

BAB 7: Evaluasi, Monitoring, dan Pemeliharaan Solusi AI

7.1 Pendahuluan

Pada bab-bab sebelumnya, pembaca telah diajak memahami AI dari pondasinya sampai pada konteks penerapannya di lingkungan pemerintah. Bab 1 membangun pemahaman dasar tentang AI dalam transformasi digital sektor publik. Bab 2 menjelaskan jenis solusi AI, data, prompting, dan agentic AI. Bab 3 membantu memilih use case yang layak. Bab 4 membahas bagaimana solusi AI dirancang dan diintegrasikan ke dalam proses bisnis dan aplikasi organisasi. Bab 5 memberi pagar utama berupa keamanan, etika, risiko, dan tata kelola. Bab 6 kemudian menempatkan AI dalam konteks kepatuhan, kebijakan nasional, SPBE, interoperabilitas, dan tata kelola data di ekosistem pemerintah Indonesia.

Setelah seluruh pondasi itu dibangun, kini pembahasan bergerak ke pertanyaan yang sangat penting, tetapi sering kurang mendapat perhatian, yaitu: **bagaimana memastikan solusi AI tetap berguna, tetap terkendali, dan tetap layak dipakai setelah diimplementasikan?** Inilah fokus utama Bab 7.

Pertanyaan ini sangat penting karena dalam banyak organisasi, perhatian terbesar sering tercurah pada tahap awal: memilih use case, membangun sistem, melakukan integrasi, lalu meluncurkannya. Setelah itu, ada kecenderungan menganggap pekerjaan utama sudah selesai. Padahal, untuk AI, keadaan justru sering berbalik. Tantangan besar sering muncul **setelah** sistem mulai digunakan dalam kerja nyata. Pada tahap inilah organisasi mulai melihat:

- apakah hasil AI benar-benar membantu pengguna;
- apakah kualitasnya tetap stabil;
- apakah ada jenis kasus tertentu yang sering gagal;
- apakah data atau perilaku pengguna berubah;
- apakah output AI mulai menurun kualitasnya;
- apakah prompt yang semula efektif tetap relevan;
- apakah agent bekerja sesuai batas yang ditetapkan;
- dan apakah sistem masih pantas dipakai dalam konteks yang sama.

Dengan kata lain, AI tidak cukup hanya **dibangun**. AI harus **dievaluasi, dimonitor, dan dipelihara**.

Hal ini sejalan dengan acuan standar internasional. **ISO/IEC 22989** menegaskan bahwa siklus hidup sistem AI tidak berhenti pada desain, pengembangan, dan deployment. Standar tersebut menempatkan **operation and monitoring** sebagai tahap penting, lalu menegaskan perlunya **continuous validation** dan **re-evaluation** karena sistem AI cenderung lebih iteratif, lebih dinamis, dan kurang sepenuhnya dapat diprediksi dibanding sistem non-AI biasa. Standar yang sama juga menyebut bahwa hasil operasi perlu dievaluasi terhadap tujuan dan risiko, dan bahwa data uji yang representatif perlu diperbarui bila kondisi input berubah.

Kerangka yang serupa juga terlihat pada **ISO/IEC 23053**, yang menggambarkan AI berbasis machine learning sebagai sistem yang berjalan melalui suatu *pipeline* dari perumusan masalah, persiapan data, pemodelan, verifikasi dan validasi, deployment, hingga operation. Standar tersebut juga menekankan konsep-konsep seperti *drift*, *retraining*, *generalization*, dan pentingnya pemantauan pasca deployment sebagai bagian dari realitas operasional AI.

Pada tingkat kompetensi kerja, arah yang sama juga tampak dalam SKKNI AI. Unit kompetensi mengenai **mengintegrasikan** dan **merawat solusi AI** menekankan bahwa setelah sistem diintegrasikan dan diuji, organisasi perlu memastikan bahwa hasil monitoring sesuai dengan kebutuhan perawatan komponen arsitektur solusi AI, melakukan penanganan masalah berdasarkan log dan keluaran monitoring, serta mendokumentasikan aktivitas perawatan sesuai standar dokumentasi yang berlaku.

Dari sini terlihat bahwa AI bukan produk sekali jadi. AI adalah **kapabilitas yang hidup di dalam organisasi**. Ia dipengaruhi oleh:

- perubahan data;
- perubahan dokumen sumber;
- perubahan kebijakan;
- perubahan kebutuhan pengguna;
- perubahan model atau vendor;
- perubahan prompt;
- perubahan workflow;
- dan perubahan risiko yang baru terlihat setelah sistem dipakai.

Karena itu, Bab 7 sangat penting bagi peserta lintas bidang. Banyak orang mengira evaluasi dan monitoring AI adalah pekerjaan yang terutama milik tim teknis. Anggapan ini hanya sebagian benar. Memang, tim teknis punya peran penting dalam memantau performa sistem, latency, error, drift, atau kebutuhan retraining. Namun, evaluasi AI juga menyangkut hal-hal yang hanya bisa dibaca dengan baik oleh pemilik proses, pengguna, pimpinan, dan pengelola tata kelola, misalnya:

- apakah AI masih memberi manfaat nyata;
- apakah pengguna merasa terbantu atau justru terbebani;
- apakah kualitas informasi yang dihasilkan masih cukup baik;
- apakah fairness tetap terjaga;
- apakah agentic workflow masih berada dalam batas yang dapat diterima;
- dan apakah sistem masih layak dipertahankan dalam bentuk yang sama.

Dengan demikian, evaluasi, monitoring, dan pemeliharaan AI adalah pekerjaan **lintas fungsi**.

7.1.1 Evaluasi AI

Salah satu hal penting yang perlu dipahami sejak awal bab ini adalah bahwa evaluasi AI tidak boleh dibatasi hanya pada ukuran teknis. Memang, aspek teknis seperti:

- akurasi,
- relevansi,
- waktu respons,
- error rate,
- drift,
- atau stabilitas sistem

sangat penting. Namun, bagi organisasi pemerintah, AI tidak diimplementasikan hanya untuk mencapai metrik teknis. AI diterapkan karena diharapkan memberi nilai bagi proses, layanan, dan kerja organisasi.

Karena itu, evaluasi AI perlu mencakup dua lapisan besar sekaligus:

- **evaluasi teknis**, untuk melihat apakah sistem bekerja sesuai fungsi dan performa yang diharapkan;
- **evaluasi manfaat organisasi**, untuk melihat apakah sistem benar-benar membantu tujuan organisasi.

Misalnya, sebuah knowledge assistant bisa saja cukup baik secara teknis dalam menjawab pertanyaan, tetapi bila pegawai tetap harus membuka banyak dokumen karena tidak percaya pada hasilnya, maka manfaat organisasionalnya mungkin rendah. Sebaliknya, sebuah sistem ringkasan mungkin tidak sempurna secara linguistik, tetapi bila sangat membantu orientasi awal pimpinan dan staf, maka nilainya tetap besar.

Karena itu, Bab 7 akan berulang kali menekankan bahwa AI perlu dilihat dari sisi **kualitas teknis** dan **kegunaan nyata**.

Setelah AI dipakai dalam pekerjaan sehari-hari, organisasi perlu membangun **monitoring**. Monitoring di sini tidak hanya berarti melihat apakah server hidup atau apakah aplikasi bisa dibuka. Monitoring AI perlu menjawab pertanyaan seperti:

- apakah kualitas keluaran tetap stabil;
- apakah ada pola kegagalan baru;
- apakah latency meningkat;
- apakah sistem disalahgunakan;
- apakah ada insiden terkait data, prompt, atau vendor;
- dan apakah perilaku agent atau output generative AI mulai menyimpang dari batas semula.

Monitoring penting karena lingkungan operasional selalu lebih kompleks daripada pengujian awal. Data masuk bisa lebih berantakan. Pertanyaan pengguna bisa lebih ambigu. Variasi dokumen bisa lebih luas. Agent bisa menghadapi situasi yang tidak pernah dibayangkan sebelumnya. Tanpa monitoring, organisasi akan terlambat menyadari bahwa sistem sebenarnya mulai kehilangan kualitas atau mulai menimbulkan risiko.

Dalam software biasa, pemeliharaan sering dipahami sebagai memperbaiki bug, memperbarui aplikasi, atau menyesuaikan fitur. Pada AI, pemeliharaan mencakup lebih banyak unsur. Yang mungkin perlu dipelihara bukan hanya aplikasi, tetapi juga:

- data dan sumber dokumen;
- metadata;
- model;
- pipeline;
- prompt;
- retrieval corpus;
- agent workflow;
- integrasi API;
- logika guardrail;
- serta dokumentasi perubahan.

Ini penting karena AI sangat dipengaruhi oleh kualitas pondasi yang menopangnya. Sebuah sistem berbasis dokumen bisa menurun kualitasnya hanya karena repositori sumber tidak diperbarui. Sistem generative AI bisa memburuk hasilnya karena prompt tidak lagi sesuai konteks. Agentic AI bisa menjadi lebih berisiko karena workflow-nya tidak lagi sejalan dengan proses organisasi yang telah berubah.

Dengan kata lain, pemeliharaan AI adalah pemeliharaan **ekosistem komponen**, bukan hanya pemeliharaan aplikasi.

Judul fokus bab ini, yaitu “menjaga AI tetap berguna dan terkendali”, sangat penting untuk dipahami. Dua kata ini—**berguna** dan **terkendali**—perlu selalu berjalan bersama.

AI yang berguna tetapi tidak terkendali dapat:

- membocorkan data,
- menghasilkan output yang menyesatkan,
- memperkuat over-automation,
- atau menimbulkan ketergantungan yang sulit diawasi.

Sebaliknya, AI yang terlalu dikendalikan tetapi tidak lagi berguna juga bermasalah, karena organisasi mengeluarkan perhatian, sumber daya, dan biaya untuk sistem yang sebenarnya tidak memberi nilai yang cukup.

Karena itu, organisasi perlu menjaga keseimbangan:

- AI harus tetap memberi manfaat nyata;
- tetapi manfaat itu harus tetap berada dalam batas pengawasan, risiko, dan tata kelola yang dapat diterima.

Inilah yang membuat evaluasi, monitoring, dan pemeliharaan menjadi sangat strategis. Ketiganya membantu organisasi memutuskan apakah sistem:

- masih layak dipakai;
- perlu diperbaiki;
- perlu dibatasi;
- perlu diperbarui;
- atau bahkan perlu dihentikan.

7.2 Evaluasi teknis dan evaluasi manfaat organisasi

Setelah pada bagian pendahuluan dijelaskan bahwa AI tidak cukup hanya dibangun lalu dibiarkan berjalan, langkah berikutnya adalah memahami **apa yang sebenarnya perlu dievaluasi** dari sebuah solusi AI. Pada titik ini, banyak organisasi langsung memikirkan ukuran-ukuran teknis seperti akurasi, kecepatan respons, atau tingkat kesalahan. Semua itu memang penting. Namun, untuk organisasi pemerintah, evaluasi AI tidak boleh berhenti di sana. Sebuah sistem bisa saja tampak baik secara teknis, tetapi belum tentu memberi manfaat yang cukup bagi proses kerja, layanan, atau pengambilan keputusan awal. Sebaliknya, ada sistem yang secara teknis belum sempurna, tetapi tetap memberi nilai karena sangat membantu pekerjaan nyata.

Karena itu, evaluasi AI perlu dibangun di atas dua lapisan yang saling melengkapi, yaitu:

- **evaluasi teknis**, untuk menilai apakah sistem bekerja dengan mutu yang cukup;
- dan **evaluasi manfaat organisasi**, untuk menilai apakah sistem benar-benar membantu tujuan organisasi.

Pembedaan ini sangat penting. Tanpa evaluasi teknis, organisasi tidak akan tahu apakah sistem cukup andal, cukup relevan, dan cukup stabil. Tanpa evaluasi manfaat organisasi, organisasi bisa terjebak pada situasi di mana AI tampak “bagus” di atas kertas, tetapi tidak benar-benar dipakai, tidak mengurangi beban kerja, tidak memperbaiki layanan, atau tidak mendukung sasaran organisasi yang lebih besar.

Dalam banyak proyek teknologi, ada kecenderungan menganggap bahwa jika sistem sudah mencapai performa teknis tertentu, maka sistem otomatis dianggap berhasil. Pada AI, anggapan ini kurang memadai. Hal ini karena AI diterapkan bukan untuk menunjukkan bahwa model bisa bekerja, tetapi untuk membantu organisasi mencapai tujuan tertentu.

Sebagai contoh:

- sebuah knowledge assistant mungkin punya tingkat relevansi jawaban yang cukup tinggi, tetapi bila pegawai tetap lebih memilih mencari manual karena tidak percaya pada hasilnya, maka manfaat organisasionalnya rendah;
- sebuah sistem ringkasan bisa menghasilkan teks yang cukup baik, tetapi bila formatnya tidak sesuai kebutuhan pimpinan atau staf, maka manfaatnya terbatas;
- sebuah sistem klasifikasi aduan bisa menunjukkan performa model yang baik pada data uji, tetapi bila petugas tetap harus mengoreksi hampir semua hasil dalam praktik, maka nilai operasionalnya perlu dipertanyakan.

Sebaliknya, ada juga situasi di mana sistem belum sempurna secara teknis, tetapi manfaatnya tetap nyata. Misalnya:

- sistem ringkasan belum selalu menghasilkan narasi yang ideal, tetapi cukup membantu staf mempercepat orientasi awal terhadap dokumen panjang;
- retrieval assistant belum selalu memberi jawaban terbaik, tetapi sangat membantu pegawai menemukan sumber awal yang benar;
- atau drafting awal masih perlu banyak penyuntingan, tetapi sudah mengurangi waktu kerja repetitif secara cukup berarti.

Karena itu, evaluasi teknis dan evaluasi manfaat organisasi harus dibedakan, tetapi tidak dipisahkan. Keduanya perlu dibaca bersama.

7.2.1 Evaluasi teknis

Evaluasi teknis adalah penilaian terhadap mutu fungsi dan performa sistem AI dari sisi teknis. Tujuannya adalah melihat apakah solusi AI:

- bekerja sesuai fungsi yang dirancang;
- menghasilkan keluaran yang cukup tepat atau cukup relevan;
- cukup stabil dalam berbagai kondisi input;
- cukup cepat untuk kebutuhan operasional;
- dan cukup dapat diandalkan untuk konteks penggunaan yang telah ditentukan.

Evaluasi teknis tentu berbeda tergantung jenis use case. Pada sistem klasifikasi, organisasi mungkin perlu melihat:

- ketepatan kategori;
- konsistensi hasil;
- pola kesalahan antar kelas;
- dan performa pada variasi dokumen yang berbeda.

Pada sistem retrieval atau knowledge assistant, organisasi mungkin perlu melihat:

- relevansi dokumen yang ditemukan;
- kualitas jawaban terhadap pertanyaan;
- ketepatan sumber rujukan;
- dan kemampuan sistem menangani pertanyaan yang ambigu.

Pada sistem ringkasan, organisasi mungkin perlu melihat:

- kesetiaan terhadap sumber;
- kelengkapan unsur penting;
- struktur keluaran;
- dan apakah hasil terlalu panjang, terlalu pendek, atau terlalu bebas.

Pada sistem generative AI untuk drafting, evaluasi teknis bisa mencakup:

- kesesuaian format;
- kelengkapan unsur yang diminta;
- konsistensi gaya;
- dan tingkat kesalahan faktual atau tambahan isi yang tidak perlu.

Pada agentic AI, evaluasi teknis juga perlu melihat:

- apakah langkah-langkah workflow dijalankan sesuai batas;
- apakah agent memilih sumber yang benar;
- apakah keluaran antara cukup masuk akal;
- dan apakah tindakan agent tetap berada dalam ruang kerja yang dirancang.

Dengan demikian, evaluasi teknis bukan hanya satu angka, tetapi serangkaian pengamatan terhadap kualitas sistem dari sisi fungsi dan performanya.

Satu hal yang sangat penting adalah bahwa evaluasi teknis harus **kontekstual**. Artinya, ukuran teknis yang dipakai harus sesuai dengan tujuan dan risiko use case yang sedang dibangun. Organisasi tidak sebaiknya memakai metrik hanya karena metrik itu populer atau mudah dihitung.

Sebagai contoh:

- untuk use case klasifikasi surat masuk, yang terpenting mungkin bukan sekadar akurasi umum, tetapi apakah kategori yang paling penting tertangani dengan baik;
- untuk retrieval dokumen, yang penting bukan hanya apakah sistem memberi jawaban, tetapi apakah jawaban itu mengarahkan pengguna ke sumber resmi yang tepat;
- untuk ringkasan briefing, yang penting bukan hanya panjang teks, tetapi apakah hasil cukup membantu orientasi awal dan tidak menghilangkan hal yang penting;
- untuk agentic AI, yang penting bukan hanya apakah output akhirnya bagus, tetapi juga apakah alur langkahnya tetap terkendali.

Dengan kata lain, evaluasi teknis perlu mengikuti fungsi sistem, bukan sebaliknya.

7.2.2 Evaluasi manfaat organisasi

Jika evaluasi teknis melihat mutu kerja sistem, maka **evaluasi manfaat organisasi** melihat apakah AI benar-benar membantu organisasi mencapai perubahan yang diharapkan. Pertanyaan utamanya bukan lagi “apakah sistem bekerja dengan baik”, tetapi “apakah keberadaan sistem ini membuat organisasi bekerja lebih baik”.

Evaluasi manfaat organisasi dapat mencakup hal-hal seperti:

- apakah waktu kerja untuk tugas tertentu berkurang;
- apakah beban kerja repetitif menurun;
- apakah akses terhadap informasi menjadi lebih cepat;
- apakah kualitas orientasi awal terhadap dokumen meningkat;
- apakah konsistensi proses membaik;
- apakah pengguna merasa lebih terbantu;
- apakah layanan menjadi lebih tertata;
- dan apakah pimpinan atau unit kerja memperoleh nilai tambah yang nyata dari sistem.

Pada sektor publik, manfaat organisasi juga dapat mencakup:

- peningkatan kualitas tata kelola;
- penguatan dokumentasi dan pengetahuan internal;
- peningkatan keteraturan administrasi;
- peningkatan kesiapan analisis awal;
- atau peningkatan koordinasi dan akses terhadap informasi yang benar.

Yang penting dipahami adalah bahwa **manfaat organisasi tidak selalu identik dengan penghematan waktu saja**. Dalam banyak kasus, manfaat yang paling besar justru berupa:

- berkurangnya kebingungan;
- meningkatnya konsistensi penggunaan dokumen resmi;
- meningkatnya kemampuan pegawai menemukan informasi yang tepat;
- atau membaiknya kualitas langkah awal sebelum analisis atau keputusan manusia dilakukan.

Dibanding evaluasi teknis, manfaat organisasi sering lebih sulit diukur karena:

- tidak selalu muncul dalam bentuk angka yang sederhana;
- bisa bergantung pada persepsi dan kebiasaan pengguna;
- dan kadang baru terasa setelah sistem dipakai cukup lama.

Misalnya, organisasi bisa dengan mudah menghitung waktu respons sistem, tetapi lebih sulit menilai apakah:

- sistem benar-benar mengurangi beban kognitif pengguna;
- kualitas briefing pimpinan menjadi lebih baik;
- atau pegawai merasa lebih yakin menggunakan sumber yang resmi.

Meski demikian, manfaat organisasi tetap harus diupayakan untuk dievaluasi. Kalau tidak, organisasi akan terlalu mudah terjebak pada metrik teknis dan lupa pada tujuan awal penggunaan AI.

Karena itu, manfaat organisasi dapat dinilai melalui kombinasi:

- indikator kuantitatif;
- pengamatan terhadap perubahan proses;
- umpan balik pengguna;
- dan penilaian terhadap dampak pada kerja nyata.

7.2.3 Perbandingan evaluasi teknis dan manfaat organisasi

Agar lebih jelas, perhatikan beberapa contoh berikut.

Contoh 1: knowledge assistant internal

Evaluasi teknis dapat melihat:

- relevansi jawaban;
- ketepatan dokumen sumber;
- waktu respons;
- stabilitas sistem.

Evaluasi manfaat organisasi dapat melihat:

- apakah pegawai lebih cepat menemukan informasi;
- apakah pertanyaan berulang ke unit tertentu berkurang;
- apakah penggunaan dokumen resmi menjadi lebih konsisten;
- apakah pengguna merasa terbantu dalam pekerjaan sehari-hari.

Contoh 2: sistem ringkasan bahan briefing

Evaluasi teknis dapat melihat:

- kesetiaan terhadap sumber;
- struktur hasil;
- kelengkapan poin penting;
- dan adanya tambahan informasi yang tidak seharusnya.

Evaluasi manfaat organisasi dapat melihat:

- apakah staf lebih cepat menyiapkan bahan;
- apakah pimpinan lebih mudah memperoleh orientasi awal;
- apakah waktu telaah awal berkurang;
- apakah kualitas diskusi awal meningkat.

Contoh 3: klasifikasi surat masuk atau aduan

Evaluasi teknis dapat melihat:

- akurasi kategori;
- pola kesalahan;
- performa pada variasi dokumen.

Evaluasi manfaat organisasi dapat melihat:

- apakah triase lebih cepat;
- apakah distribusi awal lebih tertib;
- apakah beban kerja manual berkurang;
- apakah proses tindak lanjut menjadi lebih terarah.

Dari contoh-contoh ini terlihat jelas bahwa satu sistem dapat memiliki hasil teknis tertentu, tetapi nilai organisasionalnya tetap perlu dibaca secara terpisah.

Agar evaluasi manfaat organisasi tidak kabur, organisasi perlu kembali pada **sasaran awal use case**. Pada Bab 3 telah dibahas pentingnya membedakan sasaran bisnis dan sasaran teknis. Bab ini menegaskan kembali hal tersebut.

Jika sejak awal sasaran organisasionalnya adalah:

- mempercepat pencarian informasi,
- mengurangi beban membaca,
- meningkatkan keteraturan triase,
- atau memperkuat orientasi awal terhadap dokumen,
maka evaluasi manfaat harus melihat apakah sasaran itu benar-benar tercapai.

Tanpa hubungan ke sasaran awal, evaluasi manfaat akan mudah berubah menjadi penilaian yang terlalu umum seperti “pengguna suka” atau “sistem terlihat berguna”, tanpa kejelasan apa sebenarnya yang diperbaiki.

Karena kedua jenis evaluasi ini berbeda, pihak yang terlibat juga sering berbeda. Evaluasi teknis biasanya lebih banyak melibatkan:

- tim teknis;
- pengembang;
- pengelola model atau integrasi;
- dan pihak yang memahami perilaku sistem secara operasional.

