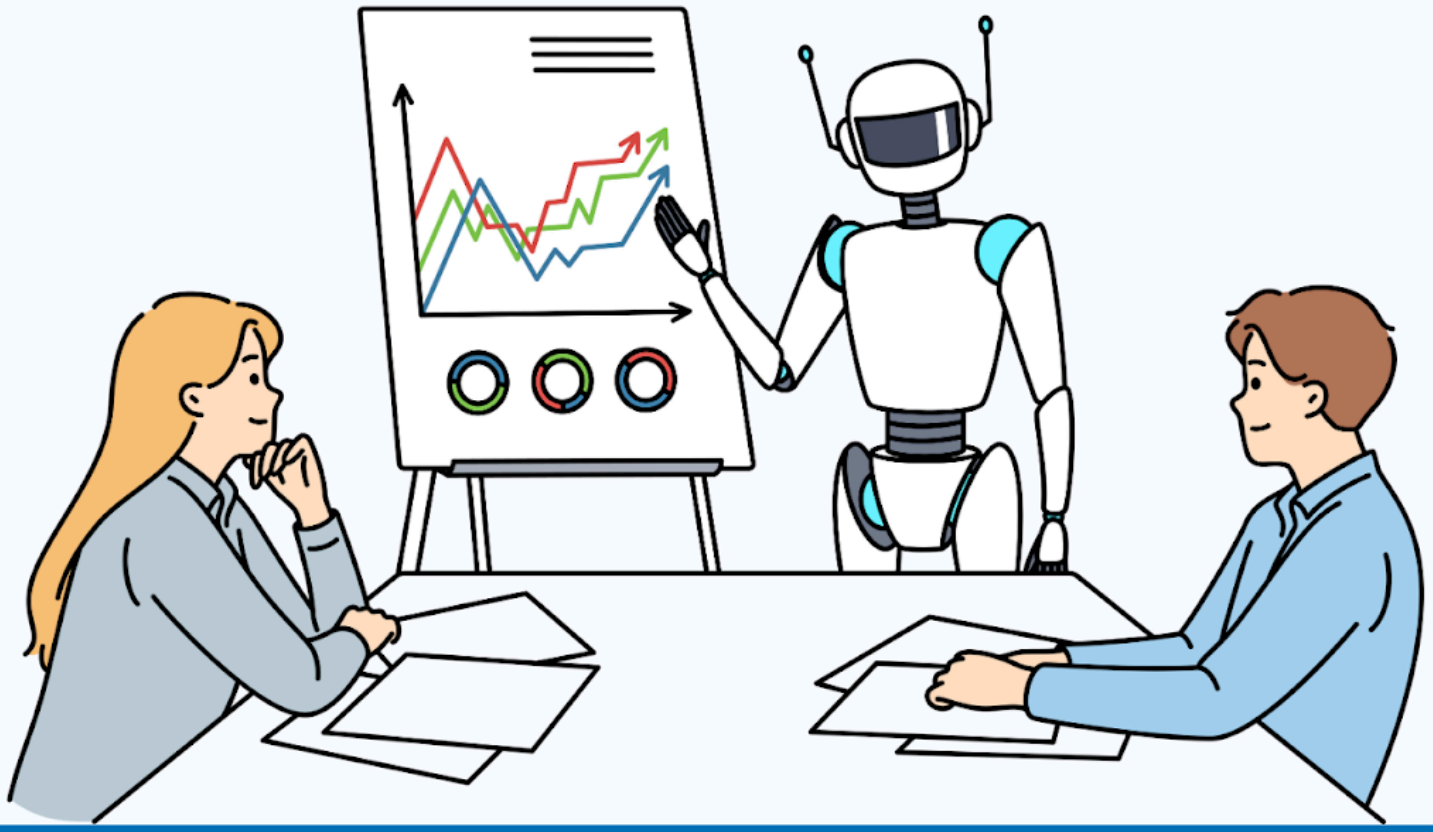
ARTIFICIAL INTELLIGENCE

AI Bukan Cuma Coding: Kerja Tim, Etika, dan Ide Brilian!



UNTUK SMP KELAS IX SEMESTER 2

Anas Nasrulloh
Onno W. Purbo

Buku AI SMP Kelas 9 Semester 2

AI Bukan Cuma Coding:
Kerja Tim, Etika, dan Ide Brilian!

