Panduan Praktis Pengadaan Barang dan Jasa Bidang IT

Berbasis Open Source, Berorientasi TKDN Tinggi, Aman, Patuh Regulasi, dan Siap Audit

Onno W. Purbo
onno@indo.net.id
onno@itts.ac.id
Twitter: onnowpurbo

Institut Teknologi Tangerang Selatan
OnnoCenter
2026

Kata Pengantar

Alhamdulillahirabbil ‘alamin, puji syukur kehadirat Allah SWT karena atas rahmat dan karunia-Nya buku ini dapat disusun. Saya juga menyampaikan **terima kasih** kepada semua pihak yang telah memberi gagasan, dorongan, dan inspirasi dalam penulisan buku ini.

Dasar pemikiran utama buku ini adalah **kedaulatan bangsa Indonesia di bidang IT**. Pengadaan barang dan jasa bidang IT tidak seharusnya hanya dipahami sebagai urusan belanja teknologi, tetapi juga sebagai bagian dari upaya membangun **kemandirian nasional**, memperkuat ekosistem digital dalam negeri, dan membuka ruang agar **UMKM IT Indonesia** dapat berkiprah lebih besar. Indonesia juga perlu mengurangi ketergantungan yang berlebihan agar tidak terus-menerus **terkunci oleh vendor asing**.

Gagasan awal buku ini terpicu oleh **I Made Wiryana** dalam sebuah webinar yang diselenggarakan untuk persiapan **Cloud Regional Open Source Summit** di **UNJAYA Yogyakarta, April 2026**. Dalam forum tersebut, beliau secara tegas menekankan perlunya **Panduan Praktis Pengadaan Barang dan Jasa bidang IT** yang dapat dipakai secara nyata di lapangan.

Buku ini ditulis dengan bantuan **Artificial Intelligence (AI)**, terutama untuk membantu penyusunan dan pengembangan materi. Namun demikian, **penulis tetap bertanggung jawab sepenuhnya** atas seluruh isi buku ini. Saya juga **mohon maaf jika masih banyak kekurangan** di dalamnya.

Semoga buku ini dapat memberi manfaat dan **berkontribusi sebesar-besarnya bagi Indonesia**, khususnya dalam memperkuat kemandirian, kapasitas, dan kedaulatan digital nasional.

Jakarta, April 2026
Penulis

Executive Summary

Buku ini lahir dari satu kegelisahan strategis: **pengadaan barang dan jasa bidang IT di Indonesia masih terlalu sering diperlakukan sebagai urusan belanja teknologi, padahal sesungguhnya ia adalah instrumen pembangunan kapasitas nasional, penguatan layanan publik, dan penjagaan kedaulatan digital**. Buku ini menegaskan bahwa **objective utamanya adalah mendukung kedaulatan bangsa Indonesia dan memperkuat kemandirian bangsa Indonesia di bidang IT**, dengan cara mengurangi ketergantungan berlebihan pada vendor asing, memperluas ruang tumbuh bagi talenta serta UMKM IT nasional, dan memastikan belanja digital benar-benar berubah menjadi manfaat jangka panjang bagi negara.

Di titik inilah terlihat **kekosongan yang selama ini sangat terasa**: banyak instansi memiliki aturan umum pengadaan, tetapi belum memiliki panduan praktis yang benar-benar memahami kekhasan proyek IT—yang menyangkut *arsitektur*, integrasi, keamanan, keberlanjutan, SDM, audit, dan risiko *vendor lock-in*. Buku ini hadir untuk menutup celah itu. Ia bukan sekadar kumpulan pasal, melainkan **panduan kerja yang menghubungkan regulasi, desain sistem, KAK, spesifikasi, HPS, kontrak, keamanan, PDP, cloud, SDM, sampai kesiapan audit** dalam satu alur yang utuh dan dapat dipakai di lapangan.

Buku ini juga menempatkan *open source* pada posisi yang tepat: **bukan slogan murah-meriah, tetapi instrumen strategis** untuk mengurangi ketergantungan, memperbesar nilai lokal, memperkuat jasa dan talenta dalam negeri, serta membangun sistem yang lebih mudah dikendalikan, diintegrasikan, dan diaudit. Namun buku ini tetap realistis: tidak semua proyek harus *open source* penuh. Yang utama adalah memilih pendekatan yang **paling bermanfaat, paling aman, paling akuntabel, dan paling memperbesar kapasitas nasional**.

Titik-titik strategis yang harus dikuasai dari buku ini:

- **memahami pengadaan IT sebagai keputusan strategis, bukan sekadar transaksi belanja**;
- **menyusun KAK berbasis masalah, output, dan manfaat, bukan merek**;
- **menulis spesifikasi yang non-diskriminatif dan berorientasi fungsi/kinerja**;
- **menghitung HPS dan *total cost of ownership* secara realistis**;
- **mengunci deliverable, SLA, UAT, source code, dokumentasi, dan *exit plan***;
- **menjaga kepatuhan pada SPBE, PDP, keamanan siber, cloud, dan PSE**;
- **menguatkan SDM, sertifikasi, serta jasa lokal agar nilai belanja tetap tinggal di dalam negeri**;
- **mendorong *open source* dan TKDN tinggi secara cerdas, pragmatis, dan siap audit**.

Inti strategi agar IT bisa mandiri dan berdaulat di Indonesia adalah ini: **belanja negara harus diarahkan bukan hanya untuk membeli teknologi, tetapi untuk membangun penguasaan, kendali, interoperabilitas, keamanan, SDM, dan keberlanjutan ekosistem digital nasional**. Dengan demikian, pengadaan IT tidak berhenti pada serah terima proyek, tetapi menjadi jalan nyata menuju **Indonesia yang lebih mandiri, lebih berdaulat, dan lebih kuat secara digital**.

Daftar Isi

Kata Pengantar **2**
Executive Summary **3**
Daftar Isi **4**
Bagian I. Fondasi Strategis Pengadaan IT Nasional **14**
Bab 1. Mengapa Pengadaan IT Tidak Sama dengan Pengadaan Biasa 14
1.1 IT bukan sekadar membeli barang, tetapi membeli layanan, arsitektur, integrasi, keamanan, dan keberlanjutan 14
1.2 Mengapa banyak proyek IT gagal meski tendernya selesai 15
1.3 Risiko utama: vendor lock-in, lisensi mahal, sistem tidak terintegrasi, SDM tidak siap, dan audit bermasalah 16
a. Vendor lock-in 16
b. Lisensi mahal dan biaya tidak transparan 16
c. Sistem tidak terintegrasi 17
d. SDM tidak siap 17
e. Audit bermasalah 17
1.4 Mengapa pendekatan open source + TKDN tinggi + jasa lokal kuat menjadi penting 18
1.5 Posisi buku ini: praktis, patuh aturan, tetapi tetap pro-inovasi 19
Penutup 20
Bab 2. Kerangka Hukum Inti Pengadaan IT di Indonesia 21
2.1 Perpres PBJ terbaru dan arah kebijakannya 21
2.2 Perencanaan pengadaan sebagai fondasi keberhasilan proyek IT 22
2.3 Pelaksanaan pengadaan melalui penyedia 23
2.4 Masa transisi aturan terbaru dan implikasinya 24
2.5 Posisi katalog elektronik, toko daring, dan pengadaan digital 25
2.6 Kapan IT masuk barang, jasa lainnya, atau jasa konsultansi 26
2.7 Konsekuensi salah klasifikasi paket 28
Penutup 29
Tabel Summary. Kerangka Hukum Inti Pengadaan IT di Indonesia 30
Ringkasan 31
Bab 3. Open Source, Produk Dalam Negeri, dan TKDN Tinggi 32
3.1 Perbedaan open source, free software, dan gratis 32
3.2 Hubungan open source dengan kedaulatan digital 33
3.3 Mengapa open source bisa memperbesar porsi nilai lokal 34
3.4 TKDN, P3DN, dan belanja produk dalam negeri 35
3.5 Cara membaca “nilai lokal” di proyek IT 36
3.6 Strategi menyusun paket agar tidak didominasi lisensi asing 37
3.7 Model hybrid: perangkat komersial + stack open source + jasa lokal 38
3.8 Metrik praktis untuk “TKDN tinggi” pada proyek software dan jasa 39
3.9 Kapan open source bukan pilihan terbaik 40
3.10 Matriks keputusan: OSS penuh, hybrid, atau proprietary terbatas 41
Penutup 42
Tabel Summary. Open Source, Produk Dalam Negeri, dan TKDN Tinggi 43

Ringkasan..... 45
Bab 4. Pengadaan IT dalam Kerangka SPBE dan Transformasi Digital.....46
4.1 SPBE sebagai konteks pengadaan IT pemerintah.....46
4.2 Arsitektur SPBE nasional: proses bisnis, data, aplikasi, infrastruktur, keamanan47
4.3 Interoperabilitas data dan integrasi antar sistem.....48
4.4 Hubungan pengadaan dengan layanan publik digital..... 49
4.5 Menghindari aplikasi silo..... 50
Penutup.....51
Tabel Summary Bab 4. Pengadaan IT dalam Kerangka SPBE dan Transformasi Digital..... 52
Ringkasan..... 53
Bab 5. Proses Perencanaan Pengadaan IT dari Awal.....54
5.1 Identifikasi masalah bisnis dan kebutuhan layanan..... 54
5.2 Memetakan pemilik proses, pengguna, admin, operator, pimpinan..... 55
5.3 Menyusun business case dan manfaat terukur.....56
5.4 Analisis kondisi as-is dan target to-be.....57
5.5 Menentukan strategi: bangun baru, kembangkan, integrasi, migrasi, sewa layanan, atau adopsi OSS.....58
5.6 Menentukan tahapan proyek: cepat, bertahap, atau transformasi penuh..... 59
5.7 Manajemen risiko sejak perencanaan.....60
5.8 RUP, pemaketan, metode pemilihan, dan jadwal.....61
5.9 Kapan proyek sebaiknya dipecah menjadi beberapa paket.....62
5.10 Kesalahan klasik perencanaan pengadaan IT..... 63
Penutup..... 64
Tabel Summary. Proses Perencanaan Pengadaan IT dari Awal.....65
Ringkasan..... 67
Bab 6. Desain Sistem sebagai Dasar KAK dan Spesifikasi..... 68
6.1 Mengapa desain sistem harus muncul sebelum tender..... 68
6.2 Arsitektur bisnis..... 69
6.3 Arsitektur data..... 70
6.4 Arsitektur aplikasi..... 71
6.5 Arsitektur infrastruktur..... 72
6.6 Arsitektur keamanan..... 73
6.7 Integrasi, API, message bus, dan pertukaran data..... 74
6.8 On-premise, colocation, cloud, dan hybrid cloud..... 75
6.9 Backup, disaster recovery, high availability, dan observability..... 76
6.10 Dokumen desain minimal yang wajib ada sebelum pengadaan..... 77
Penutup..... 78
Tabel Summary. Desain Sistem sebagai Dasar KAK dan Spesifikasi..... 79
Ringkasan..... 81
Bagian III. KAK, Spesifikasi, HPS, dan Kontrak..... 82
Bab 7. Menulis KAK yang Benar untuk Pengadaan IT.....82
7.1 Fungsi KAK dalam siklus PBJ..... 82
7.2 KAK harus mulai dari masalah dan output, bukan merek..... 83

7.3 Struktur KAK yang ideal..... 84
7.4 Latar belakang dan urgensi..... 85
7.5 Tujuan umum dan tujuan khusus..... 86
7.6 Ruang lingkup dan batas pekerjaan..... 87
7.7 Output, outcome, dan indikator manfaat..... 88
7.8 Tahapan pekerjaan..... 89
7.9 Jadwal dan milestone..... 90
7.10 Resiko dan mitigasi..... 91
7.11 KAK untuk hardware..... 92
7.12 KAK untuk software..... 93
7.13 KAK untuk jasa pengembangan..... 94
7.14 KAK untuk managed service..... 95
7.15 KAK untuk cloud/SaaS..... 96
7.16 KAK untuk LMS/e-learning..... 97
7.17 KAK untuk e-commerce/marketplace..... 98
7.18 Checklist KAK yang siap tender..... 99
Penutup..... 100
Tabel Summary. Menulis KAK yang Benar untuk Pengadaan IT..... 101
Ringkasan..... 104
Bab 8. Menyusun Spesifikasi Teknis yang Tidak Diskriminatif..... 104
8.1 Prinsip spesifikasi berbasis fungsi dan kinerja..... 106
8.2 Kapan penyebutan merek dibolehkan dan bagaimana membatasi risikonya..... 107
8.3 Menulis spesifikasi server, storage, jaringan, dan endpoint..... 108
8.4 Menulis spesifikasi software dan aplikasi..... 110
8.5 Menulis spesifikasi open source: lisensi, repository, dokumentasi, kompatibilitas, audit kode..... 111
8.6 Menulis spesifikasi cloud: SLA, region, enkripsi, backup, exit plan..... 112
8.7 Menulis spesifikasi LMS..... 113
8.8 Menulis spesifikasi e-commerce..... 114
8.9 Menulis spesifikasi keamanan..... 115
8.10 Menulis spesifikasi layanan dukungan..... 116
8.11 Tingkat layanan dan parameter mutu..... 117
8.12 Checklist anti-diskriminasi..... 118
Penutup..... 119
Tabel Summary. Menyusun Spesifikasi Teknis yang Tidak Diskriminatif..... 120
Ringkasan..... 122
Bab 9. HPS, TCO, dan Biaya Nyata Proyek IT..... 123
9.1 HPS bukan sekadar harga barang..... 123
9.2 Total Cost of Ownership untuk proyek IT..... 124
9.3 Komponen CAPEX dan OPEX..... 125
9.4 Lisensi, subscription, cloud bill, implementasi, migrasi, support, training..... 126
9.5 Biaya keamanan, audit, backup, dan DR..... 127
9.6 Biaya integrasi dan perubahan organisasi..... 128
9.7 Menggunakan SBM, SBK, struktur skala upah, dan benchmark pasar..... 129

9.8 Menghitung biaya jasa SDM per peran..... 130
9.9 Menghindari HPS terlalu rendah..... 131
9.10 Model HPS untuk OSS vs proprietary..... 132
9.11 Model HPS untuk proyek multi-tahun..... 133
Penutup..... 134
Tabel Summary. HPS, TCO, dan Biaya Nyata Proyek IT..... 135
Ringkasan..... 137
Bab 10. Rancangan Kontrak, Deliverable, UAT, dan SLA..... 138
10.1 Menyelaraskan KAK, spesifikasi, HPS, dan kontrak..... 138
10.2 Kontrak berbasis barang vs berbasis layanan..... 139
10.3 Deliverable minimal proyek IT..... 140
10.4 Source code, repo, dokumentasi, akun admin, SOP, diagram, pelatihan..... 141
10.5 UAT, security test, integration test, performance test..... 142
10.6 SLA, SLO, masa garansi, masa stabilisasi..... 143
10.7 Serah terima dan knowledge transfer..... 144
10.8 Klausul exit management dan transisi vendor..... 145
10.9 Klausul kerahasiaan, hak pakai, dan hak modifikasi..... 146
10.10 Sengketa, penalti, dan remedy..... 147
Penutup..... 148
Tabel Summary. Rancangan Kontrak, Deliverable, UAT, dan SLA..... 149
Ringkasan..... 151
Bagian IV. Kepatuhan: PDP, Keamanan, Cloud, PSE, eLearning, eCommerce..... 152
Bab 11. Pelindungan Data Pribadi dalam Pengadaan IT..... 152
11.1 Data pribadi sebagai objek yang harus dipetakan sejak awal..... 152
11.2 Peran pengendali dan prosesor..... 153
11.3 Dasar pemrosesan data..... 154
11.4 Klasifikasi data..... 155
11.5 Privacy by design dan privacy by default..... 156
11.6 Klausul PDP dalam KAK dan kontrak..... 157
11.7 Data retention, penghapusan, dan akses subjek data..... 158
11.8 Insiden kebocoran data dan kewajiban respons..... 159
11.9 Checklist PDP untuk vendor..... 160
Penutup..... 160
Tabel Summary. Pelindungan Data Pribadi dalam Pengadaan IT..... 161
Ringkasan..... 162
Bab 12. Keamanan Siber, Audit, dan Ketahanan Operasional..... 163
12.1 Keamanan sebagai bagian inti spesifikasi..... 163
12.2 Kontrol akses, MFA, logging, enkripsi, hardening..... 164
12.3 Vulnerability management dan patching..... 165
12.4 SIEM, SOC, IR, dan forensik dasar..... 166
12.5 Backup, DR, dan BCP..... 167
12.6 Audit keamanan SPBE..... 168
12.7 Kematangan keamanan dan kesiapan audit..... 169
12.8 Uji keamanan sebelum go-live..... 170

12.9 Contoh klausul keamanan untuk kontrak..... 171
Penutup..... 172
Tabel Summary. Keamanan Siber, Audit, dan Ketahanan Operasional..... 173
Ringkasan..... 174
Bab 13. Cloud, Pusat Data, dan PSE.....175
13.1 Model cloud: IaaS, PaaS, SaaS..... 175
13.2 Public cloud, private cloud, hybrid cloud..... 176
13.3 Kapan cloud layak dipilih.....177
13.4 Risiko vendor lock-in pada cloud.....178
13.5 Exit plan, portabilitas, dan cadangan data..... 179
13.6 Kewajiban PSE.....180
13.7 Keandalan dan tanggung jawab penyelenggara sistem elektronik..... 181
13.8 TTE, log, bukti elektronik, dan tata kelola layanan elektronik..... 182
13.9 Checklist due diligence penyedia cloud/SaaS.....183
Penutup..... 184
Tabel Summary. Cloud, Pusat Data, dan PSE..... 185
Ringkasan..... 186
Bab 14. Pengadaan eLearning / LMS / Pendidikan Digital..... 187
14.1 Ciri khusus pengadaan LMS..... 187
14.2 Integrasi dengan sistem akademik.....188
14.3 Konten, video, kuis, ujian, bank soal, analitik belajar.....189
14.4 Privasi mahasiswa, dosen, dan peserta.....190
14.5 Aksesibilitas, mobile access, dan beban bandwidth..... 191
14.6 Proctoring dan resiko etika.....192
14.7 LMS open source vs proprietary..... 193
14.8 Model operasi dan dukungan helpdesk.....194
Penutup..... 195
Tabel Summary. Pengadaan eLearning / LMS / Pendidikan Digital..... 196
Ringkasan..... 197
Bab 15. Pengadaan eCommerce / Marketplace / PMSE..... 198
15.1 Ciri khusus sistem e-commerce..... 198
15.2 Perdagangan melalui Sistem Elektronik.....199
15.3 Perlindungan konsumen dan tata kelola pelaku usaha.....200
15.4 Pembayaran digital dan integrasi pihak ketiga.....201
15.5 Keamanan transaksi, log, dan fraud control.....202
15.6 Persyaratan bagi marketplace, portal penjualan, dan social commerce..... 203
15.7 KAK untuk e-commerce pemerintah/BLUD/BUMD..... 204
15.8 Checklist kepatuhan PMSE.....205
Penutup.....206
Tabel Summary. Pengadaan eCommerce / Marketplace / PMSE.....207
Ringkasan..... 209
Bagian V. SDM, Sertifikasi, Honor, Gaji, dan Fee..... 210
Bab 16. Menentukan Kebutuhan SDM dalam Proyek IT.....210
16.1 Peran inti proyek IT: PM, BA, system analyst, architect, developer, QA, security,

DevOps, support..... 210
16.2 Peran minimum vs peran ideal..... 211
16.3 Kapan satu orang boleh merangkap..... 212
16.4 Kualifikasi pendidikan vs kompetensi nyata..... 213
16.5 Pengalaman minimum yang masuk akal..... 214
16.6 Menilai portofolio, bukan hanya CV..... 215
Penutup..... 216
Tabel Summary. Menentukan Kebutuhan SDM dalam Proyek IT..... 217
Ringkasan..... 218
Bab 17. Sertifikat, SKKNI, dan Kualitas SDM..... 219
17.1 Mengapa sertifikat harus dikaitkan dengan peran, bukan formalitas..... 219
17.2 BNSP, LSP, dan SKKNI..... 220
17.3 Sertifikasi untuk keamanan, cloud, data, audit, dan pengembangan..... 221
17.4 Cara menulis syarat sertifikasi yang adil dan relevan..... 222
17.5 Sertifikasi nasional vs internasional..... 223
17.6 Matriks jabatan–kompetensi–sertifikat–pengalaman..... 224
17.7 Model evaluasi kualitas personel inti..... 225
Penutup..... 225
Tabel Summary. Sertifikat, SKKNI, dan Kualitas SDM..... 226
Ringkasan..... 227
Bab 18. Honor, Gaji, Fee, dan Biaya SDM..... 228
18.1 Membedakan gaji, honorarium, fee, tarif harian, tarif bulanan, dan biaya tenaga ahli..... 228
18.2 Struktur dan skala upah..... 229
18.3 Upah minimum vs benchmark pasar..... 230
18.4 SBM/SBK untuk instansi pemerintah..... 231
18.5 Benchmark pasar IT untuk perhitungan HPS..... 232
18.6 Menyusun rate card per peran..... 233
18.7 Produktivitas, level senioritas, dan kompleksitas..... 234
18.8 Pajak atas penghasilan jasa/honorarium..... 235
18.9 Kesalahan umum: fee terlalu murah, personel fiktif, atau senioritas semu..... 236
18.10 Template perhitungan biaya SDM proyek IT..... 237
Penutup..... 238
Tabel Summary. Honor, Gaji, Fee, dan Biaya SDM..... 239
Ringkasan..... 241
Bagian VI. Panduan Praktis per Jenis Pengadaan..... 242
Bab 19. Pengadaan Hardware, Jaringan, dan Infrastruktur..... 242
19.1 Server, storage, endpoint, UPS, rack, firewall, switch, AP..... 242
19.2 Sertifikasi perangkat dan kompatibilitas..... 243
19.3 Redundansi dan kapasitas..... 244
19.4 Support, warranty, spare part, dan lifecycle..... 245
Penutup..... 246
Tabel Summary. Pengadaan Hardware, Jaringan, dan Infrastruktur..... 247
Ringkasan..... 248

Bab 20. Pengadaan Software dan Pengembangan Aplikasi....249
20.1 Build vs buy vs adopt OSS....249
20.2 Web app, mobile app, ERP, middleware, API....250
20.3 Source code, repo, CI/CD, dokumentasi....252
20.4 UAT, pentest, dan acceptance....254
Penutup....255
Tabel Summary. Pengadaan Software dan Pengembangan Aplikasi....256
Ringkasan....257
Bab 21. Pengadaan Managed Service, SOC, dan Operasi TI....258
21.1 Helpdesk, NOC, SOC, monitoring, backup, patching....258
21.2 SLA operasional....259
21.3 Shift, coverage, eskalasi, dan pelaporan....260
21.4 Bukti layanan dan audit trail....261
Penutup....262
Tabel Summary. Pengadaan Managed Service, SOC, dan Operasi TI....263
Ringkasan....264
Bab 22. Tata Kelola Proyek, Metode Kerja, dan Pengendalian Delivery....265
22.1 Waterfall vs agile vs hybrid....265
22.2 Sprint, backlog, change request, governance board....267
22.3 Dokumentasi rapat, status report, risk register, issue log....269
22.4 Menjaga proyek tetap on-time, on-budget, dan on-scope....271
Penutup....272
Tabel Summary. Tata Kelola Proyek, Metode Kerja, dan Pengendalian Delivery....273
Ringkasan....274
Bagian VII. Template, Checklist, dan Lampiran Siap Pakai....275
Bab 23. Template KAK Pengadaan IT....275
23.1 Template KAK hardware....275
23.2 Template KAK software....277
23.3 Template KAK cloud/SaaS....278
23.4 Template KAK LMS....279
23.5 Template KAK e-commerce....280
23.6 Template KAK managed service....281
Ringkasan singkat....282
Bab 24. Template Spesifikasi Teknis....283
24.1 Template spesifikasi server/jaringan....283
24.2 Template spesifikasi aplikasi....285
24.3 Template spesifikasi keamanan....286
24.4 Template spesifikasi cloud....287
24.5 Template spesifikasi layanan SDM....288
Ringkasan singkat....289
Bab 25. Template HPS dan Biaya SDM....290
25.1 Template HPS barang....290
25.2 Template HPS jasa....292
25.3 Template TCO 3 tahun dan 5 tahun....293

25.4 Template rate card SDM....294
25.5 Template pembanding pasar....295
Ringkasan singkat....296
Catatan pakai cepat....296
Bab 26. Template Evaluasi dan Due Diligence Vendor....297
26.1 Checklist legal vendor....297
26.2 Checklist teknis....298
26.3 Checklist keamanan....299
26.4 Checklist PSE/PDP....300
26.5 Checklist kapasitas SDM....301
26.6 Checklist keberlanjutan layanan....302
Ringkasan singkat....303
Catatan pakai cepat....303
Bab 27. Checklist Audit dan Kesiapan Serah Terima....304
27.1 Checklist dokumen....304
27.2 Checklist keamanan....305
27.3 Checklist hak akses....306
27.4 Checklist source code....307
27.5 Checklist backup dan DR....308
27.6 Checklist pelatihan dan SOP....309
27.7 Checklist berita acara....310
Ringkasan singkat....311
Catatan pakai cepat....311
Lampiran A. Matriks Regulasi: aturan – pasal/tema – relevansi untuk pengadaan IT 312
Ringkasan pakai cepat....316
Catatan pakai....317
Lampiran B. Matriks Open Source – Manfaat – Risiko – Kontrol....318
Ringkasan pakai cepat....322
Catatan pakai....323
Lampiran C. Matriks TKDN/P3DN untuk Proyek IT....324
Matriks baca cepat: komponen – peluang TKDN/P3DN – strategi paket....328
Indikator praktis P3DN untuk proyek IT....329
Catatan pakai....330
Ringkasan singkat....331
Lampiran D. Matriks Peran SDM – Sertifikasi – Pengalaman – Output....332
Matriks pakai cepat: peran – bukti utama yang paling meyakinkan....336
Contoh cara membaca matriks ini....337
Catatan pakai....338
Ringkasan singkat....338
Lampiran E. Contoh Perhitungan HPS Proyek Software Open Source....339
Asumsi proyek contoh....339
Struktur HPS proyek OSS: komponen utama....340
Contoh rate card SDM proyek OSS....341
Contoh komponen non-SDM....342

Contoh rekap HPS implementasi awal OSS..... 343
Contoh TCO 3 tahun untuk proyek OSS..... 344
Perbandingan ringkas: OSS vs proprietary..... 345
Template cepat isi HPS proyek OSS..... 346
Ringkasan singkat..... 347
Catatan pakai..... 347
Lampiran F. Contoh Perhitungan HPS Managed Service..... 348
Contoh kasus sederhana..... 348
1. Hitung biaya orangnya..... 348
a. Support Engineer..... 348
b. Ops / System Administrator..... 349
c. Service Lead / Koordinator..... 349
Total biaya SDM..... 349
2. Tambahkan biaya pendukung..... 350
a. Transisi awal layanan..... 350
b. Laporan bulanan dan review SLA..... 350
c. Update SOP / dokumentasi..... 350
d. Cadangan operasional kecil..... 350
Total biaya pendukung..... 350
3. Hitung total sebelum pajak..... 350
4. Tambahkan PPN 11%..... 351
5. Total HPS..... 351
6. Kenapa bisa mahal padahal tidak beli barang?..... 351
7. Hal yang sering lupa dihitung..... 351
8. Rumus paling mudah..... 352
9. Format cepat yang bisa dipakai..... 352
Inti paling sederhananya..... 352
Lampiran G. Contoh SLA dan UAT..... 353
A. Contoh SLA sederhana..... 353
1. Contoh SLA untuk aplikasi internal..... 353
2. Arti level gangguan..... 354
3. Contoh SLA untuk managed service..... 354
4. Bukti SLA..... 354
B. Contoh UAT sederhana..... 355
1. Tujuan UAT..... 355
2. Contoh UAT sederhana untuk aplikasi layanan internal..... 355
3. Format UAT paling sederhana..... 356
4. Contoh cara menulis hasil UAT..... 356
5. Kapan UAT dinyatakan selesai..... 356
6. Hal penting yang sering dilupakan saat UAT..... 357
C. Contoh kalimat siap pakai untuk buku / KAK / kontrak..... 357
Contoh kalimat SLA..... 357
Contoh kalimat UAT..... 357
D. Inti paling sederhananya..... 357

Lampiran H. Checklist PDP dan Keamanan..... 358
Ringkasan pakai cepat..... 361
Catatan pakai..... 361
Lampiran I. Checklist Due Diligence Cloud / PSE..... 362
Ringkasan pakai cepat..... 364
Penilaian sederhana..... 364
Glossary..... 365
Referensi..... 373
Tentang Penulis..... 376

Bagian I. Fondasi Strategis Pengadaan IT Nasional

Bab 1. Mengapa Pengadaan IT Tidak Sama dengan Pengadaan Biasa

1.1 IT bukan sekadar membeli barang, tetapi membeli layanan, arsitektur, integrasi, keamanan, dan keberlanjutan

Banyak orang masih menganggap pengadaan IT sama seperti membeli meja, kendaraan, atau perlengkapan kantor: tentukan spesifikasi, bandingkan harga, pilih penyedia, lalu selesai. Cara pandang ini keliru. Dalam proyek IT, yang dibeli hampir tidak pernah hanya barang. Organisasi sesungguhnya sedang membeli **kemampuan layanan**, **arsitektur sistem**, **integrasi data**, **keamanan**, **dukungan operasional**, dan **keberlanjutan jangka panjang**.

Server tanpa desain sistem yang tepat hanya menjadi perangkat mahal. Aplikasi tanpa integrasi hanya menciptakan silo. Lisensi tanpa SDM yang mampu mengelola akan berubah menjadi beban tahunan.

Dalam kerangka *SPBE*, teknologi juga tidak berdiri sendiri. Tata kelola digital pemerintah harus memastikan keterpaduan antara proses bisnis, data dan informasi, aplikasi, infrastruktur, serta keamanan. Karena itu, pengadaan IT seharusnya dipahami sebagai bagian dari **arsitektur layanan**, bukan sekadar daftar perangkat atau fitur.

Hal yang harus dipastikan sejak awal:

- layanan apa yang sebenarnya ingin dibangun;
- proses bisnis mana yang ingin diperbaiki;
- data apa yang harus saling terhubung;
- siapa operator dan penanggung jawabnya;
- bagaimana mekanisme pengamanannya;
- bagaimana sistem akan dipelihara setelah proyek selesai.

Jadi, dalam pengadaan IT, organisasi tidak sedang membeli benda mati. Organisasi sedang membeli **kemampuan digital** yang harus **hidup, terhubung, aman, digunakan, dipelihara, dan dapat diaudit**. (JDIH LKPP)

1.2 Mengapa banyak proyek IT gagal meski tendernya selesai

Secara administratif, banyak proyek IT terlihat sukses. Kontrak ditandatangani, penyedia dipilih, barang datang, aplikasi di-*deploy*, dan dokumen serah terima tersedia. Namun beberapa bulan kemudian, sistem jarang dipakai, integrasi tidak berjalan, data menjadi kacau, pengguna mengeluh, biaya tambahan muncul, dan auditor mulai bertanya.

Ini menunjukkan satu hal penting: **tender yang selesai tidak otomatis berarti transformasi digital berhasil**.

Beberapa penyebab yang paling sering muncul adalah sebagai berikut:

- **Kebutuhan ditulis terlalu teknis, tetapi tidak berbasis hasil.**
  Organisasi tahu ingin membeli server, aplikasi, atau cloud, tetapi belum jelas layanan apa yang harus membaik, waktu proses apa yang harus turun, risiko apa yang harus berkurang, dan output apa yang benar-benar harus bisa dipakai.
- **Desain sistem datang terlambat.**
  Kadang desain baru dipikirkan saat tender sudah berjalan. Akibatnya, penyedia menafsirkan kebutuhan secara sempit, sementara pengguna berharap hasil yang jauh lebih luas.
- **Mutu deliverable lemah.**
  Dalam proyek IT, kegagalan sering tidak langsung terlihat di hari pertama. Masalah muncul perlahan: performa menurun, integrasi rapuh, dokumentasi minim, ketergantungan pada satu orang terlalu tinggi, atau perubahan kecil justru memicu gangguan besar.
- **Tidak siap menghadapi fase pasca-implementasi.**
  Sistem yang sudah berjalan tetap membutuhkan *patching*, monitoring, backup, pelatihan, *incident response*, pembaruan keamanan, dan penyesuaian kebutuhan. Jika ini diabaikan, proyek yang tampak selesai akan cepat menurun kualitasnya.

Karena itu, ukuran keberhasilan pengadaan IT tidak boleh berhenti pada kalimat “paket telah dibelanjakan.” Ukuran yang lebih tepat adalah:

- apakah sistem benar-benar dipakai;
- apakah sistem aman;
- apakah sistem terintegrasi;
- apakah dokumentasinya memadai;
- apakah sistem dapat dipelihara;
- apakah sistem memberi manfaat nyata.

Pengadaan IT yang baik harus menjembatani **prosedur PBJ** dengan **keberhasilan layanan**. (JDIH LKPP)

1.3 Risiko utama: vendor lock-in, lisensi mahal, sistem tidak terintegrasi, SDM tidak siap, dan audit bermasalah

Pengadaan IT memiliki sejumlah risiko strategis yang sering muncul berulang.

a. *Vendor lock-in*

Risiko pertama adalah ketergantungan berlebihan pada satu penyedia, satu produk, satu format data, atau satu ekosistem teknologi. Kondisi ini membuat organisasi sulit bermigrasi, sulit bernegosiasi, dan sulit membangun kapasitas internal.

Dampaknya tidak hanya terasa di awal, tetapi juga membesar dalam tahun-tahun berikutnya. Organisasi memang mungkin membeli kenyamanan jangka pendek, tetapi kehilangan fleksibilitas jangka panjang.

Karena itu, isu seperti berikut harus dianggap strategis:

- kontrol atas *source code*;
- kepemilikan deliverable;
- interoperabilitas;
- penguatan jasa lokal.

b. Lisensi mahal dan biaya tidak transparan

Dalam IT, harga pembelian awal hampir tidak pernah mencerminkan biaya sesungguhnya. Masih ada biaya lain yang sering muncul kemudian, seperti:

- implementasi;
- migrasi;
- integrasi;
- pelatihan;
- peningkatan kapasitas;
- pemeliharaan;
- pembaruan versi;
- support;
- storage dan backup;
- biaya keluar saat sistem akan diganti.

Karena itu, pendekatan yang benar adalah melihat **whole-life costing** atau *total cost of ownership*, bukan hanya harga awal.

c. Sistem tidak terintegrasi

SPBE menekankan pentingnya interoperabilitas antar proses bisnis dan antar sistem elektronik. Jika aplikasi dibangun secara sektoral tanpa arsitektur bersama, hasilnya adalah aplikasi silo: berjalan sendiri-sendiri, tetapi tidak saling berbicara.

Dampaknya bisa berupa:

- duplikasi input;
- data tidak konsisten;
- pelayanan lambat;
- biaya integrasi yang meledak di belakang hari.

d. SDM tidak siap

Banyak proyek IT gagal bukan karena teknologinya buruk, tetapi karena organisasi belum siap menjalankannya. Admin, operator, pengguna, dan pengambil keputusan belum cukup kuat kapasitasnya. Akibatnya, sistem lebih banyak “dijalankan” oleh vendor daripada dimiliki oleh institusi.

Jika ini terjadi, serah terima proyek hanya menjadi formalitas.

e. Audit bermasalah

Dalam konteks *SPBE* dan *PDP*, proyek IT harus sejak awal memikirkan jejak bukti dan kepatuhan. Jika tidak ada logging, dokumentasi arsitektur, pengelolaan akses, dokumentasi perubahan, dan bukti uji terima yang memadai, maka proyek akan lemah saat diperiksa auditor atau saat terjadi insiden.

Pelajaran pentingnya: dalam pengadaan IT, risiko tidak selalu muncul sebagai kegagalan besar yang langsung terlihat. Sering kali resikonya muncul dalam bentuk kecil, seperti:

- dokumen yang kurang lengkap;
- hak akses yang tidak jelas;
- integrasi yang terus ditunda;
- pelatihan yang minim;
- kontrak yang tidak tegas.

Namun akumulasi masalah kecil itulah yang pada akhirnya membuat proyek menjadi **mahal, rapuh, dan sulit dipertanggungjawabkan**. (JDIH LKPP)

1.4 Mengapa pendekatan open source + TKDN tinggi + jasa lokal kuat menjadi penting

Pendekatan **open source + TKDN tinggi + jasa lokal kuat** penting bukan karena alasan romantisme teknologi, melainkan karena menjawab tiga persoalan sekaligus:

- ketergantungan pada pihak luar;
- kebocoran nilai ekonomi;
- lemahnya kapasitas internal.

Pendekatan ini tidak berarti semua proyek IT harus diwajibkan memakai *open source*. Namun secara strategi pengadaan, pendekatan ini selaras dengan prinsip pengadaan yang:

- non-diskriminatif;
- mendukung interoperabilitas;
- lebih mudah mengendalikan deliverable;
- membuka ruang bagi jasa integrasi, migrasi, dukungan, pelatihan, dan dokumentasi yang dapat dikerjakan pelaku lokal.

Dari sisi kebijakan, arah keberpihakan pada produk dalam negeri juga semakin kuat. Dalam proyek digital, nilai lokal tidak selalu paling besar pada perangkat fisik. Justru sering kali nilai terbesar ada pada:

- desain;
- integrasi;
- kustomisasi;
- migrasi data;
- support;
- operasi;
- pelatihan;
- dokumentasi;
- audit;
- pengamanan.

Di sinilah *open source* menjadi strategis. Dengan mengurangi dominasi biaya lisensi asing, lebih banyak nilai ekonomi bisa dialihkan ke pekerjaan yang dikerjakan talenta dan perusahaan lokal.

Pendekatan ini juga lebih sehat untuk jangka panjang. Jika organisasi memakai *stack* yang lebih terbuka, format data yang lebih terbuka, serta kontrak yang menjamin dokumentasi dan transfer pengetahuan, maka pergantian penyedia akan lebih mudah dikelola. Organisasi tidak mudah disandera oleh lisensi, format tertutup, atau ketergantungan personal.

Namun perlu ditegaskan, pendekatan ini **bukan berarti anti-proprietary**. Dalam kasus tertentu, solusi tertutup tetap dapat dibenarkan, misalnya karena:

- kebutuhan keamanan tertentu;

- kompatibilitas dengan sistem lain;
- kebutuhan industri yang sangat khusus;
- pertimbangan waktu.

Prinsip utamanya sederhana: pilih solusi yang **paling bermanfaat**, **paling dapat dipertanggungjawabkan**, **paling aman**, **paling mendukung interoperabilitas**, dan **paling memperbesar kapasitas nasional**. (JDIH LKPP)

1.5 Posisi buku ini: praktis, patuh aturan, tetapi tetap pro-inovasi

Buku ini sebaiknya diposisikan secara tegas sebagai panduan kerja strategis bagi:

- PPK;
- PA/KPA;
- UKPBJ;
- tim teknis;
- auditor internal;
- pimpinan transformasi digital;
- perguruan tinggi;
- BLUD;
- BUMD;
- penyedia lokal.

Fungsi buku ini bukan sebagai kumpulan pasal, melainkan sebagai alat bantu agar pembaca bisa menghubungkan:

- regulasi;
- desain sistem;
- kebutuhan layanan;
- keamanan;
- SDM;
- audit;

ke dalam satu alur yang utuh dan masuk akal.

Karena itu, nada dasar buku ini sebaiknya:

- **patuh aturan**, tetapi tidak birokratis;
- **kritis**, tetapi tidak sinis;
- **pro-inovasi**, tetapi tidak naif.

Pengadaan IT memang membutuhkan keberanian untuk mendorong:

- *output-based procurement*;
- spesifikasi non-merek;
- penguatan deliverable;
- pengelolaan risiko sejak awal;
- evaluasi berbasis manfaat.

Namun semua itu tetap harus berpijak pada regulasi PBJ, kerangka *SPBE*, kewajiban keamanan, dan kepatuhan terhadap pelindungan data pribadi.

Buku ini juga perlu jujur bahwa tidak ada satu resep yang cocok untuk semua proyek IT. Tidak semua proyek tepat memakai *open source* penuh. Tidak semua persoalan selesai dengan TKDN. Dan tidak semua inovasi layak dipilih hanya karena tampak modern.

Yang dibutuhkan pembaca adalah **kerangka berpikir yang tajam**, khususnya untuk menjawab pertanyaan berikut:

- bagaimana menilai kebutuhan;
- bagaimana memilih strategi;
- bagaimana menulis KAK;
- bagaimana menyusun spesifikasi;
- bagaimana menghitung HPS;
- bagaimana menilai SDM;
- bagaimana mengunci deliverable;
- bagaimana menjaga kepatuhan;
- bagaimana menghindari jebakan yang sama berulang kali.

Pada akhirnya, posisi buku ini adalah sebagai **jembatan**:

- antara hukum dan praktik;
- antara pengadaan dan arsitektur;
- antara inovasi dan akuntabilitas;
- antara efisiensi anggaran dan kedaulatan digital.

Penutup

Bab ini menanamkan satu gagasan utama sejak awal: **pengadaan IT adalah keputusan strategis tentang masa depan organisasi, bukan sekadar transaksi belanja**.

Jika sifat pengadaan IT dipahami secara keliru, hasilnya adalah proyek yang mungkin selesai secara administratif, tetapi gagal memberi manfaat. Sebaliknya, jika pengadaan dibangun di atas arsitektur yang jelas, interoperabilitas, keamanan, penguatan SDM, deliverable yang tegas, serta keberpihakan yang cerdas pada *open source*, jasa lokal, dan produk dalam negeri, maka belanja digital dapat benar-benar berubah menjadi **kapasitas nasional** dan **layanan publik yang lebih kuat**. (JDIH LKPP)

Bab 2. Kerangka Hukum Inti Pengadaan IT di Indonesia

2.1 Perpres PBJ terbaru dan arah kebijakannya

Fondasi hukum pengadaan barang/jasa pemerintah saat ini tetap bertumpu pada **Perpres 16 Tahun 2018**, yang kemudian diubah melalui **Perpres 12 Tahun 2021** dan kembali diperbarui melalui **Perpres 46 Tahun 2025**. Perubahan terakhir ini penting karena menunjukkan arah kebijakan yang semakin tegas: mendorong penggunaan produk dalam negeri, mempercepat pelaksanaan PBJ, dan mengoptimalkan manfaat belanja pemerintah. Untuk konteks buku ini, artinya jelas: pengadaan IT tidak lagi layak dipahami sekadar sebagai pembelian perangkat atau aplikasi, melainkan sebagai instrumen kebijakan yang harus menghasilkan manfaat operasional, ekonomi, dan kelembagaan sekaligus. (JDIH LKPP)

Dalam naskah konsolidasi Perpres 46 Tahun 2025, **RUP** tetap diposisikan sebagai daftar rencana PBJ K/L/PD, sedangkan **lokapasar (*e-marketplace*)** dipahami sebagai pasar elektronik untuk memenuhi kebutuhan barang/jasa pemerintah. Bagi pengadaan IT, ini berarti sejak tahap perencanaan organisasi sudah harus menentukan jalur pengadaannya secara tepat, misalnya:

- melalui katalog;
- melalui toko daring;
- melalui pemilihan penyedia biasa;
- atau melalui kombinasi beberapa mekanisme.

Pengadaan digital dengan demikian bukan lagi lapisan tambahan, tetapi sudah menjadi bagian inti dari rezim PBJ modern. (JDIH LKPP)

Arah kebijakan ini juga harus dibaca bersama kerangka **SPBE**, yang menuntut tata kelola digital yang terpadu, efisien, dan berkelanjutan. Dalam SPBE, arsitektur digital dipahami sebagai keterhubungan antara:

- proses bisnis;
- data;
- aplikasi;
- infrastruktur;
- keamanan.

Karena itu, hukum PBJ dan arsitektur digital tidak boleh berjalan sendiri-sendiri. Jika aturan PBJ dibaca tanpa memahami arsitektur layanan, proyek IT akan cenderung formalistik. Sebaliknya, jika arsitektur digital dibangun tanpa disiplin PBJ, proyek akan rapuh dari sisi akuntabilitas.

2.2 Perencanaan pengadaan sebagai fondasi keberhasilan proyek IT

Secara normatif, pengadaan yang sehat selalu dimulai dari **perencanaan**, bukan dari memilih vendor. **Peraturan LKPP Nomor 11 Tahun 2021** menegaskan bahwa pedoman perencanaan pengadaan dibuat untuk menyamakan pemahaman, menjadi rujukan pelaksana, dan memastikan pengadaan selaras dengan tujuan, kebijakan, prinsip, serta etika PBJ. Dalam aturan yang sama juga ditegaskan bahwa **PPK** bertugas menyusun perencanaan pengadaan sesuai kebutuhan K/L/PD sebagaimana tercantum dalam dokumen kerja dan anggaran. (JDIH LKPP)

Untuk bidang IT, pesan ini jauh lebih penting daripada yang tampak di permukaan. Banyak proyek digital gagal bukan karena pemilihannya salah, tetapi karena kebutuhan sejak awal tidak pernah dirumuskan dengan jelas. Organisasi sering belum tegas menjawab apakah yang dibutuhkan sesungguhnya adalah:

- server;
- aplikasi;
- integrasi;
- migrasi;
- *managed service*;
- pelatihan;
- atau kombinasi dari semuanya.

Dalam proyek IT, kesalahan di tahap perencanaan hampir selalu terbawa ke tahap kontrak, lalu meledak saat implementasi. Karena itu, perencanaan harus mampu memetakan secara utuh:

- masalah bisnis yang ingin diselesaikan;
- layanan yang ingin diperbaiki;
- kondisi *as-is*;
- target *to-be*;
- risiko;
- ketergantungan pada sistem lama;
- kebutuhan data;
- kebutuhan keamanan;
- kapasitas SDM internal.

Pendekatan ini juga sejalan dengan pandangan pengadaan modern yang menekankan bahwa biaya tidak boleh dilihat hanya dari harga awal. Literatur pengadaan menunjukkan bahwa biaya sepanjang siklus hidup sistem seringkali jauh lebih menentukan, termasuk biaya:

- operasional;
- pemeliharaan;
- perubahan ruang lingkup;
- kebutuhan keterampilan tim.

Untuk proyek IT, cara berpikir seperti ini sangat relevan karena beban terbesar justru sering muncul setelah kontrak mulai berjalan.

2.3 Pelaksanaan pengadaan melalui penyedia

Setelah perencanaan matang, organisasi masuk ke tahap pelaksanaan pengadaan melalui penyedia. **Peraturan LKPP Nomor 12 Tahun 2021** menjadi pedoman utama untuk tahap ini, dan statusnya telah diperbarui melalui **Peraturan LKPP Nomor 4 Tahun 2024**. Artinya, secara praktis pelaku pengadaan tidak cukup hanya membaca teks tahun 2021, tetapi harus memahami bahwa rezim tersebut sudah mengalami pembaruan. (JDIH LKPP)

Dalam Perpres konsolidasi, pengadaan melalui penyedia dipahami sebagai cara memperoleh barang/jasa yang disediakan pelaku usaha berdasarkan kontrak. Bagi proyek IT, skema ini sangat lazim karena banyak kebutuhan digital memang dipenuhi melalui penyedia, seperti:

- perangkat;
- aplikasi;
- integrasi sistem;
- layanan cloud;
- jasa pengembangan;
- audit keamanan;
- dukungan operasional.

Tantangan khasnya adalah bahwa dalam proyek IT, yang dibeli sering merupakan gabungan antara:

- barang berwujud;
- barang tak berwujud;
- jasa implementasi;
- jasa dukungan;
- jasa konsultansi.

Karena itu, dokumen pemilihan harus sangat presisi dalam membaca substansi pekerjaannya. (JDIH LKPP)

Perpres juga memberi ruang bagi berbagai metode evaluasi, termasuk:

- sistem nilai;
- penilaian biaya selama umur ekonomis;
- harga terendah.

Untuk proyek IT, ruang ini sangat penting. Pengadaan aplikasi, integrasi sistem, atau layanan digital yang kompleks sering kali tidak layak ditentukan hanya berdasarkan harga terendah. Kualitas teknis, keamanan, arsitektur, dokumentasi, dan kapasitas tim pelaksana sangat menentukan keberhasilan hasil akhir. Jadi, hukum memang memberi ruang, tetapi hasilnya tetap sangat ditentukan oleh kualitas perencanaan dan metode pemilihan yang digunakan. (JDIH LKPP)

2.4 Masa transisi aturan terbaru dan implikasinya

Salah satu persoalan nyata di lapangan adalah bahwa pengadaan sering berlangsung dalam masa transisi regulasi. Praktik PBJ tidak berhenti hanya karena aturan baru diterbitkan. Organisasi tetap harus menyesuaikan:

- prosedur;
- aplikasi;
- template;
- pola kerja;
- kebiasaan administratif.

Dalam konteks saat ini, **Perpres 46 Tahun 2025** menjadi titik penting karena kembali mengubah rezim PBJ yang sebelumnya telah diperbarui melalui **Perpres 12 Tahun 2021**. (JDIH LKPP)

Bagi pengadaan IT, implikasinya cukup besar. Beberapa hal yang harus diwaspadai antara lain:

- penggunaan template lama yang sudah tidak sesuai dengan arah kebijakan baru;
- perlunya penyesuaian terhadap percepatan PBJ;
- perlunya perhatian lebih kuat terhadap penggunaan produk dalam negeri;
- perubahan operasional pada platform pengadaan digital.

Contohnya, pada tahun 2025 LKPP menerbitkan surat edaran tentang penonaktifan **Katalog Elektronik Versi 5** untuk etalase tertentu. Ini menunjukkan bahwa aspek teknis platform juga dapat berubah dan berdampak langsung pada praktik pengadaan pemerintah. (JDIH LKPP)

Masa transisi juga menuntut pejabat pengadaan, PPK, tim teknis, dan auditor internal untuk disiplin memeriksa status aturan:

- mana yang masih berlaku;
- mana yang sudah diubah;
- mana yang dicabut;
- mana yang hanya berfungsi sebagai pedoman teknis.

Dalam proyek IT, kelalaian di titik ini bisa memunculkan dua masalah sekaligus: kesalahan prosedural dan kegagalan memahami kebutuhan digital yang terus berkembang. Karena itu, literasi regulasi dalam pengadaan IT bukan urusan administratif semata, melainkan bagian dari manajemen risiko proyek. (JDIH LKPP)

2.5 Posisi katalog elektronik, toko daring, dan pengadaan digital

Dalam rezim PBJ saat ini, **katalog elektronik**, **toko daring**, dan ekosistem pengadaan digital harus dipahami sebagai instrumen utama, bukan sekadar pilihan tambahan. Dalam Perpres konsolidasi, katalog elektronik dijelaskan sebagai platform yang memuat informasi mengenai:

- barang/jasa;
- harga;
- penyedia atau pelaksana swakelola;
- informasi lain yang relevan.

Pengelolaannya dapat dilakukan oleh LKPP maupun oleh K/L/PD/instansi lainnya. Bahkan, aturan tersebut juga membuka ruang untuk penghentian transaksi *e-purchasing* atas produk impor yang memiliki substitusi produk dalam negeri. Ini sangat relevan dengan arah kebijakan TKDN tinggi dan keberpihakan pada nilai lokal. (JDIH LKPP)

Sementara itu, **Peraturan LKPP Nomor 9 Tahun 2021** menegaskan bahwa toko daring dan katalog elektronik dimaksudkan agar pengadaan menjadi:

- cepat;
- mudah;
- transparan;
- tercatat secara elektronik.

Namun aturan yang sama juga menjelaskan bahwa barang/jasa di toko daring idealnya memiliki karakter:

- standar atau dapat distandarkan;
- berisiko rendah;
- harga yang sudah terbentuk di pasar.

Pelajaran pentingnya untuk pengadaan IT adalah: **tidak semua kebutuhan IT cocok masuk toko daring**. Yang mungkin cocok adalah kebutuhan yang relatif standar, seperti:

- laptop umum;
- periferal standar;
- lisensi yang sangat baku.

Sebaliknya, kebutuhan yang kompleks seperti sistem informasi dengan integrasi mendalam, migrasi data, *customization*, dan pengujian keamanan tidak layak diperlakukan seperti pembelian komoditas biasa. (JDIH LKPP)

Karena itu, katalog dan toko daring perlu dibaca secara strategis. Keduanya sangat bermanfaat untuk kebutuhan yang:

- standar;
- rutin;
- mudah dibandingkan.

Tetapi untuk paket yang kompleks, keduanya hanyalah salah satu instrumen, bukan jawaban tunggal. Kesalahan yang sering terjadi adalah memaksakan kebutuhan yang rumit ke kanal yang terlalu sederhana, lalu berharap hasilnya tetap matang. Akibatnya, kontrak mungkin selesai secara formal, tetapi persoalan layanan tidak pernah benar-benar terselesaikan. (JDIH LKPP)

2.6 Kapan IT masuk barang, jasa lainnya, atau jasa konsultansi

Ini adalah salah satu titik paling kritis dalam pengadaan IT. Secara hukum, **barang** dalam Perpres mencakup setiap benda, baik:

- berwujud maupun tidak berwujud;
- bergerak maupun tidak bergerak;
- yang dapat diperdagangkan, dipakai, digunakan, atau dimanfaatkan.

Dalam dunia digital, rumusan ini penting karena banyak objek pengadaan tidak lagi berupa barang fisik. Misalnya:

- lisensi;
- *subscription*;
- software;
- dataset;
- aset digital tertentu.

Semua itu dapat masuk kategori barang, tergantung pada substansi kebutuhannya. (JDIH LKPP)

Namun tidak semua pengadaan IT tepat diklasifikasikan sebagai barang. Jika inti kebutuhannya lebih berupa implementasi, integrasi, konfigurasi, dukungan operasional, *managed service*, *helpdesk*, *deployment*, pengujian, atau pemeliharaan, maka secara substansi sering lebih dekat ke **jasa lainnya**. Dalam Perpres, barang/jasa lainnya dapat dievaluasi dengan:

- sistem nilai;
- biaya selama umur ekonomis;
- harga terendah.

Artinya, hukum tetap memberi ruang untuk pemilihan berbasis mutu jika sifat pekerjaannya memang menuntut itu. (JDIH LKPP)

Sebaliknya, bila inti kebutuhan adalah:

- kajian;
- perencanaan;
- penyusunan arsitektur;
- audit;
- *assessment*;
- desain detail;
- blueprint;
- advis profesional berbasis tenaga ahli,

maka paket tersebut lebih tepat dibaca sebagai **jasa konsultansi**. Dalam Perpres, jasa konsultansi dapat dievaluasi dengan metode:

- kualitas dan biaya;
- kualitas;
- pagu anggaran;
- biaya terendah.

Untuk pekerjaan tertentu, ruang lingkup, jenis tenaga ahli, dan waktu penyelesaiannya juga harus dijelaskan secara rinci dalam KAK. Ini menunjukkan bahwa klasifikasi paket tidak boleh ditentukan dari nama proyek, tetapi harus mengikuti substansi output dan karakter kerjanya. (JDIH LKPP)

Dalam praktik IT, satu proyek bahkan bisa memuat ketiga unsur sekaligus, misalnya:

- perangkat keras sebagai barang;
- implementasi dan operasi sebagai jasa lainnya;
- kajian dan desain sebagai jasa konsultansi.

Karena itu, perencanaan memegang peran sangat penting untuk menentukan apakah pekerjaan harus dipaketkan sebagai satu kesatuan yang sah dan rasional, atau dipisahkan agar evaluasi, kontrak, dan akuntabilitasnya lebih tepat. Prinsip dasarnya satu: **klasifikasi harus mengikuti substansi pekerjaan, bukan kenyamanan administrasi**. (JDIH LKPP)

2.7 Konsekuensi salah klasifikasi paket

Salah klasifikasi paket dalam pengadaan IT bukan kesalahan kecil. Dampaknya dapat menjalar ke seluruh siklus proyek. Jika pekerjaan yang seharusnya masuk kategori jasa konsultansi dipaksa menjadi barang atau jasa lainnya, maka akibatnya bisa berupa:

- metode evaluasi yang tidak tepat;
- bobot tenaga ahli yang terabaikan;
- KAK yang dangkal;
- kontrak yang tidak sesuai substansi;
- hasil proyek yang lemah.

Sebaliknya, jika kebutuhan yang sebenarnya standar diperlakukan terlalu kompleks, maka proses justru menjadi:

- lebih lambat;
- lebih mahal;
- kurang efisien.

Dengan kata lain, salah klasifikasi bukan hanya mengganggu proses, tetapi juga dapat merusak dokumen pemilihan, struktur kontrak, dan hasil akhir proyek. (JDIH LKPP)

Dalam proyek IT, salah klasifikasi juga dapat menimbulkan persoalan audit. Misalnya:

- paket integrasi sistem yang sangat teknis dan operasional ditulis seperti konsultansi umum, sehingga deliverable menjadi kabur;
- paket audit keamanan yang seharusnya menuntut kompetensi ahli justru diperlakukan seperti pembelian komoditas.

Ketika auditor memeriksa, organisasi akan sulit menjelaskan mengapa metode, kualifikasi, dan output yang diminta tidak selaras dengan substansi pekerjaannya. Karena itu, salah klasifikasi tidak hanya memperbesar risiko prosedural, tetapi juga melemahkan pertanggungjawaban substantif.

Lebih jauh lagi, salah klasifikasi dapat menimbulkan biaya tersembunyi. Literatur pengadaan mengingatkan bahwa biaya proyek sangat dipengaruhi oleh:

- ruang lingkup;
- keterampilan tim;
- biaya berulang dan tidak berulang;
- perubahan teknis di tengah jalan.

Jika sejak awal klasifikasi paket keliru, maka estimasi biaya, struktur kontrak, dan distribusi risiko ikut menjadi keliru. Hasilnya, organisasi mungkin tetap “menang tender”, tetapi kehilangan kendali atas biaya, mutu, dan manfaat.

Penutup

Bab ini menegaskan bahwa kerangka hukum pengadaan IT di Indonesia sebenarnya sudah cukup kaya, tetapi tidak otomatis sederhana. Kuncinya bukan menghafal seluruh pasal, melainkan memahami hubungan antara:

- Perpres PBJ terbaru;
- pedoman perencanaan;
- pedoman pelaksanaan melalui penyedia;
- mekanisme katalog dan toko daring;
- klasifikasi barang, jasa lainnya, dan jasa konsultansi.

Dari pemahaman itulah pengadaan IT bisa dibangun dengan lebih cerdas: **patuh aturan, tepat metode, tidak salah klasifikasi, dan lebih siap menghasilkan sistem yang benar-benar bermanfaat**. (JDIH LKPP)

Tabel Summary. Kerangka Hukum Inti Pengadaan IT di Indonesia

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
2.1 Perpres PBJ terbaru dan arah kebijakannya	Kerangka utama PBJ pemerintah harus dibaca bukan hanya sebagai aturan belanja, tetapi sebagai alat untuk mendorong **manfaat anggaran, percepatan PBJ, dan penggunaan produk dalam negeri**. Dalam konteks IT, ini berarti pengadaan harus dilihat sebagai instrumen strategis, bukan sekadar transaksi.	PPK dan tim teknis perlu memastikan bahwa paket IT tidak hanya sah secara prosedural, tetapi juga memberi nilai nyata: memperkuat layanan, mendukung interoperabilitas, dan membuka ruang bagi solusi lokal.
2.2 Perencanaan pengadaan sebagai fondasi keberhasilan proyek IT	Titik awal keberhasilan proyek IT ada pada **perencanaan**, bukan pada pemilihan vendor. Kebutuhan harus dirumuskan dari layanan yang ingin dibangun, masalah yang ingin diselesaikan, risiko yang ingin dikurangi, dan kapasitas organisasi yang tersedia.	Sebelum tender dimulai, organisasi harus jelas tentang proses bisnis, data, integrasi, keamanan, SDM, dan model operasi pasca-implementasi. Tanpa ini, dokumen pengadaan akan rapi tetapi hasil proyek rapuh.
2.3 Pelaksanaan pengadaan melalui penyedia	Pengadaan melalui penyedia adalah mekanisme utama untuk banyak kebutuhan IT, tetapi bidang IT sering memadukan **barang, layanan implementasi, dukungan, integrasi, dan operasi** dalam satu paket. Karena itu, pemahaman substansi pekerjaan jauh lebih penting daripada sekadar nama paket.	Dokumen pemilihan harus menegaskan *deliverable*, mutu teknis, peran tim, dokumentasi, SLA, dan hasil yang benar-benar dapat dipakai. Evaluasi tidak boleh semata-mata berorientasi harga jika paketnya kompleks.
2.4 Masa transisi aturan terbaru dan implikasinya	Pengadaan IT sering berjalan dalam situasi perubahan regulasi dan perubahan platform digital pengadaan. Karena itu, membaca **status aturan terbaru** menjadi bagian dari manajemen risiko, bukan pekerjaan administratif semata.	Unit kerja harus rajin memperbarui template, pendekatan, dan pemahaman hukum agar tidak memakai acuan lama yang sudah tidak relevan. Dalam proyek IT, keterlambatan menyesuaikan aturan dapat berujung pada kesalahan prosedur dan kesalahan desain paket.

2.5 Posisi katalog elektronik, toko daring, dan pengadaan digital	Katalog elektronik dan toko daring sangat penting, tetapi **tidak semua kebutuhan IT cocok** dimasukkan ke kanal yang sederhana. Mekanisme digital paling tepat untuk barang atau layanan yang standar, berisiko rendah, dan mudah dibandingkan.	Untuk laptop standar, periferal umum, atau komponen rutin, kanal digital bisa sangat efektif. Tetapi untuk integrasi sistem, migrasi data, audit keamanan, atau pengembangan aplikasi kompleks, pendekatan yang lebih kaya dan lebih ketat biasanya diperlukan.
2.6 Kapan IT masuk barang, jasa lainnya, atau jasa konsultansi	Klasifikasi paket IT harus mengikuti **substansi output**, bukan sekadar nama proyek. Ada kebutuhan yang lebih tepat diperlakukan sebagai barang, ada yang sebagai jasa lainnya, dan ada yang sebagai jasa konsultansi.	Penentuan klasifikasi harus dilakukan sejak perencanaan. Ini akan memengaruhi metode pemilihan, cara evaluasi, isi KAK, persyaratan tenaga ahli, struktur kontrak, dan ekspektasi hasil kerja.
2.7 Konsekuensi salah klasifikasi paket	Salah klasifikasi paket dapat merusak seluruh rantai pengadaan: metode evaluasi menjadi tidak tepat, dokumen pemilihan menjadi lemah, kontrak menjadi kabur, dan audit menjadi berisiko.	Dalam proyek IT, salah klasifikasi sering membuat organisasi membeli sesuatu yang secara administratif tampak benar, tetapi secara substantif gagal menjawab kebutuhan layanan, keamanan, dan keberlanjutan.

Ringkasan

Bab 2 menegaskan bahwa pengadaan IT harus berdiri di atas fondasi hukum yang benar, tetapi keberhasilan nyata baru muncul jika aturan itu diterjemahkan dengan cerdas ke dalam perencanaan, klasifikasi paket, metode pemilihan, dan pengendalian hasil kerja. Pengadaan IT tidak cukup patuh secara prosedural; ia juga harus tepat secara substansi, selaras dengan kebutuhan layanan, dan siap dipertanggungjawabkan saat implementasi maupun audit.

Bab 3. *Open Source*, Produk Dalam Negeri, dan TKDN Tinggi

3.1 Perbedaan *open source*, *free software*, dan gratis

Salah satu salah paham yang paling sering muncul dalam pengadaan IT adalah menganggap *open source* sama dengan “gratis”. Ini tidak tepat. *Open source* berbicara tentang lisensi, keterbukaan *source code*, serta hak untuk mempelajari, memodifikasi, dan mendistribusikan sesuai ketentuan lisensinya. *Free software* menekankan kebebasan pengguna, bukan harga. Sementara itu, “gratis” hanya berarti tidak ada biaya lisensi di awal.

Karena itu, sebuah perangkat lunak bisa saja gratis tetapi tetap tertutup. Sebaliknya, solusi *open source* bisa tetap menimbulkan biaya untuk:

- implementasi;
- integrasi;
- migrasi;
- pelatihan;
- pemeliharaan;
- dukungan profesional.

Maka, dalam pengadaan pemerintah, pertanyaan yang tepat bukan “apakah ini gratis?”, melainkan:

- apa sebenarnya yang dibeli;
- siapa yang menguasai sistem setelah proyek selesai;
- berapa biaya total selama siklus hidup layanan.

Di sinilah *open source* menjadi relevan. Ia bukan janji bahwa semuanya otomatis murah, tetapi peluang untuk menggeser fokus belanja dari lisensi semata ke **nilai guna, penguasaan teknis, dan jasa bernilai tambah**.

Jadi, sikap yang sehat sejak awal adalah ini: **tidak semua yang gratis itu strategis, dan tidak semua yang berbayar itu buruk**. Yang perlu diuji adalah apakah solusi tersebut mendukung:

- interoperabilitas;
- keberlanjutan;
- kontrol atas *deliverable*;
- keamanan;
- pertumbuhan kapasitas lokal.

3.2 Hubungan *open source* dengan kedaulatan digital

Hubungan *open source* dengan kedaulatan digital terletak pada kemampuannya untuk **mengurangi ketergantungan yang tidak perlu**. Dalam kerangka SPBE, pemerintah dituntut membangun layanan yang terintegrasi, berbasis proses bisnis, data, aplikasi, infrastruktur, dan keamanan yang saling terhubung. Prinsip integrasi layanan, *bagi pakai data*, basis data terintegrasi, dan sistem aplikasi yang saling tersambung membuat ketergantungan berlebihan pada format tertutup, *stack* tertutup, atau satu penyedia tunggal menjadi risiko strategis. (JDIH LKPP)

Open source memang tidak otomatis menjamin kedaulatan digital. Namun ia dapat menjadi alat yang kuat untuk mencapainya. Ketika kode, dokumentasi, standar integrasi, dan kendali konfigurasi lebih terbuka, organisasi memiliki ruang lebih besar untuk:

- melakukan audit teknis;
- bermigrasi;
- mengembangkan sistem lebih lanjut;
- mengganti penyedia tanpa harus memulai dari nol.

Itulah sebabnya *open source* relevan sebagai strategi pengadaan: bukan karena ia selalu terbaik dalam semua keadaan, tetapi karena ia memberi peluang bagi organisasi untuk lebih berdaulat atas arsitektur dan masa depan sistemnya sendiri.

Tetapi kedaulatan digital tidak lahir hanya dari slogan “pakai OSS”. Ia tetap memerlukan:

- SDM yang mampu mengelola;
- kontrak yang menegaskan hak atas *source code* dan *deliverable*;
- dokumentasi yang rapi;
- arsitektur yang tidak bergantung pada satu penyedia.

Tanpa itu, sebuah sistem bisa saja berbasis *open source*, tetapi praktiknya tetap tersandera oleh vendor tertentu.

3.3 Mengapa *open source* bisa memperbesar porsi nilai lokal

Dalam banyak proyek IT, sebagian besar belanja justru bocor ke lisensi asing, *subscription* tertutup, dan biaya eksternal yang tidak banyak meninggalkan kemampuan di dalam negeri. *Open source* dapat mengubah struktur ini.

Ketika biaya lisensi dapat ditekan, ruang anggaran bisa dialihkan ke hal-hal yang lebih produktif secara lokal, seperti:

- desain sistem;
- integrasi;
- migrasi;
- *customization*;
- dukungan operasi;
- pelatihan;
- dokumentasi;
- keamanan;
- penguatan tim lokal.

Di sinilah “nilai lokal” menjadi lebih nyata. Nilai itu tidak selalu tampak pada label produk akhir, tetapi terlihat pada **kapasitas yang tumbuh di dalam negeri**.

Sebagai strategi, *open source* juga selaras dengan beberapa prinsip penting pengadaan, antara lain:

- non-diskriminatif;
- mendukung interoperabilitas;
- memperkuat jasa lokal;
- meningkatkan kontrol atas *deliverable*;
- membuka ruang bagi penggunaan produk dalam negeri.

Secara praktis, *open source* memperbesar porsi nilai lokal karena persaingan penyedia tidak lagi bertumpu pada hak eksklusif atas lisensi, tetapi pada **kemampuan kerja nyata**: siapa yang paling baik dalam mengimplementasikan, mendukung, mengamankan, dan mengembangkan sistem.

3.4 TKDN, P3DN, dan belanja produk dalam negeri

Bab ini perlu berpijak pada kebijakan nasional yang jelas. **Perpres 46 Tahun 2025** diterbitkan untuk meningkatkan penggunaan produk dalam negeri dalam PBJ pemerintah, mempercepat pelaksanaan PBJ, dan mengoptimalkan kemanfaatan anggaran belanja pemerintah. Ini menunjukkan bahwa pengadaan bukan proses yang netral secara ekonomi, melainkan instrumen untuk menggerakkan kapasitas nasional. (JDIH LKPP – Perpres 46 Tahun 2025) (JDIH LKPP)

Dorongan itu diperkuat oleh **Inpres 2 Tahun 2022**, yang secara tegas memerintahkan percepatan penggunaan produk dalam negeri serta produk UMKM dan koperasi dalam PBJ pemerintah. Ringkasan resminya juga menegaskan arahan seperti:

- pembentukan Tim P3DN;
- penyusunan *roadmap*;
- penayangan produk dalam negeri di katalog elektronik;
- pencantuman kewajiban penggunaan produk dalam negeri dalam kontrak kerja sama.

(JDIH LKPP – Inpres 2 Tahun 2022) (JDIH LKPP)

Namun untuk proyek IT, membaca TKDN dan P3DN tidak boleh terlalu sempit. Memang ada komponen yang benar-benar berupa barang fisik dan bisa memiliki sertifikat TKDN. Tetapi ada juga bagian proyek IT yang nilainya justru dominan pada:

- jasa;
- integrasi;
- migrasi;
- operasi;
- pelatihan;
- dokumentasi.

Karena itu, pembaca perlu dibantu membedakan antara **TKDN formal pada barang** dan **nilai lokal yang nyata secara ekonomi** meskipun tidak selalu tampil sebagai sertifikat barang. Untuk konteks layanan verifikasi dan kebijakan TKDN industri, rujukan resminya dapat diarahkan ke layanan verifikasi TKDN Kemenperin. (BBT Kemenperin – Verifikasi TKDN) (Kemenperin)

3.5 Cara membaca “nilai lokal” di proyek IT

Dalam pengadaan IT, “nilai lokal” tidak boleh dipersempit hanya pada tempat barang dirakit. Dalam proyek digital, nilai sering tersebar ke banyak lapisan, seperti:

- perangkat;
- integrasi;
- migrasi data;
- *support*;
- pelatihan;
- dokumentasi;
- operasi;
- pengamanan.

Kalau organisasi hanya fokus pada perangkat, sebagian besar nilai lokal yang sesungguhnya bisa dibangun justru akan terlewat.

Contohnya sederhana. Sebuah server mungkin impor. Tetapi pekerjaan seperti:

- desain arsitektur;
- instalasi;
- *hardening*;
- virtualisasi;
- *backup*;
- integrasi dengan aplikasi lain;
- pemantauan;
- penyusunan SOP;
- *knowledge transfer*;
- dukungan operasional,

dapat seluruhnya dilakukan oleh SDM dalam negeri. Dari sudut pandang pembangunan kapasitas, justru nilai seperti inilah yang sangat penting karena meninggalkan kompetensi yang menetap di institusi dan di pasar tenaga kerja lokal.

Untuk proyek perangkat lunak, pendekatan ini bahkan lebih relevan. Banyak sistem informasi tidak menghasilkan “produk fisik” yang mudah diberi label nasionalitas. Namun proyek itu tetap bisa sangat pro-dalam-negeri bila porsi utama pekerjaannya berada pada tim lokal, *support* lokal, pusat kendali lokal, dokumentasi lokal, dan pengembangan lanjutan lokal.

3.6 Strategi menyusun paket agar tidak didominasi lisensi asing

Salah satu tugas strategis PPK dan tim teknis adalah menyusun paket dengan cerdas. Jika paket dibentuk secara keliru, struktur biaya sejak awal sudah terkunci untuk menguntungkan lisensi asing dan menekan ruang bagi jasa lokal. Sebaliknya, jika paket dirancang dengan baik, organisasi dapat memperbesar porsi anggaran untuk:

- integrasi;
- migrasi;
- dokumentasi;
- pelatihan;
- keamanan;
- operasi lokal.

Beberapa strategi yang dapat digunakan antara lain:

- memisahkan kebutuhan lisensi inti dari kebutuhan implementasi;
- memisahkan biaya migrasi dari biaya *managed service*;
- menyusun spesifikasi berbasis fungsi dan interoperabilitas, bukan merek;
- membaca biaya berdasarkan *total cost of ownership*, bukan harga awal semata.

Dengan pendekatan ini, pembeli bisa melihat lebih jelas mana biaya yang hanya menghasilkan hak pakai, dan mana biaya yang benar-benar membangun kemampuan organisasi.

Dalam proyek perangkat lunak, perhatian khusus juga perlu diberikan pada model *subscription* yang tampak ringan di awal, tetapi mahal dalam jangka panjang. Karena itu, pertanyaan yang tepat bukan hanya:

- berapa biaya tahun ini;
- tetapi juga berapa biaya 3 tahun;
- berapa biaya 5 tahun;
- dan berapa biaya keluar jika suatu saat harus pindah.

Pendekatan seperti ini membuat pengadaan lebih strategis dan tidak mudah terjebak pada harga masuk yang menipu.

3.7 Model *hybrid*: perangkat komersial + *stack open source* + jasa lokal

Dalam praktik, pendekatan yang paling sehat seringkali bukan yang ekstrim, melainkan **model hybrid**. Ini realistis, karena kebutuhan di lapangan jarang hitam-putih.

Dalam model ini:

- ada komponen perangkat atau *appliance* tertentu yang lebih rasional dibeli secara komersial;
- ada lapisan sistem operasi, basis data, *middleware*, *container*, *monitoring*, atau LMS yang layak dibangun di atas *stack open source*;
- ada lapisan jasa lokal yang mengikat semuanya agar benar-benar bekerja.

Model *hybrid* memberi beberapa keuntungan:

- organisasi tetap pragmatis terhadap kebutuhan yang spesifik;
- kendali tidak sepenuhnya jatuh ke satu vendor;
- perubahan tidak harus dilakukan sekaligus;
- ruang bagi integrator, tim implementasi, penyedia *support*, dan pelatih lokal tetap besar.

Dalam operasi berkelanjutan, model seperti ini juga sering lebih aman. Solusi komersial dipakai hanya pada titik yang benar-benar memberi nilai, sementara lapisan lain dibangun dengan komponen yang lebih terbuka dan lebih mudah dipelihara. Hasil akhirnya bukan dogma, melainkan arsitektur yang lebih seimbang antara:

- keandalan;
- biaya;
- kontrol;
- keberlanjutan.

3.8 Metrik praktis untuk “TKDN tinggi” pada proyek software dan jasa

Untuk proyek IT, istilah “TKDN tinggi” tidak cukup jika hanya menjadi slogan. Ia harus diterjemahkan menjadi ukuran yang bisa dibaca dan dipertanggungjawabkan. Untuk barang, rujukannya dapat lebih formal melalui sertifikat TKDN. Namun untuk proyek software dan jasa, organisasi perlu memakai metrik praktis tambahan. (BBT Kemenperin – Verifikasi TKDN) (Kemenperin)

Beberapa metrik yang bisa dipakai antara lain:

- persentase jam kerja yang dikerjakan SDM di dalam negeri;
- persentase biaya proyek yang jatuh ke jasa lokal;
- porsi *support* dan operasi yang dikelola lokal;
- kepemilikan dokumentasi dan repositori di lingkungan pengguna;
- ketersediaan pelatihan bagi admin/operator lokal;
- kemampuan melakukan pengembangan lanjutan tanpa ketergantungan absolut pada vendor asing.

Metrik seperti ini sering lebih jujur untuk proyek digital daripada sekedar bertanya apakah produk akhirnya “lokal” atau “tidak lokal”.