Sementara itu, evaluasi manfaat organisasi sering perlu melibatkan:

- pemilik proses;
- pengguna akhir;
- pimpinan unit;
- dan pihak yang merasakan langsung perubahan dalam kerja.

Ini penting karena tim teknis mungkin tahu apakah model membaik, tetapi belum tentu tahu apakah pengguna merasa pekerjaan mereka lebih ringan. Sebaliknya, pengguna bisa merasa sistem membantu, tetapi tidak selalu tahu apakah sistem mulai mengalami drift atau penurunan kualitas tertentu.

Dalam evaluasi AI, organisasi sering bertanya apakah sebaiknya memakai indikator kuantitatif atau kualitatif. Jawaban yang lebih tepat adalah: **keduanya penting**.

Indikator kuantitatif berguna untuk melihat pola yang lebih objektif dan konsisten, misalnya:

- waktu respons;
- tingkat koreksi;
- jumlah penggunaan;
- tingkat keberhasilan jawaban;
- atau lama waktu penyelesaian tugas tertentu.

Namun, indikator kuantitatif tidak selalu cukup. AI juga perlu dibaca melalui penilaian kualitatif, misalnya:

- apakah hasil AI benar-benar terasa membantu;
- apakah pengguna merasa lebih percaya diri atau justru lebih bingung;
- apakah hasil AI sesuai dengan kebutuhan pimpinan atau unit;
- apakah ada konteks tertentu yang sering tidak tertangani dengan baik.

Pendekatan gabungan ini sangat penting terutama di sektor publik, karena banyak nilai organisasi tidak sepenuhnya tercermin dalam satu angka tunggal.

Tujuan evaluasi bukan hanya untuk “menilai”, tetapi untuk **membantu keputusan tindak lanjut**. Dari evaluasi teknis dan manfaat organisasi, instansi dapat memutuskan:

- apakah sistem cukup baik untuk dilanjutkan;
- apakah perlu perbaikan pada data, model, atau prompt;
- apakah perlu pelatihan pengguna;
- apakah use case perlu dipersempit;
- apakah perlu kontrol tambahan;
- atau apakah manfaatnya ternyata terlalu kecil untuk membenarkan pemeliharaan lebih lanjut.

Dengan kata lain, evaluasi yang baik menjadi dasar pengelolaan sistem sepanjang siklus hidupnya.

Evaluasi teknis dan evaluasi manfaat organisasi adalah dua lapisan utama dalam penilaian solusi AI. Evaluasi teknis melihat mutu kerja sistem dari sisi fungsi, performa, stabilitas, dan kualitas keluaran. Evaluasi manfaat organisasi melihat apakah sistem benar-benar membantu tujuan organisasi, memperbaiki proses, mengurangi beban kerja, meningkatkan akses informasi, atau memberi nilai yang nyata bagi pengguna dan pimpinan.

7.3 Fairness, explainability, dan reliability

Setelah pada subbagian sebelumnya dibahas bahwa solusi AI perlu dievaluasi dari dua sisi sekaligus—yaitu sisi teknis dan sisi manfaat organisasi—langkah berikutnya adalah memperjelas tiga kualitas penting yang sangat menentukan apakah AI layak terus dipakai dalam organisasi, yaitu **fairness**, **explainability**, dan **reliability**. Ketiga aspek ini penting karena organisasi pemerintah tidak cukup hanya mengetahui bahwa sistem “berfungsi”. Yang lebih penting adalah apakah sistem:

- memperlakukan variasi kasus secara cukup adil;
- dapat dijelaskan dengan tingkat yang memadai;
- dan tetap cukup andal dalam penggunaan nyata.

Dalam konteks sektor publik, tiga hal ini memiliki bobot yang besar. AI bisa saja cepat dan tampak membantu, tetapi bila hasilnya cenderung merugikan jenis kasus tertentu, sulit dipahami oleh pengguna, atau terlalu mudah berubah kualitasnya dari waktu ke waktu, maka organisasi akan sulit mempertahankan kepercayaan terhadap sistem tersebut. Karena itu, evaluasi AI yang matang perlu bergerak lebih jauh daripada sekadar angka performa umum. Organisasi perlu melihat **siapa yang mungkin dirugikan, apa yang bisa dijelaskan, dan seberapa jauh sistem dapat diandalkan**.

Hal ini sejalan dengan kerangka etika nasional. **Konsep Pedoman Etika Kecerdasan Artifisial** menekankan bahwa AI perlu dilaksanakan secara bertanggung jawab, adil, transparan, menghormati privasi, menjamin keamanan, dan memungkinkan pengawasan manusia. Dokumen tersebut juga secara eksplisit menempatkan pertanyaan tentang deteksi dan mitigasi bias, skema pertanggungjawaban, serta kemampuan menjelaskan proses pembuatan keputusan AI kepada pengguna sebagai bagian dari penilaian etika sistem. Selain itu, dokumen yang sama menekankan perlunya pemantauan dan evaluasi berkala, langkah perlindungan, serta transparansi dan akuntabilitas sistem yang dioperasikan.

Pada tingkat standar internasional, **ISO/IEC 22989** menegaskan bahwa hasil operasi AI perlu dievaluasi terhadap tujuan dan risiko yang telah diidentifikasi, serta bahwa pembaruan dapat dilakukan untuk meningkatkan **performance and reliability**. Standar ini juga menegaskan bahwa AI systems do not understand dan tetap memerlukan desain, rekayasa, dan pengawasan manusia sepanjang siklus hidupnya. Sementara itu, **ISO/IEC 23053** menjelaskan konsep **generalization**, overfitting, data drift, concept drift, retraining, dan guardrails sebagai bagian dari bagaimana model AI dapat atau tidak dapat mempertahankan performanya pada data yang belum pernah dilihat sebelumnya.

Dengan landasan tersebut, fairness, explain ability, dan reliability dapat dipahami sebagai tiga pertanyaan evaluatif yang saling melengkapi:

- **fairness** menanyakan apakah AI menghasilkan pola yang cukup wajar dan tidak merugikan konteks tertentu secara tidak proporsional;
- **explain ability** menanyakan apakah penggunaan, keluaran, dan dasar kerja AI dapat dijelaskan pada tingkat yang diperlukan;
- **reliability** menanyakan apakah AI tetap cukup konsisten, stabil, dan layak diandalkan dalam konteks operasionalnya.

7.3.1 Fairness

7.3.1.1 Fairness sebagai kualitas evaluasi, bukan asumsi

Pada Bab 5 telah dibahas fairness dalam kaitannya dengan etika dan nilai publik. Pada bab ini, fairness perlu dibaca lebih khusus sebagai **kualitas yang harus dievaluasi secara terus-menerus**. Artinya, organisasi tidak boleh menganggap bahwa fairness sudah selesai diperiksa hanya karena sistem tampak berjalan baik pada uji awal.

Fairness dalam evaluasi AI berarti organisasi perlu bertanya:

- apakah sistem bekerja cukup baik pada berbagai jenis input yang relevan;
- apakah ada kategori, konteks, kelompok pengguna, atau jenis dokumen tertentu yang secara konsisten mendapatkan hasil yang lebih buruk;
- dan apakah pola hasil AI dapat menciptakan perlakuan yang tidak wajar atau tidak proporsional dalam workflow organisasi.

Dalam konteks sektor publik, fairness tidak selalu berarti statistik yang rumit. Kadang fairness dapat dibaca dari pertanyaan yang sederhana tetapi penting, misalnya:

- apakah sistem klasifikasi lebih sering salah pada variasi bahasa tertentu;
- apakah ringkasan lebih sering menonjolkan sumber yang dominan dan mengabaikan sumber yang lebih halus tetapi penting;
- apakah retrieval lebih mudah menemukan dokumen dari unit tertentu karena struktur datanya lebih rapi, sementara sumber lain yang relevan tertinggal;
- atau apakah agentic workflow lebih sering gagal pada kasus pengecualian yang justru penting bagi organisasi.

7.3.1.2 Fairness perlu diuji pada variasi kasus, bukan hanya rata-rata

Salah satu kelemahan umum dalam evaluasi AI adalah terlalu bergantung pada ukuran rata-rata. Sebuah sistem bisa memiliki performa umum yang tampak baik, tetapi menyimpan kelemahan serius pada jenis kasus tertentu. Karena itu, fairness perlu diuji pada **variasi kasus**, bukan hanya pada agregat umum.

Misalnya, untuk sistem klasifikasi:

- lihat bukan hanya akurasi keseluruhan, tetapi juga pola salah kategori per kelas;
- lihat apakah kategori minoritas cenderung lebih sering salah;
- lihat apakah variasi format dokumen mempengaruhi hasil secara tidak proporsional.

Untuk sistem retrieval:

- lihat apakah semua jenis sumber yang resmi punya peluang wajar untuk ditemukan;
- lihat apakah pertanyaan tertentu selalu diarahkan ke sumber yang sama walaupun sebenarnya ada sumber lain yang lebih tepat.

Untuk sistem generative AI:

- lihat apakah gaya bahasa, struktur input, atau panjang dokumen tertentu membuat hasil menjadi jauh lebih buruk;
- lihat apakah sistem lebih sering “percaya diri” pada konteks yang sebenarnya tidak cukup didukung sumber.

Fairness seperti ini sangat relevan untuk organisasi pemerintah karena proses kerja birokrasi sering berhadapan dengan variasi bentuk dokumen, variasi konteks layanan, dan variasi intensitas informasi yang tidak selalu merata.

7.3.1.3 Fairness juga terkait dengan penggunaan hasil

Fairness tidak hanya berada pada level model, tetapi juga pada level **bagaimana hasil AI dipakai**. Sebuah hasil yang sekadar memberi usulan mungkin punya implikasi fairness yang berbeda dibanding hasil yang langsung mempengaruhi alur tindak lanjut. Karena itu, organisasi perlu melihat bukan hanya apakah model fair, tetapi juga:

- apakah pengguna terlalu percaya pada jenis hasil tertentu;
- apakah workflow memberi ruang cukup untuk koreksi;
- apakah kasus pengecualian tetap dapat naik ke pemeriksaan manusia;
- dan apakah organisasi memiliki mekanisme untuk menangkap keluhan atau koreksi bila hasil AI dianggap tidak adil.

7.3.2 explainability

explainability dapat dipahami sebagai kemampuan organisasi untuk menjelaskan, pada tingkat yang memadai, bagaimana AI digunakan, dasar apa yang dipakai, apa arti hasilnya, dan bagaimana hasil itu seharusnya ditafsirkan. explainability bukan selalu berarti menjelaskan seluruh detail matematis model. Dalam banyak konteks organisasi pemerintah, yang lebih penting adalah bahwa:

- pengguna memahami fungsi AI;
- pengelola memahami ruang lingkup dan keterbatasan sistem;
- pimpinan memahami bagaimana hasil AI dipakai dalam proses;
- dan bila perlu, ada penjelasan yang memadai tentang dasar keluaran tertentu.

Konsep Pedoman Etika AI secara langsung mengangkat pertanyaan **“Apakah proses pembuatan keputusan oleh KA dapat dijelaskan kepada pengguna?”** sebagai bagian dari penilaian etika. Dokumen itu juga menampilkan spektrum dari tidak ada transparansi, dokumentasi minimal, penjelasan internal, sampai penjelasan publik. Ini menunjukkan bahwa explainability tidak bersifat hitam-putih; ada tingkat kedalaman penjelasan yang berbeda sesuai konteks.

7.3.2.1 explainability berbeda menurut use case

Tingkat explainability yang dibutuhkan tidak selalu sama untuk semua use case. Untuk sistem ringkasan internal, explainability yang memadai mungkin berarti:

- pengguna tahu bahwa hasil adalah ringkasan bantuan awal;
- sumber dokumen tersedia;
- dan ada kejelasan bahwa hasil perlu diverifikasi bila akan dipakai lebih lanjut.

Untuk sistem knowledge assistant, explainability yang memadai mungkin berarti:

- jawaban terhubung ke sumber rujukan;
- pengguna dapat membuka dokumen asli;
- dan sistem cukup jelas membedakan antara jawaban berbasis sumber dan penjelasan generatif.

Untuk sistem klasifikasi, explainability mungkin berarti:

- kategori hasil terlihat jelas;
- ada alasan atau indikator yang membantu petugas memahami mengapa sistem mengusulkan kategori tertentu;
- dan petugas dapat mengoreksi hasil tanpa kebingungan.

Untuk agentic AI, explainability menjadi lebih penting lagi, karena organisasi perlu tahu:

- langkah apa saja yang dijalankan agent;
- sumber apa yang dipilih;
- bagian mana yang dihasilkan sendiri;
- dan di titik mana manusia harus masuk.

Dengan demikian, explainability selalu terkait dengan **apa yang perlu dijelaskan, kepada siapa, dan untuk tujuan apa.**

7.3.2.2 explainability penting untuk akuntabilitas dan kepercayaan

Dalam sektor publik, explainability penting karena berkaitan langsung dengan:

- akuntabilitas;
- kepercayaan pengguna;
- kemampuan audit;
- dan kualitas pengambilan keputusan manusia.

Jika pengguna tidak memahami bagaimana membaca hasil AI, mereka bisa jatuh ke dua ekstrem:

- terlalu percaya pada hasil;
- atau sama sekali tidak mau memakainya.

Jika pimpinan atau pengelola tidak dapat menjelaskan bagaimana sistem dipakai, maka keputusan untuk mempertahankan atau memperluas sistem menjadi lemah. Jika organisasi tidak memiliki dokumentasi yang memadai, maka ketika terjadi masalah, sulit menjelaskan apa yang sebenarnya dilakukan sistem.

Karena itu, explainability bukan aksesori. Ia adalah bagian dari kelayakan operasional dan tata kelola AI.

7.3.2.3 explainability memerlukan dokumentasi dan desain antarmuka

Agar AI cukup dapat dijelaskan, organisasi tidak cukup hanya mengandalkan kemampuan lisan pengembang atau operator. explainability perlu dibangun melalui:

- **dokumentasi**, yang menjelaskan tujuan, ruang lingkup, sumber, model, prompt, batas penggunaan, dan hasil pengujian;
- **desain antarmuka**, yang memberi label yang jelas, menampilkan sumber bila relevan, dan memudahkan pengguna memahami status hasil AI;
- serta **pelatihan pengguna**, agar orang tahu kapan hasil dapat dipakai sebagai bantuan awal dan kapan harus diperiksa lebih jauh.

7.3.3 Reliability

Reliability dapat dipahami sebagai tingkat keandalan sistem AI dalam menghasilkan keluaran yang cukup konsisten, cukup stabil, dan cukup dapat dipercaya untuk konteks penggunaannya. Reliability tidak berarti sistem harus selalu sempurna. Yang lebih penting adalah bahwa organisasi tahu:

- seberapa stabil sistem bekerja;
- dalam kondisi apa kualitasnya menurun;
- seberapa sering error atau kegagalan muncul;
- dan apakah hasilnya cukup dapat diandalkan untuk peran yang diberikan kepadanya.

ISO/IEC 22989 menegaskan bahwa pada tahap operation and monitoring, AI perlu dipantau untuk operasi normal dan insiden, diperbaiki bila terjadi error, diperbarui untuk memenuhi kebutuhan baru dan untuk meningkatkan **performance and reliability**, serta dievaluasi kembali terhadap tujuan dan risiko yang telah ditetapkan. Ini menunjukkan bahwa reliability bukan hal yang otomatis muncul dari deployment awal; ia harus dijaga sepanjang operasi.

7.3.3.1 Reliability berkaitan dengan generalization dan drift

Dalam AI berbasis machine learning, reliability sangat erat dengan **generalization**, yaitu kemampuan model membuat prediksi atau keluaran yang dapat diterima pada data yang belum pernah dilihat sebelumnya. ISO/IEC 23053 menjelaskan bahwa model yang generalizes well adalah model yang memiliki akurasi prediksi yang dapat diterima pada input yang sebelumnya tidak terlihat. Standar yang sama juga menjelaskan bahwa overfitting akan membuat model gagal melakukan generalization dengan baik.

Selain itu, reliability juga dipengaruhi oleh:

- **data drift**, ketika karakteristik statistik data produksi berubah dari data yang dipakai untuk pengembangan;
- **concept drift**, ketika hubungan antara input dan target berubah;
- dan kebutuhan **retraining** atau guard-rails agar perilaku model tetap berada dalam batas yang dapat diterima.

Bagi organisasi pemerintah, ini berarti reliability tidak boleh dinilai hanya dari hasil uji awal. Sistem bisa tampak baik saat pengembangan, tetapi menurun ketika dipakai pada dokumen, pertanyaan, atau perilaku pengguna yang berbeda dari data awal.

7.3.3.2 Reliability juga mencakup stabilitas operasional

Selain kualitas model, reliability juga mencakup **stabilitas operasional**. Ini berarti organisasi perlu melihat:

- apakah sistem cukup tersedia saat dibutuhkan;
- apakah waktu respons tetap masuk akal;
- apakah integrasi antar komponen cukup stabil;
- apakah perubahan kecil pada input membuat hasil terlalu liar;
- dan apakah sistem tetap cukup baik setelah pembaruan tertentu.

Untuk generative AI dan retrieval-based systems, reliability juga berarti melihat apakah:

- format output konsisten;
- kualitas jawaban tidak berubah drastis hanya karena variasi prompt kecil;
- sumber tetap muncul dengan benar;
- dan sistem tidak terlalu sering gagal menemukan jawaban yang sebenarnya ada.

Untuk agentic AI, reliability mencakup:

- kestabilan langkah workflow;
- konsistensi pemilihan tool atau sumber;
- serta kemampuan sistem untuk berhenti atau fallback ketika kondisi tidak jelas.

7.3.3.3 Reliability harus dibaca sesuai peran sistem

Satu hal penting adalah bahwa reliability harus selalu dibaca sesuai **peran sistem**. Jika sistem hanya dipakai sebagai alat bantu orientasi awal, maka tingkat reliability yang dibutuhkan berbeda dibanding bila sistem dipakai sebagai komponen penting dalam alur kerja yang lebih sensitif.

Misalnya:

- untuk ringkasan awal internal, reliability mungkin cukup bila hasil biasanya membantu dan kesalahannya mudah dikenali manusia;
- untuk knowledge assistant yang dipakai luas, reliability perlu lebih tinggi karena banyak pengguna akan bergantung pada sistem untuk navigasi informasi;
- untuk agentic workflow yang menghubungkan beberapa langkah kerja, reliability perlu lebih kuat karena satu kesalahan dapat merambat ke tahap berikutnya.

Karena itu, organisasi sebaiknya tidak menilai reliability secara abstrak, tetapi mengaitkannya dengan **fungsi dan dampak** sistem.

Fairness, explainability, dan reliability adalah tiga kualitas penting yang membantu organisasi menilai apakah solusi AI benar-benar layak dipakai secara berkelanjutan. Fairness membantu melihat apakah hasil AI cukup wajar dan tidak merugikan konteks tertentu secara tidak proporsional. explainability membantu memastikan bahwa penggunaan dan hasil AI dapat dijelaskan pada tingkat yang memadai bagi pengguna, pengelola, dan organisasi. Reliability membantu menilai apakah sistem cukup stabil, cukup konsisten, dan cukup dapat diandalkan dalam konteks operasionalnya. Kerangka etika nasional menekankan keadilan, transparansi, akuntabilitas, dan pemantauan berkala, sementara standar internasional menegaskan pentingnya evaluasi hasil operasi, generalization, retraining, continuous validation, dan peningkatan reliability sepanjang siklus hidup AI.

7.4 Monitoring drift, latency, misuse, dan incident

Setelah pada subbagian sebelumnya dibahas bahwa solusi AI perlu dinilai dari sisi **fairness**, **explainability**, dan **reliability**, langkah berikutnya adalah melihat apa yang perlu **dipantau secara terus-menerus** ketika sistem sudah berjalan dalam lingkungan organisasi. Di sinilah konsep **monitoring** menjadi sangat penting.

Monitoring pada AI tidak boleh dipahami secara sempit hanya sebagai pemantauan apakah server aktif atau apakah aplikasi bisa diakses. Tentu saja aspek-aspek tersebut tetap penting, tetapi untuk AI, monitoring perlu bergerak lebih jauh. Organisasi perlu memantau apakah:

- kualitas sistem berubah dari waktu ke waktu;
- waktu responsnya tetap sesuai kebutuhan kerja;
- sistem digunakan sebagaimana mestinya;
- ada pola penyalahgunaan atau penggunaan di luar tujuan;
- dan ada insiden yang menunjukkan bahwa AI tidak lagi berjalan dalam batas yang aman, layak, dan dapat dipertanggungjawabkan.

Dalam praktik, empat hal yang sangat penting untuk dipantau adalah:

- **drift**,
- **latency**,
- **misuse**,
- dan **incident**.

Keempatnya perlu dibahas secara khusus karena seringkali masalah utama AI tidak muncul pada tahap desain awal, tetapi baru terlihat setelah sistem benar-benar dipakai oleh pengguna dalam situasi kerja nyata.

Dalam konteks standar internasional, monitoring bukan pekerjaan tambahan setelah implementasi, melainkan bagian inti dari siklus hidup AI. **ISO/IEC 22989** menempatkan **operation and monitoring** sebagai salah satu fase penting dalam AI system life cycle, lalu menegaskan perlunya **continuous validation** dan **re-evaluation** karena AI system bersifat lebih iteratif dan kurang sepenuhnya dapat diprediksi dibanding sistem non-AI biasa. Standar tersebut juga menjelaskan bahwa hasil operasi perlu dievaluasi terhadap tujuan dan risiko yang telah diidentifikasi, dan bahwa sistem dapat diperbarui untuk meningkatkan performa serta reliability.

Hal yang sejalan juga tampak pada SKKNI terkait pemeliharaan solusi AI, yang menegaskan bahwa hasil monitoring perlu dipastikan kesesuaiannya dengan kebutuhan perawatan komponen solusi AI, bahwa masalah perlu ditangani berdasarkan log dan keluaran monitoring, dan bahwa seluruh aktivitas perawatan harus didokumentasikan.

Dari sini terlihat bahwa monitoring AI bukan sekadar “melihat dashboard”, tetapi bagian dari mekanisme organisasi untuk menjaga agar sistem tetap:

- berguna,
- terkendali,
- dan selaras dengan tujuan awalnya.

Pada software biasa, monitoring sering berfokus pada hal-hal seperti:

- uptime,
- error rate,
- pemakaian CPU atau memori,
- dan keberhasilan transaksi.

Pada AI, semua itu masih penting, tetapi belum cukup. AI bekerja di atas data, model, prompt, dan perilaku pengguna yang dapat berubah. Akibatnya, sistem bisa tampak “sehat” dari sisi infrastruktur, tetapi sebenarnya kualitas hasilnya menurun atau risikonya meningkat.