Jumlah Pertemuan: 14

Anas Nasrulloh, M.Kom.
Onno W. Purbo

Institut Teknologi Tangerang Selatan (ITTS)
2025

Daftar Isi

Daftar Isi **3**
Lisensi & Catatan Karya **6**
Disclaimer **6**
Kata Pengantar **7**
BAB 1: Penguatan App Inventor & AI Model 8
Tujuan Pembelajaran 8
Peta Konsep 8
Apersepsi 8
Penjelasan Konsep (Teori) 9
Mengenal Debugging: “Do It” untuk Uji Coba Cepat 9
Peringatan dan Catatan Kesalahan: Show Warning & Logcat 9
Strategi Debugging: Pecah Masalah, Uji Fitur, dan Gunakan Notifier 10
Penanganan Kesalahan: Error Handling Sederhana 10
Fitur Tambahan: TinyDB, List, dan AI Sederhana 11
AI Sederhana – Membuat Aplikasi Jadi “Lebih Pintar” 13
Penutup 13
Contoh Kasus atau Ilustrasi 13
Contoh Soal dan Pembahasan 15
Fakta Menarik / Fun Facts 18
Tips Belajar atau Tips Cepat Menghafal 18
Aktivitas Siswa 20
Rangkuman 22
Latihan Soal 22
Tugas Proyek 25
BAB 2: Proyek Kolaboratif: Tim Membangun Aplikasi 28
Tujuan Pembelajaran 28
Peta Konsep 28
Apersepsi 28
Penjelasan Konsep (Teori) 29
Kerja Tim: Kunci Sukses dalam Membangun Aplikasi AI 29
Mengapa Harus Bekerja dalam Tim? 29
Siapa Saja yang Ada dalam Tim Proyek Aplikasi? 30
Kunci Komunikasi Efektif dalam Tim 30
Cara Kerja Efisien di App Inventor (Teknik Modular dan Pengelolaan File) 31
Kesimpulan: Bersama Kita Bisa! 31
Contoh Kasus atau Ilustrasi 31
Contoh Soal dan Pembahasan 34
Fakta Menarik / Fun Facts 36
Tips Belajar atau Tips Cepat Menghafal 37
Aktivitas Siswa 38
Rangkuman 40
Latihan Soal 40

Tugas Proyek..... 43
BAB 3: Tanya Jawab AI dengan ChatGPT & Gemini.....47
Tujuan Pembelajaran..... 47
Peta Konsep.....47
Apersepsi..... 47
Penjelasan Konsep (Teori)..... 48
Pengertian AI Generatif.....48
AI Generatif untuk Belajar: Teman Pintar yang Siap Bantu!.....48
Cara Bertanya yang Efektif ke AI (Dasar Prompt Engineering)..... 49
Contoh Kasus atau Ilustrasi..... 50
Contoh Soal dan Pembahasan..... 53
Fakta Menarik / Fun Facts.....55
Tips Belajar atau Tips Cepat Menghafal..... 55
Aktivitas Siswa..... 57
Rangkuman.....59
Latihan Soal..... 60
Tugas Proyek..... 63
BAB 4: Diskusi Etika Penggunaan AI di Masyarakat..... 66
Tujuan Pembelajaran..... 66
Peta Konsep.....66
Apersepsi..... 66
Penjelasan Konsep (Teori)..... 67
Apa Itu Etika AI?.....67
Isu-Isu Etika dalam Penggunaan AI.....67
Contoh Kasus atau Ilustrasi..... 69
Contoh Soal dan Pembahasan..... 71
Fakta Menarik / Fun Facts..... 74
Tips Belajar atau Tips Cepat Menghafal..... 75
Aktivitas Siswa..... 76
Rangkuman.....79
Latihan Soal..... 79
Tugas Proyek..... 83
BAB 5: Dokumentasi Proyek dengan ChatGPT.....86
Tujuan Pembelajaran..... 86
Peta Konsep.....86
Apersepsi..... 86
Penjelasan Konsep (Teori)..... 87
Mengapa Dokumentasi itu Penting?..... 87
Jenis-Jenis Dokumentasi Proyek Aplikasi.....88
Peran AI (seperti ChatGPT) dalam Membantu Dokumentasi..... 88
Strategi Kerja Sama Manusia dan AI (Human-in-the-Loop).....88
Contoh Kasus atau Ilustrasi..... 89
Contoh Soal dan Pembahasan..... 91
Fakta Menarik / Fun Facts.....94

Tips Belajar atau Tips Cepat Menghafal..... 94
Aktivitas Siswa..... 95
Rangkuman..... 98
Latihan Soal..... 98
Tugas Proyek..... 102
BAB 6: Presentasi Proyek Akhir..... 105
Tujuan Pembelajaran..... 105
Peta Konsep..... 105
Apersepsi..... 105
Penjelasan Konsep (Teori)..... 106
Tujuan Presentasi Proyek Akhir..... 106
Struktur Presentasi yang Efektif..... 107
Tips untuk Presentasi yang Menarik..... 109
Contoh Kasus atau Ilustrasi..... 109
Contoh Soal dan Pembahasan..... 112
Fakta Menarik / Fun Facts..... 114
Tips Belajar atau Tips Cepat Menghafal..... 116
Aktivitas Siswa..... 117
Rangkuman..... 120
Latihan Soal..... 120
Tugas Proyek..... 123
BAB: Mengenal Domain .id..... 127
Apa Itu Domain?..... 128
Apa Itu .id?..... 128
Keunggulan domain .id..... 128
Siapa Itu PANDI?..... 129
Kenapa Kamu Harus Tahu Ini?..... 129
Lebih dalam tentang PANDI..... 129
Tata Kelola Internet..... 131
Institut Teknologi Tangerang Selatan (ITTS) dan Penulis..... 132
Tentang Penulis..... 133