Buku ini juga bisa mendorong pembaca membuat **matriks nilai lokal** pada setiap proyek, agar dapat dibaca dengan jelas:

- mana nilai yang datang dari barang;
- mana dari jasa;
- mana dari transfer pengetahuan;
- mana dari operasi;
- mana dari keberlanjutan.

Dengan begitu, istilah TKDN tinggi dalam proyek digital tidak berhenti sebagai kosmetik, tetapi menjadi alat evaluasi yang nyata.

3.9 Kapan *open source* bukan pilihan terbaik

Bab ini tetap perlu jujur dan tidak ideologis. Ada situasi ketika *open source* bukan pilihan terbaik, misalnya ketika:

- organisasi membutuhkan fitur yang sangat spesifik dan stabilitasnya hanya terbukti pada solusi tertentu;
- ada kebutuhan interoperabilitas dengan ekosistem yang sudah terlanjur tertutup;
- kemampuan SDM internal masih sangat terbatas;
- belum ada mitra lokal yang memadai;
- risiko transisi terlalu besar dibanding manfaat jangka pendek.

Ada juga keadaan ketika biaya adopsi *open source* justru tinggi karena:

- migrasi data;
- pelatihan;
- perubahan proses kerja;
- *support* jangka panjang.

Kalau semua itu tidak dihitung dengan benar, keputusan “beralih ke OSS” bisa terlihat bagus secara slogan, tetapi lemah secara eksekusi.

Karena itu, keputusan tetap harus berbasis:

- *total cost of ownership*;
- risiko;
- kesiapan SDM;
- kebutuhan keamanan;
- konsekuensi operasional.

Selain itu, untuk sistem yang memproses data pribadi dan layanan kritikal, keamanan dan pelindungan data tetap bersifat mutlak. Dalam konteks itu, pertanyaan yang benar bukan semata OSS atau bukan OSS, tetapi:

- apakah sistem aman;
- apakah terdokumentasi;
- apakah dapat diaudit;
- apakah dapat dioperasikan secara bertanggung jawab.

3.10 Matriks keputusan: OSS penuh, *hybrid*, atau *proprietary* terbatas

Bagian ini sebaiknya ditutup dengan satu prinsip yang tegas: pilih **OSS penuh** bila kebutuhan, kapasitas, dan pasar lokal mendukung; pilih **hybrid** bila organisasi membutuhkan keseimbangan antara pragmatisme dan kontrol; pilih **proprietary terbatas** hanya bila memang ada alasan kuat dan dapat dipertanggungjawabkan.

Secara ringkas:

- **OSS penuh** cocok ketika kebutuhan relatif umum, komunitas atau dukungan komersial tersedia, interoperabilitas penting, dan organisasi ingin memperbesar penguasaan teknis internal.
- **Hybrid** cocok ketika ada kebutuhan khusus pada lapisan tertentu, tetapi organisasi tetap ingin mengurangi dominasi lisensi dan menjaga ruang bagi jasa lokal.
- **Proprietary terbatas** layak dipilih bila ada kebutuhan yang sangat spesifik, risiko operasional sangat tinggi, atau pasar pengganti yang realistis belum tersedia.

Namun jika solusi *proprietary* dipilih, penggunaannya sebaiknya tetap dikawal dengan:

- kontrak yang kuat;
- kewajiban dokumentasi;
- hak akses yang jelas;
- strategi keluar yang masuk akal.

Jadi, inti kebijakan bab ini bukan “semua harus OSS”, melainkan bahwa setiap keputusan harus diarahkan pada:

- manfaat publik;
- pengurangan ketergantungan yang tidak perlu;
- penguatan kapasitas nasional;
- akuntabilitas jangka panjang.

Arah kebijakan ini sejalan dengan penekanan PBJ terbaru pada penggunaan produk dalam negeri dan optimalisasi manfaat belanja pemerintah. (JDIH LKPP – Perpres 46 Tahun 2025) (JDIH LKPP)

Penutup

Bab ini menegaskan bahwa *open source* bukan slogan murah-meriah, melainkan instrumen strategis. Nilainya bukan karena semua harus serba terbuka, tetapi karena ia dapat membantu negara dan organisasi:

- mengurangi *vendor lock-in*;
- memperbesar nilai lokal;
- memperkuat jasa dan talenta dalam negeri;
- membangun sistem yang lebih mudah diintegrasikan dan dikendalikan.

Namun keberhasilan pendekatan ini tetap ditentukan oleh:

- desain paket;
- kualitas kontrak;
- kesiapan SDM;
- keamanan;
- disiplin membaca biaya sepanjang siklus hidup.

Di titik itulah agenda produk dalam negeri dan TKDN tinggi bertemu dengan agenda yang lebih besar, yaitu **kedaulatan digital yang pragmatis, terukur, dan siap audit**. (JDIH LKPP – Perpres 46 Tahun 2025) (JDIH LKPP)

.

Tabel Summary. *Open Source*, Produk Dalam Negeri, dan TKDN Tinggi

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
3.1 Perbedaan *open source*, *free software*, dan gratis	*Open source* bukan berarti sekadar gratis. Yang utama adalah **keterbukaan lisensi, akses ke kode sumber, hak mempelajari, memodifikasi, dan mendistribusikan sesuai lisensi**. Karena itu, pengadaan harus menilai manfaat strategis, bukan hanya biaya lisensi awal.	PPK dan tim teknis perlu menghitung **biaya total siklus hidup**: implementasi, migrasi, pelatihan, *support*, keamanan, dan pemeliharaan, bukan hanya harga masuk.
3.2 Hubungan *open source* dengan kedaulatan digital	*Open source* penting karena dapat mengurangi **ketergantungan yang tidak perlu** pada vendor, format tertutup, dan *stack* tertutup. Dalam konteks SPBE, hal ini sejalan dengan kebutuhan integrasi proses bisnis, data, aplikasi, infrastruktur, dan keamanan.	Gunakan *open source* sebagai alat untuk memperbesar **kontrol atas arsitektur, migrasi, audit teknis, dan keberlanjutan sistem**, bukan sebagai slogan semata.
3.3 Mengapa *open source* bisa memperbesar porsi nilai lokal	Dengan menekan dominasi lisensi, *open source* dapat memperbesar porsi anggaran untuk **integrasi, migrasi, *customization*, pelatihan, dokumentasi, operasi, dan pengamanan** yang dapat dikerjakan pelaku lokal.	Struktur paket pengadaan sebaiknya diarahkan agar lebih banyak nilai jatuh pada **kerja nyata SDM dan perusahaan dalam negeri**, bukan hanya hak pakai teknologi asing.
3.4 TKDN, P3DN, dan belanja produk dalam negeri	Kebijakan nasional makin tegas mendorong **penggunaan produk dalam negeri** dan percepatan P3DN dalam PBJ pemerintah. Untuk bidang IT, ini berarti pengadaan harus dibaca juga sebagai instrumen pembangunan kapasitas nasional.	Dalam menyusun paket IT, pembaca perlu menilai bukan hanya aspek kepatuhan, tetapi juga bagaimana belanja tersebut benar-benar memperkuat ekosistem teknologi nasional.
3.5 Cara membaca "nilai lokal" di proyek IT	Nilai lokal dalam proyek IT tidak hanya terletak pada perangkat, tetapi juga pada **integrasi, migrasi data, *support*, pelatihan, dokumentasi, operasi, dan pengamanan**.	Evaluasi proyek IT perlu membaca **nilai ekonomi lokal secara lebih luas**, termasuk kemampuan yang tertinggal di institusi setelah proyek selesai.

3.6 Strategi menyusun paket agar tidak didominasi lisensi asing	Paket yang disusun buruk akan mendorong dominasi lisensi asing. Paket yang disusun cerdas akan memberi ruang lebih besar untuk **jasa lokal, transfer pengetahuan, interoperabilitas, dan pengendalian biaya**.	Pisahkan secara jelas komponen lisensi, implementasi, migrasi, *support*, dan operasi agar organisasi tahu mana biaya akses, mana biaya pembangunan kapasitas.
3.7 Model *hybrid*: perangkat komersial + *stack open source* + jasa lokal	Pendekatan paling sehat sering kali bukan ekstrem penuh, melainkan **model hybrid**: memakai komponen komersial bila perlu, tetapi membangun sebanyak mungkin lapisan sistem di atas komponen yang lebih terbuka dan didukung jasa lokal.	Model *hybrid* cocok untuk instansi yang ingin tetap pragmatis, tetapi juga ingin mengurangi *vendor lock-in* dan memperbesar ruang bagi talenta dalam negeri.
3.8 Metrik praktis untuk "TKDN tinggi" pada proyek *software* dan jasa	Untuk proyek digital, frasa **TKDN tinggi** tidak cukup sebagai slogan. Ia perlu diterjemahkan ke dalam metrik praktis seperti porsi jam kerja lokal, porsi biaya untuk jasa lokal, kepemilikan dokumentasi, dan kemampuan pengembangan lanjutan di dalam negeri.	Setiap proyek IT sebaiknya punya **matriks nilai lokal** agar keberpihakan pada dalam negeri bisa diukur dan dipertanggungjawabkan.
3.9 Kapan *open source* bukan pilihan terbaik	*Open source* bukan jawaban otomatis untuk semua kasus. Ada situasi ketika kebutuhan fitur, risiko operasional, kesiapan SDM, atau biaya transisi membuat solusi lain lebih rasional.	Keputusan harus berbasis **TCO, risiko, keamanan, kesiapan organisasi, dan keberlanjutan operasi**, bukan preferensi ideologis.
3.10 Matriks keputusan: OSS penuh, *hybrid*, atau *proprietary* terbatas	Pilihan terbaik bergantung pada konteks. **OSS penuh** cocok bila kebutuhan umum dan dukungan memadai; ***hybrid*** cocok bila dibutuhkan keseimbangan; ***proprietary* terbatas** layak bila ada alasan kuat yang dapat dipertanggungjawabkan.	Organisasi perlu membuat keputusan yang **pragmatis tetapi tetap strategis**: mengurangi ketergantungan, menjaga keamanan, memperbesar nilai lokal, dan tetap siap audit.

Ringkasan

Bab 3 menegaskan bahwa *open source*, produk dalam negeri, dan TKDN tinggi harus dibaca sebagai strategi pengadaan yang cerdas, bukan slogan. Tujuannya bukan sekadar menghemat lisensi, tetapi **mengurangi *vendor lock-in*, memperbesar nilai lokal, memperkuat jasa dan talenta dalam negeri, serta membangun sistem yang lebih mudah diintegrasikan, dikendalikan, dan diaudit**. Namun semua itu hanya akan berhasil bila diterjemahkan ke dalam desain paket, kontrak, metrik nilai lokal, dan keputusan teknologi yang realistis.

Bab 4. Pengadaan IT dalam Kerangka SPBE dan Transformasi Digital

4.1 SPBE sebagai konteks pengadaan IT pemerintah

Pengadaan IT pemerintah tidak boleh dipahami hanya sebagai urusan membeli perangkat, aplikasi, atau jasa teknis. Dalam konteks Indonesia, pengadaan digital harus ditempatkan dalam kerangka **Sistem Pemerintahan Berbasis Elektronik (SPBE)**. **Perpres 95 Tahun 2018** menegaskan bahwa SPBE adalah penyelenggaraan pemerintahan yang memanfaatkan teknologi informasi dan komunikasi untuk memberikan layanan kepada pengguna SPBE. Dalam regulasi yang sama, tata kelola SPBE juga ditegaskan sebagai kerangka untuk memastikan pengaturan, pengarahan, dan pengendalian dilakukan secara terpadu. Artinya, proyek IT pemerintah sejak awal harus dibaca sebagai bagian dari tata kelola pemerintahan digital, bukan sebagai proyek teknis yang berdiri sendiri. (Perpres 95 Tahun 2018) (Peraturan BPK)

Implikasinya besar. Ketika sebuah instansi membeli aplikasi, membangun pusat data, mengadakan jaringan, atau menyewa layanan digital, yang sebenarnya sedang dibentuk bukan hanya aset teknologi, tetapi juga **kapasitas layanan pemerintahan**. Karena itu, pertanyaan dasarnya bukan lagi “barang apa yang dibeli?”, melainkan:

- layanan apa yang diperkuat;
- proses bisnis apa yang diperbaiki;
- data apa yang harus dibagi-pakai;
- risiko apa yang harus dikendalikan.

Jika konteks SPBE diabaikan, pengadaan mudah jatuh menjadi belanja parsial: sibuk di administrasi, tetapi lemah di manfaat. (Perpres 95 Tahun 2018) (Peraturan BPK)

Perpres SPBE juga menunjukkan bahwa pembangunan SPBE nasional diarahkan untuk mendukung layanan yang terpadu dan menyeluruh. Dalam arah strategisnya, SPBE dikaitkan dengan integrasi layanan perencanaan, penganggaran, pengadaan barang dan jasa pemerintah, akuntabilitas kinerja, kepegawaian, kearsipan, dan pengaduan publik. Pesannya jelas: pengadaan IT pemerintah harus mendukung ekosistem layanan digital pemerintahan, bukan sekadar memenuhi daftar kebutuhan unit kerja secara terpisah. (Perpres 95 Tahun 2018 PDF) (Peraturan BPK)

Pelajaran strategisnya sederhana: pengadaan IT pemerintah yang baik selalu dimulai dari pemahaman bahwa proyek tersebut adalah bagian dari SPBE. Karena itu, proyek harus dinilai dari kontribusinya terhadap:

- tata kelola;
- kualitas layanan;
- efisiensi proses;
- integrasi antarunit. (Perpres 95 Tahun 2018) (Peraturan BPK)

4.2 Arsitektur SPBE nasional: proses bisnis, data, aplikasi, infrastruktur, keamanan

Salah satu alasan banyak proyek IT pemerintah gagal terhubung satu sama lain adalah karena pengadaan dilakukan sebelum organisasi benar-benar berpikir dalam kerangka arsitektur. Padahal, Perpres SPBE sangat jelas: **arsitektur SPBE** adalah kerangka dasar yang mendeskripsikan integrasi proses bisnis, data dan informasi, infrastruktur SPBE, aplikasi SPBE, dan keamanan SPBE untuk menghasilkan layanan yang terintegrasi. Ini bukan istilah tambahan, melainkan inti dari cara berpikir pengadaan IT pemerintah yang benar. (Perpres 95 Tahun 2018) (Peraturan BPK)

Pengaturan arsitektur SPBE nasional kemudian dipertegas lagi melalui **Perpres 132 Tahun 2022** tentang Arsitektur SPBE Nasional. Perpres ini menegaskan arah kebijakan dan strategi arsitektur SPBE nasional, termasuk domain-domain utamanya. Dengan demikian, pengadaan IT tidak boleh hanya ditarik ke satu sisi teknis semata, tetapi harus selalu menimbang secara bersamaan:

- proses bisnis;
- data dan informasi;
- aplikasi;
- infrastruktur;
- keamanan. (Perpres 132 Tahun 2022) (Peraturan BPK)

Bagi pembaca buku ini, pesan praktisnya sangat penting. Bila suatu instansi hendak mengadakan aplikasi baru, pertanyaan yang harus dijawab bukan hanya soal fiturnya. Instansi juga harus menjelaskan:

- proses bisnis apa yang berubah;
- data apa yang dipakai dan dihasilkan;
- aplikasi apa yang harus dihubungkan;
- infrastruktur apa yang menopang;
- kontrol keamanan apa yang wajib diterapkan.

Tanpa lima pertanyaan ini, pengadaan mudah berubah menjadi proyek fitur, bukan proyek layanan. (Perpres 95 Tahun 2018, Perpres 132 Tahun 2022) (Peraturan BPK)

Arsitektur SPBE juga bukan dokumen mati. Ia harus direviu dan disesuaikan ketika kebutuhan, evaluasi, atau risiko digital berubah. Artinya, pengadaan IT tidak boleh sepenuhnya kaku. Ia harus cukup disiplin untuk patuh pada arsitektur, tetapi juga cukup lincah untuk mengikuti perubahan organisasi dan perkembangan risiko. Ini adalah pembacaan yang masuk akal dari kerangka SPBE nasional. (Perpres 132 Tahun 2022) (Peraturan BPK)

4.3 Interoperabilitas data dan integrasi antar sistem

Salah satu penyakit paling mahal dalam belanja digital pemerintah adalah sistem yang berjalan sendiri-sendiri. Aplikasi dibuat, anggaran terserap, tetapi data tidak terhubung, proses kerja tetap berulang, dan pengguna harus menginput informasi yang sama berkali-kali. Dari sisi teknologi, ini pemborosan. Dari sisi pelayanan publik, ini kegagalan desain. Karena itu, **interoperabilitas** harus ditempatkan sebagai syarat strategis dalam pengadaan IT. (Perpres 95 Tahun 2018) (Peraturan BPK)

Arah kebijakan SPBE memang bukan membuat aplikasi sebanyak mungkin, melainkan membangun layanan yang terintegrasi, berbagi data, dan lebih efisien. Dalam peta strategis SPBE, ada penekanan pada portal layanan publik, portal administrasi pemerintahan, portal data nasional, serta integrasi layanan lintas fungsi pemerintahan. Maka, pengadaan yang tidak memikirkan integrasi sejak awal pada dasarnya sedang menunda masalah ke masa depan. (Perpres 95 Tahun 2018 PDF) (Peraturan BPK)

Dalam praktik pengadaan, interoperabilitas setidaknya berarti tiga hal:

- spesifikasi meminta format data yang terbuka dan mudah dipertukarkan;
- sistem memiliki mekanisme integrasi yang jelas, seperti API dan dokumentasi pertukaran data;
- ada bukti bahwa sistem dapat dihubungkan dengan ekosistem yang sudah ada, bukan hanya berjalan saat demo.

Jika tiga hal ini diabaikan, organisasi hanya membeli aplikasi yang terlihat modern tetapi menambah beban integrasi di belakang hari. Ini adalah inferensi langsung dari tuntutan keterpaduan layanan dalam kerangka SPBE. (Perpres 95 Tahun 2018) (Peraturan BPK)

Interoperabilitas juga tidak hanya soal teknis. Ketika data bergerak antar sistem, keamanan dan pelindungan data pribadi harus tetap dijaga. **UU 27 Tahun 2022 tentang Pelindungan Data Pribadi** mengatur bahwa pengendali data pribadi wajib melindungi dan memastikan keamanan data pribadi yang diproses. Jadi, integrasi antar sistem harus dinilai bukan hanya dari kelancaran teknisnya, tetapi juga dari sisi legalitas, keamanan, dan akuntabilitas. (UU 27 Tahun 2022, PDF UU 27 Tahun 2022) (Peraturan BPK)

Intinya, pengadaan IT yang baik tidak berhenti pada aplikasi yang “jadi”. Ia harus sampai pada sistem yang:

- bisa berbagi data dengan aman;
- terhubung secara masuk akal;
- mengurangi duplikasi proses. (Perpres 95 Tahun 2018, UU 27 Tahun 2022) (Peraturan BPK)

4.4 Hubungan pengadaan dengan layanan publik digital

Pengadaan IT terlalu sering berhenti pada spesifikasi teknis, ruang lingkup kerja, dan administrasi kontrak. Semua itu penting, tetapi tidak cukup. Ujung dari SPBE adalah **layanan**. Karena itu, pengadaan IT pemerintah harus selalu dikaitkan dengan kualitas layanan publik digital yang ingin dihasilkan. Jika hubungan ini putus, proyek akan mudah terlihat sukses di atas kertas tetapi gagal di mata warga. (Perpres 95 Tahun 2018) (Peraturan BPK)

Perpres SPBE sejak bagian pertimbangannya menekankan bahwa SPBE diperlukan untuk mewujudkan tata kelola yang bersih, efektif, transparan, akuntabel, dan pelayanan publik yang berkualitas serta terpercaya. Artinya, alasan negara berinvestasi pada IT bukan sekadar modernisasi simbolik, melainkan untuk membuat pelayanan lebih baik. Karena itu, setiap paket pengadaan IT idealnya dapat dijelaskan dampaknya terhadap:

- waktu layanan;
- akurasi layanan;
- keterjangkauan layanan;
- transparansi layanan;
- kepuasan pengguna. (Perpres 95 Tahun 2018) (Peraturan BPK)

Di titik ini, pengadaan IT harus belajar berpikir dari sisi pengguna. Pertanyaannya menjadi lebih substantif:

- apakah sistem ini mempermudah warga atau justru menambah friksi;
- apakah layanan menjadi lebih cepat atau hanya memindahkan antrean manual ke antrean digital;
- apakah integrasi benar-benar mengurangi pengisian data berulang;
- apakah pengguna merasa sistem ini jelas, aman, dan dapat dipercaya.

Dengan cara ini, pengadaan IT tidak lagi dipandang sebagai tujuan, melainkan sebagai sarana untuk menghasilkan layanan publik digital yang lebih baik. Kesimpulan ini selaras dengan tujuan SPBE dalam regulasi resminya. (Perpres 95 Tahun 2018) (Peraturan BPK)

Pendekatan berbasis layanan juga akan memperbaiki kualitas KAK dan evaluasi. Jika layanan publik dijadikan acuan, ruang lingkup kerja akan lebih mudah ditulis dalam bentuk hasil, misalnya:

- waktu proses turun;
- ketersediaan layanan naik;
- integrasi berjalan;
- *audit trail* tersedia;
- dokumentasi lengkap.

Dengan begitu, pengadaan menjadi lebih substantif dan tidak sekadar mengejar daftar fitur. (Perpres 95 Tahun 2018 PDF) (Peraturan BPK)

4.5 Menghindari aplikasi silo

Aplikasi silo adalah salah satu bentuk kegagalan paling umum sekaligus paling mahal dalam transformasi digital pemerintah. Ia muncul ketika unit kerja membeli atau membangun sistem berdasarkan kepentingan sempitnya sendiri, tanpa disiplin arsitektur bersama. Dalam jangka pendek, situasinya mungkin terlihat produktif. Tetapi dalam jangka menengah, organisasi dibebani banyak aplikasi yang berdiri sendiri, sulit diintegrasikan, memakai definisi data yang berbeda, dan menuntut biaya tambahan hanya untuk membuat semuanya saling bicara. Ini bertentangan dengan semangat keterpaduan dalam SPBE. (Perpres 95 Tahun 2018, Perpres 132 Tahun 2022) (Peraturan BPK)

Cara menghindarinya bukan sekadar melarang unit membuat aplikasi baru. Yang lebih penting adalah membangun disiplin pada tahap pengadaan. Sebelum sebuah paket disetujui, harus ada pertanyaan tegas seperti:

- apakah kebutuhan ini bisa dipenuhi dengan optimalisasi sistem yang sudah ada;
- apakah ini duplikasi fungsi;
- bagaimana hubungan sistem baru dengan arsitektur SPBE instansi;
- data apa yang harus dibagi;
- apakah ada API atau standar integrasi;
- siapa yang akan mengelola dan memelihara setelah proyek selesai.

Tanpa pertanyaan seperti ini, pengadaan justru berubah menjadi mesin produksi silo digital. Ini adalah inferensi praktis yang sangat langsung dari prinsip arsitektur dan interoperabilitas SPBE. (Perpres 95 Tahun 2018, Perpres 132 Tahun 2022) (Peraturan BPK)

Silo juga sering muncul karena kontrak terlalu fokus pada pembangunan aplikasi, tetapi lemah pada kewajiban integrasi, dokumentasi, migrasi data, pengujian antarsistem, dan pengelolaan hak akses. Karena itu, *deliverable* tidak boleh berhenti pada aplikasi yang hidup di server. Ia harus mencakup pula:

- dokumentasi arsitektur;
- dokumentasi integrasi;
- skema data;
- SOP;
- bukti uji integrasi.

Tanpa semua itu, ketika vendor pergi, institusi hanya mewarisi sistem yang aktif tetapi sulit dipahami dan sulit dikembangkan. Pendekatan ini konsisten dengan kebutuhan akuntabilitas dan keamanan dalam SPBE serta UU PDP. (Perpres 95 Tahun 2018, UU 27 Tahun 2022) (Peraturan BPK)

Dari sisi tata kelola, aplikasi silo bertentangan dengan semangat SPBE yang menuntut keterpaduan, efisiensi, dan layanan terintegrasi. Dari sisi anggaran, silo menciptakan pemborosan berlapis. Dari sisi audit, silo menyulitkan jejak data, akuntabilitas, dan pembuktian proses. Jadi, melawan silo bukan hanya agenda teknis, tetapi juga agenda manajemen, anggaran, dan akuntabilitas. (Perpres 95 Tahun 2018) (Peraturan BPK)

Penutup

Bab ini menegaskan bahwa pengadaan IT pemerintah harus selalu dibaca dalam kerangka **SPBE** dan **transformasi digital**, bukan sebagai belanja teknologi yang berdiri sendiri. Kunci berpikirnya ada pada lima hal:

- memahami SPBE sebagai konteks utama;
- menempatkan pengadaan di dalam arsitektur proses–data–aplikasi–infrastruktur–keamanan;
- menuntut interoperabilitas yang nyata;
- mengaitkan pengadaan dengan mutu layanan publik digital;
- secara sadar mencegah lahirnya aplikasi silo.

Jika lima hal ini dijaga, pengadaan IT tidak hanya sah secara prosedural, tetapi juga jauh lebih mungkin menghasilkan layanan yang **terintegrasi, aman, bermanfaat, dan siap audit**. (Perpres 95 Tahun 2018, Perpres 132 Tahun 2022, UU 27 Tahun 2022) (Peraturan BPK)

Tabel Summary Bab 4. Pengadaan IT dalam Kerangka SPBE dan Transformasi Digital

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
4.1 SPBE sebagai konteks pengadaan IT pemerintah	Pengadaan IT pemerintah harus dibaca sebagai bagian dari **SPBE**, bukan sekadar belanja perangkat atau aplikasi. Artinya, proyek digital harus dinilai dari kontribusinya pada tata kelola, efisiensi proses, integrasi layanan, dan kualitas pelayanan publik.	PPK, tim teknis, dan pimpinan harus mulai dari pertanyaan layanan: **masalah apa yang diselesaikan, proses apa yang diperbaiki, dan manfaat apa yang dihasilkan**, bukan hanya barang apa yang dibeli.
4.2 Arsitektur SPBE nasional: proses bisnis, data, aplikasi, infrastruktur, keamanan	Arsitektur SPBE menuntut integrasi lima domain utama: **proses bisnis, data dan informasi, aplikasi, infrastruktur, dan keamanan**. Pengadaan yang mengabaikan salah satu domain ini berisiko menghasilkan sistem yang tidak utuh dan sulit dipertahankan.	Setiap paket IT harus diuji terhadap lima pertanyaan dasar: proses bisnis apa yang berubah, data apa yang dipakai, aplikasi apa yang harus terhubung, infrastruktur apa yang menopang, dan kontrol keamanan apa yang wajib ada.
4.3 Interoperabilitas data dan integrasi antar sistem	Interoperabilitas bukan tambahan, tetapi syarat strategis. Tujuannya adalah mencegah duplikasi proses, mengurangi *silo*, dan memungkinkan pertukaran data yang aman antar sistem.	Spesifikasi dan kontrak harus minta **format data terbuka, API/integrasi yang jelas, dokumentasi pertukaran data, serta kontrol keamanan dan pelindungan data pribadi**.

4.4 Hubungan pengadaan dengan layanan publik digital	Ujung dari pengadaan IT pemerintah adalah **layanan publik digital yang lebih baik**, bukan sekadar selesainya proyek. Karena itu, keberhasilan harus diukur dari manfaat bagi pengguna, bukan hanya dari serah terima teknis.	KAK dan evaluasi perlu diarahkan pada hasil seperti **waktu layanan lebih cepat, akurasi meningkat, akses lebih mudah, integrasi berjalan, dan audit trail tersedia**.
4.5 Menghindari aplikasi silo	Aplikasi silo muncul ketika pengadaan dilakukan secara sektoral tanpa disiplin arsitektur bersama. Akibatnya, sistem berdiri sendiri, data tidak nyambung, biaya integrasi membengkak, dan manfaat layanan melemah.	Sebelum menyetujui paket baru, instansi harus mengecek: **apakah ini duplikasi fungsi, apakah bisa memakai sistem yang ada, bagaimana integrasinya, siapa pengelolanya, dan apa bukti uji integrasinya**.

Ringkasan

Bab 4 menegaskan bahwa pengadaan IT pemerintah hanya akan bernilai tinggi bila ditempatkan di dalam kerangka SPBE dan transformasi digital. Artinya, pengadaan harus mengikuti arsitektur yang jelas, menuntut interoperabilitas yang nyata, berorientasi pada mutu layanan publik digital, dan secara sadar mencegah lahirnya aplikasi silo. Dengan begitu, belanja IT tidak berhenti sebagai proyek teknis, tetapi berubah menjadi **alat untuk membangun layanan pemerintah yang terintegrasi, aman, efisien, dan siap audit**.

Bab 5. Proses Perencanaan Pengadaan IT dari Awal

5.1 Identifikasi masalah bisnis dan kebutuhan layanan

Perencanaan pengadaan IT yang sehat selalu dimulai dari **masalah bisnis** dan **kebutuhan layanan**, bukan dari daftar barang, merek, atau teknologi tertentu. Ini sejalan dengan **Pedoman Perencanaan Pengadaan LKPP**, yang menempatkan perencanaan sebagai fondasi agar pengadaan selaras dengan tujuan, kebijakan, prinsip, dan etika PBJ pemerintah. Dalam bahasa sederhana: sebelum bicara server, aplikasi, cloud, atau lisensi, organisasi harus lebih dulu jujur menjawab masalah apa yang sebenarnya ingin diselesaikan. (Peraturan LKPP Nomor 11 Tahun 2021)

Dalam proyek IT, kesalahan yang paling mahal sering terjadi ketika kebutuhan teknis dirumuskan terlalu cepat. Instansi merasa membutuhkan aplikasi baru, padahal akar masalahnya mungkin justru ada pada:

- proses kerja yang buruk;
- data yang tidak rapi;
- integrasi yang belum ada;
- tata kelola yang lemah.

Akibatnya, pengadaan bergerak ke solusi yang terlihat modern, tetapi tidak menyentuh akar persoalan.

Dalam konteks SPBE, logika ini menjadi semakin penting. **Perpres 95 Tahun 2018** memandang layanan digital pemerintah sebagai hasil integrasi proses bisnis, data, aplikasi, infrastruktur, dan keamanan. Artinya, kebutuhan pengadaan IT tidak boleh didefinisikan hanya sebagai kebutuhan teknologi, tetapi sebagai kebutuhan layanan yang ditopang teknologi. (Perpres 95 Tahun 2018)

Intinya jelas: **mulai dari masalah dan layanan, bukan dari produk**. Pengadaan IT yang dibalik urutannya hampir selalu menjadi mahal, sulit dipakai, dan lemah manfaat.

5.2 Memetakan pemilik proses, pengguna, admin, operator, pimpinan

Setelah masalah diidentifikasi, langkah berikutnya adalah menentukan siapa yang memiliki proses, siapa yang menggunakan sistem, siapa yang mengelola, siapa yang mengambil keputusan, dan siapa yang menanggung risiko operasional. Banyak proyek IT terlihat rapi di atas kertas, tetapi gagal karena peran-peran ini tidak dibedakan sejak awal.

Masalah yang sering muncul antara lain:

- pengguna akhir dianggap sama dengan admin;
- pimpinan hanya hadir saat peluncuran;
- operator tidak dilibatkan saat kebutuhan dirumuskan;
- tidak ada pemilik proses yang benar-benar jelas.

Padahal, tiap peran memiliki fungsi yang berbeda:

- **Pemilik proses** menjelaskan kebutuhan substantif dan hasil yang diinginkan.
- **Pengguna** memvalidasi kenyataan lapangan dan hambatan operasional.
- **Admin dan operator** memastikan sistem nantinya benar-benar bisa dikelola.
- **Pimpinan** menentukan prioritas, dukungan anggaran, dan legitimasi perubahan.

Tanpa kombinasi ini, proyek IT mudah terjebak dalam desain teknis yang tidak cocok dengan realitas kerja.

Dari perspektif pengadaan, pemetaan pemangku kepentingan juga membantu mencegah konflik di belakang hari. Kebingungan peran, kebutuhan yang berubah-ubah, dan sinyal yang tidak konsisten kepada penyedia hampir selalu berakar pada lemahnya konsolidasi sejak awal. Karena itu, salah satu keluaran penting tahap perencanaan seharusnya adalah **peta pemangku kepentingan** yang jelas.

Minimal, peta itu menjawab:

- siapa pemilik proses;
- siapa pengguna utama;
- siapa admin;
- siapa operator;
- siapa tim keamanan;
- siapa pengelola data;
- siapa pejabat penentu;
- siapa penerima akhir hasil kerja.

Peta ini tampak sederhana, tetapi sering menjadi pembeda antara proyek yang terkendali dan proyek yang liar.

5.3 Menyusun *business case* dan manfaat terukur

Banyak pengadaan IT gagal bukan karena teknologinya buruk, tetapi karena proyek tidak pernah dibangun di atas *business case* yang jernih. *Business case* menjelaskan mengapa proyek perlu dijalankan, manfaat apa yang dituju, biaya dan resikonya, serta mengapa opsi tertentu lebih layak dibanding alternatif lain.

Dalam praktik, pertanyaan yang harus dijawab adalah: **mengapa proyek ini layak dibiayai?**

Manfaat dalam proyek IT juga harus dibuat terukur. Tidak cukup hanya menulis:

- “meningkatkan layanan”;
- “mendukung digitalisasi”;
- “memperkuat sistem.”

Rumusan seperti itu terlalu umum. Yang lebih berguna adalah indikator yang lebih konkret, misalnya:

- waktu proses berkurang;
- input ganda menurun;
- tingkat kegagalan transaksi turun;
- *uptime* meningkat;
- kualitas data membaik;
- temuan audit berkurang;
- kepuasan pengguna meningkat.

Dengan ukuran seperti ini, organisasi akan lebih mudah menyusun:

- KAK;
- HPS;
- indikator kinerja;
- uji manfaat proyek.

Dari sudut tata kelola PBJ, *business case* yang baik juga mencegah pengadaan impulsif. Tidak semua masalah harus dijawab dengan sistem baru. Kadang yang dibutuhkan justru:

- perbaikan SOP;
- pelatihan SDM;
- optimalisasi sistem lama;
- integrasi layanan yang sudah ada.

Karena itu, pengadaan IT yang matang harus memiliki **alasan bisnis yang tegas**, **manfaat yang terukur**, dan **argumen yang bisa dipertanggungjawabkan** kepada pimpinan, auditor, dan pengguna.

5.4 Analisis kondisi *as-is* dan target *to-be*

Sebelum menulis solusi, organisasi harus memahami kondisi saat ini (*as-is*) dan kondisi target (*to-be*). Banyak proyek IT melompat terlalu cepat ke rancangan masa depan tanpa memahami realitas yang sedang berjalan. Akibatnya, target terlihat rapi di presentasi, tetapi bertabrakan dengan kondisi lapangan.

Analisis *as-is* idealnya mencakup:

- proses bisnis yang sedang berjalan;
- aplikasi yang sudah dipakai;
- data yang tersedia dan kualitasnya;
- ketergantungan manual;
- masalah keamanan;
- kapasitas SDM;
- kontrak aktif;
- keterbatasan infrastruktur.

Dari sana, organisasi baru bisa merumuskan *to-be* yang realistis, misalnya:

- proses yang lebih pendek;
- integrasi yang lebih rapi;
- peran yang lebih jelas;
- dashboard yang relevan;
- pengamanan data yang lebih kuat.

Tanpa analisis ini, pengadaan IT cenderung menghasilkan solusi yang tidak sesuai medan tempur nyata.

Dalam konteks transformasi digital pemerintah, pendekatan *as-is* ke *to-be* juga penting untuk mencegah pemborosan. **Perpres 95 Tahun 2018** mendorong integrasi layanan dan menghindari pembangunan aplikasi sejenis jika fungsi serupa sebenarnya sudah tersedia. (Perpres 95 Tahun 2018)

Jadi, analisis *as-is* dan *to-be* bukan latihan administratif. Ia adalah alat untuk memastikan bahwa pengadaan IT benar-benar bergerak dari realitas menuju perbaikan yang masuk akal.

5.5 Menentukan strategi: bangun baru, kembangkan, integrasi, migrasi, sewa layanan, atau adopsi OSS

Setelah masalah dan kondisi dipetakan, organisasi masuk ke pertanyaan strategis: **apa bentuk intervensi yang paling tepat?** Tidak semua kebutuhan harus dijawab dengan membangun sistem baru. Dalam banyak kasus, pilihan yang lebih sehat justru:

- mengembangkan sistem lama;
- mengintegrasikan sistem yang sudah ada;
- melakukan migrasi;
- menyewa layanan;
- mengadopsi *open source*.

Keputusan strategi seharusnya mempertimbangkan beberapa hal sekaligus:

- urgensi kebutuhan;
- kesiapan SDM;
- biaya sepanjang siklus hidup;
- ketergantungan vendor;
- kemampuan integrasi;
- risiko keamanan;
- kapasitas pemeliharaan.

Setiap opsi punya konsekuensi sendiri:

- **Bangun baru** memberi ruang desain penuh, tetapi mahal dan berisiko.
- **Kembangkan sistem lama** bisa lebih cepat, tetapi dibatasi arsitektur warisan.
- **Integrasi** cocok bila fungsi utama sudah ada, tetapi data belum terhubung.
- **Migrasi** relevan bila sistem lama sudah menjadi beban.
- **Sewa layanan** menarik untuk kebutuhan tertentu, tetapi harus dibaca hati-hati dari sisi biaya jangka panjang, pelindungan data, dan strategi keluar.
- **Adopsi OSS** dapat strategis jika mendukung interoperabilitas, nilai lokal, dan pengurangan *vendor lock-in*.

Yang penting, organisasi harus menahan godaan untuk memilih solusi yang paling trendi. Solusi yang modern belum tentu solusi yang paling tepat.

Karena itu, tahap ini idealnya menghasilkan **matriks opsi** yang memuat:

- alternatif solusi;
- kelebihan;
- risiko;
- estimasi biaya;
- kebutuhan SDM;
- dampak ke pengguna;
- konsekuensi jangka panjang.

Dengan begitu, keputusan teknologi menjadi lebih rasional dan lebih mudah dipertanggungjawabkan.

5.6 Menentukan tahapan proyek: cepat, bertahap, atau transformasi penuh

Salah satu sumber kegagalan proyek IT adalah ambisi yang tidak proporsional. Organisasi ingin semuanya selesai sekaligus: seluruh modul hidup, seluruh data pindah, seluruh unit terhubung, dan seluruh pengguna langsung beradaptasi. Dalam praktik, pendekatan seperti ini sering terlalu berat.

Karena itu, sejak tahap perencanaan harus ditentukan apakah proyek lebih cocok dijalankan sebagai:

- intervensi cepat;
- implementasi bertahap;
- transformasi penuh.

Secara umum:

- **Model cepat** cocok bila kebutuhan mendesak, ruang lingkup terbatas, dan manfaat bisa segera dirasakan.
- **Model bertahap** cocok bila sistem kompleks, integrasi banyak, data belum rapi, atau SDM masih berproses.
- **Transformasi penuh** hanya layak bila dukungan pimpinan kuat, *business case* jelas, kesiapan organisasi cukup, dan manajemen perubahan disiapkan serius.

Kesalahan yang paling sering terjadi adalah memilih transformasi penuh untuk organisasi yang sebenarnya belum siap. Hasilnya biasanya:

- proyek melebar;
- jadwal tertunda;
- kualitas turun;
- penerimaan hasil kerja dipaksakan.

Dari sudut pengadaan, penentuan tahapan ini memengaruhi:

- pemaketan;
- jadwal;
- HPS;
- manajemen kontrak;
- strategi pengujian.

Tahapan yang realistis membuat ruang lingkup lebih tajam dan risiko lebih terkendali. Jadi, **tempo proyek adalah keputusan strategis**, bukan urusan teknis kecil.

5.7 Manajemen risiko sejak perencanaan

Dalam pengadaan IT, risiko tidak muncul belakangan. Risiko sudah ada sejak fase perencanaan. Itulah sebabnya manajemen risiko harus masuk sejak awal, bukan dijadikan lampiran kosmetik.

Risiko yang perlu dibaca sejak awal meliputi:

- kebutuhan yang kabur;
- salah klasifikasi paket;
- masalah integrasi;
- risiko keamanan;
- risiko pelindungan data pribadi;
- keterlambatan;
- ketergantungan vendor;
- penolakan pengguna;
- kesiapan SDM.

Untuk proyek yang melibatkan data pribadi, dimensinya lebih serius. **UU 27 Tahun 2022 tentang Pelindungan Data Pribadi** menempatkan pelindungan data sebagai kewajiban hukum. Karena itu, sejak tahap perencanaan harus sudah dipikirkan:

- jenis data apa yang diproses;
- siapa pengendalinya;
- siapa yang mengakses;
- bagaimana kontrol keamanannya;
- bagaimana pembuktian kepatuhannya.

(UU 27 Tahun 2022)

Jika risiko seperti ini baru dibahas saat implementasi, biasanya biaya koreksinya jauh lebih mahal.

Karena itu, salah satu keluaran minimal dari tahap perencanaan seharusnya adalah **register risiko awal** yang berisi:

- daftar risiko utama;
- probabilitas;
- dampak;
- indikator peringatan dini;
- strategi mitigasi.

Dengan itu, proyek masuk ke tahap pengadaan dengan mata terbuka, bukan dengan optimisme kosong.

5.8 RUP, pemaketan, metode pemilihan, dan jadwal

Setelah fondasi kebutuhan dan strategi terbentuk, perencanaan harus diterjemahkan ke dalam instrumen PBJ yang formal: **RUP, pemaketan, metode pemilihan, dan jadwal**.

Dalam rezim PBJ yang berlaku, **RUP** adalah daftar rencana pengadaan yang akan dilaksanakan oleh K/L/PD. Kualitas RUP sangat dipengaruhi oleh kualitas berpikir pada tahap sebelumnya. Jika kebutuhan masih kabur, maka RUP hanya menjadi daftar belanja. Jika perencanaan matang, RUP menjadi peta eksekusi yang masuk akal. (Perpres 46 Tahun 2025 – konsolidasi PBJ)

Pemaketan dalam proyek IT harus dilakukan hati-hati. Paket yang terlalu besar bisa:

- menutup kompetisi;
- memperbesar risiko kegagalan tunggal;
- menyulitkan pengendalian.

Sebaliknya, paket yang terlalu kecil bisa:

- memecah tanggung jawab;
- menciptakan celah integrasi;
- menambah beban koordinasi.

Karena itu, pemaketan harus mengikuti:

- substansi kerja;
- keterkaitan antar-komponen;
- urgensi;
- kapasitas pengelolaan.

Jadwal juga tidak boleh diperlakukan sebagai formalitas. Dalam proyek IT, jadwal yang tidak realistis hampir selalu mendorong dua hal:

- mutu menurun;
- konflik bertambah.

Maka, penjadwalan harus memberi ruang untuk:

- klarifikasi kebutuhan;
- persiapan pemilihan;
- pelaksanaan;
- pengembangan;
- integrasi;
- pengujian;
- pelatihan;
- masa stabilisasi;
- serah terima.

Jadwal yang sehat memberi ruang untuk kualitas. Jadwal yang dipaksa biasanya hanya memindahkan masalah ke tahap berikutnya.

5.9 Kapan proyek sebaiknya dipecah menjadi beberapa paket

Tidak semua proyek IT ideal dijalankan dalam satu paket besar. Ada situasi ketika proyek justru lebih sehat bila dipecah menjadi beberapa paket, terutama jika pekerjaannya sangat berbeda sifat.

Misalnya, pekerjaan seperti berikut sering layak dipisahkan:

- kajian dan desain;
- pengadaan perangkat;
- implementasi aplikasi;
- integrasi;
- pelatihan;
- audit keamanan.

Memisahkan paket bisa memberi manfaat berupa:

- evaluasi yang lebih tepat;
- kompetisi yang lebih terbuka;
- hasil kerja yang lebih terukur.

Namun pemecahan paket juga tidak boleh sembarangan. Jika komponen yang sangat saling tergantung dipisah tanpa kendali arsitektur yang kuat, organisasi bisa menghadapi masalah baru seperti:

- saling lempar tanggung jawab antar vendor;
- integrasi terlambat;
- spesifikasi tidak sinkron;
- tidak ada pihak yang benar-benar memegang hasil akhir.

Karena itu, pertanyaan yang sehat bukan “lebih enak satu paket atau banyak paket?”, tetapi: **bagaimana menyusun paket agar kompetitif, akuntabel, dan tetap menghasilkan sistem yang utuh?**

Dalam beberapa kasus, satu paket *end-to-end* masuk akal. Dalam kasus lain, paket terpisah justru lebih aman. Yang penting, keputusan itu harus mengikuti logika manfaat dan risiko, bukan sekadar kenyamanan administrasi.

5.10 Kesalahan klasik perencanaan pengadaan IT

Bab ini layak ditutup dengan daftar kesalahan klasik, karena sebagian besar masalah proyek digital sebenarnya terus berulang.

Kesalahan yang paling umum antara lain:

- memulai dari solusi, bukan dari masalah;
- tidak melibatkan pemilik proses dan pengguna sejak awal;
- mengabaikan kondisi *as-is*;
- tidak menyusun *business case* dan manfaat terukur;
- mencampur terlalu banyak tujuan dalam satu paket;
- menunda pembahasan risiko;
- menyusun jadwal yang tidak realistis;
- memilih strategi hanya karena trend, bukan karena konteks.

Kesalahan lain yang juga sangat sering adalah tidak memikirkan pengelolaan pasca-implementasi. Sistem selesai dibangun, tetapi:

- admin belum siap;
- SOP belum ada;
- dokumentasi tidak lengkap;
- tidak ada rencana transisi.

Dalam perspektif pengadaan dan kontrak, ini berbahaya, karena proyek yang tampak selesai justru belum siap dioperasikan.

Ringkasnya, kesalahan klasik perencanaan pengadaan IT hampir selalu berakar pada satu hal: **kurangnya disiplin berpikir di awal**. Dan justru di situlah proyek terbaik dibedakan dari proyek yang nanti sibuk memadamkan kebakaran.

Penutup

Bab ini menegaskan bahwa **perencanaan adalah titik paling menentukan dalam pengadaan IT**. Di sinilah organisasi menetapkan:

- masalah yang ingin diselesaikan;
- siapa pemilik prosesnya;
- manfaat yang dituju;
- kondisi awal dan target;
- strategi solusi;
- tahapan proyek;
- risiko utama;
- struktur RUP dan pemaketan;
- batas realistis dari apa yang bisa dicapai.

Jika tahap ini kuat, maka KAK, spesifikasi, HPS, kontrak, dan pelaksanaan akan jauh lebih terkendali. Jika tahap ini lemah, maka hampir seluruh persoalan berikutnya sebenarnya hanyalah akibat yang datang terlambat. (Peraturan LKPP Nomor 11 Tahun 2021, Perpres 46 Tahun 2025)

Tabel Summary. Proses Perencanaan Pengadaan IT dari Awal

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
5.1 Identifikasi masalah bisnis dan kebutuhan layanan	Perencanaan pengadaan IT harus dimulai dari **masalah bisnis dan kebutuhan layanan**, bukan dari daftar barang, merek, atau teknologi yang ingin dibeli.	Sebelum menyusun paket, instansi harus menjawab: **masalah apa yang ingin diselesaikan, layanan apa yang ingin diperbaiki, dan manfaat apa yang diharapkan**.
5.2 Memetakan pemilik proses, pengguna, admin, operator, pimpinan	Proyek IT memerlukan peta peran yang jelas: siapa pemilik proses, siapa pengguna utama, siapa admin/operator, siapa pengelola data, dan siapa pengambil keputusan. Tanpa ini, sistem mudah lahir tanpa pemilik yang jelas.	Tim perencana perlu membuat **peta pemangku kepentingan** sejak awal agar kebutuhan tidak bias, konflik dapat dikurangi, dan hasil proyek benar-benar bisa dioperasikan.
5.3 Menyusun *business case* dan manfaat terukur	Setiap proyek IT perlu punya **alasan bisnis yang jelas**, manfaat yang terukur, serta argumentasi mengapa opsi tertentu dipilih dibanding alternatif lain.	Manfaat proyek perlu diterjemahkan ke indikator konkret, seperti **waktu proses lebih singkat, penginputan ganda berkurang, uptime meningkat, atau kualitas data membaik**.
5.4 Analisis kondisi *as-is* dan target *to-be*	Organisasi harus memahami kondisi nyata saat ini (*as-is*) sebelum menetapkan kondisi target (*to-be*). Tanpa itu, solusi yang dibeli sering tidak cocok dengan realitas lapangan.	Sebelum membuat KAK, perlu dilakukan pemetaan proses berjalan, aplikasi yang sudah ada, kualitas data, kapasitas SDM, infrastruktur, dan hambatan utama agar target perbaikan tetap realistis.
5.5 Menentukan strategi: bangun baru, kembangkan, integrasi, migrasi, sewa layanan, atau adopsi OSS	Tidak semua kebutuhan harus dijawab dengan membangun sistem baru. Ada banyak opsi strategi, dan masing-masing harus dinilai dari biaya, risiko, kapasitas, integrasi, dan keberlanjutan.	Tim perencana sebaiknya membuat **matriks opsi solusi** agar keputusan teknologi lebih rasional dan tidak sekadar mengikuti tren.

5.6 Menentukan tahapan proyek: cepat, bertahap, atau transformasi penuh	Tempo proyek adalah keputusan strategis. Tidak semua organisasi siap untuk transformasi penuh sekaligus; banyak yang lebih aman memulai secara cepat atau bertahap.	Penentuan tahapan akan memengaruhi ruang lingkup, HPS, jadwal, risiko, dan kualitas hasil. **Tahapan yang realistis biasanya lebih aman daripada ambisi besar yang terlalu cepat.**
5.7 Manajemen risiko sejak perencanaan	Risiko proyek IT sudah muncul sejak awal, bukan belakangan. Risiko kebutuhan, integrasi, keamanan, data pribadi, keterlambatan, dan ketergantungan vendor harus dibaca sejak tahap perencanaan.	Setiap proyek sebaiknya punya **register risiko awal** yang memuat risiko utama, dampak, indikator dini, dan strategi mitigasinya.
5.8 RUP, pemaketan, metode pemilihan, dan jadwal	Perencanaan yang baik harus diterjemahkan ke dalam **RUP, pemaketan, metode pemilihan, dan jadwal** yang realistis. RUP yang baik bukan sekadar daftar belanja, tetapi peta eksekusi proyek.	Pemaketan dan jadwal perlu disusun berdasarkan substansi kerja, keterkaitan komponen, tingkat kompleksitas, dan kapasitas pengelolaan internal.
5.9 Kapan proyek sebaiknya dipecah menjadi beberapa paket	Tidak semua proyek IT ideal dijalankan sebagai satu paket besar. Dalam beberapa kasus, pemecahan paket membuat evaluasi lebih tepat, kompetisi lebih sehat, dan hasil kerja lebih terukur.	Keputusan memecah atau menggabungkan paket harus mempertimbangkan **keterkaitan teknis, risiko koordinasi, kontrol arsitektur, dan kapasitas tim internal**.
5.10 Kesalahan klasik perencanaan pengadaan IT	Kesalahan paling umum adalah mulai dari solusi, tidak melibatkan pemilik proses, tidak memahami kondisi *as-is*, tidak punya *business case*, menunda pembahasan risiko, dan membuat jadwal yang tidak realistis.	Daftar kesalahan klasik ini bisa dipakai sebagai **checklist pra-pengadaan** agar organisasi tidak mengulangi pola gagal yang sama.

Ringkasan

Bab 5 menegaskan bahwa kualitas pengadaan IT sangat ditentukan oleh kualitas perencanaannya. Di tahap inilah organisasi menetapkan masalah yang ingin diselesaikan, pemilik proses yang harus dilibatkan, manfaat yang ingin dicapai, kondisi awal dan target, strategi solusi, tahapan proyek, risiko utama, serta struktur paket dan jadwal yang masuk akal. Jika perencanaan kuat, maka tahapan berikutnya akan jauh lebih terkendali. Jika perencanaan lemah, maka hampir semua masalah yang muncul kemudian sebenarnya hanya akibat yang tertunda.

Bab 6. Desain Sistem sebagai Dasar KAK dan Spesifikasi

6.1 Mengapa desain sistem harus muncul sebelum tender

Salah satu kesalahan paling sering dalam pengadaan IT adalah memulai tender ketika desain sistem belum jelas. Organisasi sudah ingin membeli aplikasi, server, lisensi, atau jasa implementasi, tetapi belum memiliki gambaran yang cukup tentang proses bisnis, data, integrasi, arsitektur aplikasi, kebutuhan infrastruktur, dan kontrol keamanan. Akibatnya, tender berjalan di atas asumsi: vendor menafsirkan ruang lingkup secara sempit, pengguna berharap hasil yang lebih luas, dan kontrak akhirnya menjadi ruang salah paham. Dalam kerangka **SPBE**, pendekatan seperti ini memang bermasalah karena arsitektur SPBE didefinisikan sebagai integrasi proses bisnis, data dan informasi, infrastruktur, aplikasi, dan keamanan untuk menghasilkan layanan yang terintegrasi. (Perpres 95 Tahun 2018) (Peraturan BPK)

Kalau desain belum ada sebelum tender, biasanya muncul lima risiko langsung:

- spesifikasi menjadi kabur atau terlalu umum;
- ruang lingkup kerja mudah bergeser di tengah jalan;
- biaya tambahan muncul karena banyak hal baru diketahui saat implementasi;
- integrasi dan keamanan dibahas terlambat;
- dokumen serah terima menjadi lemah karena hasil yang diharapkan tidak didefinisikan dengan presisi.

Karena itu, prinsip yang sehat harus tegas: **desain sistem mendahului tender**, bukan lahir setelah vendor terpilih. Vendor boleh membantu mematangkan rancangan implementasi, tetapi organisasi tetap harus masuk ke pengadaan dengan kerangka desain yang cukup untuk mengendalikan mutu, biaya, dan risiko. Pembacaan ini sejalan dengan fungsi perencanaan pengadaan dalam pedoman LKPP dan posisi arsitektur dalam SPBE. (Peraturan LKPP Nomor 11 Tahun 2021, Perpres 95 Tahun 2018) (JDIH LKPP)

6.2 Arsitektur bisnis

Desain sistem yang baik selalu dimulai dari **arsitektur bisnis**. Lapisan ini menjelaskan proses apa yang dijalankan, siapa aktornya, keputusan apa yang dibuat, layanan apa yang diberikan, dan keluaran apa yang dihasilkan. Banyak proyek IT gagal karena terlalu cepat membahas layar, server, atau bahasa pemrograman, padahal proses bisnis dasarnya sendiri belum dipetakan dengan rapi. Dalam kerangka SPBE, domain proses bisnis memang menjadi salah satu unsur utama arsitektur nasional. (Perpres 95 Tahun 2018, Perpres 132 Tahun 2022) (Peraturan BPK)

Arsitektur bisnis penting karena ia menerjemahkan kebutuhan organisasi ke dalam bahasa yang bisa dipakai untuk merancang sistem. Jika proses bisnis masih kabur, aplikasi yang dibangun biasanya hanya mendigitalisasi kekacauan lama. Akibatnya, sistem terlihat modern, tetapi tetap lambat, tetap berulang, dan tetap membuat pengguna frustrasi.

Dalam konteks pengadaan, arsitektur bisnis minimal seharusnya menghasilkan:

- peta proses utama;
- titik keputusan;
- peran aktor;
- alur persetujuan;
- pengecualian proses;
- target perbaikan layanan.

Dari sini barulah kebutuhan aplikasi, data, integrasi, dan beban operasional dapat diturunkan secara masuk akal. Tanpa arsitektur bisnis, KAK cenderung penuh fitur tetapi miskin logika layanan. Ini adalah inferensi yang wajar dari tuntutan arsitektur SPBE yang menempatkan proses bisnis sebagai fondasi desain layanan digital pemerintah. (Perpres 95 Tahun 2018, Perpres 132 Tahun 2022) (Peraturan BPK)

6.3 Arsitektur data

Setelah proses bisnis dipahami, lapisan berikutnya adalah **arsitektur data**. Ini mencakup data apa yang dibutuhkan, dari mana asalnya, siapa pemiliknya, bagaimana kualitasnya, bagaimana relasinya, siapa yang boleh mengakses, bagaimana data berpindah antar sistem, dan bagaimana ia disimpan. Banyak proyek digital gagal justru di sini: aplikasi sudah jadi, tetapi datanya duplikat, kotor, tidak sinkron, atau sulit dipertukarkan. Dalam SPBE, domain data dan informasi memang merupakan komponen inti arsitektur nasional, bukan sekadar pelengkap. (Perpres 95 Tahun 2018, Perpres 132 Tahun 2022) (Peraturan BPK)

Untuk proyek yang menyentuh data pribadi, arsitektur data menjadi jauh lebih penting. **UU 27 Tahun 2022 tentang Pelindungan Data Pribadi** mengatur kewajiban pengendali dan prosesor data pribadi, termasuk perlindungan dalam rangkaian pemrosesan data. Itu berarti desain data tidak cukup bicara tabel dan relasi, tetapi juga harus memikirkan tujuan pemrosesan, klasifikasi data, kontrol akses, retensi, dan tanggung jawab hukum. (UU 27 Tahun 2022) (Peraturan BPK)

Dalam pengadaan, arsitektur data minimal perlu menjawab:

- data master apa yang dipakai;
- sumber kebenaran utamanya berada di mana;
- data apa yang ditarik dari sistem lain;
- bagaimana validasi dilakukan;
- bagaimana pembaruan atau penghapusan dijalankan;
- bagaimana jejak perubahan dicatat.

Tanpa ini, spesifikasi akan mudah jatuh menjadi sekadar daftar input-output, tanpa tata kelola data yang sehat. Kesimpulan ini mengikuti langsung tuntutan SPBE atas arsitektur data dan tuntutan UU PDP atas keamanan pemrosesan data pribadi. (Perpres 95 Tahun 2018, UU 27 Tahun 2022) (Peraturan BPK)

6.4 Arsitektur aplikasi

Sesudah proses dan data, organisasi perlu merumuskan **arsitektur aplikasi**. Di lapisan ini, fokusnya adalah aplikasi apa saja yang sudah ada, fungsi inti apa yang harus tersedia, modul mana yang saling bergantung, layanan mana yang bersifat umum dan mana yang khusus, serta bagaimana hubungan antara aplikasi baru dan aplikasi lama. Tanpa arsitektur aplikasi, instansi sangat mudah membangun aplikasi silo yang nantinya justru menambah beban integrasi. Perpres SPBE secara eksplisit memasukkan domain aplikasi SPBE ke dalam arsitektur nasional. (Perpres 95 Tahun 2018, Perpres 132 Tahun 2022) (Peraturan BPK)

Artinya, pengadaan aplikasi baru tidak boleh dilakukan seolah-olah sistem itu berdiri sendiri. Setiap paket harus diuji terhadap pertanyaan dasar seperti:

- fungsi ini sebenarnya sudah ada di sistem mana;
- apakah benar perlu aplikasi baru;
- bagaimana hubungannya dengan portal atau layanan yang sudah berjalan;
- bagaimana ia diintegrasikan ke ekosistem instansi.

Arsitektur aplikasi yang baik juga membantu menahan godaan membeli solusi “semua dalam satu” hanya karena terlihat praktis. Kadang solusi monolitik memang masuk akal, tetapi dalam banyak kasus pendekatan modular lebih sehat karena memudahkan integrasi, pengujian, penggantian komponen, dan pengembangan bertahap. Ini adalah pertimbangan desain yang masuk akal berdasarkan tujuan interoperabilitas dan arsitektur aplikasi dalam SPBE. (Perpres 95 Tahun 2018 PDF, Perpres 132 Tahun 2022 PDF) (Peraturan BPK)

Sebelum tender dimulai, organisasi seharusnya sudah memiliki peta aplikasi yang cukup jelas, setidaknya meliputi:

- modul inti;
- dependensi;
- integrasi;
- antarmuka pengguna utama;
- batas antara fungsi wajib dan fungsi yang bisa dikembangkan bertahap.

6.5 Arsitektur infrastruktur

Sistem yang baik tidak akan bertahan tanpa **arsitektur infrastruktur** yang realistis. Ini mencakup komputasi, penyimpanan, jaringan, segmentasi, kapasitas, ketersediaan daya, lokasi penempatan, lingkungan operasi, pemantauan, dan kesiapan dukungan. Banyak pengadaan IT terlalu fokus pada fitur aplikasi, padahal kegagalan layanan sering justru lahir dari fondasi infrastruktur yang tidak cocok dengan beban nyata. Dalam SPBE, domain infrastruktur jelas menjadi salah satu pilar arsitektur. (Perpres 95 Tahun 2018, Perpres 132 Tahun 2022) (Peraturan BPK)

Karena itu, pengadaan sistem tidak boleh hanya menanyakan “aplikasinya bisa jalan atau tidak”, tetapi juga:

- di atas infrastruktur seperti apa sistem dijalankan;
- dengan kapasitas berapa;
- untuk beban seperti apa;
- dengan redundansi apa;
- dan siapa yang mengelolanya.

Arsitektur infrastruktur yang sehat minimal menjawab:

- estimasi pengguna;
- volume transaksi;
- kebutuhan storage;
- pola pertumbuhan;
- kebutuhan jaringan;
- kebutuhan akses dari luar;
- segmentasi lingkungan produksi dan nonproduksi;
- kebutuhan monitoring.

Tanpa jawaban ini, spesifikasi server, jaringan, atau cloud akan mudah salah sasaran: terlalu kecil dan cepat gagal, atau terlalu besar dan boros. Jika desain infrastrukturnya matang, organisasi bisa menulis kebutuhan berbasis kapasitas, performa, kompatibilitas, dan keandalan, bukan berbasis merek tertentu. Itu jauh lebih sehat dari sisi persaingan dan akuntabilitas, dan sejalan dengan semangat perencanaan pengadaan dalam pedoman LKPP. (Peraturan LKPP Nomor 11 Tahun 2021, Perpres 95 Tahun 2018) (JDIH LKPP)

6.6 Arsitektur keamanan

Di banyak proyek, keamanan masih diperlakukan sebagai tambahan yang nanti dibahas belakangan. Ini berbahaya. **Arsitektur keamanan** harus muncul sejak desain awal karena hampir semua keputusan sistem punya implikasi keamanan: struktur akses, segmentasi jaringan, identitas, enkripsi, logging, *audit trail*, manajemen kerentanan, sampai respons insiden. Jika keamanan masuk terlambat, biaya koreksinya biasanya tinggi dan hasilnya sering tambal sulam. Dalam SPBE, domain keamanan SPBE memang menjadi salah satu domain arsitektur nasional. (Perpres 95 Tahun 2018, Perpres 132 Tahun 2022) (Peraturan BPK)

Untuk sistem yang memproses data pribadi, desain keamanan juga harus dibaca sebagai bagian dari kepatuhan. **UU PDP** menuntut perlindungan data pribadi oleh pengendali dan prosesor data, sehingga isu keamanan dan privasi tidak bisa dipisahkan dari desain sistem. (UU 27 Tahun 2022, PDF UU 27 Tahun 2022) (Peraturan BPK)

Dalam praktik pengadaan, arsitektur keamanan minimal perlu menjawab:

- model akses berbasis peran;
- autentikasi;
- kebutuhan MFA;
- logging;
- enkripsi saat transit dan saat tersimpan;
- segmentasi lingkungan;
- mekanisme backup;
- prosedur patching;
- kebutuhan audit keamanan.

Kalau hal-hal ini belum jelas sebelum tender, spesifikasi keamanan biasanya berubah menjadi tempelan umum yang sulit diuji saat serah terima. Kesimpulan itu merupakan inferensi langsung dari kewajiban arsitektur keamanan dalam SPBE dan kewajiban perlindungan data dalam UU PDP. (Perpres 95 Tahun 2018, UU 27 Tahun 2022) (Peraturan BPK)

6.7 Integrasi, API, *message bus*, dan pertukaran data

Sistem modern hampir tidak pernah hidup sendirian. Karena itu, desain harus memberi tempat khusus pada integrasi, API, *message bus*, dan pertukaran data. Ini bukan topik teknis kecil. Dalam banyak proyek pemerintah, justru titik paling sulit ada di sini: sistem internal harus berbicara dengan sistem lain, data harus mengalir lintas fungsi, dan perubahan di satu aplikasi tidak boleh merusak keseluruhan ekosistem. Dalam kerangka SPBE, integrasi memang menjadi tujuan eksplisit dari arsitektur nasional. (Perpres 95 Tahun 2018, Perpres 132 Tahun 2022) (Peraturan BPK)

Praktiknya, desain integrasi harus menjawab beberapa pertanyaan penting:

- sistem mana yang menjadi sumber data utama;
- data apa yang dikonsumsi dan dipublikasikan;
- apakah pertukaran dilakukan secara sinkron, asinkron, atau *batch*;
- apakah dibutuhkan *message broker* atau cukup API langsung;
- bagaimana autentikasi antar sistem dilakukan;
- bagaimana skema kesalahan dan *retry*;
- bagaimana versi dikelola.

Semua pertanyaan ini perlu cukup terjawab sebelum tender agar vendor tidak membangun integrasi secara improvisasi. Intinya, integrasi yang baik tidak bisa lahir dari kalimat umum seperti “sistem harus terintegrasi”. Ia harus diturunkan ke rancangan teknis dan tata kelola pertukaran data yang jelas. Itu adalah konsekuensi praktis dari arsitektur SPBE yang menekankan keterpaduan proses, data, aplikasi, infrastruktur, dan keamanan. (Perpres 95 Tahun 2018 PDF, Perpres 132 Tahun 2022 PDF) (Peraturan BPK)

6.8 *On-premise*, *colocation*, cloud, dan *hybrid cloud*

Pemilihan model penempatan sistem adalah keputusan strategis, bukan soal tren. Pilihan **on-premise**, **colocation**, **cloud**, dan **hybrid cloud** masing-masing membawa implikasi yang berbeda terhadap biaya, keamanan, kapasitas tim, fleksibilitas, dan ketergantungan vendor. Karena itu, pilihan ini harus menjadi bagian dari desain sistem, bukan sekadar opsi belanja di akhir. Pembacaan ini konsisten dengan pendekatan perencanaan pengadaan LKPP dan arsitektur infrastruktur SPBE. (Peraturan LKPP Nomor 11 Tahun 2021, Perpres 95 Tahun 2018) (JDIH LKPP)

Secara ringkas:

- **On-premise** memberi kontrol fisik dan operasional lebih besar, tetapi menuntut kesiapan ruang, daya, pendingin, keamanan fisik, dan tim pengelola.
- **Colocation** memberi sebagian keuntungan kontrol, dengan beban fasilitas yang lebih ringan.
- **Cloud** memberi elastisitas dan kecepatan, tetapi harus dibaca hati-hati dari sisi portabilitas, *lock-in*, biaya berulang, lokasi data, dan strategi keluar.
- **Hybrid cloud** sering menjadi jalan tengah yang masuk akal ketika sebagian beban kerja perlu fleksibel, tetapi sebagian lain perlu kontrol lebih ketat.

Pilihan ini tidak boleh dibuat hanya karena “semua orang pindah ke cloud” atau “lebih aman kalau semua di dalam sendiri”. Desain yang sehat justru harus mulai dari:

- karakter beban kerja;
- pola akses;
- sensitivitas data;
- kebutuhan integrasi;
- kesiapan SDM;
- kebutuhan keberlanjutan.

Dari sana barulah dipilih model yang paling masuk akal. Jadi, sebelum KAK dan spesifikasi ditulis, organisasi seharusnya sudah bisa menjelaskan mengapa model penempatan tertentu dipilih, apa risiko utamanya, dan bagaimana mitigasinya.

6.9 *Backup*, *disaster recovery*, *high availability*, dan *observability*

Banyak proyek IT hanya sibuk memikirkan cara sistem hidup, tetapi tidak cukup serius memikirkan bagaimana sistem tetap hidup saat gangguan terjadi. Karena itu, desain harus menutup empat unsur penting: **backup**, **disaster recovery**, **high availability**, dan **observability**.

Keempat unsur ini menjawab pertanyaan yang berbeda:

- **Backup**: apakah data bisa dipulihkan?
- **Disaster recovery**: bagaimana layanan dipulihkan saat terjadi gangguan besar?
- **High availability**: bagaimana layanan tetap tersedia saat sebagian komponen gagal?
- **Observability**: apakah tim benar-benar bisa melihat kondisi sistem, mendeteksi anomali, dan menemukan akar masalah dengan cepat?

Dalam SPBE, layanan pemerintah diarahkan untuk dikelola secara terpadu, dipantau, dievaluasi, dan ditingkatkan kualitasnya secara berkelanjutan. Karena itu, sistem pemerintah memang tidak boleh didesain hanya untuk keadaan normal. Ia harus siap menghadapi perubahan, gangguan, dan kebutuhan pemulihan. (Perpres 95 Tahun 2018 PDF) (Peraturan BPK)

Dalam pengadaan, empat hal ini harus diterjemahkan ke ukuran yang jelas, seperti:

- frekuensi backup;
- retensi;
- target recovery time;
- target recovery point;
- kebutuhan redundansi;
- monitoring log dan metrik;
- alerting;
- kewajiban dokumentasi dan uji pemulihan.

Kalau tidak, semua orang akan berkata “sistem aman dan andal”, tetapi tidak ada standar nyata untuk membuktikannya. Ini merupakan inferensi praktis yang kuat dari kewajiban tata kelola SPBE yang terencana, dipantau, dan dievaluasi. (Perpres 95 Tahun 2018) (Peraturan BPK)

6.10 Dokumen desain minimal yang wajib ada sebelum pengadaan

Bagian ini perlu ditutup dengan pertanyaan praktis: **dokumen apa minimal yang harus ada sebelum pengadaan dimulai?** Ini penting karena banyak organisasi sebenarnya tahu apa yang diinginkan, tetapi belum mengubah pengetahuan itu menjadi dokumen yang cukup kuat untuk mengendalikan tender dan kontrak.

Setidaknya, sebelum pengadaan IT yang cukup signifikan dimulai, organisasi idealnya memiliki:

- ringkasan masalah dan tujuan layanan;
- peta proses bisnis inti;
- gambaran kondisi *as-is* dan target *to-be*;
- peta aktor dan peran;
- rancangan arsitektur data, aplikasi, infrastruktur, dan keamanan pada tingkat yang memadai;
- kebutuhan integrasi dan pertukaran data;
- pilihan model penempatan sistem;
- kebutuhan backup, DR, HA, dan monitoring;
- asumsi volume dan kapasitas;
- daftar risiko utama dan mitigasi awal.

Semua ini tidak harus berbentuk dokumen yang terlalu akademik, tetapi harus cukup jelas untuk menjadi dasar KAK dan spesifikasi. Prinsip ini sejalan dengan **Pedoman Perencanaan Pengadaan LKPP**, yang menempatkan perencanaan sebagai dasar agar pengadaan selaras dengan tujuan dan kebutuhan riil instansi. (Peraturan LKPP Nomor 11 Tahun 2021) (JDIH LKPP)

Dari sudut pengelolaan kontrak, dokumentasi yang baik juga penting untuk mutu dan audit. Karena itu, disiplin desain dokumen sejak awal akan sangat membantu sampai tahap pelaksanaan dan serah terima. Prinsip akhirnya sederhana: **tender yang baik lahir dari desain yang cukup matang, dan desain yang matang selalu meninggalkan dokumen yang bisa dibaca, diuji, dan dipertanggungjawabkan**. Pembacaan ini merupakan sintesis dari pedoman perencanaan LKPP, arsitektur SPBE, dan kewajiban akuntabilitas pemrosesan data. (Peraturan LKPP Nomor 11 Tahun 2021, Perpres 95 Tahun 2018, UU 27 Tahun 2022) (JDIH LKPP)

Penutup

Bab ini menegaskan bahwa **desain sistem adalah jembatan antara kebutuhan organisasi dan dokumen pengadaan**. Tanpa desain, KAK dan spesifikasi akan lemah, vendor mudah menafsirkan sesukanya, risiko integrasi membesar, keamanan dibahas terlambat, dan biaya mudah melonjak. Sebaliknya, bila organisasi lebih dulu membangun arsitektur bisnis, data, aplikasi, infrastruktur, keamanan, integrasi, model penempatan, serta strategi ketahanan operasional yang cukup jelas, maka pengadaan bisa bergerak dengan lebih disiplin, lebih kompetitif, dan lebih siap audit. Dalam kerangka SPBE, inilah cara agar belanja IT tidak hanya membeli teknologi, tetapi benar-benar membangun layanan digital yang terintegrasi, aman, dan berkelanjutan. (Perpres 95 Tahun 2018, Perpres 132 Tahun 2022, UU 27 Tahun 2022, Peraturan LKPP Nomor 11 Tahun 2021) (JDIH LKPP)

Tabel Summary. Desain Sistem sebagai Dasar KAK dan Spesifikasi

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
6.1 Mengapa desain sistem harus muncul sebelum tender	Tender IT yang dimulai tanpa desain sistem yang cukup jelas hampir selalu melahirkan spesifikasi kabur, ruang lingkup yang mudah bergeser, dan konflik tafsir antara pengguna dan penyedia. Dalam kerangka SPBE, desain sistem adalah fondasi, bukan pelengkap.	Sebelum menyusun KAK dan spesifikasi, instansi perlu menyiapkan **kerangka desain sistem** yang cukup untuk mengendalikan mutu, biaya, integrasi, keamanan, dan hasil kerja.
6.2 Arsitektur bisnis	Arsitektur bisnis menjelaskan proses, aktor, keputusan, layanan, dan keluaran yang menjadi dasar sistem. Tanpa ini, proyek IT mudah hanya mendigitalisasi kekacauan lama.	KAK harus bertumpu pada **peta proses bisnis**, titik keputusan, peran aktor, dan target perbaikan layanan, bukan hanya daftar fitur aplikasi.
6.3 Arsitektur data	Arsitektur data menjawab apa data yang dipakai, dari mana asalnya, siapa pemiliknya, bagaimana kualitas dan relasinya, serta bagaimana akses dan perubahannya dikendalikan. Ini sangat penting untuk integritas layanan dan audit.	Sebelum tender, organisasi perlu mendefinisikan **data master, sumber kebenaran, pola integrasi data, validasi, kontrol akses, dan jejak perubahan**, terutama untuk data pribadi.
6.4 Arsitektur aplikasi	Arsitektur aplikasi mencegah lahirnya aplikasi silo dengan memetakan modul, fungsi inti, dependensi, dan hubungan antara aplikasi baru dan lama.	Pengadaan aplikasi baru harus diuji terhadap pertanyaan: **apakah fungsi ini sudah ada, apakah perlu aplikasi baru, dan bagaimana integrasinya dengan ekosistem yang sudah berjalan**.
6.5 Arsitektur infrastruktur	Infrastruktur adalah fondasi daya tahan sistem. Kebutuhan kapasitas, jaringan, penyimpanan, segmentasi, lokasi penempatan, dan *monitoring* harus dipetakan sejak awal agar spesifikasi tidak salah arah.	Spesifikasi infrastruktur sebaiknya ditulis berbasis **kapasitas, performa, kompatibilitas, ketersediaan, dan pertumbuhan beban**, bukan berbasis merek tertentu.