Misalnya:

- server berjalan normal, tetapi retrieval mulai sering mengambil sumber yang kurang tepat;
- aplikasi responsif, tetapi prompt yang dulu efektif sekarang menghasilkan ringkasan yang lebih lemah;
- integrasi tidak rusak, tetapi agent mulai sering gagal pada variasi kasus yang baru;
- atau sistem tidak mengalami error teknis, tetapi pengguna mulai memakai AI di luar ruang lingkup yang dimaksudkan.

Karena itu, monitoring AI perlu mencakup:

- kondisi teknis sistem,
- kualitas perilaku sistem,
- pola penggunaan manusia,
- dan tanda-tanda bahwa risiko mulai meningkat.

7.4.1 Monitoring drift

Salah satu hal yang paling penting dipantau pada AI adalah **drift**. Secara umum, drift berarti bahwa sesuatu dalam lingkungan sistem berubah, sehingga kualitas atau perilaku AI juga ikut berubah. Pada AI berbasis machine learning, **ISO/IEC 23053** menjelaskan konsep seperti **data drift** dan **concept drift** sebagai perubahan yang dapat membuat sistem yang semula bekerja baik menjadi kurang akurat atau kurang sesuai terhadap data produksi yang baru.

Dalam konteks organisasi, drift dapat dipahami secara lebih praktis sebagai situasi ketika:

- karakteristik input nyata tidak lagi sama seperti saat sistem diuji;
- sumber dokumen berubah;
- istilah atau pola bahasa bergeser;
- kategori kerja berubah;
- konteks kebijakan berubah;
- atau perilaku pengguna berubah sehingga sistem menghadapi pola interaksi yang baru.

Dengan kata lain, drift adalah sinyal bahwa lingkungan kerja AI tidak lagi sama seperti saat sistem pertama kali dinilai layak. Dalam implementasi AI di organisasi pemerintah, drift bisa muncul dalam beberapa bentuk.

1. Drift pada data

Ini terjadi ketika data produksi berubah karakteristiknya. Misalnya:

- format dokumen menjadi berbeda;
- bahasa yang dipakai dalam aduan berubah;
- jenis pertanyaan pengguna bergeser;
- atau distribusi kategori kasus tidak lagi sama.

2. Drift pada sumber dokumen

Untuk sistem retrieval atau knowledge assistant, drift bisa terjadi ketika:

- dokumen baru ditambahkan;
- dokumen lama sudah tidak relevan tetapi masih ikut terbaca;
- metadata berubah;
- atau sumber resmi belum diperbarui.

3. Drift pada konteks kebijakan

Dalam organisasi pemerintah, kebijakan, prosedur, atau aturan dapat berubah. Jika AI tetap bekerja di atas dasar yang lama, maka sistem bisa mulai memberi keluaran yang kurang tepat walaupun secara teknis tidak ada error.

4. Drift pada perilaku pengguna

Pengguna bisa mulai memakai sistem dengan cara yang berbeda dari asumsi awal. Misalnya:

- mereka mengajukan pertanyaan yang lebih kompleks;
- menggunakan AI untuk konteks yang semula tidak dirancang;
- atau terlalu mengandalkan hasil AI untuk tugas yang lebih sensitif.

5. Drift pada integrasi dan workflow

Pada agentic AI atau sistem yang terhubung ke banyak aplikasi, perubahan pada salah satu tool, API, atau proses organisasi bisa membuat perilaku sistem ikut berubah.

Drift penting dipantau karena ia sering menjadi penyebab utama turunnya kualitas AI setelah implementasi awal. Sebuah sistem bisa lulus pengujian awal dengan baik, tetapi beberapa bulan kemudian:

- hasil klasifikasi memburuk;
- retrieval menjadi kurang relevan;
- ringkasan makin sering kehilangan unsur penting;
- atau agent makin sering berhenti di titik yang salah.

Tanpa monitoring drift, organisasi mungkin baru menyadari masalah setelah:

- pengguna mulai kehilangan kepercayaan;
- kualitas kerja menurun;
- atau insiden tertentu terjadi.

Karena itu, drift perlu diperlakukan sebagai tanda dini bahwa sistem perlu:

- ditinjau ulang;
- disesuaikan prompt-nya;
- diperbarui sumber dokumennya;
- dievaluasi workflow-nya;
- atau, pada beberapa konteks, dilakukan retraining atau penataan ulang model.

7.4.2 Monitoring latency

Latency adalah waktu yang dibutuhkan sistem untuk memberikan hasil setelah menerima input. Dalam AI, latency sangat penting karena AI mungkin secara kualitas cukup baik, tetapi bila hasilnya terlalu lambat, maka manfaat organisasionalnya dapat turun tajam.

Sebagai contoh:

- knowledge assistant yang menjawab terlalu lama akan diabaikan pengguna;
- sistem ringkasan yang membutuhkan waktu terlalu panjang akan kurang berguna untuk briefing cepat;
- klasifikasi surat atau aduan yang lambat akan menghambat triase;
- dan agentic workflow yang terlalu panjang dapat membuat pengguna kehilangan konteks atau kesabaran.

Karena itu, latency bukan hanya urusan performa teknis. Ia juga bagian dari kelayakan operasional.

Latency AI bisa berubah dari waktu ke waktu karena banyak sebab, misalnya:

- volume penggunaan meningkat;
- integrasi dengan layanan eksternal menjadi lebih lambat;
- model yang dipakai berubah;
- dokumen sumber bertambah besar;
- agent memakai lebih banyak langkah atau tool;
- atau infrastruktur pendukung mengalami beban yang lebih tinggi.

Dalam praktik, sistem yang semula terasa cepat saat uji coba bisa menjadi lambat ketika dipakai banyak pengguna atau ketika dokumen dan variasi kasus bertambah. Karena itu, organisasi perlu memantau latency bukan hanya sekali, tetapi secara berkala.

Tidak semua use case membutuhkan latency yang sama. Untuk tugas tertentu, keterlambatan beberapa detik mungkin masih dapat diterima. Namun, pada konteks lain, keterlambatan kecil saja bisa membuat AI terasa tidak berguna.

Misalnya:

- untuk orientasi awal terhadap dokumen yang panjang, latency sedikit lebih lama mungkin masih dapat diterima;
- untuk klasifikasi awal surat atau aduan dalam workflow operasional, latency yang terlalu tinggi akan terasa lebih mengganggu;
- untuk agentic AI, yang penting bukan hanya total latency, tetapi juga apakah pengguna tahu bahwa sistem masih bekerja, pada tahap mana ia berada, dan kapan hasil yang masuk akal dapat diharapkan.

Karena itu, monitoring latency sebaiknya dihubungkan dengan:

- jenis tugas;
- harapan pengguna;
- dan ritme proses kerja.

7.4.3 Monitoring misuse

Misuse adalah penggunaan AI dengan cara yang tidak sesuai dengan tujuan, batas, atau aturan yang telah ditetapkan. Misuse tidak selalu berarti niat buruk. Kadang misuse terjadi karena:

- pengguna tidak memahami batas sistem;
- kebijakan internal belum jelas;
- prompt atau output terlalu terbuka untuk ditafsirkan;
- atau AI terasa sangat membantu sehingga pengguna mulai memakainya untuk hal-hal di luar konteks.

Dalam organisasi pemerintah, misuse bisa muncul dalam berbagai bentuk, misalnya:

- memasukkan data sensitif ke layanan yang tidak seharusnya;
- memakai AI untuk menyusun keluaran yang seharusnya tidak boleh langsung diotomasi;
- memperlakukan draft AI sebagai dokumen final tanpa verifikasi;
- menggunakan retrieval assistant untuk pertanyaan di luar corpus resminya;
- atau memakai agentic AI untuk workflow yang melampaui batas yang dirancang.

Misuse penting dipantau karena banyak risiko AI justru tidak muncul karena sistem “jahat”, tetapi karena sistem dipakai di luar konteksnya. Sebuah solusi AI bisa cukup aman dan cukup berguna bila dipakai sesuai tujuan, tetapi menjadi bermasalah ketika:

- pengguna terlalu percaya pada hasilnya;
- konteks tugas berubah;
- atau ruang pakainya diperluas tanpa evaluasi ulang.

Dalam organisasi pemerintah, misuse sangat penting karena dapat menyentuh:

- perlindungan data;
- akuntabilitas;
- integritas informasi;
- dan kualitas layanan.

Karena itu, monitoring misuse membantu organisasi melihat:

- apakah pengguna mengikuti batas penggunaan;
- apakah pelatihan pengguna cukup;
- apakah kebijakan internal cukup jelas;
- dan apakah antarmuka atau workflow justru mendorong penggunaan yang salah.

Beberapa tanda misuse yang perlu diperhatikan antara lain:

- pengguna mulai memasukkan jenis data yang seharusnya tidak diproses;
- hasil AI terlalu sering dipakai langsung tanpa pemeriksaan;
- pertanyaan atau instruksi pengguna semakin menjauh dari ruang lingkup use case awal;
- agentic AI dipakai pada langkah yang tidak memiliki guardrail memadai;
- atau terjadi pola penggunaan yang menunjukkan bahwa AI dianggap sebagai pengganti pertimbangan manusia, bukan sebagai alat bantu.

Monitoring misuse seringkali membutuhkan kombinasi:

- logging penggunaan,
- observasi terhadap pola kerja,
- pelaporan pengguna,
- dan review berkala oleh pemilik proses atau pengelola sistem.

7.4.4 Monitoring incident

Incident dalam konteks AI adalah kejadian yang menunjukkan bahwa sistem:

- gagal bekerja sebagaimana mestinya;
- menimbulkan risiko yang nyata;
- atau menghasilkan dampak yang memerlukan penanganan, eskalasi, atau peninjauan lebih lanjut.

Incident dapat berbentuk teknis, tetapi juga bisa berbentuk operasional atau tata kelola. Ini penting karena dalam AI, insiden tidak selalu sama dengan “server down”.

Incident bisa berupa:

- keluaran yang sangat menyesatkan;
- kebocoran informasi melalui output;
- penggunaan di luar tujuan yang berdampak nyata;
- prompt injection yang berhasil mempengaruhi sistem;
- agent melakukan langkah yang tidak semestinya;
- atau perubahan perilaku model yang membuat hasil menjadi tidak dapat diterima.

Tidak semua error adalah incident, dan tidak semua incident muncul sebagai error teknis. Sebuah sistem bisa mengalami error kecil tanpa dampak besar, sementara di sisi lain sebuah jawaban AI yang terdengar meyakinkan tetapi salah pada konteks sensitif bisa menjadi incident yang serius.

Karena itu, organisasi perlu punya cara untuk membedakan:

- masalah teknis rutin;
- kegagalan kualitas yang perlu diperbaiki;
- dan incident yang memerlukan respons lebih formal.

Pembedaan ini penting agar organisasi tidak:

- terlalu reaktif terhadap semua hal kecil;
- tetapi juga tidak meremehkan kejadian yang sebenarnya punya implikasi besar.

Dalam konteks instansi pemerintah, incident AI dapat mencakup hal-hal seperti:

- sistem menampilkan informasi internal yang tidak boleh keluar;
- knowledge assistant memberi jawaban keliru yang kemudian dipakai sebagai dasar kerja resmi;
- agentic workflow mengambil tindakan atau langkah di luar batas yang ditetapkan;
- sistem retrieval terus-menerus menarik dokumen yang salah setelah pembaruan data tertentu;
- model vendor berubah dan menyebabkan kualitas hasil turun tajam tanpa pemberitahuan yang cukup;
- atau sistem dipakai oleh pengguna untuk tujuan yang bertentangan dengan kebijakan internal.

Incident seperti ini perlu dicatat, ditangani, dan dipakai sebagai bahan evaluasi ulang.

Pelaporan incident penting karena tanpa mekanisme pelaporan yang jelas, organisasi hanya akan mengetahui masalah-masalah AI secara kebetulan atau setelah dampaknya membesar. Organisasi perlu punya jalur yang cukup jelas untuk:

- menerima laporan dari pengguna;
- menilai tingkat keparahan insiden;
- menentukan siapa yang harus dilibatkan;
- mencatat bukti yang relevan;
- dan memutuskan apakah sistem perlu diperbaiki, dibatasi, atau dihentikan sementara.

Dalam konteks AI, incident juga penting sebagai bahan pembelajaran organisasi. seringkali justru dari incident nyata, organisasi mulai memahami bahwa:

- guardrail terlalu lemah;
- pelatihan pengguna kurang;
- sumber dokumen belum tertata;
- atau vendor perlu diawasi lebih ketat.

7.5 Evaluasi prompt dan kualitas keluaran generative AI

Setelah pada subbagian sebelumnya dibahas pentingnya monitoring drift, latency, misuse, dan incident, langkah berikutnya adalah membahas area yang semakin penting dalam praktik organisasi, yaitu **evaluasi prompt dan kualitas keluaran generative AI**. Bagian ini perlu dibahas secara khusus karena generative AI memiliki karakter yang berbeda dari banyak bentuk AI lainnya. Pada generative AI, hasil sistem tidak hanya dipengaruhi oleh model dan data, tetapi juga sangat dipengaruhi oleh **cara instruksi diberikan, konteks yang disediakan, batas yang ditetapkan, dan bagaimana keluaran itu dipakai**.

Dalam banyak organisasi, generative AI mulai digunakan untuk:

- ringkasan dokumen;
- drafting awal;
- penyusunan briefing;
- transformasi format;
- tanya jawab berbasis dokumen;
- atau bantuan penulisan dan analisis awal.

Manfaatnya bisa besar. Namun, justru karena generative AI terasa luwes dan mudah dipakai, ada kecenderungan untuk menilai hasilnya hanya dari kesan permukaan: apakah kalimatnya rapi, apakah jawaban terdengar masuk akal, atau apakah teks tampak profesional. Cara penilaian seperti ini tidak cukup. Organisasi perlu mengevaluasi dua hal sekaligus:

- **prompt**, sebagai instrumen yang mengarahkan perilaku sistem;
- dan **kualitas keluaran**, sebagai hasil nyata yang akan mempengaruhi kerja pengguna dan proses organisasi.

Dengan kata lain, untuk generative AI, pertanyaan evaluatifnya bukan hanya “apakah model ini bagus”, tetapi juga:

- apakah instruksi yang dipakai cukup tepat;
- apakah prompt ini masih sesuai untuk konteks kerja saat ini;
- apakah hasil yang keluar setia pada sumber dan tujuan;
- apakah kualitas keluaran cukup stabil;
- dan apakah keluaran itu aman, berguna, dan dapat dipertanggungjawabkan.

Pada AI generatif, prompt bukan sekadar cara pengguna “bertanya ke sistem”. Dalam organisasi, prompt adalah bagian dari desain sistem kerja. Prompt menentukan:

- fokus tugas;
- batas keluaran;
- format hasil;
- hubungan dengan sumber;
- dan ruang gerak model.

Karena itu, prompt perlu dievaluasi karena beberapa alasan.

Pertama, prompt yang tampak baik pada awalnya belum tentu tetap baik setelah konteks penggunaan berubah. Misalnya:

- struktur dokumen organisasi berubah;
- kebutuhan pimpinan berubah;
- sumber dokumen bertambah;
- atau pengguna mulai memakai prompt untuk jenis tugas yang lebih beragam.

Kedua, prompt yang terlalu umum sering menghasilkan keluaran yang:

- terlalu bebas;
- kurang terarah;
- menambahkan isi di luar kebutuhan;
- atau tidak cukup konsisten antar pengguna.

Ketiga, prompt yang tidak cukup jelas dapat membuat model:

- terlalu yakin ketika sumber sebenarnya lemah;
- gagal menampilkan batas ketidakpastian;
- atau menghasilkan format yang tidak sesuai dengan alur kerja organisasi.

Keempat, pada organisasi yang mulai mengintegrasikan generative AI ke workflow, prompt seringkali berubah menjadi semacam **aturan kerja**. Bila aturan ini tidak dievaluasi, kualitas sistem akan mudah bergeser tanpa disadari.

Karena itu, prompt harus diperlakukan sebagai objek evaluasi, bukan hanya alat yang diasumsikan selalu netral dan selalu memadai.

7.5.1 Prompt sebagai bagian dari sistem

Pada penggunaan kasual, prompt sering dipahami sebagai keterampilan individual. Namun, dalam organisasi—terutama bila generative AI dipakai berulang untuk tugas tertentu—prompt sebaiknya dilihat sebagai bagian dari **arsitektur kerja**.

Sebagai contoh:

- prompt untuk ringkasan briefing pimpinan;
- prompt untuk menyusun draft FAQ layanan;
- prompt untuk mengekstraksi isu utama dari dokumen;
- prompt untuk membandingkan dua naskah kebijakan;
- prompt untuk menyusun ringkasan aduan ke format tertentu.

Bila prompt-prompt seperti ini dipakai terus dalam organisasi, maka prompt bukan lagi sekadar improvisasi pengguna, tetapi menjadi bagian dari mekanisme operasional. Karena itu, prompt perlu dievaluasi sebagaimana komponen sistem lain dievaluasi:

- apakah masih sesuai tujuan;
- apakah masih menghasilkan keluaran yang cukup baik;
- apakah terlalu sempit atau terlalu luas;
- dan apakah perlu disesuaikan.

Evaluasi prompt dapat dilakukan melalui beberapa pertanyaan utama.

1. Kejelasan tujuan

Apakah prompt cukup jelas menjelaskan tugas yang diminta?

Prompt yang terlalu kabur akan memberi ruang terlalu besar bagi model untuk “menebak” tujuan, dan ini meningkatkan risiko hasil yang tidak tepat.

2. Kecukupan konteks

Apakah prompt memberi konteks yang cukup?

Prompt yang terlalu sedikit konteksnya sering menghasilkan jawaban umum, dangkal, atau salah sasaran.

3. Kejelasan batas

Apakah prompt membatasi apa yang boleh dan tidak boleh dilakukan model?

Ini penting terutama untuk mencegah keluaran yang terlalu spekulatif, terlalu luas, atau menambahkan informasi di luar sumber.

4. Kejelasan format keluaran

Apakah prompt menjelaskan bentuk hasil yang dibutuhkan?

Misalnya:

- poin-poin ringkas;
- tabel;
- format memo;
- ringkasan 1 paragraf;
- daftar isu dan tindak lanjut.

Prompt yang tidak mengarahkan format sering membuat hasil sulit dipakai di workflow nyata.

5. Hubungan dengan sumber

Apakah prompt meminta model tetap setia pada sumber?

Untuk tugas-tugas berbasis dokumen, ini sangat penting agar sistem tidak terlalu generatif dan tidak mudah ber confabulation.

6. Kesesuaian dengan pengguna dan workflow

Apakah prompt menghasilkan keluaran yang memang cocok untuk pengguna akhir?

Prompt yang baik secara linguistik belum tentu cocok untuk ritme kerja organisasi.

Dari sini terlihat bahwa evaluasi prompt tidak hanya bersifat bahasa, tetapi juga bersifat operasional.

Dalam organisasi, prompt tidak boleh dianggap statis. Ada beberapa alasan mengapa prompt yang semula cukup baik bisa menurun kualitasnya dari waktu ke waktu.

- **Pertama, dokumen sumber berubah.** Prompt yang dirancang untuk satu jenis struktur dokumen bisa menjadi kurang tepat ketika format sumber berubah.
- **Kedua, harapan pengguna berubah.** Misalnya, awalnya pengguna hanya membutuhkan ringkasan umum, tetapi kemudian mereka memerlukan format yang lebih tajam, lebih singkat, atau lebih berorientasi tindak lanjut.
- **Ketiga, use case meluas.** Prompt yang awalnya dibuat untuk satu tugas kadang mulai dipakai untuk tugas lain, padahal tidak dirancang untuk itu.
- **Keempat, vendor atau model berubah.** Model yang berbeda dapat bereaksi berbeda terhadap prompt yang sama. Karena itu, prompt yang semula efektif bisa menghasilkan keluaran yang tidak lagi stabil setelah pembaruan model.
- **Kelima, pengguna belajar “mengakali” prompt.** Kadang pengguna mulai menambah instruksi sendiri, menyalin prompt ke konteks lain, atau memperluas penggunaannya di luar tujuan awal.

Karena itu, prompt perlu diperlakukan sebagai sesuatu yang perlu direview dan disesuaikan, bukan diabaikan setelah sekali dibuat.

7.5.2 Kualitas keluaran generative AI

Salah satu jebakan terbesar dalam penggunaan generative AI adalah menilai hasil hanya dari **kefasihan bahasa**. Teks yang rapi, lancar, dan terdengar meyakinkan sering dianggap sebagai tanda kualitas yang baik. Padahal, dalam konteks organisasi pemerintah, kualitas keluaran perlu dinilai jauh lebih dalam.

Keluaran generative AI yang baik tidak cukup hanya:

- enak dibaca;
- terdengar profesional;
- atau terlihat pintar.

Yang jauh lebih penting adalah apakah hasil itu:

- sesuai tujuan;
- setia pada sumber;
- cukup lengkap;
- tidak menyesatkan;
- aman dipakai dalam workflow;
- dan sesuai dengan tingkat pengawasan manusia yang dirancang.

Karena itu, evaluasi kualitas keluaran generative AI harus melampaui kualitas bahasa permukaan.

Secara umum, kualitas keluaran generative AI dapat dinilai melalui beberapa dimensi berikut.

1. Relevansi

Apakah hasil benar-benar menjawab tugas yang diminta?

Sebuah keluaran bisa fasih, tetapi tidak benar-benar relevan dengan pertanyaan atau kebutuhan kerja.

2. Kesetiaan terhadap sumber

Untuk tugas berbasis dokumen, apakah keluaran tetap setia pada isi sumber?

Ini penting untuk mencegah tambahan isi yang tidak sah, penghilangan hal penting, atau penafsiran yang berlebihan.

3. Kelengkapan unsur penting

Apakah hasil mencakup unsur yang benar-benar diperlukan?

Misalnya, ringkasan briefing mungkin perlu mencakup isu utama, risiko, dan tindak lanjut, bukan hanya narasi umum.

4. Struktur dan keterpakaian

Apakah hasil mudah dipakai dalam workflow nyata?

Hasil yang benar secara isi tetapi formatnya tidak sesuai tetap bisa kurang berguna.

5. Keamanan dan kepatutan

Apakah hasil mengandung informasi yang seharusnya tidak keluar, terlalu spekulatif, atau melampaui ruang lingkup yang aman?

6. Konsistensi

Apakah keluaran cukup stabil untuk input yang sejenis?

Konsistensi penting terutama bila AI dipakai berulang dalam tugas yang serupa.