Lisensi & Catatan Karya

Karya ini dilisensikan di bawah lisensi **Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)**.
Artinya, **siapapun bebas untuk menggunakan, menyalin, membagikan, dan mengadaptasi** materi ini, **dengan syarat**:

- **Memberikan atribusi yang sesuai** (menyebut sumber asli),
- **Menyertakan lisensi yang sama** jika dimodifikasi atau dikembangkan lebih lanjut.

Lihat detail lisensi di sini https://creativecommons.org/licenses/by-nd/4.0/legalcode.id

Desain Cover: Irwan Siswanto

Disclaimer

Materi pembelajaran ini dibuat dengan **dana swadaya masyarakat Indonesia** dan **kontribusi sukarela dari para dosen di Institut Teknologi Tangerang Selatan(ITTS)**. Karya ini didistribusikan secara bebas untuk mendukung pendidikan digital dan kecerdasan buatan di kalangan pelajar Indonesia.

Kami **memohon maaf** jika terdapat kekurangan dalam isi maupun penyajian materi ini. Kritik dan saran sangat kami harapkan untuk terus menyempurnakan karya serupa di masa depan.

Mohon doa agar amal ibadah mereka yang terlibat di kegiatan ini diterima dan dilipatgandakan.

Kata Pengantar

Buku ini disusun sebagai bentuk partisipasi sederhana dari sivitas akademika Institut Teknologi Tangerang Selatan (ITTS) dalam mendukung kemajuan pendidikan di Indonesia. Di tengah pesatnya perkembangan teknologi, kami terdorong untuk menghadirkan materi pembelajaran yang relevan dan dapat diakses secara luas oleh masyarakat, khususnya generasi muda.

Dengan semangat berbagi dan memberdayakan, kami berharap buku ini dapat menjadi langkah awal yang bermakna dalam memperkenalkan konsep dasar kecerdasan buatan (AI) kepada pelajar tingkat SMP dan SMA. Materi disusun secara sistematis dan aplikatif, mencakup antara lain:

- Dasar-dasar *Computational Thinking*
- Pembuatan aplikasi Android menggunakan App Inventor 2
- Pemrograman visual dengan Scratch
- Pengenalan AI melalui Google Teachable Machine
- Eksplorasi AI generatif seperti ChatGPT, Gemini, dan Grok
- Pembelajaran AI berbasis data dengan Orange Data Mining, disertai contoh yang sederhana

Penyusunan dan penerbitan buku ini sepenuhnya dibiayai secara mandiri berkat dukungan berbagai pihak yang memiliki kepedulian terhadap kemajuan ilmu pengetahuan di tanah air. Untuk itu, kami mengucapkan terima kasih yang tulus atas segala bantuan dan dorongan yang telah diberikan.

Kami menyadari bahwa karya ini masih jauh dari sempurna. Karena itu, segala bentuk kritik, saran, maupun masukan yang membangun sangat kami hargai dan nantikan. Silakan sampaikan melalui kontak resmi ITTS atau langsung kepada tim penulis. Pembaruan dan perbaikan akan terus kami lakukan demi meningkatkan kualitas isi dan manfaat buku ini.

Semoga buku ini dapat menjadi pijakan awal bagi para pelajar dalam memahami dunia kecerdasan buatan, serta turut menginspirasi langkah kecil menuju masa depan Indonesia yang lebih cerah.

Dengan rendah hati, kami memohon doa dan dukungan dari para pembaca agar upaya sederhana ini dapat membawa manfaat yang luas dan menjadi amal jariyah bagi semua penulis yang terlibat.