6.6 Arsitektur keamanan	Keamanan harus menjadi bagian dari desain awal, bukan tambahan di belakang. Struktur akses, logging, enkripsi, segmentasi, *patching*, dan audit trail harus dipikirkan sebelum tender.	KAK dan spesifikasi harus meminta **kontrol keamanan yang dapat diuji**, termasuk autentikasi, kontrol berbasis peran, logging, enkripsi, dan kebutuhan audit keamanan.
6.7 Integrasi, API, *message bus*, dan pertukaran data	Sistem modern harus dirancang untuk terhubung. Integrasi tidak cukup ditulis sebagai slogan; ia harus diterjemahkan ke rancangan teknis dan tata kelola pertukaran data yang jelas.	Dokumen pengadaan harus menjelaskan **sumber data utama, pola pertukaran data, kebutuhan API, autentikasi antarsistem, skema kegagalan, dan dokumentasi integrasi**.
6.8 *On-premise*, *colocation*, cloud, dan *hybrid cloud*	Model penempatan sistem adalah keputusan strategis yang mempengaruhi biaya, kontrol, fleksibilitas, keamanan, dan ketergantungan vendor. Tidak boleh dipilih hanya berdasarkan tren.	Sebelum tender, instansi perlu menetapkan **alasan pemilihan model penempatan**, risiko utamanya, dan strategi mitigasinya berdasarkan karakter beban kerja dan sensitivitas data.
6.9 *Backup*, *disaster recovery*, *high availability*, dan *observability*	Sistem tidak cukup hanya hidup; ia harus tetap bisa dipulihkan, tetap tersedia saat gangguan, dan bisa dipantau dengan baik. Empat hal ini menentukan ketahanan operasional layanan digital.	Pengadaan harus memuat kebutuhan yang jelas tentang **backup, target pemulihan, redundansi, monitoring, alerting, serta uji pemulihan** agar klaim keandalan dapat dibuktikan.
6.10 Dokumen desain minimal yang wajib ada sebelum pengadaan	Desain yang matang harus meninggalkan dokumen yang cukup untuk menjadi dasar KAK, spesifikasi, HPS, kontrak, dan audit. Tanpa dokumen ini, tender berjalan di atas asumsi.	Minimal harus ada **ringkasan tujuan layanan, peta proses, *as-is*/*to-be*, rancangan arsitektur inti, kebutuhan integrasi, model penempatan, kebutuhan ketahanan operasional, asumsi kapasitas, dan risiko utama**.

Ringkasan

Bab 6 menegaskan bahwa desain sistem adalah dasar substantif dari KAK dan spesifikasi. Tanpa desain yang cukup matang, pengadaan IT mudah menghasilkan spesifikasi kabur, integrasi yang lemah, keamanan yang tertunda, dan biaya yang membesar di tengah jalan. Sebaliknya, jika organisasi lebih dulu menata arsitektur bisnis, data, aplikasi, infrastruktur, keamanan, integrasi, model penempatan, serta strategi ketahanan operasional, maka tender akan jauh lebih terarah, kompetitif, dan siap audit.

Bagian III. KAK, Spesifikasi, HPS, dan Kontrak

Bab 7. Menulis KAK yang Benar untuk Pengadaan IT

7.1 Fungsi KAK dalam siklus PBJ

Dalam pengadaan IT, **KAK bukan dokumen pelengkap**. KAK adalah penghubung antara masalah organisasi, desain sistem, strategi pengadaan, spesifikasi teknis, HPS, metode pemilihan, hingga kontrak. Karena itu, mutu KAK sangat menentukan mutu seluruh siklus PBJ.

Secara regulatif, posisi ini sejalan dengan **Peraturan LKPP Nomor 11 Tahun 2021** tentang pedoman perencanaan pengadaan, serta **Peraturan LKPP Nomor 12 Tahun 2021** tentang pelaksanaan pengadaan melalui penyedia yang kemudian diubah oleh **Peraturan LKPP Nomor 4 Tahun 2024**. Artinya, KAK tidak boleh dipandang sebagai formalitas yang ditulis belakangan, melainkan sebagai instrumen inti agar proses pengadaan bergerak konsisten dari tahap perencanaan ke tahap pelaksanaan. (Peraturan LKPP No. 11 Tahun 2021, Peraturan LKPP No. 12 Tahun 2021, Peraturan LKPP No. 4 Tahun 2024) (JDIH LKPP)

Dalam praktik, KAK yang lemah biasanya menimbulkan tiga masalah sekaligus:

- penyedia menafsirkan kebutuhan terlalu sempit;
- tim internal berharap hasil yang lebih luas daripada yang tertulis;
- auditor sulit menilai apakah hasil pekerjaan benar-benar sesuai kebutuhan awal.

Karena itu, KAK yang baik harus cukup kuat untuk menjadi **sumber kebenaran operasional** selama pengadaan berlangsung.

7.2 KAK harus mulai dari masalah dan output, bukan merek

Prinsip terpenting dalam menulis KAK pengadaan IT adalah: **mulai dari masalah, kebutuhan layanan, dan output yang diinginkan, bukan dari merek, tipe, atau preferensi vendor**.

Dalam praktik, godaan untuk menulis KAK dari sisi produk memang besar. Instansi sering merasa sudah tahu ingin membeli apa, lalu menyusun dokumen seolah pilihan teknologinya sudah final. Cara seperti ini berbahaya karena:

- mengunci kompetisi terlalu dini;
- menyempitkan ruang alternatif solusi;
- melemahkan justifikasi pengadaan.

Pendekatan yang lebih sehat adalah memulai dari pertanyaan mendasar:

- layanan apa yang bermasalah;
- proses apa yang lambat;
- data apa yang tidak sinkron;
- risiko apa yang harus ditekan;
- kemampuan apa yang ingin dibangun.

Setelah itu, barulah kebutuhan diterjemahkan ke dalam:

- output;
- fungsi;
- performa;
- integrasi;
- keamanan;
- dukungan.

Urutannya penting. Jika dibalik, KAK bisa tampak teknis, tetapi sebenarnya miskin alasan dan miskin arah.

7.3 Struktur KAK yang ideal

KAK pengadaan IT yang baik seharusnya tidak terlalu tipis, tetapi juga tidak berubah menjadi esai panjang yang sulit dipakai. Struktur idealnya harus cukup untuk menjelaskan:

- mengapa proyek perlu dilakukan;
- hasil apa yang diinginkan;
- ruang lingkupnya apa;
- batasannya di mana;
- tahapan kerjanya bagaimana;
- risikonya apa;
- keberhasilannya diukur dengan apa.

Secara praktis, struktur KAK yang ideal minimal memuat:

- latar belakang dan urgensi;
- tujuan umum dan tujuan khusus;
- gambaran masalah dan kondisi saat ini;
- ruang lingkup dan batas pekerjaan;
- output, outcome, dan indikator manfaat;
- tahapan pekerjaan;
- jadwal dan milestone;
- kebutuhan integrasi dan keamanan;
- risiko dan mitigasi;
- asumsi penting;
- kewajiban dokumentasi;
- kriteria hasil kerja yang akan diuji.

Struktur seperti ini membuat KAK tidak hanya menjelaskan **apa yang dikerjakan**, tetapi juga **mengapa, untuk siapa, dan dengan standar seperti apa**.

KAK juga harus sinkron dengan spesifikasi, HPS, dan rancangan kontrak. Jika empat dokumen ini tidak nyambung, masalah biasanya muncul kemudian:

- spesifikasi menuntut hal yang tidak ada di KAK;
- HPS tidak mencerminkan ruang lingkup nyata;
- kontrak memuat kewajiban yang tidak pernah dirumuskan sejak awal.

7.4 Latar belakang dan urgensi

Bagian latar belakang sering diremehkan, padahal justru di sinilah organisasi menjelaskan alasan keberadaan proyek. Latar belakang yang baik tidak cukup berhenti pada kalimat umum seperti “untuk mendukung transformasi digital” atau “untuk meningkatkan pelayanan”. Rumusan seperti itu terlalu kabur.

Latar belakang yang lebih kuat harus menjelaskan masalah nyata, misalnya:

- sistem lama tidak terintegrasi;
- proses terlalu manual;
- beban layanan meningkat;
- data tidak konsisten;
- risiko keamanan membesar;
- tuntutan kepatuhan makin tinggi.

Urgensi juga perlu dijelaskan secara substantif. Misalnya:

- apakah ada risiko gangguan layanan bila sistem tidak diperbarui;
- apakah ada tenggat regulatif;
- apakah ada kebutuhan interoperabilitas dengan aplikasi umum SPBE;
- apakah ada risiko pelindungan data yang makin besar.

Dalam kerangka SPBE, layanan digital pemerintah memang diarahkan pada keterpaduan proses bisnis dan layanan pemerintahan. Karena itu, latar belakang proyek IT pemerintah tidak boleh dilepaskan dari tata kelola digital yang lebih luas. (Perpres No. 95 Tahun 2018 tentang SPBE) (Peraturan BPK)

Latar belakang yang baik membantu dua hal sekaligus:

- membuat KAK lebih mudah dipahami oleh pimpinan, auditor, dan pihak nonteknis;
- memberi dasar yang kuat untuk menolak perluasan ruang lingkup yang tidak relevan di tengah jalan.

7.5 Tujuan umum dan tujuan khusus

Setelah latar belakang, KAK harus menjabarkan **tujuan umum** dan **tujuan khusus**. Banyak KAK berhenti pada tujuan yang terlalu abstrak, misalnya “membangun sistem informasi terintegrasi”. Ini belum cukup.

Tujuan umum memang boleh bersifat strategis. Tetapi tujuan khusus harus lebih operasional, misalnya:

- mempercepat proses persetujuan;
- mengurangi input data berulang;
- meningkatkan ketersediaan layanan;
- menyediakan *audit trail*;
- memperkuat pelindungan data;
- memungkinkan integrasi dengan sistem lain.

Perbedaan antara tujuan umum dan tujuan khusus penting karena akan mempengaruhi:

- spesifikasi;
- indikator manfaat;
- uji terima;
- evaluasi hasil kerja.

Tujuan umum memberi arah besar. Tujuan khusus memberi pegangan operasional. Karena itu, tujuan harus ditulis dengan bahasa yang:

- terukur;
- realistis;
- terkait langsung dengan manfaat layanan.

Tujuan bukan slogan. Ia adalah **kompas kerja**.

7.6 Ruang lingkup dan batas pekerjaan

Salah satu fungsi paling penting dari KAK adalah menetapkan **ruang lingkup** dan **batas pekerjaan**. Dalam proyek IT, bagian ini sangat krusial karena ruang lingkup paling mudah melebar.

Ruang lingkup yang baik harus menjelaskan apa saja yang termasuk, misalnya:

- analisis;
- desain;
- pengembangan;
- pengadaan perangkat;
- konfigurasi;
- integrasi;
- migrasi data;
- pengujian;
- pelatihan;
- dokumentasi;
- masa stabilisasi;
- dukungan awal.

Tetapi sama pentingnya, KAK juga harus menjelaskan apa yang **tidak termasuk**. Contohnya:

- modul tertentu akan dikembangkan pada fase berikutnya;
- integrasi dengan sistem tertentu belum masuk paket ini;
- operasi jangka panjang akan diadakan melalui kontrak terpisah.

Batas pekerjaan yang jelas membuat:

- HPS lebih sehat;
- jadwal lebih realistis;
- evaluasi hasil kerja lebih adil.

Dalam proyek IT, *scope management* yang lemah hampir selalu berujung pada ekspektasi yang liar dan konflik yang mahal.

7.7 Output, outcome, dan indikator manfaat

KAK pengadaan IT yang matang harus membedakan dengan jelas antara **output**, **outcome**, dan **indikator manfaat**.

Secara sederhana:

- **Output** adalah hasil langsung pekerjaan, misalnya aplikasi aktif, server terpasang, dashboard tersedia, dokumen desain selesai, pelatihan terlaksana.
- **Outcome** adalah perubahan yang dihasilkan, misalnya proses lebih cepat, integrasi berjalan, kualitas data membaik, gangguan berkurang.
- **Indikator manfaat** adalah ukuran yang membantu membuktikan apakah perubahan itu benar-benar terjadi.

Kesalahan yang paling sering terjadi adalah hanya menulis output teknis, lalu menganggap proyek selesai ketika semua komponen telah dikirim. Padahal layanan bisa saja tetap buruk.

Karena itu, KAK sebaiknya berani menulis indikator yang lebih substantif, misalnya:

- waktu proses turun sekian persen;
- jumlah entri ganda menurun;
- *uptime* mencapai target tertentu;
- jumlah temuan audit berkurang.

Pendekatan ini sejalan dengan semangat SPBE yang berorientasi pada kualitas layanan, bukan sekadar keberadaan aplikasi. (Perpres No. 95 Tahun 2018) (JDIH LKPP)

7.8 Tahapan pekerjaan

KAK pengadaan IT sebaiknya menjelaskan **tahapan pekerjaan** secara eksplisit. Proyek digital hampir tidak pernah sehat jika diperlakukan sebagai satu blok kerja yang homogen.

Biasanya, tahap pekerjaan mencakup:

- persiapan;
- analisis detail;
- desain;
- pengembangan atau konfigurasi;
- integrasi;
- migrasi data;
- pengujian;
- pelatihan;
- *go-live*;
- stabilisasi.

Tahapan yang jelas membantu memecah risiko. Misalnya:

- hasil analisis harus disetujui sebelum pengembangan lanjut;
- integrasi diuji sebelum migrasi besar dilakukan;
- pelatihan dilakukan sebelum serah terima akhir.

Dengan cara ini, organisasi tidak dipaksa menerima hasil yang belum matang hanya karena semua pekerjaan dibungkus dalam satu jadwal umum.

Bagi penyedia, KAK yang bertahap juga lebih adil, karena:

- ekspektasi per fase jelas;
- deliverable per fase jelas;
- titik evaluasi lebih transparan.

7.9 Jadwal dan milestone

Di banyak KAK, jadwal ditulis terlalu umum, misalnya hanya “pekerjaan selesai dalam 180 hari.” Untuk proyek IT, itu tidak cukup. KAK perlu memuat **jadwal** dan **milestone** yang lebih rinci agar organisasi dapat mengendalikan ritme pekerjaan dan mendeteksi keterlambatan lebih awal.

Milestone penting karena memberi titik ukur nyata, misalnya:

- kapan desain selesai;
- kapan lingkungan uji siap;
- kapan integrasi mulai diuji;
- kapan pelatihan dilakukan;
- kapan masa stabilisasi dimulai.

Jadwal yang baik juga harus realistis. Dalam proyek IT, tekanan waktu yang berlebihan sering dibayar dengan:

- mutu yang turun;
- dokumentasi yang lemah;
- masalah yang hanya dipindahkan ke masa operasi.

Karena itu, jadwal harus mempertimbangkan:

- kompleksitas integrasi;
- kesiapan data;
- waktu validasi pengguna;
- masa uji yang memadai.

Jadwal yang baik bukan jadwal yang sekadar terlihat cepat, tetapi jadwal yang masih memberi ruang untuk kualitas, perbaikan, dan pembuktian hasil.

7.10 Resiko dan mitigasi

KAK yang baik tidak hanya menceritakan pekerjaan ideal, tetapi juga mengakui bahwa proyek IT mengandung resiko. Karena itu, bagian **resiko dan mitigasi** seharusnya ditulis secara eksplisit.

Risiko yang umum antara lain:

- kebutuhan yang belum stabil;
- integrasi yang rumit;
- keterbatasan data;
- kesiapan pengguna yang rendah;
- risiko keamanan;
- ketergantungan pada pihak ketiga;
- potensi keterlambatan.

Untuk sistem yang memproses data pribadi, dimensinya bahkan memiliki konsekuensi hukum. **UU No. 27 Tahun 2022 tentang Pelindungan Data Pribadi** mengatur kewajiban pengendali data pribadi dan memuat kewajiban pemberitahuan dalam hal kegagalan pelindungan data pribadi. Karena itu, sejak KAK pun organisasi sudah perlu memikirkan:

- jenis risiko data;
- kebutuhan kontrol;
- tanggung jawab penanganan;
- mekanisme pemantauan risiko. (UU No. 27 Tahun 2022, PDF UU No. 27 Tahun 2022) (Peraturan BPK)

Bagian mitigasi tidak harus panjang, tetapi harus operasional:

- apa risikonya;
- siapa penanggung jawabnya;
- bagaimana cara mengendalikannya;
- bagaimana cara memantaunya.

KAK yang jujur soal risiko biasanya justru lebih kuat daripada KAK yang berpura-pura semua akan berjalan mulus.

7.11 KAK untuk *hardware*

KAK untuk *hardware* harus tetap dimulai dari fungsi layanan, bukan dari merek. Misalnya:

- kebutuhan komputasi untuk aplikasi tertentu;
- kapasitas penyimpanan;
- kebutuhan redundansi;
- kebutuhan jaringan;
- kebutuhan *endpoint* untuk unit layanan.

Jadi, mulailah dari:

- kapasitas;
- performa;
- kompatibilitas;
- keandalan;
- lingkungan operasi.

Dalam pengadaan *hardware*, KAK juga perlu menjelaskan konteks operasional, seperti:

- dipakai di pusat data atau kantor biasa;
- perlu rack atau tidak;
- perlu dual power supply atau tidak;
- butuh manajemen jarak jauh atau tidak;
- kebutuhan garansi;
- ketersediaan suku cadang;
- kebutuhan instalasi.

Dengan begitu, spesifikasi nanti turun dari kebutuhan nyata, bukan dari brosur pabrikan.

KAK *hardware* yang baik juga sebaiknya memuat:

- kewajiban dokumentasi konfigurasi;
- daftar aset;
- hasil uji fungsi;
- bila relevan, integrasi dengan sistem yang sudah ada.

7.12 KAK untuk *software*

Untuk *software*, KAK harus lebih fokus pada:

- fungsi bisnis;
- integrasi;
- data;
- keamanan;
- deliverable;

bukan sekadar daftar menu.

Dalam proyek perangkat lunak, masalah terbesar sering muncul ketika dokumen kebutuhan terlalu dangkal: hanya menyebut modul, tetapi tidak menjelaskan proses, aktor, data, kondisi pengecualian, dan hasil yang diharapkan.

KAK *software* sebaiknya minimal memuat:

- tujuan sistem;
- fungsi inti;
- profil pengguna;
- kebutuhan integrasi;
- kebutuhan data;
- lingkungan pengembangan dan produksi;
- kebutuhan keamanan;
- dokumentasi;
- pengujian;
- pelatihan;
- masa stabilisasi.

Jika sistem menyentuh data pribadi, aspek pelindungan data harus masuk sejak awal sebagai kebutuhan desain dan operasional, sesuai kerangka UU PDP. (UU No. 27 Tahun 2022) (Peraturan BPK)

Untuk pendekatan *open source*, KAK juga bisa menegaskan:

- kebutuhan repositori kode;
- dokumentasi instalasi;
- lisensi yang jelas;
- kompatibilitas;
- hak akses admin.

7.13 KAK untuk jasa pengembangan

Pada jasa pengembangan, KAK harus lebih rinci dalam menjelaskan ruang lingkup kerja intelektual dan teknis. Ini mencakup:

- analisis kebutuhan;
- desain;
- pengembangan;
- pengujian;
- dokumentasi;
- deployment;
- pelatihan;
- perbaikan bug;
- masa stabilisasi.

Banyak konflik proyek muncul karena KAK hanya menulis “pengembangan aplikasi” tanpa menjelaskan standar hasil kerja di setiap tahap.

KAK jasa pengembangan juga sebaiknya menjelaskan *way of working*, misalnya:

- kebutuhan workshop;
- validasi berkala;
- demo periodik;
- pengelolaan perubahan;
- mekanisme persetujuan.

Tanpa itu, proyek mudah tergelincir menjadi pola “vendor bekerja sendiri, pengguna baru melihat hasil di akhir”, yang hampir selalu mahal secara koreksi.

Yang tidak kalah penting, KAK harus menegaskan *deliverable* non-fisik, seperti:

- dokumen analisis;
- dokumen desain;
- *source code*;
- skrip deployment;
- hasil pengujian;
- manual admin;
- manual pengguna;
- materi pelatihan;
- berita acara tiap fase.

Dalam proyek digital, justru inilah warisan penting yang menentukan apakah institusi nanti benar-benar memiliki hasilnya.

7.14 KAK untuk *managed service*

KAK untuk *managed service* berbeda karena inti yang dibeli bukan pembangunan satu kali, melainkan **layanan berkelanjutan**. Karena itu, fokusnya harus pada:

- cakupan layanan;
- jam layanan;
- parameter performa;
- waktu respons;
- waktu pemulihan;
- pelaporan;
- eskalasi;
- pengelolaan insiden;
- keamanan operasional;
- tata kelola.

Dalam model ini, indikator kinerja menjadi sangat penting. Karena itu, KAK *managed service* harus cukup tajam untuk menjawab:

- layanan apa yang dikelola;
- batas tanggung jawab penyedia sampai di mana;
- target SLA apa yang wajib dipenuhi;
- bagaimana pembuktiannya;
- apa konsekuensinya jika target tidak tercapai.

Jika semua ini tidak ditulis sejak KAK, kontrak layanan akan cenderung kabur dan sulit dikendalikan.

7.15 KAK untuk *cloud/SaaS*

KAK untuk *cloud/SaaS* harus lebih hati-hati karena yang dibeli sering bukan kepemilikan aset, melainkan **akses layanan**. Karena itu, fokus KAK harus pada:

- jenis layanan;
- ketersediaan;
- lokasi data atau *region* bila relevan;
- keamanan;
- enkripsi;
- backup;
- integrasi;
- portabilitas data;
- hak akses admin;
- *exit plan* bila layanan dihentikan atau dipindah.

Pendekatan ini penting untuk mencegah *vendor lock-in*. Jika KAK *cloud/SaaS* hanya bicara kapasitas dan biaya bulanan, organisasi bisa terjebak pada platform yang sulit ditinggalkan.

Karena itu, sejak awal KAK harus menanyakan:

- bagaimana data diekspor;
- bagaimana konfigurasi didokumentasikan;
- bagaimana transisi dilakukan bila kontrak berakhir.

Jika layanan memproses data pribadi, kebutuhan pelindungan data dan tanggung jawab saat terjadi insiden harus ditulis lebih tegas, karena ini bukan hanya isu teknis, tetapi juga isu akuntabilitas menurut UU PDP. (UU No. 27 Tahun 2022) (Peraturan BPK)

7.16 KAK untuk LMS/*e-learning*

Untuk LMS/*e-learning*, KAK harus menekankan bahwa yang dibeli bukan hanya platform, tetapi **ekosistem pembelajaran digital**. Karena itu, ruang lingkupnya tidak cukup hanya menjelaskan modul pengguna dan materi.

KAK perlu memuat kebutuhan seperti:

- manajemen kelas;
- konten;
- video;
- kuis;
- ujian;
- bank soal;
- analitik pembelajaran;
- integrasi dengan sistem akademik atau kepegawaian;
- *mobile access*;
- dukungan bandwidth;
- *helpdesk*.

Karena LMS sering memproses data peserta, dosen, guru, pegawai, atau siswa, aspek privasi dan pengelolaan akun juga harus masuk dalam KAK. Selain itu, untuk lembaga pendidikan atau pelatihan, stabilitas layanan saat periode ujian atau pelatihan massal perlu dijelaskan sejak awal agar kapasitas dan dukungan tidak salah hitung.

KAK LMS yang baik akan membedakan dengan jelas antara:

- kebutuhan platform;
- migrasi konten;
- integrasi;
- pelatihan admin/pengajar;
- dukungan operasional.

7.17 KAK untuk *e-commerce/marketplace*

KAK untuk *e-commerce/marketplace* harus lebih luas daripada sekadar etalase digital. Sistem seperti ini menyentuh banyak hal sekaligus, antara lain:

- katalog produk;
- transaksi;
- pembayaran;
- log aktivitas;
- promosi;
- pengelolaan penjual;
- pengelolaan pembeli;
- keamanan transaksi;
- integrasi pihak ketiga;
- kontrol fraud.

Karena itu, KAK *e-commerce* harus memperjelas:

- kewajiban log transaksi;
- bukti elektronik;
- pengelolaan data pengguna;
- keamanan pembayaran;
- tanggung jawab jika terjadi insiden.

Jika sistem memproses data pribadi, maka kewajiban pelindungan data tetap harus diperhatikan sejak tahap KAK, sesuai UU PDP. (UU No. 27 Tahun 2022) (Peraturan BPK)

Dengan kata lain, KAK *e-commerce* yang baik harus ditulis dari perspektif:

- operasi bisnis;
- kepercayaan pengguna;
- keamanan;
- jejak bukti.

Bukan hanya dari sisi tampilan antarmuka.

7.18 Checklist KAK yang siap tender

Bab ini layak ditutup dengan **checklist KAK yang siap tender**. Sebuah KAK pengadaan IT dapat dianggap siap bila setidaknya menjawab pertanyaan berikut:

- apakah masalah dan urgensinya jelas;
- apakah tujuan umum dan tujuan khususnya jelas;
- apakah ruang lingkup dan batas pekerjaan sudah tegas;
- apakah output, outcome, dan indikator manfaat sudah ditulis;
- apakah tahapan, jadwal, dan milestone masuk akal;
- apakah risiko utama dan mitigasinya sudah dikenali;
- apakah kebutuhan integrasi, keamanan, dan data sudah masuk;
- apakah deliverable dan bukti hasil kerja cukup rinci;
- apakah dokumen ini sudah selaras dengan spesifikasi, HPS, dan rancangan kontrak.

Checklist ini tampak sederhana, tetapi sangat strategis. Ia membantu tim membedakan antara:

- KAK yang hanya “selesai ditulis”;
- dan KAK yang benar-benar siap dipakai untuk mengendalikan tender.

Dalam pengadaan IT, perbedaan ini sangat besar. KAK yang matang akan:

- mengurangi sengketa;
- memperbaiki kualitas kompetisi;
- memperbesar peluang proyek benar-benar berhasil.

Penutup

Bab ini menegaskan bahwa **KAK adalah alat kendali utama dalam pengadaan IT**. KAK yang baik harus lahir dari:

- masalah nyata;
- desain sistem yang cukup matang;
- tujuan yang terukur;
- ruang lingkup yang disiplin;
- tahapan kerja yang jelas;
- risiko yang diakui;
- deliverable yang bisa dibuktikan.

Ia juga harus cukup kuat untuk menjaga agar proses pengadaan tetap:

- patuh aturan;
- tidak diskriminatif;
- berorientasi pada output;
- siap diaudit.

Jika KAK ditulis dengan benar, maka spesifikasi, HPS, evaluasi, kontrak, dan pelaksanaan akan jauh lebih sinkron. Sebaliknya, jika KAK lemah, dokumen lain biasanya hanya menjadi tambalan atas masalah yang sudah lahir sejak awal. Posisi KAK dalam siklus ini sejalan dengan pedoman perencanaan dan pedoman pelaksanaan pengadaan LKPP. (Peraturan LKPP No. 11 Tahun 2021, Peraturan LKPP No. 12 Tahun 2021, Peraturan LKPP No. 4 Tahun 2024) (JDIH LKPP)

Tabel Summary. Menulis KAK yang Benar untuk Pengadaan IT

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
7.1 Fungsi KAK dalam siklus PBJ	**KAK adalah dokumen inti**, bukan pelengkap. KAK menjembatani masalah organisasi, desain sistem, strategi pengadaan, spesifikasi, HPS, metode pemilihan, hingga kontrak.	KAK harus disusun setelah masalah, target layanan, arsitektur, dan strategi solusi cukup jelas, agar seluruh dokumen pengadaan tetap sinkron.
7.2 KAK harus mulai dari masalah dan output, bukan merek	KAK yang sehat dimulai dari **masalah, kebutuhan layanan, dan output**, bukan dari merek, tipe, atau vendor tertentu. Pendekatan berbasis merek terlalu dini berisiko mengunci kompetisi dan melemahkan justifikasi kebutuhan.	Tim perencana harus menulis kebutuhan dalam bahasa fungsi, manfaat, performa, integrasi, dan keamanan, lalu baru menurunkannya ke spesifikasi teknis.
7.3 Struktur KAK yang ideal	Struktur KAK harus cukup lengkap untuk menjelaskan **mengapa proyek perlu dilakukan, apa hasilnya, ruang lingkupnya, batasnya, risikonya, dan bagaimana keberhasilannya diukur**.	Gunakan struktur minimal: latar belakang, tujuan, kondisi saat ini, ruang lingkup, output/outcome, tahapan, jadwal, risiko, integrasi, keamanan, dokumentasi, dan indikator hasil.
7.4 Latar belakang dan urgensi	Latar belakang harus menjelaskan **masalah nyata dan alasan substantif** proyek dijalankan, bukan sekadar kalimat umum seperti “mendukung transformasi digital.”	Tulis urgensi berbasis fakta: proses terlalu manual, sistem tidak terintegrasi, risiko keamanan meningkat, kebutuhan regulasi, atau beban layanan yang naik.
7.5 Tujuan umum dan tujuan khusus	Tujuan umum memberi arah strategis, sedangkan tujuan khusus memberi pegangan operasional yang lebih terukur.	Rumuskan tujuan dengan bahasa yang bisa diterjemahkan ke spesifikasi, indikator manfaat, dan uji terima, misalnya percepatan proses, pengurangan input ganda, atau peningkatan uptime.

7.6 Ruang lingkup dan batas pekerjaan	KAK harus menjelaskan **apa yang termasuk dan apa yang tidak termasuk**. Tanpa batas yang jelas, ruang lingkup proyek IT mudah melebar dan sulit dikendalikan.	Tulis secara tegas komponen pekerjaan: analisis, desain, integrasi, migrasi, pelatihan, dokumentasi, stabilisasi, serta pengecualian yang tidak termasuk paket.
7.7 Output, outcome, dan indikator manfaat	KAK yang matang membedakan **output teknis**, **outcome layanan**, dan **indikator manfaat**. Proyek tidak boleh dianggap berhasil hanya karena barang atau aplikasi sudah dikirim.	Selain menyebut aplikasi/modul/server, KAK juga perlu menulis manfaat yang ingin dicapai seperti integrasi berjalan, waktu proses turun, atau kualitas data membaik.
7.8 Tahapan pekerjaan	Proyek IT perlu dipecah ke dalam tahap yang jelas: analisis, desain, pengembangan, integrasi, migrasi, pengujian, pelatihan, *go-live*, dan stabilisasi.	Tahapan membantu mengendalikan mutu, mengurangi risiko, dan mencegah organisasi menerima hasil yang belum matang.
7.9 Jadwal dan milestone	Jadwal tidak cukup hanya menyebut total durasi. KAK harus memuat **milestone** yang memungkinkan pengawasan progres dan deteksi dini keterlambatan.	Tetapkan titik ukur seperti selesai desain, siap uji integrasi, pelatihan, *go-live*, dan akhir masa stabilisasi agar proyek lebih terkendali.
7.10 Risiko dan mitigasi	KAK yang baik mengakui risiko utama proyek IT, termasuk integrasi, data, keamanan, keterlambatan, dan kesiapan pengguna.	Tambahkan daftar risiko utama beserta mitigasinya agar proyek lebih realistis dan lebih siap menghadapi perubahan atau gangguan.
7.11 KAK untuk hardware	KAK hardware harus berbasis **fungsi layanan, kapasitas, performa, kompatibilitas, dan keandalan**, bukan merek.	Tulis kebutuhan operasional seperti kapasitas, lingkungan penggunaan, garansi, instalasi, dokumentasi konfigurasi, dan hasil uji fungsi.

7.12 KAK untuk software	KAK software harus menekankan **fungsi bisnis, data, integrasi, keamanan, dokumentasi, dan pelatihan**, bukan sekadar daftar menu.	Pastikan KAK memuat kebutuhan pengguna, modul inti, integrasi, hak akses, keamanan, dokumentasi, pengujian, dan masa stabilisasi.
7.13 KAK untuk jasa pengembangan	Pada jasa pengembangan, KAK harus menjelaskan *deliverable* intelektual dan teknis secara rinci, bukan hanya menyebut “bangun aplikasi”.	Cantumkan dokumen analisis, desain, source code, skrip deployment, hasil uji, manual admin, manual pengguna, dan materi pelatihan sebagai hasil wajib.
7.14 KAK untuk managed service	Untuk *managed service*, yang dibeli adalah **layanan berkelanjutan**, sehingga fokusnya ada pada cakupan layanan, jam layanan, respons, pemulihan, pelaporan, dan SLA.	KAK harus menegaskan KPI, SLA, batas tanggung jawab, mekanisme eskalasi, dan pembuktian kinerja layanan.
7.15 KAK untuk cloud/SaaS	KAK cloud/SaaS harus berhati-hati karena yang dibeli biasanya **akses layanan**, bukan kepemilikan aset. Fokusnya adalah ketersediaan, portabilitas data, keamanan, hak akses, dan *exit plan*.	Tulis kebutuhan ekspor data, dokumentasi konfigurasi, lokasi data bila relevan, tanggung jawab insiden, dan strategi keluar untuk mengurangi *vendor lock-in*.
7.16 KAK untuk LMS/e-learning	LMS bukan sekadar platform, tetapi **ekosistem pembelajaran digital** yang mencakup kelas, konten, ujian, analitik, integrasi, dan dukungan pengguna.	Bedakan kebutuhan platform, migrasi konten, pelatihan admin/pengajar, integrasi sistem, dan dukungan operasional agar ruang lingkup lebih presisi.
7.17 KAK untuk e-commerce/marketplace	KAK e-commerce harus menekankan aspek transaksi, log aktivitas, keamanan, pembayaran, pengelolaan pengguna, dan kontrol fraud, bukan hanya tampilan etalase.	Pastikan KAK memuat kebutuhan bukti elektronik, keamanan transaksi, pengelolaan data pengguna, integrasi pihak ketiga, dan audit trail.

7.18 Checklist KAK yang siap tender	KAK siap tender bila masalahnya jelas, tujuannya terukur, ruang lingkupnya tegas, risikonya dikenali, *deliverable*-nya rinci, dan selaras dengan spesifikasi, HPS, serta kontrak.	Gunakan checklist ini sebagai *quality gate* sebelum tender agar tim bisa membedakan dokumen yang sekadar selesai ditulis dari dokumen yang benar-benar siap dipakai.

Ringkasan

Bab 7 menegaskan bahwa KAK adalah alat kendali utama dalam pengadaan IT. KAK yang benar harus lahir dari masalah nyata, tujuan yang jelas, ruang lingkup yang disiplin, tahapan kerja yang masuk akal, risiko yang diakui, dan *deliverable* yang dapat dibuktikan. Dengan KAK seperti itu, proses pengadaan akan lebih **patuh aturan, tidak diskriminatif, berorientasi output, dan siap audit**. Sebaliknya, bila KAK lemah, maka spesifikasi, HPS, evaluasi, dan kontrak biasanya hanya menjadi tambalan atas masalah yang sudah lahir sejak awal.

Bab 8. Menyusun Spesifikasi Teknis yang Tidak Diskriminatif

8.1 Prinsip spesifikasi berbasis fungsi dan kinerja

Dalam pengadaan IT, spesifikasi teknis yang baik harus disusun berdasarkan **fungsi, kinerja, kompatibilitas, keandalan, dan hasil layanan yang dibutuhkan**, bukan berdasarkan identitas produk tertentu. Prinsip ini penting untuk menjaga agar pengadaan tetap kompetitif, sehat, dan dapat dipertanggungjawabkan. Dalam praktik PBJ, pendekatan seperti ini sejalan dengan arah pedoman perencanaan dan pelaksanaan pengadaan LKPP, yang menempatkan kebutuhan sebagai dasar penyusunan dokumen pengadaan. (JDIH LKPP)

Pendekatan berbasis fungsi berarti organisasi menjelaskan **apa yang harus bisa dilakukan** oleh barang atau sistem. Sementara itu, pendekatan berbasis kinerja berarti organisasi menjelaskan **seberapa baik** barang atau sistem tersebut harus bekerja. Dalam konteks IT, ini dapat diterjemahkan ke dalam kebutuhan seperti:

- kapasitas komputasi;
- *throughput* jaringan;
- waktu respons aplikasi;
- jumlah pengguna simultan;
- target *uptime*;
- kebutuhan integrasi;
- kemampuan pemulihan layanan.

Dengan cara ini, spesifikasi tetap menjaga mutu, tetapi tidak mematikan persaingan. (JDIH LKPP)

Dari sudut tata kelola, spesifikasi yang diskriminatif adalah tanda bahaya. Jika spesifikasi diam-diam menguntungkan pemasok tertentu, maka ruang persaingan menyempit dan risiko penyimpangan ikut membesar. Karena itu, prinsip dasarnya sederhana: **spesifikasi harus cukup presisi untuk menjaga mutu, tetapi cukup terbuka untuk menjaga kompetisi**. Terlalu longgar membuat mutu tidak terkendali. Terlalu sempit membuat kompetisi mati. Ini adalah inferensi yang konsisten dengan prinsip umum pengadaan yang sehat dan tidak diskriminatif dalam kerangka PBJ. (JDIH LKPP)

8.2 Kapan penyebutan merek dibolehkan dan bagaimana membatasi risikonya

Dalam praktik, ada situasi ketika penyebutan merek atau produk tertentu sulit dihindari. Misalnya, ketika organisasi harus menjaga kompatibilitas dengan sistem yang sudah ada, ketika diperlukan suku cadang yang benar-benar spesifik, atau ketika lingkungan teknis yang berjalan memang hanya bisa mendukung kelas produk tertentu. Namun, penyebutan merek dalam pengadaan IT harus diperlakukan sebagai **pengecualian yang dijustifikasi**, bukan kebiasaan default. Prinsip kehati-hatian ini sejalan dengan arah umum regulasi PBJ yang menuntut kebutuhan disusun secara objektif dan dapat dipertanggungjawabkan. (JDIH LKPP)

Risiko utama penyebutan merek adalah:

- kompetisi menyempit;
- harga menjadi tidak sehat;
- inovasi tertutup;
- organisasi terjebak pada ketergantungan jangka panjang.

Karena itu, jika penyebutan merek memang tidak terhindarkan, risikonya perlu dibatasi dengan cara:

- menjelaskan alasan teknis dan operasional secara terbuka;
- menyebutkan parameter kompatibilitas atau kelas produk yang dibutuhkan;
- membuka ruang kesetaraan secara fungsional, bila memang memungkinkan dan dapat dibuktikan.

Yang paling penting, tim pengadaan harus jujur membedakan antara **kebutuhan kompatibilitas yang sah** dan **preferensi yang malas diuji**. Banyak penyebutan merek sebenarnya bukan kebutuhan teknis, melainkan kebiasaan lama. Di sinilah pentingnya desain sistem yang matang: jika arsitektur dan kebutuhan sudah jelas, kebutuhan riil akan lebih mudah dipisahkan dari preferensi vendor. (JDIH LKPP)

8.3 Menulis spesifikasi server, *storage*, jaringan, dan *endpoint*

Untuk komponen infrastruktur seperti server, *storage*, jaringan, dan *endpoint*, spesifikasi sebaiknya ditulis berbasis **kapasitas, performa, keandalan, kompatibilitas, lingkungan penggunaan, dan kemudahan pengelolaan**. Jangan mulai dari nama produk. Mulailah dari **beban layanan yang harus ditopang**. Pendekatan ini juga sejalan dengan semangat arsitektur SPBE yang menempatkan infrastruktur sebagai salah satu domain utama layanan digital pemerintah. (Peraturan BPK)

Untuk server, misalnya, kebutuhan dapat ditulis dalam bentuk:

- kelas prosesor;
- jumlah inti atau *logical compute*;
- kapasitas memori;
- kemampuan ekspansi;
- kebutuhan virtualisasi;
- redundansi daya;
- antarmuka jaringan;
- kompatibilitas dengan sistem operasi atau *hypervisor*;
- dukungan manajemen jarak jauh, bila memang dibutuhkan.

Untuk *storage*, fokus bisa diarahkan pada:

- kapasitas efektif;
- performa IOPS atau *throughput*;
- redundansi;
- kebutuhan *snapshot*;
- replikasi;
- integrasi dengan server atau virtualisasi;
- kebutuhan *backup*.

Untuk jaringan, spesifikasi bisa menekankan:

- jumlah port;
- kecepatan;
- dukungan VLAN;
- *routing*;
- keamanan;
- kemampuan manajemen;
- kompatibilitas dengan infrastruktur yang ada;
- kebutuhan *redundancy* atau *stacking*, bila relevan.

Sedangkan untuk *endpoint* seperti PC, laptop, atau perangkat kerja pengguna, spesifikasi harus mengacu pada profil beban kerja, misalnya:

- administrasi biasa;
- desain;
- analitik;
- operator layanan;
- mobilitas lapangan.

Dengan pendekatan ini, pengadaan tidak jatuh pada dua ekstrem: terlalu rendah sampai menghambat kerja, atau terlalu tinggi sampai boros. Ini adalah penerapan praktis dari spesifikasi berbasis fungsi dan kinerja. (JDIH LKPP)

8.4 Menulis spesifikasi *software* dan aplikasi

Untuk *software* dan aplikasi, spesifikasi tidak boleh berhenti pada daftar modul. Ia harus menjelaskan **fungsi bisnis, aktor pengguna, alur proses, data utama, integrasi, keamanan, kapasitas, dan hasil kerja yang diharapkan**. Banyak spesifikasi aplikasi tampak lengkap karena menuliskan banyak fitur, tetapi tetap lemah karena tidak menjelaskan konteks penggunaannya. Pendekatan ini selaras dengan arsitektur SPBE yang menghubungkan proses bisnis, data, aplikasi, infrastruktur, dan keamanan dalam satu kerangka layanan. (Peraturan BPK)

Spesifikasi aplikasi yang sehat setidaknya memuat:

- tujuan sistem;
- jenis pengguna;
- fungsi inti;
- kebutuhan *workflow*;
- kebutuhan pelaporan;
- kebutuhan integrasi;
- kebutuhan *audit trail*;
- kebutuhan kontrol akses;
- kebutuhan performa;
- kebutuhan dokumentasi;
- kebutuhan pengujian;
- hasil nonfungsional seperti kemudahan administrasi, pemeliharaan, dan interoperabilitas.

Dengan demikian, penyedia tidak hanya dinilai dari tampilan antarmuka, tetapi dari kemampuannya mendukung layanan secara nyata. (Peraturan BPK)

Spesifikasi aplikasi juga sebaiknya membedakan antara:

- kebutuhan wajib;
- kebutuhan penting;
- kebutuhan pengembangan bertahap.

Pembedaan ini penting agar paket tetap realistis dan tidak dipenuhi fitur yang bagus di atas kertas tetapi tidak pernah dipakai. Dalam pengadaan IT, kualitas sering justru lahir dari spesifikasi yang tajam dan relevan, bukan dari dokumen yang paling panjang. (JDIH LKPP)

8.5 Menulis spesifikasi *open source*: lisensi, repository, dokumentasi, kompatibilitas, audit kode

Jika organisasi memilih pendekatan *open source*, spesifikasi harus ditulis lebih cermat. Tidak cukup hanya menulis “menggunakan *open source*”. Spesifikasi perlu menjelaskan hal-hal yang benar-benar penting, seperti:

- jenis lisensi yang dapat diterima;
- repository kode;
- dokumentasi instalasi dan konfigurasi;
- kompatibilitas dengan sistem lain;
- bila relevan, kemampuan audit kode.

Pendekatan ini penting agar penggunaan OSS tidak berhenti sebagai label, tetapi benar-benar menjadi strategi pengadaan yang mendukung interoperabilitas, kontrol atas *deliverable*, dan penguatan kapasitas lokal. Ini merupakan inferensi yang masuk akal dari arah umum pengadaan dan strategi arsitektur terbuka dalam layanan digital. (JDIH LKPP)

Pada proyek OSS, kontrol atas *deliverable* sangat penting. Organisasi sebaiknya memastikan bahwa hasil pekerjaan tidak hanya berupa aplikasi yang berjalan, tetapi juga:

- repositori kode yang dapat diakses sesuai haknya;
- dokumentasi *deployment*;
- daftar dependensi;
- petunjuk *build*;
- skrip instalasi;
- konfigurasi lingkungan;
- dokumentasi perubahan.

Tanpa itu, proyek bisa saja memakai OSS, tetapi secara praktik tetap sulit dipelihara dan tetap bergantung pada vendor tertentu. (JDIH LKPP)

Spesifikasi juga sebaiknya menulis kebutuhan kompatibilitas, misalnya dengan:

- sistem operasi;
- basis data;
- browser;
- standar data;
- API;
- infrastruktur yang sudah ada.

Jika proyek menyentuh area sensitif, kebutuhan audit kode atau minimal transparansi komponen dan dependensi juga patut diminta. (Peraturan BPK)

8.6 Menulis spesifikasi *cloud*: SLA, *region*, enkripsi, *backup*, *exit plan*

Untuk *cloud*, spesifikasi harus lebih luas daripada sekadar kapasitas komputasi atau ruang simpan. Yang dibeli sering kali bukan aset, melainkan **layanan** yang harus tetap tersedia, aman, bisa dipantau, dan bisa ditinggalkan bila perlu. Karena itu, unsur seperti SLA, *region*, enkripsi, *backup*, dan *exit plan* memang harus menjadi bagian inti spesifikasi. Prinsip ini juga konsisten dengan kebutuhan pengelolaan layanan digital pemerintah yang andal dan berkelanjutan. (Peraturan BPK)

Bagian SLA sebaiknya menjelaskan:

- tingkat ketersediaan;
- waktu respons insiden;
- waktu pemulihan;
- mekanisme pembuktian kinerja.

Bagian *region* penting terutama bila ada pertimbangan:

- sensitivitas data;
- latensi;
- kebijakan internal organisasi.

Enkripsi sebaiknya ditulis untuk:

- data saat transit;
- data saat tersimpan.

Backup juga tidak boleh diasumsikan otomatis aman hanya karena memakai *cloud*. Frekuensi, retensi, dan mekanisme pemulihan tetap harus ditulis. Yang sering dilupakan justru *exit plan*, yaitu:

- bagaimana data diekspor;
- bagaimana konfigurasi didokumentasikan;
- bagaimana migrasi keluar dilakukan;
- bagaimana akun dan akses ditutup saat kontrak berakhir.

Jika layanan *cloud* memproses data pribadi, spesifikasi juga harus selaras dengan prinsip pemrosesan data pribadi dalam **UU No. 27 Tahun 2022**, yang mengatur asas dan kewajiban pengendali maupun prosesor data pribadi. (Peraturan BPK)

Pelajaran pentingnya: *cloud* bukan alasan untuk menurunkan kedalaman spesifikasi. Justru karena bentuknya layanan, spesifikasinya harus lebih tajam dalam menjelaskan hak, risiko, dan standar layanan. (Peraturan BPK)

8.7 Menulis spesifikasi LMS

Untuk LMS, spesifikasi yang baik harus berpijak pada **fungsi pembelajaran**, bukan hanya pada fitur halaman depan. Artinya, organisasi perlu menulis kebutuhan seperti:

- manajemen kelas;
- pengelolaan materi;
- kuis dan ujian;
- bank soal;
- pelaporan kemajuan;
- analitik;
- manajemen peran;
- integrasi dengan sistem lain;
- kapasitas pengguna;
- akses *mobile*;
- dukungan saat beban puncak.

Spesifikasi seperti ini membuat dokumen menjadi dekat dengan kebutuhan layanan belajar yang sesungguhnya, bukan sekadar daftar fitur antarmuka. (JDIH LKPP)

Jika LMS dipakai untuk instansi pemerintah, sekolah, kampus, atau pelatihan besar, spesifikasi juga harus memikirkan realitas operasional, misalnya:

- apakah banyak pengguna masuk bersamaan;
- apakah ada ujian daring;
- apakah materi video dominan;
- apakah koneksi pengguna beragam;
- apakah admin lokal perlu kemampuan kelola yang kuat tanpa selalu bergantung pada vendor.

Dengan demikian, spesifikasi menjadi lebih realistis dan lebih siap dipakai untuk mengukur hasil layanan. (Peraturan BPK)

LMS juga sering memproses data peserta, pengajar, hasil belajar, dan rekam aktivitas. Karena itu, kebutuhan seperti berikut harus masuk:

- *audit trail*;
- kontrol akses;
- pengelolaan data;
- akuntabilitas aktivitas.

Tanpa itu, sistem bisa terlihat baik secara antarmuka, tetapi lemah saat ditanya soal keamanan, akuntabilitas, atau pembuktian aktivitas. (Peraturan BPK)

8.8 Menulis spesifikasi *e-commerce*

Untuk *e-commerce* atau *marketplace*, spesifikasi harus memotret operasi digital secara utuh. Yang dibutuhkan bukan hanya tampilan katalog, tetapi juga:

- manajemen produk;
- manajemen pengguna;
- transaksi;
- log aktivitas;
- pembayaran;
- notifikasi;
- pengelolaan pesanan;
- *audit trail*;
- keamanan;
- integrasi pihak ketiga.

Kategori ini memang berbeda dari aplikasi administrasi biasa karena membawa risiko transaksi, bukti elektronik, dan tata kelola yang lebih berat. (Peraturan BPK)

Spesifikasi *e-commerce* yang baik juga harus mengantisipasi kebutuhan pembuktian elektronik. Kerangka **UU ITE**, yang telah diubah terakhir melalui **UU No. 1 Tahun 2024**, menempatkan pemanfaatan teknologi informasi dalam kerangka kepastian hukum, manfaat, dan perlindungan pengguna. Dalam konteks pengadaan, artinya sistem transaksi digital seharusnya memiliki pencatatan, jejak aktivitas, dan kontrol yang cukup untuk mendukung keandalan dan pembuktian bila terjadi sengketa atau insiden. (Peraturan BPK)

Selain itu, bila sistem memproses data pribadi pelanggan, merchant, atau pihak lain, maka spesifikasi juga harus selaras dengan **UU PDP**, misalnya dengan memastikan:

- tujuan pemrosesan jelas;
- data akurat dan dapat dipertanggungjawabkan;
- perlindungan diterapkan sepanjang siklus pemrosesan.

Dengan kata lain, spesifikasi *e-commerce* yang baik harus ditulis dari perspektif:

- operasi bisnis;
- kepercayaan pengguna;
- keamanan;
- jejak bukti.

Bukan hanya dari sisi tampilan antarmuka. (Peraturan BPK)

8.9 Menulis spesifikasi keamanan

Keamanan tidak boleh ditulis hanya sebagai kalimat generik seperti “sistem harus aman.” Spesifikasi keamanan harus memuat **kontrol yang dapat diuji**. Dalam konteks SPBE, keamanan memang merupakan salah satu domain arsitektur nasional, dan audit infrastruktur, aplikasi, serta keamanan SPBE menjadi bagian penting tata kelola. Karena itu, keamanan harus disiapkan bukan hanya untuk melindungi sistem, tetapi juga agar sistem siap diaudit. (Peraturan BPK)

Spesifikasi keamanan yang sehat biasanya mencakup:

- model autentikasi;
- otorisasi berbasis peran;
- *logging* dan *audit trail*;
- enkripsi;
- manajemen kata sandi;
- MFA bila perlu;
- pengelolaan sesi;
- pemisahan lingkungan pengembangan, uji, dan produksi;
- *patch management*;
- *backup*;
- monitoring;
- kewajiban uji keamanan sesuai risiko sistem.

Dengan begitu, keamanan menjadi bagian nyata dari spesifikasi, bukan tambahan kosmetik. (Peraturan BPK)

Pada sistem yang memproses data pribadi, keamanan juga tidak dapat dipisahkan dari kepatuhan. **UU PDP** memandang pelindungan data pribadi sebagai upaya menyeluruh sepanjang rangkaian pemrosesan data. Karena itu, spesifikasi keamanan harus dirancang sebagai bagian dari tata kelola pemrosesan, bukan sekadar tempelan teknis. (Peraturan BPK)

8.10 Menulis spesifikasi layanan dukungan

Banyak pengadaan IT cukup serius menulis spesifikasi untuk barang dan aplikasi, tetapi lemah pada layanan dukungan. Padahal justru dukungan inilah yang sering menentukan apakah sistem benar-benar bisa dipakai setelah serah terima. Karena itu, spesifikasi layanan dukungan harus cukup jelas, setidaknya menjawab:

- apa saja yang didukung;
- pada jam berapa layanan diberikan;
- lewat kanal apa dukungan diberikan;
- berapa waktu responnya;
- siapa PIC-nya;
- bagaimana eskalasinya;
- bagaimana pelaporannya.

Jika bagian ini kabur, organisasi akan sulit menilai apakah layanan benar-benar diberikan sesuai kebutuhan. (JDIH LKPP)

Layanan dukungan juga perlu membedakan jenis layanannya, misalnya:

- *helpdesk*;
- *incident response*;
- *bug fixing*;
- *preventive maintenance*;
- *patching*;
- monitoring;
- administrasi akun;
- pendampingan operasional;
- dukungan integrasi.

Kalau semua dicampur dalam satu istilah “support”, organisasi akan kesulitan menagih mutu layanan secara objektif. (JDIH LKPP)

Spesifikasi dukungan yang baik juga sebaiknya menyebut *deliverable* administratif, seperti:

- tiket layanan;
- laporan berkala;
- dokumentasi perubahan;
- notulen rapat operasional;
- materi pelatihan tambahan;
- daftar insiden yang telah ditangani.

Dengan begitu, dukungan tidak hanya diukur dari klaim, tetapi dari bukti yang bisa diperiksa. (JDIH LKPP)

8.11 Tingkat layanan dan parameter mutu

Untuk proyek IT, tingkat layanan dan parameter mutu tidak boleh menjadi lampiran kabur. Ia harus ditulis sebagai bagian inti spesifikasi. Parameter mutu membantu organisasi memastikan bahwa penyedia tidak hanya mengirimkan barang atau menyalakan sistem, tetapi juga memenuhi standar layanan yang dijanjikan. Pendekatan ini sangat relevan untuk pengadaan IT yang semakin banyak berbasis layanan. (JDIH LKPP)

Parameter mutu dapat berupa:

- *uptime*;
- jumlah insiden kritis;
- waktu respons;
- waktu pemulihan;
- jumlah cacat yang diterima saat UAT;
- kualitas dokumentasi;
- kehadiran tim kunci;
- kualitas pelatihan;
- kepuasan pengguna;
- jumlah pelanggaran terhadap SLA.

Prinsip pentingnya sederhana: **kalau mutu penting, mutunya harus ditulis**. Jika tidak ditulis, ia sulit ditagih. Kalau sulit ditagih, ia mudah dikorbankan saat proyek ditekan oleh waktu atau biaya. (JDIH LKPP)

8.12 *Checklist* anti-diskriminasi

Bab ini layak ditutup dengan *checklist* anti-diskriminasi, karena banyak masalah spesifikasi bukan muncul dari niat buruk yang terang-terangan, melainkan dari kebiasaan, kemalasan teknis, atau warisan dokumen lama. Sebuah spesifikasi layak diuji dengan beberapa pertanyaan sederhana, misalnya:

- apakah kebutuhan ditulis berbasis fungsi dan hasil;
- apakah parameter kinerja dijelaskan;
- apakah kompatibilitas dijelaskan tanpa menyempitkan pasar secara tidak perlu;
- apakah penyebutan merek, bila ada, benar-benar dijustifikasi;
- apakah ruang persaingan masih terbuka;
- apakah kebutuhan keamanan dan integrasi ditulis secara objektif;
- apakah hasil kerja bisa diuji saat penerimaan.

Checklist seperti ini sangat berguna sebagai *quality gate* sebelum dokumen pemilihan dilepas. Ia membantu tim membedakan antara spesifikasi yang benar-benar lahir dari kebutuhan dan spesifikasi yang diam-diam membela satu produk. Dalam pengadaan IT, disiplin kecil seperti ini bisa mengubah kualitas persaingan secara signifikan. (JDIH LKPP)

Ringkasnya, spesifikasi yang tidak diskriminatif **bukan berarti spesifikasi yang lemah**. Justru sebaliknya, ia adalah spesifikasi yang lebih matang, karena disusun dari kebutuhan nyata, kinerja yang terukur, dan kontrol hasil yang bisa diuji. (JDIH LKPP)

Penutup

Bab ini menegaskan bahwa spesifikasi teknis adalah arena paling sensitif dalam pengadaan IT. Di sinilah kebutuhan organisasi diterjemahkan menjadi bahasa pasar. Jika spesifikasi ditulis dengan baik, pengadaan akan menjadi:

- lebih terbuka;
- lebih kompetitif;
- lebih aman dari sengketa;
- lebih mudah diaudit.

Sebaliknya, jika spesifikasi ditulis buruk atau diskriminatif, maka:

- harga menjadi tidak sehat;
- pilihan mengecil;
- risiko penyimpangan meningkat;
- hasil proyek cenderung tidak optimal.

Karena itu, spesifikasi harus disusun berbasis **fungsi, kinerja, mutu, keamanan, interoperabilitas, dan keberlanjutan**, lalu diuji dengan *checklist* anti-diskriminasi sebelum masuk ke proses pemilihan. Prinsip ini konsisten dengan kerangka PBJ, SPBE, UU PDP, dan UU ITE yang sama-sama menuntut akuntabilitas, keamanan, dan keterukuran dalam penyelenggaraan sistem elektronik dan layanan publik digital. (JDIH LKPP)

Tabel Summary. Menyusun Spesifikasi Teknis yang Tidak Diskriminatif

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
8.1 Prinsip spesifikasi berbasis fungsi dan kinerja	Spesifikasi teknis yang sehat harus ditulis berbasis **fungsi, kinerja, kompatibilitas, keandalan, dan hasil layanan**, bukan identitas produk tertentu. Pendekatan ini menjaga mutu tanpa mematikan persaingan.	Tim perencana harus menulis “apa yang harus bisa dilakukan” dan “seberapa baik harus bekerja”, bukan langsung menunjuk merek atau produk.
8.2 Kapan penyebutan merek dibolehkan dan bagaimana membatasi risikonya	Penyebutan merek hanya boleh diperlakukan sebagai **pengecualian yang harus dijustifikasi**, misalnya untuk kompatibilitas atau kebutuhan teknis tertentu yang benar-benar sah.	Bila merek harus disebut, alasannya harus ditulis terbuka, dibatasi pada kebutuhan nyata, dan ruang pembuktian kesetaraan fungsional tetap dijaga bila memungkinkan.
8.3 Menulis spesifikasi server, storage, jaringan, dan endpoint	Spesifikasi infrastruktur harus bertumpu pada **kapasitas, performa, redundansi, kompatibilitas, lingkungan penggunaan, dan kemudahan pengelolaan**, bukan brosur vendor.	Untuk server, storage, jaringan, dan endpoint, kebutuhan harus ditulis dari sudut beban kerja dan konteks operasional agar tidak terlalu rendah, terlalu tinggi, atau terlalu sempit.
8.4 Menulis spesifikasi software dan aplikasi	Spesifikasi aplikasi tidak cukup berisi daftar modul. Ia harus menjelaskan **fungsi bisnis, aktor, workflow, data, integrasi, keamanan, kapasitas, dan hasil kerja**.	Tim harus membedakan kebutuhan wajib, penting, dan pengembangan bertahap agar spesifikasi tetap tajam, realistis, dan tidak dipenuhi fitur yang tidak relevan.
8.5 Menulis spesifikasi open source: lisensi, repository, dokumentasi, kompatibilitas, audit kode	Spesifikasi *open source* harus jelas soal **lisensi, repositori kode, dokumentasi instalasi/konfigurasi, kompatibilitas, dan bila perlu audit kode**. OSS tidak cukup disebut sebagai slogan.	Pastikan pengadaan OSS menuntut kontrol atas *deliverable*: repository, dokumentasi, daftar dependensi, skrip deployment, dan hak akses yang cukup bagi organisasi.

8.6 Menulis spesifikasi cloud: SLA, region, enkripsi, backup, exit plan	Spesifikasi cloud harus menekankan **tingkat layanan, lokasi layanan/region, enkripsi, backup, dan strategi keluar**, karena yang dibeli adalah layanan yang harus tetap tersedia dan bisa ditinggalkan bila perlu.	Dokumen pengadaan cloud harus meminta SLA yang terukur, portabilitas data, dokumentasi konfigurasi, dan *exit plan* untuk mengurangi *vendor lock-in*.
8.7 Menulis spesifikasi LMS	LMS harus dibaca sebagai **ekosistem pembelajaran digital**, bukan sekadar platform. Spesifikasinya harus mencakup kelas, materi, ujian, analitik, integrasi, kapasitas, dan dukungan pengguna.	Saat menulis spesifikasi LMS, perhatikan realitas operasional: beban puncak, kebutuhan admin lokal, pengelolaan konten, dan audit trail aktivitas belajar.
8.8 Menulis spesifikasi e-commerce	Spesifikasi e-commerce harus mencakup **produk, pengguna, transaksi, pembayaran, log aktivitas, notifikasi, audit trail, keamanan, dan integrasi pihak ketiga**. Ia tidak boleh berhenti pada tampilan katalog.	Sistem transaksi digital harus ditulis dengan perhatian pada bukti elektronik, kontrol fraud, log transaksi, dan pelindungan data pribadi agar siap operasi dan siap sengketa bila diperlukan.
8.9 Menulis spesifikasi keamanan	Keamanan harus ditulis sebagai **kontrol yang dapat diuji**, bukan kalimat umum. Ini mencakup autentikasi, otorisasi, logging, enkripsi, patching, backup, monitoring, dan audit trail.	KAK dan spesifikasi harus menyebut kebutuhan keamanan yang objektif dan terukur agar perlindungan sistem dan kesiapan audit dapat dibuktikan saat pengujian maupun operasi.
8.10 Menulis spesifikasi layanan dukungan	Dukungan operasional harus dispesifikkan dengan jelas: **cakupan layanan, jam layanan, kanal dukungan, waktu respons, eskalasi, pelaporan, dan deliverable administratif**.	Jangan cukup menulis “support tersedia”; rincikan jenis dukungan, target respons, jenis laporan, dan bukti layanan agar kebutuhan pasca implementasi benar-benar terlindungi.
8.11 Tingkat layanan dan parameter mutu	Tingkat layanan dan parameter mutu adalah alat untuk memastikan penyedia tidak hanya mengirim hasil, tetapi juga memenuhi **standar layanan dan kualitas yang dijanjikan**.	Tulis KPI dan parameter mutu seperti uptime, respons insiden, waktu pemulihan, kualitas dokumentasi, kualitas

		pelatihan, atau hasil UAT agar mutu bisa ditagih.
8.12 Checklist anti-diskriminasi	Spesifikasi perlu diuji dengan checklist: apakah sudah berbasis fungsi, apakah kinerja dijelaskan, apakah kompatibilitas dijustifikasi, apakah merek disebut hanya bila sah, dan apakah hasil kerja bisa diuji.	Gunakan checklist ini sebagai *quality gate* sebelum dokumen pemilihan dilepas agar spesifikasi tidak diam-diam mengunci satu produk atau vendor.

Ringkasan

Bab 8 menegaskan bahwa spesifikasi teknis adalah titik paling sensitif dalam pengadaan IT, karena disinilah kebutuhan organisasi diterjemahkan menjadi bahasa pasar. Bila spesifikasi ditulis berbasis fungsi, kinerja, mutu, keamanan, interoperabilitas, dan keberlanjutan, maka pengadaan akan lebih terbuka, lebih kompetitif, lebih mudah diaudit, dan lebih kecil resikonya terhadap sengketa maupun *vendor lock-in*. Sebaliknya, spesifikasi yang diskriminatif atau dangkal akan mempersempit persaingan, melemahkan mutu, dan memperbesar risiko penyimpangan.

Bab 9. HPS, TCO, dan Biaya Nyata Proyek IT

9.1 HPS bukan sekadar harga barang

Dalam pengadaan IT, **HPS tidak boleh dipahami hanya sebagai angka untuk membeli perangkat atau lisensi**. HPS harus dibaca sebagai perkiraan biaya yang wajar untuk menghasilkan layanan digital yang benar-benar dibutuhkan organisasi, termasuk sistem, integrasi, dokumentasi, pengujian, pelatihan, dan dukungan. Arah ini sejalan dengan kerangka perencanaan dan pelaksanaan pengadaan dalam **Peraturan LKPP Nomor 11 Tahun 2021** dan **Peraturan LKPP Nomor 12 Tahun 2021** yang kemudian diubah oleh **Peraturan LKPP Nomor 4 Tahun 2024**, serta dengan perubahan umum rezim PBJ melalui **Perpres 46 Tahun 2025**. (JDIH LKPP)

Masalah klasik di lapangan adalah HPS disusun hanya dari *copy-paste* harga perangkat, sementara komponen kerja yang justru paling menentukan tidak dihitung secara memadai. Yang sering tertinggal antara lain:

- analisis;
- integrasi;
- migrasi;
- *hardening*;
- *support*;
- pelatihan;
- dokumentasi;
- stabilisasi.

Akibatnya, tender tampak murah di awal, tetapi proyek menjadi rapuh, penuh perubahan, atau gagal menghasilkan layanan yang benar-benar siap dipakai. Karena itu, HPS proyek IT harus dipahami sebagai **harga proyek layanan digital secara utuh**, bukan sekadar harga komponen yang terlihat. Pembacaan ini merupakan inferensi yang konsisten dengan fungsi perencanaan dan pelaksanaan PBJ dalam regulasi resmi LKPP. (JDIH LKPP)

9.2 *Total Cost of Ownership* untuk proyek IT

Dalam proyek IT, angka pembelian awal hampir tidak pernah mencerminkan biaya sesungguhnya. Yang lebih penting adalah **Total Cost of Ownership (TCO)**, yaitu seluruh biaya yang timbul sepanjang siklus hidup sistem: pengadaan awal, implementasi, operasi, pemeliharaan, pembaruan, dukungan, perubahan, sampai biaya keluar dari teknologi tersebut. Pendekatan seperti ini selaras dengan praktik pengadaan yang sehat, karena mendorong pembeli melihat biaya secara menyeluruh, bukan hanya harga masuk. (JDIH LKPP)

Dalam proyek digital, biaya yang terlihat kecil di awal sering menjadi mahal pada tahun-tahun berikutnya. Misalnya:

- *subscription* naik;
- kebutuhan kapasitas membesar;
- tarif *support* meningkat;
- migrasi ternyata lebih rumit dari perkiraan.

Karena itu, HPS yang sehat sebaiknya minimal dibaca dalam horizon **3 tahun**, dan untuk sistem strategis lebih baik lagi dalam horizon **5 tahun**. Ini memang bukan angka yang ditetapkan eksplisit dalam regulasi yang saya buka, tetapi merupakan inferensi manajerial yang masuk akal dari kebutuhan menghitung biaya proyek secara utuh dan berkelanjutan. (JDIH LKPP)

TCO penting untuk mencegah keputusan yang tampak hemat tetapi sebenarnya boros. Sistem yang murah saat pembelian bisa menjadi mahal bila bergantung pada:

- lisensi tahunan;
- *managed service* mahal;
- pelatihan terus-menerus;
- biaya integrasi berulang.

Sebaliknya, solusi yang tampak lebih mahal di awal bisa lebih efisien dalam jangka menengah bila dokumentasinya baik, integrasinya sehat, dan ketergantungan vendornya lebih rendah. Jadi, HPS harus berfungsi sebagai **alat berpikir strategis**, bukan hanya angka pembanding penawaran. (JDIH LKPP)

9.3 Komponen CAPEX dan OPEX

Agar HPS sehat, organisasi perlu membedakan **CAPEX** dan **OPEX** secara jelas. CAPEX biasanya terkait investasi awal, seperti:

- perangkat keras;
- instalasi awal;
- pengembangan awal;
- *setup* infrastruktur;
- pembangunan sistem.

Sementara itu, OPEX biasanya terkait biaya berulang, seperti:

- *subscription*;
- tagihan *cloud*;
- konektivitas;
- *support*;
- pemeliharaan;
- monitoring;
- audit berkala;
- lisensi tahunan;
- tenaga operasional.

Dalam pengadaan IT, kegagalan membedakan dua kategori ini sering membuat anggaran tampak aman di tahun pertama, tetapi menimbulkan beban pada tahun berikutnya. (JDIH LKPP)

Untuk konteks penganggaran pemerintah pusat, terdapat rujukan resmi berupa **PMK 32 Tahun 2025** tentang **Standar Biaya Masukan Tahun Anggaran 2026** dan **PMK 3 Tahun 2026** tentang **Standar Biaya Keluaran dalam Penganggaran Kementerian Negara/Lembaga**. Kedua regulasi ini penting sebagai jangkar formal, tetapi untuk proyek IT banyak komponen teknis tetap harus dijustifikasi dengan data pasar dan struktur biaya yang relevan. (Kementerian Keuangan Republik Indonesia)

Dalam praktik, pembagian CAPEX–OPEX juga membantu organisasi memilih strategi teknologi. Sistem *on-premise* bisa lebih berat di CAPEX tetapi lebih terkendali pada sebagian OPEX. Sebaliknya, *cloud* atau SaaS bisa lebih ringan di CAPEX, tetapi OPEX dapat terus berjalan dan meningkat sesuai skala penggunaan. Karena itu, yang penting bukan memilih mana yang tampak murah di awal, tetapi memahami **profil biaya sepanjang umur layanan**. (Kementerian Keuangan Republik Indonesia)

9.4 Lisensi, *subscription*, *cloud bill*, implementasi, migrasi, *support*, *training*

Komponen biaya proyek IT hampir selalu lebih luas daripada yang dibayangkan pada awal. Di luar lisensi atau perangkat, organisasi juga harus menghitung:

- *subscription*;
- tagihan *cloud*;
- implementasi;
- migrasi data;
- *support*;
- *training*;
- stabilisasi pasca-*go-live*.

Jika komponen-komponen ini tidak dihitung sejak awal, HPS akan tampak rendah tetapi tidak realistis. Ini adalah inferensi praktis yang konsisten dengan fungsi HPS sebagai perkiraan biaya yang wajar dalam kerangka PBJ. (JDIH LKPP)

Lisensi dan *subscription* juga harus dibaca berbeda. Lisensi bisa bersifat sekali beli atau periodik. *Subscription* hampir selalu berulang dan sering meningkat sesuai kapasitas, jumlah pengguna, atau fitur. Tagihan *cloud* bahkan lebih dinamis lagi karena dipengaruhi oleh:

- *compute*;
- *storage*;
- *bandwidth*;
- *backup*;
- *observability*;
- layanan tambahan lainnya.

Tanpa asumsi penggunaan yang realistis, HPS *cloud* bisa tampak murah saat tender tetapi tidak tahan menghadapi operasi nyata. (Kementerian Keuangan Republik Indonesia)

Sementara itu, implementasi dan migrasi hampir selalu lebih rumit daripada perkiraan awal. Pembersihan data, pemetaan *field*, penyesuaian proses, integrasi antar sistem, pelatihan admin, dan pendampingan pengguna adalah **biaya nyata**, bukan pelengkap. *Training* juga tidak boleh diperlakukan sebagai sesi simbolik. Untuk sistem yang dipakai luas, pelatihan admin, operator, dan pengguna kunci harus dihitung sebagai bagian dari keberhasilan proyek, bukan sebagai bonus vendor. (JDIH LKPP)

9.5 Biaya keamanan, audit, *backup*, dan DR

Salah satu kekeliruan paling berbahaya dalam proyek IT adalah menganggap keamanan, audit, *backup*, dan *disaster recovery* sebagai biaya opsional. Untuk sistem yang memproses layanan publik, data penting, atau data pribadi, keempatnya harus dipandang sebagai **biaya wajib**. Dalam kerangka SPBE, keamanan merupakan salah satu domain arsitektur nasional, dan SPBE sendiri diarahkan untuk menghasilkan layanan yang efektif, efisien, berkesinambungan, dan berkualitas. (JDIH LKPP)

Dari sisi pelindungan data, **UU 27 Tahun 2022** mewajibkan pemrosesan data pribadi dilakukan secara sah dan dapat dipertanggungjawabkan, serta menempatkan kewajiban pada pengendali dan prosesor data pribadi. Konsekuensinya, biaya untuk:

- *hardening*;
- audit keamanan;
- *logging*;
- *backup*;
- *recovery testing*;
- kontrol akses

bukan biaya mewah, melainkan bagian dari kepatuhan dan keberlanjutan layanan. (JDIH LKPP)

Backup harus dihitung dari:

- frekuensi;
- retensi;
- media;
- uji pemulihan.

DR harus dihitung dari:

- target waktu pemulihan;
- redundansi;
- skenario gangguan.

Audit dapat mencakup:

- audit konfigurasi;
- audit aplikasi;
- audit keamanan;
- penilaian kepatuhan.

Jika semua itu tidak masuk HPS sejak awal, organisasi biasanya hanya punya dua pilihan buruk: hidup dengan resiko tinggi, atau menambah biaya besar di belakang hari. (JDIH LKPP)

9.6 Biaya integrasi dan perubahan organisasi

Proyek IT jarang gagal karena aplikasi tidak bisa dipasang. Ia lebih sering gagal karena **integrasi** dan **perubahan organisasi** tidak dihitung secara jujur. Integrasi menuntut pekerjaan nyata seperti:

- analisis sistem lama;
- pemetaan data;
- pengembangan API;
- pengujian lintas sistem;
- sinkronisasi hak akses;
- dokumentasi teknis.

Jika HPS hanya menghitung pembangunan modul baru tanpa integrasi yang memadai, organisasi sebenarnya sedang membeli sistem yang tampak selesai tetapi belum benar-benar nyambung dengan dunia nyata. (JDIH LKPP)

Biaya perubahan organisasi juga sering diabaikan. Padahal transformasi digital hampir selalu menuntut:

- penyesuaian SOP;
- pelatihan ulang;
- pendampingan pengguna;
- komunikasi internal;
- pembentukan peran baru;
- penataan tata kelola data.

Dalam kerangka SPBE, layanan pemerintah memang diarahkan pada integrasi proses bisnis, data, aplikasi, infrastruktur, dan keamanan. Itu berarti biaya perubahan proses bukan pengeluaran pinggiran, melainkan bagian inti dari keberhasilan implementasi. (JDIH LKPP)

Kesimpulannya, HPS yang tidak memasukkan integrasi dan perubahan organisasi biasanya hanya menghitung **menyalakan sistem**, bukan **membuat sistem benar-benar dipakai dan bekerja**. (JDIH LKPP)

9.7 Menggunakan SBM, SBK, struktur skala upah, dan *benchmark* pasar

Untuk menyusun HPS yang kredibel, organisasi tidak cukup mengandalkan intuisi. Perlu kombinasi antara **rujukan resmi** dan **benchmark pasar**. Pada sisi resmi, ada **PMK 32 Tahun 2025** untuk SBM TA 2026 dan **PMK 3 Tahun 2026** untuk SBK dalam penganggaran K/L. Pada sisi pasar, proyek IT tetap memerlukan pembanding harga aktual untuk perangkat, layanan *cloud*, jasa implementasi, dan tenaga ahli, karena banyak komponen teknis memang tidak bisa seluruhnya diturunkan dari tabel biaya resmi. (Kementerian Keuangan Republik Indonesia)

Pendekatan ini penting terutama untuk proyek IT, karena pasar tenaga digital bergerak lebih cepat daripada banyak referensi administrasi. Menggunakan rujukan resmi tanpa membaca pasar bisa membuat HPS terlalu rendah. Sebaliknya, memakai klaim pasar tanpa pembanding resmi bisa membuat HPS sulit dipertanggungjawabkan. Karena itu, praktik yang sehat adalah:

- memakai SBM/SBK sebagai jangkar formal;
- menguatkannya dengan pembanding pasar yang terdokumentasi;
- memastikan logika HPS dapat dijelaskan, bukan hanya angkanya.

Ini merupakan sintesis praktis dari fungsi SBM/SBK dan kebutuhan kewajaran harga dalam PBJ. (Kementerian Keuangan Republik Indonesia)

9.8 Menghitung biaya jasa SDM per peran

Dalam banyak proyek IT, biaya terbesar justru ada pada **manusia**, bukan perangkat. Karena itu, HPS sebaiknya menghitung biaya jasa SDM **per peran**, bukan sebagai satu angka gelondongan yang kabur.

Perhitungan per peran berarti organisasi membedakan kebutuhan, misalnya untuk:

- *project manager*;
- *business analyst*;
- *system analyst*;
- *architect*;
- *developer*;
- QA;
- *security specialist*;
- DevOps;
- *support*;
- trainer.

Tiap peran memiliki tarif, produktivitas, senioritas, dan output yang berbeda. Menyatukan semuanya dalam satu angka “jasa pengembangan” membuat HPS sulit diuji kewajarannya. Pendekatan per peran juga lebih konsisten dengan logika SBK sebagai indeks biaya untuk menghasilkan keluaran tertentu. (Kementerian Keuangan Republik Indonesia)

Pendekatan ini penting untuk menghindari dua jebakan:

- personel fiktif;
- senioritas semu.