7. Tingkat koreksi yang dibutuhkan

Berapa banyak penyuntingan atau koreksi manusia yang biasanya dibutuhkan sebelum hasil bisa dipakai?

Ini penting untuk membaca manfaat nyata, bukan hanya kualitas semu.

Dari sini terlihat bahwa kualitas keluaran generative AI harus dibaca sebagai **kualitas operasional**, bukan hanya kualitas bahasa.

7.5.3 Evaluasi keluaran perlu sesuai dengan jenis tugas

Seperti pada evaluasi teknis lain, kualitas keluaran generative AI harus dibaca sesuai jenis tugas.

Pada **ringkasan**, fokus evaluasi mungkin pada:

- kesetiaan terhadap sumber;
- kelengkapan unsur penting;
- dan apakah hasil terlalu umum atau terlalu bebas.

Pada **drafting awal**, fokus evaluasi mungkin pada:

- kelengkapan struktur;
- kesesuaian gaya dan format;
- serta jumlah koreksi substantif yang dibutuhkan.

Pada **tanya jawab berbasis dokumen**, fokus evaluasi mungkin pada:

- relevansi jawaban;
- ketepatan sumber;
- dan sejauh mana jawaban melampaui apa yang didukung dokumen.

Pada **transformasi format**, fokus evaluasi mungkin pada:

- ketepatan pemetaan isi ke format baru;
- apakah ada bagian yang hilang;
- dan apakah struktur keluaran benar-benar lebih berguna bagi pengguna.

Artinya, organisasi tidak sebaiknya memakai satu standar kualitas yang sama untuk semua jenis keluaran.

Sama seperti fairness dan reliability, kualitas keluaran generative AI juga tidak boleh dinilai hanya dari beberapa contoh terbaik. Evaluasi perlu dilakukan pada **variasi kasus**, misalnya:

- dokumen yang pendek dan yang panjang;
- dokumen yang rapi dan yang berantakan;
- pertanyaan yang jelas dan yang ambigu;
- sumber yang lengkap dan yang terbatas;
- serta tugas yang sederhana dan yang sedikit lebih kompleks.

Ini penting karena generative AI sering tampak sangat baik pada contoh-contoh yang ideal, tetapi mulai melemah ketika berhadapan dengan variasi yang lebih nyata.

Bagi organisasi pemerintah, pendekatan ini sangat penting karena dokumen dan kebutuhan kerja di lapangan jarang seragam sempurna.

7.5.4 Umpan balik pengguna sebagai sumber utama evaluasi

Dalam banyak kasus, kualitas keluaran generative AI paling cepat terlihat dari **umpan balik pengguna**. Pengguna yang benar-benar memakai sistem akan tahu:

- apakah hasil membantu atau justru menambah kerja;
- apakah ringkasan sering kehilangan unsur penting;
- apakah draft terlalu sering harus ditulis ulang;
- apakah jawaban cukup bisa dipercaya;
- dan apakah format keluaran cocok dengan kebutuhan kerja.

Karena itu, evaluasi generative AI sebaiknya tidak hanya bertumpu pada penilaian tim teknis atau demonstrasi internal. Organisasi perlu menangkap:

- koreksi pengguna;
- pola penolakan hasil;
- keluhan berulang;
- serta bagian mana dari hasil yang sering disunting.

Umpan balik ini sangat penting untuk memperbaiki prompt, sumber, guardrail, maupun integrasi ke workflow.

Prompt dan kualitas keluaran tidak boleh dievaluasi secara terpisah. seringkali, masalah pada keluaran bukan berarti modelnya buruk, tetapi karena:

- prompt terlalu umum;
- tujuan tugas kurang jelas;
- format tidak diarahkan;
- atau konteks sumber tidak diberikan dengan cukup baik.

Sebaliknya, prompt yang tampak rapi juga belum tentu baik bila hasilnya:

- sering membingungkan pengguna;
- tidak sesuai workflow;
- atau menghasilkan keluaran yang terlalu banyak perlu dikoreksi.

Karena itu, organisasi perlu melihat hubungan sebab-akibat antara prompt dan keluaran:

- prompt seperti apa yang menghasilkan kualitas terbaik;
- prompt seperti apa yang justru terlalu bebas;
- dan prompt seperti apa yang paling stabil untuk use case tertentu.

Pada akhirnya, evaluasi prompt dan keluaran generative AI bukan hanya tugas teknis, tetapi juga bagian dari **governance**. Ini karena prompt bisa menjadi bagian dari aturan kerja organisasi, dan keluaran generative AI bisa mempengaruhi:

- kualitas dokumen;
- orientasi awal pimpinan;
- kecepatan pemrosesan informasi;
- dan kualitas komunikasi internal atau layanan.

Karena itu, organisasi perlu mendokumentasikan:

- prompt penting yang dipakai berulang;
- perubahan prompt yang signifikan;
- batas penggunaan keluaran;
- serta hasil evaluasi kualitas yang relevan.

Pendekatan ini membantu generative AI tidak tumbuh sebagai praktik improvisasi yang liar, tetapi sebagai komponen kerja yang dapat ditinjau dan diperbaiki secara sistematis.

Evaluasi prompt dan kualitas keluaran generative AI sangat penting karena pada sistem generatif, hasil tidak hanya dipengaruhi oleh model, tetapi juga sangat ditentukan oleh instruksi, konteks, batas, dan cara hasil dipakai dalam workflow. Prompt perlu dievaluasi dari sisi kejelasan tujuan, kecukupan konteks, kejelasan batas, kesesuaian format, hubungan dengan sumber, dan kecocokannya dengan kebutuhan pengguna. Kualitas keluaran perlu dinilai bukan hanya dari kefasihan bahasa, tetapi dari relevansi, kesetiaan terhadap sumber, kelengkapan unsur penting, keterpakaian, keamanan, konsistensi, dan tingkat koreksi yang dibutuhkan.

7.6 Evaluasi dan pengawasan agentic AI

Setelah pada subbagian sebelumnya dibahas evaluasi prompt dan kualitas keluaran generative AI, langkah berikutnya adalah membahas bentuk AI yang lebih kompleks, yaitu **agentic AI**. Bagian ini perlu dibahas secara khusus karena agentic AI berbeda dari sistem yang hanya menerima satu instruksi lalu menghasilkan satu keluaran. Agentic AI dapat membantu menjalankan **beberapa langkah kerja secara berurutan**, seperti:

- memahami tujuan;
- mencari sumber;
- memilih dokumen atau data;
- merangkum hasil;
- menyusun draft;
- menggunakan tool;
- dan dalam beberapa kasus meneruskan hasil ke langkah berikutnya dalam workflow.

Karena itu, evaluasi dan pengawasan agentic AI tidak cukup hanya melihat “apakah output akhirnya bagus”. Organisasi juga perlu menilai:

- bagaimana agent memahami tujuan;
- langkah apa saja yang dijalankan;
- sumber apa yang dipilih;
- apakah urutan langkah tetap sesuai batas yang dirancang;
- kapan agent berhenti atau menyerahkan ke manusia;
- dan apakah keseluruhan workflow tetap berada dalam kontrol organisasi.

Dengan kata lain, jika pada generative AI evaluasi banyak berfokus pada prompt dan kualitas keluaran, maka pada agentic AI evaluasi harus mencakup **perilaku sistem dalam menjalankan alur kerja**.

7.6.1 Agentic AI harus dievaluasi sebagai workflow

Kesalahan umum dalam menilai agentic AI adalah memperlakukannya seperti model generatif biasa. Organisasi hanya melihat apakah jawaban akhir agent terlihat baik, padahal masalah utama bisa terjadi pada langkah-langkah sebelumnya.

Sebagai contoh:

- agent bisa menemukan jawaban yang tampak baik, tetapi sebenarnya memilih sumber yang salah;
- agent bisa menghasilkan draft yang rapi, tetapi langkah pencariannya mengabaikan dokumen yang lebih resmi;
- agent bisa menyelesaikan tugas, tetapi melangkah terlalu jauh tanpa persetujuan manusia;
- atau agent bisa terlihat efisien, tetapi sebenarnya tidak reliabel karena urutan langkahnya terlalu mudah berubah.

Karena itu, agentic AI harus dievaluasi sebagai **workflow yang dibantu AI**, bukan sekadar sebagai model bahasa yang menghasilkan teks. Yang dinilai bukan hanya hasil akhir, tetapi juga:

- kualitas langkah-langkah antara;
- kestabilan alur;
- batas kewenangan;
- dan keterlacakan tindakan.

Secara umum, evaluasi agentic AI dapat dilakukan melalui beberapa dimensi utama berikut.

1. Kesesuaian tujuan dan interpretasi tugas

Organisasi perlu melihat apakah agent memahami tujuan yang diberikan dengan benar. Banyak masalah agentic AI justru dimulai dari salah tafsir terhadap tugas. Bila tujuan awal dibaca terlalu sempit, terlalu luas, atau melenceng, maka seluruh langkah berikutnya bisa ikut salah.

Pertanyaan evaluatifnya antara lain:

- apakah agent memahami ruang lingkup tugas dengan benar;
- apakah agent membedakan antara bantuan awal dan tindakan yang lebih sensitif;
- apakah agent tahu kapan tugas membutuhkan persetujuan manusia;
- dan apakah agent memahami konteks batas yang dirancang organisasi.

2. Kualitas pemilihan sumber atau tool

Agentic AI seringkali bergantung pada kemampuan memilih:

- dokumen;
- basis pengetahuan;
- aplikasi;
- atau tool tertentu.

Karena itu, kualitas pemilihan sumber harus dievaluasi secara khusus.

Organisasi perlu menilai:

- apakah agent cenderung memilih sumber yang resmi dan relevan;
- apakah ada pola mengabaikan sumber penting;
- apakah agent terlalu mudah memakai tool tertentu walaupun tidak perlu;
- dan apakah agent dapat membedakan sumber yang otoritatif dan yang hanya kebetulan mudah diakses.

3. Urutan langkah dan logika workflow

Agentic AI dinilai tidak hanya dari apa yang dilakukan, tetapi juga **bagaimana urutan langkah itu dibangun**. Organisasi perlu melihat:

- apakah langkah-langkah yang dipilih masuk akal;
- apakah ada langkah yang seharusnya dilakukan tetapi dilewati;
- apakah ada langkah yang seharusnya tidak dilakukan tetapi justru muncul;
- dan apakah alur kerja agent cukup konsisten untuk tugas yang sejenis.

4. Kualitas keluaran antara

Pada agentic AI, keluaran antara bisa sangat menentukan hasil akhir. Misalnya:

- hasil pencarian awal;
- ringkasan awal dari beberapa sumber;
- daftar prioritas;
- atau struktur draft.

Jika keluaran antara sudah lemah, maka hasil akhir bisa tampak rapi tetapi dibangun di atas dasar yang lemah.

Karena itu, evaluasi agentic AI perlu cukup sering memeriksa artefak antara, bukan hanya artefak final.

5. Kualitas keluaran akhir

Tentu saja keluaran akhir tetap perlu dinilai. Namun, penilaiannya perlu mempertimbangkan:

- kesetiaan terhadap sumber;
- kesesuaian dengan tujuan;
- kualitas struktur dan format;
- keamanan isi;
- serta apakah hasil akhir telah melampaui kewenangan yang seharusnya dimiliki agent.

6. Kepatuhan terhadap batas dan guardrail

Salah satu aspek paling penting pada agentic AI adalah apakah agent tetap berada dalam **batas yang telah dirancang**. Organisasi perlu menilai:

- apakah agent hanya memakai tool yang diperbolehkan;
- apakah agent berhenti ketika kondisi tidak jelas;
- apakah agent menyerahkan ke manusia pada titik yang benar;
- dan apakah agent tidak mengambil langkah di luar ruang tugasnya.

7. Konsistensi perilaku

Agentic AI perlu dievaluasi dari sisi konsistensi. Untuk tugas yang mirip, apakah agent cenderung:

- mengikuti alur yang cukup serupa;
 - menghasilkan kualitas yang stabil;
 - dan tetap menjaga pola kehati-hatian yang sama?
- Jika perilaku agent terlalu liar atau terlalu berubah-ubah untuk kasus yang mirip, maka reliability-nya perlu dipertanyakan.

7.6.2 Pengawasan agentic AI

Evaluasi awal penting, tetapi untuk agentic AI, **pengawasan saat operasi** bahkan sering lebih penting. Ini karena agentic AI bergerak dalam ruang yang lebih dinamis. Ia bisa bertemu:

- data baru;
- dokumen baru;
- variasi pertanyaan yang tidak diprediksi;
- tool yang diperbarui;
- atau perubahan proses organisasi.

Karena itu, pengawasan agentic AI harus melihat tidak hanya apakah sistem “berjalan”, tetapi apakah ia tetap:

- patuh pada batas;
- cukup aman;
- dan cukup dapat diprediksi dalam perilakunya.

7.6.3 Human oversight pada agentic AI

Pada agentic AI, *human oversight* harus lebih nyata daripada pada banyak use case AI lain. Ini karena agent tidak hanya memberi saran, tetapi berpotensi:

- memutuskan urutan langkah;
- memilih sumber;
- membentuk sintesis;
- atau bahkan memicu langkah berikutnya.

Karena itu, organisasi perlu menetapkan secara jelas:

- di titik mana manusia masuk;
- hasil antara mana yang harus diperiksa;
- langkah mana yang memerlukan persetujuan;
- kondisi apa yang memaksa agent berhenti;
- dan siapa yang bertanggung jawab atas keluaran akhirnya.

Dalam praktik, pengawasan manusia pada agentic AI bisa muncul dalam bentuk:

- verifikasi hasil pencarian sebelum sintesis;
- persetujuan sebelum agent melanjutkan ke drafting;
- review manusia pada ringkasan atau prioritas yang dihasilkan;
- atau pemeriksaan akhir sebelum keluaran dipakai dalam proses resmi.

Semakin tinggi risiko use case, semakin kuat pengawasan manusia yang diperlukan.

Pengawasan agentic AI sangat bergantung pada **logging dan keterlacakan**. Organisasi perlu dapat melihat:

- tujuan awal yang diberikan;
- sumber apa yang diakses agent;
- tool apa yang dipakai;
- langkah apa yang dijalankan;
- keluaran antara apa yang dihasilkan;
- dan di titik mana manusia masuk atau seharusnya masuk.

Tanpa logging yang cukup, organisasi akan kesulitan menjawab pertanyaan seperti:

- mengapa agent menghasilkan keluaran tertentu;
- langkah mana yang menyebabkan kesalahan;
- apakah masalah berasal dari sumber, dari prompt, dari workflow, atau dari penggunaan tool;
- dan bagaimana memperbaiki perilaku agent di masa depan.

Karena itu, agentic AI yang tidak cukup terlacak sebaiknya diperlakukan dengan sangat hati-hati, terutama di sektor publik.

Salah satu inti pengawasan agentic AI adalah memastikan bahwa **batas kewenangan agent tetap dijaga**. Organisasi perlu memantau:

- apakah agent mulai digunakan untuk konteks yang tidak dirancang;
- apakah pengguna mendorong agent menyelesaikan tugas yang terlalu luas;
- apakah tool atau akses agent bertambah tanpa evaluasi;
- dan apakah agent mulai diperlakukan sebagai pengganti pertimbangan manusia, bukan sebagai alat bantu.

Pengawasan ini penting karena agentic AI mudah mengalami pergeseran penggunaan. Awalnya hanya dipakai untuk orientasi internal, lalu perlahan mulai dipakai untuk drafting eksternal, lalu mungkin dipakai untuk menyiapkan bahan yang lebih sensitif. Bila pergeseran seperti ini tidak diawasi, maka risiko bisa meningkat jauh lebih cepat daripada kemampuan organisasi mengendalikannya.

Pertanyaan penting dalam evaluasi agentic AI bukan apakah agent selalu sempurna, melainkan apakah agent **cukup baik dan cukup terkendali untuk peran yang diberikan**. Ini berarti organisasi perlu menilai beberapa hal sekaligus:

- apakah kualitas hasil cukup membantu;
- apakah langkah-langkahnya cukup masuk akal;
- apakah perilakunya cukup konsisten;
- apakah kesalahan masih mudah dikenali dan diperbaiki manusia;
- dan apakah risiko yang tersisa masih dapat diterima.

Bila agent menghasilkan hasil yang tampak membantu tetapi:

- terlalu sulit diawasi,
- terlalu sering berubah perilaku,
- terlalu mudah melampaui batas,
- atau terlalu sulit ditelusuri,
maka sistem itu belum layak dianggap matang.

Dengan kata lain, kualitas agentic AI tidak hanya diukur dari kecerdasan hasil, tetapi juga dari **kedewasaan pengawasannya**.

1. Evaluasi agentic AI perlu berbasis skenario

Salah satu cara terbaik menilai agentic AI adalah menggunakan **skenario**. Karena agent bekerja melalui urutan langkah, evaluasi berbasis skenario membantu organisasi melihat bagaimana agent bertindak pada:

- kasus normal;
- kasus ambigu;
- kasus yang kekurangan sumber;
- kasus yang mengandung pengecualian;
- dan kasus yang seharusnya memicu eskalasi ke manusia.

Pendekatan ini penting karena agent sering terlihat sangat baik pada skenario ideal, tetapi mulai bermasalah ketika berhadapan dengan variasi kasus yang lebih realistis.

Bagi organisasi pemerintah, evaluasi berbasis skenario sangat berguna karena banyak proses birokrasi justru penuh dengan pengecualian, konteks khusus, dan kebutuhan kehati-hatian yang tidak selalu muncul dalam contoh-contoh ideal.

2. Evaluasi agentic AI sebagai proses berkelanjutan

Seperti pada aspek AI lainnya, evaluasi agentic AI tidak selesai pada satu putaran pengujian. Agent perlu terus dievaluasi karena:

- sumber dan dokumen berubah;
- tool atau API berubah;
- pengguna berubah cara memakai sistem;
- kebutuhan organisasi berubah;
- dan risiko baru bisa muncul setelah sistem dipakai lebih luas.

Karena itu, agentic AI perlu berada dalam siklus:

- evaluasi awal,
- penggunaan terbatas,
- monitoring,
- review berkala,
- dan re-evaluation sebelum ruang lingkupnya diperluas.

Pendekatan seperti ini jauh lebih sehat daripada langsung memperlakukan agent sebagai komponen yang boleh dipakai luas tanpa pembelajaran bertahap.

7.6.4 Implikasi praktis bagi instansi pemerintah

Bagi instansi pemerintah, evaluasi dan pengawasan agentic AI dapat diterjemahkan ke beberapa langkah praktis:

- mulai dari use case yang sempit dan relatif terkendali;
- nilai bukan hanya output akhir, tetapi juga langkah-langkah antara;
- tetapkan titik persetujuan manusia dengan jelas;
- batasi tool, sumber, dan tindakan yang boleh dipakai agent;
- pastikan logging cukup untuk audit dan perbaikan;
- gunakan evaluasi berbasis skenario;
- dan jangan memperluas ruang lingkup agent sebelum kualitas dan kontrolnya cukup terbukti.

Ini penting karena dalam sektor publik, agentic AI yang terlalu cepat diperluas justru berisiko menimbulkan masalah akuntabilitas, keamanan, dan kualitas proses.

Evaluasi dan pengawasan agentic AI harus lebih luas daripada evaluasi AI generatif biasa. Yang dinilai bukan hanya kualitas keluaran akhir, tetapi juga pemahaman tujuan, pemilihan sumber dan tool, urutan langkah, kualitas keluaran antara, kepatuhan terhadap guardrail, konsistensi perilaku, dan keterlacakan tindakan. Pengawasan agentic AI juga memerlukan *human oversight* yang nyata, logging yang cukup, pemantauan atas batas kewenangan, dan evaluasi berbasis skenario agar organisasi dapat memastikan bahwa agent tetap membantu tanpa melampaui ruang kendali yang dirancang.

7.7 Continuous validation dan re-evaluation

Setelah membahas evaluasi teknis, manfaat organisasi, fairness, explainability, reliability, monitoring drift, latency, misuse, incident, serta evaluasi dan pengawasan agentic AI, langkah berikutnya adalah memahami satu prinsip penting dalam pengelolaan solusi AI, yaitu bahwa **penilaian terhadap AI tidak pernah benar-benar selesai**. Di sinilah konsep **continuous validation** dan **re-evaluation** menjadi sangat penting.

Pada sistem biasa, organisasi sering merasa cukup aman bila sistem telah diuji sebelum diluncurkan dan sesekali diperbaiki bila ada masalah. Pada AI, pendekatan seperti itu tidak memadai. AI bekerja dalam lingkungan yang berubah. Data dapat berubah. Dokumen sumber bertambah atau diperbarui. Prompt dan workflow disesuaikan. Vendor memperbarui model. Pengguna mengubah cara memakai sistem. Semua perubahan ini bisa mempengaruhi kualitas, fairness, keamanan, dan manfaat AI. Karena itu, organisasi perlu membangun kebiasaan untuk **terus memvalidasi** apakah sistem masih bekerja sesuai yang diharapkan, dan **mengevaluasi ulang** apakah sistem masih layak dipakai dalam bentuk yang sama.

7.7.1 Continuous validation sebagai bagian dari siklus hidup AI

Dalam standar internasional, prinsip ini ditegaskan dengan cukup jelas. **ISO/IEC 22989** menjelaskan bahwa setelah deployment dan operation, AI system dapat memerlukan **continuous validation** untuk memastikan bahwa sistem tetap memenuhi tujuan dan persyaratan yang ditetapkan. Standar yang sama juga menyebut bahwa bila sistem menggunakan *continuous learning* atau bila data produksi berubah, data uji yang dipakai untuk validasi juga mungkin perlu diperbarui agar tetap representatif terhadap kondisi nyata. Selain itu, standar tersebut menekankan perlunya **re-evaluation** terhadap hasil operasi dibandingkan dengan tujuan dan risiko yang telah diidentifikasi sebelumnya.

Ini menunjukkan bahwa continuous validation bukan tambahan opsional. Ia adalah bagian dari cara AI seharusnya dikelola. Organisasi tidak cukup hanya berkata, “sistem ini sudah pernah diuji.” Pertanyaan yang lebih relevan adalah, “apakah sistem ini **masih** memenuhi tujuan, kualitas, dan batas risiko yang kita tetapkan?”

Secara sederhana, **continuous validation** berarti proses memeriksa secara berkelanjutan apakah sistem AI:

- masih menghasilkan keluaran yang cukup baik;
- masih bekerja pada jenis input yang relevan;
- masih cukup sesuai dengan kebutuhan pengguna;
- dan masih berada dalam batas risiko yang dapat diterima.