Tangerang Selatan, Agustus 2025
Penulis

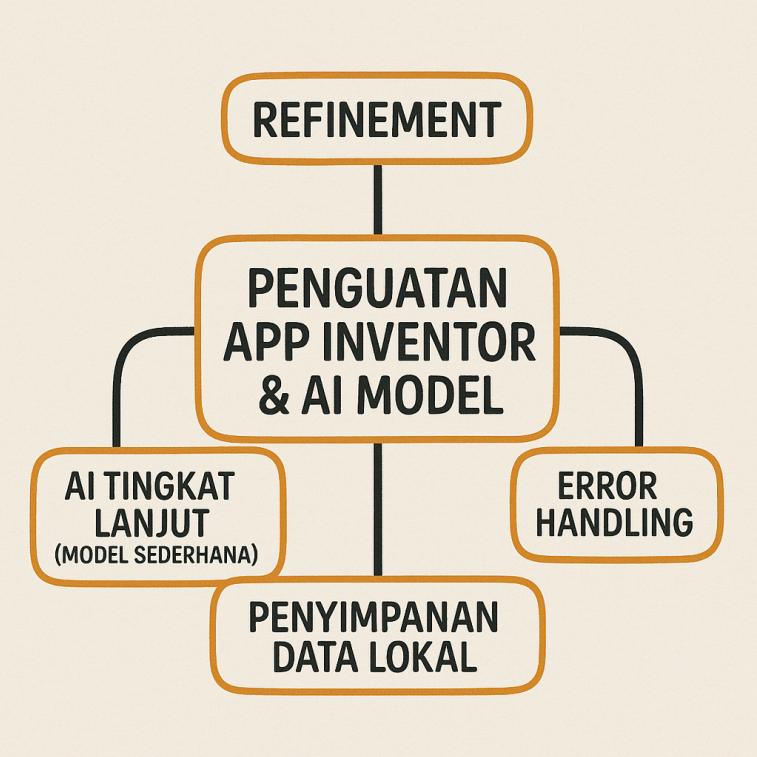
BAB 1: Penguatan App Inventor & AI Model

Tujuan Pembelajaran

Siswa mampu:

- Siswa dapat memperdalam keterampilan membuat aplikasi menggunakan **App Inventor**.
- Siswa mampu menguji dan memperbaiki kesalahan pada aplikasi secara sistematis (**debugging**).
- Siswa dapat menggunakan fitur lanjutan seperti **Do It, Notifier, TinyDB**, dan **List**.
- Siswa mulai mengenal cara menambahkan fitur **AI sederhana** ke dalam aplikasi.

Peta Konsep



Apersepsi

Halo, para inovator muda!
Kalian pasti pernah merasa bangga saat aplikasi buatan sendiri akhirnya bisa berjalan, bukan? Tapi, coba bayangkan dulu—apa yang akan terjadi kalau tombolnya ditekan dua kali? Bagaimana kalau tiba-tiba koneksi internet terputus? Atau, bisakah orang lain langsung mengerti cara menggunakan aplikasi tanpa harus bertanya? Nah, inilah tantangan sebenarnya dalam membuat aplikasi yang tidak hanya sekedar bisa dijalankan, tapi juga kuat, mudah digunakan, dan tahan terhadap kesalahan.

Di bab ini, kalian akan berperan sebagai **detektif aplikasi**! Kita akan belajar bagaimana cara **mendeteksi dan memperbaiki error**, menyusun fitur yang lebih stabil, **menyimpan data** agar tidak hilang saat aplikasi ditutup, dan yang tak kalah seru—**menambahkan sentuhan kecerdasan buatan (AI)** agar aplikasi jadi lebih pintar dan interaktif.

Petualangan coding yang seru dan menantang ini siap dimulai. Yuk, buktikan bahwa kamu adalah pengembang muda yang kreatif dan tangguh!

Penjelasan Konsep (Teori)

Saat membuat aplikasi, kita tidak selalu langsung berhasil. Kadang, ada bagian yang tidak bekerja sesuai harapan. Nah, disinilah kita perlu melakukan proses **debugging**, yaitu mencari dan memperbaiki kesalahan (*error*) dalam program agar aplikasi bisa berjalan dengan baik. Seperti detektif yang mencari tahu siapa pelakunya, debugging mengajak kita untuk menyelidiki apa yang salah dan memperbaikinya.

Mengenal Debugging: “Do It” untuk Uji Coba Cepat

Do It adalah fitur keren di App Inventor yang memungkinkan kita menjalankan *sebagian kecil* dari program secara langsung, tanpa harus menjalankan seluruh aplikasi.

Cara Kerja:

- Klik kanan pada blok kode tertentu.
- Pilih “Do It” untuk melihat hasil dari blok tersebut secara instan.

Contoh:
Ingin tahu apakah hasil penjumlahan di blokmu benar? Gunakan “Do It” untuk mengeceknya langsung!

Manfaat:

- Menghemat waktu saat uji coba.
- Memudahkan kita menemukan letak kesalahan.
- Cocok untuk eksperimen saat aplikasi masih dalam tahap pengembangan.

Peringatan dan Catatan Kesalahan: Show Warning & Logcat

App Inventor punya sistem yang cerdas untuk memberitahu jika ada yang salah dalam programmu.

Show Warning

- Jika blokmu belum lengkap atau salah sambung, akan muncul **tanda kuning (⚠️)**.

- Ini adalah tanda peringatan awal agar kamu bisa segera memperbaikinya.

Logcat (di Android Studio)

- Merupakan catatan lengkap semua kejadian yang terjadi di aplikasi, termasuk kesalahan tersembunyi.
- Cocok digunakan jika kamu mulai menjelajah dunia coding yang lebih kompleks.

Manfaat:

- Membantu mengidentifikasi jenis dan lokasi error.
- Memperkuat pemahaman tentang bagaimana aplikasi bekerja “di balik layar”.