Ketika HPS ditulis terlalu gelondongan, proyek mudah diisi personel yang tidak selevel dengan harga yang dibayar, atau justru terlalu murah sampai menarik tim yang tidak memadai. Karena itu, biaya jasa SDM sebaiknya dihitung dari kombinasi:

- peran;
- level;
- beban kerja;
- durasi;
- hasil yang diharapkan.

Ini adalah praktik yang lebih akuntabel dan lebih mudah dijustifikasi. (Kementerian Keuangan Republik Indonesia)

9.9 Menghindari HPS terlalu rendah

Salah satu penyakit kronis proyek IT adalah **HPS terlalu rendah**. Sepintas ini terlihat hemat. Padahal dampaknya sering merusak:

- penyedia menawar tidak sehat;
- kualitas personel turun;
- ruang lingkup ditekan diam-diam;
- dokumentasi dipangkas;
- pengujian dipersingkat;
- *support* pasca-implementasi menjadi minimal.

Pada akhirnya, yang tampak hemat di awal berubah menjadi biaya koreksi, risiko audit, atau sistem yang tidak dipakai. Fungsi HPS sebagai perkiraan biaya yang wajar justru menuntut organisasi menghindari jebakan ini. (JDIH LKPP)

HPS terlalu rendah biasanya lahir dari tiga sumber:

- hanya menghitung barang utama dan mengabaikan pekerjaan pendukung;
- memakai pembanding yang tidak setara;
- gagal membaca realitas pasar SDM digital.

Karena itu, strategi terbaik untuk menghindarinya adalah menghitung biaya dari bawah ke atas: komponen teknis, jasa SDM, risiko, *support*, stabilisasi, dan kebutuhan pasca-*go-live*, lalu membandingkannya dengan SBM/SBK dan data pasar yang relevan. (Kementerian Keuangan Republik Indonesia)

Intinya, **HPS yang terlalu rendah bukan tanda efisiensi**. Dalam proyek IT, itu sering justru tanda bahwa organisasi belum jujur terhadap biaya nyata yang dibutuhkan untuk menghasilkan pelayanan yang aman, stabil, dan dapat dipertanggungjawabkan. (Kementerian Keuangan Republik Indonesia)

9.10 Model HPS untuk OSS vs *proprietary*

Salah satu bagian paling penting dalam HPS proyek IT adalah membedakan model biaya untuk **OSS** dan **proprietary**. Pada solusi *proprietary*, porsi biaya biasanya lebih besar pada:

- lisensi;
- *subscription*;
- hak akses platform.

Pada solusi OSS, porsi biaya sering bergeser ke:

- implementasi;
- integrasi;
- *hardening*;
- dokumentasi;
- pelatihan;
- *support*;
- operasi lokal.

Artinya, HPS OSS tidak otomatis lebih murah, tetapi struktur biayanya sering lebih sehat untuk membangun kapasitas dan nilai lokal. Ini adalah inferensi strategis yang konsisten dengan pergeseran biaya dari lisensi ke jasa implementasi dalam praktik proyek digital. (JDIH LKPP)

Model HPS OSS juga tetap harus menghitung biaya nyata, seperti:

- kualitas tim integrator;
- kesiapan dokumentasi;
- pengelolaan dependensi;
- audit keamanan;
- *support*;
- keberlanjutan.

Sebaliknya, model *proprietary* harus lebih disiplin menghitung:

- biaya lisensi berulang;
- biaya keluar;
- kompatibilitas;
- potensi *lock-in*.

Jadi, pembeda utamanya bukan “mana yang gratis dan mana yang berbayar”, melainkan **ke mana nilai proyek mengalir**. Pada OSS, nilai bisa lebih banyak masuk ke jasa lokal dan penguasaan sistem. Pada *proprietary*, nilai sering lebih banyak masuk ke hak akses dan dependensi platform. HPS yang baik harus membuat perbedaan struktur biaya ini terlihat jelas. (JDIH LKPP)

9.11 Model HPS untuk proyek multi-tahun

Untuk sistem yang strategis, besar, atau bertahap, pendekatan paling sehat sering kali adalah **proyek multi-tahun**. Ini relevan karena banyak proyek IT gagal justru karena memaksa seluruh beban biaya dan risiko masuk ke satu tahun anggaran, padahal transformasi digital secara alami berlangsung bertahap.

Pada proyek multi-tahun, HPS sebaiknya dibangun dalam lapisan, misalnya:

- biaya tahun pertama untuk analisis, desain, fondasi, atau implementasi awal;
- biaya tahun berikutnya untuk pengembangan lanjutan, integrasi tambahan, optimasi, *support*, atau ekspansi kapasitas.

Model seperti ini membuat proyek lebih realistis dan memungkinkan organisasi menyesuaikan ritme dengan kesiapan SDM, perubahan kebutuhan, dan pembelajaran lapangan. Gagasan ini juga sejalan dengan pembacaan SPBE sebagai proses yang terencana dan bertahap, serta dengan struktur penganggaran yang memakai SBM dan SBK sebagai acuan tahunan. (JDIH LKPP)

Dalam model multi-tahun, HPS sebaiknya tidak hanya memecah angka per tahun, tetapi juga menjelaskan:

- logika tahapan;
- manfaat tiap fase;
- risiko tiap fase;
- biaya keberlanjutan setelah fase implementasi utama selesai.

Dengan begitu, proyek multi-tahun tidak hanya rapi secara angka, tetapi juga jelas secara strategi dan akuntabilitas. (JDIH LKPP)

Penutup

Bab ini menegaskan bahwa **HPS proyek IT harus dibangun dari biaya nyata, bukan ilusi harga murah**. Ia harus melihat proyek secara utuh, meliputi:

- barang;
- *software*;
- lisensi;
- *subscription*;
- *cloud*;
- integrasi;
- migrasi;
- keamanan;
- *backup*;
- DR;
- *support*;
- *training*;
- perubahan organisasi;
- biaya SDM per peran.

Ia juga harus dibaca dalam logika **TCO**, memakai **SBM/SBK** dan *benchmark* pasar secara seimbang, serta cukup cerdas untuk membedakan struktur biaya **OSS** dan **proprietary**, termasuk kebutuhan proyek **multi-tahun**. Jika HPS ditulis dengan matang, pengadaan akan lebih sehat, hasil lebih realistis, dan risiko kegagalan lebih kecil. Jika HPS ditulis dangkal, maka hampir semua masalah berikutnya hanya menunggu waktu untuk muncul. (JDIH LKPP)

Tabel Summary. HPS, TCO, dan Biaya Nyata Proyek IT

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
9.1 HPS bukan sekadar harga barang	**HPS proyek IT harus dibaca sebagai harga layanan digital secara utuh**, bukan sekadar harga perangkat atau lisensi. Ia harus mencerminkan biaya untuk menghasilkan sistem, integrasi, dokumentasi, pelatihan, dan dukungan yang benar-benar dibutuhkan.	Saat menyusun HPS, tim tidak boleh berhenti pada daftar barang. HPS harus mencakup komponen kerja end-to-end agar tender tidak tampak murah di awal tetapi gagal saat implementasi.
9.2 Total Cost of Ownership untuk proyek IT	Biaya awal hampir tidak pernah menjadi biaya sesungguhnya. Yang lebih penting adalah **TCO**, yaitu seluruh biaya sepanjang siklus hidup sistem: pengadaan, implementasi, operasi, pemeliharaan, perubahan, dan biaya keluar.	HPS sebaiknya dibaca minimal dalam horizon 3 tahun, dan untuk sistem strategis lebih baik 5 tahun, agar keputusan tidak terjebak pada harga masuk yang menipu.
9.3 Komponen CAPEX dan OPEX	Proyek IT harus membedakan **CAPEX** dan **OPEX**. CAPEX biasanya terkait investasi awal, sedangkan OPEX terkait biaya berulang seperti subscription, cloud, support, monitoring, dan operasi.	Tim anggaran perlu membaca profil biaya tahunan proyek agar sistem yang tampak murah di tahun pertama tidak berubah menjadi beban besar di tahun berikutnya.
9.4 Lisensi, subscription, cloud bill, implementasi, migrasi, support, training	Biaya nyata proyek IT hampir selalu lebih luas daripada yang terlihat. Lisensi dan perangkat hanyalah sebagian; implementasi, migrasi, support, training, dan stabilisasi sering justru menentukan keberhasilan.	HPS harus memasukkan semua komponen pendukung secara eksplisit agar proyek tidak kekurangan biaya untuk pekerjaan yang paling menentukan mutu dan keberlanjutan.

9.5 Biaya keamanan, audit, backup, dan DR	Keamanan, audit, backup, dan *disaster recovery* bukan biaya opsional, melainkan bagian dari **kepatuhan, ketahanan layanan, dan kesiapan audit**.	Dalam HPS, perlu dihitung biaya *hardening*, audit keamanan, logging, backup, recovery testing, dan kontrol akses agar sistem tidak hanya berjalan, tetapi juga aman dan dapat dipertanggungjawabkan.
9.6 Biaya integrasi dan perubahan organisasi	Banyak proyek gagal bukan karena aplikasi tidak jadi, tetapi karena **integrasi** dan **perubahan organisasi** tidak dihitung dengan jujur. Ini adalah biaya nyata, bukan biaya tambahan pinggiran.	HPS harus mencakup biaya pemetaan data, API, pengujian integrasi, revisi SOP, komunikasi perubahan, pendampingan pengguna, dan pelatihan agar sistem benar-benar dipakai.
9.7 Menggunakan SBM, SBK, struktur skala upah, dan benchmark pasar	HPS yang kredibel harus memakai kombinasi **rujukan resmi** dan **benchmark pasar**. SBM/SBK memberi jangkar formal, sedangkan data pasar membantu membaca realitas biaya teknis dan SDM.	Tim penyusun HPS harus bisa menjelaskan logika angka: mana yang dirujuk dari standar resmi, mana yang berasal dari pembanding pasar, dan mengapa keduanya dipakai bersama.
9.8 Menghitung biaya jasa SDM per peran	Dalam banyak proyek IT, biaya terbesar ada pada **tenaga ahli**. Karena itu, HPS harus menghitung biaya SDM **per peran**, bukan angka gelondongan yang kabur.	Hitung kebutuhan peran seperti PM, BA, architect, developer, QA, security, DevOps, support, dan trainer berdasarkan level, durasi, dan beban kerja agar kewajarannya bisa diuji.
9.9 Menghindari HPS terlalu rendah	**HPS terlalu rendah bukan efisiensi**, tetapi sering menjadi tanda bahwa biaya nyata belum dihitung jujur. Dampaknya adalah kualitas turun, personel lemah, dokumentasi dipangkas, dan support minim.	Hindari HPS terlalu rendah dengan menghitung proyek dari bawah ke atas: komponen teknis, SDM, risiko, support, stabilisasi, dan kebutuhan pasca go-live.

9.10 Model HPS untuk OSS vs proprietary	Struktur biaya **OSS** dan **proprietary** berbeda. Pada OSS, biaya sering bergeser ke integrasi, support, dokumentasi, pelatihan, dan operasi lokal; pada proprietary, biaya sering lebih berat pada lisensi dan ketergantungan platform.	HPS harus menunjukkan ke mana nilai proyek mengalir, sehingga pembeli dapat menilai bukan hanya harga, tetapi juga dampaknya pada kapasitas lokal dan risiko lock-in.
9.11 Model HPS untuk proyek multi-tahun	Untuk sistem besar atau strategis, pendekatan **multi-tahun** sering lebih sehat daripada memaksa semua beban biaya ke satu tahun anggaran.	HPS multi-tahun perlu menjelaskan biaya per fase, manfaat tiap fase, risiko tiap fase, dan biaya keberlanjutan setelah implementasi utama selesai agar ritme proyek lebih realistis dan terkendali.

Ringkasan

Bab 9 menegaskan bahwa HPS proyek IT harus dibangun dari biaya nyata, bukan ilusi harga murah. Karena itu, penyusunan HPS harus memakai logika **TCO**, membedakan **CAPEX** dan **OPEX**, menghitung lisensi, cloud, integrasi, migrasi, keamanan, backup, DR, support, training, dan biaya SDM per peran, lalu menyeimbangkannya dengan **SBM/SBK** dan **benchmark pasar**. Dengan pendekatan ini, pengadaan menjadi lebih realistis, lebih sehat, dan lebih kecil resikonya terhadap kegagalan implementasi maupun audit.

Bab 10. Rancangan Kontrak, *Deliverable*, UAT, dan SLA

10.1 Menyelaraskan KAK, spesifikasi, HPS, dan kontrak

Kontrak pengadaan IT yang baik tidak pernah berdiri sendiri. Ia harus menjadi kelanjutan yang logis dari **KAK, spesifikasi, dan HPS**. Jika empat dokumen ini tidak sinkron, masalah hampir pasti muncul saat pelaksanaan. Bisa terjadi kebutuhan yang ditulis di KAK tidak seluruhnya masuk ke kontrak, spesifikasi menuntut hal yang tidak tercermin dalam HPS, atau kontrak terlalu umum sehingga hasil kerja sulit ditagih. Posisi ini sejalan dengan **Peraturan LKPP Nomor 12 Tahun 2021** tentang pedoman pelaksanaan pengadaan melalui penyedia yang telah diubah oleh **Peraturan LKPP Nomor 4 Tahun 2024**. (JDIH LKPP)

Secara praktis, penyelarasan berarti setiap unsur penting harus bisa ditelusuri silang:

- masalah dan output ada di KAK;
- parameter teknis ada di spesifikasi;
- kewajaran biaya ada di HPS;
- kewajiban, bukti hasil, cara uji, jadwal, dan konsekuensi hukum ada di kontrak.

Kalau satu mata rantai ini putus, kontrak akan sulit dipakai untuk mengendalikan proyek. Itu sebabnya kontrak tidak boleh dipandang hanya sebagai dokumen hukum, tetapi juga sebagai **instrumen operasional** yang menjaga konsistensi dari perencanaan sampai pelaksanaan. (JDIH LKPP)

10.2 Kontrak berbasis barang vs berbasis layanan

Dalam pengadaan IT, sangat penting membedakan **kontrak berbasis barang** dan **kontrak berbasis layanan**. Banyak paket digital terlihat seperti pembelian barang, padahal nilai utamanya justru ada pada layanan: implementasi, integrasi, migrasi, dukungan, operasi, dan pemeliharaan. Jika kontrak salah membaca sifat pekerjaan, maka ukuran keberhasilannya juga akan salah.

Pada kontrak berbasis barang, fokus umumnya ada pada:

- spesifikasi fisik atau lisensi;
- jumlah;
- kompatibilitas;
- instalasi dasar;
- garansi;
- uji fungsi.

Pada kontrak berbasis layanan, fokusnya bergeser ke:

- *deliverable* bertahap;
- kinerja tim;
- kualitas hasil;
- SLA;
- pelaporan;
- perbaikan insiden;
- tata kelola layanan.

Dalam proyek IT modern, yang paling sering terjadi justru campuran keduanya. Karena itu, kontrak harus jujur membaca mana yang benar-benar barang dan mana yang sesungguhnya layanan. Pembacaan ini juga selaras dengan kerangka **SPBE**, yang memandang IT pemerintah sebagai bagian dari layanan pemerintahan yang terintegrasi, bukan sekadar aset teknologi. (Peraturan BPK)

10.3 *Deliverable* minimal proyek IT

Salah satu akar sengketa paling umum dalam proyek IT adalah *deliverable* yang tidak didefinisikan cukup rinci. Organisasi merasa membeli sistem yang siap dipakai, sementara penyedia merasa cukup menyerahkan aplikasi yang “sudah jalan”. Karena itu, kontrak IT harus menetapkan *deliverable* minimal secara eksplisit.

Deliverable proyek IT biasanya tidak cukup hanya berupa aplikasi atau perangkat yang aktif. Bergantung pada konteks proyek, kontrak sebaiknya mencakup:

- dokumen analisis;
- dokumen desain;
- konfigurasi lingkungan;
- *source code* atau akses yang relevan;
- skrip *deployment*;
- hasil pengujian;
- dokumentasi integrasi;
- manual admin;
- manual pengguna;
- SOP operasional;
- materi pelatihan;
- berita acara tiap fase;
- laporan stabilisasi.

Pelajaran pentingnya sederhana: **kalau hasil kerja penting, hasil kerja itu harus ditulis**. Hal-hal yang tidak tertulis sering berubah menjadi wilayah abu-abu, dan wilayah abu-abu dalam proyek IT hampir selalu menjadi sumber debat mahal di tahap akhir. Penyelarasan ini juga merupakan konsekuensi logis dari fungsi kontrak dalam rezim pelaksanaan PBJ. (JDIH LKPP)

10.4 *Source code*, repo, dokumentasi, akun admin, SOP, diagram, pelatihan

Pada proyek perangkat lunak dan layanan digital, kontrak yang baik harus tegas mengatur soal:

- *source code*;
- repositori;
- dokumentasi;
- akun admin;
- SOP;
- diagram sistem;
- pelatihan.

Tanpa pengaturan yang jelas, organisasi berisiko menerima sistem yang tampak aktif tetapi tidak benar-benar dikuasai. Contohnya:

- aplikasi berjalan, tetapi repo tidak diserahkan;
- akun admin utama tetap dipegang vendor;
- diagram arsitektur tidak tersedia;
- SOP tidak jelas;
- pelatihan hanya formalitas singkat.

Dalam jangka pendek, kondisi ini membuat organisasi bergantung. Dalam jangka menengah, ia melemahkan audit, pemeliharaan, dan transisi vendor.

Karena itu, kontrak sebaiknya membedakan secara tegas antara:

- hak akses operasional;
- hak pakai;
- hak modifikasi;
- dokumen teknis yang wajib diserahkan.

Untuk proyek OSS, repo, dokumentasi *build/deploy*, dependensi, dan konfigurasi menjadi sangat penting. Untuk proyek *proprietary*, organisasi setidaknya tetap harus mengamankan dokumentasi, akun admin yang memang menjadi hak pengguna, konfigurasi, dan SOP operasi yang memadai. Tujuannya jelas: mengurangi *vendor lock-in* dan memastikan organisasi benar-benar dapat mengelola hasil proyeknya. (JDIH LKPP)

10.5 UAT, *security test*, *integration test*, *performance test*

Kontrak IT yang sehat harus jelas soal **mekanisme uji terima**. Tidak cukup hanya menulis “akan dilakukan UAT”. Organisasi perlu mendefinisikan jenis pengujian, kriteria lulus, pihak yang menguji, mekanisme perbaikan, dan bukti hasilnya.

Minimal, kontrak sebaiknya membedakan empat lapisan pengujian:

- **UAT**, untuk membuktikan fungsi bisnis;
- **integration test**, untuk membuktikan hubungan antar sistem;
- **performance test**, untuk membuktikan sistem mampu menahan beban yang relevan;
- **security test**, untuk membuktikan kontrol keamanan minimum telah diterapkan sesuai risiko.

Kalau lapisan-lapisan ini tidak dipisah, penerimaan hasil kerja akan mudah jatuh pada penilaian visual semata, padahal masalah paling serius justru sering tersembunyi di integrasi, beban, dan keamanan.

Pendekatan ini juga selaras dengan SPBE, yang menempatkan layanan digital pemerintah dalam kerangka proses, data, aplikasi, infrastruktur, dan keamanan yang terintegrasi. Karena itu, uji terima tidak boleh berhenti pada “aplikasi bisa dibuka”, tetapi harus sampai pada pembuktian bahwa sistem memang siap dipakai secara fungsional dan aman. (Peraturan BPK)

10.6 SLA, SLO, masa garansi, masa stabilisasi

Untuk proyek IT, terutama yang berlanjut ke operasi, kontrak harus membedakan dengan jelas antara:

- **SLA**: komitmen layanan yang dikontrakkan dan bisa ditagih;
- **SLO**: target operasional internal untuk menjaga layanan;
- **masa garansi**: kewajiban perbaikan atas cacat atau kegagalan tertentu;
- **masa stabilisasi**: periode setelah *go-live* ketika sistem dimonitor dan disempurnakan secara intensif.

Dalam praktik, kontrak harus menulis parameter yang bisa dibuktikan, misalnya:

- *uptime*;
- waktu respons insiden;
- waktu pemulihan;
- kualitas *bug fix*;
- masa observasi pasca-*go-live*;
- konsekuensi bila target tidak tercapai.

Kalau tingkat layanan penting, maka tingkat layanan itu harus **ditulis dan diukur**. Dalam proyek IT pemerintah yang semakin banyak berbasis layanan, bagian ini tidak boleh menjadi lampiran kabur. Ia harus menjadi alat ukur yang konkret untuk menagih mutu layanan. Pendekatan ini juga sejalan dengan arah pengadaan melalui penyedia dalam regulasi LKPP. (JDIH LKPP)

10.7 Serah terima dan *knowledge transfer*

Dalam proyek IT, serah terima tidak boleh dipersempit menjadi tanda tangan berita acara. Serah terima yang sehat harus berarti organisasi menerima bukan hanya sistem atau barang, tetapi juga **pengetahuan minimum** untuk mengoperasikannya dengan aman dan masuk akal.

Ini penting karena banyak proyek terlihat selesai secara administratif, tetapi sebenarnya belum siap dipakai tanpa vendor. Misalnya:

- admin belum paham;
- SOP belum dijalankan;
- diagram belum dibaca tim internal;
- akun belum diserahterimakan penuh.

Jika keadaannya seperti itu, maka proyek sesungguhnya belum benar-benar selesai secara substantif.

Serah terima yang baik sebaiknya mencakup:

- daftar aset atau komponen;
- dokumen teknis;
- dokumen operasional;
- akun dan kredensial sesuai otorisasi;
- pelatihan admin/operator;
- hasil uji;
- daftar isu terbuka;
- rencana penanganan selama stabilisasi.

Dengan begitu, serah terima menjadi titik transisi yang sehat dari pembangunan ke pengelolaan, bukan sekadar penutupan formal kontrak. Ini juga merupakan konsekuensi logis dari pengadaan IT yang dipahami sebagai pembangunan layanan, bukan hanya pengiriman barang. (Peraturan BPK)

10.8 Klausul *exit management* dan transisi vendor

Salah satu ciri kontrak IT yang matang adalah keberanian memikirkan akhir hubungan sejak awal. Di sinilah fungsi klausul **exit management** dan **transisi vendor**.

Klausul ini penting untuk menjawab pertanyaan yang sering dihindari:

- apa yang terjadi bila kontrak berakhir;
- apa yang terjadi bila vendor diganti;
- bagaimana bila sistem dipindahkan ke platform lain.

Tanpa klausul ini, organisasi bisa menghadapi kesulitan besar saat membutuhkan:

- migrasi data;
- transfer dokumentasi;
- bantuan transisi;
- koordinasi dengan penyedia baru.

Karena itu, kontrak sebaiknya menulis secara terang:

- kewajiban penyerahan data;
- format ekspor;
- dokumentasi konfigurasi;
- dukungan masa transisi;
- kerja sama dengan vendor baru;
- batas biaya transisi;
- larangan menghambat alih kelola.

Tujuan klausul ini bukan memusuhi vendor, tetapi melindungi kesinambungan layanan publik dan mengurangi risiko *vendor lock-in*. Dalam konteks layanan digital pemerintah, ini adalah bagian dari manajemen risiko yang sehat. (Peraturan BPK)

10.9 Klausul kerahasiaan, hak pakai, dan hak modifikasi

Kontrak IT juga harus jernih membedakan tiga isu yang sering bercampur:

- **kerahasiaan**;
- **hak pakai**;
- **hak modifikasi**.

Kerahasiaan menyangkut informasi, data, konfigurasi, dan akses yang tidak boleh dibuka sembarangan. Hak pakai menyangkut ruang penggunaan atas perangkat lunak, sistem, atau materi tertentu. Hak modifikasi menyangkut apakah pengguna atau pihak yang ditunjuk boleh mengubah, mengembangkan, atau menyesuaikan hasil kerja.

Pemisahan ini penting agar organisasi tidak salah mengira bahwa hak pakai otomatis berarti hak ubah, atau sebaliknya. Pada solusi OSS, hak modifikasi cenderung lebih terbuka sesuai lisensinya, tetapi kontrak tetap harus mengatur penyerahan hasil kerja dan tanggung jawab operasional. Pada solusi *proprietary*, hak modifikasi bisa terbatas, tetapi organisasi tetap harus melindungi hak pakai yang cukup, dokumentasi, dan keberlangsungan layanan.

Untuk sistem yang memproses data pribadi atau informasi sensitif, klausul kerahasiaan juga harus selaras dengan kewajiban pelindungan data. **UU 27 Tahun 2022** menempatkan kewajiban pada pengendali dan prosesor data pribadi, sehingga kontrak tidak boleh kabur soal:

- siapa mengakses apa;
- dalam kapasitas apa;
- dengan kewajiban keamanan seperti apa. (Peraturan BPK)

10.10 Sengketa, penalti, dan *remedy*

Kontrak IT yang matang tidak alergi pada kemungkinan sengketa. Ia justru menulis mekanisme penyelesaian, penalti, dan *remedy* secara proporsional. Namun penalti dalam proyek IT harus diperlakukan hati-hati. Terlalu lemah membuat organisasi kehilangan alat kendali; terlalu keras bisa merusak ruang penyelesaian masalah secara profesional.

Karena itu, keseimbangannya penting: **cukup tegas untuk menjaga akuntabilitas, tetapi tidak merusak kemungkinan penyelesaian yang jujur dan profesional**.

Secara praktis, kontrak sebaiknya memisahkan antara jenis pelanggaran, misalnya:

- keterlambatan;
- kegagalan memenuhi SLA;
- kegagalan lulus uji;
- pelanggaran keamanan;
- pelanggaran kerahasiaan.

Tiap jenis pelanggaran bisa punya *remedy* berbeda, misalnya:

- perbaikan wajib;
- *service credit*;
- penggantian personil;
- perpanjangan stabilisasi;
- denda;
- terminasi untuk kondisi berat.

Yang terpenting, semua ini harus bisa ditautkan kembali ke *deliverable*, pengujian, dan tingkat layanan yang memang sudah ditulis sejak awal. Kalau tidak, penalti dan *remedy* hanya akan menjadi ancaman abstrak yang sulit dijalankan. Pendekatan ini konsisten dengan perlunya kontrak yang sinkron dengan KAK, spesifikasi, dan parameter layanan. (JDIH LKPP)

Penutup

Bab ini menegaskan bahwa kontrak pengadaan IT bukan sekadar instrumen hukum, tetapi juga **instrumen kendali mutu, layanan, risiko, dan transisi**. Kontrak yang baik harus:

- sinkron dengan KAK, spesifikasi, dan HPS;
- jujur membedakan barang dan layanan;
- merinci *deliverable* teknis maupun non teknis;
- mengatur *source code*, dokumentasi, akun admin, dan pengujian;
- memuat SLA, garansi, stabilisasi, dan serah terima;
- mengatur *knowledge transfer*, *exit management*, hak pakai, dan hak modifikasi;
- menyediakan mekanisme sengketa dan *remedy* yang proporsional.

Jika semua ini ditulis dengan disiplin, pengadaan akan lebih siap audit, lebih kecil resikonya terhadap *vendor lock-in*, dan lebih besar peluangnya menghasilkan sistem yang benar-benar bisa dipakai dan dipelihara. Arah ini selaras dengan kerangka pelaksanaan PBJ melalui penyedia, SPBE, dan kewajiban pelindungan data pribadi. (JDIH LKPP)

Tabel Summary. Rancangan Kontrak, *Deliverable*, UAT, dan SLA

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
10.1 Menyelaraskan KAK, spesifikasi, HPS, dan kontrak	Kontrak IT yang baik harus menjadi kelanjutan logis dari **KAK, spesifikasi, dan HPS**. Jika empat dokumen ini tidak sinkron, maka ruang lingkup, biaya, mutu, dan kewajiban hasil kerja akan mudah bertabrakan saat pelaksanaan.	Sebelum kontrak ditetapkan, tim harus memastikan setiap kewajiban utama dapat ditelusuri silang: masalah dan output di KAK, parameter teknis di spesifikasi, kewajaran biaya di HPS, dan mekanisme tagih-uji di kontrak.
10.2 Kontrak berbasis barang vs berbasis layanan	Pengadaan IT sering merupakan **campuran barang dan layanan**. Salah membaca sifat paket akan membuat parameter keberhasilan menjadi keliru.	Kontrak harus membedakan mana bagian yang diukur dari pengiriman barang, dan mana yang diukur dari kualitas layanan seperti implementasi, integrasi, support, operasi, dan pemulihan insiden.
10.3 Deliverable minimal proyek IT	Proyek IT tidak cukup hanya menyerahkan aplikasi atau perangkat yang aktif. Kontrak harus menetapkan **deliverable minimal** secara eksplisit agar tidak terjadi sengketa tafsir.	Kontrak perlu mencantumkan deliverable teknis dan nonteknis: dokumen analisis, desain, konfigurasi, hasil pengujian, dokumentasi integrasi, manual, materi pelatihan, dan laporan stabilisasi.
10.4 Source code, repo, dokumentasi, akun admin, SOP, diagram, pelatihan	Pada proyek digital, kontrol atas **source code, repo, dokumentasi, akun admin, SOP, diagram, dan pelatihan** menentukan apakah organisasi sungguh menguasai hasil proyek atau tetap tergantung pada vendor.	Kontrak harus jelas soal apa yang wajib diserahkan, hak akses apa yang diberikan, dan bagaimana transfer pengetahuan dilakukan agar sistem tidak menjadi “kotak hitam” bagi institusi.
10.5 UAT, security test, integration test, performance test	Uji terima harus dirancang sebagai **mekanisme pembuktian**, bukan formalitas. UAT saja tidak cukup; proyek IT juga perlu pengujian integrasi, performa, dan keamanan sesuai risiko sistem.	Kontrak harus menuliskan jenis pengujian, kriteria lulus, siapa yang menguji, bukti hasil uji, serta mekanisme perbaikan bila hasil belum memenuhi syarat.

10.6 SLA, SLO, masa garansi, masa stabilisasi	SLA, SLO, garansi, dan stabilisasi adalah empat hal yang berbeda dan harus dibedakan jelas dalam kontrak. Semuanya penting untuk memastikan layanan tetap terjaga setelah go-live.	Tuliskan parameter yang bisa dibuktikan seperti uptime, waktu respons, waktu pemulihan, masa observasi pasca go-live, dan konsekuensi bila target tidak tercapai.
10.7 Serah terima dan knowledge transfer	Serah terima proyek IT tidak boleh hanya berupa tanda tangan berita acara. Serah terima harus berarti bahwa institusi telah menerima **sistem sekaligus pengetahuan minimum** untuk mengoperasikannya.	Pastikan serah terima mencakup aset/komponen, dokumen teknis, dokumen operasional, akun sesuai haknya, hasil uji, pelatihan admin/operator, dan daftar isu terbuka.
10.8 Klausul exit management dan transisi vendor	Kontrak IT yang matang harus memikirkan **akhir hubungan** sejak awal. Klausul exit management penting untuk mencegah vendor lock-in dan melindungi kesinambungan layanan.	Kontrak perlu mengatur ekspor data, dokumentasi konfigurasi, bantuan masa transisi, kerja sama dengan penyedia baru, dan larangan menghambat alih kelola.
10.9 Klausul kerahasiaan, hak pakai, dan hak modifikasi	Kontrak harus membedakan dengan jelas antara **kerahasiaan, hak pakai, dan hak modifikasi**, karena ketiganya tidak sama dan berimplikasi langsung pada kepatuhan dan kemandirian operasional.	Tim harus menulis secara tegas siapa boleh mengakses apa, apa yang boleh digunakan, apa yang boleh diubah, dan batas kewajiban kerahasiaan terutama bila sistem memproses data pribadi.
10.10 Sengketa, penalti, dan remedy	Kontrak harus punya mekanisme **sengketa, penalti, dan remedy** yang proporsional. Tanpa itu, organisasi lemah saat menagih kewajiban; tetapi kalau terlalu kaku, ruang penyelesaian profesional juga bisa rusak.	Pisahkan jenis pelanggaran seperti keterlambatan, gagal lulus uji, gagal memenuhi SLA, pelanggaran keamanan, atau pelanggaran kerahasiaan, lalu tautkan remedy-nya ke deliverable dan standar layanan yang sudah ditulis.

Ringkasan

Bab 10 menegaskan bahwa kontrak pengadaan IT adalah instrumen kendali utama atas mutu, layanan, risiko, dan transisi. Kontrak yang baik harus sinkron dengan KAK, spesifikasi, dan HPS; harus merinci deliverable teknis maupun non teknis; harus mengatur uji terima, SLA, garansi, stabilisasi, serah terima, transfer pengetahuan, dan exit management; serta harus tegas soal hak pakai, hak modifikasi, kerahasiaan, dan remedy bila terjadi pelanggaran. Dengan kontrak seperti itu, pengadaan menjadi lebih siap audit, lebih kecil resikonya terhadap sengketa dan *vendor lock-in*, serta lebih besar peluangnya menghasilkan sistem yang benar-benar bisa dipakai dan dipelihara.

Bagian IV. Kepatuhan: PDP, Keamanan, Cloud, PSE, *eLearning*, *eCommerce*

Bab 11. Pelindungan Data Pribadi dalam Pengadaan IT

11.1 Data pribadi sebagai objek yang harus dipetakan sejak awal

Dalam banyak proyek IT, pelindungan data pribadi masih sering dibahas terlalu akhir, seolah-olah itu hanya urusan tim hukum atau keamanan setelah sistem hampir selesai. Pendekatan ini keliru. Dalam pengadaan IT, data pribadi harus dipetakan sejak awal, bahkan sebelum KAK dan spesifikasi ditulis final. Alasannya sederhana: jenis data yang diproses akan menentukan desain arsitektur, kontrol akses, model integrasi, retensi, kebutuhan keamanan, sampai isi kontrak. **UU No. 27 Tahun 2022** mendefinisikan *Data Pribadi* sebagai data tentang orang perseorangan yang teridentifikasi atau dapat diidentifikasi, baik secara langsung maupun tidak langsung, dan mendefinisikan *Pelindungan Data Pribadi* sebagai upaya menyeluruh dalam rangkaian pemrosesan data pribadi. (UU No. 27 Tahun 2022, JDIH Komdigi – UU No. 27 Tahun 2022) (Peraturan BPK)

Artinya, ketika proyek mengelola data seperti:

- akun pegawai;
- data mahasiswa;
- data pasien;
- data pelanggan;
- log aktivitas;
- lokasi;
- riwayat transaksi;
- identitas pengguna,

maka proyek tersebut sudah masuk ke wilayah tata kelola PDP. Karena itu, tahap awal pengadaan seharusnya menghasilkan **peta data pribadi** yang cukup jelas, yaitu:

- data apa yang diproses;
- dari mana sumbernya;
- untuk tujuan apa dipakai;
- siapa yang mengakses;
- di sistem mana disimpan;
- ke mana dibagikan;
- risiko apa yang melekat.

Tanpa peta ini, organisasi akan kesulitan menulis kontrol yang tepat dalam desain, spesifikasi, kontrak, dan operasional. Itu adalah konsekuensi langsung dari definisi dan ruang lingkup pelindungan data dalam UU PDP. (UU No. 27 Tahun 2022) (Peraturan BPK)

11.2 Peran pengendali dan prosesor

Salah satu sumber kebingungan paling umum dalam proyek IT adalah tidak jelasnya siapa **Pengendali Data Pribadi** dan siapa **Prosesor Data Pribadi**. Dalam banyak pengadaan, instansi mengira penyedia “yang mengelola data”, padahal dari sisi hukum belum tentu penyedia memegang posisi pengendali. Di sinilah pembedaan peran menjadi sangat penting. **UU PDP** memang mengatur kewajiban bagi Pengendali Data Pribadi dan Prosesor Data Pribadi, termasuk situasi ketika pemrosesan dilakukan oleh dua atau lebih pengendali yang harus diatur lewat perjanjian mengenai peran, tanggung jawab, hubungan antar pihak, tujuan pemrosesan yang saling berkaitan, dan narahubung yang ditunjuk. (UU No. 27 Tahun 2022, JDIH Komdigi – UU PDP) (Peraturan BPK)

Dalam praktik pengadaan, pembagian peran harus ditulis tegas. Secara umum:

- instansi atau badan publik biasanya bertindak sebagai **pengendali** untuk tujuan layanan publiknya;
- penyedia dapat bertindak sebagai **prosesor** jika memproses data atas nama dan atas instruksi pengendali;
- pada proyek integrasi lintas lembaga atau *shared service*, hubungan antar pihak bisa lebih kompleks dan harus dirumuskan lebih rinci.

Kalau peran ini kabur, maka tanggung jawab juga akan kabur saat terjadi insiden, sengketa, atau audit. Karena itu, penegasan peran tidak boleh ditunda sampai akhir proyek. (UU No. 27 Tahun 2022) (Peraturan BPK)

11.3 Dasar pemrosesan data

Pengadaan IT tidak cukup hanya bertanya, “apakah sistem ini berguna?” Ia juga harus bertanya: **apa dasar pemrosesan data pribadinya?** Sistem yang baik secara teknis tetap bisa bermasalah bila dasar pemrosesannya tidak jelas. **UU PDP** secara tegas mewajibkan Pengendali Data Pribadi memiliki dasar pemrosesan Data Pribadi. Untuk konteks pelayanan publik, dasar itu dapat terkait kewajiban hukum, kepentingan umum dalam rangka penyelenggaraan negara, atau dasar lain yang sah menurut UU PDP. Penjelasan UU juga memberi contoh kepentingan umum dalam penyelenggaraan negara seperti administrasi kependudukan, jaminan sosial, perpajakan, kepabeanan, dan pelayanan perizinan berusaha terintegrasi secara elektronik. (UU No. 27 Tahun 2022, PDF UU No. 27 Tahun 2022) (Peraturan BPK)

Implikasinya sangat praktis. KAK dan desain sistem seharusnya sejak awal menjelaskan:

- tujuan pemrosesan;
- kategori data yang dibutuhkan;
- dasar hukum atau dasar pemrosesannya.

Pendekatan ini membantu membatasi pengumpulan data agar tidak berlebihan dan mencegah proyek memproses data hanya karena “sekalian bisa”. Dalam tata kelola PDP, pengumpulan data yang tidak perlu justru memperbesar risiko tanpa menambah manfaat. (UU No. 27 Tahun 2022) (Peraturan BPK)

11.4 Klasifikasi data

Tidak semua data dalam proyek IT memiliki tingkat sensitivitas yang sama. Karena itu, salah satu langkah penting dalam pengadaan adalah **klasifikasi data**. Tanpa klasifikasi, organisasi akan cenderung memberi perlakuan terlalu longgar pada data sensitif, atau sebaliknya terlalu berat pada data yang resikonya rendah. Dalam kerangka **UU PDP**, klasifikasi penting karena ada pemrosesan yang berpotensi risiko tinggi, termasuk pemrosesan skala besar, pemrosesan data pribadi yang bersifat spesifik, pencocokan atau penggabungan data, penggunaan teknologi baru, atau pemrosesan yang berdampak signifikan terhadap subjek data. Dalam kondisi seperti itu, UU PDP mewajibkan **penilaian dampak pelindungan data pribadi**. (UU No. 27 Tahun 2022, PDF UU PDP) (Peraturan BPK)

Secara praktis, proyek pengadaan sebaiknya minimal membedakan:

- data identitas dasar;
- data transaksi;
- log aktivitas;
- data perilaku atau *profiling*;
- data sensitif atau spesifik, bila ada;
- data hasil integrasi dari pihak lain.

Dari klasifikasi itu, organisasi baru bisa menurunkan kontrol yang proporsional, seperti:

- kontrol akses;
- enkripsi;
- *logging*;
- kewajiban audit;
- retensi;
- kebutuhan respons insiden.

Klasifikasi data pada dasarnya adalah fondasi untuk menentukan tingkat perlindungan yang tepat. (UU No. 27 Tahun 2022) (Peraturan BPK)

11.5 *Privacy by design* dan *privacy by default*

Di banyak proyek digital, privasi masih diperlakukan sebagai tambahan: nanti setelah sistem jadi, baru dipikirkan persetujuan, penghapusan, atau pembatasan akses. Cara ini tidak cukup. Pengadaan IT yang baik harus mendorong pendekatan ***privacy by design* dan *privacy by default***. Walau istilah itu tidak selalu ditulis dengan frasa yang sama dalam UU PDP, semangatnya sangat selaras dengan kewajiban dalam undang-undang tersebut: pelindungan harus dibangun ke dalam desain, bukan ditempelkan belakangan. **UU PDP** mewajibkan pengendali data melindungi dan memastikan keamanan data pribadi yang diprosesnya, dan untuk pemrosesan berisiko tinggi mewajibkan penilaian dampak PDP. (UU No. 27 Tahun 2022) (Peraturan BPK)

Dalam konteks pengadaan, prinsip ini harus diterjemahkan ke spesifikasi dan kontrak. Misalnya:

- data yang wajib dikumpulkan dibatasi pada yang benar-benar diperlukan;
- *audit trail* diaktifkan;
- akun admin dibatasi berdasarkan peran;
- pengaturan default tidak membuka data lebih luas dari yang perlu;
- perubahan konfigurasi yang menyentuh data dicatat.

Privasi yang baik lahir dari keputusan desain yang disiplin, bukan hanya dari kebijakan tertulis. (UU No. 27 Tahun 2022) (Peraturan BPK)

11.6 Klausul PDP dalam KAK dan kontrak

Jika pelindungan data pribadi memang penting, maka ia harus masuk ke **KAK** dan **kontrak**. Tidak cukup hanya disebut dalam presentasi atau rapat koordinasi. Dalam KAK, unsur PDP sebaiknya muncul pada:

- latar belakang;
- ruang lingkup;
- kebutuhan integrasi;
- kebutuhan keamanan;
- kebutuhan *logging*;
- kebutuhan retensi;
- *deliverable* dokumentasi.

Dalam kontrak, unsur PDP perlu diterjemahkan menjadi kewajiban yang lebih operasional, seperti:

- siapa pengendali dan siapa prosesor;
- tujuan pemrosesan;
- batas instruksi pemrosesan;
- kewajiban kerahasiaan;
- kewajiban keamanan;
- notifikasi insiden;
- bantuan pada audit;
- bantuan untuk pemenuhan hak subjek data;
- penghapusan atau pengembalian data setelah kontrak berakhir.

Pendekatan ini konsisten dengan ketentuan UU PDP mengenai kewajiban pengendali, prosesor, dan pengaturan hubungan antar pengendali bila ada lebih dari satu pihak. (UU No. 27 Tahun 2022) (Peraturan BPK)

Untuk proyek yang melibatkan PSE privat atau *cloud/SaaS*, klausul kontrak juga perlu lebih hati-hati terhadap akses, lokasi data bila relevan, dan mekanisme respons terhadap permintaan akses atau penegakan hukum. Dalam konteks ini, rujukan penting ada pada **Permen Kominfo No. 5 Tahun 2020 tentang Penyelenggara Sistem Elektronik Lingkup Privat**, sebagaimana disebut juga dalam evaluasi regulasi JDIH Komdigi telah diubah oleh **Permen Kominfo No. 10 Tahun 2021**. (Permen Kominfo No. 5 Tahun 2020, PDF Permen Kominfo No. 5 Tahun 2020) (JDIH Kemkomdigi)

11.7 *Data retention*, penghapusan, dan akses subjek data

Banyak sistem rajin mengumpulkan data, tetapi lemah saat ditanya: **berapa lama data disimpan, kapan dihapus, dan bagaimana subjek data dapat mengakses atau mengoreksinya**. Di sinilah tata kelola retensi menjadi penting. Dalam kerangka **UU PDP**, pemrosesan data harus sesuai tujuan dan dapat dipertanggungjawabkan. Itu berarti organisasi tidak semestinya menyimpan data selamanya tanpa dasar yang jelas. Hak subjek data dalam UU PDP juga menuntut organisasi menyiapkan proses untuk akses, koreksi, dan permintaan lain yang relevan secara tertib. (UU No. 27 Tahun 2022, PDF UU PDP) (Peraturan BPK)

Dalam pengadaan, hal ini sebaiknya ditulis ke desain dan spesifikasi, setidaknya meliputi:

- kebijakan retensi per kategori data;
- mekanisme penghapusan atau anonimisasi;
- log perubahan;
- antarmuka atau prosedur untuk memenuhi hak subjek data secara terkendali.

Tanpa itu, sistem mudah berubah menjadi gudang data tanpa tata kelola. (UU No. 27 Tahun 2022) (Peraturan BPK)

11.8 Insiden kebocoran data dan kewajiban respons

Tidak ada sistem yang kebal seratus persen. Karena itu, pengadaan IT harus mengasumsikan bahwa insiden bisa terjadi dan sejak awal menyiapkan **kewajiban respons**. **UU PDP** mewajibkan pengendali data melindungi dan memastikan keamanan data pribadi, dan untuk pemrosesan berisiko tinggi menuntut perlindungan yang berkelanjutan. Dalam praktik, itu berarti organisasi harus punya kemampuan yang jelas untuk:

- mendeteksi insiden;
- melakukan eskalasi;
- mengisolasi masalah;
- melaporkan;
- memulihkan layanan.

Dalam kerangka SPBE, audit aplikasi dan keamanan juga menegaskan bahwa kontrol teknis dan tata kelola insiden tidak bisa dipisah dari layanan digital pemerintah. (UU No. 27 Tahun 2022, Perpres No. 95 Tahun 2018 tentang SPBE) (Peraturan BPK)

Karena itu, kontrak dan SOP sebaiknya menulis setidaknya:

- definisi insiden;
- jalur pelaporan;
- batas waktu notifikasi internal;
- kewajiban bantuan investigasi;
- preservasi log;
- tindakan pemulihan;
- peran masing-masing pihak.

Untuk sistem yang melibatkan vendor, klausul seperti ini sangat penting agar saat insiden terjadi tidak ada saling lempar tanggung jawab. (UU No. 27 Tahun 2022) (Peraturan BPK)

11.9 *Checklist* PDP untuk vendor

Bab ini layak ditutup dengan *checklist* PDP untuk vendor, karena banyak organisasi terlalu fokus pada fitur dan harga, tetapi kurang menilai kesiapan vendor dalam mengelola data pribadi. *Checklist* minimum sebaiknya mencakup:

- apakah vendor memahami perannya sebagai pengendali atau prosesor;
- apakah vendor memiliki kebijakan keamanan dan kerahasiaan;
- apakah ada kontrol akses berbasis peran;
- apakah *logging* dan *audit trail* tersedia;
- apakah ada mekanisme notifikasi insiden;
- apakah ada prosedur retensi dan penghapusan data;
- apakah ada dukungan untuk pemenuhan hak subjek data;
- apakah vendor mampu mendukung penilaian dampak PDP untuk pemrosesan berisiko tinggi;
- apakah dokumentasi teknis dan kontraktualnya cukup jelas.

Untuk layanan *cloud* atau PSE privat, *due diligence* tambahan juga penting, terutama dengan merujuk pada kerangka PSE lingkup privat di Komdigi. (Permen Kominfo No. 5 Tahun 2020, UU No. 27 Tahun 2022) (JDIH Kemkomdigi)

Pesan akhirnya sederhana: **vendor yang baik bukan hanya yang bisa membangun sistem, tetapi juga yang bisa menunjukkan bahwa sistem itu dikelola dengan disiplin pelindungan data yang memadai**. (UU No. 27 Tahun 2022, Permen Kominfo No. 5 Tahun 2020) (Peraturan BPK)

Penutup

Bab ini menegaskan bahwa **pelindungan data pribadi harus masuk ke jantung pengadaan IT**, bukan menjadi catatan tambahan di akhir proyek. Organisasi perlu:

- memetakan data sejak awal;
- membedakan peran pengendali dan prosesor;
- memastikan dasar pemrosesan jelas;
- mengklasifikasikan data;
- membangun *privacy by design* dan *privacy by default*;
- menerjemahkan semuanya ke KAK, spesifikasi, kontrak, retensi, SOP insiden, dan evaluasi vendor.

Dengan pendekatan seperti ini, proyek IT bukan hanya lebih patuh hukum, tetapi juga lebih aman, lebih terukur, dan lebih siap diaudit. Arah ini didukung langsung oleh **UU No. 27 Tahun 2022**, dan untuk konteks sistem elektronik privat juga perlu dibaca bersama regulasi **PSE Lingkup Privat** di Komdigi. (UU No. 27 Tahun 2022, Permen Kominfo No. 5 Tahun 2020, Perpres No. 95 Tahun 2018) (Peraturan BPK)

Tabel Summary. Pelindungan Data Pribadi dalam Pengadaan IT

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
11.1 Data pribadi sebagai objek yang harus dipetakan sejak awal	Dalam pengadaan IT, **data pribadi harus dipetakan sejak awal**, bukan dibahas belakangan. Jenis data yang diproses akan mempengaruhi desain sistem, kontrol akses, integrasi, retensi, keamanan, dan isi kontrak.	Sebelum KAK dan spesifikasi difinalkan, instansi perlu membuat **peta data pribadi**: jenis data, sumber, tujuan pemrosesan, pihak yang mengakses, lokasi penyimpanan, alur berbagi data, dan risiko utamanya.
11.2 Peran pengendali dan prosesor	Salah satu titik paling penting adalah membedakan **pengendali data pribadi** dan **prosesor data pribadi**. Jika peran ini kabur, tanggung jawab juga akan kabur saat terjadi insiden atau sengketa.	Dalam KAK dan kontrak, harus jelas siapa yang menentukan tujuan pemrosesan, siapa yang memproses atas instruksi, siapa PIC-nya, dan bagaimana hubungan antar pihak diatur.
11.3 Dasar pemrosesan data	Sistem yang baik secara teknis tetap bermasalah bila **dasar pemrosesan data** tidak jelas. Pengendali data wajib memiliki dasar pemrosesan yang sah.	KAK, desain, dan arsitektur data harus menjelaskan tujuan pemrosesan, kategori data yang diperlukan, dan dasar hukumnya agar pengumpulan data tidak berlebihan dan lebih mudah dipertanggungjawabkan.
11.4 Klasifikasi data	Tidak semua data punya sensitivitas yang sama. Karena itu, proyek IT perlu **klasifikasi data** agar perlindungannya proporsional dengan tingkat risiko.	Tim perlu membedakan data identitas dasar, data transaksi, log aktivitas, data sensitif/spesifik, dan data hasil integrasi agar kontrol akses, enkripsi, audit, dan retensinya bisa dirancang dengan tepat.
11.5 Privacy by design dan privacy by default	Pelindungan data tidak boleh menjadi tempelan setelah sistem selesai. Prinsipnya, privasi harus dibangun **di dalam desain** dan **di konfigurasi default** sistem.	Spesifikasi dan desain harus sejak awal memuat minimisasi data, pembatasan akses, logging, enkripsi, retensi, penghapusan, dan pengaturan default yang tidak terlalu terbuka.

11.6 Klausul PDP dalam KAK dan kontrak	Jika PDP penting, maka PDP harus tertulis dalam **KAK** dan **kontrak**. Tidak cukup hanya dibahas di rapat atau presentasi.	Dalam KAK, PDP harus muncul pada ruang lingkup, keamanan, logging, dan deliverable. Dalam kontrak, harus ada pengaturan tentang peran para pihak, tujuan pemrosesan, keamanan, notifikasi insiden, audit, dan penghapusan/pengembalian data.
11.7 Data retention, penghapusan, dan akses subjek data	Sistem tidak cukup hanya bisa mengumpulkan data; ia juga harus mampu mengelola **siklus hidup data**, termasuk berapa lama disimpan, kapan dihapus, dan bagaimana subjek data mengakses atau mengoreksinya.	Dalam desain dan spesifikasi, perlu ada kebijakan retensi per kategori data, mekanisme penghapusan/anonimisasi, log perubahan, dan prosedur pemenuhan hak subjek data.
11.8 Insiden kebocoran data dan kewajiban respons	Pengadaan IT harus mengasumsikan bahwa **insiden bisa terjadi** dan karena itu perlu menyiapkan tanggung jawab respons sejak awal.	Kontrak dan SOP harus mengatur definisi insiden, jalur pelaporan, batas waktu eskalasi, bantuan investigasi, preservasi log, dan langkah pemulihan agar tidak terjadi saling lempar tanggung jawab saat insiden muncul.
11.9 Checklist PDP untuk vendor	Vendor yang baik bukan hanya yang bisa membangun sistem, tetapi juga yang mampu menunjukkan **kesiapan tata kelola pelindungan data pribadi**.	Gunakan checklist vendor: pemahaman peran, kontrol akses, logging, notifikasi insiden, retensi/penghapusan data, dukungan hak subjek data, kesiapan audit, dan dokumentasi teknis/kontraktual yang memadai.

Ringkasan

Bab 11 menegaskan bahwa pelindungan data pribadi harus ditempatkan di jantung pengadaan IT, bukan sebagai catatan tambahan di akhir proyek. Organisasi perlu memetakan data sejak awal, memperjelas peran pengendali dan prosesor, memastikan dasar pemrosesan sah, mengklasifikasikan data, menerapkan *privacy by design* dan *privacy by default*, lalu menerjemahkan semuanya ke dalam KAK, spesifikasi, kontrak, retensi, SOP insiden, dan evaluasi vendor. Dengan pendekatan ini, pengadaan IT menjadi **lebih patuh hukum, lebih aman, lebih siap audit, dan lebih bertanggung jawab terhadap warga atau pengguna**.

Bab 12. Keamanan Siber, Audit, dan Ketahanan Operasional

12.1 Keamanan sebagai bagian inti spesifikasi

Dalam banyak proyek IT, keamanan masih sering diperlakukan sebagai lapisan tambahan: sistem dibangun lebih dulu, urusan keamanan dibahas belakangan. Pendekatan ini keliru dan mahal. Dalam pengadaan IT, keamanan harus menjadi **bagian inti dari spesifikasi**, bukan lampiran di akhir. Jika keamanan tidak masuk sejak tahap kebutuhan, desain, KAK, spesifikasi, HPS, dan kontrak, organisasi biasanya hanya membeli sistem yang “berfungsi”, tetapi belum tentu aman, tahan gangguan, dan siap diaudit.

Kerangka **SPBE** sendiri menempatkan keamanan sebagai salah satu domain arsitektur nasional, sejajar dengan proses bisnis, data, aplikasi, dan infrastruktur. **Perpres 95 Tahun 2018** juga menegaskan bahwa manajemen keamanan informasi bertujuan menjamin keberlangsungan SPBE dengan meminimalkan dampak risiko keamanan informasi. Artinya, keamanan bukan urusan teknis sempit, melainkan bagian dari tata kelola layanan pemerintahan digital. (Peraturan BPK)

Karena itu, spesifikasi pengadaan IT yang sehat sebaiknya menulis kebutuhan keamanan dalam bentuk yang bisa diuji, seperti:

- model akses
- *logging*
- enkripsi
- *patching*
- *backup*
- pemantauan
- uji keamanan
- tanggung jawab saat insiden

Kalau keamanan dianggap penting, maka keamanan harus terlihat jelas di dokumen pengadaan. Ini adalah konsekuensi praktis dari penempatan keamanan sebagai domain inti SPBE. (Peraturan BPK)

12.2 Kontrol akses, MFA, logging, enkripsi, *hardening*

Lapisan pertama keamanan operasional biasanya ada pada **kontrol akses, MFA, logging, enkripsi, dan hardening**. Ini terdengar teknis, tetapi justru merupakan kontrol dasar yang paling menentukan apakah sistem akan mudah disalahgunakan atau tidak.

Secara praktis:

- **Kontrol akses** berarti sistem membedakan siapa yang boleh melihat, mengubah, menghapus, atau mengekspor data.
- **MFA** penting terutama untuk akses administratif, akses jarak jauh, dan sistem yang memproses data sensitif.
- **Logging** dan *audit trail* dibutuhkan agar aktivitas penting dapat ditelusuri.
- **Enkripsi** penting untuk melindungi data saat transit maupun saat tersimpan.
- **Hardening** berarti menonaktifkan konfigurasi lemah, akun default, layanan yang tidak perlu, dan celah dasar yang membuat sistem mudah diserang.

Dalam pengadaan, lima hal ini seharusnya ditulis sebagai kebutuhan minimum, bukan fitur premium. (Peraturan BPK)

Kerangka hukum Indonesia juga mendukung pendekatan ini. **PP 71 Tahun 2019** mewajibkan Penyelenggara Sistem Elektronik menyelenggarakan sistem secara andal, aman, dan bertanggung jawab. Sementara itu, **UU 1 Tahun 2024** sebagai perubahan kedua atas UU ITE tetap menjadi dasar hukum penting bagi penyelenggaraan sistem elektronik dan bukti elektronik. Jadi, kontrol dasar keamanan bukan hanya kebutuhan teknis, tetapi bagian dari kewajiban penyelenggaraan sistem elektronik yang sah. (Peraturan BPK)

12.3 *Vulnerability management* dan *patching*

Banyak organisasi membeli sistem yang baik di hari pertama, tetapi lupa bahwa ancaman terus berubah. Karena itu, *vulnerability management* dan *patching* harus dipahami sebagai **proses berkelanjutan**, bukan pekerjaan sekali selesai.

Secara sederhana:

- **Vulnerability management** berarti organisasi memiliki cara untuk mengidentifikasi kelemahan, memprioritaskan risiko, memperbaiki celah, dan memverifikasi hasil perbaikannya.
- **Patching** berarti pembaruan keamanan dilakukan secara teratur, terencana, dan terdokumentasi.

Tanpa dua hal ini, sistem yang awalnya aman akan cepat menjadi rapuh. Bahkan kontrol akses dan enkripsi yang baik pun tidak cukup jika sistem operasi, aplikasi, atau dependensi dibiarkan usang.

Dalam pengadaan, kebutuhan ini harus diterjemahkan menjadi kewajiban yang jelas, misalnya:

- siapa yang memonitor kerentanan
- seberapa cepat *patch* kritikal harus dipasang
- bagaimana pengujian *patch* dilakukan
- bagaimana pengecualian dikelola
- bagaimana bukti pelaksanaan disimpan

Keamanan operasional yang matang biasanya lebih dekat ke disiplin proses daripada sekadar pembelian alat. Ini selaras dengan prinsip sistem elektronik yang harus andal, aman, dan bertanggung jawab menurut PP 71 Tahun 2019. (Peraturan BPK)

12.4 SIEM, SOC, IR, dan forensik dasar

Untuk sistem yang penting, organisasi tidak cukup hanya punya pencegahan. Mereka juga perlu kemampuan untuk **mendeteksi, merespons, dan menyelidiki insiden**. Di sinilah peran SIEM, SOC, IR, dan forensik dasar menjadi relevan.

Secara ringkas:

- **SIEM** membantu mengumpulkan dan menganalisis log dari berbagai sumber.
- **SOC** adalah fungsi atau layanan yang memantau kejadian keamanan dan melakukan triase.
- **IR** (*incident response*) adalah proses menangani insiden secara sistematis: deteksi, isolasi, analisis, pemulihan, dan pelaporan.
- **Forensik dasar** penting agar bukti digital tidak rusak dan akar kejadian bisa dipahami.

Tidak semua organisasi harus membangun SOC penuh sejak awal. Tetapi untuk layanan yang penting, setidaknya harus ada jawaban atas pertanyaan berikut:

- siapa yang memantau
- bagaimana log dikumpulkan
- siapa yang merespons
- bagaimana bukti dipertahankan bila insiden terjadi

Dalam konteks SPBE, audit keamanan dan manajemen keamanan informasi memang menuntut adanya proses yang terencana, dievaluasi, dan diperbaiki secara berkelanjutan. Jadi, pembahasan SIEM dan SOC bukan kemewahan teknis, melainkan bagian dari kesiapan organisasi menjaga layanan digital tetap dapat dipercaya. (Peraturan BPK)

12.5 *Backup*, DR, dan BCP

Jika keamanan berbicara tentang pencegahan dan deteksi, maka **backup, disaster recovery (DR), dan business continuity planning (BCP)** berbicara tentang ketahanan. Ini penting karena tidak ada sistem yang bebas dari kemungkinan gangguan, baik akibat serangan siber, salah konfigurasi, kesalahan manusia, kerusakan perangkat, maupun kegagalan layanan pihak ketiga.

Perannya berbeda-beda:

- **Backup** menjawab apakah data bisa dipulihkan.
- **DR** menjawab bagaimana layanan dipulihkan setelah gangguan serius.
- **BCP** menjawab bagaimana proses penting organisasi tetap berjalan walau sistem utama terganggu.

Dalam banyak proyek, *backup* memang ada, tetapi tidak pernah diuji. DR disebut di proposal, tetapi tidak pernah dijabarkan. BCP dianggap urusan manajemen, bukan bagian dari proyek IT. Pendekatan seperti ini berbahaya, karena ketahanan layanan justru diuji pada hari terburuk, bukan pada hari biasa.

PP 71 Tahun 2019 menekankan bahwa sistem elektronik harus andal, aman, dan bertanggung jawab. Dalam lingkungan SPBE, manajemen keamanan informasi juga diarahkan untuk menjamin keberlangsungan layanan. Ini memberi dasar kuat bahwa *backup*, DR, dan BCP harus dipandang sebagai kebutuhan inti dalam pengadaan sistem penting, bukan biaya tambahan yang bisa ditunda. (Peraturan BPK)

12.6 Audit keamanan SPBE

Dalam pengadaan IT pemerintah, pembahasan keamanan tidak lengkap tanpa menyentuh **audit keamanan SPBE**. **Perpres 95 Tahun 2018** secara jelas mengatur bahwa audit TIK dalam SPBE mencakup audit Infrastruktur SPBE, audit Aplikasi SPBE, dan audit Keamanan SPBE. Audit tersebut meliputi pemeriksaan tata kelola, fungsionalitas, kinerja, dan aspek teknis lainnya. Artinya, sejak tahap pengadaan pun sistem seharusnya sudah didesain agar **bisa diaudit**, bukan hanya bisa dipakai. (Peraturan BPK)

Implikasinya sangat praktis. Sistem perlu memiliki:

- dokumentasi yang cukup
- *logging* yang memadai
- kontrol perubahan yang jelas
- bukti pengujian
- pemisahan peran yang bisa ditelusuri

Tanpa itu, auditor akan kesulitan memverifikasi apa yang sebenarnya terjadi di sistem, dan organisasi akan kesulitan menunjukkan kepatuhan maupun kewajaran pengelolaannya.

Audit keamanan juga penting karena memaksa organisasi membaca keamanan secara lebih objektif. Tidak cukup mengatakan “sistem kami aman”. Yang penting adalah apakah ada:

- bukti tata kelola
- bukti kontrol
- bukti pengujian
- jejak aktivitas yang memadai

Audit yang baik bukan musuh proyek IT; ia adalah cermin kedewasaan tata kelola. (Peraturan BPK)

12.7 Kematangan keamanan dan kesiapan audit

Tidak semua organisasi berada pada tingkat kesiapan yang sama. Karena itu, penting membedakan antara **kematangan keamanan** dan **kesiapan audit**.

Secara sederhana:

- **Kematangan keamanan** berbicara tentang seberapa konsisten organisasi mengelola kontrol: apakah masih reaktif, apakah sudah terdokumentasi, apakah sudah diukur, dan apakah diperbaiki berkelanjutan.
- **Kesiapan audit** berarti organisasi tidak hanya melakukan kontrol, tetapi juga bisa membuktikannya.

Banyak instansi merasa sudah punya keamanan karena memakai *firewall*, antivirus, atau layanan *cloud*. Tetapi ketika ditanya soal:

- bukti *review* akses
- bukti *patching*
- bukti uji *restore backup*
- bukti penanganan insiden

jawabannya belum tentu siap.

Dalam pengadaan, dua konsep ini penting untuk menentukan ruang lingkup yang realistis. Organisasi dengan kematangan rendah mungkin perlu memulai dari:

- kontrol dasar
- dokumentasi
- SOP

Organisasi dengan kematangan lebih tinggi bisa masuk ke:

- SIEM
- SOC
- otomasi
- KPI keamanan yang lebih rinci

Keamanan yang sehat dibangun bertahap, tetapi arah kedewasaannya harus jelas. Ini merupakan inferensi yang kuat dari tuntutan audit dan manajemen keamanan berkelanjutan dalam SPBE. (Peraturan BPK)

12.8 Uji keamanan sebelum *go-live*

Salah satu kebiasaan buruk dalam proyek IT adalah meluncurkan sistem karena tenggat sudah dekat, walaupun pengujian keamanan belum memadai. Bab ini perlu tegas: **uji keamanan sebelum *go-live* bukan kemewahan, melainkan syarat minimum untuk sistem yang penting**.

Jenis uji keamanan tentu berbeda sesuai risiko sistem. Untuk sebagian sistem, pemeriksaan konfigurasi dan *vulnerability scanning* mungkin cukup sebagai *baseline*. Tetapi untuk sistem lain, terutama yang memproses data pribadi, transaksi, atau layanan kritikal, pengujian perlu lebih dalam, misalnya:

- *penetration testing*
- *review* hak akses
- validasi enkripsi
- uji *logging*
- uji pemulihan

Yang penting, hasil uji tidak hanya ada, tetapi juga **ditindaklanjuti** sebelum sistem dinyatakan siap produksi.

Pendekatan ini sejalan dengan **UU PDP** dan **PP 71 Tahun 2019**, yang sama-sama menuntut penyelenggaraan sistem yang aman, andal, dan bertanggung jawab. Sistem yang langsung *go-live* tanpa uji keamanan yang memadai pada dasarnya memindahkan risiko ke pengguna dan institusi. Murah di jadwal, mahal di insiden. (Peraturan BPK)

12.9 Contoh klausul keamanan untuk kontrak

Bab ini layak ditutup dengan **klausul keamanan untuk kontrak**, karena keamanan yang tidak diterjemahkan ke dalam kontrak sering berakhir hanya sebagai niat baik tanpa alat tagih.

Secara substansi, kontrak keamanan yang sehat sebaiknya memuat setidaknya:

- kewajiban kontrol akses dan pengelolaan akun
- kewajiban MFA untuk akses administratif atau akses tertentu
- kewajiban *logging* dan retensi log
- kewajiban enkripsi sesuai kebutuhan
- kewajiban *patching* dan penanganan kerentanan
- kewajiban *backup* dan uji pemulihan
- kewajiban notifikasi insiden
- hak audit atau hak meminta bukti kontrol
- kewajiban menjaga kerahasiaan
- kewajiban kerja sama saat investigasi atau transisi vendor

Jika sistem memproses data pribadi, klausul ini juga harus selaras dengan klausul PDP dan kewajiban pengendali/prosesor menurut **UU PDP**. (Peraturan BPK)

Prinsipnya sederhana: **apa yang dianggap penting untuk keamanan harus bisa ditagih dalam kontrak**. Kalau tidak ditulis, ia mudah dianggap di luar ruang lingkup. Kalau dianggap di luar ruang lingkup, ia akan menjadi celah saat masalah muncul.

Penutup

Bab ini menegaskan bahwa **keamanan siber, audit, dan ketahanan operasional harus diperlakukan sebagai bagian inti dari pengadaan IT**, bukan lapisan tambahan setelah proyek hampir selesai.

Organisasi perlu:

- menulis kontrol dasar seperti akses, MFA, *logging*, enkripsi, dan *hardening*
- membangun proses *vulnerability management* dan *patching*
- menyiapkan kemampuan deteksi dan respons insiden
- memastikan *backup*, DR, dan BCP
- mendesain sistem agar siap diaudit dalam kerangka SPBE

Dengan begitu, pengadaan IT tidak hanya menghasilkan sistem yang berjalan, tetapi juga sistem yang:

- aman
- tangguh
- dapat dibuktikan pengelolaannya
- lebih siap menghadapi gangguan nyata

Arah ini didukung oleh **Perpres 95 Tahun 2018**, **PP 71 Tahun 2019**, dan **UU 1 Tahun 2024** sebagai perubahan kedua UU ITE. (Peraturan BPK)

Tabel Summary. Keamanan Siber, Audit, dan Ketahanan Operasional

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
12.1 Keamanan sebagai bagian inti spesifikasi	Keamanan harus menjadi **bagian inti dari spesifikasi**, bukan tambahan di belakang. Jika tidak masuk sejak tahap kebutuhan, desain, KAK, HPS, dan kontrak, organisasi hanya membeli sistem yang berjalan, tetapi belum tentu aman, tahan gangguan, dan siap audit.	Kebutuhan keamanan harus tertulis jelas sejak awal: kontrol akses, logging, enkripsi, patching, backup, uji keamanan, dan tanggung jawab saat insiden.
12.2 Kontrol akses, MFA, logging, enkripsi, hardening	Lapisan pertama keamanan operasional ada pada **kontrol akses, MFA, logging, enkripsi, dan hardening**. Ini adalah kontrol dasar yang paling menentukan apakah sistem mudah disalahgunakan atau tidak.	Dalam pengadaan, lima kontrol ini sebaiknya ditulis sebagai kebutuhan minimum, terutama untuk akses administratif, data sensitif, dan sistem yang terhubung ke internet.
12.3 Vulnerability management dan patching	Sistem yang aman di hari pertama bisa cepat menjadi rapuh bila tidak ada **pengelolaan kerentanan** dan **patching** yang disiplin. Keamanan harus dipelihara terus-menerus, bukan sekali selesai.	Kontrak dan SOP perlu menetapkan siapa memantau kerentanan, batas waktu patch kritikal, mekanisme pengujian patch, dan bukti pelaksanaannya.
12.4 SIEM, SOC, IR, dan forensik dasar	Untuk sistem penting, organisasi perlu kemampuan **mendeteksi, merespons, dan menyelidiki** insiden. Karena itu, SIEM, SOC, IR, dan forensik dasar menjadi bagian dari kesiapan keamanan operasional.	Tidak semua organisasi harus membangun SOC penuh, tetapi minimal harus jelas siapa memantau log, siapa merespons insiden, dan bagaimana bukti digital dijaga saat kejadian.
12.5 Backup, DR, dan BCP	Ketahanan layanan tidak cukup dengan sistem yang aman; sistem juga harus **bisa dipulihkan** dan proses penting organisasi harus tetap berjalan saat terjadi gangguan.	Pengadaan harus memasukkan kebutuhan backup, target pemulihan, skenario DR, dan rencana kesinambungan bisnis agar layanan tidak lumpuh saat insiden atau kegagalan besar terjadi.

12.6 Audit keamanan SPBE	Audit keamanan SPBE menuntut sistem tidak hanya bisa dipakai, tetapi juga **bisa diaudit**. Audit mencakup infrastruktur, aplikasi, dan keamanan SPBE.	Sistem harus dirancang dengan dokumentasi, logging, kontrol perubahan, dan bukti pengujian yang cukup agar auditor dapat memverifikasi tata kelola dan kontrol secara objektif.
12.7 Kematangan keamanan dan kesiapan audit	Keamanan yang sehat tidak diukur hanya dari alat yang dipakai, tetapi dari **kematangan proses** dan kemampuan organisasi untuk membuktikan kontrol yang dijalankan.	Organisasi dengan kematangan rendah sebaiknya fokus dulu pada kontrol dasar, SOP, dan dokumentasi. Yang lebih matang bisa melangkah ke otomasi, pengukuran KPI keamanan, dan monitoring yang lebih canggih.
12.8 Uji keamanan sebelum go-live	**Uji keamanan sebelum go-live** adalah syarat minimum, terutama untuk sistem yang memproses data pribadi, transaksi, atau layanan penting. Sistem tidak boleh masuk produksi hanya karena tenggat.	KAK, spesifikasi, dan kontrak harus meminta uji keamanan yang proporsional: baseline scan, review konfigurasi, penetration test, review hak akses, dan tindak lanjut hasil uji sebelum sistem dinyatakan siap produksi.
12.9 Contoh klausul keamanan untuk kontrak	Keamanan yang tidak diterjemahkan ke dalam kontrak akan sulit ditagih. Karena itu, klausul keamanan harus memuat kewajiban kontrol, patching, logging, backup, notifikasi insiden, hak audit, dan kerja sama investigasi.	Semua kontrol yang dianggap penting harus tertulis dalam kontrak agar tidak dianggap “di luar ruang lingkup” ketika terjadi insiden, audit, atau sengketa layanan.

Ringkasan

Bab 12 menegaskan bahwa keamanan siber, audit, dan ketahanan operasional harus diperlakukan sebagai bagian inti dari pengadaan IT, bukan sebagai tambahan setelah proyek hampir selesai. Organisasi perlu menanamkan kontrol dasar seperti akses, MFA, logging, enkripsi, dan hardening; membangun proses vulnerability management dan patching; menyiapkan deteksi dan respons insiden; memastikan backup, DR, dan BCP; serta merancang sistem agar siap diaudit dalam kerangka SPBE. Dengan pendekatan seperti ini, pengadaan IT tidak hanya menghasilkan sistem yang berjalan, tetapi juga **sistem yang aman, tangguh, terkelola, dan lebih siap menghadapi gangguan nyata**.

Bab 13. *Cloud*, Pusat Data, dan PSE

13.1 Model *cloud*: IaaS, PaaS, SaaS

Dalam pengadaan IT, istilah *cloud* sering dipakai terlalu longgar, seolah semua layanan awan itu sama. Padahal dari sisi pengadaan, **IaaS, PaaS, dan SaaS** membawa konsekuensi yang berbeda terhadap ruang lingkup, tanggung jawab, HPS, kontrol, dan risiko. Karena itu, langkah pertama yang sehat adalah membedakan model layanannya secara jelas.

Secara ringkas:

- **IaaS** biasanya memberi fondasi komputasi: mesin virtual, penyimpanan, jaringan, dan komponen infrastruktur dasar.
- **PaaS** mengurangi sebagian beban itu dengan menyediakan platform pengembangan atau *runtime* yang lebih terkelola.
- **SaaS** lebih jauh lagi: organisasi membeli akses ke aplikasi yang sudah jadi, sementara sebagian besar pengelolaan teknis berada di pihak penyedia.

Perbedaan ini penting karena apa yang dibeli, apa yang diuji, dan apa yang harus dikontrakkan akan berubah menurut model layanannya. Semakin tinggi lapisan layanan yang dibeli, semakin besar perhatian harus diberikan pada:

- hak akses;
- portabilitas data;
- konfigurasi;
- integrasi;
- *exit plan*.

Sebaliknya, semakin rendah lapisannya, semakin besar beban organisasi pada:

- operasi;
- keamanan;
- *patching*;
- kapasitas SDM internal.

Jadi, model *cloud* bukan sekadar pilihan teknis, tetapi juga pilihan tata kelola dan pembagian tanggung jawab. Ini konsisten dengan kewajiban penyelenggara sistem elektronik untuk menjaga keandalan, keamanan, dan tanggung jawab sistemnya. (Peraturan BPK)

13.2 *Public cloud*, *private cloud*, *hybrid cloud*

Selain model layanan, pengadaan juga harus membedakan **public cloud, private cloud, dan hybrid cloud**. Banyak keputusan yang salah justru lahir bukan dari pemilihan vendor, tetapi dari asumsi keliru tentang lingkungan penempatan sistem.

Secara umum:

- **Public cloud** biasanya unggul dalam elastisitas, kecepatan penyediaan, dan variasi layanan.
- **Private cloud** memberi kontrol yang lebih besar terhadap lingkungan, kebijakan, dan dalam beberapa kasus kebutuhan tertentu atas data atau infrastruktur.
- **Hybrid cloud** sering menjadi jalan tengah: sebagian beban kerja ditempatkan pada lingkungan yang sangat terkontrol, sementara sebagian lain memanfaatkan elastisitas *public cloud*.

Namun setiap model membawa konsekuensi berbeda. Pada *public cloud*, perhatian biasanya lebih besar pada:

- portabilitas;
- ketergantungan platform;
- tata kelola akses;
- pembacaan biaya berulang.

Pada *private cloud*, tantangannya lebih besar pada:

- investasi;
- kapasitas operasional;
- disiplin pengelolaan.

Pilihan di antara ketiganya tidak boleh didasarkan pada slogan seperti “*cloud* pasti lebih modern” atau “*private* pasti lebih aman”. Yang seharusnya menjadi dasar adalah:

- karakter beban kerja;
- sensitivitas data;
- pola integrasi;
- kebutuhan latensi;
- kesiapan SDM;
- kebutuhan audit;
- strategi keberlanjutan layanan.