Continuous validation bukan berarti setiap hari organisasi harus melakukan pengujian besar-besaran. Yang lebih penting adalah bahwa organisasi memiliki mekanisme yang teratur untuk memeriksa:

- apakah performa sistem berubah;
- apakah kualitas keluaran tetap layak;
- apakah fairness tetap terjaga;
- apakah sumber yang dipakai tetap relevan;
- dan apakah perilaku sistem tetap sesuai dengan desain awalnya.

Pada use case yang berbeda, bentuk continuous validation juga bisa berbeda. Untuk sistem klasifikasi, continuous validation mungkin berarti memeriksa sampel hasil terbaru dan membandingkannya dengan penilaian manusia. Untuk knowledge assistant, continuous validation bisa berarti memeriksa relevansi jawaban, sumber yang dipakai, dan pertanyaan yang sering gagal dijawab dengan baik. Untuk generative AI, continuous validation bisa berarti meninjau kualitas ringkasan, draft, atau jawaban terhadap kumpulan kasus uji yang diperbarui. Untuk agentic AI, continuous validation juga mencakup evaluasi langkah-langkah workflow, bukan hanya keluaran akhirnya.

Ada beberapa alasan utama mengapa continuous validation diperlukan.

- Pertama, **lingkungan penggunaan AI berubah**. Data produksi jarang tetap sama selamanya. Dokumen baru muncul, kebijakan berubah, variasi pertanyaan bertambah, dan konteks penggunaan menjadi lebih luas.
- Kedua, **perilaku sistem dapat bergeser tanpa disadari**. Prompt yang semula efektif bisa menjadi kurang tepat. Model vendor yang diperbarui bisa bereaksi berbeda. Retrieval corpus yang bertambah bisa mengubah hasil pencarian. Agent workflow yang diperluas bisa menimbulkan pola baru.
- Ketiga, **pengguna belajar memakai sistem dengan cara baru**. Kadang pengguna menjadi lebih terampil, tetapi kadang mereka juga mulai mendorong sistem melampaui konteks awalnya. Ini mempengaruhi bagaimana AI harus divalidasi.
- Keempat, **risiko tidak selalu muncul di awal**. Banyak masalah fairness, misuse, atau reliability justru baru terlihat setelah sistem digunakan dalam jangka waktu tertentu.

Karena itu, continuous validation membantu organisasi menangkap perubahan lebih dini, sebelum kualitas turun terlalu jauh atau risiko menjadi lebih besar.

7.7.2 Continuous validation berbeda dari monitoring biasa

Continuous validation memang terkait erat dengan monitoring, tetapi keduanya tidak sama. **Monitoring** lebih berfokus pada pengamatan berkelanjutan terhadap tanda-tanda operasional, seperti drift, latency, misuse, atau incident. Sementara itu, **continuous validation** lebih berfokus pada pertanyaan evaluatif: apakah sistem **masih layak** menurut kriteria yang telah ditetapkan.

Dengan kata lain:

- monitoring membantu organisasi melihat apa yang sedang terjadi;
- continuous validation membantu organisasi menilai apa arti dari perubahan itu terhadap kelayakan sistem.

Contohnya, monitoring dapat menunjukkan bahwa latency meningkat atau bahwa pola pertanyaan pengguna berubah. Continuous validation kemudian menilai apakah perubahan itu cukup signifikan untuk menurunkan kualitas sistem atau memerlukan penyesuaian lebih lanjut.

Dalam praktik organisasi, continuous validation dapat mencakup beberapa hal berikut.

1. Kualitas keluaran

Apakah hasil AI masih cukup akurat, cukup relevan, cukup setia pada sumber, dan cukup berguna untuk konteks kerjanya?

2. Kesesuaian dengan tujuan awal

Apakah sistem masih membantu tujuan organisasi yang semula menjadi dasar implementasi? Atau manfaatnya justru menurun?

3. Fairness

Apakah ada variasi kasus tertentu yang mulai lebih sering dirugikan? Apakah distribusi kesalahan berubah?

4. explainability

Apakah hasil AI masih cukup dapat dijelaskan kepada pengguna dan pengelola? Apakah perubahan model atau prompt membuat hasil menjadi lebih sulit dipahami?

5. Reliability

Apakah sistem masih cukup stabil dan konsisten pada kondisi penggunaan yang nyata?

6. Kepatuhan terhadap guardrail

Apakah sistem tetap berada dalam batas penggunaan yang dirancang? Apakah agent masih patuh pada ruang kewenangannya? Apakah keluaran generatif tetap sesuai pembatasan?

7. Kesesuaian sumber dan data

Apakah data, dokumen, metadata, dan sumber yang dipakai sistem masih mutakhir dan benar-benar layak menjadi dasar keluaran AI?

Dengan melihat hal-hal ini, organisasi dapat menjaga agar validasi tidak terlalu sempit.

7.7.3 Re-evaluation: kapan evaluasi ulang diperlukan

Jika continuous validation adalah pemeriksaan berkelanjutan, maka **re-evaluation** adalah penilaian ulang yang lebih menyeluruh ketika ada alasan yang cukup kuat. Re-evaluation diperlukan ketika organisasi perlu kembali ke pertanyaan dasar: **apakah sistem ini masih pantas digunakan seperti sekarang?**

Beberapa kondisi yang biasanya memicu re-evaluation antara lain:

- perubahan besar pada data atau sumber;
- perubahan model atau vendor;
- perubahan workflow organisasi;
- perluasan ruang lingkup use case;
- munculnya incident yang penting;
- penurunan kualitas yang konsisten;
- perubahan kebijakan atau regulasi;
- atau umpan balik pengguna yang menunjukkan bahwa manfaat sistem tidak lagi jelas.

Re-evaluation berarti organisasi tidak hanya memperbaiki detail kecil, tetapi meninjau ulang:

- tujuan penggunaan;
- manfaat aktual;
- risiko yang kini muncul;
- kesesuaian kontrol;
- dan apakah sistem perlu dipertahankan, diubah, dipersempit, atau diganti.

Penting untuk dipahami bahwa re-evaluation bukan tanda bahwa organisasi gagal. Justru, re-evaluation adalah tanda bahwa organisasi mengelola AI secara matang. Sistem yang terus dipakai tanpa pernah dievaluasi ulang biasanya justru lebih berisiko, karena organisasi tidak pernah bertanya apakah asumsi awalnya masih berlaku.

Dalam konteks sektor publik, re-evaluation sangat penting karena perubahan eksternal dan internal cukup sering terjadi. Kebijakan dapat diperbarui. Kebutuhan pimpinan dapat berubah. Dokumen resmi bertambah. Layanan berkembang. Sistem yang beberapa bulan lalu berguna bisa jadi hari ini perlu penyesuaian agar tetap relevan dan patuh.

7.7.4 Continuous validation untuk generative AI

Pada generative AI, continuous validation sangat penting karena kualitas sistem tidak hanya bergantung pada model, tetapi juga pada:

- prompt;
- jenis tugas;
- sumber yang dipakai;
- dan ekspektasi pengguna.

Continuous validation pada generative AI dapat meliputi:

- pemeriksaan berkala terhadap ringkasan, draft, atau jawaban;
- review terhadap tingkat kesetiaan pada sumber;
- penilaian terhadap pola confabulation;
- evaluasi apakah format keluaran masih sesuai kebutuhan kerja;
- dan pemantauan apakah pengguna mulai memakai sistem di luar konteks yang semula dirancang.

Pada sistem generatif, hasil yang tampak lancar belum tentu valid. Karena itu, validasi berkelanjutan sangat penting agar kualitas tidak dinilai hanya dari kelancaran bahasa.

7.7.5 Continuous validation untuk agentic AI

Pada agentic AI, continuous validation perlu lebih luas lagi. Yang perlu divalidasi bukan hanya hasil akhir, tetapi juga:

- urutan langkah yang dijalankan;
- pemilihan sumber dan tool;
- kepatuhan terhadap batas kewenangan;
- titik eskalasi ke manusia;
- dan konsistensi perilaku workflow.

Jika agentic AI mulai dipakai lebih luas, maka re-evaluation juga perlu dilakukan lebih hati-hati. Agent yang awalnya cukup aman pada ruang lingkup sempit belum tentu tetap aman bila:

- tugasnya diperluas;
- tool yang diakses bertambah;
- atau ia dihubungkan ke proses yang lebih sensitif.

Karena itu, untuk agentic AI, continuous validation dan re-evaluation sebaiknya dianggap sebagai syarat dasar, bukan tambahan.

Agar prinsip ini tidak berhenti sebagai gagasan, organisasi perlu membangun mekanisme yang nyata. Bentuknya tidak harus rumit, tetapi perlu cukup jelas. Beberapa langkah yang dapat dilakukan antara lain:

- menetapkan sampel kasus yang ditinjau berkala;
- memperbarui data kasus uji agar tetap representatif;
- melibatkan pengguna dalam review kualitas hasil;
- menangkap koreksi manusia sebagai sinyal kualitas;
- menjadwalkan review berkala terhadap prompt, sumber, dan workflow;
- serta menentukan kondisi apa yang otomatis memicu re-evaluation.

Pendekatan ini membantu organisasi berpindah dari pola reaktif menuju pola pengelolaan yang lebih sistematis.

Continuous validation dan re-evaluation juga sangat terkait dengan **dokumentasi perubahan**. Bila prompt diubah, model diperbarui, retrieval corpus dibersihkan, atau workflow agent diperluas, maka organisasi perlu tahu:

- apa yang berubah;
- kapan berubah;
- mengapa berubah;
- dan apakah perubahan itu memperbaiki atau justru menurunkan kualitas.

Tanpa dokumentasi seperti ini, organisasi akan sulit menilai apakah hasil validasi berkelanjutan mencerminkan kemajuan, penurunan, atau hanya perubahan kondisi yang tidak tercatat.

Pada akhirnya, continuous validation dan re-evaluation penting karena menjadi dasar bagi keputusan organisasi. Dari hasil proses ini, organisasi dapat menentukan:

- apakah sistem masih layak dipertahankan;
- apakah perlu penyesuaian kecil;
- apakah perlu pembaruan besar;
- apakah ruang lingkupnya harus dipersempit;
- atau apakah sistem perlu dihentikan dan diganti dengan pendekatan lain.

Dengan kata lain, validasi berkelanjutan bukan hanya kegiatan evaluasi, tetapi bagian dari keputusan strategis tentang masa depan sistem AI.

Continuous validation dan re-evaluation adalah bagian penting dari pengelolaan AI yang matang. AI tidak dapat dianggap cukup divalidasi sekali di awal, karena data, dokumen, konteks, model, prompt, vendor, dan perilaku pengguna dapat berubah dari waktu ke waktu. Standar AI menegaskan perlunya operation and monitoring, continuous validation, dan re-evaluation agar sistem tetap sesuai dengan tujuan, kualitas, dan batas risiko yang telah ditetapkan.

7.8 Pemeliharaan data, model, pipeline, aplikasi, prompt, dan agent workflow

Setelah membahas evaluasi, monitoring, *continuous validation*, dan *re-evaluation*, langkah berikutnya adalah melihat bagaimana semua temuan itu diterjemahkan ke dalam pekerjaan yang sangat konkret, yaitu **pemeliharaan solusi AI**. Bagian ini penting karena pada akhirnya AI tidak cukup hanya dinilai. AI juga harus **dirawat**.

Dalam organisasi, istilah pemeliharaan kadang dipahami sempit sebagai memperbaiki bug aplikasi atau memperbarui versi perangkat lunak. Pada solusi AI, pemeliharaan jauh lebih luas. Yang perlu dirawat bukan hanya aplikasi, tetapi juga:

- **data** yang menjadi pondasi sistem;
- **model** atau engine inferensi;
- **pipeline** yang memproses aliran data dan logika kerja;
- **aplikasi** yang menjadi titik interaksi pengguna;
- **prompt** yang mengarahkan perilaku generative AI;
- serta **agent workflow** yang mengatur langkah-langkah kerja agentic AI.

Ini sejalan dengan SKKNI unit **Merawat Solusi Artificial Intelligence (AI)**, yang menegaskan bahwa hasil monitoring harus dipastikan kesesuaiannya dengan kebutuhan perawatan komponen arsitektur lengkap solusi AI, bahwa kebutuhan perawatan ditentukan sesuai prosedur perawatan, bahwa komponen arsitektur lengkap solusi AI dirawat sesuai rencana, dan bahwa seluruh aktivitas perawatan didokumentasikan sesuai standar dokumentasi yang berlaku. Dokumen yang sama juga menjelaskan bahwa komponen yang perlu dirawat mencakup hal-hal seperti strategi AI, tata kelola, faktor manusia, ketahanan siber, kompetensi AI, kualitas data, arsitektur dan infrastruktur data, pengukuran kinerja, etika, serta komponen teknis seperti perangkat keras, perangkat lunak, data, model atau knowledge, privacy compliance, dan security architecture.

Pada tingkat standar internasional, **ISO/IEC 22989** menegaskan bahwa pada tahap operation and monitoring, AI system perlu dipantau, diperbaiki bila mengalami error, diperbarui pada software, model, dan hardware untuk memenuhi kebutuhan baru serta meningkatkan performance and reliability, dan diberikan dukungan pengguna yang memadai. Standar ini juga menempatkan *continuous validation* dan *re-evaluation* sebagai kelanjutan alami dari operasi sistem. Sementara itu, **ISO/IEC 23053** menjelaskan bahwa retraining dapat diperlukan karena data drift, concept drift, perubahan kebutuhan, atau optimasi model, dan bahwa continuous learning menuntut pemantauan performa model serta guard-rails terhadap perilaku yang dapat diterima.

Dari sini terlihat bahwa pemeliharaan AI bukan pekerjaan sampingan setelah implementasi, melainkan bagian inti dari menjaga agar AI tetap:

- relevan,
- aman,
- andal,
- dan bernilai bagi organisasi.

Mengapa pemeliharaan AI lebih luas daripada pemeliharaan sistem biasa

Pada software biasa, perubahan perilaku sistem sering dikaitkan langsung dengan perubahan kode. Pada AI, perilaku sistem dapat berubah walaupun kode aplikasi utamanya hampir tidak berubah sama sekali. Hal ini bisa terjadi karena:

- data baru masuk dengan karakteristik yang berbeda;
- dokumen sumber berubah;
- metadata diperbarui;
- model diganti atau disetel ulang;
- prompt diubah;
- retrieval corpus bertambah;
- vendor memperbarui layanan;
- atau agent diberi tool baru.

Akibatnya, pemeliharaan AI harus melihat sistem sebagai **gabungan komponen** yang saling mempengaruhi. Sebuah knowledge assistant bisa menurun kualitasnya bukan karena aplikasinya rusak, tetapi karena dokumen sumber tidak diperbarui. Sebuah generative AI bisa memberi hasil yang lebih lemah karena prompt tidak lagi sesuai. Sebuah agentic AI bisa menjadi lebih berisiko karena workflow-nya berkembang tanpa evaluasi ulang. Karena itu, pemeliharaan AI perlu dibaca secara sistemik, bukan hanya teknis sempit.

Pemeliharaan data

Data sebagai pondasi yang terus berubah

Data adalah salah satu komponen yang paling penting sekaligus paling mudah berubah dalam solusi AI. Dalam konteks pemerintah, data yang dipakai AI bisa berupa:

- dataset terstruktur;
- metadata;
- dokumen kebijakan;
- SOP;
- surat;
- aduan;
- arsip pengetahuan internal;
- atau log operasional.

Semua sumber ini tidak bersifat tetap. Dokumen dapat bertambah, diperbarui, dicabut, atau diganti versinya. Metadata dapat berubah. Kategori dapat disesuaikan. Oleh karena itu, data tidak cukup “disiapkan” sekali di awal. Data harus **dirawat**.

Apa yang perlu dirawat dari data

Pemeliharaan data dapat mencakup:

- pembaruan sumber resmi;
- pembersihan data yang usang atau tidak lagi relevan;
- penataan versi dokumen;
- penyelarasan metadata;
- pemeriksaan kualitas label atau kategori;
- pembatasan akses sesuai kebijakan;
- dan peninjauan apakah data yang dipakai masih benar-benar diperlukan untuk use case yang ada.

Untuk AI berbasis dokumen, pemeliharaan data juga berarti menjaga agar retrieval corpus tetap:

- resmi,
- mutakhir,
- tertata,
- dan cukup dapat ditelusuri.

Bila ini tidak dilakukan, sistem bisa mulai menghasilkan jawaban yang tampak baik tetapi sebenarnya berbasis dokumen yang sudah tidak berlaku. Karena itu, pada banyak use case pemerintah, merawat data sama pentingnya dengan merawat model.

Pemeliharaan model

Model perlu dijaga kesesuaiannya dengan konteks

Pada solusi AI berbasis machine learning atau layanan model tertentu, model perlu dirawat agar tetap sesuai dengan konteks operasional. **ISO/IEC 23053** menjelaskan bahwa retraining dapat diperlukan karena data drift, concept drift, perubahan target, atau optimasi model, dan bahwa model yang tidak lagi mewakili data produksi akan kehilangan akurasi prediksinya.

Dalam praktik organisasi, pemeliharaan model dapat mencakup:

- pemantauan apakah performa model menurun;
- penyesuaian parameter;
- retraining dengan data yang lebih representatif;
- fine-tuning untuk konteks baru;
- atau bahkan penggantian model bila pendekatan yang lebih baik sudah tersedia.

Model bukan hanya diganti ketika rusak

Penting dipahami bahwa model tidak hanya perlu disentuh ketika “gagal total”. Kadang model masih berjalan, tetapi:

- performanya menurun perlahan;
- kualitasnya tidak lagi sebanding dengan risiko;
- atau ada model lain yang lebih cocok, lebih transparan, atau lebih mudah dikendalikan.

Karena itu, pemeliharaan model juga berarti mempertanyakan secara berkala:

- apakah model ini masih pilihan yang tepat;
- apakah pembaruan vendor perlu diterima atau ditahan dulu;
- dan apakah modifikasi terhadap model sudah terdokumentasi dengan baik. NIST AI 600-1 juga menekankan pentingnya mendokumentasikan bagaimana model pra-latih telah diadaptasi, termasuk fine-tuning, retrieval-augmented generation, parameter adjustment, dan perubahan lain yang mempengaruhi integritas serta privasi.

Pemeliharaan pipeline

Pipeline sebagai tulang punggung aliran kerja AI

Pipeline adalah rangkaian aliran data dan proses yang menghubungkan input, pemrosesan, model, validasi, dan keluaran. Dalam banyak kasus, pipeline justru menentukan apakah solusi AI dapat bekerja stabil dalam organisasi. Pipeline mencakup:

- pengambilan data atau dokumen;
- pembersihan dan transformasi;
- indexing atau retrieval;
- inferensi model;
- penyusunan output;
- logging;
- dan terkadang integrasi ke aplikasi lain.

ISO/IEC 23053 memandang AI berbasis ML sebagai sistem yang dibangun melalui ML pipeline, dan menunjukkan bahwa berbagai komponen data, tool, dan proses bekerja saling terhubung dalam seluruh alur pengembangan sampai operasi.

Apa yang perlu dirawat dari pipeline

Pemeliharaan pipeline dapat mencakup:

- memastikan aliran data tetap benar;
- memperbaiki titik transformasi yang menimbulkan error;
- menyesuaikan integrasi ketika sumber berubah;
- mengurangi keterlambatan proses;
- memperbaiki indexing;
- menata ulang alur bila retrieval atau inferensi tidak lagi efisien;
- dan menjaga logging agar cukup kaya untuk audit dan perbaikan.

Pada organisasi pemerintah, pipeline sering bersinggungan dengan banyak sumber dan aplikasi. Karena itu, pipeline yang tidak dirawat dapat membuat sistem tampak baik di permukaan, padahal:

- datanya tidak lagi sinkron;
- dokumen yang masuk tidak utuh;
- atau hasil yang keluar tidak lagi sesuai dengan kebutuhan proses.

Pemeliharaan aplikasi

Aplikasi tetap menjadi ruang kerja pengguna

Walaupun AI sering dilihat dari sudut model atau data, pengguna berinteraksi melalui **aplikasi**. Karena itu, aplikasi tetap merupakan komponen penting yang harus dirawat. Pemeliharaan aplikasi mencakup:

- stabilitas antarmuka;
- kemudahan koreksi hasil AI;
- kejelasan penandaan bahwa hasil adalah bantuan awal;
- keterhubungan ke sumber;
- pengelolaan hak akses;
- dan kenyamanan penggunaan dalam ritme kerja nyata.

Bila aplikasi tidak dirawat dengan baik, kualitas AI yang bagus sekalipun bisa sulit dirasakan. Misalnya:

- hasil AI tidak mudah diperiksa;
- sumber tidak tampil jelas;
- koreksi pengguna tidak bisa ditangkap;
- atau antarmuka terlalu lambat sehingga AI terasa mengganggu alur kerja.

Karena itu, pemeliharaan aplikasi perlu dibaca sebagai bagian dari menjaga manfaat organisasi, bukan sekadar menjaga tampilan.

Pemeliharaan prompt

Prompt sebagai komponen kerja yang hidup

Pada generative AI, prompt adalah bagian dari sistem. Ia mengarahkan tujuan, batas, format, dan perilaku keluaran. Karena itu, prompt juga harus dirawat. Prompt yang dulu efektif belum tentu tetap efektif ketika:

- jenis dokumen berubah;
- pengguna membutuhkan format yang berbeda;
- sumber bertambah;
- model berubah;
- atau organisasi memperluas ruang lingkup tugas.

Pedoman keamanan AI juga menekankan dokumentasi terhadap system prompts, limitations, guardrails, dan failure modes sebagai bagian dari pengelolaan aset AI.

Apa yang perlu dirawat dari prompt

Pemeliharaan prompt dapat meliputi:

- memperjelas instruksi yang terlalu umum;
- memperbarui batas penggunaan;
- menyesuaikan format keluaran dengan kebutuhan terbaru;
- memperkuat hubungan prompt dengan sumber resmi;
- mengurangi ruang bagi keluaran yang terlalu bebas;
- dan mendokumentasikan perubahan penting agar dampaknya dapat dievaluasi.

Dalam konteks organisasi, prompt yang dirawat dengan baik membantu menjaga kualitas keluaran lebih konsisten dan mengurangi variasi yang tidak perlu antar pengguna.

Pemeliharaan agent workflow

Agent workflow perlu dipandang sebagai objek pemeliharaan tersendiri

Pada agentic AI, yang perlu dirawat bukan hanya model dan prompt, tetapi juga **agent workflow**. Ini mencakup:

- urutan langkah;
- pemilihan sumber atau tool;
- aturan eskalasi ke manusia;
- guardrails;
- batas tindakan;
- dan kondisi fallback.