Strategi Debugging: Pecah Masalah, Uji Fitur, dan Gunakan Notifier

Debugging yang baik perlu strategi, seperti saat memperbaiki sepeda: kita periksa satu bagian dulu, bukan langsung bongkar semua!

Langkah-Langkah Strategis:

1. **Pecah Masalah Menjadi Bagian Kecil**
   - Periksa satu fitur dulu, misalnya hanya tombol “Simpan”.
2. **Uji Satu Fitur dalam Satu Waktu**
   - Jangan langsung semua fitur dijalankan. Fokus pada satu saja.
3. **Gunakan Notifier**
   - Tampilkan nilai suatu variabel dengan kotak pesan agar mudah dicek.



Manfaat:

- Mempercepat proses perbaikan.
- Menghindari kebingungan saat debugging.
- Bisa digunakan untuk aplikasi besar maupun kecil.

Penanganan Kesalahan: Error Handling Sederhana

Apa yang terjadi kalau aplikasi tiba-tiba error? Kita butuh sistem yang *ramah pengguna* untuk menangani hal ini.

Konsep Try-Catch (Gaya App Inventor)

- Walaupun tidak ada blok khusus *try-catch*, kita bisa menerapkannya secara logika.
- Misalnya: saat data cuaca gagal dimuat, tampilkan pesan seperti:

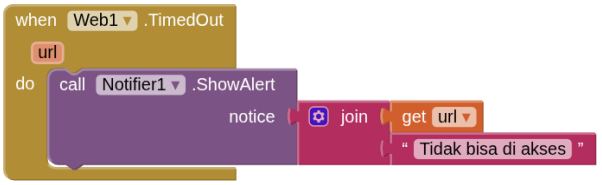
  “Tidak bisa mengambil data cuaca. Cek koneksi internet kamu ya!”

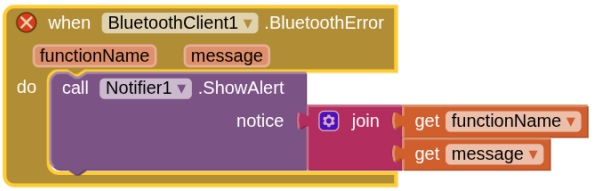
Blok When .Error

- Beberapa komponen seperti Web atau Bluetooth punya blok error bawaan.

  Contoh:

When Web1.TimeOut
→ Notifier1.ShowAlert("URL tidak ditemukan atau server sedang offline.")





Pesan Error yang Baik

- Hindari menulis hanya "ERROR 404".
- Gunakan pesan jelas dan bersahabat:

  “Maaf, data yang kamu cari tidak ditemukan. Silakan coba beberapa saat lagi.”

Manfaat:

- Pengguna merasa lebih tenang saat error terjadi.
- Meningkatkan kepercayaan terhadap aplikasi.
- Membantu pengguna mengerti apa yang sedang terjadi.

Fitur Tambahan: TinyDB, List, dan AI Sederhana

Mari kita jelajahi fitur-fitur lanjutan di App Inventor yang bikin aplikasi jadi makin canggih!

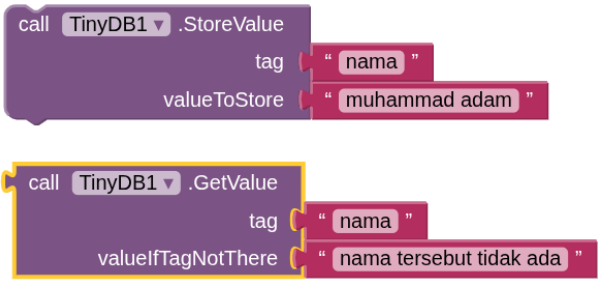
TinyDB – Menyimpan Data Secara Permanen

- Seperti buku catatan digital.

- Data tetap tersimpan meskipun aplikasi ditutup.

Blok penting:

- `StoreValue`: untuk menyimpan data.
- `GetValue`: untuk mengambil data yang disimpan.



Contoh:

- Menyimpan skor tertinggi game.
- Menyimpan nama pengguna.

Manfaat:

- Pengguna tidak perlu mengisi ulang data.
- Aplikasi jadi lebih praktis dan nyaman digunakan.

List dan ListPicker – Daftar Data yang Rapi

- List menyimpan banyak data dalam satu variabel.
- Bisa ditambah, dihapus, dan diakses berdasarkan urutan.

Blok penting:

- `make a list`, `add items to list`, `select list item`.



Contoh:

- Daftar kota.
- Menu makanan.

Manfaat:

- Menyederhanakan tampilan aplikasi.
- Mempermudah interaksi pengguna.

AI Sederhana – Membuat Aplikasi Jadi “Lebih Pintar”

- AI bisa mengenali gambar, suara, atau tulisan.
- Bisa dilatih menggunakan alat seperti **Teachable Machine**.