Karena itu, keputusan penempatan *cloud* adalah keputusan arsitektur sekaligus keputusan pengadaan. Prinsip ini selaras dengan SPBE yang menempatkan infrastruktur dan keamanan sebagai bagian dari arsitektur layanan digital pemerintah. (Peraturan BPK)

13.3 Kapan *cloud* layak dipilih

Cloud tidak selalu menjadi jawaban terbaik, tetapi dalam banyak situasi memang sangat layak dipilih. Kuncinya adalah membaca konteks penggunaan secara jujur.

Cloud biasanya layak dipilih ketika organisasi membutuhkan:

- kecepatan penyediaan;
- skalabilitas;
- elastisitas beban;
- kemudahan ekspansi;
- pengurangan beban pengelolaan fasilitas fisik tertentu.

Untuk kebutuhan seperti pengembangan, pengujian, *sandbox*, analitik, atau layanan digital yang bertumbuh cepat, *cloud* sering memberi keuntungan operasional yang nyata.

Tetapi *cloud* tidak otomatis cocok untuk semua kasus. Ia bisa menjadi pilihan yang buruk bila organisasi:

- lemah dalam tata kelola vendor;
- tidak disiplin membaca biaya bulanan;
- belum siap mengelola identitas dan akses;
- lemah dalam dokumentasi konfigurasi.

Dalam kondisi seperti itu, *cloud* justru bisa mempercepat masalah, bukan menyelesaikannya. Jadi, pemilihan *cloud* seharusnya dipandang sebagai **keputusan kesiapan**, bukan sekadar keputusan teknologi. Pembacaan ini konsisten dengan kewajiban umum sistem elektronik untuk diselenggarakan secara andal, aman, dan bertanggung jawab. (Peraturan BPK)

13.4 Risiko *vendor lock-in* pada *cloud*

Salah satu risiko paling serius pada *cloud* adalah **vendor lock-in**. Ini bukan sekadar soal sulit pindah vendor. Ini adalah situasi ketika data, konfigurasi, integrasi, *workflow*, atau bahkan kompetensi tim sudah terlalu bergantung pada satu platform tertentu, sehingga biaya keluar menjadi mahal, migrasi menjadi rumit, dan ruang negosiasi organisasi menjadi lemah.

Vendor lock-in pada *cloud* bisa muncul dalam beberapa bentuk:

- penggunaan layanan yang sangat spesifik vendor dan sulit digantikan;
- format data atau proses ekspor yang tidak disiapkan dengan baik;
- dokumentasi konfigurasi yang tidak lengkap;
- pengetahuan praktis yang hanya dimiliki vendor atau integrator.

Pada layanan SaaS, *lock-in* sering lebih halus: aplikasi terasa nyaman dipakai, tetapi saat organisasi ingin pindah, ternyata data, *workflow*, dan integrasinya tidak mudah dipindahkan.

Karena itu, pendekatan sehat terhadap *cloud* selalu bertanya sejak awal:

- jika suatu hari harus pindah, apa yang akan dibawa;
- dalam format apa;
- dengan biaya berapa;
- dalam waktu berapa.

Kalau pertanyaan ini tidak bisa dijawab saat pengadaan, biasanya organisasi sedang masuk ke *cloud* dengan mata setengah tertutup. Risiko ini juga relevan dalam kerangka PSE privat karena ketergantungan layanan harus dibaca bersama kepatuhan penyelenggara sistem elektronik. (JDIH Kemkomdigi)

13.5 *Exit plan*, portabilitas, dan cadangan data

Karena risiko *lock-in* nyata, maka **exit plan, portabilitas, dan cadangan data** harus menjadi bagian eksplisit dari pengadaan *cloud*.

Secara praktis:

- **Exit plan** berarti organisasi sudah memikirkan bagaimana keluar dari layanan.
- **Portabilitas** berarti sistem, data, dan konfigurasi tidak terkurung dalam bentuk yang menyulitkan migrasi.
- **Cadangan data** berarti organisasi tidak menggantungkan kelangsungan layanan pada satu titik kegagalan atau satu relasi kontraktual.

Dalam praktik kontrak, tiga hal ini sebaiknya ditulis jelas, misalnya:

- format ekspor data yang terbaca;
- dokumentasi konfigurasi dan integrasi;
- bantuan transisi;
- batas waktu penghapusan data setelah terminasi;
- hak organisasi atas salinan data cadangan yang wajar.

Untuk *cloud* dan SaaS, pengadaan yang tidak menulis *exit plan* pada dasarnya sedang membeli kenyamanan hari ini dengan risiko besar di hari esok. Prinsip ini juga selaras dengan kewajiban sistem elektronik untuk dikelola secara bertanggung jawab dan dengan kerangka PSE privat yang menuntut tata kelola yang lebih jelas. (Peraturan BPK)

13.6 Kewajiban PSE

Untuk sistem elektronik yang beroperasi di Indonesia, isu **Penyelenggara Sistem Elektronik (PSE)** tidak bisa diabaikan. **PP 71 Tahun 2019** menjadi fondasi umum rezim penyelenggaraan sistem elektronik di Indonesia. Sementara itu, **Permen Kominfo Nomor 5 Tahun 2020** tentang PSE Lingkup Privat, yang kemudian **diubah oleh Permen Kominfo Nomor 10 Tahun 2021**, mengatur kewajiban yang lebih teknis bagi PSE lingkup privat, termasuk pendaftaran dan kewajiban tertentu dalam rangka pengawasan maupun penegakan hukum. (Peraturan BPK)

Dalam konteks pengadaan, ini berarti organisasi perlu membaca sejak awal:

- apakah penyedia *cloud* atau SaaS yang dipakai masuk kategori PSE lingkup privat;
- bagaimana status kepatuhannya dibuktikan;
- apakah narahubung dan mekanisme responsnya jelas;
- apakah kewajiban relevan telah tercermin dalam kontrak dan tata kelola layanannya.

Bagi pembeli pemerintah, kampus, BLUD, BUMD, atau entitas publik lain, kepatuhan PSE bukan isu pinggiran. Ia adalah bagian dari **due diligence penyedia**. (JDIH Kemkomdigi)

13.7 Keandalan dan tanggung jawab penyelenggara sistem elektronik

Selain kewajiban PSE, pengadaan *cloud* juga harus menilai **keandalan dan tanggung jawab penyelenggara sistem elektronik**. Ini penting karena yang dibeli bukan hanya kapasitas teknis, tetapi juga *trust*.

PP 71 Tahun 2019 menegaskan bahwa penyelenggara sistem elektronik wajib menyelenggarakan sistem secara **andal, aman, dan bertanggung jawab**. Ini adalah dasar normatif yang sangat penting untuk pengadaan, karena berarti penyedia tidak cukup hanya "punya layanan", tetapi juga harus bisa menunjukkan tata kelola yang memadai. (Peraturan BPK)

Dalam praktik *due diligence*, keandalan penyedia sebaiknya dibaca melalui beberapa hal, seperti:

- arsitektur ketersediaan layanan;
- bukti SLA;
- riwayat gangguan;
- model *support*;
- jalur eskalasi;
- dokumentasi keamanan;
- *backup*;
- DR;
- mekanisme pemulihan.

Pada titik ini, pembeli sebaiknya menghindari dua ekstrem:

- percaya penuh pada materi pemasaran;
- menuntut jaminan yang mustahil.

Yang dibutuhkan adalah pembuktian yang wajar dan dapat diverifikasi. (Peraturan BPK)

13.8 TTE, log, bukti elektronik, dan tata kelola layanan elektronik

Dalam layanan digital, tidak cukup sistem itu “jalan”. Ia juga harus bisa menghasilkan **jejak elektronik yang dapat dipercaya**. Itulah sebabnya TTE, log, bukti elektronik, dan tata kelola layanan elektronik menjadi sangat penting.

UU ITE, yang telah mengalami perubahan kedua melalui **UU Nomor 1 Tahun 2024**, tetap menjadi fondasi penting bagi pengakuan atas dokumen elektronik, informasi elektronik, transaksi elektronik, dan bukti elektronik di Indonesia. Ini relevan langsung untuk pengadaan *cloud*, SaaS, dan layanan elektronik, karena banyak proses digital pada akhirnya harus bisa dipertanggungjawabkan. (Peraturan BPK)

Dalam konteks pengadaan, pertanyaan yang sehat bukan hanya soal fitur dan kapasitas, tetapi juga:

- apakah sistem punya log yang memadai;
- apakah jejak aktivitas bisa diakses dan diekspor;
- apakah ada dukungan TTE bila diperlukan;
- apakah tata kelola layanan elektronik cukup untuk kebutuhan pembuktian, audit, dan akuntabilitas.

Tanpa itu, layanan mungkin nyaman dipakai, tetapi lemah ketika diuji dalam konteks pengawasan, sengketa, atau pemeriksaan. (Peraturan BPK)

13.9 *Checklist due diligence* penyedia *cloud*/SaaS

Bab ini layak ditutup dengan **checklist due diligence penyedia *cloud*/SaaS**, karena banyak kesalahan pengadaan lahir bukan dari KAK yang buruk, tetapi dari asumsi bahwa semua penyedia *cloud* “kurang lebih sama.” Padahal tidak.

Checklist minimum sebaiknya mencakup beberapa lapisan:

Dari sisi model layanan

- jelas apakah yang dibeli IaaS, PaaS, atau SaaS.

Dari sisi arsitektur

- jelas apakah layanan berada di *public*, *private*, atau *hybrid environment*.

Dari sisi kontrak

- ada SLA;
- ada *exit plan*;
- ada portabilitas data.

Dari sisi kepatuhan

- status PSE jelas;
- rezim hukum yang relevan jelas.

Dari sisi keamanan

- ada kontrol akses;
- ada *logging*;
- ada enkripsi;
- ada *backup*;
- ada DR;
- ada notifikasi insiden.

Dari sisi operasional

- ada *support*;
- ada jalur eskalasi;
- ada dokumentasi.

Dari sisi pembuktian

- ada log;
- ada ekspor data;
- ada bukti layanan elektronik yang memadai.

Inti dari *due diligence* bukan mencari vendor yang tampak paling besar atau paling terkenal, tetapi penyedia yang paling masuk akal dari sisi:

- keandalan;
- kepatuhan;
- kontrol;
- portabilitas;
- kesesuaian dengan kebutuhan organisasi.

Dengan *checklist* yang disiplin, pembeli lebih kecil risikonya masuk ke layanan yang tampak mudah di depan tetapi sulit dikelola di belakang. (Peraturan BPK)

Penutup

Bab ini menegaskan bahwa pengadaan *cloud*, pusat data, dan layanan PSE harus dibaca sebagai **keputusan tata kelola**, bukan sekadar keputusan infrastruktur.

Organisasi perlu:

- memahami model layanan dan model penempatan;
- menilai kapan *cloud* layak dipilih;
- membaca risiko *vendor lock-in*;
- menulis *exit plan* dan portabilitas data;
- memeriksa kewajiban PSE;
- menilai keandalan penyelenggara sistem elektronik;
- memastikan jejak elektronik dan tata kelola layanan mendukung audit dan pembuktian.

Bila semua ini dipikirkan sejak awal, *cloud* bisa menjadi instrumen efisiensi dan percepatan. Tetapi bila diabaikan, *cloud* justru bisa menjadi sumber ketergantungan, biaya tersembunyi, dan risiko kepatuhan yang berat. Kerangka ini didukung oleh **PP 71 Tahun 2019**, **Permen Kominfo 5 Tahun 2020 jo. Permen Kominfo 10 Tahun 2021**, serta **UU 1 Tahun 2024** sebagai perubahan kedua UU ITE. (Peraturan BPK)

Tabel Summary. Cloud, Pusat Data, dan PSE

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
13.1 Model cloud: *IaaS*, *PaaS*, *SaaS*	Model cloud harus dibedakan secara tegas karena **IaaS, PaaS, dan SaaS** membawa konsekuensi yang berbeda pada ruang lingkup, tanggung jawab, kontrol, dan risiko.	Sebelum memilih penyedia, instansi harus jelas: apakah yang dibeli hanya fondasi infrastruktur, platform terkelola, atau aplikasi jadi. Ini akan mempengaruhi KAK, spesifikasi, HPS, dan kontrak.
13.2 *Public cloud*, *private cloud*, *hybrid cloud*	Pilihan lingkungan cloud tidak boleh didasarkan pada slogan. **Public, private, dan hybrid cloud** masing-masing punya konsekuensi berbeda terhadap elastisitas, kontrol, integrasi, audit, dan beban operasional.	Keputusan harus berbasis karakter beban kerja, sensitivitas data, kebutuhan latensi, pola integrasi, kesiapan SDM, dan strategi keberlanjutan layanan.
13.3 Kapan cloud layak dipilih	Cloud layak dipilih ketika organisasi butuh **kecepatan, elastisitas, skalabilitas, dan efisiensi operasional tertentu**, tetapi tidak semua organisasi siap memanfaatkannya dengan disiplin.	Pilih cloud jika manfaat operasionalnya jelas dan organisasi siap mengelola biaya, identitas, akses, konfigurasi, dan pengawasan vendor secara tertib.
13.4 Risiko *vendor lock-in* pada cloud	Risiko terbesar pada cloud sering bukan gangguan teknis, tetapi ***vendor lock-in***: data, konfigurasi, integrasi, dan kompetensi terlalu bergantung pada satu platform.	Saat menyusun pengadaan, tim harus bertanya sejak awal: bagaimana jika suatu hari harus pindah, apa yang bisa dibawa, dalam format apa, dan dengan biaya berapa.
13.5 *Exit plan*, portabilitas, dan cadangan data	Karena risiko lock-in nyata, **exit plan, portabilitas, dan cadangan data** harus menjadi bagian eksplisit dari pengadaan cloud dan SaaS.	Kontrak harus menulis prosedur keluar, format ekspor data, dokumentasi konfigurasi, bantuan masa transisi, dan pengelolaan cadangan data agar layanan tidak terkurung pada satu vendor.

13.6 Kewajiban PSE	Pengadaan cloud/SaaS harus memperhatikan **rezim Penyelenggara Sistem Elektronik (PSE)**, karena kepatuhan penyedia merupakan bagian dari due diligence.	Sebelum memilih penyedia, instansi perlu memastikan status kepatuhan PSE, narahubung, dan kesiapan penyedia memenuhi kewajiban yang relevan secara hukum dan operasional.
13.7 Keandalan dan tanggung jawab penyelenggara sistem elektronik	Yang dibeli dari penyedia cloud bukan hanya layanan teknis, tetapi juga **keandalan dan tanggung jawab penyelenggara sistem elektronik**.	Due diligence harus menilai SLA, arsitektur ketersediaan, model support, penanganan gangguan, backup, DR, dokumentasi keamanan, dan jalur eskalasi.
13.8 TTE, log, bukti elektronik, dan tata kelola layanan elektronik	Layanan cloud/SaaS harus mampu menghasilkan **jejak elektronik yang dapat dipercaya**: log, bukti aktivitas, dukungan pembuktian, dan bila relevan dukungan terhadap layanan elektronik yang memakai TTE.	Pengadaan tidak boleh hanya fokus pada fitur dan kapasitas. Ia juga harus memastikan log memadai, aktivitas bisa ditelusuri, data bisa diekspor, dan layanan cukup kuat untuk audit maupun pembuktian.
13.9 Checklist due diligence penyedia cloud/SaaS	Tidak semua penyedia cloud/SaaS setara. Karena itu perlu **checklist due diligence** yang disiplin untuk menilai kecocokan layanan dengan kebutuhan organisasi.	Gunakan checklist yang mencakup model layanan, arsitektur, SLA, exit plan, portabilitas data, kepatuhan PSE, keamanan, support, dan bukti layanan agar risiko tersembunyi bisa dibaca sejak awal.

Ringkasan

Bab 13 menegaskan bahwa pengadaan cloud, pusat data, dan layanan PSE adalah keputusan tata kelola, bukan sekadar keputusan infrastruktur. Organisasi harus memahami model layanan dan model penempatan, menilai kapan cloud layak dipilih, membaca risiko *vendor lock-in*, menyiapkan *exit plan* dan portabilitas data, memeriksa kepatuhan PSE, menilai keandalan penyelenggara sistem elektronik, dan memastikan log serta bukti elektronik mendukung audit dan pembuktian. Dengan pendekatan seperti ini, cloud dapat menjadi alat percepatan yang sehat. Tanpa itu, cloud mudah berubah menjadi sumber ketergantungan, biaya tersembunyi, dan risiko kepatuhan.

Bab 14. Pengadaan *eLearning* / LMS / Pendidikan Digital

14.1 Ciri khusus pengadaan LMS

Pengadaan LMS tidak bisa diperlakukan seperti pengadaan aplikasi administrasi biasa. LMS bukan sekadar perangkat lunak untuk mengunggah materi. Ia adalah **ekosistem layanan belajar digital** yang menyentuh proses pembelajaran, interaksi pengajar-peserta, penilaian, dokumentasi aktivitas belajar, analitik, dan sering kali juga integrasi dengan sistem akademik atau kepegawaian. Karena itu, pengadaan LMS harus dimulai dari pemahaman atas **proses belajar**, bukan sekadar daftar fitur aplikasi.

Ada beberapa ciri khas LMS yang membuatnya berbeda.

Pertama, LMS sangat padat interaksi. Berbeda dengan aplikasi administrasi yang dipakai oleh sedikit operator inti, LMS digunakan oleh banyak kelompok pengguna dengan perilaku yang sangat beragam, seperti:

- dosen
- guru
- trainer
- mahasiswa
- siswa
- pegawai
- admin akademik
- tim dukungan

Akibatnya, keberhasilan LMS tidak cukup diukur dari “aplikasi berjalan”, tetapi dari apakah pengguna benar-benar bisa:

- masuk ke sistem
- mengakses materi
- mengerjakan tugas
- mengikuti kuis
- menerima umpan balik

tanpa friksi yang berlebihan.

Kedua, LMS sangat bergantung pada konten. Sistem ini harus mampu menangani:

- dokumen
- video
- kuis
- ujian
- forum
- bank soal
- rekaman kehadiran
- analitik progres belajar

Karena itu, ruang lingkup pengadaan LMS jauh lebih kompleks daripada sekadar pengadaan aplikasi web. Kalau ruang lingkupnya ditulis terlalu dangkal, hasil akhirnya sering hanya berupa platform yang hidup, tetapi tidak sungguh mendukung pembelajaran.

14.2 Integrasi dengan sistem akademik

Salah satu titik paling kritis dalam pengadaan LMS adalah **integrasi dengan sistem akademik, sistem kepegawaian, atau sistem administrasi pelatihan**. Tanpa integrasi yang sehat, LMS akan menjadi pulau tersendiri: akun dibuat manual, kelas dimasukkan ulang, nilai diekspor-impor lewat spreadsheet, dan data peserta tidak pernah benar-benar sinkron.

Dalam jangka pendek, ini merepotkan. Dalam jangka panjang, ini menciptakan:

- data ganda
- beban admin berulang
- konflik operasional
- ketidakpastian sumber data utama

Dalam kerangka **SPBE**, integrasi layanan dibangun di atas arsitektur proses bisnis, data, aplikasi, infrastruktur, dan keamanan. Karena itu, LMS di instansi pemerintah, kampus negeri, atau lembaga pelatihan publik tidak seharusnya diposisikan sebagai aplikasi yang berdiri sendiri. Ia harus dibaca sebagai bagian dari ekosistem layanan digital yang lebih luas. **Perpres 95 Tahun 2018** memang menekankan integrasi layanan antarinstansi serta manajemen data dan layanan SPBE. (Perpres No. 95 Tahun 2018) (Peraturan BPK)

Secara praktis, pertanyaan integrasi LMS seharusnya dijawab sejak awal pengadaan:

- apakah akun ditarik dari sistem akademik
- apakah kelas dan mata kuliah dibentuk otomatis
- bagaimana sinkronisasi daftar peserta dilakukan
- apakah nilai atau status kelulusan perlu dikirim balik ke sistem utama
- apakah ada *single sign-on*
- apakah API tersedia dan terdokumentasi

LMS yang bagus tetapi tidak terintegrasi sering akhirnya dipakai lewat banyak jalan memutar yang melelahkan pengguna dan admin. (Perpres No. 95 Tahun 2018) (Peraturan BPK)

14.3 Konten, video, kuis, ujian, bank soal, analitik belajar

Banyak proyek LMS terlalu fokus pada platform, padahal nilai riilnya justru terletak pada **pengelolaan konten dan proses belajar**.

Konten perlu dipahami sebagai objek operasional. Artinya, pengadaan harus menjawab:

- siapa yang membuat konten
- siapa yang mengunggah
- siapa yang memperbarui
- siapa yang mengarsipkan

Video menambah tantangan tersendiri, terutama pada:

- kapasitas penyimpanan
- kebutuhan *bandwidth*
- pengalaman pengguna

Kuis dan ujian menuntut:

- kestabilan sistem
- pengelolaan waktu
- pengacakan soal
- keamanan akses
- kejelasan hasil

Bank soal juga tidak cukup hanya banyak. Ia perlu:

- terstruktur
- memiliki kategori
- mendukung tingkat kesulitan
- terjaga mutunya

Sementara itu, analitik belajar penting karena LMS yang baik tidak hanya menayangkan materi, tetapi juga membantu membaca:

- progres belajar
- keterlibatan peserta
- tingkat ketuntasan
- potensi masalah pembelajaran

Dalam pengadaan, semua ini harus diterjemahkan ke spesifikasi dan kontrak. Jadi, jangan cukup menulis “tersedia fitur kuis” atau “mendukung video”. Pertanyaan yang lebih berguna adalah:

- berapa banyak peserta yang mungkin mengakses secara serentak
- bagaimana kebijakan penyimpanan video
- apakah bank soal mendukung kategori dan tingkat kesulitan
- apakah analitik tersedia untuk admin dan pengajar

- apakah hasil evaluasi dapat diekspor untuk kebutuhan akademik atau audit

LMS yang hanya kuat di katalog fitur sering justru lemah dalam operasi belajar sehari-hari.

14.4 Privasi mahasiswa, dosen, dan peserta

LMS hampir selalu memproses **data pribadi**. Data yang diproses bisa meliputi:

- nama
- email
- NIM atau NIP
- nilai
- kehadiran
- aktivitas belajar
- log akses
- rekam tugas
- bahkan kadang rekaman video atau suara

Karena itu, pengadaan LMS tidak bisa dilepaskan dari pelindungan data pribadi. **UU No. 27 Tahun 2022** menegaskan bahwa pemrosesan data pribadi harus dilakukan secara terbatas dan spesifik, sah secara hukum, transparan, sesuai tujuan, serta dapat dipertanggungjawabkan. UU yang sama juga membedakan data pribadi umum dan data pribadi yang bersifat spesifik, serta mewajibkan adanya dasar pemrosesan yang sah. (UU No. 27 Tahun 2022, JDIH Komdigi – UU PDP) (Peraturan BPK)

Untuk LMS, implikasinya jelas. Sistem tidak boleh:

- mengumpulkan data “sekalian saja”
- membuka data lebih luas dari kebutuhan
- membiarkan akses tanpa kontrol yang proporsional

Secara praktis, pengadaan LMS sebaiknya menuliskan setidaknya:

- kategori data yang diproses
- tujuan pemrosesan
- peran pengendali dan prosesor jika ada vendor
- kebutuhan *logging*
- retensi nilai dan aktivitas belajar
- hak akses dosen, admin, dan mahasiswa
- SOP saat ada permintaan koreksi atau insiden data

LMS yang nyaman digunakan tetapi lemah dari sisi privasi pada akhirnya menjadi resiko besar bagi institusi. (UU No. 27 Tahun 2022) (Peraturan BPK)

14.5 Aksesibilitas, *mobile access*, dan beban *bandwidth*

Salah satu kesalahan klasik dalam pengadaan pendidikan digital adalah menganggap semua pengguna punya perangkat bagus dan koneksi cepat. Kenyataannya sering tidak demikian. Karena itu, **aksesibilitas, akses seluler, dan beban bandwidth** harus diperlakukan sebagai unsur inti, bukan tambahan.

Aksesibilitas berarti sistem dapat digunakan oleh beragam pengguna, termasuk mereka yang memiliki keterbatasan tertentu atau kondisi penggunaan yang tidak ideal. *Mobile access* penting karena dalam banyak konteks di Indonesia, ponsel justru menjadi perangkat utama untuk belajar. Sementara itu, beban *bandwidth* penting karena LMS yang terlalu berat akan sangat menyulitkan pengguna dengan koneksi terbatas.

Dalam pengadaan, pertanyaan yang sehat adalah:

- apakah LMS nyaman dipakai melalui ponsel
- apakah antarmukanya responsif
- apakah materi bisa diakses dalam format ringan
- apakah video punya opsi kualitas berbeda
- apakah notifikasi dan aktivitas inti tetap nyaman di perangkat bergerak
- berapa perkiraan beban akses pada jam puncak

LMS yang “lengkap” tetapi berat dan sulit diakses justru akan menurunkan partisipasi belajar.

14.6 *Proctoring* dan resiko etika

Ketika LMS dipakai untuk ujian, banyak institusi tergoda memakai **proctoring digital** sebagai solusi pengawasan. Namun, *proctoring* bukan sekadar fitur tambahan. Ia membawa risiko etika, privasi, dan pengalaman pengguna yang serius.

Dari sudut pandang pengadaan, pertanyaannya bukan hanya “apakah *proctoring* tersedia”, tetapi:

- apakah benar diperlukan
- apakah proporsional
- apakah dapat dipertanggungjawabkan

Beberapa model *proctoring* meminta akses ke:

- kamera
- mikrofon
- perekaman layar
- pemantauan perilaku
- analisis otomatis

Artinya, data yang diproses menjadi jauh lebih sensitif, risiko salah deteksi meningkat, dan beban etikanya bertambah. Dalam konteks **UU PDP**, ini berarti perlu dipikirkan secara serius:

- dasar pemrosesan
- tujuan
- proporsionalitas
- retensi rekaman
- hak akses
- perlindungan atas data yang terkumpul

Teknologi pengawasan yang canggih belum tentu menghasilkan evaluasi yang adil. Untuk sebagian konteks, *proctoring* mungkin memang relevan. Tetapi dalam konteks lain, desain evaluasi yang lebih baik justru bisa jauh lebih sehat daripada pengawasan digital yang terlalu invasif. (UU No. 27 Tahun 2022) (Peraturan BPK)

14.7 LMS *open source* vs *proprietary*

Pemilihan antara LMS *open source* dan LMS *proprietary* sebaiknya dibaca dengan kacamata yang sama seperti bab-bab sebelumnya: **bukan soal gratis lawan berbayar, tetapi soal struktur biaya, kontrol, interoperabilitas, dukungan, dan kapasitas internal**.

LMS *open source* biasanya memberi ruang lebih besar untuk:

- modifikasi
- kontrol konfigurasi
- dokumentasi yang lebih terbuka
- penguatan jasa lokal

Model ini sering lebih sehat bila institusi ingin:

- menghindari *lock-in*
- menumbuhkan kapasitas admin internal
- mengintegrasikan sistem secara lebih leluasa

Tetapi *open source* tetap memerlukan biaya nyata, seperti:

- implementasi
- integrasi
- *hardening*
- dokumentasi
- *support*
- pelatihan
- operasi

Sementara itu, LMS *proprietary* bisa menawarkan:

- implementasi yang lebih cepat
- antarmuka yang lebih matang
- layanan terkelola yang lebih rapi

Namun model ini menuntut perhatian lebih besar pada:

- lisensi atau *subscription*
- portabilitas data
- integrasi
- hak akses admin
- *exit plan*

Jadi, pertanyaannya bukan “mana yang lebih modern”, tetapi **mana yang paling masuk akal untuk kebutuhan, kapasitas SDM, dan strategi jangka menengah institusi**.

14.8 Model operasi dan dukungan *helpdesk*

LMS yang bagus saat demo bisa gagal total ketika semester dimulai jika model operasi dan dukungan *helpdesk*-nya lemah. Karena itu, kualitas operasi hampir sama pentingnya dengan kualitas platform.

Model operasi harus menjawab pertanyaan seperti:

- siapa admin utama
- siapa pengelola konten
- siapa yang menambah kelas atau akun
- siapa yang menangani gangguan
- bagaimana eskalasi berjalan
- kapan dukungan tersedia

Helpdesk sangat penting karena pengguna LMS biasanya sangat beragam, dan masalah yang muncul sering sederhana tetapi sangat mengganggu, misalnya:

- lupa akun
- gagal akses kelas
- gagal unggah tugas
- tautan video tidak berjalan
- kuis tidak muncul
- nilai tidak tampil

Kalau semua masalah kecil ini tidak tertangani cepat, kepercayaan terhadap sistem akan turun.

Dari perspektif kontrak dan layanan, *helpdesk* LMS sebaiknya punya parameter yang jelas, seperti:

- jam layanan
- waktu respons
- jalur eskalasi
- dokumentasi tiket
- pelaporan insiden
- dukungan saat periode puncak seperti ujian atau awal semester

Pelajaran akhirnya sederhana: dalam pengadaan LMS, kualitas operasi sering sama pentingnya dengan kualitas platform. Sistem belajar digital yang berhasil bukan hanya yang canggih, tetapi yang benar-benar **bisa dipakai, didukung, dan dipelihara terus-menerus**.

Penutup

Bab ini menegaskan bahwa pengadaan LMS dan pendidikan digital harus dibaca sebagai **pengadaan layanan belajar**, bukan sekadar pengadaan aplikasi.

Karena itu, instansi perlu memberi perhatian serius pada:

- ciri khusus LMS
- integrasi dengan sistem akademik
- tata kelola konten dan evaluasi
- privasi peserta
- aksesibilitas dan keterbatasan *bandwidth*
- risiko etika dari *proctoring*
- pilihan antara *open source* dan *proprietary*
- model operasi dan *helpdesk* yang benar-benar siap dipakai

Bila semua ini dirancang sejak awal, LMS dapat menjadi pengungkit kualitas pembelajaran. Tetapi bila diabaikan, LMS mudah berubah menjadi platform yang hidup di server, namun tidak hidup di proses belajar.

Tabel Summary. Pengadaan *eLearning* / LMS / Pendidikan Digital

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
14.1 Ciri khusus pengadaan LMS	LMS bukan sekadar aplikasi unggah materi, tetapi **ekosistem layanan belajar digital** yang menyentuh proses pembelajaran, interaksi, penilaian, analitik, dan dukungan pengguna.	Pengadaan LMS harus dimulai dari kebutuhan proses belajar, profil pengguna, dan model operasi pembelajaran, bukan sekadar dari daftar fitur aplikasi.
14.2 Integrasi dengan sistem akademik	LMS yang tidak terintegrasi dengan sistem akademik, kepegawaian, atau pelatihan akan mudah menjadi **pulau tersendiri**: akun manual, kelas ganda, nilai dipindah tangan, dan data tidak sinkron.	KAK dan spesifikasi harus menjelaskan integrasi akun, kelas, peserta, nilai, *single sign-on*, dan API agar LMS benar-benar menjadi bagian dari ekosistem digital institusi.
14.3 Konten, video, kuis, ujian, bank soal, analitik belajar	Nilai riil LMS ada pada **pengelolaan konten dan proses belajar**, bukan hanya pada platform. Video, kuis, ujian, bank soal, dan analitik harus dibaca sebagai objek operasional yang perlu ditata.	Spesifikasi LMS harus menulis kebutuhan operasional yang konkret: kapasitas akses serentak, penyimpanan konten, pengelolaan bank soal, stabilitas ujian, dan pelaporan progres belajar.
14.4 Privasi mahasiswa, dosen, dan peserta	LMS hampir selalu memproses **data pribadi** seperti identitas, nilai, kehadiran, log aktivitas, dan kadang rekaman video. Karena itu, privasi harus menjadi bagian inti pengadaan LMS.	Pengadaan harus menjelaskan kategori data, tujuan pemrosesan, kontrol akses, logging, retensi data, dan peran pengendali/prosesor bila ada vendor pengelola.
14.5 Aksesibilitas, mobile access, dan beban bandwidth	LMS yang baik harus mempertimbangkan **keragaman perangkat dan kualitas koneksi pengguna**. Platform yang berat atau tidak nyaman di ponsel akan menurunkan partisipasi belajar.	Spesifikasi perlu meminta antarmuka responsif, dukungan akses mobile, opsi konten yang lebih ringan, dan perhitungan beban akses pada jam puncak agar sistem tetap inklusif.

14.6 Proctoring dan risiko etika	*Proctoring* digital bukan sekadar fitur ujian, tetapi membawa **risiko etika, privasi, dan salah deteksi**. Karena itu, ia harus dinilai secara proporsional, bukan otomatis dianggap wajib.	Sebelum membeli proctoring, institusi harus menilai kebutuhan, dasar pemrosesan data, proporsionalitas, retensi rekaman, dan dampaknya pada keadilan serta pengalaman peserta.
14.7 LMS open source vs proprietary	Pilihan antara **LMS open source** dan **proprietary** harus dibaca dari sisi biaya, kontrol, interoperabilitas, dukungan, dan kapasitas internal, bukan sekadar gratis vs berbayar.	Open source cocok bila institusi ingin kontrol dan penguatan kapasitas lokal; proprietary bisa cocok untuk implementasi cepat, tetapi harus dibaca hati-hati dari sisi lisensi, integrasi, dan exit plan.
14.8 Model operasi dan dukungan helpdesk	LMS yang baik saat demo bisa gagal saat semester dimulai bila **operasi dan helpdesk**-nya lemah. Kualitas dukungan sering sama pentingnya dengan kualitas platform.	Kontrak dan model operasi harus menjelaskan admin utama, pengelola konten, jalur eskalasi, jam layanan, waktu respons, dokumentasi tiket, dan dukungan pada periode puncak seperti awal semester atau ujian.

Ringkasan

Bab 14 menegaskan bahwa pengadaan LMS dan pendidikan digital harus dibaca sebagai pengadaan layanan belajar, bukan sekadar pengadaan aplikasi. Karena itu, institusi harus memperhatikan integrasi dengan sistem akademik, tata kelola konten dan evaluasi, privasi peserta, aksesibilitas dan keterbatasan bandwidth, risiko etika dari proctoring, pilihan open source vs proprietary, serta model operasi dan helpdesk yang benar-benar siap dipakai. Dengan pendekatan ini, LMS dapat menjadi pengungkit kualitas pembelajaran. Tanpa itu, ia mudah berubah menjadi platform yang aktif secara teknis tetapi lemah dalam proses belajar yang nyata.

Bab 15. Pengadaan *eCommerce* / *Marketplace* / PMSE

15.1 Ciri khusus sistem *e-commerce*

Pengadaan sistem *e-commerce* tidak bisa diperlakukan seperti pengadaan aplikasi administrasi biasa. Sistem ini bukan sekadar katalog digital atau formulir pemesanan. Ia adalah **mesin transaksi, mesin kepercayaan, dan mesin bukti**. Di dalamnya ada produk, pengguna, harga, promosi, stok, pesanan, pembayaran, notifikasi, log aktivitas, komplain, pembatalan, dan sering kali integrasi dengan banyak pihak ketiga. Karena itu, *e-commerce* selalu membawa beban tata kelola yang lebih berat dibanding aplikasi internal biasa.

Ada beberapa ciri khusus yang perlu dibaca sejak awal.

Pertama, transaksi berlangsung sangat dinamis. Pengguna datang dari luar organisasi, perilakunya tidak sepenuhnya bisa diprediksi, dan volume interaksinya dapat melonjak tiba-tiba.

Kedua, sistem ini bergantung pada **bukti elektronik**. Hampir setiap tindakan penting harus bisa dibuktikan, misalnya:

- siapa memesan;
- kapan membayar;
- harga berapa;
- promo apa yang berlaku;
- siapa mengubah status transaksi;
- apa yang terjadi saat muncul sengketa.

Ketiga, eksposur risikonya lebih tinggi. Sistem seperti ini lebih dekat ke:

- penipuan;
- penyalahgunaan akun;
- sengketa transaksi;
- insiden data;

dibanding aplikasi administrasi internal.

Karena itu, pengadaan *e-commerce* harus dimulai dari pemahaman bahwa yang dibeli bukan hanya aplikasi, tetapi **ekosistem transaksi elektronik** yang harus andal, aman, dapat dibuktikan, dan dapat diaudit. Pembacaan ini sejalan dengan kerangka PMSE dan sistem elektronik dalam regulasi Indonesia. (Peraturan BPK)

15.2 Perdagangan melalui Sistem Elektronik

Kerangka dasar untuk membaca bab ini adalah **Perdagangan Melalui Sistem Elektronik (PMSE)**. **PP 80 Tahun 2019** secara resmi mengatur PMSE sebagai perdagangan yang transaksinya dilakukan melalui serangkaian perangkat dan prosedur elektronik. PP ini penting karena menempatkan transaksi digital bukan hanya sebagai urusan teknis, tetapi sebagai rezim perdagangan yang membawa kewajiban bagi pelaku usaha dan penyelenggara. (Peraturan BPK)

Bagi pengadaan IT, implikasinya sangat praktis. Ketika organisasi membangun:

- *marketplace*;
- portal penjualan;
- kanal pemesanan digital;
- platform transaksi lain;

maka pengadaannya tidak cukup dinilai dari sisi antarmuka, fitur promosi, atau integrasi pembayaran. Ia juga harus dibaca dari kacamata kepatuhan PMSE, karena sistem itu nantinya akan hidup dalam lingkungan hukum perdagangan elektronik. (Peraturan BPK)

Untuk entitas pemerintah, BLUD, atau BUMD, poin ini bahkan lebih sensitif. Ketika layanan digital menyentuh:

- penjualan barang;
- tiket layanan;
- jasa laboratorium;
- layanan kesehatan berbayar;
- distribusi hasil usaha BLUD;
- kanal transaksi BUMD;

maka organisasi harus jernih membedakan:

- mana yang murni aplikasi layanan publik;
- mana yang sudah masuk wilayah PMSE;
- mana yang memerlukan tata kelola campuran.

Kesalahan membaca posisi ini bisa membuat proyek baik secara teknis, tetapi lemah secara kepatuhan. Ini merupakan inferensi yang masuk akal dari ruang lingkup PMSE dalam PP 80 Tahun 2019 dan pengaturan lebih rinci di Permendag 31 Tahun 2023. (Peraturan BPK)

15.3 Perlindungan konsumen dan tata kelola pelaku usaha

Salah satu ciri sistem *e-commerce* yang sehat adalah kemampuannya menjaga **kepercayaan pengguna**. Karena itu, pengadaan sistem seperti ini tidak boleh mengabaikan perlindungan konsumen dan tata kelola pelaku usaha.

> **Permendag 31 Tahun 2023** memberi kerangka yang lebih rinci untuk perdagangan melalui sistem elektronik, termasuk definisi PMSE, pengaturan pelaku usaha, penyelenggara perdagangan melalui sistem elektronik, serta model-model perdagangan digital yang lebih mutakhir seperti *social commerce*. Ini penting karena pasar digital berkembang lebih cepat daripada banyak kebiasaan administrasi pengadaan. (Peraturan BPK)
>
> Dalam praktik pengadaan, perlindungan konsumen berarti beberapa hal harus tertanam ke dalam sistem:
>
> - informasi produk atau layanan yang jelas;
> - harga yang transparan;
> - syarat transaksi yang dapat dibaca;
> - status pesanan yang dapat dilacak;
> - mekanisme pengaduan;
> - notifikasi yang tertib;
> - rekam jejak transaksi yang bisa ditelusuri.
>
> Tata kelola pelaku usaha juga penting, khususnya untuk *marketplace* multi-merchant, karena platform tidak hanya mengelola pembeli, tetapi juga:
>
> - penjual;
> - produk;
> - promosi;
> - komisi;
> - disiplin operasional merchant.
>
> Kalau tata kelola pelaku usaha lemah, masalah konsumen akan cepat berubah menjadi masalah platform. Ini adalah inferensi yang kuat dari kerangka PMSE dalam PP 80 Tahun 2019 dan Permendag 31 Tahun 2023. (Peraturan BPK)

15.4 Pembayaran digital dan integrasi pihak ketiga

Salah satu bagian paling sensitif dalam sistem *e-commerce* adalah **pembayaran digital** dan **integrasi pihak ketiga**. Banyak proyek mengira bahwa pembayaran hanyalah fitur tambahan. Padahal di sinilah risiko operasional, risiko sengketa, dan risiko keamanan sering terkumpul paling besar.

Secara praktis, pembayaran digital berarti sistem harus siap berhubungan dengan pihak luar, seperti:

- *payment gateway*;
- bank;
- kanal pembayaran;
- sistem verifikasi transaksi;
- penyedia *anti fraud*;
- layanan verifikasi identitas.

Setiap integrasi semacam ini menambah kompleksitas. Karena itu, pengadaan harus menulis dengan jelas:

- jenis metode pembayaran yang didukung;
- alur status transaksi;
- mekanisme rekonsiliasi;
- log notifikasi;
- penanganan *callback* yang gagal;
- prosedur untuk pembayaran tertunda;
- siapa yang bertanggung jawab saat terjadi ketidaksesuaian antara status pembayaran dan status pesanan.

Semakin banyak pihak ketiga yang terlibat, semakin penting pula:

- kontrak;
- SLA;
- *logging*;
- pembagian tanggung jawab.

Kalau integrasi hanya diperlakukan sebagai koneksi teknis, organisasi akan kesulitan saat insiden terjadi. Pembayaran digital bukan sekadar tombol bayar; ia adalah sistem bukti, kepercayaan, dan rekonsiliasi. Relevansi ini didukung oleh kerangka PMSE dan rezim sistem elektronik di Indonesia. (Peraturan BPK)

15.5 Keamanan transaksi, log, dan *fraud control*

Pada sistem *e-commerce*, keamanan tidak cukup dibaca sebagai keamanan server atau keamanan aplikasi. Ia harus dibaca sebagai **keamanan transaksi**. Ini mencakup integritas proses pesanan, perlindungan akun, validitas status pembayaran, pencatatan aktivitas, dan pengendalian penyalahgunaan.

Di titik ini, **UU ITE** tetap sangat relevan karena memberi landasan atas:

- informasi elektronik;
- dokumen elektronik;
- transaksi elektronik;
- alat bukti elektronik.

Perubahan kedua melalui **UU 1 Tahun 2024** menegaskan bahwa rezim ITE tetap menjadi fondasi penting bagi kepastian hukum dalam ekosistem digital. Untuk pengadaan *e-commerce*, artinya sistem harus dirancang agar jejak elektroniknya cukup kuat untuk pembuktian, audit, dan penanganan sengketa. (Peraturan BPK)

Secara praktis, sistem sebaiknya memiliki *logging* yang memadai, misalnya:

- siapa login;
- siapa mengubah pesanan;
- kapan pembayaran masuk;
- siapa membatalkan;
- perubahan apa yang terjadi pada harga atau promo;
- bagaimana alur notifikasi berjalan.

Fraud control juga perlu dipikirkan sejak awal, misalnya:

- deteksi pola transaksi tidak wajar;
- pembatasan percobaan login;
- kontrol perubahan akun;
- validasi transaksi bernilai tinggi;
- pemisahan peran admin.

Kalau log lemah dan kontrol penipuan minim, organisasi akan kehilangan dua hal sekaligus: **uang dan bukti**. Kebutuhan ini konsisten dengan pengakuan alat bukti elektronik dalam rezim UU ITE. (Peraturan BPK)

15.6 Persyaratan bagi *marketplace*, portal penjualan, dan *social commerce*

Tidak semua sistem *e-commerce* punya karakter yang sama. Karena itu, pengadaan perlu membedakan antara:

- *marketplace*;
- portal penjualan;
- *social commerce*.

Marketplace biasanya mempertemukan banyak penjual dan banyak pembeli, sehingga tata kelolanya lebih kompleks, misalnya pada:

- *onboarding* merchant;
- pengelolaan katalog;
- komisi;
- kepatuhan merchant;
- pengelolaan sengketa;
- pengendalian kualitas informasi produk.

Portal penjualan cenderung lebih sederhana bila hanya dipakai oleh satu institusi atau satu kelompok penyedia yang lebih terkendali.

Social commerce menambah lapisan lain, karena aktivitas promosi, komunikasi, dan transaksi bisa bercampur dalam satu ekosistem digital.

Di sini, **Permendag 31 Tahun 2023** penting karena memberi kerangka yang lebih mutakhir terhadap PMSE dan *social commerce*. (Peraturan BPK)

Implikasinya pada pengadaan sangat nyata:

- sistem *marketplace* memerlukan spesifikasi dan kontrak yang lebih berat pada tata kelola merchant dan sengketa;
- portal penjualan lebih fokus pada katalog, pembayaran, dan layanan pelanggan;
- *social commerce* menuntut perhatian lebih pada pemisahan fungsi promosi, transaksi, kontrol pelaku usaha, dan akuntabilitas aktivitas digital.

Kalau model bisnisnya tidak jelas, sistem yang dibeli sering tidak pas dengan kenyataan operasionalnya. Ini adalah inferensi yang didukung oleh perbedaan model PMSE yang diatur dalam regulasi perdagangan elektronik. (Peraturan BPK)

15.7 KAK untuk *e-commerce* pemerintah/BLUD/BUMD

Bab ini juga harus menjawab kebutuhan praktis pembaca: bagaimana menulis **KAK untuk e-commerce pemerintah, BLUD, atau BUMD**.

KAK untuk entitas publik sebaiknya dimulai dari **tujuan layanan** dan **model transaksi**. Misalnya, apakah sistem dipakai untuk:

- penjualan produk daerah;
- tiket layanan;
- jasa laboratorium;
- layanan kesehatan berbayar;
- distribusi hasil usaha BLUD;
- kanal transaksi BUMD.

Setelah itu, KAK harus menjelaskan ruang lingkup seperti:

- katalog;
- akun pengguna;
- pengelolaan merchant bila ada;
- metode pembayaran;
- integrasi pembayaran;
- log transaksi;
- notifikasi;
- pengaduan;
- *dashboard*;
- pelaporan;
- keamanan;
- *audit trail*;
- dukungan operasional.

Hal yang sering dilupakan adalah membedakan **output teknis** dan **output tata kelola**.

Output teknis bisa berupa:

- aplikasi;
- *dashboard*;
- API;
- integrasi pembayaran.

Output tata kelola bisa berupa:

- SOP operasional;
- SOP komplain;
- SOP *refund* bila relevan;
- log audit;
- dokumentasi rekonsiliasi;
- pelatihan admin.

Untuk entitas publik, bagian tata kelola ini sering sama pentingnya dengan fitur, karena berkaitan langsung dengan akuntabilitas. KAK *e-commerce* yang hanya bicara fitur biasanya lemah saat diuji oleh audit dan operasi nyata. Pembacaan ini selaras dengan PP 80 Tahun 2019 dan Permendag 31 Tahun 2023. (Peraturan BPK)

15.8 *Checklist* kepatuhan PMSE

Bab ini layak ditutup dengan **checklist kepatuhan PMSE**, karena pada akhirnya pengadaan sistem *e-commerce* harus bisa menjawab pertanyaan sederhana: apakah platform ini hanya terlihat modern, atau juga tertib secara tata kelola.

Checklist minimum sebaiknya mencakup:

- model transaksi yang jelas;
- posisi sistem dalam kerangka PMSE terbaca;
- informasi produk atau layanan transparan;
- mekanisme pembayaran dan rekonsiliasi terdokumentasi;
- status transaksi dapat ditelusuri;
- log aktivitas memadai;
- kontrol *fraud* tersedia;
- mekanisme komplain dan penanganan sengketa ada;
- kategori pelaku usaha atau merchant dikelola;
- privasi dan pelindungan data pribadi diperhatikan;
- kontrak dengan penyedia dan pihak ketiga cukup kuat untuk audit dan transisi.

Dasar normatif utamanya dapat dibaca dari **PP 80 Tahun 2019** untuk PMSE dan **Permendag 31 Tahun 2023** untuk pengaturan yang lebih rinci dan lebih mutakhir. (Peraturan BPK)

Intinya, *checklist* PMSE membantu organisasi membedakan antara platform yang sekadar **“bisa jualan”** dan platform yang benar-benar siap dipakai secara **tertib, aman, dan akuntabel**. Dalam pengadaan IT, perbedaan ini sangat besar. (Peraturan BPK)

Penutup

Bab ini menegaskan bahwa pengadaan *e-commerce*, *marketplace*, dan PMSE harus dibaca sebagai **pengadaan sistem transaksi elektronik** yang menuntut tata kelola jauh lebih berat daripada aplikasi biasa.

Karena itu, organisasi perlu:

- memahami ciri khusus *e-commerce*;
- membaca kerangka PMSE;
- menjaga perlindungan konsumen dan tata kelola pelaku usaha;
- menata pembayaran digital dan integrasi pihak ketiga;
- menulis keamanan transaksi serta log dan *fraud control*;
- membedakan kebutuhan *marketplace*, portal penjualan, dan *social commerce*;
- menerjemahkan semuanya ke KAK dan *checklist* kepatuhan yang bisa dipakai di lapangan.

Bila semua ini disiapkan sejak awal, platform digital dapat menjadi mesin layanan dan pendapatan yang sehat. Tetapi bila diabaikan, ia mudah berubah menjadi sumber sengketa, kebocoran bukti, dan risiko kepatuhan. Kerangka ini didukung oleh **PP 80 Tahun 2019**, **Permendag 31 Tahun 2023**, dan **UU 1 Tahun 2024** sebagai perubahan kedua UU ITE. (Peraturan BPK)

Tabel Summary. Pengadaan *eCommerce* / *Marketplace* / PMSE

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
15.1 Ciri khusus sistem e-commerce	Sistem *e-commerce* bukan sekadar katalog digital, tetapi **mesin transaksi, mesin kepercayaan, dan mesin bukti**. Karena itu, ia membawa beban tata kelola yang lebih berat daripada aplikasi administrasi biasa.	Pengadaan *e-commerce* harus dimulai dari pemahaman atas alur transaksi, bukti elektronik, risiko sengketa, dan kebutuhan audit, bukan hanya dari desain antarmuka atau daftar fitur.
15.2 Perdagangan melalui sistem elektronik	Pengadaan *e-commerce* harus dibaca dalam kerangka **PMSE**, karena sistem transaksi digital hidup di lingkungan hukum perdagangan elektronik, bukan sekadar lingkungan teknis aplikasi web.	Sebelum menyusun KAK dan spesifikasi, instansi perlu memastikan apakah sistem masuk wilayah PMSE, apa implikasi kepatuhannya, dan bagaimana posisi layanan tersebut dalam model bisnis atau layanan publik yang dijalankan.
15.3 Perlindungan konsumen dan tata kelola pelaku usaha	Sistem *e-commerce* yang sehat harus menjaga **kepercayaan pengguna** melalui informasi yang jelas, status transaksi yang tertib, mekanisme pengaduan, dan tata kelola pelaku usaha yang rapi.	KAK dan spesifikasi harus menulis kebutuhan transparansi informasi, status pesanan, notifikasi, komplain, dan tata kelola merchant/penjual agar masalah konsumen tidak berubah menjadi masalah platform.
15.4 Pembayaran digital dan integrasi pihak ketiga	Pembayaran digital bukan fitur tambahan, melainkan salah satu titik paling sensitif dalam sistem *e-commerce* karena melibatkan **rekonsiliasi, status transaksi, dan tanggung jawab lintas pihak**.	Pengadaan harus menjelaskan metode pembayaran, alur status, rekonsiliasi, callback/notifikasi, penanganan mismatch, dan pembagian tanggung jawab dengan penyedia pihak ketiga seperti *payment gateway* atau bank.

15.5 Keamanan transaksi, log, dan fraud control	Pada *e-commerce*, keamanan harus dibaca sebagai **keamanan transaksi**, bukan hanya keamanan server. Log dan *fraud control* menjadi inti untuk menjaga uang, akun, dan bukti elektronik.	Sistem harus punya audit trail yang memadai, pencatatan perubahan status transaksi, kontrol login dan akun, serta mekanisme deteksi pola penyalahgunaan agar sengketa dan penipuan bisa ditangani.
15.6 Persyaratan bagi marketplace, portal penjualan, dan social commerce	Tidak semua sistem *e-commerce* sama. **Marketplace**, **portal penjualan**, dan ***social commerce*** punya karakter operasional dan risiko tata kelola yang berbeda.	Sebelum merancang sistem, organisasi harus jelas model bisnisnya agar spesifikasi, KAK, dan kontrak tidak salah sasaran terhadap pengelolaan merchant, transaksi, promosi, dan sengketa.
15.7 KAK untuk e-commerce pemerintah/BLUD/BUMD	KAK *e-commerce* untuk entitas publik harus menulis bukan hanya fitur, tetapi juga **model transaksi, output tata kelola, keamanan, rekonsiliasi, pengaduan, dan audit trail**.	Dalam KAK, tim perlu membedakan output teknis seperti aplikasi dan integrasi, dengan output tata kelola seperti SOP komplain, SOP operasional, log audit, dan pelatihan admin.
15.8 Checklist kepatuhan PMSE	Checklist PMSE membantu membedakan antara platform yang sekadar “bisa jualan” dan platform yang **siap dipakai secara tertib, aman, dan akuntabel**.	Gunakan checklist yang mencakup model transaksi, transparansi informasi, pembayaran, log, fraud control, pengaduan, tata kelola merchant, privasi, dan kekuatan kontrak dengan pihak ketiga sebelum sistem dioperasikan.

Ringkasan

Bab 15 menegaskan bahwa pengadaan e-commerce, marketplace, dan PMSE harus dibaca sebagai pengadaan sistem transaksi elektronik yang menuntut tata kelola lebih berat daripada aplikasi biasa. Karena itu, organisasi harus memahami model transaksi dan posisi PMSE-nya, menjaga perlindungan konsumen dan tata kelola pelaku usaha, menata pembayaran digital dan integrasi pihak ketiga, memperkuat keamanan transaksi, log, dan *fraud control*, lalu menerjemahkan semuanya ke dalam KAK dan checklist kepatuhan yang bisa dipakai di lapangan. Dengan pendekatan ini, platform digital dapat menjadi mesin layanan dan pendapatan yang sehat. Tanpa itu, ia mudah berubah menjadi sumber sengketa, kehilangan bukti, dan risiko kepatuhan.

Bagian V. SDM, Sertifikasi, Honor, Gaji, dan Fee

Bab 16. Menentukan Kebutuhan SDM dalam Proyek IT

16.1 Peran inti proyek IT: PM, BA, *system analyst*, *architect*, *developer*, QA, *security*, DevOps, *support*

Salah satu kesalahan paling mahal dalam pengadaan IT adalah menganggap proyek dapat berjalan hanya dengan “programmer” dan “admin”. Cara pandang ini terlalu sempit. Proyek IT yang sehat hampir selalu membutuhkan kombinasi peran yang berbeda, karena yang dibangun bukan hanya kode, tetapi juga layanan, integrasi, keamanan, operasi, dan keberlanjutan. Ini juga sejalan dengan arah **SPBE**, yang menuntut integrasi proses bisnis, data, aplikasi, infrastruktur, dan keamanan sebagai satu kesatuan layanan digital. (Peraturan BPK)

Dalam praktik, peran inti proyek IT biasanya meliputi:

- **Project Manager (PM)** untuk mengendalikan ruang lingkup, jadwal, risiko, dan koordinasi.
- **Business Analyst (BA)** untuk menerjemahkan kebutuhan bisnis dan layanan.
- **System Analyst** untuk menurunkan kebutuhan ke rancangan sistem.
- **Architect** untuk menjaga konsistensi arsitektur aplikasi, data, infrastruktur, dan keamanan.
- **Developer** untuk membangun atau menyesuaikan solusi.
- **QA** untuk menguji mutu dan konsistensi hasil.
- **Security** untuk menjaga kontrol keamanan.
- **DevOps** atau peran operasional sejenis untuk *deployment*, automasi, lingkungan, dan stabilitas.
- **Support** untuk memastikan sistem benar-benar bisa dipakai setelah *go-live*.

Kalau proyek hanya diisi oleh peran teknis yang sempit, maka yang biasanya hilang adalah jembatan antara kebutuhan layanan dan sistem yang benar-benar berjalan di lapangan. Dalam dokumen arah SPBE, peningkatan kapasitas ASN penyelenggara SPBE, standar kompetensi teknis, pelatihan, dan sertifikasi memang ditegaskan sebagai bagian dari fondasi pembangunan SPBE nasional. (Peraturan BPK)

Intinya, menentukan kebutuhan SDM proyek IT harus dimulai dari dua pertanyaan dasar:

- fungsi apa yang benar-benar harus dijalankan;
- risiko apa yang harus dikendalikan.

Bukan dari kebiasaan menyebut jabatan secara generik. (Peraturan BPK)

16.2 Peran minimum vs peran ideal

Tidak semua proyek IT membutuhkan formasi besar. Karena itu, penting membedakan antara **peran minimum** dan **peran ideal**.

Peran minimum adalah kombinasi paling sedikit yang masih memungkinkan proyek berjalan dengan sehat. Pada proyek kecil, misalnya, organisasi masih bisa bekerja dengan kombinasi seperti:

- PM;
- analis yang cukup kuat;
- pengembang;
- QA;
- *support* dasar;

selama kompleksitasnya rendah dan arsitekturnya tidak rumit.

Sementara itu, **peran ideal** adalah formasi yang lebih lengkap, di mana fungsi-fungsi seperti:

- analisis bisnis;
- analisis sistem;
- arsitektur;
- keamanan;
- DevOps;
- dukungan operasional;

dipisahkan secara lebih jelas sehingga mutu, pengendalian, dan akuntabilitas lebih terjaga.

Pembedaan ini penting agar penyusunan kebutuhan personel tetap realistis. Proyek kecil tidak harus dibebani struktur seperti proyek nasional. Tetapi proyek yang besar, integratif, dan sensitif juga tidak boleh dipaksa memakai formasi kecil hanya demi terlihat hemat. Arah umum ini selaras dengan logika SPBE yang menekankan penguatan kapasitas SDM dan kompetensi teknis yang proporsional dengan kebutuhan layanan digital. (Peraturan BPK)

Pelajaran pentingnya adalah ini: **peran minimum boleh dipakai, tetapi harus jujur terhadap risiko**. Sedangkan **peran ideal** biasanya memberi kualitas lebih baik, tetapi tetap harus disesuaikan dengan skala, kompleksitas, dan beban layanan yang dituju. (Peraturan BPK)

16.3 Kapan satu orang boleh merangkap

Dalam pengadaan IT, pertanyaan “bolehkah satu orang merangkap?” hampir selalu muncul. Jawabannya: **boleh dalam kondisi tertentu, tetapi tidak boleh sembarangan**.

Perangkapan peran masih bisa diterima bila beberapa syarat terpenuhi:

- kompleksitas proyek rendah atau menengah;
- beban kerja masih realistis;
- tidak ada benturan besar antara fungsi pengendalian dan fungsi pelaksanaan.

Misalnya, dalam proyek kecil:

- seorang analis bisa merangkap PM;
- *developer* senior bisa merangkap *system analyst* pada sistem yang sederhana.

Tetapi ada kombinasi yang sebaiknya dihindari atau dibatasi ketat, misalnya:

- QA yang sepenuhnya melebur dengan *developer* tanpa kontrol lain;
- fungsi keamanan yang hanya formalitas pada sistem berisiko tinggi.

Merangkap memang sering tampak efisien di awal, tetapi bisa mahal di belakang. Ketika terlalu banyak fungsi ditumpuk pada sedikit orang, yang biasanya dikorbankan adalah:

- dokumentasi;
- pengujian;
- *review* independen;
- dukungan pasca implementasi.

Dari sisi tata kelola, hal ini juga menciptakan ketergantungan personel yang tinggi. Kalau satu orang kunci keluar, pengetahuan proyek ikut hilang. Karena itu, merangkap boleh dipakai sebagai strategi pragmatis, tetapi bukan alasan untuk menghapus fungsi-fungsi penting. Prinsip ini sangat selaras dengan kebutuhan penguatan kapasitas, standar kompetensi, dan pengelolaan layanan digital dalam SPBE. (Peraturan BPK)

16.4 Kualifikasi pendidikan vs kompetensi nyata

Banyak dokumen pengadaan terlalu fokus pada ijazah formal dan lupa pada pertanyaan yang lebih penting: **apakah orang ini benar-benar mampu menjalankan perannya**.

Pendidikan formal tetap penting karena memberi:

- dasar berpikir;
- kedisiplinan akademik;
- kadang relevansi bidang.

Tetapi dalam proyek IT, kompetensi nyata sering jauh lebih menentukan. Seseorang bisa memiliki gelar yang baik, tetapi:

- tidak pernah menangani integrasi;
- tidak pernah mengelola *deployment*;
- tidak terbiasa menulis dokumentasi yang rapi;
- tidak pernah menghadapi pengguna riil.

Sebaliknya, ada personel dengan jalur pendidikan yang tidak sepenuhnya linier, tetapi terbukti kuat di lapangan melalui:

- proyek yang relevan;
- portofolio;
- penguasaan teknis yang jelas.

Dalam konteks SPBE, arah kebijakan nasional juga tidak berhenti pada pendidikan formal. Dokumen strategis SPBE menekankan:

- standar kompetensi teknis SPBE;
- jabatan fungsional terkait;
- pelatihan;
- sertifikasi kompetensi.

Ini menunjukkan bahwa kualitas SDM digital memang dibaca sebagai gabungan antara struktur jabatan, pembelajaran, dan kompetensi yang dapat diukur, bukan sekadar ijazah. (Peraturan BPK)

Karena itu, syarat pendidikan dalam pengadaan sebaiknya dipakai sebagai **pagar yang wajar**, bukan sebagai satu-satunya alat seleksi. Yang lebih penting adalah kesesuaian antara:

- peran;
- kompetensi nyata;
- pengalaman;
- kemampuan menghasilkan *deliverable* yang memang dibutuhkan proyek. (Peraturan BPK)

16.5 Pengalaman minimum yang masuk akal

Bab ini juga perlu menjawab pertanyaan praktis: **berapa pengalaman minimum yang masuk akal untuk tiap peran**.

Pengalaman minimum sebaiknya tidak ditulis seragam untuk semua posisi. Beberapa peran seperti:

- PM;
- *architect*;
- *security lead*;

biasanya menuntut pengalaman yang lebih kuat karena memegang keputusan yang berdampak besar pada mutu, risiko, dan arah proyek.

Sementara itu, peran seperti:

- *developer*;
- QA;
- *support*;
- peran operasional lain;

tetap butuh pengalaman, tetapi ukurannya perlu lebih proporsional dengan kompleksitas sistem.

Yang paling penting bukan sekadar jumlah tahun, tetapi **relevansi pengalaman**. Lima tahun mengerjakan aplikasi yang sama sekali tidak terkait bisa lebih lemah daripada tiga tahun menangani sistem yang sangat mirip dengan kebutuhan proyek saat ini.

Karena itu, pengalaman minimum yang sehat sebaiknya dirumuskan dengan mengaitkan:

- peran;
- kompleksitas;
- jenis sistem;
- skala layanan.

Bukan satu angka generik yang ditempel ke seluruh posisi. Pendekatan seperti ini juga lebih masuk akal dalam kerangka penguatan kompetensi teknis SPBE yang berbasis fungsi dan kebutuhan nyata. (Peraturan BPK)

16.6 Menilai portofolio, bukan hanya CV

Dalam proyek IT, CV memang penting, tetapi **portofolio sering jauh lebih jujur**. CV memberi gambaran riwayat, pendidikan, sertifikat, dan jabatan. Tetapi portofolio menunjukkan apa yang benar-benar pernah dikerjakan.

Portofolio membantu menjawab pertanyaan seperti:

- apakah seseorang pernah membangun sistem yang mirip;
- apakah pernah menangani integrasi;
- apakah pernah bekerja di lingkungan dengan banyak pengguna;
- apakah pernah menangani migrasi, keamanan, atau beban operasional nyata;
- apakah ada artefak yang bisa dinilai, seperti rancangan, dokumentasi, demo, repository, studi kasus, atau bukti hasil proyek sebelumnya.

Menilai portofolio juga membantu mengurangi dua resiko besar:

- **senioritas semu**;
- **personel kertas**.

Ada personel yang di CV terlihat sangat meyakinkan, tetapi kontribusinya pada proyek-proyek sebelumnya tidak substantif. Sebaliknya, ada yang CV-nya sederhana tetapi portofolionya kuat, relevan, dan jelas menunjukkan kemampuan *problem solving*.

Kesimpulannya sederhana: **CV memberi cerita, tetapi portofolio memberi bukti**. Dalam proyek IT, bukti hampir selalu lebih penting daripada cerita. Pendekatan ini sangat sejalan dengan kebutuhan pengadaan yang berbasis mutu hasil kerja, bukan sekadar formalitas administratif. (JDIH LKPP)

Penutup

Bab ini menegaskan bahwa menentukan kebutuhan SDM proyek IT adalah **keputusan strategis** yang akan memengaruhi mutu, kecepatan, risiko, dan keberlanjutan hasil proyek.

Karena itu, organisasi perlu:

- memahami peran inti yang benar-benar dibutuhkan;
- membedakan antara formasi minimum dan formasi ideal;
- berhati-hati saat mengizinkan perangkapan peran;
- menyeimbangkan pendidikan formal dengan kompetensi nyata;
- menetapkan pengalaman minimum yang relevan;
- menilai portofolio secara serius, bukan hanya CV.

Jika hal-hal ini dibaca dengan jujur sejak awal, pengadaan akan jauh lebih mampu menghadirkan tim yang benar-benar dapat:

- membangun;
- menguji;
- mengamankan;
- mengoperasikan sistem.

Sebaliknya, jika diabaikan, proyek akan mudah jatuh pada jebakan klasik: tim terlihat lengkap di atas kertas, tetapi lemah saat berhadapan dengan kenyataan. Arah ini selaras dengan penguatan kapasitas ASN penyelenggara SPBE, standar kompetensi teknis, pelatihan, dan sertifikasi yang ditekankan dalam dokumen strategis SPBE nasional. (Peraturan BPK)

Tabel Summary. Menentukan Kebutuhan SDM dalam Proyek IT

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
16.1 Peran inti proyek IT: PM, BA, system analyst, architect, developer, QA, security, DevOps, support	Proyek IT yang sehat tidak cukup dijalankan oleh “programmer” dan “admin” saja. Ia membutuhkan kombinasi peran yang berbeda karena yang dibangun bukan hanya kode, tetapi juga **layanan, integrasi, keamanan, operasi, dan keberlanjutan**.	Saat menyusun kebutuhan personel, instansi harus mulai dari fungsi yang harus dijalankan dan risiko yang harus dikendalikan, lalu menurunkannya ke peran inti yang relevan.
16.2 Peran minimum vs peran ideal	Tidak semua proyek memerlukan formasi besar. Karena itu, penting membedakan **peran minimum** yang masih aman dan **peran ideal** yang memberi mutu serta kontrol lebih baik.	Formasi minimum boleh dipakai untuk proyek kecil atau sederhana, tetapi proyek besar, integratif, atau sensitif tidak boleh dipaksa memakai tim yang terlalu tipis hanya demi terlihat hemat.
16.3 Kapan satu orang boleh merangkap	Perangkapan peran boleh dilakukan dalam kondisi tertentu, tetapi tidak boleh menghilangkan fungsi penting seperti kontrol mutu, keamanan, atau dukungan operasional.	Satu orang bisa merangkap bila kompleksitas proyek rendah-menengah, beban kerja masih realistis, dan tidak ada benturan serius antara fungsi pelaksana dan fungsi pengendalian.
16.4 Kualifikasi pendidikan vs kompetensi nyata	Pendidikan formal penting, tetapi dalam proyek IT **kompetensi nyata** sering lebih menentukan daripada ijazah saja. Yang dibutuhkan adalah kecocokan antara peran, kemampuan, dan bukti kerja.	Syarat pendidikan sebaiknya dijadikan pagar yang wajar, tetapi evaluasi personel harus tetap melihat kemampuan teknis, pengalaman relevan, dan hasil kerja yang pernah ditangani.
16.5 Pengalaman minimum yang masuk akal	Pengalaman minimum tidak boleh ditulis seragam untuk semua posisi. Yang paling penting bukan hanya jumlah tahun, tetapi **relevansi pengalaman** terhadap jenis sistem, skala layanan, dan kompleksitas proyek.	Saat menulis syarat pengalaman, instansi perlu mengaitkannya dengan peran, kompleksitas proyek, dan risiko. Hindari syarat terlalu rendah yang melemahkan mutu atau terlalu tinggi yang menutup pasar tanpa alasan kuat.

16.6 Menilai portofolio, bukan hanya CV	Dalam proyek IT, **CV memberi cerita, tetapi portofolio memberi bukti**. Penilaian personel yang hanya bertumpu pada CV sangat rawan tertipu senioritas semu atau personel kertas.	Evaluasi personil inti sebaiknya melihat proyek yang benar-benar pernah dikerjakan, artefak hasil kerja, pengalaman integrasi, dokumentasi, demo, studi kasus, atau bukti kontribusi nyata, bukan hanya daftar jabatan dan sertifikat.

Ringkasan

Bab 16 menegaskan bahwa menentukan kebutuhan SDM proyek IT adalah keputusan strategis yang langsung mempengaruhi mutu, kecepatan, risiko, dan keberlanjutan hasil proyek. Organisasi perlu memahami peran inti yang benar-benar dibutuhkan, membedakan formasi minimum dan ideal, berhati-hati saat mengizinkan perangkapan peran, menyeimbangkan pendidikan formal dengan kompetensi nyata, menetapkan pengalaman minimum yang relevan, dan menilai portofolio secara serius, bukan hanya CV. Dengan pendekatan seperti ini, pengadaan akan lebih mungkin menghadirkan tim yang benar-benar mampu membangun, menguji, mengamankan, dan mengoperasikan sistem secara bertanggung jawab.

Bab 17. Sertifikat, SKKNI, dan Kualitas SDM

17.1 Mengapa sertifikat harus dikaitkan dengan peran, bukan formalitas

Dalam banyak pengadaan IT, sertifikat masih sering diperlakukan seperti ornamen administratif: semakin banyak sertifikat, dianggap semakin baik. Cara pandang ini keliru. Sertifikat hanya benar-benar bernilai jika dikaitkan dengan **peran yang memang akan dijalankan** dalam proyek. Kalau tidak, ia berubah menjadi formalitas yang mahal, mempersempit pasar, tetapi tidak otomatis menaikkan mutu hasil kerja.

Contohnya, seorang **security lead** memang lebih masuk akal diminta memiliki sertifikasi yang relevan dengan keamanan informasi atau audit keamanan. Seorang **cloud engineer** lebih masuk akal diuji lewat sertifikasi atau bukti kompetensi *cloud*. Seorang **project manager** lebih tepat dinilai dari kombinasi pengalaman memimpin proyek, kemampuan koordinasi, dan sertifikasi manajemen proyek yang relevan. Tetapi meminta seluruh personel proyek memiliki sertifikat yang sama, atau meminta sertifikat yang tidak nyambung dengan fungsi nyata, justru menunjukkan bahwa persyaratan itu lahir dari kebiasaan, bukan dari kebutuhan.

Pendekatan ini juga sejalan dengan arah SPBE yang menekankan peningkatan kapasitas SDM, standar kompetensi teknis, pelatihan, dan sertifikasi kompetensi sebagai bagian dari fondasi pembangunan SPBE nasional. (Peraturan BPK)

Intinya, pengadaan yang matang tidak bertanya:

- sertifikat apa yang sedang populer;
- label apa yang paling bergengsi;

melainkan:

- kompetensi apa yang dibutuhkan untuk peran ini;
- bukti apa yang paling masuk akal untuk menilainya.

17.2 BNSP, LSP, dan SKKNI

Untuk konteks Indonesia, pembahasan sertifikasi SDM IT tidak lengkap tanpa memahami hubungan antara **BNSP**, **LSP**, dan **SKKNI**.

Secara sederhana:

- **SKKNI** memberi kerangka kompetensi kerja nasional yang bisa dipakai untuk mendefinisikan kemampuan pada bidang atau jabatan tertentu. (BNSP)
- **LSP** adalah lembaga yang menyelenggarakan sertifikasi profesi berdasarkan skema tertentu. BNSP menampilkan daftar LSP resmi, termasuk yang relevan dengan bidang informatika. (BNSP)
- **BNSP** berperan sebagai otoritas sertifikasi personel nasional dan menjalankan sistem sertifikasi kompetensi profesi di Indonesia. (BNSP)

Dalam logika pengadaan, tiga unsur ini penting karena memberi pijakan yang lebih objektif saat organisasi ingin menulis persyaratan kompetensi. Dengan pendekatan berbasis SKKNI/LSP/BNSP, syarat kompetensi bisa lebih dekat ke **peran kerja nyata**, misalnya:

- pengembangan perangkat lunak;
- administrasi sistem;
- pengelolaan jaringan;
- keamanan;
- data;
- fungsi teknis lain yang memang dibutuhkan proyek.

Namun, bab ini juga perlu jujur: keberadaan sertifikat nasional tidak otomatis menutup kebutuhan akan pembuktian lain. Tetap perlu dibaca:

- pengalaman;
- portofolio;
- relevansi peran.

Sertifikat nasional memberi struktur. Kualitas nyata tetap harus dibuktikan lewat pekerjaan.

17.3 Sertifikasi untuk keamanan, *cloud*, data, audit, dan pengembangan

Tidak semua domain IT membutuhkan jenis sertifikasi yang sama. Karena itu, pengadaan harus membedakan kebutuhan sertifikasi menurut bidang kompetensinya.

Untuk memudahkan, kebutuhan sertifikasi bisa dibaca per rumpun:

- **Keamanan**: relevan untuk peran yang menangani kontrol keamanan, audit keamanan, respons insiden, atau *hardening*.
- **Cloud**: relevan ketika proyek memang bertumpu pada layanan *cloud* tertentu, arsitektur *cloud*, atau operasi *cloud*.
- **Data**: relevan untuk peran yang menangani rekayasa data, basis data, analitik, atau tata kelola data.
- **Audit**: penting bila proyek membutuhkan penilaian kepatuhan, audit keamanan, atau evaluasi tata kelola.
- **Pengembangan**: sertifikasi kadang bermanfaat, tetapi untuk banyak peran pengembangan, portofolio, kualitas artefak, dan pengalaman proyek sejenis sering lebih jujur daripada sekadar sertifikat pelatihan pemrograman.

Pelajaran strategisnya: sertifikasi paling berguna pada domain yang memang membutuhkan disiplin, standar, dan kontrol yang jelas. Tetapi untuk sebagian peran teknis, sertifikasi sebaiknya dibaca sebagai **pelengkap pembuktian**, bukan pengganti bukti kerja nyata. Pendekatan ini selaras dengan ekosistem BNSP–LSP yang berbasis skema kompetensi, bukan sekadar label umum. (BNSP)

17.4 Cara menulis syarat sertifikasi yang adil dan relevan

Ini salah satu bagian paling praktis. Dalam pengadaan IT, syarat sertifikasi yang salah tulis bisa menjadi:

- diskriminatif;
- tidak relevan;
- terlalu sempit;
- atau justru menurunkan kualitas persaingan.

Syarat sertifikasi yang sehat sebaiknya memenuhi beberapa prinsip:

- terkait langsung dengan peran;
- proporsional dengan kompleksitas proyek;
- tidak menjadikan satu sertifikat tertentu sebagai satu-satunya jalan, jika kompetensi yang dibutuhkan sebenarnya bisa dibuktikan lewat jalur setara;
- dibaca bersama dengan pengalaman dan portofolio.

Misalnya, untuk peran keamanan, organisasi dapat menulis bahwa personil inti harus memiliki **sertifikasi kompetensi yang relevan dengan keamanan informasi atau audit keamanan**, dan/atau bukti pengalaman kuat pada proyek sejenis, sesuai model evaluasi yang dipakai. Untuk peran *cloud*, organisasi dapat mensyaratkan sertifikasi atau pembuktian kompetensi teknis yang relevan dengan arsitektur dan operasi *cloud* yang memang dipakai proyek.

Dengan cara ini, pengadaan tetap menjaga mutu tanpa mengunci pasar pada satu label. Pendekatan seperti ini juga lebih selaras dengan fungsi BNSP, LSP, dan SKKNI yang menekankan kompetensi kerja, bukan sekadar nama sertifikat. (BNSP)

Kesalahan umum yang sebaiknya dihindari adalah:

- meminta terlalu banyak sertifikat untuk satu peran;
- meminta sertifikat yang tidak nyambung dengan pekerjaan;
- memakai sertifikat sebagai pengganti seluruh proses evaluasi kualitas personel.