Agent workflow yang awalnya cukup baik dapat menjadi kurang layak bila:

- organisasi mengubah proses bisnis;
- sumber data bertambah;
- tool yang dipakai agent berubah;
- atau pengguna mulai mendorong agent menangani tugas yang lebih luas.

Apa yang perlu dirawat dari agent workflow

Pemeliharaan agent workflow dapat mencakup:

- menyederhanakan langkah yang terlalu kompleks;
- memperjelas kapan agent harus berhenti;
- menghapus tool yang tidak lagi perlu atau terlalu berisiko;
- memperbarui aturan sumber resmi;
- menambah titik persetujuan manusia;
- dan meninjau ulang workflow bila hasil antara sering bermasalah.

Pada sektor publik, ini sangat penting karena agentic AI mudah mengalami **creep**, yaitu meluas perlahan dari bantuan awal ke ruang tindakan yang lebih sensitif tanpa evaluasi yang cukup. Karena itu, workflow agent harus dirawat dengan disiplin, bukan dibiarkan berkembang secara improvisational.

Pemeliharaan membutuhkan rencana, bukan reaksi sesaat

Salah satu pelajaran utama dari sub bagian ini adalah bahwa pemeliharaan AI sebaiknya tidak dilakukan hanya secara reaktif ketika masalah sudah membesar. SKKNI menekankan pentingnya **rencana perawatan**, yang disusun berdasarkan hasil evaluasi monitoring dan metrik kesuksesan solusi AI yang terpasang. Dokumen tersebut juga menegaskan bahwa rencana perawatan meliputi komponen yang perlu diperbaiki karena performansinya berada di bawah kinerja saat solusi AI mulai dipasang.

Ini berarti organisasi sebaiknya memiliki:

- indikator apa yang menunjukkan komponen tertentu perlu dirawat;
- siapa yang bertanggung jawab untuk tiap komponen;
- bagaimana perubahan dilakukan;
- dan bagaimana dokumentasi pemeliharaan dijaga.

Pendekatan seperti ini membuat pemeliharaan menjadi bagian dari operasi rutin, bukan sekadar respons darurat.

Pemeliharaan sebagai pekerjaan lintas fungsi

Pemeliharaan AI juga tidak bisa dibebankan pada satu tim saja. Karena komponen yang dirawat beragam, maka pihak yang terlibat juga beragam:

- pengelola data merawat kualitas dan sumber data;
- tim teknis merawat model, pipeline, aplikasi, dan integrasi;
- pemilik proses menilai apakah perubahan masih sesuai kebutuhan kerja;
- pengelola tata kelola dan keamanan menilai risiko perubahan;
- pengguna memberi umpan balik terhadap kualitas hasil;
- dan pimpinan memberi arah jika perubahan menyangkut ruang lingkup atau prioritas organisasi.

Dengan kata lain, pemeliharaan AI adalah pekerjaan lintas fungsi yang memadukan perspektif teknis, operasional, dan tata kelola.

Dokumentasi pemeliharaan sangat penting

Karena pemeliharaan AI menyentuh banyak komponen yang saling mempengaruhi, dokumentasi menjadi sangat penting. Tanpa dokumentasi, organisasi akan sulit menilai:

- apa yang berubah;
- mengapa berubah;
- apakah perubahan memperbaiki kualitas;
- dan apakah ada risiko baru yang muncul.

SKKNI secara jelas menegaskan bahwa seluruh aktivitas kegiatan perawatan solusi AI perlu didokumentasikan sesuai standar dokumentasi yang berlaku. Ini penting agar pemeliharaan AI tidak bergantung pada ingatan informal tim tertentu.

Pemeliharaan solusi AI mencakup jauh lebih banyak daripada memperbaiki aplikasi. Dalam organisasi, yang perlu dirawat adalah data, model, pipeline, aplikasi, prompt, dan agent workflow, karena seluruh komponen itu bersama-sama membentuk kualitas, keamanan, dan manfaat AI. SKKNI menegaskan bahwa perawatan harus ditentukan dari hasil monitoring, dilakukan sesuai rencana, dan didokumentasikan dengan baik, sementara standar AI menempatkan repair, update, support, continuous validation, dan re-evaluation sebagai bagian dari operasi dan pemeliharaan sistem.

7.9 Update, patch, retraining, dan dokumentasi perubahan

Setelah pada subbagian sebelumnya dibahas bahwa solusi AI perlu dipelihara pada berbagai komponennya—mulai dari data, model, pipeline, aplikasi, prompt, hingga *agent workflow*—langkah berikutnya adalah melihat bentuk tindakan pemeliharaan yang lebih spesifik, yaitu **update**, **patch**, **retraining**, dan **dokumentasi perubahan**. Bagian ini penting karena dalam praktik organisasi, solusi AI hampir tidak pernah benar-benar statis. Seiring waktu, sistem perlu diperbarui, disesuaikan, diperbaiki, atau ditingkatkan. Namun, perubahan pada AI tidak boleh dilakukan secara serampangan. Setiap perubahan dapat mempengaruhi:

- kualitas hasil,
- stabilitas perilaku sistem,
- keamanan,
- fairness,
- kepatuhan,
- dan tingkat kepercayaan pengguna.

Karena itu, organisasi tidak cukup hanya memiliki kemampuan untuk “mengubah” sistem. Organisasi perlu memiliki kemampuan untuk **mengelola perubahan** secara tertib dan terdokumentasi.

Pada standar internasional, **ISO/IEC 22989** menegaskan bahwa pada tahap operation and monitoring, AI system dapat diperbaiki bila ada error, lalu diperbarui pada software, model, dan hardware untuk memenuhi kebutuhan baru dan untuk meningkatkan **performance and reliability**. Standar yang sama juga menegaskan perlunya *continuous validation* dan *re-evaluation* setelah perubahan, agar hasil operasi tetap sesuai dengan tujuan dan risiko yang telah ditetapkan.

Sementara itu, **ISO/IEC 23053** menjelaskan bahwa retraining dapat terjadi karena beberapa sebab, seperti:

- **data drift**,
- **concept drift**,
- perubahan tugas atau target,
- atau kebutuhan optimasi model.

Standar tersebut juga menekankan bahwa perubahan model dan data dapat mempengaruhi *generalization* dan perilaku sistem pada data baru.

Di tingkat implementasi yang lebih praktis, **NIST AI 600-1** menekankan pentingnya pencatatan perubahan, *version history*, metadata, dan *change management records*, termasuk pada sistem yang melibatkan komponen generative AI dan plugin pihak ketiga. Dokumen ini juga menekankan perlunya dokumentasi terhadap bagaimana model pra-latih telah diadaptasi, termasuk *fine-tuning*, retrieval-augmented generation, dan perubahan lain yang berpengaruh pada integritas, privasi, dan keamanan sistem.

Dengan landasan tersebut, subbagian ini akan membahas empat bentuk utama pengelolaan perubahan pada solusi AI:

- **update**,
- **patch**,
- **retraining**,
- dan **dokumentasi perubahan**.

Keempatnya penting karena membantu organisasi menjaga agar AI tidak hanya tetap berjalan, tetapi tetap **layak dipakai**.

7.9.1 Update

Secara umum, **update** adalah pembaruan terhadap komponen sistem untuk menyesuaikan dengan kebutuhan baru, memperbaiki performa, meningkatkan kualitas, atau menjaga kesesuaian dengan konteks organisasi yang terus berubah. Pada solusi AI, update bisa mencakup:

- pembaruan aplikasi;
- pembaruan model;
- pembaruan sumber dokumen;
- pembaruan metadata;
- pembaruan prompt;
- pembaruan *agent workflow*;
- pembaruan integrasi API;
- atau pembaruan komponen infrastruktur pendukung.

Berbeda dari software biasa, update pada AI bisa mempengaruhi perilaku sistem secara lebih luas. Misalnya:

- model baru dapat memberi jawaban yang berbeda untuk pertanyaan yang sama;
- prompt baru dapat mengubah gaya dan struktur keluaran;
- pembaruan corpus dokumen dapat mengubah hasil retrieval;
- dan update agent workflow dapat mengubah urutan langkah yang diambil sistem.

Karena itu, update pada AI perlu dipandang bukan sekadar peremajaan teknis, tetapi juga potensi perubahan perilaku.

Update biasanya diperlukan ketika:

- organisasi memiliki kebutuhan baru;
- sumber data atau dokumen berubah;
- pengguna membutuhkan format hasil yang lebih sesuai;
- model atau layanan yang dipakai sudah tidak optimal;
- ada fitur baru yang lebih baik;
- atau sistem perlu disesuaikan dengan perubahan kebijakan, proses, atau tata kelola.

Dalam konteks pemerintah, update juga bisa dibutuhkan ketika:

- dokumen regulatif berubah;
- klasifikasi administratif diperbarui;
- sistem pemerintahan digital lain berubah;
- atau ada kebutuhan interoperabilitas baru dengan unit atau instansi lain.

Dengan kata lain, update tidak selalu muncul karena sistem “rusak”. Kadang update justru diperlukan karena organisasi berkembang dan AI harus tetap relevan.

Satu hal yang sangat penting adalah bahwa update pada solusi AI tidak sebaiknya langsung diterapkan tanpa evaluasi. Karena perilaku AI sangat dipengaruhi oleh banyak komponen, update yang tampak kecil bisa menghasilkan dampak yang besar.

Sebagai contoh:

- pembaruan model vendor bisa mengubah gaya keluaran generative AI;
- penambahan dokumen baru ke corpus bisa menggeser hasil retrieval;
- perubahan prompt sistem bisa membuat jawaban menjadi lebih panjang, lebih singkat, atau lebih spekulatif;
- perubahan workflow agent bisa mempercepat alur, tetapi sekaligus mengurangi titik pengawasan manusia.

Karena itu, setiap update yang penting perlu:

- diuji;
- dibandingkan dengan perilaku sebelumnya;
- dilihat dampaknya pada fairness, explainability, dan reliability;
- dan, bila perlu, diterapkan secara bertahap.

7.9.2 Patch

Jika update cenderung bersifat pembaruan atau penyesuaian yang lebih umum, maka **patch** biasanya merujuk pada perbaikan yang lebih spesifik terhadap masalah tertentu. Patch bisa diterapkan untuk:

- memperbaiki bug;
- menutup kerentanan keamanan;
- memperbaiki integrasi yang rusak;
- membenarkan prompt yang menghasilkan keluaran bermasalah;
- atau mengatasi perilaku tertentu yang tidak diinginkan pada sistem.

Dalam lingkungan AI, patch bisa muncul pada beberapa lapisan:

- patch pada aplikasi atau layanan;
- patch pada koneksi API;
- patch pada indexing atau retrieval layer;
- patch pada guardrail;
- patch pada *prompt template*;
- atau patch pada batas dan aturan *agent workflow*.

Patch penting karena banyak masalah AI tidak memerlukan perubahan besar, tetapi tetap tidak boleh dibiarkan. Misalnya:

- output tertentu terlalu sering menambahkan informasi yang tidak ada di sumber;
- satu jalur integrasi tertentu menyebabkan keterlambatan besar;
- pengguna dengan hak akses tertentu masih melihat kutipan sumber yang seharusnya tidak muncul;
- atau agent terlalu sering memilih tool tertentu padahal tidak relevan.

Masalah seperti ini kadang tidak memerlukan retraining model, tetapi tetap memerlukan perbaikan cepat dan tepat. Di sinilah patch berperan.

Walaupun patch sering terlihat kecil, ia tidak boleh diperlakukan sembarangan. Patch yang dilakukan terburu-buru tanpa dokumentasi atau pengujian bisa memunculkan masalah baru. Karena itu, organisasi perlu memastikan bahwa patch:

- punya alasan yang jelas;
- diuji setidaknya pada konteks yang relevan;
- tidak bertabrakan dengan komponen lain;
- dan dicatat dalam riwayat perubahan.

Hal ini penting terutama untuk generative AI dan agentic AI, karena perubahan kecil pada prompt, guardrail, atau langkah kerja bisa berdampak luas terhadap perilaku sistem.

7.9.3 Retraining

Retraining adalah proses melatih ulang model dengan data baru, data yang diperbarui, atau konfigurasi yang disesuaikan agar model kembali sesuai dengan kebutuhan dan konteks operasional. Retraining biasanya lebih relevan pada solusi AI berbasis machine learning yang modelnya memang dikembangkan atau dikendalikan oleh organisasi, atau pada sistem di mana ada ruang penyesuaian model yang cukup besar.

ISO/IEC 23053 menjelaskan bahwa retraining dapat dipicu oleh:

- data drift;
- concept drift;
- perubahan target;
- penambahan data baru;
- atau kebutuhan optimasi model.

Dalam praktik, retraining berarti organisasi tidak sekadar memperbaiki permukaan sistem, tetapi memperbarui dasar pembelajaran model agar ia kembali lebih sesuai dengan kondisi nyata.

Retraining dapat menjadi relevan ketika:

- kualitas prediksi atau klasifikasi menurun secara konsisten;
- jenis data baru muncul dan tidak tercakup cukup baik oleh model lama;
- distribusi kategori berubah;
- definisi target atau label berubah;
- atau organisasi menemukan bahwa model lama terlalu bergantung pada pola yang tidak lagi relevan.

Misalnya:

- sistem klasifikasi aduan yang dulu cukup baik menjadi lemah karena jenis aduan baru meningkat;
- model penandaan prioritas tidak lagi sesuai karena proses kerja berubah;
- atau sistem prediksi sederhana menjadi kurang akurat karena perilaku data produksi bergeser.

Penting dipahami bahwa retraining tidak selalu menjadi jawaban atas semua masalah AI. Kadang penurunan kualitas justru disebabkan oleh:

- sumber dokumen yang buruk;
- metadata yang tidak tertata;
- prompt yang kurang tepat;
- integrasi retrieval yang bermasalah;
- atau misuse oleh pengguna.

Jika masalah sebenarnya ada di prompt atau di corpus dokumen, retraining model mungkin tidak akan membantu banyak. Karena itu, sebelum melakukan retraining,

organisasi perlu cukup yakin bahwa masalah memang berada pada model atau hubungan model dengan data.

Setelah retraining dilakukan, model tidak boleh langsung dianggap lebih baik. Organisasi perlu melakukan validasi ulang untuk menilai:

- apakah performa memang membaik;
- apakah ada kategori atau jenis kasus tertentu yang justru memburuk;
- apakah fairness tetap terjaga;
- dan apakah perilaku model tetap sesuai dengan batas yang ditetapkan.

Dengan kata lain, retraining bukan akhir dari masalah, tetapi bagian dari siklus:

- identifikasi penurunan kualitas,
- perbaikan model,
- validasi ulang,
- lalu monitoring berkelanjutan.

7.9.4 Dokumentasi perubahan

Dari semua komponen pengelolaan perubahan, **dokumentasi perubahan** adalah salah satu yang paling sering diremehkan, padahal sangat penting. Tanpa dokumentasi, organisasi akan sulit menjawab pertanyaan:

- apa yang berubah;
- kapan perubahan dilakukan;
- mengapa perubahan dilakukan;
- siapa yang menyetujui perubahan;
- apa dampak yang diharapkan;
- dan apakah perubahan tersebut benar-benar memperbaiki sistem.

Dalam AI, dokumentasi perubahan lebih penting lagi karena perilaku sistem bisa dipengaruhi oleh banyak hal sekaligus:

- model baru,
- data baru,
- prompt baru,
- workflow baru,
- sumber baru,
- atau patch kecil di guardrail.

Jika semua itu tidak dicatat dengan baik, organisasi akan sulit menelusuri penyebab peningkatan atau penurunan kualitas.

Dokumentasi perubahan sebaiknya mencatat minimal:

- komponen apa yang diubah;
- versi sebelum dan sesudah perubahan;
- alasan perubahan;
- data atau sumber yang terpengaruh;
- risiko yang dipertimbangkan;
- hasil pengujian atau validasi setelah perubahan;
- siapa yang melakukan dan menyetujui perubahan;
- serta apakah perubahan diterapkan penuh, bertahap, atau masih dalam masa uji.

Untuk generative AI, dokumentasi perubahan juga sangat penting pada:

- perubahan *prompt template*;
- perubahan *system prompt*;
- perubahan aturan guardrail;
- dan perubahan sumber atau corpus dokumen.

Untuk agentic AI, dokumentasi perubahan penting pada:

- urutan workflow;
- tool yang dipakai agent;
- titik persetujuan manusia;
- dan batas tindakan yang diperbolehkan.

Dokumentasi perubahan bukan hanya bermanfaat untuk tim teknis. Ia juga penting untuk:

- audit;
- evaluasi berkala;
- pengelolaan risiko;
- pelaporan ke pimpinan;
- serta pembelajaran organisasi.

Misalnya, bila kualitas ringkasan menurun setelah pembaruan model vendor, dokumentasi perubahan membantu organisasi memahami apakah penurunan itu berkaitan dengan:

- model baru;
- prompt yang ikut berubah;
- dokumen sumber yang bertambah;
- atau perubahan pada workflow.

Tanpa dokumentasi, organisasi hanya akan menebak-nebak.

Pada akhirnya, dokumentasi perubahan adalah bagian dari **governance**. Ia membantu memastikan bahwa AI tidak berubah diam-diam di balik layar tanpa kendali organisasi. Ini penting terutama di sektor publik, karena penggunaan AI harus dapat dijelaskan dan dipertanggungjawabkan secara kelembagaan.

Dengan dokumentasi yang baik, organisasi dapat lebih mudah:

- menjelaskan evolusi sistem;
- menunjukkan bahwa perubahan dilakukan secara sadar dan teruji;
- dan menentukan kapan sebuah sistem layak dipertahankan, diperbaiki, atau diganti.

Update, patch, retraining, dan dokumentasi perubahan adalah bagian penting dari pemeliharaan solusi AI yang matang. Update membantu menyesuaikan sistem dengan kebutuhan dan konteks baru. Patch memperbaiki masalah spesifik tanpa harus selalu melakukan perubahan besar. Retraining diperlukan ketika model tidak lagi cukup mewakili data dan kebutuhan operasional. Dokumentasi perubahan menjaga agar seluruh modifikasi sistem dapat ditelusuri, dievaluasi, dan dipertanggungjawabkan. Standar AI dan pedoman tata kelola menegaskan pentingnya repair, update, retraining, continuous validation, dan pencatatan perubahan untuk menjaga performance, reliability, keamanan, dan akuntabilitas sistem sepanjang siklus hidupnya.

7.10 Business continuity dan fallback

Setelah membahas evaluasi, monitoring, pemeliharaan, update, patch, retraining, dan dokumentasi perubahan, langkah berikutnya adalah melihat satu aspek yang sangat penting dalam penggunaan AI di organisasi, yaitu **business continuity** dan **fallback**. Bagian ini penting karena sebaik apa pun sistem AI dirancang, diuji, dan dipelihara, organisasi tetap harus siap menghadapi kenyataan bahwa:

- sistem dapat gagal;
- layanan vendor dapat terganggu;
- kualitas hasil dapat turun mendadak;
- sumber data dapat tidak tersedia;
- atau AI dapat berada pada kondisi yang membuat hasilnya tidak layak digunakan.

Dalam situasi seperti itu, pertanyaan pentingnya bukan lagi hanya “mengapa AI gagal”, tetapi juga: **apa yang dilakukan organisasi agar proses kerja tetap berjalan?** Inilah inti dari business continuity dan fallback.

Bagi organisasi pemerintah, hal ini sangat penting karena banyak proses tidak boleh berhenti hanya karena AI terganggu. Walaupun AI dapat membantu mempercepat kerja, memperbaiki orientasi awal, atau mengurangi beban repetitif, tanggung jawab organisasi tetap berjalan. Layanan tetap harus diberikan. Dokumen tetap harus diproses. Informasi tetap harus dikelola. Karena itu, AI tidak boleh menjadi satu-satunya titik tumpu yang bila gagal langsung membuat proses organisasi lumpuh.

Pada tingkat keamanan dan tata kelola, kebutuhan ini juga sudah ditegaskan. **NIST AI 600-1** menekankan pentingnya *contingency planning* dan kesiapan menghadapi masalah pada pihak ketiga, termasuk kebutuhan akan rencana bila layanan AI eksternal mengalami gangguan atau perubahan yang berdampak pada operasi organisasi. Sementara itu, **ISO/IEC 22989** menegaskan bahwa pada tahap operation and monitoring, AI system perlu dipantau untuk operasi normal dan insiden, didukung dengan mekanisme perbaikan, dan dievaluasi ulang terhadap tujuan serta risiko yang telah ditetapkan.

Dari sudut kompetensi operasional, arah yang sama juga tampak pada SKKNI AI yang menekankan penanganan masalah berdasarkan hasil monitoring dan log, serta penyesuaian perawatan komponen solusi AI berdasarkan kebutuhan operasional yang nyata.

Dengan dasar itu, business continuity dan fallback dapat dipahami sebagai dua hal yang saling melengkapi:

- **business continuity** berarti kemampuan organisasi menjaga agar proses kerja penting tetap berjalan meskipun solusi AI mengalami gangguan, penurunan kualitas, atau tidak dapat dipakai sementara;
- **fallback** berarti mekanisme alternatif yang digunakan ketika AI tidak layak, tidak tersedia, atau tidak boleh dipakai pada suatu kondisi.

Dalam banyak organisasi, ada kecenderungan untuk menempatkan AI sebagai alat bantu tambahan. Pada tahap awal, ini memang benar. Namun, seiring waktu, ketika AI mulai:

- dipakai lebih sering;
 - diintegrasikan ke workflow;
 - diandalkan oleh banyak pengguna;
 - atau menjadi bagian dari proses rutin;
- maka ketergantungan organisasi terhadap AI juga meningkat.

Di sinilah risiko muncul. Bila organisasi tidak menyiapkan business continuity, maka ketika AI:

- lambat,
 - salah,
 - tidak tersedia,
 - atau harus dimatikan sementara,
- proses kerja bisa terganggu lebih besar daripada yang diperkirakan.

Sebagai contoh:

- jika knowledge assistant internal menjadi sumber utama pegawai untuk mencari rujukan, lalu sistem tidak tersedia, apakah pegawai masih tahu jalur manual mencari sumber resmi?
- jika ringkasan AI menjadi bagian rutin briefing, lalu sistem memberi keluaran yang meragukan, apakah staf masih punya prosedur kerja untuk menyiapkan bahan secara manual?
- jika agentic workflow dipakai untuk triase atau sintesis awal, lalu kualitasnya turun, apakah ada mekanisme untuk kembali ke alur kerja yang lebih aman tanpa menghentikan seluruh proses?