Contoh Penggunaan:

- Mengenali wajah senyum.
- Membaca tulisan dari gambar.

Manfaat:

- Menjadikan aplikasi lebih interaktif.
- Melatih cara berpikir logis dan sistematis saat membuat program.

Penutup

Debugging bukan sekadar memperbaiki error, tapi juga tentang **melatih cara berpikir logis**, **mencari solusi**, dan **membuat aplikasi yang lebih baik**. Dengan mengenal fitur-fitur lanjutan di App Inventor, kamu bukan hanya belajar coding, tapi juga membangun aplikasi cerdas yang siap digunakan oleh banyak orang.

Yuk terus eksplorasi! Jadilah pengembang muda yang tangguh dan kreatif! 💡📱

Contoh Kasus atau Ilustrasi

Kasus 1: Kalkulator Ngambek!

Ilustrasi Kasus:
Bayangkan kamu membuat aplikasi kalkulator sederhana di ponsel. Tapi... begitu kamu tekan "Hitung" setelah mengetik simbol aneh atau huruf, aplikasi langsung *crash*! Waduh, aplikasi kamu jadi ngambek, tuh!

Apa yang Terjadi?
Komputer tidak paham simbol yang bukan angka. Jadi kalau kamu ketik “2 + a”, sistem bingung dan menyerah.

Solusi Menantang:
Tambahkan pemeriksaan: apakah input yang diketik itu benar-benar angka? Kalau bukan, memunculkan pesan seperti, *"Masukkan hanya angka, yaa!"* supaya pengguna tidak bingung. Gunakan komponen **Notifier** agar peringatannya muncul seperti pop-up.

Kasus 2: Catatan Hilang Misterius

Ilustrasi Kasus:
Kamu sudah menulis banyak catatan penting di aplikasi buatanmu. Tapi saat besok dibuka lagi... *loh kok kosong semua?* 😱

Apa yang Terjadi?
Aplikasi belum menyimpan datanya ke tempat yang bisa dibaca ulang saat dibuka kembali, seperti ke dalam TinyDB.

Solusi Menantang:
Gunakan **TinyDB.StoreValue** saat pengguna menekan tombol “Simpan”. Lalu, saat aplikasi dibuka ulang, gunakan **TinyDB.GetValue** untuk menampilkan catatan-catatan sebelumnya. Gunakan **ListView** agar tampilannya rapi.

Kasus 3: Data Tidak Tersimpan Padahal Sudah Tekan Simpan

Ilustrasi Kasus:
seorang teman menggunakan aplikasi buatanmu untuk mencatat skor game. Tapi tiba-tiba, nilainya tidak tersimpan. Temanmu bilang, “Aku sudah tekan ‘Simpan’ lho!”

Apa yang Terjadi?
Mungkin penyimpanan gagal karena error, tapi aplikasi tidak memberi tahu apa-apa.

Solusi Menantang:
Tambahkan logika untuk memeriksa apakah penyimpanan berhasil. Kalau gagal, memunculkan pesan seperti: *“Oops! Gagal menyimpan. Pastikan memori cukup, ya!”* supaya pengguna tahu apa yang salah dan tidak panik.

Kasus 4: Tombol Tidak Bekerja!

Ilustrasi Kasus:
Kamu membuat tombol “Hitung” di aplikasi kalkulator. Tapi, setiap kali diklik, tidak ada yang terjadi! Padahal kamu sudah yakin codingnya benar...

Apa yang Terjadi?
Kemungkinan blok "When Button.Click" belum diisi dengan benar, atau salah satu blok di dalamnya tidak bekerja seperti yang diharapkan.

Solusi Menantang:
Gunakan fitur **“Do It”** pada blok-blok tertentu untuk mengecek apakah nilainya muncul seperti yang diharapkan. Ini seperti memperbesar kaca pembesar dan menyelidiki setiap bagian kode!

Kasus 5: Aplikasi Pelupa

Ilustrasi Kasus:
Kamu membuat aplikasi “Daftar Belanja”. Saat pengguna mengetik barang-barang dan menekan “Simpan”, data berhasil muncul. Tapi begitu aplikasi ditutup dan dibuka lagi, datanya hilang seperti lenyap ditelan bumi!

Apa yang Terjadi?
Aplikasi tidak menyimpan data ke tempat permanen.

Solusi Menantang:
Gunakan list (daftar) untuk menyimpan semua item, lalu simpan ke **TinyDB**. Saat aplikasi dibuka lagi, ambil kembali data dari TinyDB dan tampilkan dalam **ListView**. Sekarang aplikasimu jadi pintar dan tidak pelupa lagi!

Kasus 6: Error Mendadak Saat Input Kosong

Ilustrasi Kasus:
Kamu punya aplikasi penjumlahan. Tapi saat pengguna tidak mengisi angka sama sekali lalu menekan “Hitung”... aplikasi langsung error. Waduh!

Apa yang Terjadi?
Aplikasi tidak tahu cara menangani input kosong.