Sertifikat membantu seleksi. Ia tidak boleh mematikan akal sehat seleksi.

17.5 Sertifikasi nasional vs internasional

Dalam praktik pengadaan, sering muncul pertanyaan: mana yang lebih penting, **sertifikasi nasional** atau **internasional**?

Jawaban yang sehat bukan memilih salah satunya secara ideologis, melainkan membaca **fungsi dan konteks proyek**.

Sertifikasi nasional yang bertumpu pada SKKNI/LSP/BNSP punya kelebihan pada:

- kesesuaian dengan ekosistem kompetensi kerja Indonesia;
- kemudahan dikaitkan dengan jabatan dan skema profesi lokal.

Sertifikasi internasional sering kuat pada:

- teknologi vendor tertentu;
- praktik global;
- kerangka kompetensi yang dipakai luas di industri.

Untuk proyek tertentu, keduanya bisa sama-sama relevan. Karena itu, pendekatan yang sehat adalah:

- bila proyek dekat dengan standar kompetensi kerja nasional dan kebutuhan peran umum, sertifikasi nasional bisa sangat memadai;
- bila proyek menyangkut teknologi tertentu, ekosistem *cloud* tertentu, atau metode audit yang memang bertumpu pada kerangka global, sertifikasi internasional bisa relevan.

Yang penting adalah **relevansi**, bukan gengsi. Karena itu, syarat pengadaan sebaiknya memberi ruang yang masuk akal: menerima sertifikasi nasional yang relevan, sertifikasi internasional yang setara relevansinya, atau kombinasi keduanya bila memang dibutuhkan. Ini lebih adil dan lebih sehat bagi pasar. (BNSP)

17.6 Matriks jabatan–kompetensi–sertifikat–pengalaman

Bab ini menjadi jauh lebih kuat bila pembaca diajak melihat hubungan antarunsur secara sistematis. Karena itu, pendekatan **matriks jabatan–kompetensi–sertifikat–pengalaman** sangat berguna.

Matriks ini membantu organisasi berhenti dari kebiasaan menulis syarat personel secara acak. Dengan matriks, organisasi bisa menulis syarat berdasarkan logika yang lebih sehat:

- untuk **project manager**, kompetensi utamanya bukan pemrograman, tetapi manajemen ruang lingkup, koordinasi, risiko, komunikasi, dan pengendalian *delivery*;
- untuk **security lead**, kompetensi utamanya ada pada kontrol keamanan, identifikasi risiko, audit, atau *incident response*;
- untuk **data engineer**, fokusnya pada rekayasa data, kualitas data, *pipeline*, dan stabilitas operasional;
- untuk **developer**, bobot terbesar bisa jadi ada pada portofolio dan pengalaman membangun sistem serupa, sementara sertifikat bersifat pendukung.

Dengan matriks seperti ini, tim pengadaan bisa menulis syarat yang lebih presisi. Bukan lagi “semua personel harus bersertifikat”, tetapi:

- untuk jabatan A, kompetensi utamanya apa;
- bukti sertifikasinya apa bila relevan;
- pengalaman minimumnya apa.

Pendekatan seperti ini membuat persyaratan:

- lebih mudah dipertanggungjawabkan;
- lebih adil;
- lebih kecil resikonya menjadi formalitas kosong.

Ringkasnya, matriks ini adalah cara paling sehat untuk menghubungkan kebutuhan proyek dengan kualitas personel secara rasional.

17.7 Model evaluasi kualitas personel inti

Bagian terakhir ini sangat penting, karena sertifikat, pengalaman, dan portofolio pada akhirnya harus masuk ke **model evaluasi kualitas personel inti**.

Model evaluasi yang sehat sebaiknya tidak bertumpu pada satu faktor tunggal. Evaluasi yang baik biasanya merupakan kombinasi dari beberapa unsur:

- **Sertifikat**, sebagai indikator formal kompetensi tertentu.
- **Pengalaman**, untuk menunjukkan relevansi jam terbang.
- **Portofolio**, untuk menunjukkan bukti hasil kerja nyata.
- **Wawancara teknis atau presentasi singkat**, bila metode pengadaan dan ruang evaluasi memungkinkan, untuk menguji apakah personel benar-benar memahami konteks proyek atau hanya muncul di CV.

Pendekatan seperti ini lebih sehat karena mengurangi dua risiko utama:

- **personel kertas**;
- **senioritas semu**.

Yang dicari bukan personel yang tampak paling lengkap di CV, tetapi personel yang paling masuk akal untuk menyelesaikan proyek. Ini juga sejalan dengan kebutuhan penguatan kompetensi teknis dan kualitas SDM digital yang ditekankan dalam kerangka SPBE. (Peraturan BPK)

Penutup

Bab ini menegaskan bahwa **sertifikat, SKKNI, dan kualitas SDM harus dibaca sebagai bagian dari strategi mutu proyek, bukan formalitas administrasi**.

Karena itu, organisasi perlu:

- mengaitkan sertifikat dengan peran;
- memahami fungsi BNSP–LSP–SKKNI;
- membedakan kebutuhan sertifikasi menurut domain kompetensi;
- menulis syarat yang adil dan relevan;
- membaca secara sehat perbandingan sertifikat nasional dan internasional;
- menerjemahkannya ke dalam matriks jabatan–kompetensi–sertifikat–pengalaman;
- membangun model evaluasi personil inti yang seimbang.

Dengan pendekatan seperti ini, pengadaan akan jauh lebih mampu memilih personel yang benar-benar bisa bekerja, bukan sekadar tampak meyakinkan di atas kertas. (BNSP)

Tabel Summary. Sertifikat, SKKNI, dan Kualitas SDM

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
17.1 Mengapa sertifikat harus dikaitkan dengan peran, bukan formalitas	Sertifikat hanya bernilai jika dikaitkan dengan **peran yang benar-benar dijalankan** dalam proyek. Tanpa kaitan itu, sertifikat mudah berubah menjadi formalitas administratif yang mempersempit pasar tanpa menambah mutu.	Saat menulis syarat personel, instansi harus mulai dari fungsi dan risiko proyek, lalu menentukan kompetensi dan bukti sertifikasi yang memang relevan untuk peran tersebut.
17.2 BNSP, LSP, dan SKKNI	Dalam konteks Indonesia, pembahasan sertifikasi SDM IT harus memahami hubungan **BNSP, LSP, dan SKKNI** sebagai fondasi sertifikasi profesi dan kompetensi kerja nasional.	Gunakan skema berbasis SKKNI/LSP/BNSP untuk merumuskan kebutuhan kompetensi yang lebih objektif dan dekat dengan fungsi kerja nyata, bukan sekadar menyalin syarat sertifikat dari proyek lain.
17.3 Sertifikasi untuk keamanan, cloud, data, audit, dan pengembangan	Tidak semua domain IT membutuhkan tipe sertifikasi yang sama. **Keamanan, cloud, data, audit, dan pengembangan** punya kebutuhan pembuktian kompetensi yang berbeda.	Sertifikasi harus dipilih sesuai domain proyek. Untuk beberapa peran, sertifikat sangat penting; untuk peran lain, portofolio dan pengalaman bisa lebih menentukan daripada label sertifikat.
17.4 Cara menulis syarat sertifikasi yang adil dan relevan	Syarat sertifikasi yang baik harus **terkait dengan peran, proporsional dengan kompleksitas proyek, dan tidak diskriminatif**. Sertifikasi tidak boleh dijadikan satu-satunya bukti mutu bila kompetensi dapat dibuktikan dengan cara setara.	Tulis syarat sertifikasi dengan bahasa yang fokus pada kompetensi relevan, bukan menutup pasar pada satu label tertentu tanpa alasan teknis yang kuat.

17.5 Sertifikasi nasional vs internasional	Sertifikasi nasional dan internasional sama-sama bisa relevan. Yang penting bukan gengsi labelnya, tetapi **kecocokan dengan konteks proyek, teknologi, dan fungsi kerja**.	Dalam pengadaan, sebaiknya beri ruang yang masuk akal untuk sertifikasi nasional yang relevan, sertifikasi internasional yang setara, atau kombinasi keduanya bila memang diperlukan.
17.6 Matriks jabatan–kompetensi–sertifikat–pengalaman	Pengadaan akan lebih sehat bila kebutuhan personel dibangun dengan **matriks yang menghubungkan jabatan, kompetensi inti, sertifikasi yang relevan, dan pengalaman minimum**.	Gunakan matriks ini agar syarat personel lebih presisi, mudah dipertanggungjawabkan, dan tidak jatuh menjadi daftar sertifikat acak yang tidak nyambung dengan proyek.
17.7 Model evaluasi kualitas personel inti	Kualitas personel inti sebaiknya dinilai dari kombinasi **sertifikat, pengalaman, portofolio, dan bukti kontribusi nyata**, bukan dari administrasi semata.	Model evaluasi yang sehat perlu memberi bobot seimbang pada kompetensi formal, pengalaman relevan, kualitas portofolio, dan bila memungkinkan pembuktian langsung seperti presentasi atau wawancara teknis untuk personil kunci.

Ringkasan

Bab 17 menegaskan bahwa sertifikat, SKKNI, dan kualitas SDM harus dibaca sebagai bagian dari strategi mutu proyek, bukan formalitas administrasi. Organisasi perlu mengaitkan sertifikasi dengan peran yang nyata, memahami fondasi BNSP–LSP–SKKNI, membedakan kebutuhan sertifikasi menurut domain kompetensi, menulis syarat yang adil dan relevan, lalu menerjemahkannya ke dalam matriks jabatan–kompetensi–sertifikat–pengalaman dan model evaluasi personil inti yang seimbang. Dengan pendekatan ini, pengadaan akan lebih mampu memilih personel yang benar-benar bisa bekerja, bukan sekadar tampak kuat di atas kertas.

Bab 18. Honor, Gaji, Fee, dan Biaya SDM

18.1 Membedakan gaji, honorarium, fee, tarif harian, tarif bulanan, dan biaya tenaga ahli

Salah satu sumber kekacauan paling umum dalam pengadaan jasa IT adalah mencampuradukkan istilah **gaji, honorarium, fee, tarif harian, tarif bulanan, dan biaya tenaga ahli**. Padahal masing-masing punya logika yang berbeda. Jika istilahnya kabur, biasanya yang ikut kacau adalah HPS, ekspektasi kontrak, dan penilaian kewajaran biaya.

Secara sederhana:

- **Gaji** biasanya terkait hubungan kerja yang lebih tetap dan masuk ke rezim pengupahan ketenagakerjaan.
- **Honorarium** lebih dekat ke imbalan atas kegiatan, penugasan, atau kontribusi tertentu yang tidak identik dengan penggajian reguler.
- **Fee** lazim dipakai untuk jasa profesional atau imbalan atas layanan tertentu.
- **Tarif harian** dan **tarif bulanan** adalah cara menghitung biaya jasa berdasarkan satuan waktu.
- **Biaya tenaga ahli** dalam proyek IT seharusnya dibaca sebagai biaya menghadirkan kompetensi setiap peran untuk menghasilkan *deliverable* tertentu, bukan sekadar membayar kehadiran orang.

Karena itu, pembeli harus disiplin sejak awal. Jika proyek berbicara tentang kebutuhan personil inti vendor, jangan otomatis menyebut semuanya “gaji”. Jika pekerjaan berbicara tentang narasumber, reviewer, asesor, atau pelatih, jangan otomatis menyebut semuanya “fee”. Yang lebih penting adalah membaca:

- bentuk hubungan;
- satuan kerja;
- output yang diharapkan;
- dasar penganggarannya.

Kejelasan ini akan menentukan cara menghitung, perlakuan pajak, dan kewajaran pembanding biaya.

18.2 Struktur dan skala upah

Dalam konteks ketenagakerjaan Indonesia, **struktur dan skala upah** adalah konsep penting yang tidak boleh diabaikan saat organisasi membahas biaya SDM. **Permenaker Nomor 1 Tahun 2017** memang secara khusus mengatur struktur dan skala upah, dan aturan ini tetap menjadi salah satu rujukan utama dalam pembahasan pengupahan. (Permenaker No. 1 Tahun 2017) (peraturan.bpk.go.id)

Struktur dan skala upah penting karena membantu organisasi membaca hubungan antara:

- jabatan;
- masa kerja;
- kompetensi;
- tingkat imbalan;

secara lebih tertib. Dalam proyek IT, ini relevan terutama ketika organisasi ingin menyusun pembanding kewajaran biaya personel, *rate card*, atau menilai apakah usulan biaya dari penyedia terlalu rendah atau terlalu tinggi dibanding level peran yang ditawarkan. Struktur yang sehat membantu membedakan secara lebih rasional antara:

- junior;
- madya;
- senior;
- lead;
- peran manajerial. (Permenaker No. 1 Tahun 2017) (peraturan.bpk.go.id)

Namun, bab ini juga perlu jujur: struktur dan skala upah ketenagakerjaan tidak otomatis identik dengan *rate card* proyek. Dalam proyek IT, biaya personel sering mencakup lebih dari sekadar upah dasar. Biasanya ada juga:

- *overhead*;
- alat kerja;
- biaya koordinasi;
- jaminan keberlanjutan;
- risiko *delivery*.

Karena itu, struktur dan skala upah penting sebagai fondasi pembacaan, tetapi tidak cukup sebagai satu-satunya alat menghitung biaya proyek. (Permenaker No. 1 Tahun 2017, PP No. 36 Tahun 2021) (peraturan.bpk.go.id)

18.3 Upah minimum vs *benchmark* pasar

Kesalahan yang sangat sering terjadi adalah memakai **upah minimum** sebagai patokan untuk menilai kewajaran biaya SDM IT. Ini berbahaya. Upah minimum adalah jaring pengaman dasar dalam rezim pengupahan, bukan harga pasar untuk kompetensi digital yang spesifik. **PP 36 Tahun 2021 tentang Pengupahan** dan **PP 49 Tahun 2025** yang mengubahnya kembali menunjukkan bahwa pengupahan memang diatur secara normatif oleh negara, termasuk dalam kerangka kebijakan pengupahan dan struktur skala upah. (PP No. 36 Tahun 2021, PP No. 49 Tahun 2025) (peraturan.bpk.go.id)

Untuk proyek IT, memakai upah minimum sebagai patokan langsung akan sangat menyesatkan. Peran seperti:

- *architect*;
- *security lead*;
- *data engineer*;
- DevOps;
- PM proyek digital kompleks;

bekerja pada pasar kompetensi yang berbeda dari pekerjaan umum. Karena itu, yang lebih sehat adalah membedakan antara:

- batas dasar normatif;
- *benchmark* pasar nyata.

Upah minimum memberi dasar legal paling bawah. *Benchmark* pasar memberi gambaran biaya yang wajar untuk menghadirkan kompetensi yang benar-benar dibutuhkan proyek.

Pelajaran strategisnya begini: upah minimum penting untuk memastikan proyek tidak jatuh ke praktik biaya yang tidak masuk akal secara sosial dan legal. Tetapi untuk menyusun HPS dan *rate card* SDM IT, pembeli tetap harus melihat:

- pasar;
- senioritas;
- kompleksitas;
- risiko *delivery*.

Kalau tidak, proyek akan tampak hemat, tetapi personilnya tidak akan selevel dengan yang dibutuhkan. Dasar normatifnya datang dari regulasi pengupahan, sedangkan pembacaan pasar tetap perlu dibangun secara terpisah dan terdokumentasi. (PP No. 36 Tahun 2021, PP No. 49 Tahun 2025) (peraturan.bpk.go.id)

18.4 SBM/SBK untuk instansi pemerintah

Bagi instansi pemerintah, pembahasan biaya SDM tidak bisa dilepaskan dari **SBM** dan **SBK**. Untuk tahun anggaran 2026, **PMK Nomor 32 Tahun 2025** menetapkan **Standar Biaya Masukan Tahun Anggaran 2026**, yaitu satuan biaya berupa harga satuan, tarif, dan indeks sebagai acuan penyusunan RKA TA 2026. Sementara itu, **PMK Nomor 3 Tahun 2026** mengatur **Standar Biaya Keluaran dalam Penganggaran Kementerian Negara/Lembaga**. Keduanya menjadi jangkar formal dalam pembahasan kewajaran biaya di lingkungan pemerintah. (PMK 32 Tahun 2025, PMK 3 Tahun 2026) (jdih.kemenkeu.go.id)

Namun, pembaca juga perlu sadar bahwa pada proyek IT, tidak semua komponen jasa digital dapat dibaca memadai hanya dari SBM/SBK. Untuk personil inti vendor, *rate card* teknis, biaya integrasi, atau peran digital yang sangat spesifik, organisasi sering tetap perlu *benchmark* pasar yang terdokumentasi dengan baik.

Jadi, pendekatan yang sehat bukan memilih salah satu, melainkan menggabungkan:

- **SBM/SBK** sebagai jangkar formal;
- **pembanding pasar** sebagai alat membaca realitas teknis.

Dengan cara ini, biaya SDM bisa tetap akuntabel secara administrasi, tetapi juga realistis terhadap kebutuhan proyek. (PMK 32 Tahun 2025, PMK 3 Tahun 2026) (jdih.kemenkeu.go.id)

18.5 *Benchmark* pasar IT untuk perhitungan HPS

Karena pasar tenaga digital bergerak cepat, ***benchmark* pasar IT** menjadi sangat penting dalam penyusunan HPS. *Benchmark* membantu organisasi menjawab pertanyaan yang tidak bisa dijawab cukup dari aturan umum, misalnya:

- berapa kisaran biaya wajar untuk *developer* backend menengah;
- berapa untuk *security engineer* senior;
- berapa untuk *cloud architect*;
- berapa untuk *data analyst*;
- berapa untuk QA automation;
- berapa untuk PM aplikasi publik berskala besar.

Tanpa *benchmark*, organisasi mudah terjebak pada dua kesalahan:

- terlalu murah hingga hanya menarik personel yang tidak memadai;
- terlalu mahal tanpa dasar yang kuat.

Tetapi *benchmark* juga harus dipakai dengan disiplin. Ia sebaiknya mempertimbangkan:

- jenis peran;
- senioritas;
- kompleksitas sistem;
- lokasi atau ekosistem pasar kerja;
- model kerja penuh waktu atau paruh waktu;
- apakah peran itu terikat target *delivery* tertentu.

Benchmark bukan sekadar mengambil angka dari internet atau cerita rekan. Ia harus:

- terdokumentasi;
- bisa dijelaskan;
- konsisten dengan kebutuhan proyek.

SBM/SBK tetap menjadi jangkar resmi, tetapi *benchmark* pasar diperlukan untuk membaca kenyataan tenaga digital yang bergerak lebih cepat daripada tabel administratif. (PMK 32 Tahun 2025, PMK 3 Tahun 2026) (jdih.kemenkeu.go.id)

18.6 Menyusun *rate card* per peran

Salah satu alat paling penting dalam bab ini adalah ***rate card* per peran**. *Rate card* adalah jembatan antara kebutuhan SDM dan HPS yang bisa dihitung secara lebih masuk akal.

Rate card yang sehat tidak ditulis gelondongan. Ia sebaiknya dibagi per peran, misalnya:

- PM;
- BA;
- *system analyst*;
- *architect*;
- *developer*;
- QA;
- *security*;
- DevOps;
- *support*;
- trainer;
- peran lain yang memang relevan.

Setelah itu, tiap peran juga perlu dibedakan menurut:

- level senioritas;
- atau setidaknya tingkat kompleksitas kompetensinya.

Pendekatan ini membantu menghindari jebakan klasik:

- semua orang dihitung seolah setara;
- semua orang dihitung mahal padahal kontribusinya berbeda.

Dengan *rate card*, pembeli bisa melihat biaya per peran, durasi keterlibatan, dan nilai kontribusi dengan lebih jernih. Ini sangat membantu saat:

- menguji penawaran vendor;
- menjelaskan HPS kepada auditor;
- menjelaskan HPS kepada pimpinan.

Struktur ini juga lebih selaras dengan logika SBK dan standar biaya yang membedakan keluaran serta komponen biaya secara lebih sistematis. (PMK 3 Tahun 2026, PMK 32 Tahun 2025) (jdih.kemenkeu.go.id)

18.7 Produktivitas, level senioritas, dan kompleksitas

Biaya SDM IT tidak bisa dihitung hanya dari jabatan. Ia juga dipengaruhi oleh:

- produktivitas;
- level senioritas;
- kompleksitas pekerjaan.

Orang yang lebih senior memang biasanya lebih mahal. Tetapi senioritas bukan sekadar label tahun pengalaman. Ia berkaitan dengan kemampuan untuk:

- mengambil keputusan;
- memecahkan masalah yang ambigu;
- bekerja di sistem yang kompleks;
- menurunkan risiko proyek.

Di sisi lain, tidak semua pekerjaan memerlukan orang paling senior. Beberapa pekerjaan rutin bisa ditangani oleh level menengah atau junior yang diawasi dengan baik. Jadi, biaya SDM yang sehat selalu bertanya:

- bagian mana yang benar-benar memerlukan senioritas tinggi;
- bagian mana yang bisa diisi level lain tanpa menurunkan mutu secara signifikan.

Produktivitas juga sangat penting. Dua orang dengan tarif sama bisa menghasilkan kualitas dan kecepatan kerja yang sangat berbeda. Karena itu, biaya SDM yang tampak murah bisa sangat mahal bila:

- produktivitasnya rendah;
- penempatan senioritasnya salah;
- kompleksitas pekerjaannya tidak dibaca dengan jujur.

Dalam konteks HPS dan evaluasi penawaran, organisasi sebaiknya membaca biaya SDM bukan sekadar dari gelar atau jabatan, tetapi dari hubungan antara **level orang, beban kerja, dan risiko delivery**.

18.8 Pajak atas penghasilan jasa/honorarium

Bab ini juga perlu menyentuh **pajak atas penghasilan jasa atau honorarium**, karena dalam praktik pengadaan, angka bruto dan neto sering tercampur.

Untuk pembaca buku ini, pelajaran utamanya sederhana: biaya SDM yang ditulis dalam HPS atau penawaran harus dibaca dengan jelas:

- apakah sudah termasuk pajak atau belum;
- bentuk hubungan kerjanya seperti apa;
- konsekuensi administrasinya apa.

Tanpa kejernihan ini, organisasi bisa:

- salah membaca kewajaran biaya;
- salah menaksir nilai riil yang diterima personel atau penyedia;
- mengalami masalah saat pembayaran dan audit.

Bab ini memang tidak perlu berubah menjadi panduan pajak yang teknis. Tetapi setidaknya harus menekankan bahwa pengadaan IT perlu disiplin membedakan antara:

- nilai jasa;
- pajak yang melekat;
- konsekuensi administrasi menurut model kontrak atau jenis penghasilan.

Kekaburan pada titik ini terlihat kecil, tetapi sering sangat mengganggu di tahap pelaksanaan.

18.9 Kesalahan umum: *fee* terlalu murah, personel fiktif, atau senioritas semu

Bagian ini sangat penting karena justru di sinilah pembaca biasanya paling membutuhkan panduan yang jujur.

Kesalahan pertama adalah **fee terlalu murah**. Ini sering dipuji sebagai efisiensi, padahal biasanya hanya menunda masalah. Dampaknya bisa berupa:

- personel bagus tidak masuk;
- dokumentasi dipangkas;
- pengujian dikecilkan;
- *support* menjadi minim.

Pada akhirnya, proyek tampak jalan, tetapi rapuh.

Kesalahan kedua adalah **personel fiktif**, yaitu nama-nama kuat muncul di proposal tetapi tidak hadir sungguh-sungguh dalam *delivery*.

Kesalahan ketiga adalah **senioritas semu**. CV terlihat meyakinkan, tetapi pengalaman relevan atau kontribusi nyatanya tidak sekuat yang dibayangkan.

Intinya, biaya SDM yang sehat bukan yang paling murah di atas kertas, tetapi yang paling masuk akal untuk menghadirkan tim yang benar-benar bisa bekerja. Pendekatan ini selaras dengan kebutuhan agar HPS tidak hanya formal, tetapi mencerminkan biaya nyata proyek. (PMK 32 Tahun 2025, PMK 3 Tahun 2026) (jdih.kemenkeu.go.id)

18.10 Template perhitungan biaya SDM proyek IT

Bab ini layak ditutup dengan **template perhitungan biaya SDM proyek IT**, karena pembaca buku panduan biasanya membutuhkan alat kerja, bukan hanya konsep.

Template yang sehat setidaknya memuat:

- peran;
- level atau senioritas;
- durasi keterlibatan;
- satuan tarif, apakah harian atau bulanan;
- koefisien produktivitas atau tingkat keterlibatan;
- komponen tambahan bila ada;
- pajak;
- catatan asumsi.

Dengan format seperti ini, pembaca dapat menghitung biaya secara lebih transparan dan bisa menjelaskan logikanya saat:

- menyusun HPS;
- mengevaluasi kewajaran penawaran;
- menjawab pertanyaan auditor atau pimpinan.

Template juga membantu konsistensi antar proyek. Daripada setiap paket menyusun biaya SDM dari nol dengan logika yang berubah-ubah, organisasi bisa memakai kerangka baku yang tetap terbuka untuk penyesuaian menurut:

- jenis proyek;
- kompleksitas;
- kebutuhan riil.

Di sinilah panduan menjadi benar-benar praktis: bukan hanya menjelaskan mengapa, tetapi juga membantu menghitung bagaimana.

Penutup

Bab ini menegaskan bahwa **honor, gaji, fee, dan biaya SDM proyek IT harus dibaca dengan disiplin konsep dan disiplin angka**.

Karena itu, organisasi perlu:

- membedakan istilah dasar;
- memahami struktur dan skala upah;
- membedakan upah minimum dari *benchmark* pasar;
- memakai SBM/SBK sebagai jangkar resmi;
- menyusun *rate card* per peran;
- membaca pengaruh senioritas, produktivitas, dan kompleksitas;
- menjernihkan aspek pajak;
- menghindari jebakan klasik seperti *fee* terlalu murah, personel fiktif, dan senioritas semu.

Dengan pendekatan seperti ini, biaya SDM tidak lagi menjadi angka gelap dalam pengadaan, tetapi berubah menjadi **alat kendali mutu proyek** yang jauh lebih kuat. Dasar normatifnya bertumpu pada aturan pengupahan dan standar biaya resmi pemerintah, sementara praktik baiknya diperkuat oleh pembacaan pasar yang disiplin. (Permenaker No. 1 Tahun 2017, PP No. 36 Tahun 2021, PP No. 49 Tahun 2025, PMK 32 Tahun 2025, PMK 3 Tahun 2026) (peraturan.bpk.go.id)

Tabel Summary. Honor, Gaji, *Fee*, dan Biaya SDM

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
18.1 Membedakan gaji, honorarium, fee, tarif harian, tarif bulanan, dan biaya tenaga ahli	Dalam pengadaan IT, istilah **gaji, honorarium, *fee*, tarif harian, tarif bulanan, dan biaya tenaga ahli** tidak boleh dicampur. Masing-masing punya logika hubungan kerja, satuan biaya, dan konsekuensi administrasi yang berbeda.	Saat menyusun HPS dan kontrak, instansi harus jelas: apakah yang dibayar adalah hubungan kerja tetap, jasa profesional, penugasan tertentu, atau keterlibatan personel proyek berbasis waktu dan deliverable.
18.2 Struktur dan skala upah	**Struktur dan skala upah** membantu membaca hubungan antara jabatan, masa kerja, kompetensi, dan tingkat imbalan secara lebih tertib. Ini penting sebagai fondasi kewajaran biaya SDM.	Gunakan struktur dan skala upah sebagai dasar pembacaan level personel, tetapi jangan menyamakannya begitu saja dengan *rate card* proyek karena proyek IT juga memuat overhead dan risiko delivery.
18.3 Upah minimum vs benchmark pasar	**Upah minimum** adalah jaring pengaman normatif, bukan harga pasar untuk kompetensi digital spesifik. Untuk proyek IT, kewajaran biaya harus dibaca dengan membedakan batas dasar normatif dan *benchmark* pasar nyata.	Jangan memakai upah minimum sebagai patokan langsung untuk architect, security lead, DevOps, data engineer, atau PM proyek kompleks. Gunakan *benchmark* pasar yang relevan agar tim yang dibutuhkan benar-benar bisa dihadirkan.
18.4 SBM/SBK untuk instansi pemerintah	Bagi instansi pemerintah, **SBM/SBK** tetap penting sebagai jangkar formal dalam membaca kewajaran biaya, tetapi tidak selalu cukup untuk menangkap realitas biaya jasa digital yang sangat spesifik.	Pendekatan terbaik adalah menggabungkan SBM/SBK dengan pembanding pasar yang terdokumentasi agar HPS tetap patuh secara administrasi sekaligus realistis secara teknis.

18.5 Benchmark pasar IT untuk perhitungan HPS	**Benchmark pasar IT** dibutuhkan karena pasar tenaga digital bergerak cepat dan banyak biaya SDM teknis tidak bisa dibaca memadai hanya dari acuan umum.	Saat menyusun HPS, gunakan pembanding pasar yang relevan menurut peran, senioritas, kompleksitas sistem, dan model keterlibatan, lalu dokumentasikan logikanya dengan baik.
18.6 Menyusun rate card per peran	*Rate card* yang sehat harus dibangun **per peran**, bukan angka gelondongan. Ini membuat biaya SDM lebih transparan, dapat diuji, dan lebih mudah dijelaskan.	Pisahkan biaya untuk PM, BA, system analyst, architect, developer, QA, security, DevOps, support, trainer, dan peran lain sesuai kebutuhan proyek agar evaluasi biaya lebih masuk akal.
18.7 Produktivitas, level senioritas, dan kompleksitas	Biaya SDM IT dipengaruhi bukan hanya jabatan, tetapi juga **produktivitas, senioritas, dan kompleksitas pekerjaan**. Senioritas yang tepat bisa menurunkan risiko; senioritas yang salah tempat justru memboroskan anggaran.	Saat menghitung biaya, tentukan bagian mana yang benar-benar butuh senioritas tinggi dan mana yang bisa ditangani level lain tanpa menurunkan mutu secara signifikan.
18.8 Pajak atas penghasilan jasa/honorarium	Dalam pengadaan IT, angka bruto dan neto sering tercampur. Karena itu, pembacaan **pajak atas jasa/honorarium** harus jelas agar tidak muncul kekacauan saat pembayaran dan audit.	Dalam HPS dan evaluasi penawaran, jelaskan apakah nilai yang dipakai sudah termasuk pajak atau belum, dan pastikan model hubungan kerja/jasa dibaca konsisten dengan perlakuan administrasinya.
18.9 Kesalahan umum: fee terlalu murah, personel fiktif, atau senioritas semu	Kesalahan klasik dalam biaya SDM IT adalah ***fee* terlalu murah**, **personel fiktif**, dan **senioritas semu**. Ketiganya membuat proyek tampak murah di atas kertas tetapi rapuh di delivery.	Evaluasi biaya dan personel harus menguji kewajaran tarif, kehadiran nyata personel inti, dan relevansi pengalaman, agar proyek tidak dikuasai oleh personel “kertas” atau tim yang tidak selevel dengan kebutuhan.

18.10 Template perhitungan biaya SDM proyek IT	Template biaya SDM membantu organisasi menghitung biaya secara konsisten, transparan, dan dapat dipertanggungjawabkan. Ini membuat bab ini benar-benar operasional.	Gunakan template yang memuat peran, level, durasi, satuan tarif, tingkat keterlibatan, pajak, dan asumsi agar HPS maupun evaluasi biaya vendor lebih rapi dan konsisten antar proyek.

Ringkasan

Bab 18 menegaskan bahwa honor, gaji, *fee*, dan biaya SDM proyek IT harus dibaca dengan disiplin konsep dan disiplin angka. Organisasi perlu membedakan istilah dasar, memahami struktur dan skala upah, membedakan upah minimum dari *benchmark* pasar, memakai SBM/SBK sebagai jangkar resmi, menyusun *rate card* per peran, membaca pengaruh senioritas, produktivitas, dan kompleksitas, menjernihkan aspek pajak, lalu menghindari jebakan klasik seperti *fee* terlalu murah, personel fiktif, dan senioritas semu. Dengan pendekatan seperti ini, biaya SDM tidak lagi menjadi angka gelap dalam pengadaan, tetapi berubah menjadi **alat kendali mutu proyek** yang jauh lebih kuat.

Bagian VI. Panduan Praktis per Jenis Pengadaan

Bab 19. Pengadaan Hardware, Jaringan, dan Infrastruktur

19.1 Server, *storage*, *endpoint*, UPS, *rack*, *firewall*, *switch*, AP

Pengadaan *hardware* dan infrastruktur IT sering terlihat paling mudah karena barangnya berwujud. Padahal justru di sinilah banyak kesalahan besar terjadi. Server bisa dibeli terlalu besar atau terlalu kecil. *Storage* bisa dibeli tanpa menghitung pertumbuhan data. *Firewall* bisa dipilih karena merek, bukan karena kebutuhan. *Access point* bisa dipasang banyak, tetapi desain cakupan dan kepadatan pengguna tidak pernah dihitung. Akibatnya, organisasi membeli perangkat, tetapi tidak selalu membeli layanan yang benar-benar siap jalan.

Karena itu, titik awal pengadaan *hardware* harus tetap sama seperti bab-bab sebelumnya: mulai dari **beban kerja, risiko layanan, dan konteks operasional**, bukan dari brosur produk. Untuk tiap komponen, pertanyaannya harus berbeda:

- **Server**: berapa beban aplikasi, berapa jumlah layanan, apakah perlu virtualisasi, berapa kebutuhan ketersediaan, dan bagaimana pola pertumbuhannya.
- **Storage**: berapa kapasitas efektif, seperti apa performanya, jenis data apa yang disimpan, bagaimana retensinya, bagaimana *backup* dan kebutuhan pemulihannya.
- **Endpoint**: siapa penggunanya — administrasi ringan, operator layanan, desain, analitik, atau mobilitas lapangan.
- **UPS dan rack**: berapa kebutuhan daya, bagaimana ketahanannya, seperti apa lingkungan ruang, pendinginan, dan tata letaknya.
- **Firewall, switch, dan AP**: bagaimana desain jaringannya, segmentasinya, jumlah penggunanya, pola trafiknya, dan kebutuhan keamanannya.

Dalam kerangka SPBE, pendekatan ini sangat masuk akal karena **infrastruktur SPBE** memang ditempatkan sebagai salah satu domain inti arsitektur, sejajar dengan proses bisnis, data, aplikasi, dan keamanan. Itu berarti pengadaan perangkat tidak boleh berdiri sendiri; ia harus dibaca sebagai bagian dari layanan administrasi pemerintahan dan layanan publik berbasis elektronik yang harus andal, terintegrasi, dan berkelanjutan. (Peraturan BPK)

Pelajaran pentingnya sederhana: *hardware* yang benar bukan yang paling mahal atau paling terkenal, tetapi yang **paling tepat** untuk kebutuhan layanan, pola beban, risiko, dan kemampuan pengelolaan organisasi.

19.2 Sertifikasi perangkat dan kompatibilitas

Dalam pengadaan *hardware*, istilah “sertifikasi perangkat” sering dipakai terlalu longgar. Kadang yang dimaksud adalah sertifikasi mutu, kadang sertifikasi keselamatan, kadang kompatibilitas dengan sistem tertentu, dan kadang hanya label pemasaran. Karena itu, sertifikasi perangkat dan kompatibilitas perlu dibaca dengan hati-hati.

Prinsip dasarnya sederhana: **sertifikasi hanya penting jika memang relevan dengan fungsi perangkat dan risiko layanan**. Misalnya:

- untuk perangkat jaringan dan keamanan, kompatibilitas dengan lingkungan yang ada, dukungan standar jaringan, kemampuan manajemen, dan integrasi dengan sistem pengawasan bisa lebih penting daripada sekadar logo sertifikasi;
- untuk server dan *storage*, kompatibilitas dengan sistem operasi, *hypervisor*, platform *backup*, atau perangkat lunak manajemen sering jauh lebih menentukan daripada nama produknya;
- untuk *endpoint*, kompatibilitas dengan aplikasi kerja, kebijakan keamanan, dan lingkungan dukungan organisasi harus dibaca lebih dulu.

Dari sudut praktik pengadaan, ada dua ekstrem yang harus dihindari:

- mengabaikan kompatibilitas sehingga perangkat tidak benar-benar bisa dipakai dalam ekosistem yang ada;
- menulis syarat kompatibilitas terlalu sempit sampai hanya satu merek yang mungkin lolos.

Intinya, sertifikasi dan kompatibilitas harus dibaca sebagai alat untuk menjaga **keandalan** dan **interoperabilitas**, bukan sebagai pintu belakang untuk mengunci merek tertentu.

19.3 Redundansi dan kapasitas

Salah satu tanda kedewasaan dalam pengadaan infrastruktur adalah kemampuan membedakan antara **kapasitas normal** dan **ketahanan saat gangguan**. Di sinilah isu redundansi dan kapasitas menjadi sangat penting. Banyak organisasi membeli perangkat hanya berdasarkan kebutuhan hari ini, tanpa menghitung pertumbuhan, jam puncak, kegagalan komponen, atau kebutuhan pemulihan. Hasilnya, sistem berjalan pada hari baik, tetapi rapuh pada hari sibuk atau hari buruk.

Kapasitas seharusnya dihitung dari beban nyata, misalnya:

- jumlah pengguna;
- volume transaksi;
- ukuran data;
- pola akses;
- pertumbuhan tahunan;
- kebutuhan layanan digital yang terus bertambah.

Redundansi juga harus dibaca dari tingkat kritikalitas layanan. Pertanyaannya bukan "apakah redundansi bagus", melainkan "**seberapa jauh redundansi memang dibutuhkan**." Misalnya:

- apakah perlu *dual power supply*;
- apakah perlu *link redundancy*;
- apakah perlu *cluster*;
- apakah perlu *failover*;
- apakah perlu *hot spare*;
- atau setidaknya desain pemulihan yang masuk akal.

Tidak semua sistem butuh arsitektur yang sangat mahal. Tetapi sistem yang penting juga tidak boleh dirancang seperti aplikasi uji coba. Dalam konteks SPBE, keberlangsungan layanan bukan isu sampingan. Perpres SPBE menempatkan keamanan dan keberlangsungan sebagai bagian dari manajemen SPBE yang harus direncanakan, diterapkan, dipantau, dan dievaluasi. Artinya, pengadaan infrastruktur pemerintah yang sehat harus berpikir lebih dari sekadar "hidup saat instalasi", tetapi juga "tetap andal saat dipakai sungguh-sungguh." (Peraturan BPK)

Kesimpulannya:

- kapasitas yang sehat menjaga performa;
- redundansi yang tepat menjaga ketahanan.

Keduanya harus dihitung dari kebutuhan layanan, bukan dari kebiasaan membeli.

19.4 *Support*, *warranty*, *spare part*, dan *lifecycle*

Banyak pengadaan *hardware* berhenti pada satu pertanyaan: “barangnya datang kapan?” Padahal untuk infrastruktur IT, pertanyaan yang lebih penting justru muncul setelah itu:

- siapa yang mendukung;
- berapa lama garansinya;
- bagaimana ketersediaan suku cadangnya;
- berapa panjang siklus hidup perangkatnya.

Support penting karena perangkat tidak hanya perlu dipasang, tetapi juga dipelihara, diperbarui, dipulihkan, dan kadang diganti komponennya. **Warranty** juga harus dibaca lebih dari sekadar angka tahun. Yang perlu dilihat adalah:

- *on-site* atau tidak;
- waktu respons;
- waktu penggantian;
- cakupan komponen;
- jalur eskalasi.

Spare part penting karena perangkat yang kritikal tidak boleh bergantung pada suku cadang yang sulit didapat atau lama datangnya. Sementara itu, **lifecycle** penting karena keputusan membeli hari ini akan memengaruhi biaya dan risiko beberapa tahun ke depan, misalnya:

- kapan perangkat mulai usang;
- kapan dukungan resmi berakhir;
- kapan harus diperbarui;
- apakah organisasi punya rencana transisi.

Dari sudut layanan publik dan SPBE, isu ini juga tidak sepele. Sistem elektronik pemerintah mendukung layanan administrasi dan layanan publik berbasis elektronik. Kalau perangkat gagal dan dukungan lemah, yang terganggu bukan hanya aset, tetapi juga pelayanan dan akuntabilitas. Karena itu, pengadaan infrastruktur yang matang harus menulis *support* dan *lifecycle* sebagai bagian dari desain layanan, bukan sekadar lampiran jaminan vendor. (Peraturan BPK)

Penutup

Bab ini menegaskan bahwa pengadaan *hardware*, jaringan, dan infrastruktur harus dibaca sebagai **pengadaan fondasi layanan digital**, bukan sekadar belanja perangkat. Karena itu, organisasi perlu:

- mulai dari kebutuhan beban kerja dan konteks operasional;
- menulis kebutuhan server, *storage*, *endpoint*, UPS, *rack*, *firewall*, *switch*, dan AP secara berbasis fungsi dan kapasitas;
- memastikan sertifikasi dan kompatibilitas dibaca secara sehat;
- menghitung kapasitas dan redundansi dengan jujur;
- menata *support*, *warranty*, *spare part*, dan *lifecycle* sebagai bagian dari keberlanjutan layanan.

Dengan pendekatan seperti ini, infrastruktur tidak hanya aktif saat pemasangan, tetapi juga:

- andal saat dipakai;
- lebih siap saat terganggu;
- lebih masuk akal saat diaudit;
- lebih mudah diperbaiki saat siklus hidupnya berakhir.

Tabel Summary. Pengadaan *Hardware*, Jaringan, dan Infrastruktur

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
19.1 Server, storage, endpoint, UPS, rack, firewall, switch, AP	Pengadaan perangkat dan infrastruktur IT harus dimulai dari **beban kerja, risiko layanan, dan konteks operasional**, bukan dari brosur atau merek. Perangkat yang dibeli harus dibaca sebagai fondasi layanan digital, bukan sekadar aset fisik.	Saat menyusun kebutuhan perangkat, instansi harus memetakan beban aplikasi, pola akses, pertumbuhan kapasitas, lingkungan ruang, desain jaringan, dan kebutuhan keamanan agar perangkat yang dibeli benar-benar tepat guna.
19.2 Sertifikasi perangkat dan kompatibilitas	Sertifikasi perangkat dan kompatibilitas hanya penting bila **relevan dengan fungsi, interoperabilitas, dan risiko layanan**. Keduanya tidak boleh dipakai sebagai alasan tersembunyi untuk mengunci merek.	Dalam spesifikasi, organisasi perlu menulis kebutuhan kompatibilitas yang objektif dengan lingkungan eksisting, sambil tetap menjaga ruang kompetisi yang sehat di pasar.
19.3 Redundansi dan kapasitas	Infrastruktur yang sehat harus dihitung bukan hanya untuk **beban normal**, tetapi juga untuk **ketahanan saat gangguan**. Kapasitas menjaga performa, sedangkan redundansi menjaga layanan tetap berjalan ketika sebagian komponen gagal.	Sebelum membeli, instansi perlu menghitung jumlah pengguna, volume transaksi, pertumbuhan data, jam puncak, dan kebutuhan failover agar sistem tidak hanya hidup saat dipasang, tetapi juga andal saat dipakai nyata.
19.4 Support, warranty, spare part, dan lifecycle	Pengadaan hardware tidak berhenti pada barang datang. Nilai strategisnya justru banyak ditentukan oleh **support, warranty, ketersediaan spare part, dan pengelolaan lifecycle** perangkat.	Kontrak dan spesifikasi harus mengatur waktu respons, cakupan garansi, penggantian komponen, jalur eskalasi, ketersediaan suku cadang, serta rencana pembaruan perangkat agar layanan tidak terganggu saat infrastruktur mulai menua.

Ringkasan

Bab 19 menegaskan bahwa pengadaan hardware, jaringan, dan infrastruktur harus diperlakukan sebagai pengadaan fondasi layanan digital, bukan sekadar belanja perangkat. Karena itu, organisasi harus menulis kebutuhan perangkat berdasarkan beban kerja dan konteks operasional, menjaga sertifikasi dan kompatibilitas tetap relevan dan tidak diskriminatif, menghitung kapasitas dan redundansi secara jujur, serta memperlakukan support, warranty, spare part, dan lifecycle sebagai bagian inti dari keberlanjutan layanan. Dengan pendekatan ini, infrastruktur menjadi **lebih tepat guna, lebih tahan gangguan, dan lebih mudah dipertanggungjawabkan dalam operasi maupun audit**.

Bab 20. Pengadaan *Software* dan Pengembangan Aplikasi

20.1 *Build* vs *buy* vs *adopt OSS*

Salah satu keputusan paling strategis dalam pengadaan *software* adalah memilih antara **membangun sendiri (*build*)**, **membeli solusi jadi (*buy*)**, atau **mengadopsi *open source software* (OSS)**. Keputusan ini tidak boleh diambil hanya karena tren, kenyamanan vendor, atau preferensi pribadi tim teknis. Pilihan itu harus dibaca dari kebutuhan layanan, kapasitas organisasi, risiko integrasi, biaya jangka menengah, dan tingkat kontrol yang ingin dipertahankan organisasi. Arah ini juga selaras dengan disiplin perencanaan dan pelaksanaan pengadaan dalam pedoman LKPP. (JDIH LKPP)

Pendekatan **build** masuk akal bila organisasi punya kebutuhan yang sangat khas, proses bisnisnya tidak mudah dipenuhi oleh produk generik, atau sistem tersebut memang menjadi aset strategis yang perlu dikuasai lebih dalam. Tetapi keputusan *build* juga membawa tanggung jawab yang besar, karena organisasi harus sungguh siap pada:

- analisis;
- desain;
- pengembangan;
- pengujian;
- dokumentasi;
- operasi;
- pengelolaan perubahan.

Kalau proyek memilih *build* tanpa kesiapan analisis dan pengendalian ruang lingkup, hasilnya sering menjadi mahal dan lambat. Itu merupakan inferensi praktis yang sangat masuk akal dari prinsip perencanaan pengadaan yang menuntut kebutuhan dirumuskan dengan jelas sejak awal. (JDIH LKPP)

Pendekatan **buy** cocok bila kebutuhan relatif standar, waktu implementasi harus cepat, pasar menyediakan solusi yang matang, dan biaya penyesuaian masih masuk akal. Tetapi membeli bukan berarti masalah selesai. Organisasi tetap harus menilai:

- integrasi;
- hak pakai;
- portabilitas data;
- konfigurasi;
- dukungan;
- *exit plan*.

Banyak kegagalan justru muncul ketika organisasi membeli produk yang terlihat rapi saat demo, tetapi tidak cocok dengan proses bisnis, integrasi, atau model operasi yang sesungguhnya. Ini adalah konsekuensi logis dari kewajiban menyusun kebutuhan dan spesifikasi secara tepat dalam kerangka pengadaan. (JDIH LKPP)

Sementara itu, **adopt OSS** sering menjadi pilihan strategis ketika organisasi ingin menekan dominasi lisensi, memperbesar ruang integrasi, memperkuat kontrol atas *deliverable*, dan membuka peluang bagi jasa lokal. Tetapi OSS tidak boleh dibaca sebagai “gratis”. Ia tetap memerlukan biaya untuk:

- implementasi;
- *hardening*;
- dokumentasi;
- pelatihan;
- *support*;
- kadang audit kode atau pengelolaan dependensi.

Jadi, pertanyaan yang sehat bukan “mana yang paling murah”, tetapi **mana yang paling masuk akal untuk manfaat layanan, biaya total, penguasaan teknis, dan keberlanjutan**. (JDIH LKPP)

20.2 *Web app*, *mobile app*, ERP, *middleware*, API

Tidak semua pengadaan *software* memiliki karakter yang sama. Karena itu, organisasi perlu membedakan dengan jernih antara:

- *web app*;
- *mobile app*;
- ERP;
- *middleware*;
- API.

Bentuk aplikasinya berbeda, dan karena itu spesifikasi, risiko, serta model kontraknya juga berbeda. Dalam kerangka SPBE, integrasi proses bisnis, data, aplikasi, infrastruktur, dan keamanan adalah fondasi layanan digital pemerintah yang terpadu. Artinya, klasifikasi jenis aplikasi bukan soal teknis semata, tetapi juga soal tata kelola layanan. (Peraturan BPK)

Web app umumnya menjadi pilihan utama untuk sistem yang diakses lintas perangkat melalui peramban, relatif mudah diperbarui secara terpusat, dan cocok untuk banyak layanan administrasi maupun publik. Tantangannya sering ada pada:

- performa;
- pengalaman pengguna;
- integrasi;
- kompatibilitas lingkungan kerja.

Mobile app lebih relevan bila layanan sangat bergantung pada penggunaan ponsel, akses cepat, notifikasi, fitur perangkat, atau mobilitas lapangan. Tetapi *mobile app* juga membawa konsekuensi tambahan, seperti:

- dukungan multi-platform;
- pembaruan versi;
- pengelolaan distribusi aplikasi;
- beban dukungan pengguna yang lebih tinggi.

ERP harus diperlakukan sangat hati-hati karena ia bukan hanya aplikasi, melainkan sistem yang memengaruhi banyak fungsi organisasi sekaligus. Pengadaan ERP biasanya menyentuh:

- proses bisnis inti;
- data lintas fungsi;
- hak akses yang kompleks;
- perubahan organisasi yang besar.

Karena itu, proyek ERP yang dibeli tanpa kesiapan *as-is/to-be*, pengelolaan *master data*, dan komitmen pimpinan sering menjadi sumber frustasi besar. Ini merupakan inferensi yang kuat dari prinsip arsitektur dan integrasi dalam SPBE. (Peraturan BPK)

Middleware dan **API** juga tidak boleh diremehkan. Keduanya sering tidak terlihat oleh pengguna akhir, tetapi justru sangat menentukan:

- interoperabilitas;
- stabilitas integrasi;
- ketahanan sistem.

Dalam kerangka SPBE, keduanya harus dipandang sebagai aset strategis, bukan sekadar sambungan teknis antar-aplikasi. Arsitektur SPBE memang dirancang untuk mendorong keterpaduan layanan, bukan proliferasi aplikasi silo. (Peraturan BPK)

Pelajaran pentingnya: bentuk aplikasinya boleh berbeda, tetapi pertanyaan dasarnya tetap sama:

- siapa penggunanya;
- proses apa yang ditopang;
- data apa yang diproses;
- bagaimana integrasinya;
- bagaimana sistem itu dipelihara setelah implementasi.

20.3 *Source code*, repo, CI/CD, dokumentasi

Dalam pengadaan *software*, banyak organisasi terlalu fokus pada antarmuka dan fitur, tetapi kurang tegas pada hal-hal yang justru menentukan **kepemilikan operasional** dan **keberlanjutan teknis**. Karena itu, *source code*, repository, CI/CD, dan dokumentasi harus dipandang sebagai bagian inti hasil proyek, bukan pelengkap.

Source code penting bukan hanya untuk proyek yang dibangun dari nol. Ia penting untuk memastikan bahwa organisasi memahami:

- apa yang sebenarnya diserahkan;
- apa yang bisa dipelihara;
- apa yang dapat dikembangkan lebih lanjut sesuai model hak pakai dan hak modifikasi di kontrak.

Repo penting sebagai sumber kebenaran kerja pengembangan, karena di sanalah biasanya tersimpan:

- histori perubahan;
- kolaborasi tim;
- kontrol versi;
- dasar audit teknis.

Tanpa repo yang dikelola sehat, organisasi sering mewarisi sistem yang aktif tetapi sulit dipahami dan sulit diteruskan. Prinsip ini sangat sejalan dengan kebutuhan *deliverable* yang jelas dalam perencanaan dan pelaksanaan pengadaan. (JDIH LKPP)

CI/CD penting karena kualitas *software* modern tidak hanya ditentukan oleh *coding*, tetapi juga oleh cara perubahan:

- dibangun;
- diuji;
- diterapkan ke lingkungan yang berbeda;
- dikendalikan saat terjadi kegagalan.

Organisasi tidak harus memaksa semua proyek memakai *pipeline* yang rumit. Tetapi untuk proyek yang cukup penting, setidaknya harus ada disiplin tentang:

- *build process*;
- *deployment*;
- *rollback*;
- pengendalian perubahan.

Ini membantu mutu, ketertelusuran, dan stabilitas operasional.

Yang paling sering diremehkan justru **dokumentasi**. Banyak vendor bisa membuat sistem berjalan, tetapi tidak semua meninggalkan dokumentasi yang layak, seperti:

- dokumen analisis;
- dokumen desain;
- konfigurasi;
- skema integrasi;
- petunjuk instalasi;
- SOP operasi;
- manual admin;
- manual pengguna;
- catatan perubahan.

Padahal dalam banyak proyek, justru dokumentasi inilah yang membedakan antara sistem yang dapat diwariskan dan sistem yang hanya dipahami oleh pembuatnya. Dari sudut pengadaan, ini selaras dengan kebutuhan *deliverable* administratif dan teknis sebagai bagian dari kendali mutu, pelaporan, audit, dan transisi. (JDIH LKPP)

Ringkasnya, pada proyek *software*, kualitas hasil tidak boleh hanya dibaca dari apa yang tampak di layar. Ia juga harus dibaca dari:

- *source code*;
- repo;
- disiplin *delivery*;
- mutu dokumentasi yang menyertai sistem itu.

20.4 UAT, *pentest*, dan *acceptance*

Bab ini layak ditutup dengan satu kenyataan yang sering menyakitkan dalam proyek *software*: banyak sistem dinyatakan “selesai” padahal belum benar-benar diterima dengan sehat. Karena itu, pembahasan tentang **UAT, *pentest*, dan *acceptance*** menjadi sangat penting.

UAT atau *user acceptance test* penting untuk membuktikan bahwa fungsi bisnis benar-benar berjalan sesuai kebutuhan pengguna. Tetapi UAT saja tidak cukup. Proyek *software* yang memproses data penting, melayani publik, atau terhubung dengan banyak sistem seharusnya juga memperhatikan:

- *pentest* atau setidaknya uji keamanan yang proporsional;
- pengujian integrasi;
- pengujian performa, bila relevan.

Dalam konteks SPBE, keamanan adalah salah satu domain arsitektur, dan layanan digital pemerintah dituntut terpadu serta andal. Karena itu, *acceptance* yang sehat memang tidak cukup hanya membuktikan fungsi, tetapi juga kesiapan integrasi, keamanan, dan operasi. (Peraturan BPK)

Maka, *acceptance* tidak boleh dimaknai sekadar sebagai “pengguna sudah sempat mencoba.” *Acceptance* harus berarti bahwa:

- ada kriteria lulus yang jelas;
- ada bukti pengujian yang memadai;
- ada daftar isu yang diketahui;
- ada keputusan sadar bahwa sistem layak masuk tahap berikutnya.

Untuk sistem tertentu, ini berarti *acceptance* baru layak dilakukan setelah:

- fungsi inti lulus UAT;
- integrasi terbukti;
- hasil uji keamanan minimum tersedia;
- performa cukup untuk beban yang relevan;
- dokumentasi minimum telah diserahkan.

Dalam konteks pengadaan, ini berarti kontrak dan rencana pengujian harus tegas. Kalau penerimaan hasil kerja terlalu kabur, organisasi akan mudah menandatangani serah terima atas sistem yang sebenarnya belum siap. Murah di administrasi, mahal di operasi. Prinsip ini selaras dengan kebutuhan sinkronisasi antara KAK, spesifikasi, kontrak, dan pelaksanaan yang diatur dalam regulasi LKPP. (JDIH LKPP)

Penutup

Bab ini menegaskan bahwa pengadaan *software* dan pengembangan aplikasi harus diperlakukan sebagai **pengadaan hasil digital yang hidup**, bukan sekadar pengadaan fitur. Karena itu, organisasi perlu:

- memilih dengan sadar antara *build*, *buy*, dan *adopt OSS*;
- membedakan karakter *web app*, *mobile app*, ERP, *middleware*, dan API;
- memastikan kontrol atas *source code*, repo, CI/CD, dan dokumentasi;
- menutup proyek dengan UAT, uji keamanan, dan *acceptance* yang benar-benar bermakna.

Dengan pendekatan seperti ini, *software* yang diadakan bukan hanya tampil baik saat demo, tetapi juga:

- masuk akal secara operasional;
- dapat dikendalikan secara teknis;
- lebih siap dipelihara dalam jangka menengah.

Arah ini konsisten dengan pedoman perencanaan dan pelaksanaan pengadaan LKPP serta dengan kerangka arsitektur SPBE yang menekankan integrasi proses bisnis, data, aplikasi, infrastruktur, dan keamanan. (JDIH LKPP)

Tabel Summary. Pengadaan *Software* dan Pengembangan Aplikasi

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
20.1 *Build* vs *buy* vs *adopt OSS*	Keputusan paling strategis dalam pengadaan *software* adalah memilih antara **membangun sendiri**, **membeli solusi jadi**, atau **mengadopsi OSS**. Pilihan ini harus dibaca dari kebutuhan layanan, kontrol yang diinginkan, biaya total, integrasi, dan kapasitas internal, bukan dari tren atau preferensi vendor.	Sebelum menyusun KAK, instansi harus menilai apakah kebutuhannya benar-benar khas, apakah pasar sudah menyediakan solusi matang, dan apakah organisasi siap menanggung biaya implementasi, integrasi, support, serta pengelolaan perubahan jangka menengah.
20.2 *Web app*, *mobile app*, ERP, *middleware*, API	Tidak semua pengadaan *software* punya karakter yang sama. **Web app, mobile app, ERP, middleware, dan API** membawa kebutuhan spesifikasi, integrasi, operasi, dan risiko yang berbeda.	Tim perencana harus jelas sejak awal: siapa pengguna utama, bagaimana pola aksesnya, seberapa besar kebutuhan integrasi, dan apakah sistem ini akan menjadi aplikasi utama, aplikasi pendukung, atau tulang punggung interoperabilitas.
20.3 *Source code*, repo, CI/CD, dokumentasi	Dalam proyek *software*, kualitas tidak cukup dibaca dari tampilan aplikasi. Yang juga menentukan adalah **source code, repo, CI/CD, dan dokumentasi**, karena di sanalah keberlanjutan teknis dan kontrol operasional dibangun.	Kontrak dan deliverable harus menegaskan kebutuhan penyerahan source code sesuai model haknya, pengelolaan repo, dokumentasi deployment, SOP operasi, manual admin/pengguna, dan catatan perubahan agar sistem tidak menjadi “kotak hitam”.
20.4 UAT, *pentest*, dan *acceptance*	Sistem tidak boleh dinyatakan selesai hanya karena bisa didemokan. Diperlukan **UAT, uji keamanan yang proporsional, dan acceptance** yang berbasis kriteria lulus yang jelas.	Dalam kontrak dan rencana uji, instansi harus menetapkan kriteria lulus fungsi bisnis, integrasi, keamanan, performa, dokumentasi minimum, dan penanganan isu sebelum serah terima akhir dilakukan.

Ringkasan

Bab 20 menegaskan bahwa pengadaan software dan pengembangan aplikasi harus diperlakukan sebagai pengadaan hasil digital yang hidup, bukan sekadar pengadaan fitur. Organisasi harus memilih secara sadar antara *build*, *buy*, dan *adopt OSS*; membedakan karakter *web app*, *mobile app*, ERP, *middleware*, dan API; memastikan kontrol atas *source code*, repo, CI/CD, dan dokumentasi; lalu menutup proyek dengan UAT, uji keamanan, dan *acceptance* yang benar-benar bermakna. Dengan pendekatan ini, *software* yang diadakan menjadi **lebih masuk akal secara operasional, lebih terkendali secara teknis, dan lebih siap dipelihara dalam jangka menengah**.

Bab 21. Pengadaan *Managed Service*, SOC, dan Operasi TI

21.1 *Helpdesk*, NOC, SOC, monitoring, *backup*, *patching*

Salah satu kesalahan paling umum dalam pengadaan IT adalah menganggap proyek selesai ketika sistem sudah *go-live*. Padahal, untuk banyak layanan digital, tantangan sesungguhnya justru dimulai setelah sistem berjalan. Di sinilah peran *managed service*, *helpdesk*, NOC, SOC, *monitoring*, *backup*, dan *patching* menjadi sangat penting. Arah ini sejalan dengan kerangka SPBE yang menempatkan manajemen layanan SPBE sebagai bagian dari penyelenggaraan layanan digital pemerintah yang harus berjalan berkesinambungan, berkualitas, responsif, dan adaptif. (Peraturan BPK)

Masing-masing fungsi ini punya peran yang berbeda. **Helpdesk** adalah lapisan dukungan pertama yang berhadapan langsung dengan pengguna dan gangguan harian. **NOC** berfokus menjaga kesehatan infrastruktur, konektivitas, kapasitas, dan ketersediaan layanan. **SOC** menangani deteksi dan respons kejadian keamanan. **Monitoring** memastikan organisasi tidak buta terhadap performa dan insiden. **Backup** menjaga data tetap dapat dipulihkan. **Patching** memastikan sistem tidak dibiarkan rapuh oleh kerentanan yang sudah diketahui. Kalau semua ini tidak dirancang sejak tahap pengadaan, organisasi biasanya hanya membeli “sistem yang hidup”, bukan “layanan yang benar-benar dikelola.”

Dalam **Perpres 95 Tahun 2018**, penyelenggaraan manajemen layanan SPBE memang diarahkan untuk mendukung agar layanan publik berbasis elektronik dan layanan administrasi pemerintahan berbasis elektronik dapat berjalan secara berkesinambungan. Perpres yang sama juga menekankan integrasi proses bisnis, data, layanan SPBE, dan penerapan keamanan SPBE. Itu berarti operasi TI bukan tambahan pascaproyek, tetapi bagian dari tata kelola layanan digital pemerintah itu sendiri. (Peraturan BPK)

Karena itu, pengadaan *managed service* harus dimulai dari pertanyaan yang sangat praktis:

- layanan apa yang harus dijaga;
- siapa pengguna yang dilayani;
- gangguan seperti apa yang paling kritikal;
- bagaimana pemantauan dilakukan;
- siapa yang memegang peran operasional;
- bukti apa yang harus tersedia agar kualitas layanan bisa diukur.

Managed service yang sehat bukan sekadar menyediakan orang jaga, tetapi menyediakan **disiplin operasi**.

21.2 SLA operasional

Kalau organisasi membeli layanan operasi TI, maka yang paling penting bukan hanya keberadaan timnya, tetapi **tingkat layanan yang mereka jaga**. Karena itu, SLA operasional menjadi pusat dari pengadaan layanan operasi.

SLA operasional sebaiknya menjawab hal-hal yang benar-benar bisa diukur, misalnya:

- waktu respons tiket;
- waktu pemulihan insiden;
- target *uptime*;
- frekuensi *backup*;
- tenggat pemasangan *patch* kritis;
- waktu eskalasi ke level berikutnya;
- ketersediaan tim pada jam tertentu;
- kualitas laporan bulanan;
- jumlah gangguan yang tidak terselesaikan melewati batas waktu.

Prinsipnya sederhana: kalau parameter layanan penting, maka parameternya harus ditulis. Kalau tidak ditulis, ia sulit ditagih. Logika ini konsisten dengan pendekatan pengadaan melalui penyedia dalam **Peraturan LKPP Nomor 12 Tahun 2021**, yang kemudian diubah dengan **Peraturan LKPP Nomor 4 Tahun 2024**. Dalam kerangka itu, layanan yang dibeli harus diterjemahkan ke kewajiban yang jelas, terukur, dan bisa dikendalikan selama kontrak berjalan. (JDIH LKPP)

Dalam praktik yang lebih matang, organisasi juga sebaiknya membedakan antara:

- **SLA**, yaitu komitmen layanan yang dikontrakkan;
- **SLO**, yaitu target operasional internal yang dipakai tim agar kualitas tidak hanya bergerak di ambang minimum kontrak.

Dengan cara ini, kontrak tetap realistis, tetapi operasi tidak sekadar diarahkan untuk “lolos penalti.” Layanan yang baik dijaga dengan standar internal yang lebih disiplin daripada angka minimum kontrak. Ini adalah inferensi manajerial yang sangat masuk akal dari model pengelolaan layanan yang terukur. (JDIH LKPP)

21.3 *Shift*, *coverage*, eskalasi, dan pelaporan

Banyak layanan operasi TI tampak baik di proposal, tetapi gagal di realitas karena empat hal tidak jelas: siapa jaga kapan, siapa menangani apa, ke siapa masalah dinaikkan, dan bagaimana semua itu dicatat. Karena itu, *shift*, *coverage*, eskalasi, dan pelaporan harus ditulis dengan tegas sejak awal.

Shift dan **coverage** harus mengikuti karakter layanan. Sistem yang dipakai jam kerja biasa mungkin cukup dengan dukungan operasional di jam kerja dan mekanisme siaga untuk insiden berat. Tetapi sistem transaksi publik, infrastruktur kritis, atau layanan yang menopang banyak unit mungkin memerlukan *coverage* yang lebih panjang, bahkan 24/7. Yang penting bukan memaksakan semua layanan menjadi 24/7, tetapi menyamakan cakupan layanan dengan risiko gangguan dan kebutuhan pengguna. Itu sejalan dengan gagasan SPBE bahwa layanan digital harus dikelola sesuai kebutuhan keberlanjutan layanannya. (Peraturan BPK)

Eskalasi juga harus tegas. Gangguan tidak boleh berhenti di operator pertama tanpa jalur naik yang jelas ke admin, engineer, vendor aplikasi, tim keamanan, atau pimpinan layanan. Pengalaman di banyak proyek menunjukkan bahwa masalah sering membesar bukan semata karena *error* awal, tetapi karena lambat naik level, lambat diputuskan, atau tidak ada pemilik tindakan yang jelas. Dalam layanan operasi, *escalation tree* yang baik sering lebih berharga daripada alat yang mahal.

Pelaporan adalah unsur yang kerap diremehkan. Padahal justru dari laporanlah organisasi bisa membaca:

- pola insiden;
- kualitas layanan;
- kapasitas tim;
- titik rapuh sistem;
- kepatuhan SLA;
- kebutuhan perbaikan.

Dalam konteks kontrak layanan, dokumentasi dan laporan bukan beban administratif semata, tetapi instrumen manajemen. Tanpa laporan yang tertib, organisasi tidak punya bahan yang cukup untuk mengendalikan vendor, memperbaiki operasi, atau membuktikan kinerja saat audit. Ini sangat selaras dengan logika pengadaan layanan melalui penyedia yang mengharuskan kewajiban kontraktual dapat dibaca dan dibuktikan. (JDIH LKPP)

Ringkasnya:

- *shift* menjaga kehadiran;
- *coverage* menjaga cakupan;
- eskalasi menjaga respons;
- pelaporan menjaga pembelajaran.

Kalau salah satu hilang, operasi mudah berubah dari disiplin menjadi sekadar pemadam kebakaran.

21.4 Bukti layanan dan *audit trail*

Bab ini layak ditutup dengan unsur yang paling menentukan akuntabilitas: **bukti layanan** dan **audit trail**. Pada akhirnya, *managed service* harus bisa dibuktikan, bukan hanya diklaim.

Bukti layanan berarti organisasi dapat menunjukkan bahwa layanan memang diberikan sesuai yang dijanjikan, misalnya melalui:

- tiket masuk;
- waktu respons;
- waktu penyelesaian;
- bukti *backup*;
- bukti *patching*;
- log perubahan;
- notulen eskalasi;
- laporan ketersediaan;
- hasil *monitoring*;
- rekaman tindakan saat insiden terjadi.

Sementara itu, **audit trail** berarti ada jejak yang cukup untuk menelusuri siapa melakukan apa, kapan, pada sistem yang mana, dan dengan hasil apa. Tanpa ini, layanan operasi mudah menjadi wilayah gelap: pekerjaan katanya sudah dilakukan, tetapi bukti minim. Dalam konteks SPBE, audit TIK memang mencakup audit Infrastruktur SPBE, audit Aplikasi SPBE, dan audit Keamanan SPBE yang memeriksa tata kelola, fungsionalitas, kinerja, dan aspek teknis lainnya. Itu berarti layanan operasi harus dirancang sejak awal agar **siap diaudit**, bukan baru panik ketika pemeriksaan datang. (Peraturan BPK)

Pada lingkungan yang memproses data pribadi atau bukti transaksi elektronik, kebutuhan *audit trail* juga bertemu langsung dengan kewajiban pelindungan data dan pembuktian elektronik. Maka, kontrak layanan yang sehat sebaiknya meminta bukti yang wajar dan operasional, seperti:

- tiket;
- *dashboard* kinerja;
- log;
- laporan berkala;
- bukti tindakan perbaikan;
- bukti pengujian *restore*;
- jejak koordinasi insiden.

Tanpa bukti layanan, SLA hanya angka. Tanpa *audit trail*, akuntabilitas hanya niat baik. Prinsip ini sangat selaras dengan kebutuhan pengelolaan layanan digital pemerintah yang berkesinambungan dan dapat dievaluasi dalam SPBE. (Peraturan BPK)

Penutup

Bab ini menegaskan bahwa pengadaan *managed service*, SOC, dan operasi TI adalah **pengadaan disiplin layanan**, bukan sekadar pengadaan orang atau alat pemantauan. Organisasi perlu:

- mendefinisikan dengan jelas peran *helpdesk*, NOC, SOC, *monitoring*, *backup*, dan *patching*;
- menulis SLA operasional yang bisa ditagih;
- mengatur *shift*, *coverage*, eskalasi, dan pelaporan secara realistis;
- memastikan semuanya meninggalkan bukti layanan dan *audit trail* yang cukup untuk pengendalian, pembelajaran, dan audit.

Dengan pendekatan seperti ini, operasi TI tidak lagi menjadi fase yang diurus seadanya setelah proyek selesai, tetapi berubah menjadi **fondasi keberlanjutan layanan digital yang benar-benar hidup**. Arah ini didukung oleh **Perpres 95 Tahun 2018** tentang SPBE dan oleh rezim pengadaan melalui penyedia di **Peraturan LKPP Nomor 12 Tahun 2021** sebagaimana telah diubah oleh **Peraturan LKPP Nomor 4 Tahun 2024**. (Peraturan BPK)

Tabel Summary. Pengadaan *Managed Service*, SOC, dan Operasi TI

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
21.1 Helpdesk, NOC, SOC, monitoring, backup, patching	Operasi TI dimulai setelah sistem *go-live*. Karena itu, pengadaan *managed service* harus dibaca sebagai pengadaan **disiplin operasi**: dukungan pengguna, pemantauan infrastruktur, respons keamanan, backup, dan patching yang berkelanjutan.	Sebelum menyusun kontrak, instansi harus menentukan layanan apa yang dijaga, siapa penggunanya, gangguan paling kritikal apa yang harus ditangani, dan bukti operasi apa yang wajib tersedia.
21.2 SLA operasional	Yang paling penting dalam layanan operasi bukan hanya keberadaan tim, tetapi **tingkat layanan** yang bisa diukur dan ditagih: respons tiket, waktu pemulihan, uptime, backup, patching, dan kualitas laporan.	Kontrak harus menuliskan SLA yang konkret dan membedakannya dari SLO internal, sehingga layanan tidak berhenti pada target minimum, tetapi tetap dijaga dengan standar operasional yang lebih disiplin.
21.3 Shift, coverage, eskalasi, dan pelaporan	Layanan operasi akan rapuh bila **shift, coverage, eskalasi, dan pelaporan** tidak diatur dengan jelas. Empat unsur ini menentukan apakah insiden bisa ditangani cepat dan apakah organisasi bisa belajar dari gangguan yang terjadi.	Model operasi harus disesuaikan dengan risiko layanan: jam layanan, cakupan siaga, jalur eskalasi, peran tiap level dukungan, dan bentuk laporan berkala harus tertulis jelas dalam desain layanan maupun kontrak.
21.4 Bukti layanan dan audit trail	*Managed service* harus bisa **dibuktikan**, bukan hanya diklaim. Karena itu, bukti layanan dan *audit trail* menjadi dasar akuntabilitas, pengendalian mutu, dan kesiapan audit.	Kontrak harus mewajibkan tiket, log, laporan SLA, bukti backup, bukti patching, jejak perubahan, dan dokumentasi insiden agar kinerja layanan dapat diverifikasi dan ditelusuri saat ada gangguan atau audit.

Ringkasan

Bab 21 menegaskan bahwa pengadaan managed service, SOC, dan operasi TI adalah pengadaan disiplin layanan, bukan sekadar pengadaan orang atau alat pemantauan. Organisasi harus mendefinisikan dengan jelas fungsi helpdesk, NOC, SOC, monitoring, backup, dan patching; menulis SLA operasional yang bisa diukur; mengatur shift, coverage, eskalasi, dan pelaporan secara realistis; lalu memastikan semua itu meninggalkan bukti layanan dan *audit trail* yang cukup untuk pengendalian, pembelajaran, dan audit. Dengan pendekatan ini, operasi TI menjadi **fondasi keberlanjutan layanan digital**, bukan fase yang dibiarkan berjalan seadanya setelah proyek selesai.

Bab 22. Tata Kelola Proyek, Metode Kerja, dan Pengendalian *Delivery*

22.1 *Waterfall* vs *agile* vs *hybrid*

Salah satu pertanyaan yang paling sering muncul dalam proyek IT adalah: **metode kerja apa yang paling tepat?** Banyak orang berharap ada satu jawaban sederhana, seolah-olah ada satu metode yang otomatis paling baik untuk semua proyek. Kenyataannya tidak demikian. Pemilihan metode kerja harus dibaca dari kebutuhan layanan, tingkat ketidakpastian, kapasitas tim, dan tuntutan akuntabilitas proyek.

Waterfall biasanya lebih cocok ketika kebutuhan relatif stabil, ruang lingkup bisa dirumuskan cukup lengkap sejak awal, ketergantungan antar tahap jelas, dan perubahan tidak diperkirakan terlalu besar. Kelebihan pendekatan ini ada pada:

- struktur tahapan yang jelas;
- dokumentasi yang lebih tertib;
- keterbacaan proses dari analisis sampai serah terima.

Namun, kelemahannya juga jelas: bila kebutuhan awal ternyata belum matang, maka perubahan akan menjadi lebih mahal dan lebih lambat.

Agile lebih masuk akal ketika kebutuhan berkembang, umpan balik pengguna perlu sering masuk, dan sistem lebih aman dibangun secara iteratif. Pendekatan ini memberi ruang untuk:

- pembelajaran bertahap;
- penyesuaian prioritas;
- *delivery* yang lebih sering;
- perbaikan yang lebih cepat.

Tetapi *agile* bukan alasan untuk bekerja tanpa target, tanpa dokumen, atau tanpa kontrol ruang lingkup. Salah kaprah yang sering terjadi adalah menganggap *agile* sebagai izin untuk “jalan dulu saja.” Padahal kalau tidak dikendalikan, cara kerja seperti ini justru membuat proyek kehilangan arah.

Karena itu, untuk banyak proyek pengadaan pemerintah dan institusi publik, pendekatan **hybrid** sering menjadi pilihan paling sehat. Pendekatan ini memungkinkan organisasi tetap menjaga disiplin pada:

- perencanaan;
- KAK;
- kontrak;
- pengendalian hasil;

namun tetap memberi ruang iterasi pada bagian tertentu. Misalnya, analisis kebutuhan, arsitektur, dan batas ruang lingkup tetap dijaga ketat, sementara pengembangan modul, prioritas *backlog*, atau penyempurnaan antarmuka dilakukan secara bertahap. Dengan begitu, *hybrid* bukan kompromi setengah hati, melainkan usaha menyatukan kepastian pengadaan dengan kenyataan proyek digital yang dinamis.

Pelajaran pentingnya sederhana: **metode kerja tidak boleh dipilih karena sedang tren**. Ia harus dipilih karena paling masuk akal untuk kebutuhan layanan, tingkat perubahan, kapasitas tim, dan kewajiban akuntabilitas proyek.

22.2 *Sprint*, *backlog*, *change request*, *governance board*

Setelah metode kerja dipilih, langkah berikutnya adalah memastikan ada mekanisme pengendalian yang nyata. Di sinilah fungsi:

- *sprint*;
- *backlog*;
- *change request*;
- *governance board*;

menjadi sangat penting. Keempatnya berkaitan langsung dengan cara proyek bergerak dari rencana ke realisasi tanpa kehilangan arah.