Business continuity diperlukan justru karena AI semakin berguna. Semakin besar manfaat AI, semakin penting organisasi menyiapkan jalan ketika AI tidak dapat dipakai.

7.10.1 Business continuity bukan berarti menolak AI

Penting untuk dipahami bahwa business continuity bukan tanda bahwa organisasi tidak percaya pada AI. Justru sebaliknya, business continuity menunjukkan bahwa organisasi cukup matang untuk memahami bahwa AI adalah bagian dari sistem kerja, dan setiap bagian sistem kerja harus punya ketahanan.

Dalam pemerintahan, pendekatan seperti ini sangat penting. Organisasi tidak bisa berkata:

- “karena AI gagal, proses tidak bisa jalan”;
 - atau “karena vendor bermasalah, layanan harus berhenti.”
- Cara berpikir seperti itu tidak sesuai dengan tanggung jawab kelembagaan.

Karena itu, menyiapkan continuity plan justru merupakan bagian dari penggunaan AI yang sehat dan bertanggung jawab.

7.10.2 Fallback

Fallback adalah mekanisme alternatif yang sudah dipikirkan sebelumnya dan dapat dipakai ketika AI:

- tidak tersedia;
- tidak menghasilkan keluaran yang cukup layak;
- tidak menemukan sumber yang cukup jelas;
- menghasilkan hasil yang meragukan;
- atau tidak boleh melanjutkan karena konteks terlalu sensitif atau ambigu.

Fallback penting karena organisasi tidak boleh memaksa AI bekerja pada semua situasi. Ada kondisi ketika sistem sebaiknya:

- berhenti;
- menyerahkan ke manusia;
- mengalihkan ke proses manual;
- atau menghasilkan keluaran yang lebih terbatas daripada mengambil risiko yang terlalu besar.

Dengan kata lain, fallback adalah bentuk kedewasaan desain. Ia menunjukkan bahwa organisasi tidak hanya merancang jalur “saat AI berjalan baik”, tetapi juga jalur “saat AI tidak boleh atau tidak bisa diandalkan”.

Fallback tidak sama dengan kegagalan. Bila fallback berjalan baik, justru itu tanda bahwa sistem dirancang secara sehat. Misalnya:

- knowledge assistant tidak menemukan sumber yang cukup pasti, lalu mengarahkan pengguna ke repository resmi dan menyatakan bahwa jawaban tidak dapat disimpulkan otomatis;
- sistem generative AI menolak menyusun jawaban final dan hanya memberi poin awal karena sumbernya lemah;
- agentic AI berhenti sebelum langkah berikutnya dan meminta persetujuan manusia;
- atau workflow kembali ke proses manual ketika latency terlalu tinggi atau kualitas hasil menurun.

Dalam semua contoh itu, fallback bukan kegagalan. Ia adalah **mekanisme perlindungan**.

Organisasi perlu cukup jelas mengenai **kondisi pemicu fallback**. Ini penting agar fallback tidak bergantung pada intuisi individu semata. Beberapa kondisi yang dapat memicu fallback antara lain:

1. Ketika kualitas hasil diragukan

Bila sistem:

- tidak menemukan sumber yang cukup kuat;
- menghasilkan ringkasan yang tampak terlalu bebas;
- atau agent memberi hasil antara yang tidak meyakinkan, maka fallback bisa berarti menghentikan otomatisasi dan menyerahkan ke manusia.

2. Ketika sumber atau data tidak tersedia

Bila dokumen yang dibutuhkan tidak dapat diakses, metadata rusak, atau corpus belum diperbarui, maka AI sebaiknya tidak dipaksa memberi jawaban penuh.

3. Ketika sistem terlalu lambat

Pada beberapa proses, latency yang terlalu tinggi membuat AI tidak lagi cocok untuk ritme kerja. Dalam kondisi ini, organisasi mungkin perlu kembali ke prosedur kerja biasa untuk sementara.

4. Ketika ada dugaan misuse atau incident

Jika terdeteksi penggunaan di luar tujuan, potensi data leakage, atau keluaran yang berisiko tinggi, fallback dapat berupa penghentian sementara fitur tertentu.

5. Ketika konteks terlalu sensitif

Untuk konteks yang menyangkut data sensitif, pengecualian penting, komunikasi resmi, atau keputusan yang lebih berdampak, fallback bisa berarti AI hanya memberi bantuan minimal atau tidak dipakai sama sekali tanpa verifikasi manusia.

6. Ketika vendor atau layanan pihak ketiga bermasalah

Jika model eksternal, plugin, atau layanan vendor tidak tersedia atau berubah perilakunya secara signifikan, organisasi perlu punya jalur kerja alternatif.

7.10.3 Bentuk-bentuk fallback

Fallback tidak harus selalu berarti “kembali sepenuhnya manual”. Ada beberapa bentuk fallback yang bisa dipilih sesuai konteks.

1. Fallback ke manusia

Ini bentuk yang paling umum. AI berhenti pada titik tertentu dan menyerahkan proses ke manusia untuk:

- memeriksa,
- melanjutkan,
- atau memutuskan langkah berikutnya.

2. Fallback ke mode terbatas

Sistem masih berjalan, tetapi fungsinya dipersempit. Misalnya:

- AI hanya menampilkan sumber tanpa menyusun jawaban;
- hanya memberi poin-poin kasar;
- atau hanya melakukan retrieval tanpa generasi bahasa yang bebas.

3. Fallback ke proses manual

Organisasi kembali ke prosedur kerja manual yang sudah dikenal, misalnya pencarian dokumen langsung, klasifikasi manual, atau penyusunan briefing tanpa bantuan AI.

4. Fallback ke sistem alternatif

Dalam beberapa kasus, organisasi mungkin punya layanan atau metode lain yang lebih sederhana tetapi lebih stabil untuk sementara.

5. Fallback ke ruang lingkup yang dipersempit

Agent atau generative AI hanya dipakai untuk jenis tugas yang lebih aman, sementara konteks yang lebih sensitif ditangguhkan.

Pilihan bentuk fallback tergantung pada risiko use case, ketergantungan organisasi, dan kesiapan proses manual atau semi-manual yang menyertainya.

Untuk sistem seperti ini, continuity plan perlu menjawab:

- jika sistem tidak menemukan jawaban yang layak, ke mana pengguna harus diarahkan;
- bagaimana pengguna tetap dapat mengakses repositori resmi secara manual;
- dan bagaimana organisasi memastikan pegawai tidak kehilangan akses terhadap sumber saat AI sedang tidak tersedia.

Fallback bisa berupa:

- penyusunan bahan awal secara manual;
- penggunaan template kerja biasa tanpa bantuan AI;
- atau pembatasan AI hanya pada ekstraksi poin, bukan narasi penuh.

Business continuity perlu memastikan bahwa:

- bila sistem klasifikasi gagal, proses manual tetap dapat dilakukan;
- kategori dan alur disposisi tetap jelas;
- dan backlog tidak membesar hanya karena AI berhenti.

Pada GenAI Fallback dapat berupa:

- hanya menampilkan dokumen sumber yang relevan tanpa jawaban generatif;
- mengurangi ruang lingkup jawaban;
- atau mewajibkan verifikasi manusia sebelum hasil ditampilkan penuh.

Untuk agentic AI, continuity dan fallback jauh lebih penting. Organisasi perlu menentukan:

- langkah mana yang boleh otomatis dan mana yang tidak;
- kapan agent harus berhenti;
- bagaimana manusia mengambil alih workflow;
- dan bagaimana jika tool tertentu tidak tersedia.

7.10.4 Business continuity memerlukan kesiapan proses manual

Salah satu hal yang sering luput adalah bahwa fallback yang baik memerlukan **proses manual yang masih hidup**. Jika organisasi terlalu lama bergantung pada AI tanpa menjaga pengetahuan kerja dasarnya, maka saat AI gagal:

- pegawai kebingungan;
- sumber tidak mudah ditemukan;
- dan proses kerja justru terganggu lebih besar.

Karena itu, business continuity juga berarti menjaga agar organisasi:

- tetap tahu jalur manual;
- tetap punya dokumentasi proses;
- dan tetap punya kapasitas manusia untuk mengambil alih bila perlu.

Ini sangat penting di sektor publik, karena kontinuitas layanan dan akuntabilitas proses tidak boleh bergantung sepenuhnya pada satu sistem AI.

Seringkali organisasi sudah mengatakan bahwa “bila AI gagal, manusia akan mengambil alih.” Pernyataan seperti ini belum cukup bila tidak diterjemahkan ke dalam workflow yang jelas. Fallback yang sehat perlu menjawab:

- siapa yang mengambil alih;
- pada titik mana pengambilalihan terjadi;
- bagaimana pengguna tahu bahwa fallback sedang aktif;
- apa yang harus dilakukan setelah fallback dipicu;
- dan bagaimana kualitas serta waktu kerja tetap dijaga.

Tanpa desain seperti ini, fallback hanya menjadi niat baik yang tidak operasional.

Contoh:

- jika knowledge assistant gagal, apakah pengguna diarahkan ke portal dokumen resmi?
- jika agent berhenti, apakah ada notifikasi yang jelas kepada petugas?
- jika sistem generatif hanya boleh memberi poin awal, apakah pengguna tahu bahwa hasil belum layak dipakai sebagai dokumen final?
- jika vendor eksternal mati, apakah sistem masih punya cache, repositori lokal, atau prosedur kerja alternatif?

Pertanyaan seperti ini membantu organisasi merancang fallback yang nyata.

Dalam banyak solusi AI modern, ketahanan organisasi juga bergantung pada:

- vendor model;
- API pihak ketiga;
- plugin;
- layanan cloud;
- dan infrastruktur digital pendukung.

Karena itu, business continuity tidak boleh dibaca hanya dari sisi pengguna akhir. Organisasi juga perlu mempertimbangkan:

- apa yang terjadi bila vendor berubah layanan;
- apa yang terjadi bila API gagal;
- apa yang terjadi bila model diperbarui dan kualitas turun;
- dan apa yang terjadi bila infrastruktur pendukung tidak stabil.

Inilah sebabnya NIST AI 600-1 menekankan pentingnya *contingency planning* untuk layanan pihak ketiga, termasuk inventarisasi pihak yang punya akses ke konten organisasi, pencatatan perubahan, dan rencana saat layanan terganggu.

7.10.5 Business continuity sebagai bagian dari tata kelola

Pada akhirnya, business continuity dan fallback bukan hanya persoalan teknis. Keduanya adalah bagian dari **tata kelola AI**. Organisasi yang matang akan bertanya:

- proses mana yang terlalu kritis untuk bergantung penuh pada AI;
- bentuk fallback apa yang paling realistis;
- siapa yang mengambil alih bila AI gagal;
- bagaimana insiden dicatat;
- dan kapan masalah kualitas cukup serius untuk memicu pembatasan atau penghentian sementara.

Dengan kata lain, continuity plan membantu organisasi menjaga agar AI tetap menjadi alat bantu, bukan sumber kerentanan kelembagaan yang baru.

Business continuity dan fallback adalah bagian penting dari pengelolaan AI yang matang. Organisasi tidak boleh hanya merancang kondisi ketika AI berjalan baik, tetapi juga harus menyiapkan kondisi ketika AI:

- tidak tersedia,
- tidak cukup layak,
- terlalu lambat,
- terlalu berisiko,
- atau tidak boleh dipakai pada konteks tertentu.

Business continuity menjaga agar proses kerja dan layanan tetap berjalan, sementara fallback menyediakan jalur alternatif ketika AI tidak dapat diandalkan. Standar dan acuan tata kelola menegaskan pentingnya pemantauan operasi,

penanganan insiden, dukungan pengguna, dan *contingency planning*, termasuk untuk layanan pihak ketiga.

7.11 Kapan sistem dibatasi, diganti, atau dihentikan

Setelah membahas evaluasi, monitoring, pemeliharaan, pembaruan, *retraining*, *business continuity*, dan *fallback*, ada satu pertanyaan penting yang sering dihindari dalam praktik organisasi, yaitu: **kapan sebuah solusi AI tidak lagi layak dipertahankan dalam bentuk yang sama?** Pertanyaan ini penting karena dalam banyak organisasi, begitu sebuah sistem AI sudah dibangun dan mulai dipakai, ada kecenderungan untuk terus mempertahankannya, meskipun tanda-tanda masalah mulai muncul. Hal ini bisa terjadi karena:

- organisasi sudah mengeluarkan waktu dan sumber daya;
- pengguna sudah terbiasa;
- pimpinan sudah menganggap sistem tersebut sebagai simbol kemajuan;
- atau tidak ada mekanisme yang jelas untuk menilai apakah sistem masih layak diteruskan.

Padahal, salah satu ciri organisasi yang matang adalah kemampuannya untuk mengatakan:

- kapan sistem cukup baik untuk diteruskan,
- kapan perlu dibatasi ruang lingkupnya,
- kapan perlu diganti dengan pendekatan lain,
- dan kapan sebaiknya dihentikan.

Dalam konteks AI, keputusan seperti ini sangat penting karena AI bukan hanya alat teknis. AI mempengaruhi:

- kualitas informasi;
- ritme kerja organisasi;
- cara pengguna mengambil keputusan awal;
- keamanan data;
- akuntabilitas proses;
- dan kepercayaan terhadap sistem digital secara keseluruhan.

Karena itu, mempertahankan sistem AI yang sebenarnya tidak lagi layak bisa lebih berbahaya daripada menghentikannya.

Ada kecenderungan dalam adopsi teknologi untuk melihat keberlanjutan sistem sebagai sesuatu yang selalu positif. Seolah-olah jika sebuah sistem terus berjalan, maka itu tanda keberhasilan. Dalam AI, cara pandang seperti ini tidak selalu benar. Sistem yang tetap hidup belum tentu tetap bermanfaat, aman, atau relevan.

Beberapa masalah yang dapat muncul bila sistem AI dipertahankan terlalu lama tanpa peninjauan yang jujur antara lain:

- kualitas keluaran menurun tetapi tidak disadari;
- pengguna mulai kehilangan kepercayaan, tetapi organisasi tidak menangkap sinyalnya;
- proses kerja tetap bergantung pada AI walaupun hasilnya makin banyak dikoreksi;
- risiko keamanan, privasi, atau bias meningkat;
- vendor atau model berubah, tetapi organisasi tidak lagi punya kendali yang cukup;
- atau solusi yang dulu berguna menjadi kurang relevan karena kebutuhan organisasi sudah berubah.

Karena itu, keputusan untuk membatasi, mengganti, atau menghentikan sistem sebaiknya tidak dipandang sebagai kegagalan, melainkan sebagai bagian dari tata kelola yang sehat.

Salah satu jebakan umum dalam pengelolaan AI adalah *sunk cost thinking*, yaitu kecenderungan mempertahankan sistem hanya karena organisasi merasa sudah terlanjur berinvestasi. Cara berpikir seperti ini berbahaya. Yang seharusnya menjadi dasar keputusan bukanlah berapa banyak waktu atau biaya yang sudah dihabiskan, tetapi:

- apakah sistem masih memberi manfaat yang cukup;
- apakah kualitasnya masih layak;
- apakah risikonya masih dapat diterima;
- dan apakah ada alternatif yang lebih tepat.

Dalam sektor publik, hal ini lebih penting lagi karena yang dipertaruhkan bukan hanya efisiensi internal, tetapi juga kualitas tata kelola, mutu layanan, dan kepercayaan terhadap institusi.

Secara umum, ketika hasil evaluasi menunjukkan bahwa suatu sistem AI tidak lagi ideal, organisasi biasanya punya tiga pilihan besar:

- **membatasi** sistem;
- **mengganti** sistem;
- atau **menghentikan** sistem.

Pilihan-pilihan ini tidak harus dipahami sebagai urutan otomatis, tetapi sebagai spektrum respons tergantung tingkat masalah, risiko, dan manfaat yang tersisa.

7.11.1 Pembatasan sebagai langkah korektif yang proporsional

Membatasi sistem berarti organisasi tidak langsung mematikan AI sepenuhnya, tetapi **memperkecil ruang lingkup penggunaan** agar risiko turun dan manfaat yang masih sah tetap dapat dipertahankan. Pembatasan sering menjadi pilihan yang sehat ketika:

- sistem masih memberi manfaat pada konteks tertentu;
- tetapi tidak lagi aman atau cukup andal untuk konteks yang lebih luas;
- atau kualitasnya hanya baik untuk jenis tugas tertentu.

Dengan kata lain, pembatasan adalah cara untuk mengatakan: “AI ini masih berguna, tetapi tidak untuk semua hal yang selama ini dibebankan padanya.”

7.11.2 Tanda-tanda bahwa sistem perlu dibatasi

Beberapa kondisi yang menunjukkan bahwa pembatasan mungkin lebih tepat daripada penerusan penuh antara lain:

7.11.2.1. Kualitas hasil tidak merata

Sistem cukup baik untuk beberapa jenis kasus, tetapi lemah untuk kasus lain. Misalnya:

- knowledge assistant baik untuk dokumen SOP, tetapi lemah untuk dokumen kebijakan yang lebih kompleks;
- generative AI baik untuk ringkasan awal, tetapi tidak layak untuk drafting final;
- agentic AI baik untuk pencarian dan sintesis awal, tetapi terlalu berisiko untuk langkah tindak lanjut otomatis.

7.11.2.2. Risiko meningkat pada konteks tertentu

Sistem mungkin cukup aman untuk penggunaan internal terbatas, tetapi terlalu berisiko untuk dipakai pada data sensitif, komunikasi resmi, atau proses yang lebih berdampak.

7.11.3. Fairness atau reliability tidak cukup stabil

Jika evaluasi menunjukkan bahwa sistem:

- sering gagal pada kategori tertentu;
- terlalu berubah-ubah pada variasi kasus tertentu;
- atau hanya reliabel pada kondisi yang sempit,
maka pembatasan ruang lingkup bisa menjadi langkah yang masuk akal.

7.11.4. Pengawasan manusia belum cukup kuat

Kadang sistem sebenarnya belum siap dipakai luas, bukan karena modelnya buruk sekali, tetapi karena kontrol prosesnya belum matang. Dalam kondisi seperti ini, pembatasan dapat dilakukan sambil memperkuat *human oversight*, logging, atau guardrail.

7.11.3 Bentuk-bentuk pembatasan

Pembatasan bisa dilakukan dalam berbagai bentuk, misalnya:

- hanya membolehkan AI untuk penggunaan internal;
- membatasi AI pada jenis dokumen tertentu;
- membatasi AI pada fungsi retrieval saja, tanpa generasi bebas;
- membatasi agent hanya pada langkah tertentu dalam workflow;
- mewajibkan verifikasi manusia untuk semua hasil pada kategori tertentu;
- atau membatasi penggunaan AI hanya untuk tim atau unit yang sudah terlatih.

Pendekatan ini penting karena tidak semua masalah harus berakhir pada penghentian total. Kadang pembatasan justru merupakan bentuk tata kelola yang paling realistis dan bertanggung jawab.

Mengganti sistem berarti organisasi memutuskan bahwa pendekatan yang saat ini dipakai tidak lagi paling tepat, sehingga perlu beralih ke:

- model lain;
- vendor lain;
- arsitektur lain;
- workflow lain;
- atau bahkan pendekatan non-AI yang lebih sesuai.

Keputusan mengganti sistem tidak selalu berarti sistem lama gagal total. Kadang sistem perlu diganti karena:

- kebutuhan organisasi berubah;
- teknologi yang dipakai sudah tidak cocok;
- struktur data berubah;
- vendor tidak lagi memenuhi syarat;
- atau ada pendekatan yang lebih aman, lebih sederhana, atau lebih bernilai.

Adapun tanda-tanda bahwa sistem perlu diganti

1. Arsitektur sudah tidak cocok

Sistem mungkin dibangun untuk kebutuhan awal yang sempit, tetapi sekarang organisasi memerlukan integrasi, interoperabilitas, atau skalabilitas yang tidak bisa dicapai dengan desain lama.

2. Vendor atau model tidak lagi memadai

Misalnya:

- vendor terlalu sering berubah tanpa kendali yang cukup;
- kualitas model terlalu tidak stabil;
- biaya tidak lagi sebanding dengan manfaat;
- atau risiko vendor menjadi terlalu besar.

3. Perbaikan kecil tidak lagi cukup

Jika masalah sistem terus berulang dan hanya ditambah dengan patch atau prompt tweak kecil, maka organisasi perlu bertanya apakah pendekatannya sendiri yang sudah tidak tepat.

4. Kebutuhan organisasi berubah secara signifikan

AI yang awalnya dibuat untuk membantu orientasi dokumen mungkin tidak lagi cukup ketika organisasi kini membutuhkan integrasi lintas sistem atau workflow yang lebih terkendali.

5. Ada pendekatan yang lebih sederhana dan lebih aman

Kadang sistem AI yang kompleks diganti bukan karena AI itu buruk, tetapi karena ternyata masalah organisasi bisa diselesaikan lebih baik dengan:

- retrieval yang lebih sederhana;
- aturan kerja yang lebih jelas;
- otomasi biasa;
- atau model yang lebih kecil tetapi lebih transparan.

Dalam organisasi pemerintah, mengganti sistem tidak boleh dilakukan secara mendadak tanpa transisi. Organisasi perlu memikirkan:

- bagaimana pengguna berpindah;
- bagaimana data dan dokumentasi dijaga;
- bagaimana proses lama tetap berjalan selama masa peralihan;
- dan bagaimana keputusan penggantian dijelaskan.

Karena itu, keputusan mengganti sistem sangat terkait dengan *business continuity* dan dokumentasi perubahan.

7.11.4 Penghentian sebagai keputusan tata kelola

Menghentikan sistem berarti organisasi memutuskan bahwa solusi AI tersebut **tidak lagi layak dipakai sama sekali** dalam bentuk operasionalnya. Ini adalah keputusan yang paling tegas, tetapi kadang justru merupakan pilihan yang paling sehat.

Penghentian perlu dipandang sebagai keputusan tata kelola yang sah ketika manfaat yang tersisa tidak lagi cukup membenarkan biaya, risiko, atau dampak negatif sistem.

1. Risiko terlalu tinggi dan sulit dikendalikan

Jika sistem:

- berulang kali menghasilkan keluaran berbahaya;
- menimbulkan incident serius;
- membocorkan data;
- atau terlalu mudah di misuse,
dan masalah itu tidak dapat diperbaiki secara proporsional, maka penghentian perlu dipertimbangkan.

2. Manfaat nyata sangat rendah

Sistem bisa saja masih berjalan, tetapi bila:

- pengguna tidak lagi memakainya;
- hasil terlalu sering dikoreksi;
- atau dampaknya pada kerja nyata sangat kecil,
maka organisasi perlu jujur menilai apakah sistem ini layak dipertahankan.