Solusi Menantang:
Tambahkan pengecekan: kalau input kosong, memunculkan pesan seperti, *“Yuk isi angka dulu sebelum menghitung!”* Ini membuat aplikasi lebih ramah dan membantu pengguna menghindari kesalahan.

Semua kasus ini bisa dijadikan bagian dari proses belajar siswa untuk **mengenal dunia debugging, error handling, dan penyimpanan data**. Penjelasannya bersifat eksploratif dan menyenangkan agar siswa termotivasi untuk memperbaiki dan mengembangkan aplikasi mereka sendiri. Jika kamu ingin versi visual (komik, flowchart, atau storyboard dari kasus-kasus ini), tinggal bilang saja!

Contoh Soal dan Pembahasan

Soal 1: Menyimpan Riwayat Perhitungan

Pertanyaan:
Bayangkan kamu membuat aplikasi kalkulator. Ketika aplikasi ditutup lalu dibuka lagi, riwayat hitungan terakhir (misalnya: 5 + 3 = 8) tidak muncul. Hmm… kenapa ya bisa begitu? Dan bagaimana cara agar riwayat itu tetap tersimpan walaupun aplikasinya ditutup?

Pembahasan:
Agar data tetap ada walaupun aplikasi ditutup, kita bisa menyimpannya ke dalam **TinyDB** – seperti lemari penyimpanan khusus di dalam aplikasi!

Langkah-langkah:

Setelah pengguna menekan tombol "=":

```
call TinyDB.StoreValue
  tag = "history"
  value = "5 + 3 = 8"
```

Saat aplikasi dibuka (Screen1.Initialize), tampilkan riwayat:

```
set LabelRiwayat.Text to call TinyDB.GetValue
  tag = "history"
  valueIfTagNotThere = "Belum ada riwayat"
```

Ini membuat aplikasi kamu terasa lebih pintar dan ramah pengguna!

Soal 2: Aplikasi Musik Error? Cegah Sebelum Terjadi!

Pertanyaan:
Kamu membuat aplikasi pemutar musik. Tapi saat file musik tidak ada, aplikasi langsung berhenti/crash. Coba tebak, bagaimana cara agar aplikasi tidak error kalau file musiknya tidak ditemukan?

Pembahasan:
Inilah pentingnya **error handling** atau penanganan kesalahan. Kita harus memeriksa dulu apakah file musik tersedia sebelum diputar.

Solusi logikanya:

```
if (fileName ≠ "")
  call Player.Source to fileName
  call Player.Start
else
  call Notifier.ShowAlert("File tidak ditemukan!")
```

Dengan begitu, pengguna tetap mendapat informasi yang sopan, bukan aplikasi yang tiba-tiba mati.

Soal 3: Tombol Absen Tak Berfungsi? Yuk Cari Masalahnya!

Pertanyaan:
Di aplikasi absensi yang kamu buat, pengguna sudah menekan tombol “Absen Masuk” tapi datanya tidak tercatat. Waduh! Apa saja langkah-langkah yang bisa kamu lakukan untuk mencari dan memperbaiki masalah ini?

Pembahasan:
Ini adalah bagian dari proses **debugging**, yaitu mencari sumber masalah.

Langkahnya bisa seperti ini:

1. Gunakan **blok Do It** untuk mengecek apakah data seperti nama, tanggal, atau jam benar-benar ada.

   Tambahkan pengecekan input:

   ```
   if (TextBoxNama.Text = "")
     call Notifier.ShowAlert("Nama harus diisi dulu!")
   else
     call TinyDB.StoreValue
   ```

2. Pastikan tombol benar-benar terhubung ke logika penyimpanan.

Dengan langkah-langkah ini, kamu bisa menemukan dan memperbaiki bug dengan percaya diri!

Soal 4: Kirim Formulir Tapi Kosong? Jangan Sampai Terjadi!

Pertanyaan:
Aplikasi formulir pendaftaran kamu sudah keren. Tapi, kadang pengguna langsung menekan tombol “Kirim” tanpa mengisi nama dan email. Gimana ya supaya aplikasi tidak membiarkan data kosong terkirim?

Pembahasan:
Solusinya adalah **validasi input**, yaitu memeriksa dulu apakah kolom sudah diisi.

Blok logikanya:

```
if (TextboxNama.Text ≠ "") and (TextboxEmail.Text ≠ "")
  call Notifier.ShowAlert("Data berhasil dikirim!")
else
  call Notifier.ShowAlert("Harap isi semua kolom sebelum mengirim.")
```

Dengan logika ini, pengguna akan mendapat pesan ramah dan jelas jika lupa mengisi data.

Soal 5: Cegah Lagu Kosong di Playlist!

Pertanyaan:
Di aplikasi pemutar lagu buatanmu, terkadang playlist kosong tapi pengguna tetap menekan tombol “Putar Lagu”. Apa yang sebaiknya dilakukan supaya aplikasi tidak bingung atau error saat itu terjadi?