Sprint berguna untuk memecah pekerjaan ke dalam periode yang lebih pendek, terukur, dan bisa ditinjau. Tetapi *sprint* hanya sehat bila ada:

- prioritas yang jelas;
- definisi selesai yang jelas;
- umpan balik yang tertata.

Kalau tidak, tim memang terlihat sibuk, tetapi hasilnya belum tentu bergerak ke tujuan yang benar.

Backlog penting sebagai alat untuk menyimpan, mengurutkan, dan memutuskan pekerjaan apa yang benar-benar menjadi prioritas. Tanpa *backlog* yang disiplin, proyek akan mudah disusupi oleh permintaan kecil yang tampak sepele, tetapi lama-lama mengubah arah dan biaya secara signifikan. Jadi, *backlog* bukan sekadar daftar kerja, tetapi alat pengendali fokus proyek.

Change request sangat penting, terutama dalam konteks pengadaan, karena proyek IT hampir selalu menghadapi perubahan. Yang berbahaya bukan perubahan itu sendiri, tetapi perubahan yang:

- tidak diakui;
- tidak dihitung;
- atau diselipkan diam-diam ke ruang lingkup awal.

Karena itu, perubahan harus dikelola secara jujur, bukan ditutup-tutupi. Dengan cara ini, organisasi bisa menilai apakah sebuah perubahan masih bisa ditampung, atau memang harus diikuti penyesuaian waktu, biaya, dan prioritas.

Di sinilah **governance board** atau forum pengambilan keputusan menjadi penting. Tidak semua perubahan cukup diselesaikan di level teknis. Perubahan yang menyentuh:

- biaya;
- jadwal;
- prioritas layanan;
- risiko keamanan;
- dampak ke pengguna;

perlu diputuskan oleh forum yang tepat. *Governance board* menjaga agar proyek tidak dikuasai oleh improvisasi sepihak vendor atau dorongan mendadak dari pengguna tertentu. Tanpa forum ini, *backlog* bisa menjadi kacau, dan *sprint* bisa sibuk tetapi proyek tetap melenceng.

22.3 Dokumentasi rapat, *status report*, *risk register*, *issue log*

Banyak proyek gagal bukan karena timnya tidak bekerja, tetapi karena jejak pengambilan keputusan dan masalahnya tidak terdokumentasi dengan baik. Karena itu, dokumentasi proyek bukan pekerjaan tambahan yang membebani, melainkan alat untuk menjaga agar fakta tetap terlihat.

Dokumentasi rapat penting agar keputusan tidak berubah hanya karena ingatan tiap orang berbeda. Dari dokumentasi rapat, organisasi bisa membedakan dengan jelas:

- apa yang baru diusulkan;
- apa yang sudah diputuskan;
- apa tindak lanjutnya;
- siapa yang bertanggung jawab.

Tanpa ini, proyek mudah tenggelam dalam perdebatan yang berulang.

Status report membantu pimpinan, PPK, tim teknis, dan vendor membaca denyut proyek secara berkala. Laporan ini seharusnya memberi gambaran tentang:

- apa yang sudah selesai;
- apa yang tertunda;
- apa hambatannya;
- keputusan apa yang dibutuhkan.

Tanpa *status report* yang disiplin, proyek sering baru terlihat bermasalah ketika keterlambatannya sudah terlalu besar.

Risk register penting untuk memisahkan antara risiko yang masih potensial dan masalah yang sudah benar-benar terjadi. Ia membantu tim membaca sejak awal kemungkinan seperti:

- keterlambatan;
- risiko integrasi;
- risiko keamanan;
- risiko data;
- risiko ketergantungan personel.

Dengan begitu, potensi masalah tidak dibiarkan berkembang diam-diam menjadi krisis.

Issue log berbeda dari *risk register*. *Issue log* dipakai untuk mencatat masalah yang sudah nyata, misalnya:

- bug besar;
- hambatan akses;
- keterlambatan data;
- keputusan yang macet;
- insiden operasional.

Di sinilah organisasi bisa melihat masalah yang benar-benar sedang mengganggu proyek dan perlu segera ditangani.

Intinya, dokumentasi proyek bukan birokrasi tambahan. Ia adalah alat untuk menjaga agar diskusi proyek tidak dikuasai oleh:

- opini;
- lupa;
- saling menyalahkan.

Dalam proyek IT yang kompleks, dokumen kecil yang konsisten sering jauh lebih berharga daripada presentasi besar yang hanya muncul sesekali.

22.4 Menjaga proyek tetap *on-time*, *on-budget*, dan *on-scope*

Pada akhirnya, tata kelola proyek dinilai dari satu pertanyaan besar: **bagaimana menjaga proyek tetap *on-time*, *on-budget*, dan *on-scope* tanpa mengorbankan mutu secara membabi buta?** Inilah inti seluruh Bab 22.

Menjaga proyek tetap **on-time** berarti jujur terhadap:

- kapasitas tim;
- kompleksitas pekerjaan;
- kenyataan integrasi;
- kebutuhan pengujian;
- beban perubahan.

Jadwal yang terlalu optimistis hampir selalu meninggalkan dua warisan:

- kualitas turun;
- konflik naik.

Ketika proyek dijalankan dengan asumsi yang tidak realistis, tahapan yang seharusnya berurutan dipaksa paralel, atau UAT dimulai terlalu cepat, keterlambatan akan cepat membesar.

Menjaga proyek tetap **on-budget** berarti disiplin terhadap ruang lingkup, perubahan, dan prioritas. Proyek digital jarang membengkak karena satu keputusan besar. Yang lebih sering terjadi adalah akumulasi dari banyak perubahan kecil yang tidak pernah dihitung dengan jujur. Karena itu, setiap tambahan seperti:

- fitur baru;
- integrasi tambahan;
- laporan;
- *dashboard*;
- perubahan proses;

harus dinilai dampaknya terhadap waktu dan biaya. *Budget control* bukan berarti menolak semua perubahan, tetapi memastikan bahwa setiap perubahan dipahami konsekuensinya oleh semua pihak.

Menjaga proyek tetap **on-scope** berarti melindungi tujuan awal tanpa menjadi kaku secara membabi buta. *Scope* yang baik harus tegas, tetapi tetap memberi ruang bagi pengelolaan perubahan secara rasional. Di sinilah hubungan sehat antara vendor dan pengguna menjadi sangat penting. Yang dibutuhkan bukan suasana kerja yang terlalu defensif, tetapi komunikasi yang tetap fokus pada fakta dan tujuan proyek.

Pelajaran akhirnya sederhana: proyek yang sehat bukan proyek yang bebas masalah. Proyek yang sehat adalah proyek yang punya cukup disiplin untuk:

- melihat masalah lebih awal;
- membahasnya dengan jujur;
- mengubahnya menjadi keputusan yang terkendali.

Itulah inti tata kelola *delivery* yang matang.

Penutup

Bab ini menegaskan bahwa tata kelola proyek, metode kerja, dan pengendalian *delivery* adalah **jembatan antara desain pengadaan yang baik dan hasil proyek yang benar-benar tercapai**. Karena itu, organisasi perlu:

- memilih metode kerja secara sadar, apakah *waterfall*, *agile*, atau *hybrid*;
- melengkapinya dengan instrumen pengendalian seperti *sprint*, *backlog*, *change request*, dan *governance board*;
- menjaga proyek dengan dokumentasi rapat, *status report*, *risk register*, dan *issue log* yang disiplin;
- memastikan proyek tetap *on-time*, *on-budget*, dan *on-scope* tanpa kehilangan mutu.

Bila semua ini dikelola dengan sehat, peluang proyek untuk benar-benar sampai ke hasil yang diharapkan akan jauh lebih besar. Tetapi bila diabaikan, proyek bisa terlihat sibuk bergerak, namun tidak selalu tiba di tujuan yang semestinya.

Tabel Summary. Tata Kelola Proyek, Metode Kerja, dan Pengendalian Delivery

Subbab	Inti Gagasan	Implikasi Praktis untuk Pengadaan IT
22.1 *Waterfall* vs *agile* vs *hybrid*	Tidak ada satu metode kerja yang otomatis terbaik untuk semua proyek. **Waterfall, agile, dan hybrid** harus dipilih berdasarkan tingkat kepastian kebutuhan, dinamika perubahan, kapasitas tim, dan kebutuhan akuntabilitas proyek.	Sebelum proyek dimulai, instansi harus menentukan pendekatan kerja yang paling masuk akal: *waterfall* untuk kebutuhan stabil, *agile* untuk iterasi yang cepat, atau *hybrid* untuk menyeimbangkan disiplin pengadaan dengan realitas proyek digital yang dinamis.
22.2 *Sprint*, *backlog*, *change request*, *governance board*	Proyek digital memerlukan instrumen pengendalian yang nyata. **Sprint** menjaga ritme kerja, **backlog** menjaga prioritas, **change request** menjaga kejujuran perubahan, dan **governance board** menjaga keputusan tetap diambil pada level yang tepat.	Dalam pengadaan, organisasi perlu menulis mekanisme pengelolaan backlog, persetujuan perubahan, dan forum pengambilan keputusan agar proyek tidak dikuasai improvisasi sepihak atau permintaan tambahan yang tidak terkendali.
22.3 Dokumentasi rapat, *status report*, *risk register*, *issue log*	Proyek sering gagal bukan karena tim tidak bekerja, tetapi karena **keputusan, risiko, dan masalah tidak terdokumentasi dengan disiplin**. Dokumentasi adalah alat kendali, bukan beban administrasi semata.	Tim proyek harus menyiapkan notulen rapat yang jelas, laporan status berkala, daftar risiko, dan catatan isu agar fakta proyek tetap terlihat dan tidak tenggelam oleh opini atau ingatan yang berbeda-beda.
22.4 Menjaga proyek tetap *on-time*, *on-budget*, dan *on-scope*	Tata kelola proyek pada akhirnya diuji dari kemampuan menjaga proyek tetap **tepat waktu, tepat biaya, dan tetap dalam ruang lingkup**, tanpa mengorbankan mutu secara membabi buta.	Pengendalian delivery harus disiplin terhadap jadwal yang realistis, perubahan yang dihitung, prioritas yang jelas, dan komunikasi yang sehat agar proyek tidak sekadar sibuk bergerak tetapi benar-benar sampai ke hasil yang dituju.

Ringkasan

Bab 22 menegaskan bahwa tata kelola proyek, metode kerja, dan pengendalian delivery adalah jembatan antara desain pengadaan yang baik dan hasil proyek yang benar-benar tercapai. Organisasi harus memilih metode kerja secara sadar, melengkapinya dengan sprint, backlog, change request, dan governance board, lalu menjaga proyek dengan dokumentasi rapat, status report, risk register, dan issue log yang disiplin. Dengan pendekatan seperti ini, peluang proyek untuk tetap **on-time, on-budget, dan on-scope** akan jauh lebih besar. Jika diabaikan, proyek bisa tampak sibuk, tetapi tidak selalu sampai ke hasil yang diharapkan.

Bagian VII. Template, Checklist, dan Lampiran Siap Pakai

Bab 23. Template KAK Pengadaan IT

23.1 Template KAK hardware

Bagian KAK	Isi yang Perlu Ditulis	Catatan Praktis
Nama paket	Pengadaan server/storage/jaringan/endpoint/infrastruktur pendukung	Tulis spesifik, jangan terlalu umum
Latar belakang	Masalah layanan saat ini, keterbatasan kapasitas, risiko gangguan, kebutuhan ekspansi, kebutuhan keamanan	Mulai dari kebutuhan layanan, bukan merek perangkat
Tujuan	Menyediakan infrastruktur yang andal, kompatibel, aman, dan siap operasional	Hubungkan dengan layanan yang ditopang
Ruang lingkup	Pengadaan perangkat, instalasi, konfigurasi, integrasi dasar, pengujian, dokumentasi, pelatihan admin	Tegaskan apa yang termasuk dan tidak termasuk
Keluaran utama	Perangkat terpasang, konfigurasi dasar selesai, dokumen konfigurasi, hasil uji fungsi, daftar aset	Jangan hanya menulis “barang diterima”
Spesifikasi umum	Kapasitas, performa, redundansi, kompatibilitas, kebutuhan daya, kebutuhan rack/UPS, kebutuhan jaringan	Gunakan spesifikasi berbasis fungsi dan performa
Keamanan	Hardening dasar, kontrol akses admin, logging, patch awal, segmentasi jaringan bila relevan	Penting untuk server, firewall, switch, AP
Dukungan purna jual	Garansi, waktu respons, penggantian komponen, ketersediaan suku cadang, jalur eskalasi	Tulis SLA support bila perangkat kritikal

Pengujian/UAT	Uji nyala, uji performa dasar, uji redundansi bila ada, uji konektivitas/integrasi dasar	Bukti uji harus menjadi bagian serah terima
Dokumen serah terima	Manual, diagram/topologi, konfigurasi awal, lisensi bila ada, berita acara, pelatihan admin	Wajib untuk audit dan pemeliharaan

23.2 Template KAK *software*

Bagian KAK	Isi yang Perlu Ditulis	Catatan Praktis
Nama paket	Pengembangan/pengadaan aplikasi web, mobile, middleware, ERP, API, atau modul tertentu	Tulis jenis aplikasi dan fungsi utamanya
Latar belakang	Masalah proses bisnis, kebutuhan digitalisasi, kebutuhan integrasi, risiko sistem lama	Jelaskan masalah nyata, bukan slogan transformasi digital
Tujuan	Membangun/mengadakan aplikasi yang mendukung proses bisnis, integrasi data, keamanan, dan layanan pengguna	Tujuan umum + tujuan khusus
Ruang lingkup	Analisis, desain, pengembangan/konfigurasi, integrasi, migrasi data, pengujian, pelatihan, stabilisasi	Batasi ruang lingkup dengan jelas
Keluaran utama	Aplikasi berjalan, source code/repo sesuai haknya, dokumentasi, hasil pengujian, manual admin/user	Deliverable non-fisik wajib ditulis
Fungsi utama	Modul, workflow, hak akses, dashboard, pelaporan, notifikasi, integrasi	Pisahkan fitur wajib dan pengembangan bertahap
Arsitektur & integrasi	Arsitektur aplikasi, API, pertukaran data, dependensi, middleware bila ada	Penting untuk menghindari aplikasi silo
Keamanan & PDP	Kontrol akses, logging, enkripsi, audit trail, pengelolaan data pribadi	Sesuaikan dengan jenis data yang diproses
Pengujian/UAT	SIT, UAT, performance test, security test/pentest sesuai risiko	Acceptance harus berbasis kriteria lulus
Dokumen serah terima	Source code, repo, skrip deployment, dokumen desain, SOP, manual, hasil uji, materi pelatihan	Jangan hanya terima file aplikasi

23.3 Template KAK *cloud/SaaS*

Bagian KAK	Isi yang Perlu Ditulis	Catatan Praktis
Nama paket	Pengadaan layanan cloud IaaS/PaaS/SaaS untuk sistem tertentu	Sebut model layanannya
Latar belakang	Kebutuhan skalabilitas, kecepatan penyediaan, efisiensi operasi, kebutuhan layanan digital	Jelaskan mengapa cloud dipilih
Tujuan	Menyediakan layanan cloud yang andal, aman, terukur, dan mendukung kebutuhan aplikasi/data	Harus terkait kebutuhan layanan
Ruang lingkup	Penyediaan layanan, konfigurasi awal, migrasi bila ada, integrasi, monitoring dasar, dukungan operasional	Jelaskan cakupan layanan vendor
Keluaran utama	Lingkungan cloud aktif, akun admin, konfigurasi awal, dokumentasi, integrasi dasar, dashboard pemantauan	Akun dan akses admin harus jelas
Parameter layanan	Kapasitas, region/lokasi layanan bila relevan, uptime, performa, backup, logging, scaling	Tulis yang benar-benar perlu diukur
Keamanan & PDP	IAM, MFA, enkripsi, log, notifikasi insiden, pengelolaan data pribadi	Penting untuk cloud/SaaS
Portabilitas & exit plan	Ekspor data, dokumentasi konfigurasi, bantuan migrasi keluar, penghapusan data pasca terminasi	Wajib untuk kurangi vendor lock-in
SLA & support	Respons insiden, waktu pemulihan, dukungan teknis, eskalasi, laporan berkala	Jangan hanya menulis “support tersedia”
Dokumen serah terima	Dokumen arsitektur, akun, konfigurasi, backup policy, SOP operasi, laporan implementasi	Bukti layanan harus bisa diaudit

23.4 Template KAK LMS

Bagian KAK	Isi yang Perlu Ditulis	Catatan Praktis
Nama paket	Pengadaan/pengembangan LMS atau platform *e-learning*	Sebut cakupan layanan belajar
Latar belakang	Kebutuhan pembelajaran digital, keterbatasan sistem saat ini, kebutuhan kelas daring, evaluasi, analitik	Fokus pada proses belajar
Tujuan	Menyediakan LMS yang mendukung kelas, materi, kuis, ujian, analitik, dan dukungan pengguna	Harus terkait layanan pembelajaran
Ruang lingkup	Platform, migrasi konten, integrasi sistem akademik, pelatihan admin/pengajar, dukungan operasional	Jelaskan porsi platform vs layanan
Keluaran utama	LMS aktif, kelas dan akun tersedia, konten termigrasi, integrasi dasar jalan, manual admin/pengajar	Deliverable pembelajaran harus konkret
Fungsi utama	Kelas, tugas, kuis, ujian, bank soal, video, forum, analitik, laporan	Bedakan fungsi wajib dan tambahan
Integrasi	SSO, sinkronisasi akun, kelas, peserta, nilai, integrasi akademik/kepegawaian	Sangat penting agar tidak jadi silo
Privasi & aksesibilitas	Data peserta, hak akses dosen/admin/peserta, *mobile access*, kebutuhan bandwidth, akses ramah perangkat	Penting untuk konteks Indonesia
Dukungan operasional	Helpdesk, dukungan masa ujian, pelatihan admin, pelaporan insiden	LMS gagal sering karena support lemah
Pengujian/UAT	UAT kelas, kuis/ujian, beban akses serentak, integrasi, keamanan dasar	Uji jam puncak bila dipakai massal

23.5 Template KAK *e-commerce*

Bagian KAK	Isi yang Perlu Ditulis	Catatan Praktis
Nama paket	Pengadaan/pengembangan *e-commerce*, *marketplace*, atau portal transaksi digital	Sebut model transaksinya
Latar belakang	Kebutuhan kanal transaksi digital, penjualan layanan/produk, keterbatasan proses manual, kebutuhan akuntabilitas	Bukan sekadar “buat toko online”
Tujuan	Menyediakan sistem transaksi elektronik yang aman, terdokumentasi, dan mudah dipakai	Tekankan transaksi dan bukti
Ruang lingkup	Katalog, akun pengguna, pembayaran, notifikasi, dashboard, log aktivitas, pengaduan, integrasi pihak ketiga	Harus jelas dari awal
Keluaran utama	Platform aktif, katalog siap, akun admin, integrasi pembayaran, log transaksi, laporan operasional	Log dan bukti transaksi wajib
Fungsi utama	Katalog, keranjang, checkout, pembayaran, status pesanan, notifikasi, refund/komplain bila relevan	Sesuaikan model bisnis
Integrasi	Payment gateway, bank, notifikasi, pengiriman/logistik bila relevan, ERP/backoffice	Tulis tanggung jawab pihak ketiga
Keamanan & PDP	Keamanan transaksi, audit trail, fraud control, kontrol akun, perlindungan data pelanggan	Sangat penting untuk PMSE
Kepatuhan	PMSE, bukti elektronik, log, tata kelola merchant bila *marketplace*	Perlu untuk pemerintah/BLUD/BUMD juga
Pengujian/UAT	UAT transaksi, uji pembayaran, uji log, uji beban, uji keamanan	Jangan hanya uji tampilan

23.6 Template KAK *managed service*

Bagian KAK	Isi yang Perlu Ditulis	Catatan Praktis
Nama paket	Pengadaan *managed service* untuk helpdesk/NOC/SOC/operasi aplikasi/infrastruktur	Sebut jenis layanan operasi
Latar belakang	Kebutuhan operasi berkelanjutan, keterbatasan SDM internal, kebutuhan pemantauan, respons insiden, patching, backup	Fokus pada kebutuhan layanan operasional
Tujuan	Menyediakan layanan operasi TI yang berkesinambungan, responsif, terdokumentasi, dan siap audit	Tujuan harus terkait mutu layanan
Ruang lingkup	Helpdesk, monitoring, backup, patching, incident response, eskalasi, pelaporan, dukungan pengguna/admin	Tulis sangat rinci agar tidak kabur
Keluaran utama	Tiket layanan, laporan SLA, bukti backup/patching, log operasi, laporan insiden, notulen eskalasi	Deliverable layanan harus berbukti
SLA operasional	Waktu respons, waktu pemulihan, uptime, frekuensi backup, tenggat patch kritis, coverage layanan	Parameter harus bisa diukur
Shift & coverage	Jam kerja, 24/7 atau tidak, on-call, level dukungan, jalur eskalasi	Sesuaikan dengan kritikalitas sistem
Keamanan & audit trail	Logging, kontrol akses, bukti tindakan, pelaporan insiden, dukungan audit	Sangat penting untuk layanan operasi
Tata kelola layanan	PIC, rapat berkala, status report, risk register, issue log, governance	Jangan hanya fokus ke operator
Serah terima & transisi	Dokumen operasi, SOP, daftar isu, handover, exit/transition plan	Penting jika vendor berganti

Ringkasan singkat

Jenis KAK	Fokus Utama
Hardware	kapasitas, kompatibilitas, instalasi, garansi, dukungan
Software	fungsi bisnis, integrasi, keamanan, source code, UAT
Cloud/SaaS	SLA, keamanan, portabilitas data, exit plan
LMS	proses belajar, integrasi akademik, konten, helpdesk
E-commerce	transaksi, pembayaran, log, fraud control, PMSE
Managed service	SLA operasional, monitoring, eskalasi, audit trail

Bab 24. Template Spesifikasi Teknis

24.1 Template spesifikasi server/jaringan

Komponen	Template Spesifikasi Teknis	Catatan Praktis
Tujuan layanan	Perangkat digunakan untuk menopang layanan/aplikasi/sistem dengan kebutuhan kapasitas, keandalan, dan keamanan tertentu	Mulai dari beban kerja, bukan merek
Server	Kapasitas komputasi minimum, jumlah core/logical compute, RAM minimum, kemampuan ekspansi, virtualisasi, redundansi daya, antarmuka jaringan, manajemen jarak jauh	Hindari menyebut merek; tulis fungsi dan performa
Storage	Kapasitas efektif minimum, tipe storage, performa minimum, dukungan snapshot/replication bila perlu, integrasi dengan server/backup	Hitung pertumbuhan data dan kebutuhan backup
Switch	Jumlah port, kecepatan port, dukungan VLAN/routing bila perlu, redundancy/stacking, fitur manajemen, kompatibilitas dengan desain jaringan	Sesuaikan dengan topologi dan segmentasi jaringan
Firewall	Throughput minimum, jumlah interface, dukungan policy management, VPN, logging, kontrol akses, kemampuan high availability bila perlu	Tulis berdasarkan risiko dan kebutuhan keamanan
Access Point (AP)	Cakupan area, jumlah pengguna simultan, standar Wi-Fi yang didukung, manajemen terpusat bila perlu, keamanan akses	Jangan hitung jumlah AP tanpa survei kebutuhan
Endpoint	Profil perangkat sesuai jenis pengguna: administrasi, operator, analitik, desain, atau lapangan; CPU, RAM, storage, layar, baterai bila relevan	Profilkan pengguna agar tidak over/under-spec
UPS & Rack	Kapasitas daya, estimasi backup time, ukuran rack, kompatibilitas ruang, kebutuhan pendinginan dan kabel	Penting untuk kesiapan operasi, bukan sekadar aksesori
Kompatibilitas	Kompatibel dengan sistem operasi, hypervisor, aplikasi, topologi, dan alat manajemen yang digunakan organisasi	Kompatibilitas harus objektif, jangan mengunci vendor

Pengujian	Uji fungsi, uji konektivitas, uji performa dasar, uji redundansi bila relevan	Bukti uji wajib masuk serah terima
Dukungan	Garansi, waktu respons, penggantian komponen, ketersediaan suku cadang, dukungan instalasi	Support harus ditulis, bukan diasumsikan

24.2 Template spesifikasi aplikasi

Komponen	Template Spesifikasi Teknis	Catatan Praktis
Tujuan aplikasi	Aplikasi harus mendukung proses bisnis/layanan tertentu, pengguna tertentu, dan target manfaat tertentu	Hubungkan dengan layanan, bukan sekadar fitur
Jenis aplikasi	Web app, mobile app, middleware, ERP, API, atau kombinasi	Jenis aplikasi memengaruhi model spesifikasi
Fungsi utama	Modul inti, workflow, dashboard, laporan, notifikasi, pencarian, ekspor data, audit trail	Bedakan fitur wajib dan pengembangan bertahap
Pengguna & hak akses	Peran pengguna, tingkat akses, otorisasi berbasis peran, admin/operator/user	Harus terkait keamanan dan tata kelola
Data & integrasi	Data utama yang diproses, integrasi dengan sistem lain, API, pertukaran data, format data	Penting untuk hindari aplikasi silo
Kinerja	Waktu respons, jumlah pengguna simultan, beban transaksi, target availability bila relevan	Jangan hanya tulis “harus cepat”
Keamanan	Logging, enkripsi, manajemen sesi, kontrol akses, hardening aplikasi, audit trail	Sesuaikan dengan risiko sistem
Dokumentasi teknis	Dokumen analisis, desain, manual admin, manual pengguna, panduan instalasi/deployment	Deliverable dokumentasi harus jelas
Source code & repo	Penyerahan source code sesuai haknya, penggunaan repo, histori perubahan, skrip deployment	Sangat penting untuk sistem custom/OSS
Pengujian	SIT, UAT, integration test, performance test, security test/pentest sesuai risiko	Acceptance harus berbasis hasil uji
Pelatihan & stabilisasi	Pelatihan admin/user, masa stabilisasi, perbaikan bug pasca go-live	Jangan lupa support awal pasca implementasi

24.3 Template spesifikasi keamanan

Komponen	Template Spesifikasi Teknis	Catatan Praktis
Kontrol akses	Akses berbasis peran, least privilege, pemisahan admin dan user, pengelolaan akun	Dasar dari semua kontrol keamanan
Autentikasi	Password policy, MFA untuk admin/akses kritikal, manajemen sesi, lockout policy	Tulis sesuai tingkat risiko
Logging & audit trail	Perekaman aktivitas penting, akses log, retensi log, kemampuan ekspor log	Penting untuk audit dan investigasi
Enkripsi	Enkripsi saat transit, enkripsi saat tersimpan, pengelolaan kunci bila relevan	Sesuaikan dengan jenis data
Hardening	Penonaktifan akun/servis default, konfigurasi aman, baseline konfigurasi	Harus diminta sejak awal
Patching	Mekanisme patching, tenggat patch kritis, pengujian patch, dokumentasi perubahan	Jangan biarkan patching jadi urusan belakangan
Monitoring keamanan	Alert insiden, integrasi SIEM/SOC bila perlu, pemantauan kejadian keamanan	Sesuaikan dengan kritikalitas sistem
Backup & recovery	Backup berkala, retensi, uji restore, target recovery bila relevan	Keamanan bukan hanya pencegahan
Uji keamanan	Vulnerability scanning, penetration test, review hak akses, verifikasi logging/enkripsi	Tulis sebagai bagian acceptance
Insiden	Kewajiban notifikasi, jalur eskalasi, preservasi log, bantuan investigasi	Penting untuk kontrak juga
Kepatuhan	Selaras dengan PDP, SPBE, audit keamanan, dan tata kelola sistem elektronik	Spesifikasi keamanan harus siap audit

24.4 Template spesifikasi cloud

Komponen	Template Spesifikasi Teknis	Catatan Praktis
Model layanan	IaaS, PaaS, SaaS	Harus jelas sejak awal
Tujuan penggunaan	Skalabilitas, elastisitas, percepatan implementasi, efisiensi operasi, atau kebutuhan khusus lain	Jelaskan kenapa cloud dipilih
Kapasitas layanan	Compute, storage, bandwidth, scaling, concurrency, environment dev/test/prod bila perlu	Hindari estimasi asal
Region/lokasi	Region layanan atau lokasi pemrosesan bila relevan	Penting untuk latensi/kepatuhan tertentu
Ketersediaan layanan	Uptime minimum, SLA, failover, DR bila perlu	Tulis yang benar-benar dapat ditagih
Keamanan	IAM, MFA, enkripsi, logging, monitoring, notifikasi insiden	Cloud tidak boleh mengurangi kedalaman keamanan
Backup & recovery	Backup policy, retensi, restore, target pemulihan, snapshot bila perlu	Jangan diasumsikan otomatis cukup
Integrasi	API, SSO, sinkronisasi data, integrasi dengan sistem internal/eksternal	Sangat penting untuk SaaS
Portabilitas data	Ekspor data, format data, dokumentasi konfigurasi, hak akses admin	Inti untuk mengurangi vendor lock-in
Exit plan	Bantuan migrasi keluar, penghapusan data pasca terminasi, handover dokumentasi	Harus tertulis sejak awal
Dukungan layanan	Helpdesk, response insiden, jalur eskalasi, laporan SLA	Cloud adalah layanan, bukan hanya resource

24.5 Template spesifikasi layanan SDM

Komponen	Template Spesifikasi Teknis	Catatan Praktis
Tujuan layanan SDM	Penyediaan personel/professional service untuk fungsi tertentu dalam proyek/operasi IT	Jangan hanya tulis “menyediakan tenaga ahli”
Peran yang dibutuhkan	PM, BA, system analyst, architect, developer, QA, security, DevOps, support, trainer, dsb.	Tulis per peran, bukan gelondongan
Ruang lingkup kerja	Tugas utama tiap peran, output yang harus dihasilkan, keterlibatan pada fase proyek	Harus terkait deliverable nyata
Kualifikasi minimum	Pendidikan, kompetensi teknis, pengalaman relevan, sertifikasi bila benar-benar relevan	Sertifikat harus dikaitkan dengan peran
Tingkat senioritas	Junior, madya, senior, lead sesuai kebutuhan	Sesuaikan dengan kompleksitas proyek
Model keterlibatan	Full-time, part-time, on-call, harian, bulanan, shift, atau berbasis milestone	Mempengaruhi HPS dan SLA
Output per peran	Dokumen analisis, desain, kode, hasil uji, laporan, dukungan operasi, pelatihan	SDM harus dinilai dari output, bukan kehadiran
Evaluasi kualitas personel	CV, portofolio, pengalaman proyek sejenis, wawancara/presentasi bila relevan	Portofolio sering lebih jujur daripada CV
SLA layanan SDM	Kehadiran, respons, kualitas deliverable, dukungan insiden, eskalasi	Penting untuk managed service/ops
Pelaporan & audit trail	Timesheet bila perlu, status report, issue log, bukti pekerjaan, notulen	Bukti kerja penting untuk pembayaran dan audit
Transisi/handover	Knowledge transfer, dokumentasi, serah terima tugas, exit personel	Hindari ketergantungan pada individu

Ringkasan singkat

Jenis template	Fokus utama
Server/Jaringan	kapasitas, kompatibilitas, redundansi, dukungan
Aplikasi	fungsi bisnis, integrasi, source code, pengujian
Keamanan	akses, logging, enkripsi, patching, insiden
Cloud	SLA, IAM, backup, portabilitas, exit plan
Layanan SDM	peran, kualifikasi, output, SLA, bukti kerja

Bab 25. Template HPS dan Biaya SDM

25.1 Template HPS barang

Komponen	Template Isi	Catatan Praktis
Nama paket	HPS pengadaan perangkat/infrastruktur/aplikasi jadi/lisensi tertentu	Sebut objek HPS dengan jelas
Dasar kebutuhan	Ringkasan kebutuhan layanan, kapasitas, jumlah unit, lokasi, dan tujuan penggunaan	HPS harus lahir dari kebutuhan, bukan angka pasar semata
Komponen utama	Unit barang/perangkat/lisensi inti yang dibeli	Tulis per item, jangan gelondongan
Komponen pendukung	Instalasi, konfigurasi awal, aksesoris, rack, UPS, kabel, training dasar, dokumentasi, garansi tambahan bila relevan	Komponen pendukung sering dilupakan
Jumlah / volume	Unit, user, node, lokasi, kapasitas, bulan/tahun lisensi	Pastikan satuan konsisten
Harga satuan	Harga per item/per lisensi/per unit kapasitas	Gunakan pembanding yang terdokumentasi
Subtotal	Jumlah × harga satuan	Pisahkan per komponen
Biaya distribusi/implementasi awal	Ongkir, instalasi, deployment awal, aktivasi bila ada	Tergantung jenis barang
Pajak	PPN/komponen pajak relevan sesuai skema pengadaan	Tegaskan bruto atau neto
Total HPS	Total seluruh komponen	Harus bisa dijelaskan logikanya

Sumber pembanding	E-katalog, toko daring, penawaran pembanding, daftar harga resmi, referensi pasar	Simpan bukti pembanding
Asumsi	Kurs, lokasi, jumlah pengguna, kebutuhan kapasitas, masa lisensi	Wajib agar auditor paham dasar hitung

25.2 Template HPS jasa

Komponen	Template Isi	Catatan Praktis
Nama paket	HPS jasa pengembangan, integrasi, managed service, audit, pelatihan, support, dsb.	Jelaskan jenis jasanya
Dasar kebutuhan	Masalah yang diselesaikan, output yang diharapkan, durasi pekerjaan, model kerja	Harus nyambung dengan KAK
Ruang lingkup jasa	Analisis, desain, implementasi, migrasi, support, pelatihan, monitoring, pelaporan, audit, dsb.	Jangan hanya tulis “jasa IT”
Komponen SDM	Peran personel, level, durasi, intensitas keterlibatan	Dasar utama HPS jasa
Komponen non-SDM	Tools, perjalanan bila relevan, environment uji, biaya komunikasi, training material, biaya operasional layanan	Sesuaikan dengan paket
Tarif per peran / komponen	Tarif harian/bulanan/berbasis milestone	Gunakan rate card yang masuk akal
Durasi / volume	Hari kerja, bulan, sprint, fase, shift, lokasi	Tulis satuan yang jelas
Subtotal per komponen	Tarif × volume	Pisah agar transparan
Cadangan risiko wajar	Hanya bila relevan dan dapat dijustifikasi	Jangan jadi ruang gelap markup
Pajak	PPN dan komponen perpajakan lain yang relevan	Jelaskan struktur nilai
Total HPS jasa	Total biaya jasa	Harus konsisten dengan ruang lingkup
Sumber pembanding	SBM/SBK, benchmark pasar, proyek sejenis, penawaran pembanding	Kombinasikan resmi + pasar
Asumsi	Kompleksitas, model keterlibatan, jumlah user/lokasi, jam layanan	Sangat penting untuk jasa IT

25.3 Template TCO 3 tahun dan 5 tahun

Komponen	Template Isi	Catatan Praktis
Objek yang dihitung	Sistem/aplikasi/cloud/infrastruktur/layanan tertentu	TCO harus per sistem/solusi
Biaya tahun 0 / awal	Pengadaan awal, lisensi awal, implementasi awal, migrasi, pelatihan awal, instalasi	Ini bukan satu-satunya biaya
Biaya tahunan rutin	Subscription, cloud bill, support, maintenance, patching, monitoring, audit berkala, backup, SDM operasi	Inti dari TCO nyata
Biaya peningkatan	Upgrade kapasitas, penambahan user, penambahan storage, ekspansi modul, integrasi tambahan	Jangan lupakan pertumbuhan
Biaya keamanan	Pentest, audit keamanan, hardening, log management, incident response	Wajib untuk sistem penting
Biaya ketahanan layanan	Backup, DR, BCP, redundansi, restore test	Sering terlupakan
Biaya SDM internal / vendor	Admin, operator, helpdesk, managed service, support engineer	TCO bukan hanya vendor bill
Biaya perubahan organisasi	Pelatihan lanjutan, sosialisasi, revisi SOP, pendampingan user	Penting untuk sistem yang hidup
Biaya keluar / exit cost	Ekspor data, migrasi keluar, transisi vendor, penutupan akun/layanan	Penting untuk cloud/SaaS/proprietary
TCO 3 tahun	Total seluruh biaya tahun 0–3	Minimal untuk banyak proyek
TCO 5 tahun	Total seluruh biaya tahun 0–5	Lebih sehat untuk sistem strategis
Analisis singkat	Mana biaya dominan, kapan biaya melonjak, apa risiko biaya tersembunyi	TCO harus dibaca, bukan hanya dijumlah

25.4 Template *rate card* SDM

Peran / Komponen	Template Isi	Catatan Praktis
Nama peran	PM, BA, system analyst, architect, developer, QA, security, DevOps, support, trainer, dsb.	Buat per peran, bukan gelondongan
Level / senioritas	Junior / Madya / Senior / Lead	Harus konsisten antar proyek
Model tarif	Harian / bulanan / shift / berbasis milestone	Pilih yang cocok dengan proyek
Tarif dasar	Nilai tarif per satuan waktu	Gunakan benchmark yang jelas
Koefisien keterlibatan	Full-time, 50%, on-call, jam tertentu, sprint tertentu	Penting untuk hitung realistis
Durasi	Hari, bulan, sprint, fase	Jangan samakan semua peran
Output utama	Deliverable atau fungsi kerja yang diharapkan dari peran itu	Bantu uji kewajaran biaya
Sertifikasi / kompetensi	Bila relevan dengan peran	Jangan jadikan formalitas
Pengalaman minimum	Tahun dan/atau jenis proyek relevan	Fokus ke relevansi, bukan angka saja
Subtotal per peran	Tarif × keterlibatan × durasi	Dasar HPS jasa SDM
Catatan kewajaran	Sumber benchmark, alasan level, alasan durasi	Penting untuk audit dan evaluasi

25.5 Template pembanding pasar

Komponen	Template Isi	Catatan Praktis
Objek pembanding	Barang, lisensi, jasa, rate card SDM, cloud service, managed service	Harus spesifik
Sumber pembanding 1	E-katalog / situs resmi / penawaran vendor / kontrak sejenis / laporan pasar	Simpan bukti
Sumber pembanding 2	Pembanding lain dari pasar	Minimal 2–3 sumber bila memungkinkan
Sumber pembanding 3	Pembanding lain yang relevan	Hindari satu sumber tunggal
Parameter yang dibandingkan	Kapasitas, fungsi, level layanan, peran, durasi, garansi, jumlah user, dll.	Bandingkan yang setara
Harga / tarif	Nilai dari masing-masing sumber	Harus apples to apples
Kelebihan / kekurangan	Catatan mutu, cakupan, dukungan, syarat kontrak, lisensi, SLA	Harga tanpa konteks menyesatkan
Penyesuaian	Penyesuaian lokasi, volume, kompleksitas, dukungan, pajak, kurs, durasi	Jelaskan jika ada
Nilai rujukan wajar	Kisaran atau nilai yang dipilih sebagai dasar	Jangan selalu ambil paling murah
Alasan pemilihan	Kenapa angka ini dipakai untuk HPS	Bagian penting untuk akuntabilitas
Tanggal pembanding	Tanggal sumber harga/tarif	Penting untuk validitas data

Ringkasan singkat

Template	Fokus utama
HPS barang	item, volume, harga satuan, komponen pendukung, pajak
HPS jasa	ruang lingkup, SDM, non-SDM, tarif, durasi, asumsi
TCO 3/5 tahun	biaya awal, biaya rutin, biaya tumbuh, biaya keluar
Rate card SDM	peran, level, tarif, durasi, output, kewajaran
Pembanding pasar	sumber, parameter setara, penyesuaian, alasan nilai rujukan

Catatan pakai cepat

Kondisi	Template yang diprioritaskan
Pengadaan server/perangkat/lisensi jadi	**Template HPS barang**
Pengembangan aplikasi / audit / managed service	**Template HPS jasa**
Keputusan buy vs cloud vs OSS jangka menengah	**Template TCO 3 tahun dan 5 tahun**
Menyusun biaya personel inti vendor	**Template rate card SDM**
Membela kewajaran HPS di audit/evaluasi	**Template pembanding pasar**

Bab 26. Template Evaluasi dan *Due Diligence* Vendor

26.1 Checklist legal vendor

Komponen	Template Isi	Catatan Praktis
Identitas badan usaha	Nama badan usaha, NIB/izin usaha relevan, NPWP, alamat, status aktif	Pastikan identitas hukum konsisten di semua dokumen
Legalitas usaha	Akta/perubahan terakhir, izin yang relevan, status operasional	Sesuaikan dengan jenis pengadaan dan sektor
Kepatuhan kontraktual	Tidak sedang dalam sengketa berat yang mengganggu delivery, tidak bermasalah secara serius pada kontrak sejenis	Riwayat sengketa penting untuk menilai risiko delivery
Kondisi keuangan	Bukti kemampuan finansial dasar, working capital memadai, tidak ada tanda tekanan keuangan serius	*Due diligence* vendor perlu membaca robustness finansial
Asuransi / tanggung jawab	Polis atau bukti perlindungan risiko yang relevan bila diperlukan	Penting untuk proyek bernilai besar atau berisiko tinggi
Subkontraktor	Daftar subkontraktor yang sudah ada/potensial, fungsi yang dialihkan	Vendor harus terbuka soal penggunaan downstream subcontractor
Kepatuhan umum	Komitmen pada kerahasiaan, anti-fraud, anti-suap, dan kepatuhan hukum	Masukkan sebagai syarat kontrak/dokumen evaluasi
Narasi kelayakan legal	Ringkasan apakah vendor layak secara legal atau ada catatan risiko	Buat singkat, tegas, dan dapat diaudit

26.2 Checklist teknis

Komponen	Template Isi	Catatan Praktis
Pengalaman proyek sejenis	Daftar proyek relevan, skala, kompleksitas, jenis sistem, sektor	Fokus pada relevansi, bukan jumlah proyek saja
Kesesuaian solusi	Kecocokan dengan kebutuhan: hardware/software/cloud/managed service/LMS/e-commerce	Uji fit terhadap kebutuhan, bukan sekadar demo
Arsitektur & integrasi	Kemampuan integrasi, API, interoperabilitas, dukungan data exchange	Penting untuk hindari aplikasi silo
Metode kerja	Waterfall/agile/hybrid, cara kelola backlog/perubahan, governance	Harus cocok dengan tipe proyek
Mutu deliverable	Dokumen analisis, desain, source code, repo, manual, laporan uji, SOP	Jangan hanya menilai UI atau brosur
Kesiapan pengujian	UAT, integration test, performance test, security test/pentest sesuai risiko	Acceptance harus berbasis bukti
Kemampuan dukungan teknis	Helpdesk, support, escalation path, coverage, dokumentasi operasional	Penting untuk proyek operasional
Referensi teknis	Kontak/jejak referensi klien sebelumnya bila tersedia	Sangat berguna untuk validasi kualitas nyata
Narasi kelayakan teknis	Ringkasan kelebihan, celah, dan risiko teknis vendor	Pakai bahasa singkat dan objektif

26.3 Checklist keamanan

Komponen	Template Isi	Catatan Praktis
Tanggung jawab keamanan	PIC keamanan vendor selama siklus kontrak	Kontrak harus jelas siapa yang accountable
Kontrol akses	IAM, role-based access, pemisahan akun admin/user, MFA bila relevan	Minimum untuk layanan kritikal
Logging & audit trail	Ketersediaan log, retensi log, kemampuan ekspor, jejak perubahan	Penting untuk audit dan insiden
Enkripsi	Enkripsi data in transit dan at rest bila relevan	Sesuaikan dengan jenis layanan dan data
Patching & vulnerability management	Proses patching, SLA patch kritis, vulnerability handling	Vendor harus punya proses berkelanjutan
Backup & recovery	Backup policy, restore test, target recovery, bukti pengujian	Jangan terima klaim tanpa bukti
Insiden keamanan	Jalur notifikasi, waktu eskalasi, bantuan investigasi, preservasi log	Wajib untuk layanan digital penting
Subkontraktor & keamanan	Apakah subkontraktor tunduk pada kontrol yang sama	Vendor tetap bertanggung jawab atas subkontraktor
Kompetensi keamanan SDM	Bukti pelatihan keamanan atau sertifikasi relevan bila diperlukan	Vendor harus bisa menunjukkan kecukupan skill keamanan
Narasi kelayakan keamanan	Ringkasan apakah kontrol keamanan memadai atau ada gap besar	Gunakan untuk keputusan risiko

26.4 Checklist PSE/PDP

Komponen	Template Isi	Catatan Praktis
Status PSE	Apakah layanan/vendor termasuk PSE yang relevan dan bagaimana status kepatuhannya	Penting untuk cloud/SaaS/layanan elektronik
Peran data	Pengendali, prosesor, atau relasi multi-pihak	Harus jelas sejak awal kontrak
Kategori data	Jenis data pribadi/non-pribadi yang diproses	Peta data wajib ada
Dasar pemrosesan	Tujuan pemrosesan dan dasar hukumnya	Penting untuk kepatuhan PDP
Retensi & penghapusan	Kebijakan simpan data, penghapusan, ekspor, anonimisasi bila relevan	Jangan biarkan data “disimpan selamanya”
Hak subjek data	Dukungan untuk akses, koreksi, atau tindakan lain yang relevan	Penting untuk sistem yang memproses data pribadi
Notifikasi insiden	Mekanisme respons insiden data dan dukungan notifikasi	Harus tertulis di kontrak dan SOP
Lokasi / alur data	Region/lokasi pemrosesan bila relevan, pembagian data ke pihak ketiga	Penting untuk cloud/SaaS tertentu
Dokumentasi kepatuhan	Kebijakan privasi, SOP data, klausul kontrak, rekam pemrosesan bila relevan	Nilai bukan hanya dari klaim verbal
Narasi kelayakan PSE/PDP	Ringkasan apakah vendor cukup matang dari sisi kepatuhan	Gunakan sebagai dasar mitigasi

26.5 Checklist kapasitas SDM

Komponen	Template Isi	Catatan Praktis
Peran inti tersedia	PM, BA, architect, developer, QA, security, DevOps, support, trainer, dsb.	Harus cocok dengan ruang lingkup proyek
Jumlah personel memadai	Jumlah tim dibanding beban delivery dan jadwal	Jangan hanya melihat satu nama kunci
Kompetensi inti	Kesesuaian skill dengan kebutuhan proyek	Sertifikat harus relevan dengan peran
Pengalaman relevan	Pengalaman pada sistem/lingkungan/kompleksitas serupa	Relevansi lebih penting daripada angka tahun semata
Portofolio nyata	Bukti proyek, artefak kerja, demo, repositori, studi kasus	Portofolio lebih jujur daripada CV saja
Ketersediaan personel	Full-time/part-time/on-call, periode keterlibatan, potensi konflik penugasan	Cegah “personel kertas”
Pelatihan & pembaruan skill	Bukti pelatihan rutin, program peningkatan kompetensi	Bagian dari kualitas jangka menengah vendor
Ketergantungan individu	Apakah delivery tergantung pada satu-dua orang saja	Risiko besar bila knowledge tidak tersebar
Narasi kapasitas SDM	Ringkasan apakah tim cukup kuat untuk delivery	Uji terhadap jadwal dan kompleksitas proyek

26.6 Checklist keberlanjutan layanan

Komponen	Template Isi	Catatan Praktis
Model support	Helpdesk, NOC, SOC, support aplikasi, coverage jam layanan	Harus sesuai kritikalitas layanan
SLA operasional	Waktu respons, waktu pemulihan, uptime, patching, backup, laporan	Jangan hanya menerima janji umum
BCP/DR	Continuity plan, disaster recovery, restore test, failover bila relevan	Vendor harus siap untuk hari buruk
Lifecycle support	Dukungan versi, patch keamanan, upgrade, EOS/EOL awareness	Penting untuk hardware/software/cloud
Portabilitas & exit plan	Ekspor data, handover, dokumentasi, bantuan transisi	Kritis untuk hindari lock-in
Bukti layanan	Tiket, log, dashboard SLA, laporan bulanan, bukti backup/patching	Layanan harus bisa diaudit
Hubungan vendor–subkontraktor	Kejelasan kontrol atas pihak ketiga dan continuity mereka	Bagian dari risiko supply chain
Rencana perbaikan berkelanjutan	Mekanisme review berkala, RCA, corrective action, improvement plan	Vendor yang baik tidak hanya reaktif
Narasi keberlanjutan layanan	Ringkasan apakah vendor siap menopang layanan dalam jangka menengah	Penting untuk managed service/cloud/SaaS

Ringkasan singkat

Checklist	Fokus utama
Legal vendor	legalitas, finansial, subkontraktor, sengketa
Teknis	pengalaman relevan, arsitektur, integrasi, deliverable
Keamanan	akses, log, patching, backup, insiden, keamanan subkontraktor
PSE/PDP	status PSE, peran data, retensi, insiden, kepatuhan privasi
Kapasitas SDM	peran inti, pengalaman, portofolio, ketersediaan tim
Keberlanjutan layanan	SLA, support, BCP/DR, exit plan, bukti layanan

Catatan pakai cepat

Kondisi	Checklist yang diprioritaskan
Pengadaan vendor baru berisiko sedang/tinggi	**Legal + Teknis + SDM**
Cloud/SaaS/layanan elektronik	**PSE/PDP + Keamanan + Keberlanjutan layanan**
Managed service/NOC/SOC/helpdesk	**Keamanan + SDM + Keberlanjutan layanan**
LMS/e-commerce/sistem publik	**Teknis + PDP + Keamanan + Bukti layanan**
Vendor memakai subkontraktor	**Legal + Keamanan + Keberlanjutan layanan**

Bab 27. Checklist Audit dan Kesiapan Serah Terima

27.1 Checklist dokumen

Komponen	Checklist	Catatan Praktis
KAK final	KAK final tersedia dan konsisten dengan kebutuhan layanan serta ruang lingkup kontrak	Pastikan versi final, bukan draft kerja
Spesifikasi teknis final	Spesifikasi final tersedia dan konsisten dengan barang/jasa yang diterima	Penting untuk pembuktian kesesuaian
HPS dan dasar perhitungannya	Dokumen HPS, pembanding pasar, asumsi, dan justifikasi tersedia	Sangat penting untuk audit kewajaran biaya
Kontrak dan adendum	Kontrak utama, adendum, SLA, lampiran deliverable tersedia lengkap	Audit sering gagal di dokumen perubahan
Dokumen desain	Dokumen analisis, desain, arsitektur, integrasi, konfigurasi tersedia sesuai jenis proyek	Wajib untuk software, cloud, dan sistem kompleks
Dokumen pengujian	UAT, hasil uji integrasi, performa, keamanan, dan uji fungsi tersedia	Acceptance harus berbasis bukti
Dokumen pelaksanaan	Notulen rapat, status report, risk register, issue log tersedia	Menunjukkan proyek dikelola, bukan sekadar dijalankan
Dokumen serah terima	BAST/BAST parsial, daftar aset, daftar akun, daftar isu terbuka tersedia	Pastikan tidak ada celah pada fase handover
Dokumen operasional	SOP, manual admin, manual pengguna, panduan backup/restore, panduan eskalasi tersedia	Wajib untuk keberlanjutan operasi
Ringkasan kesiapan dokumen	Semua dokumen minimum lengkap / ada gap yang harus ditutup	Buat status: lengkap, sebagian, belum ada

27.2 Checklist keamanan

Komponen	Checklist	Catatan Praktis
Kontrol akses	Role dan hak akses telah disetel sesuai fungsi	Audit akan melihat least privilege
MFA	MFA aktif untuk akun admin/akses kritikal bila relevan	Minimum untuk layanan penting
Logging	Log aktivitas penting aktif, retensi log jelas, akses log diatur	Penting untuk audit dan insiden
Enkripsi	Enkripsi in transit dan/atau at rest diterapkan sesuai kebutuhan	Sesuaikan dengan klasifikasi data
Hardening	Akun default, layanan tidak perlu, dan konfigurasi lemah telah ditangani	Jangan serah terima sistem mentah
Patching	Patch keamanan kritikal telah diterapkan dan terdokumentasi	Bukti patching wajib ada
Uji keamanan	Vulnerability scan/pentest/review keamanan telah dilakukan sesuai risiko	Minimal harus ada bukti tindak lanjut
Notifikasi insiden	Jalur notifikasi dan SOP insiden tersedia	Penting untuk operasi pasca serah terima
Kesiapan audit keamanan	Bukti kontrol cukup untuk audit keamanan SPBE atau audit internal	Sistem harus siap diaudit, bukan hanya dipakai

27.3 Checklist hak akses

Komponen	Checklist	Catatan Praktis
Daftar akun admin	Semua akun admin dicatat dan diserahkan sesuai otorisasi	Jangan ada akun admin “tersembunyi”
Akun vendor sementara	Akun vendor sementara ditutup/diubah/diatur sesuai kebijakan	Hindari akses tersisa pasca proyek
Hak akses pengguna	Hak akses user sudah sesuai unit/peran/fungsi	Cegah hak terlalu luas
Kredensial & reset	Mekanisme ganti password/reset admin telah dilakukan	Idealnya kredensial awal diganti saat handover
SSO/IAM	Integrasi IAM/SSO sudah diuji bila relevan	Penting untuk sistem besar
Dokumentasi akses	Daftar akun, role matrix, dan otorisasi tersedia	Wajib untuk audit dan operasi
Prosedur onboarding/offboarding	SOP tambah/hapus user/admin tersedia	Penting untuk keberlanjutan
Narasi kesiapan akses	Tidak ada ketergantungan akses yang hanya diketahui vendor	Ini titik rawan vendor lock-in operasional

27.4 Checklist *source code*

Komponen	Checklist	Catatan Praktis
Penyerahan source code	Source code diserahkan sesuai model hak pakai/hak modifikasi yang disepakati	Relevan untuk software custom/OSS
Akses repo	Repo aktif dan dapat diakses pihak yang berhak	Jangan hanya serah file zip
Histori perubahan	Commit history/log perubahan tersedia dan terbaca	Penting untuk transisi tim
Branch/tag rilis	Versi rilis yang diserahterimakan ditandai jelas	Hindari kebingungan versi produksi
Build/deploy script	Skrip build, deploy, migration, dan rollback tersedia bila relevan	Penting untuk keberlanjutan teknis
Dependensi	Daftar dependensi/library/framework tersedia	Sangat penting untuk patching dan audit
Dokumentasi teknis	README, arsitektur, konfigurasi environment, API docs tersedia	Source code tanpa dokumentasi sering tidak berguna
Lisensi komponen	Lisensi OSS atau komponen pihak ketiga terdokumentasi	Penting untuk kepatuhan
Kesiapan transisi teknis	Tim internal/vendor baru dapat memahami dan melanjutkan sistem	Ini uji nyata dari kualitas handover

27.5 Checklist *backup* dan DR

Komponen	Checklist	Catatan Praktis
Backup policy	Kebijakan backup tersedia: frekuensi, retensi, media, cakupan	Harus sesuai jenis data dan layanan
Backup berjalan	Backup telah aktif dan bukti pelaksanaannya tersedia	Jangan hanya tertulis di SOP
Restore test	Uji restore telah dilakukan dan hasilnya terdokumentasi	Bukti paling penting dari backup
DR plan	Rencana pemulihan bencana tersedia dan realistis	Sesuaikan kritikalitas sistem
Target pemulihan	RTO/RPO atau target pemulihan lain ditetapkan bila relevan	Harus masuk akal dan dapat diuji
Redundansi	Komponen redundan/failover tersedia bila memang disyaratkan	Tidak semua sistem perlu mahal, tapi yang kritikal perlu jelas
Peran saat insiden	PIC backup/restore/DR sudah ditetapkan	Tanpa PIC, DR hanya jadi dokumen
Dokumentasi pemulihan	Langkah restore, failover, verifikasi layanan tersedia	Penting untuk operasi nyata
Kesiapan keberlanjutan	Sistem tetap bisa dipulihkan tanpa vendor utama hadir langsung	Ini indikator kematangan operasional

27.6 Checklist pelatihan dan SOP

Komponen	Checklist	Catatan Praktis
Pelatihan admin	Admin telah dilatih dan daftar hadir/materi tersedia	Harus lebih dari sesi formalitas
Pelatihan operator/user kunci	User kunci/operator telah dilatih sesuai perannya	Penting untuk sistem layanan
Materi pelatihan	Slide, video, panduan, modul, FAQ tersedia	Agar pelatihan bisa diulang
Manual admin	Manual teknis admin tersedia dan bisa dipakai	Wajib untuk operasi harian
Manual pengguna	Manual user tersedia dalam format yang mudah diakses	Gunakan bahasa yang operasional
SOP operasional	SOP login, operasional rutin, backup, eskalasi, penanganan masalah tersedia	SOP adalah inti handover
SOP insiden	SOP gangguan, keamanan, dan eskalasi tersedia	Penting untuk operasi awal
Knowledge transfer	Ada bukti handover pengetahuan, bukan hanya pelatihan satu arah	Bisa berupa sesi Q&A, demo, simulasi
Kesiapan tim internal	Tim internal dapat menjalankan tugas minimum tanpa vendor terus-menerus	Ini ukuran sukses serah terima

27.7 Checklist berita acara

Komponen	Checklist	Catatan Praktis
BAST utama	Berita acara serah terima utama tersedia dan ditandatangani pihak berwenang	Dokumen inti administrasi
BAST parsial/fase	Jika proyek bertahap, berita acara per fase tersedia	Penting untuk milestone-based contract
Daftar deliverable	Lampiran daftar hasil kerja/barang/jasa yang diserahterimakan lengkap	Harus rinci, bukan kalimat umum
Daftar pengecualian / punch list	Isu terbuka, bug minor, atau pekerjaan lanjutan yang masih tersisa dicatat	Jangan sembunyikan issue terbuka
Bukti pengujian	Hasil uji yang menjadi dasar acceptance dilampirkan	Acceptance harus tertaut ke bukti
Daftar aset / akun / lisensi	Aset, akun admin, lisensi, kredensial, repo, dokumen yang berpindah dicatat	Sangat penting untuk audit
Pernyataan pelatihan / handover	Bukti pelatihan dan knowledge transfer disebut/dirujuk	Menguatkan sisi substantif serah terima
Rujukan SLA/garansi/stabilisasi	Masa garansi, stabilisasi, dan SLA pasca serah terima dirujuk dengan jelas	Banyak sengketa muncul di sini
Status final kesiapan serah terima	Siap / siap dengan catatan / belum siap	Gunakan penilaian tegas, jangan abu-abu

Ringkasan singkat

Checklist	Fokus utama
Dokumen	kelengkapan bukti administratif, teknis, kontraktual
Keamanan	kontrol minimum, uji keamanan, bukti kepatuhan
Hak akses	akun admin, role, IAM, penutupan akses vendor
Source code	repo, kode, dokumentasi, dependensi, transisi teknis
Backup & DR	backup aktif, restore test, DR plan, PIC pemulihan
Pelatihan & SOP	kesiapan admin/user, manual, SOP, knowledge transfer
Berita acara	BAST, deliverable, punch list, bukti acceptance

Catatan pakai cepat

Kondisi	Checklist yang diprioritaskan
Serah terima software custom	**Dokumen + Source code + Hak akses + Berita acara**
Serah terima cloud/SaaS	**Hak akses + Keamanan + Backup/DR + Berita acara**
Serah terima LMS/e-commerce	**Dokumen + Keamanan + Pelatihan/SOP + Berita acara**
Serah terima managed service	**Keamanan + Hak akses + Pelatihan/SOP + Bukti layanan**
Audit internal pra-BAST	**Semua checklist**

Lampiran A. Matriks Regulasi: aturan – pasal/tema – relevansi untuk pengadaan IT

Aturan	Pasal / Tema Kunci	Relevansi untuk Pengadaan IT
Perpres 16 Tahun 2018 jo. Perpres 12 Tahun 2021 jo. Perpres 46 Tahun 2025 tentang PBJ Pemerintah	Kerangka umum PBJ, perencanaan, pemaketan, metode pemilihan, kontrak, penggunaan produk dalam negeri	Menjadi **payung utama** untuk seluruh pengadaan IT: mulai dari perencanaan kebutuhan, klasifikasi paket, metode pemilihan penyedia, sampai kontrak dan akuntabilitas hasil. Sangat relevan untuk menulis KAK, spesifikasi, HPS, evaluasi vendor, dan kontrak.
Peraturan LKPP Nomor 11 Tahun 2021	Pedoman perencanaan pengadaan	Menjadi dasar untuk **identifikasi kebutuhan, penyusunan KAK, pemaketan, jadwal, dan logika HPS**. Penting agar pengadaan IT dimulai dari masalah bisnis dan kebutuhan layanan, bukan dari daftar barang.
Peraturan LKPP Nomor 12 Tahun 2021 jo. Peraturan LKPP Nomor 4 Tahun 2024	Pedoman pelaksanaan pengadaan melalui penyedia	Relevan untuk **pemilihan penyedia, evaluasi teknis, evaluasi personel inti, rancangan kontrak, deliverable, UAT, SLA, dan serah terima**. Menjadi jembatan dari perencanaan ke delivery nyata.
Inpres 2 Tahun 2022	Percepatan penggunaan produk dalam negeri dan produk UMKM/koperasi	Sangat relevan untuk pengadaan IT yang ingin mendorong **TKDN/P3DN, jasa lokal, open source, dan penguatan ekosistem dalam negeri**. Berguna saat menyusun strategi paket dan justifikasi nilai lokal.

Perpres 95 Tahun 2018 tentang SPBE	Arsitektur SPBE, aplikasi umum/khusus, interoperabilitas, keamanan SPBE, manajemen layanan SPBE, audit TIK SPBE	Menjadi **kerangka paling penting** agar pengadaan IT pemerintah tidak berdiri sendiri. Relevan untuk arsitektur proses–data–aplikasi–infrastruktur–keamanan, integrasi layanan, penggunaan aplikasi umum, interoperabilitas, manajemen layanan, dan audit SPBE. Pasal 35 juga penting karena menyatakan pembangunan aplikasi SPBE **mengutamakan penggunaan kode sumber terbuka**.
UU Nomor 27 Tahun 2022 tentang Pelindungan Data Pribadi	Definisi data pribadi, dasar pemrosesan, pengendali dan prosesor, pemrosesan berisiko tinggi, kewajiban pengendali/prosesor	Sangat relevan untuk pengadaan sistem yang memproses **data pegawai, mahasiswa, pelanggan, pasien, peserta, pengguna layanan digital, log aktivitas, dan transaksi**. Menjadi dasar untuk peta data, privacy by design, klausul PDP di KAK dan kontrak, retensi, insiden data, dan due diligence vendor.
PP 71 Tahun 2019 tentang Penyelenggaraan Sistem dan Transaksi Elektronik	Keandalan, keamanan, tanggung jawab penyelenggara sistem elektronik	Relevan untuk pengadaan **cloud, SaaS, aplikasi publik, e-commerce, dan layanan elektronik**. Menjadi dasar untuk menuntut sistem yang andal, aman, bertanggung jawab, dan siap audit. Juga penting untuk topik backup, DR, BCP, dan penyelenggaraan layanan elektronik.
UU ITE jo. UU 1 Tahun 2024	Informasi elektronik, dokumen elektronik, transaksi elektronik, bukti elektronik	Sangat relevan untuk sistem yang menghasilkan **persetujuan digital, log, transaksi, status layanan, notifikasi, audit trail, dan bukti elektronik**. Berguna untuk e-commerce, LMS, aplikasi layanan publik, dan kontrak yang mensyaratkan jejak elektronik yang dapat dibuktikan.

Permenkominfo 5 Tahun 2020 jo. Permenkominfo 10 Tahun 2021 tentang PSE Lingkup Privat	Kewajiban PSE privat, kepatuhan penyelenggara sistem elektronik privat	Penting saat pengadaan menyentuh **cloud, SaaS, platform digital, vendor aplikasi publik, marketplace, atau sistem elektronik pihak ketiga**. Relevan untuk checklist due diligence vendor, kepatuhan cloud/SaaS, dan evaluasi layanan pihak ketiga.
PP 80 Tahun 2019 tentang PMSE	Perdagangan melalui sistem elektronik	Menjadi dasar hukum utama untuk pengadaan **e-commerce, marketplace, portal transaksi digital, kanal penjualan, dan model bisnis digital**. Relevan untuk tata kelola transaksi, pembuktian, integrasi pembayaran, pelaku usaha, dan kepatuhan platform transaksi elektronik.
Permendag 31 Tahun 2023	PMSE, marketplace, social commerce, tata kelola perdagangan digital yang lebih mutakhir	Sangat relevan untuk proyek **e-commerce/marketplace/social commerce**. Berguna untuk menulis KAK, spesifikasi, dan checklist kepatuhan PMSE agar sistem tidak hanya bagus secara teknis tetapi juga tertib secara tata kelola.
Permenaker 1 Tahun 2017	Struktur dan skala upah	Relevan untuk membaca **kewajaran biaya SDM**, terutama saat menyusun rate card, membandingkan level personel, dan membedakan antara dasar pengupahan dengan biaya jasa proyek. Berguna sebagai fondasi, tetapi tidak cukup sendiri untuk membaca pasar SDM IT.
PMK SBM Tahun 2026	Standar Biaya Masukan Tahun Anggaran 2026	Penting untuk instansi pemerintah saat menyusun **HPS, honorarium, komponen biaya pendukung, dan justifikasi biaya**. Menjadi jangkar formal yang perlu dibaca bersama benchmark pasar, khususnya untuk proyek IT yang sangat teknis.

PMK SBK yang relevan	Standar Biaya Keluaran	Relevan untuk penganggaran keluaran tertentu dan membantu pembacaan kewajaran biaya dalam konteks instansi pemerintah. Dipakai bersama HPS dan pembanding pasar, bukan sebagai satu-satunya dasar untuk proyek digital kompleks.
SKKNI / BNSP / LSP	Standar kompetensi dan sertifikasi profesi	Sangat relevan untuk menyusun syarat personel inti, sertifikat yang relevan per peran, dan model evaluasi kualitas SDM proyek IT. Membantu agar syarat personel tidak berhenti pada gelar dan CV, tetapi bergerak ke kompetensi yang lebih terstruktur.
Pedoman/audit keamanan SPBE dan keamanan siber dari BSSN	Audit keamanan, kontrol minimum, tata kelola keamanan SPBE	Relevan untuk proyek yang menyentuh **keamanan, hardening, logging, audit trail, vulnerability management, SOC/SIEM, dan kesiapan audit**. Sangat penting untuk Bab keamanan, managed service, dan serah terima sistem kritikal.