3. Kualitas terus menurun dan tidak ekonomis diperbaiki

Ada situasi ketika masalah teknis, kualitas data, dan perubahan konteks sudah begitu besar sehingga biaya memperbaiki sistem tidak sebanding dengan manfaat yang mungkin didapat.

4. Konteks kebijakan atau kelembagaan berubah

Sistem bisa menjadi tidak relevan atau tidak patut dipakai lagi karena:

- aturan berubah;
- sumber resmi berubah secara besar;
- mandat organisasi berubah;
- atau pendekatan itu tidak lagi selaras dengan kebijakan dan tata kelola yang berlaku.

5. Sistem menimbulkan ketergantungan yang rapuh

Jika organisasi terlalu bergantung pada sistem yang sebenarnya tidak reliabel, dan ketergantungan itu mulai membahayakan proses kerja, maka penghentian bisa lebih sehat daripada mempertahankan ilusi bantuan.

Meskipun sebuah sistem dihentikan, proses penghentiannya tetap perlu tertib. Organisasi perlu:

- mendokumentasikan alasan penghentian;
- menentukan kapan dan bagaimana penghentian berlaku;
- memastikan proses kerja alternatif tersedia;
- mengarsipkan log, dokumentasi, atau artefak penting bila perlu;
- dan memberi tahu pihak-pihak yang terdampak.

Penghentian yang tidak tertib justru bisa menciptakan masalah baru, misalnya pengguna masih mengira sistem bisa dipakai, atau tidak ada jalur kerja alternatif yang jelas.

Agar keputusan membatasi, mengganti, atau menghentikan sistem tidak bersifat subjektif, organisasi perlu menilai beberapa hal secara sistematis.

1. Manfaat organisasi

Apakah sistem masih memberi manfaat nyata?

Apakah manfaat itu cukup besar dibanding biaya, beban pemeliharaan, dan risiko?

2. Kualitas teknis

Apakah performa sistem masih cukup baik?

Apakah kualitasnya menurun?

Apakah fairness, explainability, dan reliability masih memadai?

3. Resiko

Apakah risiko keamanan, privasi, misuse, atau over-automation masih dapat diterima?

Apakah incident yang terjadi menunjukkan masalah yang lebih mendasar?

4. Kesiapan kontrol

Apakah organisasi punya *human oversight*, guardrail, logging, dan fallback yang cukup?

Jika belum, apakah pembatasan atau penghentian sementara diperlukan?

5. Kesesuaian dengan tujuan dan kebijakan

Apakah sistem masih selaras dengan tujuan awal, konteks kerja saat ini, dan arah kebijakan organisasi?

6. Alternatif yang tersedia

Apakah ada pendekatan lain yang lebih tepat, lebih aman, atau lebih efisien?

Dengan membaca semua unsur ini bersama-sama, organisasi dapat membuat keputusan yang lebih matang.

Keputusan membatasi, mengganti, atau menghentikan sistem AI tidak sebaiknya diambil oleh satu pihak saja. Karena AI menyentuh banyak dimensi, maka keputusan seperti ini biasanya perlu melibatkan:

- pemilik proses bisnis;
- tim teknis;
- pengelola data;
- pengelola keamanan atau tata kelola;
- pengguna utama;
- dan bila relevan pimpinan yang berwenang.

Pendekatan lintas fungsi ini penting agar keputusan tidak terlalu teknis semata dan tidak juga hanya berdasarkan persepsi pengguna tanpa dasar evaluasi yang cukup.

7.11.5 Tanda kedewasaan organisasi

Pada akhirnya, kemampuan membatasi, mengganti, atau menghentikan sistem adalah tanda kedewasaan organisasi dalam mengelola AI. Organisasi yang matang:

- tidak terikat buta pada sistem yang sudah terlanjur dibangun;
- tidak mempertahankan AI hanya demi citra inovasi;
- dan tidak takut mengakui bahwa sebuah pendekatan perlu dikoreksi.

Sebaliknya, organisasi yang belum matang cenderung:

- mempertahankan sistem walaupun manfaatnya kecil;
- menutup mata terhadap penurunan kualitas;
- atau takut menghentikan sistem karena menganggap itu sama dengan mengakui kegagalan.

Dalam sektor publik, kedewasaan seperti ini sangat penting karena yang dipertaruhkan bukan hanya keberhasilan proyek, tetapi kualitas tata kelola dan tanggung jawab kelembagaan.

Sistem AI tidak harus selalu dipertahankan dalam bentuk yang sama. Ada situasi ketika sistem lebih tepat **dibatasi**, yaitu saat manfaatnya masih ada tetapi hanya aman untuk ruang lingkup yang lebih sempit. Ada situasi ketika sistem perlu **diganti**, yaitu saat arsitektur, vendor, model, atau pendekatannya tidak lagi cocok dengan kebutuhan organisasi. Ada pula situasi ketika sistem harus **dihentikan**, yaitu saat manfaatnya terlalu kecil, risikonya terlalu besar, atau kualitasnya tidak lagi layak untuk dipertahankan.

7.12 Rangkuman

Bab ini menegaskan bahwa solusi AI tidak cukup hanya dibangun, diintegrasikan, lalu dipakai. AI harus **dievaluasi, dimonitor, dan dipelihara** agar tetap berguna, tetap terkendali, dan tetap layak digunakan dalam organisasi. Jika bab-bab sebelumnya membantu pembaca memilih use case, membangun solusi, menempatkannya dalam tata kelola, serta menyelaraskannya dengan konteks pemerintahan, maka Bab 7 membantu pembaca memahami bagaimana solusi AI dijaga setelah masuk ke operasi nyata.

Pada bagian awal, bab ini menekankan bahwa setelah implementasi, organisasi harus terus bertanya:

- apakah AI masih membantu tujuan organisasi;
- apakah kualitasnya tetap stabil;
- apakah risiko masih terkendali;
- dan apakah sistem masih layak dipertahankan dalam bentuk yang sama.

Pertanyaan-pertanyaan ini penting karena AI bekerja dalam lingkungan yang berubah. Data dapat bergeser. Dokumen dapat diperbarui. Pengguna dapat berubah cara memakai sistem. Model, prompt, vendor, dan workflow juga dapat berubah. Karena itu, AI tidak bisa diperlakukan seperti sistem yang cukup diuji sekali di awal lalu dibiarkan berjalan sendiri.

Bab ini kemudian membahas **evaluasi teknis dan evaluasi manfaat organisasi**. Di sini dijelaskan bahwa AI perlu dinilai dari dua sisi sekaligus. Evaluasi teknis melihat apakah sistem bekerja dengan mutu yang cukup dari sisi fungsi, performa, relevansi, stabilitas, dan kualitas keluaran. Evaluasi manfaat organisasi melihat apakah AI benar-benar membantu tujuan kerja yang ingin dicapai, seperti:

- mengurangi beban kerja repetitif,
- mempercepat pencarian informasi,
- meningkatkan kualitas orientasi awal,
- atau memperkuat konsistensi proses.

Pembahasan ini menekankan bahwa AI yang baik bukan hanya AI yang akurat atau cepat, tetapi AI yang benar-benar memberi nilai dalam kerja nyata.

Selanjutnya, bab ini membahas **fairness, explainability, dan reliability**. Fairness dijelaskan sebagai kebutuhan untuk memastikan bahwa AI tidak menghasilkan pola yang merugikan konteks atau jenis kasus tertentu secara tidak proporsional. explainability dijelaskan sebagai kemampuan menjelaskan bagaimana AI digunakan, apa arti hasilnya, dan bagaimana hasil itu seharusnya ditafsirkan. Reliability dijelaskan sebagai keandalan sistem dalam menghasilkan keluaran yang cukup konsisten, cukup stabil, dan cukup dapat dipercaya untuk konteks penggunaannya. Ketiga aspek ini penting karena organisasi pemerintah tidak cukup hanya mengetahui bahwa sistem “berjalan”, tetapi juga perlu tahu apakah sistem:

- cukup adil,
- cukup bisa dijelaskan,
- dan cukup andal.

Bab ini juga membahas **monitoring drift, latency, misuse, dan incident**. Drift dijelaskan sebagai perubahan pada data, sumber, konteks, atau perilaku sistem yang dapat menurunkan kualitas AI dari waktu ke waktu. Latency dibahas sebagai waktu respons yang mempengaruhi kelayakan operasional AI dalam ritme kerja nyata. Misuse dijelaskan sebagai penggunaan AI di luar tujuan, ruang lingkup, atau aturan yang ditetapkan. Incident dibahas sebagai kejadian yang menunjukkan bahwa sistem gagal bekerja sebagaimana mestinya atau menimbulkan risiko yang memerlukan penanganan dan evaluasi lebih lanjut. Pembahasan ini menekankan bahwa monitoring AI harus lebih luas daripada monitoring aplikasi biasa, karena yang perlu dipantau bukan hanya infrastruktur, tetapi juga kualitas perilaku sistem dan pola penggunaan manusia.

Setelah itu, bab ini membahas **evaluasi prompt dan kualitas keluaran generative AI**. Di sini dijelaskan bahwa pada generative AI, prompt bukan sekadar instruksi biasa, tetapi bagian dari desain kerja yang membentuk perilaku sistem. Karena itu, prompt perlu dievaluasi dari sisi:

- kejelasan tujuan,
- kecukupan konteks,
- batas instruksi,
- format keluaran,
- hubungan dengan sumber,
- dan kecocokannya dengan kebutuhan pengguna.

Sementara itu, kualitas keluaran generative AI perlu dinilai dari sisi relevansi, kesetiaan pada sumber, kelengkapan unsur penting, keterpakaian dalam workflow, keamanan, konsistensi, dan tingkat koreksi yang dibutuhkan. Pembahasan ini menekankan bahwa keluaran generative AI tidak boleh dinilai hanya dari kefasihan bahasa.

Bab ini kemudian membahas **evaluasi dan pengawasan agentic AI**. Di sini dijelaskan bahwa agentic AI harus dinilai bukan hanya dari keluaran akhir, tetapi juga dari:

- cara ia memahami tujuan,
- kualitas pemilihan sumber dan tool,
- urutan langkah yang dijalankan,
- kualitas keluaran antara,
- kepatuhan terhadap batas kewenangan,
- konsistensi perilaku,
- dan keterlacakan langkah-langkahnya.

Pengawasan agentic AI menuntut *human oversight* yang lebih kuat, logging yang memadai, dan evaluasi berbasis skenario. Hal ini penting karena agentic AI tidak hanya menghasilkan teks, tetapi ikut membentuk workflow dan karena itu membawa risiko yang lebih besar.

Pada subbagian berikutnya, bab ini menjelaskan **continuous validation dan re-evaluation**. Di sini ditekankan bahwa penilaian terhadap AI tidak pernah benar-benar selesai. Organisasi perlu terus memvalidasi apakah sistem masih bekerja sesuai tujuan dan masih berada dalam batas risiko yang dapat diterima. Re-evaluation diperlukan ketika ada perubahan penting, seperti:

- drift,
- perubahan vendor,
- perubahan model,
- perluasan use case,
- penurunan kualitas,
- atau perubahan kebijakan dan kebutuhan organisasi.

Dengan demikian, validasi berkelanjutan dan evaluasi ulang diposisikan sebagai bagian penting dari siklus hidup solusi AI.

Setelah itu, bab ini membahas **pemeliharaan data, model, pipeline, aplikasi, prompt, dan agent workflow**. Pembahasan ini menekankan bahwa solusi AI harus dirawat sebagai ekosistem komponen yang saling mempengaruhi. Data perlu diperbarui dan ditata. Model perlu dipantau dan, bila perlu, disesuaikan atau dilatih ulang. Pipeline perlu dijaga alirannya agar tetap benar dan efisien. Aplikasi perlu dirawat agar tetap mendukung pengalaman pengguna dan proses koreksi. Prompt perlu diperbarui agar tetap sesuai konteks. Agent workflow perlu dijaga agar tidak meluas tanpa kontrol. Dengan kata lain, pemeliharaan AI jauh lebih luas daripada sekadar memelihara perangkat lunak biasa.

Bab ini kemudian masuk ke **update, patch, retraining, dan dokumentasi perubahan**. Update dijelaskan sebagai pembaruan yang menyesuaikan sistem dengan kebutuhan baru atau meningkatkan performa. Patch dijelaskan sebagai perbaikan yang lebih spesifik terhadap masalah tertentu. Retraining dijelaskan sebagai pelatihan ulang model ketika terjadi drift, perubahan target, atau penurunan performa yang bermakna. Dokumentasi perubahan dijelaskan sebagai pondasi penting agar organisasi dapat menelusuri apa yang berubah, mengapa berubah, siapa yang menyetujui, dan apa dampak perubahan itu terhadap kualitas dan risiko sistem. Pembahasan ini menekankan bahwa perubahan pada AI tidak boleh dilakukan secara informal tanpa pengujian dan dokumentasi.

Selanjutnya, bab ini membahas **business continuity dan fallback**. Di sini dijelaskan bahwa organisasi tidak boleh bergantung pada AI tanpa menyiapkan jalur alternatif ketika AI:

- tidak tersedia,
- terlalu lambat,
- tidak cukup layak,
- atau tidak boleh dipakai pada konteks tertentu.

Business continuity berarti kemampuan organisasi menjaga proses kerja tetap berjalan. Fallback berarti mekanisme alternatif ketika AI tidak dapat diandalkan. Fallback dapat berupa:

- pengambilalihan oleh manusia,
- pembatasan fungsi AI,
- kembali ke proses manual,
- atau penggunaan sistem alternatif yang lebih sederhana.

Pembahasan ini penting karena menegaskan bahwa AI harus menjadi penguat kapasitas organisasi, bukan sumber kerentanan baru.

Terakhir, bab ini membahas **kapan sistem dibatasi, diganti, atau dihentikan**. Ini adalah salah satu pelajaran penting dalam pengelolaan AI yang matang. Tidak semua sistem AI harus dipertahankan terus-menerus. Ada kondisi ketika sistem lebih tepat:

- dibatasi ruang lingkupnya,
- diganti pendekatannya,
- atau dihentikan sama sekali.

Keputusan seperti ini perlu didasarkan pada:

- manfaat organisasi,
- kualitas teknis,
- fairness,
- reliability,
- risiko yang tersisa,
- kesiapan kontrol,
- dan kesesuaian dengan tujuan serta kebijakan organisasi.

Bab ini menekankan bahwa kemampuan mengambil keputusan seperti ini adalah tanda kedewasaan tata kelola AI.

Secara keseluruhan, Bab 7 menunjukkan bahwa menjaga AI tetap berguna dan terkendali adalah pekerjaan yang berkelanjutan. Organisasi perlu membangun kebiasaan untuk:

- menilai kualitas dan manfaat;
- memantau perubahan dan risiko;
- merawat komponen solusi;
- mengelola perubahan secara tertib;
- menyiapkan continuity dan fallback;
- serta cukup jujur untuk membatasi, mengganti, atau menghentikan sistem bila memang diperlukan.

7.13 Pertanyaan refleksi

Jawablah pertanyaan-pertanyaan berikut berdasarkan konteks unit kerja, instansi, atau pengalaman Anda sendiri.

1. Mengapa solusi AI tidak cukup hanya diuji sebelum implementasi, tetapi perlu terus dievaluasi setelah dipakai?
2. Dalam konteks pekerjaan Anda, mana yang lebih sulit dinilai: kualitas teknis AI atau manfaat organisasionalnya? Mengapa?
3. Bagaimana Anda membedakan sistem AI yang “terlihat pintar” dari sistem AI yang benar-benar berguna bagi organisasi?
4. Dari tiga aspek berikut—fairness, explainability, reliability—mana yang menurut Anda paling menantang dijaga di instansi Anda saat ini?
5. Apa contoh drift yang paling mungkin terjadi pada use case AI di unit kerja Anda?
6. Dalam penggunaan generative AI, mengapa prompt perlu diperlakukan sebagai bagian dari sistem kerja, bukan hanya sebagai kebiasaan individual pengguna?
7. Mengapa agentic AI membutuhkan pengawasan yang lebih kuat dibanding generative AI biasa?
8. Apakah organisasi Anda saat ini sudah memiliki jalur fallback yang jelas bila AI tidak tersedia atau hasilnya diragukan?
9. Apa risiko terbesar bila organisasi mempertahankan sistem AI hanya karena sudah terlanjur dibangun, meskipun manfaatnya tidak lagi jelas?
10. Menurut Anda, kapan keputusan untuk membatasi, mengganti, atau menghentikan sistem AI menjadi keputusan yang paling bertanggung jawab?

7.14 Latihan dan tugas kecil

Latihan 1 — Membedakan evaluasi teknis dan manfaat organisasi

Pilih satu use case AI yang menurut Anda relevan untuk organisasi Anda. Lalu buat dua kolom:

Kolom A: Evaluasi teknis

- kualitas keluaran;
- relevansi;
- stabilitas;
- waktu respons;
- pola kesalahan.

Kolom B: Evaluasi manfaat organisasi

- penghematan waktu;
- pengurangan beban kerja;
- peningkatan akses informasi;
- peningkatan kualitas proses;
- manfaat bagi pengguna atau pimpinan.

Setelah itu, jawab:

1. indikator teknis apa yang paling penting untuk use case ini;
2. manfaat organisasi apa yang paling realistis diharapkan;
3. apakah keduanya sudah cukup selaras.

Tujuan latihan ini adalah membantu Anda melihat bahwa solusi AI perlu dinilai dari dua lapisan yang berbeda tetapi saling melengkapi.

Latihan 2 — Menilai fairness, explainability, dan reliability

Pilih satu solusi AI yang berbentuk:

- klasifikasi,
- retrieval,
- generative AI,
- atau agentic AI.

Lalu jawab:

1. di mana risiko fairness dapat muncul;
2. apa yang perlu dijelaskan kepada pengguna agar hasil AI dapat dipakai dengan benar;
3. apa tanda bahwa sistem sudah mulai kurang reliabel;
4. tindakan apa yang perlu dilakukan bila salah satu dari tiga aspek tersebut melemah.

Tujuan latihan ini adalah membantu Anda menghubungkan kualitas AI dengan keadilan, keterjelasan, dan keandalan dalam konteks kerja nyata.

Latihan 3 — Menyusun rancangan monitoring sederhana

Ambil satu use case AI, lalu tentukan apa yang akan dipantau untuk empat hal berikut:

1. drift;
2. latency;
3. misuse;
4. incident.

Tuliskan:

- indikator atau tanda yang akan diperhatikan;
- siapa yang memantau;
- seberapa sering dipantau;
- dan tindakan awal yang diambil bila masalah muncul.

Tujuan latihan ini adalah membantu Anda membangun pola pikir bahwa monitoring AI harus lebih luas daripada sekadar memeriksa apakah aplikasi aktif.

Latihan 4 — Meninjau prompt atau agent workflow

Pilih salah satu:

- satu prompt yang sering dipakai untuk generative AI; atau
- satu alur kerja agentic AI yang dibayangkan untuk organisasi Anda.

Lalu evaluasi:

1. apakah tujuannya jelas;
2. apakah batasnya cukup tegas;
3. apakah format hasil sesuai kebutuhan;
4. apakah sumber atau tool yang dipakai sudah tepat;
5. kapan manusia harus masuk;
6. bagian mana yang paling perlu diperbaiki.

Tujuan latihan ini adalah membantu Anda melihat bahwa prompt dan workflow agent adalah komponen yang perlu dirancang, dievaluasi, dan dirawat secara sadar.

Latihan 5 — Menentukan continuity plan dan fallback

Pilih satu solusi AI yang menurut Anda penting bagi organisasi. Lalu buat rancangan singkat:

1. apa yang terjadi bila sistem tidak tersedia;
2. apa yang terjadi bila kualitas hasil diragukan;
3. siapa yang mengambil alih;
4. proses manual atau alternatif apa yang dipakai;
5. kapan fallback harus dipicu;
6. apa yang perlu dicatat setelah fallback digunakan.

Tujuan latihan ini adalah membantu Anda menempatkan AI dalam kerangka ketahanan organisasi, bukan ketergantungan yang rapuh.

Tugas Kecil

Judul: Menyusun Rencana Evaluasi, Monitoring, dan Pemeliharaan Solusi AI

Tuliskan esai singkat sepanjang **2–3 halaman** yang memuat unsur-unsur berikut:

1. Use case AI yang dipilih

Jelaskan satu solusi AI yang paling relevan untuk unit kerja atau instansi Anda.

2. Evaluasi teknis dan manfaat organisasi

Uraikan bagaimana Anda akan menilai:

- o kualitas teknis sistem;
- o dan manfaat nyata sistem bagi organisasi.

3. Fairness, explainability, dan reliability

Jelaskan:

- o risiko fairness yang mungkin muncul;
- o apa yang perlu dijelaskan kepada pengguna;
- o dan bagaimana Anda menilai keandalan sistem.

4. Rencana monitoring

Uraikan bagaimana Anda akan memantau:

- o drift,
- o latency,
- o misuse,
- o dan incident.

5. Evaluasi generative AI atau agentic AI

Bila use case menggunakan generative AI, jelaskan bagaimana prompt dan kualitas keluaran akan dievaluasi.

Bila use case menggunakan agentic AI, jelaskan bagaimana alur kerja agent akan diawasi dan dievaluasi.

6. Continuous validation dan re-evaluation

Jelaskan kapan sistem perlu divalidasi ulang dan kondisi apa yang memicu evaluasi ulang yang lebih menyeluruh.

7. Pemeliharaan komponen solusi

Jelaskan komponen apa saja yang perlu dirawat, seperti:

- o data,
- o model,
- o pipeline,
- o aplikasi,
- o prompt,
- o atau agent workflow.

8. Update, patch, retraining, dan dokumentasi perubahan

Jelaskan:

- o perubahan seperti apa yang mungkin diperlukan;
- o bagaimana perubahan itu diuji;
- o dan bagaimana dokumentasinya dijaga.

9. Business continuity dan fallback

Jelaskan apa yang akan dilakukan organisasi bila AI gagal, tidak tersedia, atau hasilnya tidak cukup layak dipakai.

10. **Keputusan tindak lanjut**

Jelaskan kriteria Anda untuk memutuskan kapan sistem:

- tetap dipertahankan,
- dibatasi,
- diganti,
- atau dihentikan.

Tujuan tugas kecil

Tugas ini bertujuan membantu Anda:

- melihat bahwa implementasi AI tidak selesai pada saat sistem diluncurkan;
- memahami bahwa AI perlu dijaga nilainya, dipantau risikonya, dan dirawat komponennya;
- serta membangun cara berpikir bahwa AI yang matang adalah AI yang terus dievaluasi dan cukup disiplin untuk dibatasi, diubah, atau dihentikan bila diperlukan.