Ringkasan pakai cepat

Kebutuhan Pengadaan IT	Regulasi yang Paling Perlu Dibuka Dulu
Menyusun KAK, pemaketan, metode pemilihan, kontrak	**Perpres PBJ + PerLKPP 11/2021 + PerLKPP 12/2021/4/2024**
Menulis arsitektur, interoperabilitas, aplikasi pemerintah	**Perpres 95/2018 SPBE**
Sistem yang memproses data pribadi	**UU PDP + PP 71/2019**
Cloud, SaaS, PSE, platform digital pihak ketiga	**PP 71/2019 + Permenkominfo 5/2020 jo. 10/2021**
E-commerce, marketplace, social commerce	**PP 80/2019 + Permendag 31/2023 + UU ITE**
HPS, honor, biaya SDM	**PMK SBM/SBK + Permenaker 1/2017 + benchmark pasar**
Personel inti, sertifikasi, kompetensi	**SKKNI / BNSP / LSP**
Keamanan, audit, ketahanan operasional	**Perpres SPBE + UU PDP + PP 71/2019 + pedoman BSSN**

Catatan pakai

Prinsip	Makna Praktis
Jangan baca aturan secara terpisah-pisah	Pengadaan IT hampir selalu menyentuh lebih dari satu rezim: PBJ, SPBE, PDP, PSE, ITE, bahkan PMSE
Mulai dari fungsi proyek	Pilih aturan yang paling relevan dengan jenis sistem: hardware, software, cloud, LMS, e-commerce, managed service
Turunkan ke dokumen kerja	Setiap aturan yang penting harus tercermin dalam KAK, spesifikasi, HPS, kontrak, SLA, SOP, dan checklist audit
Utamakan akuntabilitas yang bisa dibuktikan	Regulasi paling berguna bila diterjemahkan menjadi bukti: log, dokumen, laporan, hasil uji, BAST, dan audit trail

Lampiran B. Matriks Open Source – Manfaat – Risiko – Kontrol

Area / Tema	Manfaat	Risiko	Kontrol yang Disarankan
Pengurangan lisensi	Menekan dominasi biaya lisensi dan membuka ruang anggaran untuk integrasi, pelatihan, dokumentasi, support, dan SDM lokal	Disalahpahami sebagai “gratis”, sehingga biaya implementasi dan operasi diremehkan	Selalu hitung **TCO 3–5 tahun**, termasuk implementasi, hardening, support, training, operasi, dan exit cost
Kontrol atas sistem	Organisasi punya peluang lebih besar menguasai konfigurasi, alur deployment, dan arah pengembangan	Tetap bisa terjadi ketergantungan pada integrator tertentu walau memakai OSS	Wajibkan **repo, source code, dokumentasi, skrip deployment, akun admin, dan knowledge transfer** dalam kontrak
Interoperabilitas	Lebih mudah mendorong standar terbuka, API, dan integrasi lintas sistem	OSS yang dipilih bisa tetap sulit diintegrasikan bila desain arsitektur buruk	Tetapkan kebutuhan **API, format data terbuka, dokumentasi integrasi, dan uji interoperabilitas** sejak KAK dan spesifikasi
Kedaulatan digital	Mengurangi ketergantungan yang tidak perlu pada format dan platform tertutup	Kedaulatan semu bila tim internal tidak mampu mengelola sistem	Pastikan ada **admin internal, SOP, dokumentasi, pelatihan, dan model operasi pasca go-live**

Peningkatan nilai lokal	Porsi belanja bisa lebih banyak jatuh ke jasa lokal: integrasi, support, pelatihan, dokumentasi, dan operasi	Nilai lokal hanya menjadi klaim, tidak benar-benar terukur	Buat **matriks nilai lokal**: porsi jam kerja lokal, porsi biaya jasa lokal, porsi support lokal, dan transfer pengetahuan
Fleksibilitas pengembangan	Lebih mudah menyesuaikan sistem dengan kebutuhan spesifik organisasi	Perubahan liar tanpa disiplin arsitektur dapat membuat sistem rapuh dan sulit dipelihara	Terapkan **governance perubahan, version control, coding standard, code review, dan acceptance test**
Kecepatan adopsi	Solusi OSS matang bisa mempercepat implementasi dibanding membangun dari nol	Organisasi tergoda memilih produk populer tanpa uji kecocokan	Lakukan **fit-gap analysis**: fungsi, integrasi, keamanan, kapasitas, dan kesiapan operasi sebelum memilih platform
Keamanan	Kode yang lebih terbuka bisa membantu audit, transparansi dependensi, dan pengurangan “black box”	Kerentanan OSS bisa diabaikan bila tidak ada patching dan dependency management	Wajibkan **vulnerability management, patching, daftar dependensi, security test/pentest, dan logging**
Privasi dan data	OSS memberi peluang kontrol lebih kuat atas lokasi, alur, dan pemrosesan data	Implementasi buruk tetap bisa melanggar PDP	Terapkan **privacy by design**, peta data pribadi, kontrol akses, retensi, logging, dan klausul PDP di kontrak

Cloud berbasis OSS	Bisa mengurangi ketergantungan pada stack proprietary tertentu	Bisa tetap terjebak lock-in jika operasinya sangat vendor-specific	Siapkan **portabilitas data, dokumentasi konfigurasi, backup independen, dan exit plan**
LMS berbasis OSS	Cocok untuk kontrol pembelajaran, modifikasi, dan penguatan admin lokal	Bisa berat di operasi jika kapasitas support internal lemah	Wajibkan **pelatihan admin, dokumentasi, uji beban, support operasional, dan integrasi akademik**
E-commerce berbasis OSS	Membuka ruang modifikasi proses bisnis dan kontrol transaksi yang lebih besar	Risiko keamanan transaksi, plugin tidak aman, dan maintenance tinggi	Terapkan **hardening, review plugin/dependensi, uji keamanan, log transaksi, fraud control, dan SOP update**
Managed service OSS	Memungkinkan operasi layanan tanpa lisensi platform yang dominan	Vendor dapat “mengunci” operasi lewat pengetahuan, bukan lisensi	Tulis **SLA, bukti layanan, audit trail, SOP, handover, dan exit management** secara tegas
Biaya SDM	Biaya lebih banyak masuk ke kompetensi nyata dan delivery	HPS bisa terlalu rendah karena OSS dianggap murah	Gunakan **rate card per peran**, benchmark pasar, dan HPS jasa end-to-end, bukan asumsi “tanpa lisensi = murah”
Sertifikasi dan SDM	Mendorong pembacaan kompetensi nyata, bukan hanya sertifikasi vendor tertentu	Tim dinilai terlalu longgar karena tidak ada sertifikat vendor khusus	Gunakan kombinasi **portofolio, pengalaman relevan, kompetensi inti, dan sertifikasi yang relevan per peran**

Audit dan akuntabilitas	OSS yang dikelola baik dapat lebih siap audit karena artefaknya lengkap dan terbuka	Jika dokumentasi lemah, audit justru lebih sulit	Pastikan ada **repo, manual, change log, hasil uji, BAST, audit trail, dan bukti acceptance**
Keberlanjutan jangka panjang	Lebih mudah memperpanjang umur sistem bila organisasi menguasai aset teknisnya	Proyek berhenti ketika komunitas lemah atau dukungan komersial tidak ada	Nilai **maturitas komunitas/dukungan komersial**, roadmap, dokumentasi, dan kemampuan mencari integrator pengganti
Pengadaan non-diskriminatif	Mendukung spesifikasi berbasis fungsi dan hasil, bukan merek	OSS bisa disalahgunakan sebagai preferensi ideologis yang juga menutup pasar	Tuliskan kebutuhan dalam bahasa **fungsi, performa, interoperabilitas, keamanan, dan outcome**, bukan label produk
Keputusan strategis	OSS memberi ruang lebih luas untuk desain solusi yang pragmatis dan berdaulat	Tidak selalu menjadi pilihan terbaik untuk semua kasus	Gunakan **matriks keputusan**: OSS penuh, hybrid, atau proprietary terbatas, berdasarkan fungsi, risiko, SDM, dan TCO

Ringkasan pakai cepat

Kondisi	Posisi yang Paling Sehat
Kebutuhan umum, integrasi penting, dukungan pasar cukup	**Prioritaskan OSS atau hybrid**
Kebutuhan sangat spesifik, pasar OSS belum matang	**Evaluasi proprietary terbatas dengan exit plan kuat**
Organisasi ingin memperbesar nilai lokal dan jasa domestik	**Gunakan OSS/hybrid + integrator lokal + deliverable lengkap**
Tim internal lemah dan dokumentasi buruk	**Jangan memilih OSS hanya karena murah**
Sistem memproses data penting/pribadi	**OSS boleh dipakai, tetapi kontrol keamanan dan PDP harus diperketat**

Catatan pakai

Prinsip	Makna Praktis
OSS bukan kewajiban umum	Ia adalah **strategi pengadaan** yang sering sangat berguna, bukan perintah tunggal untuk semua proyek
Fokus ke outcome	Ukur dari interoperabilitas, kontrol deliverable, nilai lokal, TCO, dan keberlanjutan
Jangan romantis	OSS tetap butuh biaya, SDM, support, keamanan, dan tata kelola
Jangan alergi hybrid	Banyak proyek paling sehat justru memakai **perangkat komersial + stack OSS + jasa lokal**

Lampiran C. Matriks TKDN/P3DN untuk Proyek IT

Area / Komponen Proyek IT	Cara Membaca TKDN/P3DN	Implikasi Praktis untuk Pengadaan IT
Perangkat keras (server, storage, endpoint, jaringan)	TKDN paling mudah dibaca pada komponen **barang fisik** yang memang memiliki struktur manufaktur, perakitan, atau kandungan lokal yang dapat diukur formal	Untuk pengadaan hardware, cek apakah ada **produk dalam negeri/substitusi lokal** yang relevan, lalu pastikan spesifikasi tetap berbasis fungsi, performa, dan kompatibilitas, bukan sekadar label lokal
Perangkat jaringan & keamanan	Tidak semua perangkat jaringan/firewall/switch punya alternatif lokal yang setara pada semua kelas kebutuhan	Lakukan **fit-gap**: bila ada opsi lokal yang memenuhi kebutuhan layanan, prioritaskan; bila belum, justifikasi kebutuhan teknis dan arahkan nilai lokal ke jasa integrasi, support, dan operasi
Perangkat pendukung (UPS, rack, kabel, aksesoris, enclosure)	Komponen pendukung sering justru punya peluang lebih besar untuk kandungan lokal tinggi	Saat menyusun paket, jangan fokus hanya pada item utama; komponen pendukung bisa menjadi ruang cepat untuk mendorong P3DN
Software proprietary impor	Pada software proprietary, porsi nilai sering bocor ke **lisensi asing** dan subscription berulang	Dalam proyek yang sangat bergantung pada lisensi asing, pembeli perlu jujur membaca bahwa kontribusi P3DN formal bisa rendah, lalu menyeimbangkannya dengan jasa lokal, integrasi, pelatihan, dan support

Software open source	OSS tidak otomatis berarti TKDN formal tinggi, tetapi sering sangat baik untuk **memperbesar porsi nilai lokal** pada jasa implementasi, integrasi, pelatihan, dokumentasi, dan operasi	Gunakan OSS/hybrid bila tujuannya adalah **mengurangi porsi lisensi asing** dan memperbesar belanja yang jatuh ke SDM dan perusahaan lokal
Pengembangan aplikasi custom	Pada aplikasi custom, nilai lokal lebih sering berada pada **jasa SDM, analisis, desain, coding, QA, support, dan operasi** daripada pada “produk jadi”	Untuk proyek software custom, bacalah P3DN lewat **struktur biaya jasa lokal**, bukan hanya bertanya apakah aplikasinya “lokal” atau “tidak”
Integrasi sistem	Integrasi adalah area yang sangat kuat untuk membangun nilai lokal karena bertumpu pada pemahaman proses, data, API, middleware, dan operasi	Saat memaketkan proyek, pastikan integrasi tidak tenggelam di balik lisensi; justru di sinilah jasa lokal bisa menjadi inti nilai proyek
Migrasi data	Migrasi data hampir selalu berbasis jasa dan pengetahuan lokal atas data, proses, dan sistem lama	Masukkan migrasi data sebagai komponen eksplisit dalam KAK, HPS, dan evaluasi agar nilainya terlihat sebagai bagian dari P3DN
Managed service / operasi TI	Nilai lokal pada operasi TI sangat besar karena bertumpu pada helpdesk, NOC, SOC, patching, backup, monitoring, dan dukungan pengguna	Untuk layanan operasi, ukuran P3DN yang lebih relevan adalah **berapa porsi layanan dikelola lokal**, bukan semata produk yang dipakai
Cloud / SaaS	Cloud/SaaS sering menyulitkan pembacaan TKDN formal bila platformnya milik pihak luar, tetapi masih membuka ruang besar untuk jasa lokal	Jika memakai cloud/SaaS asing, kompensasikan dengan **integrator lokal, support lokal, tata kelola lokal, pelatihan lokal, dan portabilitas data**

LMS / e-learning	LMS dapat dibaca melalui dua lapisan: platform dan jasa pendukung	Nilai lokal pada LMS sering lebih kuat di **migrasi konten, pelatihan admin/pengajar, dukungan operasional, integrasi akademik, dan helpdesk** daripada pada platform semata
E-commerce / marketplace / PMSE	Pada sistem transaksi digital, nilai lokal tersebar di pengembangan, operasi, payment integration, fraud control, dan support	Jangan hanya membaca P3DN dari platform inti; baca juga **merchant onboarding, support, rekonsiliasi, log, helpdesk, dan operasi lokal**
Keamanan siber	Layanan keamanan sering bertumpu pada jasa audit, pentest, monitoring, SOC, IR, dan forensik	Ini area kuat untuk membangun **kapasitas nasional** melalui tenaga ahli lokal, meski alat atau platform tertentu mungkin masih impor
Pelatihan dan knowledge transfer	Pelatihan sering dianggap pelengkap, padahal justru sangat penting dalam logika P3DN jangka panjang	Masukkan pelatihan, SOP, dokumentasi, dan transfer pengetahuan sebagai komponen wajib agar belanja proyek meninggalkan **kapasitas lokal**, bukan hanya sistem aktif
Dokumentasi dan SOP	Nilai lokal yang kuat sering lahir dari dokumentasi operasional, panduan administrasi, dan SOP yang benar-benar dipakai institusi	Wajibkan deliverable dokumentasi agar hasil proyek dapat diteruskan oleh tim lokal tanpa tergantung penuh pada vendor

SDM lokal per peran	Pada proyek digital, ukuran P3DN paling jujur sering ada pada **jam kerja, peran, dan biaya yang jatuh ke SDM lokal**	Buat matriks peran: PM, BA, architect, developer, QA, security, DevOps, support, trainer—lalu lihat berapa proporsi yang benar-benar dikerjakan lokal
Subkontraktor dan rantai pasok	Vendor utama bisa lokal, tetapi nilai proyek belum tentu tinggal di lokal bila sebagian besar pekerjaan kunci dialihkan keluar	Saat due diligence vendor, minta kejelasan **fungsi mana dikerjakan sendiri, mana yang disubkontrakkan, dan ke mana nilai proyek mengalir**
Paket hybrid (komersial + OSS + jasa lokal)	Banyak proyek IT paling sehat justru berada pada model hybrid: sebagian komponen komersial, sebagian OSS, dan porsi besar jasa lokal	Saat membaca TKDN/P3DN, jangan memaksa hitam-putih; gunakan model hybrid untuk menyeimbangkan **keandalan, biaya, dan nilai lokal**
Nilai lokal non-sertifikat	Tidak semua kontribusi lokal hadir dalam bentuk sertifikat TKDN formal	Untuk proyek IT, organisasi perlu membaca juga **nilai lokal ekonomis nyata**: jasa, transfer knowledge, operasi, support, dan kapasitas institusi
Produk dalam negeri vs hasil layanan	Produk dalam negeri penting, tetapi dalam IT hasil layanan sering sama pentingnya	Pengadaan yang sehat tidak berhenti pada “beli produk lokal”, tetapi juga bertanya: **apakah proyek ini meninggalkan kemampuan lokal yang bisa bertahan?**

Matriks baca cepat: komponen – peluang TKDN/P3DN – strategi paket

Komponen	Peluang TKDN/P3DN	Strategi Paket yang Disarankan
Hardware utama	Sedang–tinggi, tergantung kelas produk	Pisahkan komponen fisik secara jelas, cek substitusi lokal, tetap jaga spesifikasi non-diskriminatif
Software proprietary	Rendah pada lisensi, lebih tinggi pada jasa	Tekan dominasi lisensi, perbesar porsi integrasi, pelatihan, support lokal
Software OSS	Tinggi pada jasa dan penguasaan lokal	Gunakan untuk proyek yang ingin memperbesar nilai lokal dan kontrol teknis
Integrasi & migrasi	Tinggi	Jadikan komponen eksplisit, jangan tenggelam di bawah lisensi
Managed service	Tinggi	Fokus pada SLA, SDM lokal, audit trail, dan transfer knowledge
Cloud/SaaS	Rendah–sedang pada platform, tinggi pada jasa pendamping	Wajibkan support lokal, governance lokal, exit plan, dan dokumentasi
Keamanan	Sedang–tinggi	Perbesar porsi audit, monitoring, IR, dan pentest oleh tim lokal yang kompeten
Pelatihan & SOP	Tinggi	Wajibkan deliverable pelatihan dan SOP agar manfaat proyek tinggal di institusi

Indikator praktis P3DN untuk proyek IT

Indikator	Cara Membaca	Catatan
Persentase biaya jasa lokal	Berapa bagian dari total proyek jatuh ke jasa lokal	Sangat penting untuk software, integrasi, dan managed service
Persentase jam kerja lokal	Berapa banyak pekerjaan inti dilakukan oleh SDM lokal	Lebih jujur daripada hanya melihat nama vendor
Porsi support lokal	Apakah dukungan harian/operasional benar-benar dikerjakan lokal	Penting untuk keberlanjutan
Porsi transfer pengetahuan	Ada tidaknya pelatihan, SOP, dokumentasi, handover	Nilai lokal jangka panjang
Porsi kendali teknis internal	Apakah admin, repo, konfigurasi, dan dokumentasi bisa dikuasai institusi	Mengukur kedaulatan operasional
Porsi komponen fisik lokal	Relevan terutama untuk hardware	Lebih mudah dibaca formal melalui TKDN

Catatan pakai

Prinsip	Makna Praktis
Jangan membaca TKDN/P3DN IT terlalu sempit	Dalam proyek digital, nilai lokal sering lebih besar pada **jasa, operasi, integrasi, dan knowledge transfer** daripada pada produk fisik
Pisahkan barang dan jasa secara jujur	Ini membantu membaca mana kontribusi lokal formal dan mana kontribusi lokal ekonomis nyata
Gunakan OSS/hybrid secara strategis	Bukan karena ideologi, tetapi karena sering lebih efektif untuk memperbesar ruang jasa lokal dan penguasaan sistem
Ukur, jangan hanya klaim	Matriks P3DN IT harus dibaca lewat **angka, jam kerja, deliverable, dan kontrol operasional**, bukan hanya narasi vendor

Ringkasan singkat

Fokus	Pesan Utama
TKDN formal	Paling kuat pada barang fisik
P3DN ekonomis nyata	Paling kuat pada jasa, integrasi, support, pelatihan, dan operasi
Open source / hybrid	Sering menjadi alat terbaik untuk memperbesar nilai lokal
Due diligence vendor	Wajib untuk memastikan nilai proyek benar-benar tinggal di dalam negeri

Lampiran D. Matriks Peran SDM – Sertifikasi – Pengalaman – Output

Peran SDM	Sertifikasi / Kompetensi yang Relevan	Pengalaman Minimum yang Masuk Akal + Output yang Diharapkan
Project Manager (PM)	Sertifikasi manajemen proyek yang relevan dapat menjadi nilai tambah; kompetensi utama tetap ada pada pengelolaan ruang lingkup, jadwal, risiko, koordinasi, dan komunikasi	**Pengalaman:** memimpin proyek IT dengan kompleksitas sebanding. **Output:** project plan, timeline, risk register, issue log, status report, notulen koordinasi, kendali change request
Business Analyst (BA)	Kompetensi analisis kebutuhan, pemetaan proses bisnis, fasilitasi workshop, dan penyusunan requirement lebih penting daripada sekadar label sertifikat	**Pengalaman:** terlibat pada analisis kebutuhan sistem/proses sejenis. **Output:** dokumen kebutuhan bisnis, proses *as-is / to-be*, user story/use case, matriks kebutuhan pengguna
System Analyst	Kompetensi analisis sistem, desain logika aplikasi, data flow, integrasi, dan spesifikasi fungsional	**Pengalaman:** merancang sistem dengan integrasi atau modul sejenis. **Output:** spesifikasi fungsional, desain sistem, data flow, mapping integrasi, bahan untuk dev & QA
Solution / Enterprise Architect	Kompetensi arsitektur aplikasi, data, infrastruktur, keamanan, interoperabilitas; sertifikasi arsitektur relevan bila memang dibutuhkan	**Pengalaman:** merancang arsitektur sistem lintas modul/sistem. **Output:** dokumen arsitektur, integrasi, standar desain, keputusan teknis utama, kontrol konsistensi solusi
Software Developer / Programmer	Kompetensi coding, framework, version control, debugging, secure coding; sertifikasi dapat mendukung tetapi **portofolio biasanya lebih penting**	**Pengalaman:** membangun modul/aplikasi sejenis, bukan hanya tahun kerja umum. **Output:** source code, unit/module yang berjalan, commit repo, technical note, fix bug

Mobile Developer	Kompetensi Android/iOS/cross-platform, integrasi API, manajemen rilis aplikasi, notifikasi, optimasi perangkat	**Pengalaman:** membangun aplikasi mobile yang sudah dipakai pengguna nyata. **Output:** aplikasi mobile, build/release, dokumentasi teknis, integrasi API, support perbaikan versi
Frontend Developer / UI Engineer	Kompetensi antarmuka, responsive design, aksesibilitas, integrasi frontend-backend, performa UI	**Pengalaman:** membangun UI untuk aplikasi dengan beban dan profil user sejenis. **Output:** layar/halaman, style guide sederhana, komponen UI, hasil integrasi frontend
Backend Developer	Kompetensi API, business logic, database interaction, security dasar, performance tuning	**Pengalaman:** membangun service/backend untuk transaksi atau layanan internal/publik. **Output:** API/service, modul backend, dokumentasi endpoint, integrasi data
QA / Tester	Kompetensi test case, test scenario, defect management, regression testing, UAT support; sertifikasi testing dapat membantu bila relevan	**Pengalaman:** pengujian sistem dengan modul/proses sejenis. **Output:** test plan, test case, bug list, hasil SIT/UAT, laporan pengujian
QA Automation	Kompetensi automation testing, script test, CI/CD integration, regression automation	**Pengalaman:** membuat dan memelihara automated test untuk sistem nyata. **Output:** automated test suite, laporan hasil automation, integrasi test ke pipeline
DevOps / Release Engineer	Kompetensi CI/CD, deployment, environment management, observability, scripting, container/orchestration bila relevan	**Pengalaman:** menangani deployment dan operasi aplikasi pada lingkungan serupa. **Output:** pipeline build/deploy, skrip deployment, dokumentasi environment, release note
System Administrator	Kompetensi administrasi server, OS, user access, backup dasar, monitoring, hardening dasar	**Pengalaman:** mengelola server/VM pada skala yang relevan. **Output:** konfigurasi server, SOP admin, patching record, akun/akses, bukti backup dasar

Database Administrator (DBA)	Kompetensi desain basis data, tuning, backup/restore, akses, migrasi, integritas data	**Pengalaman:** mengelola database dengan ukuran/beban yang relevan. **Output:** skema database, backup strategy, tuning note, migration script, restore evidence
Network Engineer	Kompetensi switching, routing, VLAN, wireless, segmentasi, troubleshooting jaringan, dokumentasi topologi	**Pengalaman:** implementasi/operasi jaringan dengan topologi sebanding. **Output:** topologi jaringan, konfigurasi perangkat, hasil uji konektivitas, SOP dasar jaringan
Security Engineer / Security Analyst	Sertifikasi keamanan yang relevan dapat bernilai tambah; kompetensi inti: hardening, vulnerability management, logging, IAM, incident support	**Pengalaman:** menangani kontrol keamanan pada sistem operasional. **Output:** baseline hardening, vulnerability report, rekomendasi kontrol, review akses, log/security checklist
Security Auditor / Pentester	Sertifikasi audit keamanan/pentest dapat relevan; yang penting adalah kompetensi audit, verifikasi kontrol, dan penulisan temuan	**Pengalaman:** audit atau pentest pada sistem dengan profil risiko serupa. **Output:** laporan audit/pentest, temuan, severity, rekomendasi perbaikan, re-test result
SOC Analyst / Incident Responder	Kompetensi monitoring keamanan, triage insiden, log analysis, escalation, incident documentation	**Pengalaman:** operasi SOC/IR atau monitoring keamanan layanan nyata. **Output:** incident log, escalation record, RCA awal, notifikasi insiden, rekomendasi containment
Cloud Engineer / Cloud Architect	Sertifikasi cloud relevan bila proyek memang memakai cloud tertentu; kompetensi inti: IAM, network, compute, storage, security, cost awareness	**Pengalaman:** implementasi/operasi cloud pada model IaaS/PaaS/SaaS yang relevan. **Output:** arsitektur cloud, environment setup, IAM policy, backup policy, runbook
Data Engineer	Kompetensi pipeline data, ETL/ELT, integrasi data, kualitas data, data model	**Pengalaman:** membangun alur data pada proyek dengan volume/kompleksitas mirip. **Output:** pipeline data, mapping data, dokumentasi alur, monitoring kualitas data

Data Analyst / BI Analyst	Kompetensi analitik, dashboard, visualisasi, definisi KPI, data interpretation	**Pengalaman:** membuat dashboard/laporan untuk proses bisnis relevan. **Output:** dashboard, KPI definition, laporan analitik, dokumentasi data source
Trainer / Learning Facilitator	Kompetensi pelatihan, penyusunan materi, fasilitasi user/admin, evaluasi pemahaman	**Pengalaman:** melatih admin/user pada sistem atau platform digital sejenis. **Output:** materi pelatihan, daftar hadir, rekaman/sesi, FAQ, evaluasi pelatihan
Helpdesk / Support Engineer	Kompetensi support pengguna, ticket handling, troubleshooting dasar, eskalasi, dokumentasi layanan	**Pengalaman:** support aplikasi/infrastruktur dengan beban user relevan. **Output:** tiket layanan, status penyelesaian, FAQ, laporan support, escalation log
LMS Administrator	Kompetensi manajemen kelas, user/course, kuis/ujian, bank soal, pelaporan, support pengajar/peserta	**Pengalaman:** mengelola LMS atau platform belajar digital. **Output:** kelas aktif, akun terkelola, laporan penggunaan, SOP LMS, support issue log
E-commerce / Marketplace Operations Lead	Kompetensi operasi transaksi, katalog, order flow, payment reconciliation, complaint handling, fraud awareness	**Pengalaman:** mengelola sistem transaksi digital/marketplace. **Output:** SOP operasi transaksi, dashboard status, rekonsiliasi, log komplain, perbaikan proses
Product Owner / Service Owner	Kompetensi prioritisasi backlog, pemahaman proses bisnis, pengambilan keputusan layanan, validasi output	**Pengalaman:** memegang keputusan prioritas pada proyek digital yang sejenis
Technical Writer / Documentation Specialist	Kompetensi dokumentasi teknis, manual, SOP, struktur informasi	**Pengalaman:** menulis dokumentasi sistem/operasi yang benar-benar dipakai
Auditor / Reviewer Kepatuhan	Kompetensi audit internal/eksternal, review kontrak, review bukti layanan, review kontrol	**Pengalaman:** audit atau review proyek/layanan digital sejenis

Matriks pakai cepat: peran – bukti utama yang paling meyakinkan

Peran	Bukti yang Paling Meyakinkan
PM	pengalaman memimpin proyek serupa + contoh status report/risk handling
BA / System Analyst	contoh requirement/process mapping/spec dokumen
Architect	contoh arsitektur, integrasi, keputusan desain
Developer	portofolio, repo, modul/aplikasi yang pernah dibangun
QA	test plan, test case, defect report
Security	baseline hardening, vulnerability report, audit/pentest report
Cloud/DevOps	pipeline, environment setup, runbook, deployment evidence
Support/Helpdesk	ticket report, SOP support, escalation handling
Trainer	materi pelatihan, catatan sesi, panduan user

Contoh cara membaca matriks ini

Situasi Pengadaan	Cara Memakai Matriks
Pengadaan software custom	Fokus pada PM, BA, System Analyst, Developer, QA, DevOps, Security, Trainer
Pengadaan cloud/SaaS	Fokus pada Cloud Engineer/Architect, Security, Support, Service Owner
Pengadaan LMS	Fokus pada BA, Developer/Configurator, LMS Administrator, Trainer, Support
Pengadaan e-commerce	Fokus pada BA, Developer, Security, Data/BI, Ops Lead, Support
Managed service / operasi TI	Fokus pada Support Engineer, NOC/SOC role, System Admin, Security, Service Owner

Catatan pakai

Prinsip	Makna Praktis
Jangan minta semua sertifikat untuk semua peran	Sertifikat harus relevan dengan fungsi nyata peran itu
Portofolio lebih penting daripada CV saja	Terutama untuk developer, analyst, QA, DevOps, dan security
Pengalaman harus relevan, bukan sekadar panjang	3 tahun yang relevan bisa lebih kuat daripada 8 tahun yang tidak nyambung
Output harus bisa ditagih	Setiap peran harus dikaitkan dengan deliverable atau fungsi yang dapat dibuktikan
Sesuaikan dengan skala proyek	Proyek kecil boleh merangkap, tetapi fungsi penting tidak boleh hilang

Ringkasan singkat

Fokus	Pesan Utama
Peran	Tentukan berdasarkan fungsi proyek, bukan jabatan generik
Sertifikasi	Gunakan sebagai pendukung kompetensi, bukan formalitas
Pengalaman	Nilai relevansi terhadap sistem, skala, dan risiko
Output	Jadikan dasar evaluasi dan dasar pembayaran/acceptance peran

Lampiran E. Contoh Perhitungan HPS Proyek *Software Open Source*

Asumsi proyek contoh

Komponen	Contoh Isi	Catatan Praktis
Jenis proyek	Implementasi aplikasi layanan internal berbasis *open source*	Bisa untuk sistem administrasi, layanan internal, atau dashboard operasional
Model solusi	*Adopt OSS* + konfigurasi + integrasi + migrasi + pelatihan + stabilisasi	Menunjukkan bahwa OSS tetap butuh jasa nyata
Jumlah pengguna	300 pengguna, 20 admin/operator	Penting untuk skala implementasi dan support
Lingkungan	1 produksi, 1 uji/staging	Minimal untuk proyek yang sehat
Durasi proyek awal	4 bulan implementasi + 2 bulan stabilisasi	Pisahkan implementasi dan stabilisasi
Hosting	On-premise atau cloud yang sudah tersedia terpisah	Di contoh ini fokus ke HPS software/jasa, bukan infrastruktur
Hak hasil kerja	Konfigurasi, dokumentasi, repo perubahan, SOP, manual diserahkan ke instansi	Sangat penting untuk proyek OSS
Pendekatan biaya	Biaya utama ada pada **SDM, integrasi, migrasi, pelatihan, keamanan, dan dokumentasi**, bukan lisensi	Ini inti perbedaan HPS OSS vs proprietary

Struktur HPS proyek OSS: komponen utama

Komponen	Contoh Perhitungan	Catatan Praktis
Lisensi software inti	Rp0	Rp0 **bukan berarti proyek murah**
Analisis & desain	Dihitung sebagai jasa SDM per peran	Wajib untuk implementasi yang sehat
Konfigurasi / kustomisasi OSS	Dihitung sebagai jasa SDM	Jangan diasumsikan “tinggal install”
Integrasi	Dihitung sebagai jasa SDM + uji integrasi	Sering jadi komponen besar
Migrasi data	Dihitung sebagai jasa SDM + validasi data	Sering diremehkan
Keamanan & hardening	Dihitung sebagai jasa SDM + uji keamanan dasar	Wajib untuk sistem operasional
Dokumentasi	Dihitung sebagai output wajib	Harus muncul di HPS, bukan bonus
Pelatihan admin/user kunci	Dihitung sebagai jasa + materi	Penting untuk nilai lokal dan keberlanjutan
Stabilisasi pasca go-live	Dihitung sebagai support awal	Sangat penting agar sistem tidak rapuh
Support tahun pertama	Bisa dihitung terpisah bila diadakan dalam paket yang sama	Jangan tercampur tanpa kejelasan
Pajak	Tambahkan sesuai struktur biaya	Jelaskan bruto/neto

Contoh *rate card* SDM proyek OSS

Peran	Contoh Tarif dan Volume	Subtotal Contoh
Project Manager	Rp18.000.000/bulan × 4 bulan × 50% keterlibatan	**Rp36.000.000**
Business Analyst	Rp16.000.000/bulan × 2 bulan × 100%	**Rp32.000.000**
System Analyst / Solution Analyst	Rp18.000.000/bulan × 2 bulan × 100%	**Rp36.000.000**
Developer / Implementor OSS	Rp15.000.000/bulan × 4 bulan × 2 orang	**Rp120.000.000**
QA / Tester	Rp12.000.000/bulan × 2 bulan × 100%	**Rp24.000.000**
DevOps / Deployment Engineer	Rp17.000.000/bulan × 1 bulan × 100%	**Rp17.000.000**
Security Engineer	Rp20.000.000/bulan × 0,5 bulan × 100%	**Rp10.000.000**
Trainer / Documentation Specialist	Rp12.000.000/bulan × 1 bulan × 100%	**Rp12.000.000**
Support / Stabilisasi	Rp10.000.000/bulan × 2 bulan × 100%	**Rp20.000.000**
Subtotal SDM		**Rp307.000.000**

Catatan: angka ini hanya **contoh struktur hitung**, bukan tarif baku nasional. Saat dipakai riil, sesuaikan dengan **SBM/SBK, benchmark pasar, kompleksitas, dan lokasi**. Pendekatan per-peran seperti ini konsisten dengan Bab 9, Bab 16, dan Bab 18.

Contoh komponen non-SDM

Komponen Non-SDM	Contoh Nilai	Catatan Praktis
Workshop kebutuhan & validasi	Rp7.500.000	Bisa mencakup materi, koordinasi, dokumentasi sesi
Migrasi data (alat bantu, validasi, cleaning support)	Rp12.500.000	Di luar SDM inti bila memang ada komponen tambahan
Pelatihan admin dan user kunci	Rp8.000.000	Materi, modul, sesi pelatihan
Dokumentasi final & packaging deliverable	Rp5.000.000	Bila tidak sepenuhnya masuk tarif SDM
Uji keamanan dasar / scanning	Rp7.500.000	Minimal untuk baseline keamanan
Cadangan biaya operasional kecil implementasi	Rp5.000.000	Harus tetap wajar dan dijustifikasi
Subtotal Non-SDM	**Rp45.500.000**	

Contoh rekap HPS implementasi awal OSS

Komponen	Nilai
Subtotal SDM	Rp 307.000.000
Subtotal Non-SDM	Rp 45.500.000
Subtotal sebelum pajak	**Rp 352.500.000**
PPN 11%	**Rp 38.775.000**
Total HPS contoh	**Rp 391.275.000**

Bacaan penting: pada proyek OSS ini, **lisensi inti = Rp0**, tetapi total HPS tetap hampir **Rp400 juta** karena nilai proyek terletak pada **analisis, konfigurasi, integrasi, migrasi, keamanan, dokumentasi, pelatihan, dan stabilisasi**. Inilah logika yang harus dijelaskan kepada pimpinan atau auditor ketika ada pertanyaan “kalau open source, kenapa tetap mahal?”

Contoh TCO 3 tahun untuk proyek OSS

Komponen TCO	Tahun 1	Tahun 2	Tahun 3	Catatan
Implementasi awal	Rp 391.275.000	Rp 0	Rp 0	Sekali di awal
Support & maintenance tahunan	Rp 60.000.000	Rp 60.000.000	Rp 60.000.000	Bisa oleh vendor lokal / tim internal + vendor
Patching / hardening / audit ringan	Rp 15.000.000	Rp 15.000.000	Rp 15.000.000	Penting untuk keberlanjutan
Pelatihan lanjutan / refresh admin	Rp 7.500.000	Rp 7.500.000	Rp 7.500.000	Untuk jaga kapasitas internal
Pengembangan kecil / penyesuaian minor	Rp 20.000.000	Rp 20.000.000	Rp 20.000.000	Jangan diasumsikan nol
Total per tahun	**Rp 493.775.000**	**Rp102.500.000**	**Rp102.500.000**	
Total TCO 3 tahun			**Rp698.775.000**	

Pelajaran utamanya: biaya OSS sehat dibaca lewat **TCO**, bukan harga lisensi masuk. Setelah tahun pertama, biaya biasanya turun tajam, tetapi tetap ada biaya **support, keamanan, pelatihan, dan perubahan minor**. Ini konsisten dengan logika Bab 9 tentang TCO proyek IT.

Perbandingan ringkas: OSS vs proprietary

Komponen	OSS	Proprietary
Lisensi awal	Rendah / nol	Biasanya tinggi
Biaya implementasi	Sedang–tinggi	Sedang–tinggi
Biaya integrasi	Tetap ada	Tetap ada
Biaya dokumentasi & knowledge transfer	Harus kuat	Harus kuat
Biaya support tahunan	Ada	Ada
Risiko lock-in	Lebih rendah bila repo, data, dan SOP dikuasai	Bisa tinggi bila lisensi, data, dan konfigurasi tertutup
Ruang jasa lokal	Biasanya lebih besar	Sering lebih kecil bila dominan lisensi
Bentuk nilai proyek	Lebih banyak ke **SDM & kontrol teknis**	Lebih banyak ke **hak pakai/lisensi + vendor support**

Template cepat isi HPS proyek OSS

Komponen	Isian Template	Catatan
Nama sistem	…	Sebut sistem/layanan
Platform OSS yang diadopsi	…	Sebut nama platform bila perlu, atau “platform OSS sejenis”
Jumlah user/admin	…	Dasar kapasitas dan pelatihan
Lingkup implementasi	analisis / konfigurasi / integrasi / migrasi / training / stabilisasi	Pilih sesuai proyek
Peran SDM	PM / BA / SA / Dev / QA / DevOps / Security / Trainer / Support	Sesuaikan kebutuhan
Tarif per peran	…	Dasarkan pada benchmark + kewajaran
Durasi per peran	…	Hari/bulan/sprint
Komponen non-SDM	workshop / migrasi / pelatihan / uji keamanan / dokumentasi	Jangan dibiarkan kosong
Pajak	…	Jelaskan bruto/neto
Total HPS	…	Harus dapat dijelaskan logikanya
TCO 3 tahun	…	Minimal dibuat untuk pembacaan strategis

Ringkasan singkat

Fokus	Pesan Utama
Lisensi OSS	bisa nol
Biaya nyata proyek OSS	ada pada SDM, integrasi, migrasi, keamanan, dokumentasi, support
HPS OSS yang sehat	harus per-peran, per-output, dan berbasis TCO
Nilai lokal	biasanya lebih besar karena porsi jasa dan penguasaan teknis lebih tinggi

Catatan pakai

Prinsip	Makna Praktis
Jangan jual OSS sebagai proyek murah	jual sebagai proyek yang **lebih sehat struktur nilainya**
Pisahkan implementasi awal dan operasi tahunan	agar HPS dan TCO terbaca jernih
Dokumentasi dan handover wajib dihitung	bukan bonus vendor
Perkuat justifikasi ke auditor/pimpinan	jelaskan bahwa biaya OSS berpindah dari lisensi ke **kompetensi, integrasi, dan kontrol hasil**

Lampiran F. Contoh Perhitungan HPS Managed Service

HPS *managed service* jangan dihitung seperti beli barang.
Yang dibeli adalah **layanan operasional**: orang yang menjaga sistem, merespons gangguan, membuat laporan, dan memastikan layanan tetap jalan.

Contoh kasus sederhana

Misalnya sebuah instansi butuh layanan:

- *helpdesk* untuk pengguna
- monitoring aplikasi dan server
- pengecekan backup
- patching rutin
- bantuan saat ada gangguan
- laporan bulanan

Layanannya berlaku:

- **12 bulan**
- **jam kerja**: Senin–Jumat, pukul 08.00–17.00
- ada **on-call terbatas** kalau ada masalah penting di luar jam kerja

1. Hitung biaya orangnya

Misalnya dibutuhkan 3 peran:

a. Support Engineer

Tugasnya:

- menerima tiket
- membantu user
- mencatat gangguan
- eskalasi ke tim teknis

Biaya:

- Rp12.000.000 per bulan
- 12 bulan

Perhitungan:
Rp12.000.000 × 12 = Rp144.000.000

b. Ops / System Administrator

Tugasnya:

- cek server
- cek backup
- patching
- bantu masalah teknis

Biaya:

- Rp16.000.000 per bulan
- dipakai 50%
- 12 bulan

Perhitungan:
Rp16.000.000 × 12 × 50% = Rp96.000.000

c. Service Lead / Koordinator

Tugasnya:

- review SLA
- ikut rapat layanan
- cek laporan
- bantu eskalasi masalah besar

Biaya:

- Rp18.000.000 per bulan
- dipakai 30%
- 12 bulan

Perhitungan:
Rp18.000.000 × 12 × 30% = Rp64.800.000

Total biaya SDM

Rp144.000.000 + Rp96.000.000 + Rp64.800.000 = Rp304.800.000

2. Tambahkan biaya pendukung

Selain orang, biasanya ada biaya lain.

a. Transisi awal layanan

Untuk:

- serah terima
- baca dokumen
- cek akun
- penyesuaian SOP

Contoh:
Rp10.000.000

b. Laporan bulanan dan review SLA

Contoh:
Rp6.000.000

c. Update SOP / dokumentasi

Contoh:
Rp5.000.000

d. Cadangan operasional kecil

Contoh:
Rp4.000.000

Total biaya pendukung

Rp10.000.000 + Rp6.000.000 + Rp5.000.000 + Rp4.000.000 = Rp25.000.000

3. Hitung total sebelum pajak

- Total SDM: **Rp304.800.000**
- Total biaya pendukung: **Rp25.000.000**

Total sebelum pajak:

Rp304.800.000 + Rp25.000.000 = Rp329.800.000

4. Tambahkan PPN 11%

PPN:

11% × Rp329.800.000 = Rp36.278.000

5. Total HPS

Rp329.800.000 + Rp36.278.000 = Rp366.078.000

Jadi contoh **HPS managed service 1 tahun** adalah:

Rp366.078.000

6. Kenapa bisa mahal padahal tidak beli barang?

Karena yang dibayar adalah:

- orang yang menjaga layanan
- respons gangguan
- monitoring
- backup checking
- patching
- laporan
- koordinasi
- bukti layanan

Jadi, **managed service itu membayar keberlangsungan layanan**, bukan membeli server atau lisensi.

7. Hal yang sering lupa dihitung

Yang sering terlupa dalam HPS *managed service*:

- biaya transisi awal
- laporan bulanan
- dokumentasi
- review SLA
- backup checking
- handover di akhir kontrak
- koordinasi dengan vendor lain

8. Rumus paling mudah

Cara paling gampang:

HPS Managed Service = Biaya SDM + Biaya Pendukung + Pajak

Atau lebih rinci:

HPS = (tarif orang × lama kerja × tingkat keterlibatan) + biaya laporan/transisi/SOP + PPN

9. Format cepat yang bisa dipakai

Kalau mau menghitung cepat, isi saja:

- nama layanan
- cakupan layanan
- jam layanan
- peran yang dibutuhkan
- tarif tiap peran
- berapa bulan
- apakah full-time atau parsial
- biaya pendukung
- pajak
- total

Inti paling sederhananya

Managed service bukan beli barang.
Yang dibeli adalah **layanan operasional**.

Karena itu, HPS-nya harus dihitung dari:

- **siapa yang bekerja**
- **berapa lama**
- **apa tugasnya**
- **layanan apa yang harus dijaga**
- **laporan dan bukti apa yang harus dibuat**

Lampiran G. Contoh SLA dan UAT

A. Contoh SLA sederhana

SLA adalah janji layanan yang tertulis dan bisa diukur.
Sederhananya, SLA menjawab pertanyaan:

- kalau ada masalah, vendor harus merespons seberapa cepat
- kalau sistem gangguan, vendor harus memperbaiki dalam waktu berapa lama
- layanan normalnya harus tersedia berapa persen
- bukti layanannya apa

Dalam dokumen utama Anda, topik **SLA** memang ditempatkan sebagai bagian penting dari kontrak, layanan dukungan, cloud, dan managed service.

1. Contoh SLA untuk aplikasi internal

Misalnya sebuah aplikasi dipakai pegawai untuk pekerjaan harian.

SLA bisa ditulis seperti ini:

Jam layanan:
Senin–Jumat, pukul 08.00–17.00.

Ketersediaan layanan:
Aplikasi harus tersedia minimal **99%** pada jam layanan.

Respon gangguan:

- Gangguan kritis: respon maksimal **30 menit**
- Gangguan sedang: respon maksimal **2 jam**
- Gangguan ringan: respon maksimal **1 hari kerja**

Waktu penyelesaian:

- Gangguan kritis: maksimal **4 jam**
- Gangguan sedang: maksimal **1 hari kerja**
- Gangguan ringan: maksimal **3 hari kerja**

Backup:
Backup dilakukan **setiap hari**.

Laporan:
Vendor wajib mengirim **laporan bulanan** berisi jumlah gangguan, waktu respon, waktu perbaikan, dan penyebab gangguan utama.

2. Arti level gangguan

Supaya mudah dipahami, level gangguan harus dijelaskan.

Gangguan kritis
Sistem tidak bisa dipakai sama sekali, atau fitur utama berhenti total.

Gangguan sedang
Sistem masih bisa dipakai, tetapi ada fungsi penting yang terganggu.

Gangguan ringan
Masalah kecil, tampilan, atau bug yang tidak menghentikan pekerjaan utama.

Ini penting karena SLA yang baik harus punya ukuran yang jelas dan bisa ditagih, bukan hanya kalimat umum seperti “vendor harus support dengan baik.”

3. Contoh SLA untuk managed service

Kalau yang dibeli adalah **managed service**, SLA bisa lebih operasional.

Contohnya:

- tiket helpdesk dijawab maksimal **30 menit**
- insiden prioritas tinggi eskalasi maksimal **15 menit**
- patch keamanan kritis dipasang maksimal **7 hari**
- pengecekan backup dilakukan **harian**
- restore test dilakukan **minimal 1 kali per 6 bulan**
- laporan SLA dikirim **setiap bulan**
- rapat evaluasi layanan dilakukan **setiap bulan**

4. Bukti SLA

SLA tidak cukup ditulis. Harus ada buktinya.

Bukti yang paling sederhana:

- tiket layanan
- log gangguan
- bukti backup
- laporan bulanan
- notulen rapat evaluasi
- daftar pelanggaran SLA bila ada

Tanpa bukti, SLA hanya janji di atas kertas. Ini juga sejalan dengan penekanan dokumen Anda pada **bukti layanan dan audit trail**.

B. Contoh UAT sederhana

UAT adalah *User Acceptance Test*.
Sederhananya, UAT adalah uji coba oleh pengguna untuk memastikan sistem **sudah sesuai kebutuhan**.

UAT penting karena sistem tidak cukup hanya “bisa jalan.” Sistem harus **cocok dipakai** oleh pengguna. Dalam dokumen utama Anda, UAT memang ditempatkan sebagai bagian penting dari **kontrak, acceptance, dan serah terima**.

1. Tujuan UAT

Tujuan UAT adalah memastikan:

- fitur utama bekerja
- alur kerja sesuai kebutuhan
- hasil yang keluar benar
- pengguna bisa menjalankan tugasnya
- sistem cukup siap dipakai

2. Contoh UAT sederhana untuk aplikasi layanan internal

Misalnya ada aplikasi persuratan internal.

Beberapa skenario UAT bisa dibuat seperti ini:

Skenario 1: Login pengguna
User bisa login memakai akun yang benar.
Hasil yang diharapkan: user masuk ke dashboard sesuai perannya.

Skenario 2: Membuat surat baru
User bisa membuat draft surat.
Hasil yang diharapkan: draft tersimpan dan muncul di daftar surat.

Skenario 3: Proses persetujuan
Atasan bisa melihat dan menyetujui surat.
Hasil yang diharapkan: status surat berubah menjadi disetujui.

Skenario 4: Pencarian surat
User bisa mencari surat berdasarkan nomor atau kata kunci.
Hasil yang diharapkan: hasil pencarian sesuai data.

Skenario 5: Cetak / ekspor
Surat bisa dicetak atau diunduh.
Hasil yang diharapkan: file keluar dengan format benar.

3. Format UAT paling sederhana

Format UAT bisa sangat sederhana, misalnya:

- nomor skenario
- nama fungsi yang diuji
- langkah uji
- hasil yang diharapkan
- hasil aktual
- status: lulus / tidak lulus
- catatan

4. Contoh cara menulis hasil UAT

Contoh sederhana:

Nama fungsi: Login
Langkah uji: Masukkan username dan password yang benar
Hasil yang diharapkan: User masuk ke dashboard
Hasil aktual: User berhasil masuk
Status: Lulus

Contoh lain:

Nama fungsi: Persetujuan surat
Langkah uji: Atasan klik tombol setujui
Hasil yang diharapkan: Status berubah menjadi disetujui
Hasil aktual: Tombol bisa diklik, tetapi status tidak berubah
Status: Tidak lulus

5. Kapan UAT dinyatakan selesai

UAT sebaiknya dianggap selesai kalau:

- semua fungsi utama sudah diuji
- masalah besar sudah diperbaiki
- masalah kecil sudah dicatat
- pengguna utama setuju sistem layak dipakai
- ada berita acara atau bukti persetujuan

Literatur proyek IT yang Anda unggah juga menekankan bahwa UAT paling sehat dilakukan secara pragmatis: ruang uji dibagi ke blok fungsi yang jelas, lalu diuji berdasarkan tingkat kesiapan, bukan menunggu semua hal sempurna sekaligus.

6. Hal penting yang sering dilupakan saat UAT

Yang sering terlupakan:

- data uji belum siap
- user belum paham cara menguji
- bug kecil dan bug besar dicampur
- hasil uji tidak ditulis
- UAT dimulai terlalu cepat saat sistem belum cukup stabil

Karena itu, UAT harus sederhana, tertulis, dan fokus pada fungsi yang benar-benar penting bagi pengguna.

C. Contoh kalimat siap pakai untuk buku / KAK / kontrak

Contoh kalimat SLA

“Penyedia wajib menjaga ketersediaan layanan minimal 99% pada jam layanan yang disepakati. Penyedia wajib merespons insiden kritis paling lambat 30 menit sejak tiket diterima dan menyelesaikannya paling lambat 4 jam, kecuali terdapat kondisi force majeure atau ketergantungan pihak lain yang telah diberitahukan secara tertulis.”

Contoh kalimat UAT

“UAT dilakukan oleh pengguna yang ditunjuk oleh PPK/Tim Teknis berdasarkan skenario uji yang disepakati bersama. UAT dinyatakan lulus apabila fungsi utama berjalan sesuai kebutuhan, kesalahan kritis telah diperbaiki, dan hasil uji didokumentasikan dalam berita acara atau laporan hasil uji.”

D. Inti paling sederhananya

SLA menjawab:
“Kalau layanan terganggu, vendor harus bergerak seberapa cepat dan kualitas layanannya seperti apa?”

UAT menjawab:
“Apakah sistem ini benar-benar sudah bisa dipakai oleh pengguna?”

Lampiran H. Checklist PDP dan Keamanan

Area	Checklist	Catatan Praktis
Pemetaan data pribadi	Sudah diidentifikasi data apa saja yang diproses, sumbernya dari mana, untuk tujuan apa, dan siapa yang mengakses	Data pribadi harus dipetakan sejak awal, bukan setelah sistem jadi. UU PDP menekankan pemrosesan yang bertanggung jawab dan dapat dibuktikan.
Peran pengendali dan prosesor	Sudah jelas siapa pengendali data dan siapa prosesor data, termasuk bila ada vendor atau pihak ketiga	Ini penting untuk kontrak, insiden, dan tanggung jawab hukum.
Tujuan pemrosesan	Tujuan pengumpulan dan penggunaan data ditulis jelas dan tidak terlalu luas	Hindari pengumpulan data “sekalian saja”. Data harus relevan dengan tujuan layanan.
Dasar pemrosesan	Ada dasar pemrosesan yang sah dan bisa dijelaskan	Penting terutama untuk layanan publik, LMS, e-commerce, dan aplikasi pegawai/pelanggan.
Klasifikasi data	Sudah dibedakan mana data biasa, data sensitif/spesifik, log aktivitas, data transaksi, dan data hasil integrasi	Klasifikasi membantu menentukan tingkat kontrol dan keamanan.
Privacy by design	Desain sistem sudah membatasi data yang dikumpulkan, mengatur role akses, dan menyalakan audit trail seperlunya	Privasi harus dibangun dari desain, bukan ditempel belakangan.
Privacy by default	Pengaturan awal sistem tidak membuka data terlalu luas, akun admin dibatasi, dan fitur sensitif tidak aktif sembarangan	Default yang terlalu terbuka adalah sumber risiko umum.
Hak subjek data	Sistem/prosedur mendukung akses, koreksi, atau penggunaan data sesuai hak subjek data bila relevan	Hak subjek data harus bisa dipenuhi secara tertib.
Retensi data	Sudah ditetapkan berapa lama data disimpan dan kapan dihapus/dimusnahkan	Retensi tidak boleh dibiarkan tanpa batas tanpa alasan jelas.

Penghapusan / pemusnahan data	Ada prosedur penghapusan, anonimisasi, atau pemusnahan data	Penting untuk akhir kontrak, pergantian sistem, atau penutupan layanan.
Kontrol akses	Hak akses berbasis peran, akun admin dan user dipisah, prinsip *least privilege* diterapkan	Ini kontrol keamanan paling dasar dan paling penting.
MFA / autentikasi kuat	MFA aktif untuk akun admin dan akses kritikal bila relevan	Minimum untuk sistem penting, cloud, dan layanan publik/digital.
Logging dan audit trail	Aktivitas penting tercatat, log disimpan, retensi log jelas, dan log bisa diakses untuk audit	Audit trail penting untuk pembuktian, audit, dan investigasi.
Enkripsi	Data saat transit dan/atau saat tersimpan dilindungi sesuai kebutuhan	Sesuaikan dengan klasifikasi data dan tingkat risiko.
Hardening	Akun default, konfigurasi lemah, dan layanan yang tidak perlu sudah dinonaktifkan	Jangan serah terima sistem dalam kondisi mentah.
Patching dan kerentanan	Ada proses patching, review kerentanan, dan bukti tindak lanjut untuk celah yang ditemukan	Sistem yang tidak di-*patch* cepat menjadi rapuh.
Backup	Backup berjalan rutin dan ada bukti pelaksanaannya	Backup bukan hanya SOP, harus benar-benar berjalan.
Restore test	Pernah dilakukan uji restore dan hasilnya terdokumentasi	Ini bukti paling penting bahwa backup benar-benar berguna.
DR / pemulihan bencana	Ada rencana pemulihan untuk gangguan besar	Penting untuk sistem layanan penting atau kritikal.
Notifikasi insiden	Ada jalur pelaporan dan notifikasi bila terjadi insiden keamanan atau kegagalan pelindungan data	Wajib dipikirkan sejak kontrak dan SOP.
Bantuan investigasi insiden	Vendor atau tim operasional wajib membantu investigasi, preservasi log, dan analisis awal	Jangan sampai saat insiden semua saling lempar tanggung jawab.
Keamanan vendor / pihak ketiga	Vendor, cloud, SaaS, atau subkontraktor tunduk pada standar keamanan yang sama	Rantai pasok digital bisa jadi titik lemah jika tidak dikontrol.

Status PSE	Untuk layanan elektronik pihak ketiga, status PSE dan kepatuhannya sudah dicek bila relevan	Penting untuk cloud, SaaS, platform digital, dan layanan elektronik privat.
Dokumentasi PDP dan keamanan	Ada kebijakan, SOP, klausul kontrak, dan dokumen teknis yang mendukung kepatuhan	Tanpa dokumen, sulit membuktikan kepatuhan.
Uji keamanan sebelum go-live	Sudah ada vulnerability scan, pentest, atau uji keamanan lain sesuai tingkat risiko	Sistem penting tidak boleh langsung *go-live* tanpa uji keamanan minimum.
Kesiapan audit	Bukti kontrol cukup untuk audit internal, audit keamanan, atau audit SPBE	Sistem pemerintah harus siap diaudit, bukan hanya aktif dipakai.
Pelatihan admin dan operator	Admin/operator sudah dilatih soal keamanan, akses, backup, dan SOP insiden	Banyak risiko muncul bukan karena alat kurang, tetapi karena orang belum siap.
Serah terima akses dan dokumen	Akun admin, SOP, manual, repo, dan dokumen keamanan sudah diserahterimakan dengan jelas	Penting untuk menghindari ketergantungan berlebihan pada vendor.
Review berkala	Ada review berkala untuk akses, log, backup, patching, dan insiden	Keamanan dan PDP bukan pekerjaan sekali selesai.

Ringkasan pakai cepat

Kondisi	Checklist yang paling penting
Aplikasi internal pegawai	akses, logging, backup, retensi, SOP insiden
LMS / pendidikan digital	data peserta, hak akses, retensi, mobile access, log aktivitas
E-commerce / marketplace	data pelanggan, keamanan transaksi, audit trail, notifikasi insiden
Cloud / SaaS	PSE, IAM, enkripsi, portabilitas data, exit plan, log
Managed service	akses admin, patching, backup, bukti layanan, audit trail

Catatan pakai

Prinsip	Makna Praktis
PDP dan keamanan tidak boleh dipisah	Banyak risiko PDP justru muncul karena kontrol keamanan lemah
Yang penting bukan hanya ada kebijakan	Yang lebih penting adalah ada bukti: log, tiket, SOP, hasil uji, laporan
Mulai dari risiko layanan	Tidak semua sistem butuh kontrol yang sama, tetapi semua sistem butuh kontrol minimum
Masukkan ke dokumen pengadaan	Checklist ini paling berguna bila diterjemahkan ke KAK, spesifikasi, kontrak, SLA, dan serah terima

Lampiran I. Checklist Due Diligence Cloud / PSE

Area	Checklist	Catatan Praktis
Identitas layanan	Nama layanan cloud/SaaS, model layanan (*IaaS / PaaS / SaaS*), nama penyedia, dan kontak resmi jelas	Ini titik awal evaluasi
Model layanan	Sudah jelas apakah yang dibeli infrastruktur, platform, atau aplikasi jadi	Model layanan mempengaruhi tanggung jawab
Cakupan layanan	Fitur, kapasitas, modul, lingkungan produksi/uji, dan batas layanan tertulis jelas	Hindari ruang lingkup kabur
Status PSE	Status PSE penyedia sudah dicek bila relevan	Penting untuk layanan elektronik pihak ketiga
Legalitas vendor	Badan usaha, identitas hukum, dan dokumen legal dasar tersedia	Bagian dari due diligence minimum
Region / lokasi layanan	Lokasi pemrosesan atau region layanan diketahui bila relevan	Penting untuk latensi, tata kelola, dan data
Kepemilikan dan alur data	Jelas data milik siapa, diproses oleh siapa, dan mengalir ke mana	Harus dipahami sejak awal
Peran data	Sudah jelas siapa pengendali data dan siapa prosesor data	Penting untuk kontrak dan insiden
Jenis data yang diproses	Sudah dipetakan apakah ada data pribadi, data sensitif, log, atau data transaksi	Menentukan kontrol keamanan dan privasi
Dasar pemrosesan	Tujuan penggunaan data dan dasar pemrosesannya dapat dijelaskan	Jangan ada pemrosesan data yang tidak perlu
Kontrol akses	Ada IAM, pembagian role, pemisahan admin-user, dan pengelolaan akun yang jelas	Minimum untuk semua layanan cloud
MFA	MFA tersedia dan aktif untuk akun admin/akses kritikal	Sangat penting untuk cloud
Logging dan audit trail	Aktivitas penting tercatat, log bisa diakses, dan retensinya jelas	Penting untuk audit dan investigasi
Enkripsi	Data saat transit dan/atau saat tersimpan dilindungi sesuai kebutuhan	Sesuaikan dengan klasifikasi data

Backup	Backup tersedia, jadwalnya jelas, dan kebijakannya tertulis	Jangan hanya percaya klaim vendor
Restore test	Ada bukti uji pemulihan atau setidaknya penjelasan prosedur restore	Sangat penting untuk menilai ketahanan
DR / failover	Ada mekanisme pemulihan gangguan, failover, atau layanan pengganti bila diperlukan	Sesuaikan dengan kritikalitas sistem
Keamanan operasional	Ada patching, vulnerability management, dan review keamanan berkala	Cloud tetap butuh kontrol keamanan aktif
Notifikasi insiden	Mekanisme pelaporan insiden dan eskalasi vendor jelas	Harus masuk kontrak dan SOP
SLA layanan	Uptime, waktu respons, waktu pemulihan, dan komitmen support tertulis jelas	Jangan hanya menerima janji umum
Support dan eskalasi	Helpdesk, jam layanan, jalur eskalasi, dan PIC vendor tersedia	Penting untuk operasi nyata
Integrasi	API, SSO, sinkronisasi data, webhook, atau integrasi lain tersedia bila diperlukan	Sangat penting untuk sistem yang tidak berdiri sendiri
Portabilitas data	Data bisa diekspor dalam format yang masuk akal dan terdokumentasi	Kunci untuk mengurangi *vendor lock-in*
Exit plan	Ada prosedur keluar, migrasi, handover data, dan penutupan akun/layanan	Wajib dipikirkan sejak awal
Penghapusan data	Ada aturan penghapusan data setelah kontrak berakhir atau layanan dihentikan	Penting untuk PDP dan tata kelola
Subkontraktor / pihak ketiga	Jelas apakah vendor memakai pihak ketiga lain untuk hosting, support, atau pemrosesan	Risiko rantai pasok harus terbaca
Dokumentasi layanan	Tersedia dokumen arsitektur, panduan admin, SOP, dan dokumentasi operasional minimum	Tanpa dokumen, sulit dioperasikan dan diaudit
Bukti layanan	Vendor dapat menunjukkan dashboard, laporan SLA, log, tiket, dan bukti operasional lain	Layanan harus bisa dibuktikan
Kesiapan audit	Layanan cukup terbuka untuk audit internal, review kontrak, atau pemeriksaan teknis yang wajar	Sangat penting untuk instansi publik

Kelayakan akhir	Status akhir: layak / layak dengan catatan / belum layak	Gunakan penilaian tegas, jangan abu-abu

Ringkasan pakai cepat

Kondisi	Fokus Checklist
Cloud untuk aplikasi internal	IAM, backup, SLA, support, portabilitas data
SaaS untuk layanan publik	PSE, PDP, logging, notifikasi insiden, exit plan
LMS berbasis cloud	data peserta, akses admin, backup, support, integrasi
E-commerce / transaksi digital	keamanan transaksi, audit trail, SLA, insiden, portabilitas
Managed service berbasis cloud	support, eskalasi, bukti layanan, log, DR

Penilaian sederhana

Status	Arti
Layak	Hampir semua kontrol utama ada dan dapat dibuktikan
Layak dengan catatan	Bisa dipakai, tetapi ada beberapa gap yang harus ditutup di kontrak/SOP
Belum layak	Ada kekurangan besar pada legalitas, keamanan, data, SLA, atau exit plan

Glossary

Istilah	Arti singkat
Acceptance	Penerimaan hasil pekerjaan setelah diuji dan dinilai memenuhi syarat.
Agile	Metode kerja proyek yang iteratif, bertahap, dan adaptif terhadap perubahan.
API	*Application Programming Interface*; mekanisme agar aplikasi/sistem bisa saling terhubung dan bertukar data.
AP	*Access Point*; perangkat jaringan untuk menyediakan koneksi nirkabel.
Arsitektur Aplikasi	Rancangan struktur aplikasi, modul, dan hubungan antar aplikasi.
Arsitektur Bisnis	Gambaran proses bisnis, layanan, dan kebutuhan organisasi yang menjadi dasar sistem.
Arsitektur Data	Rancangan data, aliran data, struktur data, dan hubungan antar sumber data.
Arsitektur Infrastruktur	Rancangan server, storage, jaringan, pusat data, dan lingkungan operasi sistem.
Arsitektur Keamanan	Rancangan kontrol keamanan seperti akses, enkripsi, logging, dan perlindungan sistem.
As-is	Kondisi saat ini sebelum perubahan atau sebelum proyek dijalankan.
Audit Trail	Jejak bukti aktivitas sistem atau layanan yang dapat ditelusuri untuk audit dan investigasi.
BA	*Business Analyst*; peran yang menerjemahkan kebutuhan bisnis menjadi kebutuhan sistem.
Backup	Salinan data/sistem untuk pemulihan bila terjadi gangguan atau kehilangan data.
BAST	*Berita Acara Serah Terima*; dokumen formal serah terima hasil pekerjaan.
BCP	*Business Continuity Plan*; rencana menjaga layanan tetap berjalan saat terjadi gangguan.
Benchmark Pasar	Pembanding harga, kualitas, atau tarif berdasarkan kondisi pasar.
BLUD	*Badan Layanan Umum Daerah*.

BNSP	*Badan Nasional Sertifikasi Profesi*; lembaga nasional terkait sistem sertifikasi profesi.
Build vs Buy vs Adopt OSS	Pilihan strategi: membangun sendiri, membeli solusi jadi, atau mengadopsi solusi *open source*.
BUMD	*Badan Usaha Milik Daerah*.
Business Case	Alasan bisnis dan manfaat terukur yang menjadi dasar proyek dijalankan.
CAPEX	*Capital Expenditure*; belanja modal atau biaya investasi awal.
Change Request	Permintaan perubahan ruang lingkup, fitur, jadwal, atau deliverable proyek.
CI/CD	Praktik *Continuous Integration/Continuous Delivery* untuk pengembangan dan rilis aplikasi yang lebih tertib dan cepat.
Cloud	Layanan komputasi berbasis jaringan/internet untuk infrastruktur, platform, atau aplikasi.
Cloud Bill	Tagihan biaya penggunaan layanan cloud.
Colocation	Penempatan server milik sendiri di fasilitas pusat data pihak lain.
Deliverable	Hasil kerja yang wajib diserahkan dalam proyek, misalnya aplikasi, dokumen, source code, SOP, atau pelatihan.
DevOps	Pendekatan yang menggabungkan pengembangan aplikasi dan operasi TI agar rilis lebih cepat dan stabil.
Disaster Recovery (DR)	Kemampuan memulihkan sistem/layanan setelah insiden besar atau bencana.
Due Diligence	Pemeriksaan menyeluruh terhadap vendor/penyedia sebelum penetapan atau kontrak.
e-Commerce	Sistem perdagangan digital untuk transaksi barang/jasa secara elektronik.
Endpoint	Perangkat pengguna akhir seperti laptop, desktop, atau perangkat akses lain.
Enkripsi	Proses mengamankan data dengan mengubahnya menjadi format yang tidak mudah dibaca pihak tidak berwenang.
ERP	*Enterprise Resource Planning*; sistem terintegrasi untuk proses bisnis organisasi.
Eskalasi	Proses menaikkan penanganan isu/insiden ke level yang lebih tinggi.

Exit Management	Pengaturan transisi saat kontrak berakhir atau saat vendor diganti agar layanan tetap aman dan berlanjut.
Exit Plan	Rencana keluar dari platform, vendor, atau layanan tertentu tanpa merusak keberlanjutan sistem.
Fee	Imbalan jasa profesional atau tenaga ahli.
Firewall	Perangkat/sistem pengaman lalu lintas jaringan.
Forensik Dasar	Pemeriksaan awal insiden keamanan untuk menelusuri sebab, bukti, dan dampak.
Fraud Control	Pengendalian untuk mencegah dan mendeteksi kecurangan dalam sistem/transaksi.
Free Software	Perangkat lunak yang menekankan kebebasan penggunaan, mempelajari, memodifikasi, dan mendistribusikan ulang.
Go-live	Saat sistem resmi mulai digunakan dalam operasional nyata.
Governance Board	Forum pengendali keputusan penting proyek, perubahan, atau prioritas.
HA	*High Availability*; rancangan agar layanan tetap tersedia dengan gangguan seminimal mungkin.
Hardening	Penguatan konfigurasi sistem agar lebih aman dari serangan atau kesalahan.
Helpdesk	Layanan bantuan pengguna untuk masalah operasional atau teknis.
HPS	*Harga Perkiraan Sendiri*; estimasi biaya yang disusun sebagai dasar pengadaan.
Hybrid Cloud	Model gabungan antara infrastruktur lokal, private cloud, dan/atau public cloud.
Hybrid	Model campuran, misalnya campuran metode kerja, teknologi, atau solusi open source dan proprietary.
IaaS	*Infrastructure as a Service*; layanan cloud berbasis infrastruktur komputasi.
Incident Response (IR)	Tindakan terstruktur untuk menangani insiden keamanan atau gangguan layanan.
Indikator Manfaat	Ukuran untuk melihat apakah proyek memberi hasil/manfaat yang ditargetkan.

Integrasi	Penghubungan antar sistem, aplikasi, atau data agar bekerja sebagai satu ekosistem.
Interoperabilitas	Kemampuan antar sistem untuk bertukar dan menggunakan data/informasi secara efektif.
Issue Log	Daftar masalah yang muncul selama proyek dan status penanganannya.
Jasa Konsultansi	Paket pengadaan yang dominan pada keahlian analisis, kajian, atau advis profesional.
KAK	*Kerangka Acuan Kerja*; dokumen yang menjelaskan latar belakang, tujuan, ruang lingkup, output, tahapan, dan kebutuhan pekerjaan.
Katalog Elektronik	Sarana pengadaan digital yang memuat produk/jasa beserta spesifikasi dan harga.
Kedaulatan Digital	Kemampuan organisasi/negara mengendalikan sistem, data, kapasitas, dan arah teknologinya sendiri.
Knowledge Transfer	Alih pengetahuan dari penyedia ke organisasi agar sistem bisa dioperasikan dan dipelihara mandiri.
Layanan Dukungan	Dukungan operasional seperti helpdesk, pemeliharaan, monitoring, atau perbaikan.
Lifecycle	Siklus hidup perangkat/sistem dari pengadaan, operasi, pemeliharaan, hingga penggantian.
Lisensi	Aturan hak penggunaan perangkat lunak.
LMS	*Learning Management System*; sistem untuk pembelajaran digital, kursus, ujian, konten, dan analitik belajar.
Logging	Pencatatan aktivitas sistem sebagai bukti operasional dan keamanan.
LSP	*Lembaga Sertifikasi Profesi*; penyelenggara sertifikasi profesi pada skema tertentu.
Managed Service	Layanan TI yang dioperasikan/dikelola penyedia secara berkelanjutan berdasarkan tingkat layanan tertentu.
Marketplace	Platform digital yang mempertemukan penjual dan pembeli.
Message Bus	Mekanisme/perantara pertukaran pesan antar aplikasi atau layanan.

MFA	*Multi-Factor Authentication*; autentikasi dengan lebih dari satu faktor verifikasi.
Milestone	Titik capaian penting dalam jadwal proyek.
Migrasi	Pemindahan data, aplikasi, atau layanan dari sistem lama ke sistem baru.
Mobile App	Aplikasi yang berjalan di perangkat seluler.
Monitoring	Pemantauan kesehatan, performa, dan status layanan/sistem.
NOC	*Network Operations Center*; fungsi/tim pemantauan dan pengelolaan operasi jaringan.
Observability	Kemampuan memahami kondisi sistem melalui log, metrik, dan jejak aktivitas.
On-budget	Kondisi proyek berjalan sesuai anggaran.
On-premise	Infrastruktur/sistem yang ditempatkan dan dikelola di lingkungan sendiri.
On-scope	Kondisi proyek tetap dalam ruang lingkup yang disepakati.
On-time	Kondisi proyek berjalan sesuai jadwal.
Open Source	Perangkat lunak dengan kode sumber yang dapat diakses, dipelajari, dimodifikasi, dan dikelola lebih terbuka sesuai lisensinya.
OPEX	*Operational Expenditure*; biaya operasional berulang selama layanan berjalan.
OSS	*Open Source Software*; perangkat lunak open source.
Output	Hasil langsung yang dihasilkan proyek atau paket pekerjaan.
Outcome	Dampak atau manfaat yang dihasilkan setelah output digunakan.
PaaS	*Platform as a Service*; layanan cloud berbasis platform untuk pengembangan/operasi aplikasi.
P3DN	*Peningkatan Penggunaan Produk Dalam Negeri*.
Patching	Proses pembaruan sistem/aplikasi untuk perbaikan bug atau keamanan.
PDP	*Pelindungan Data Pribadi*; kerangka perlindungan data pribadi dalam pemrosesan data.

Pentest	*Penetration Testing*; pengujian keamanan dengan simulasi serangan.
Performance Test	Pengujian performa sistem, misalnya kecepatan, beban, dan kestabilan.
PM	*Project Manager*; penanggung jawab pengelolaan proyek.
PMSE	*Perdagangan Melalui Sistem Elektronik*.
Portabilitas	Kemudahan memindahkan data, aplikasi, atau layanan ke platform lain.
Privacy by Default	Prinsip bahwa pengaturan bawaan sistem harus melindungi privasi secara otomatis.
Privacy by Design	Prinsip bahwa pelindungan privasi harus dirancang sejak awal pembangunan sistem.
Private Cloud	Cloud yang dikhususkan untuk satu organisasi atau lingkungan terbatas.
Proctoring	Mekanisme pengawasan ujian digital.
Project Manager	Lihat **PM**.
Proprietary	Solusi tertutup yang hak akses, modifikasi, atau distribusinya dibatasi pemiliknya.
PSE	*Penyelenggara Sistem Elektronik*.
Public Cloud	Layanan cloud yang disediakan untuk banyak pengguna/organisasi melalui penyedia publik.
QA	*Quality Assurance*; fungsi penjaminan mutu hasil pekerjaan atau aplikasi.
Rack	Perangkat/fasilitas penempatan server dan perangkat pusat data.
Rate Card	Daftar tarif per peran atau level SDM untuk perhitungan biaya jasa.
Redundansi	Penyediaan komponen cadangan agar layanan tetap berjalan saat ada kegagalan.
Region	Lokasi wilayah pusat data/layanan cloud.
Remedy	Tindakan perbaikan atau pemulihan bila kewajiban kontrak tidak terpenuhi.
Repo	*Repository*; tempat penyimpanan source code dan riwayat perubahan.

Risk Register	Daftar risiko proyek beserta tingkat risiko, PIC, dan rencana mitigasi.
RUP	*Rencana Umum Pengadaan*.
SaaS	*Software as a Service*; layanan aplikasi siap pakai berbasis cloud.
SBK	*Standar Biaya Keluaran*.
SBM	*Standar Biaya Masukan*.
Security Test	Pengujian untuk menilai keamanan aplikasi/sistem.
Serah Terima	Tahap penyerahan hasil pekerjaan dari penyedia kepada pengguna/organisasi.
SIEM	*Security Information and Event Management*; pengelolaan dan analisis log/kejadian keamanan.
Silo Aplikasi	Kondisi saat aplikasi berjalan sendiri-sendiri tanpa integrasi memadai.
SKKNI	*Standar Kompetensi Kerja Nasional Indonesia*.
SLA	*Service Level Agreement*; kesepakatan tingkat layanan yang harus dipenuhi penyedia.
SLO	*Service Level Objective*; target kinerja layanan yang lebih operasional/terukur.
SOC	*Security Operations Center*; fungsi/tim pemantauan dan respons keamanan siber.
SOP	*Standard Operating Procedure*; prosedur operasional baku.
Source Code	Kode sumber aplikasi yang menjadi dasar pengembangan dan pemeliharaan sistem.
SPBE	*Sistem Pemerintahan Berbasis Elektronik*.
Spare Part	Komponen cadangan untuk perawatan/perbaikan perangkat.
Spesifikasi Teknis	Uraian kebutuhan teknis/fungsional/kinerja yang menjadi dasar pengadaan.
Sprint	Siklus kerja singkat dalam metode agile untuk menghasilkan bagian pekerjaan tertentu.
Stack	Tumpukan/komposisi teknologi yang digunakan dalam suatu sistem.
Status Report	Laporan berkala tentang kemajuan, hambatan, dan posisi proyek.

Storage	Media/infrastruktur penyimpanan data.
Support	Dukungan teknis atau operasional setelah implementasi.
System Analyst	Peran yang menganalisis kebutuhan sistem dan rancangan solusi teknis-logis.
TCO	*Total Cost of Ownership*; total biaya nyata sepanjang siklus hidup sistem, bukan hanya harga awal.
TKDN	*Tingkat Komponen Dalam Negeri*.
To-be	Kondisi target yang ingin dicapai setelah proyek/perubahan selesai.
TTE	*Tanda Tangan Elektronik*.
UAT	*User Acceptance Test*; uji penerimaan pengguna untuk memastikan hasil sesuai kebutuhan.
UKPBJ	*Unit Kerja Pengadaan Barang/Jasa*.
UPS	*Uninterruptible Power Supply*; perangkat cadangan daya listrik.
Uptime	Tingkat ketersediaan layanan/sistem selama periode tertentu.
Vendor Lock-in	Ketergantungan berlebihan pada satu vendor/produk sehingga sulit migrasi atau negosiasi.
Vulnerability Management	Pengelolaan celah keamanan, mulai dari identifikasi sampai perbaikan.
Warranty	Garansi dukungan/perbaikan dari penyedia atau pabrikan.
Waterfall	Metode proyek berurutan dari tahap awal sampai akhir secara linear.
Web App	Aplikasi yang diakses melalui browser/web.

Referensi

Aktayewa, A., Fuschi, D. L., Dauletkireeva, L., Zaurbekov, N., Assanova, K., Urmashev, B., & Niyazova, R. (2017). *Project management: Software and technologies exercises (Part 3).* Evero.

Badan Nasional Sertifikasi Profesi. (n.d.). *Sertifikasi kompetensi profesi.* https://bnsp.go.id

Badan Siber dan Sandi Negara. (n.d.). *Pedoman dan audit keamanan siber/SPBE.* https://bssn.go.id

Dąbrowski, M. (2023). *Managing IT projects: How to pragmatically deliver projects for external customers.* Apress.

Dimitrov, D. (2020). *Software project estimation: Intelligent forecasting, project control, and client relationship management.* Apress. https://doi.org/10.1007/978-1-4842-5025-9

Dominick, C., & Lunney, S. R. (2022). *The procurement game plan: Winning strategies and techniques for supply management professionals* (2nd ed.). J. Ross Publishing.

Indonesia. (2017). *Peraturan Menteri Ketenagakerjaan Republik Indonesia Nomor 1 Tahun 2017 tentang struktur dan skala upah.* https://peraturan.bpk.go.id

Indonesia. (2018). *Peraturan Presiden Republik Indonesia Nomor 16 Tahun 2018 tentang pengadaan barang/jasa pemerintah.* https://peraturan.bpk.go.id/details/73586/perpres-no-16-tahun-2018

Indonesia. (2018). *Peraturan Presiden Republik Indonesia Nomor 95 Tahun 2018 tentang sistem pemerintahan berbasis elektronik.* https://peraturan.bpk.go.id

Indonesia. (2019). *Peraturan Pemerintah Republik Indonesia Nomor 71 Tahun 2019 tentang penyelenggaraan sistem dan transaksi elektronik.* https://peraturan.bpk.go.id/Details/122030/pp-no-71-tahun-2019

Indonesia. (2019). *Peraturan Pemerintah Republik Indonesia Nomor 80 Tahun 2019 tentang perdagangan melalui sistem elektronik.* https://peraturan.bpk.go.id/Details/126143/pp-no-80-tahun-2019

Indonesia. (2020). *Peraturan Menteri Komunikasi dan Informatika Republik Indonesia Nomor 5 Tahun 2020 tentang penyelenggara sistem elektronik lingkup privat.* https://peraturan.bpk.go.id

Indonesia. (2021). *Peraturan Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah Nomor 11 Tahun 2021 tentang pedoman perencanaan pengadaan barang/jasa pemerintah.* https://peraturan.bpk.go.id

Indonesia. (2021). *Peraturan Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah Nomor 12 Tahun 2021 tentang pedoman pelaksanaan pengadaan barang/jasa pemerintah melalui penyedia.* https://peraturan.bpk.go.id/Details/169565/peraturan-lkpp-no-12-tahun-2021

Indonesia. (2021). *Peraturan Menteri Komunikasi dan Informatika Republik Indonesia Nomor 10 Tahun 2021 tentang perubahan atas Peraturan Menteri Komunikasi dan Informatika Nomor 5 Tahun 2020 tentang penyelenggara sistem elektronik lingkup privat.* https://peraturan.bpk.go.id

Indonesia. (2021). *Peraturan Presiden Republik Indonesia Nomor 12 Tahun 2021 tentang perubahan atas Peraturan Presiden Nomor 16 Tahun 2018 tentang pengadaan barang/jasa pemerintah.* https://peraturan.bpk.go.id/Details/161828/perpres-no-12-tahun-2021

Indonesia. (2022). *Instruksi Presiden Republik Indonesia Nomor 2 Tahun 2022 tentang percepatan peningkatan penggunaan produk dalam negeri dan produk usaha mikro, usaha kecil, dan koperasi dalam rangka menyukseskan Gerakan Nasional Bangga Buatan Indonesia pada pelaksanaan pengadaan barang/jasa pemerintah.* https://jdih.setkab.go.id

Indonesia. (2022). *Undang-Undang Republik Indonesia Nomor 27 Tahun 2022 tentang pelindungan data pribadi.* https://peraturan.bpk.go.id/Details/229798/uu-no-27-tahun-2022

Indonesia. (2023). *Peraturan Menteri Perdagangan Republik Indonesia Nomor 31 Tahun 2023 tentang perizinan berusaha, periklanan, pembinaan, dan pengawasan pelaku usaha dalam perdagangan melalui sistem elektronik.* https://peraturan.bpk.go.id/Details/265202/permendag-no-31-tahun-2023

Indonesia. (2024). *Peraturan Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah Nomor 4 Tahun 2024 tentang perubahan atas Peraturan Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah Nomor 12 Tahun 2021 tentang pedoman pelaksanaan pengadaan barang/jasa pemerintah melalui penyedia.* https://peraturan.bpk.go.id

Indonesia. (2024). *Undang-Undang Republik Indonesia Nomor 1 Tahun 2024 tentang perubahan kedua atas Undang-Undang Nomor 11 Tahun 2008 tentang informasi dan transaksi elektronik.* https://peraturan.bpk.go.id

Indonesia. (2025). *Peraturan Presiden Republik Indonesia Nomor 46 Tahun 2025 tentang perubahan kedua atas Peraturan Presiden Nomor 16 Tahun 2018 tentang pengadaan barang/jasa pemerintah.* https://peraturan.bpk.go.id/Details/318647/perpres-no-46-tahun-2025

Kementerian Keuangan Republik Indonesia. (2025). *Peraturan Menteri Keuangan Republik Indonesia Nomor 32 Tahun 2025 tentang standar biaya masukan tahun anggaran 2026.* https://jdih.kemenkeu.go.id/dok/pmk-32-tahun-2025

Kementerian Keuangan Republik Indonesia. (2026). *Peraturan Menteri Keuangan Republik Indonesia Nomor 3 Tahun 2026 tentang standar biaya keluaran dalam penganggaran kementerian negara/lembaga.* https://jdih.kemenkeu.go.id/dok/pmk-3-tahun-2026

Kementerian Ketenagakerjaan Republik Indonesia. (n.d.). *Standar Kompetensi Kerja Nasional Indonesia (SKKNI) bidang terkait TIK.* https://kemnaker.go.id

Lysons, K., & Farrington, B. (2020). *Procurement and supply chain management* (10th ed.). Pearson.

O'Regan, G. (2025). *Guide to software project management.* Springer.

Tentang Penulis



Onno W. Purbo, saat ini bertugas sebagai rektor di Institut Teknologi Tangerang Selatan (ITTS). Onno memperoleh gelar Ph.D bidang Electrical Engineering dari University of Waterloo, Canada, adalah seorang copyleft, educator dan ICT evangelist. Dia sudah mempublikasikan 50++ buku, termasuk free ICT ebook untuk sekolah tahun 2008. Beberapa buku terakhirnya adalah "Internet-TCP/IP: Konsep Dan Implementasi", 2018; "Sistem Operasi, Konsep Dan Membuat Linux OpenWRT Dan ROM Android", 2019; "IPv6 Untuk Mendukung Operasi Jaringan Dan Domain Name System", 2019; "Kubernetes untuk Pemula", 2024 dan "Membuat Operator Seluler 5G Sendiri", 2024.

Dia memimpin sambungan pertama Internet di Institut Teknologi Bandung, tahun 1993-2000, dan menggunakannya untuk membuat jaringan Internet pendidikan yang pertama di Indonesia. Dia membebaskan frekuensi WiFi, memperkenalkan RT/RW-net, antenna Wajanbolic dan jaringan selular OpenBTS dan private 5G sendiri. Dia memimpin jaringan telepon pertama di atas Internet, VoIP Merdeka, yang kemudian hari dikenal sebagai VoIP Rakyat berbasis SIP dan menggunakan kode area +62520 dan +62521. Dia saat ini aktif memperkenalkan e-Learning, dan menjalankan server e-Learning di http://lms.onnocenter.or.id/moodle/ 66,000++ siswa / mahasiswa dan https://opencourse.itts.ac.id 39.000+ mahasiswa secara gratis. Email: onno@indo.net.id