Pembahasan:
Sebelum memutar lagu, cek dulu apakah playlist-nya **tidak kosong**. Gunakan logika **if** agar hanya memutar saat lagu tersedia.

Contoh logikanya:

```
if (ListLagu.Length > 0)
  set Player.Source to ListLagu.Select(1)
  call Player.Start
else
  call Notifier.ShowAlert("Tidak ada lagu di daftar!")
```

Dengan cara ini, aplikasi akan lebih pintar dan tidak error saat pengguna mencoba hal-hal yang belum tersedia.

Fakta Menarik / Fun Facts

- **Ngoding = Jadi Detektif?**
  Programmer itu lebih sering jadi detektif daripada penulis kode! 50%–80% waktunya dihabiskan untuk mencari dan memperbaiki *bug*, bukan mengetik kode.
- **Bug Itu Bukan Cuma Istilah!**
  Istilah “bug” beneran datang dari serangga sungguhan! Tahun 1947, seekor ngengat nyangkut di komputer raksasa dan bikin error. Sejak itu, semua kesalahan dalam kode disebut *bug*.
- **Satu Huruf Bisa Hancurkan Raksasa!**
  Google pernah *crash* gara-gara SATU karakter kode doang. Bayangin, cuma satu huruf salah bisa bikin dunia geger!
- **Coding = Latihan Otak Super!**
  Belajar coding itu kayak nge-gym buat otak. Kamu bakal jago berpikir logis, ngatur strategi, dan mencari solusi dengan cepat dan tepat!
- **Game Pertama Bukan dari Gamer!**
  Game digital pertama diciptakan bukan oleh gamer, tapi ilmuwan nuklir! “Tennis for Two” dibuat di lab nuklir tahun 1958. Canggih banget buat zamannya!

Tips Belajar atau Tips Cepat Menghafal

Versi Singkat dan Mudah Diingat

1. **Iterasi** → *Perbaiki, Simpan, Uji Lagi*
2. **Mulai Kecil** → *Tangani bagian kecil dulu*
3. **Catat** → *Tulis semua langkah yang dicoba*
4. **Pahami** → *Fokus ke alur, bukan hafalan*

5. **Bikin Proyek** → *Belajar sambil bikin aplikasi seru*
6. **Nikmati Prosesnya** → *Belajar harus bikin penasaran*

Cerita Mudah Diingat

1. Iterasi = Tukang Bangun Rumah

> Bayangkan kamu bangun rumah. Jangan tunggu semua selesai baru dicek! Cek tiap bagian yang selesai: dinding, jendela, atap. Kalau genteng bocor, bisa langsung betulin, nggak perlu bongkar semuanya. Itu namanya **"Perbaiki, Simpan, Uji Lagi"**.

2. Mulai Kecil = Benang Kusut

> Kamu pegang benang kusut. Kalau kamu tarik semua sekaligus, malah makin kusut. Tapi kalau kamu pelan-pelan urai satu bagian, lama-lama rapi. Sama seperti debugging, **fokus di bagian kecil dulu**.

3. Catat = Buku Resep Debugging

> Pernah bikin kue dan gagal karena salah takaran? Kalau kamu catat, besok tidak salah lagi. Sama juga dengan debugging. Tulis langkahmu supaya tahu apa yang udah dicoba dan nggak mengulang kesalahan yang sama. Itu bisa jadi **buku resep pribadi**.

4. Pahami = Jalan Cerita, bukan Skrip

> Bayangin kamu nonton film. Nggak harus hafal semua dialog, tapi kamu harus ngerti jalan ceritanya. Sama kayak bikin aplikasi – kamu harus **ngerti alurnya, bukan hafal bloknya**.

5. Bikin Proyek = Game Favoritmu

> Pernah main game yang kamu suka banget? Kamu rela belajar cara main, strategi, bahkan nonton tutorial. Coba bikin aplikasi yang kamu suka juga – **semangatnya akan keluar sendiri**.

Tips Belajar Cepat

Prinsip	Makna Singkat	Analogi
Perbaiki, Simpan, Uji	Belajar bertahap, langsung uji	Tukang bangun rumah
Mulai dari yang kecil	Fokus pada masalah kecil dulu	Benang kusut
Catat langkah	Dokumentasi = Ingat + Belajar + Hindari ulang	Buku resep pribadi

Pahami alur	Mengerti lebih penting dari menghafal	Nonton film, bukan hafal skrip
Proyek seru	Belajar lebih menyenangkan lewat karya	Main game favorit

Aktivitas Siswa

Aktivitas 1: Telusuri dan Perbaiki Bug di Aplikasi Sendiri

Deskripsi Aktivitas:
Pilih satu aplikasi buatanmu di semester lalu. Cermati dengan lebih jeli – apakah ada bagian yang terasa kurang nyaman digunakan atau tidak bekerja sebagaimana mestinya?

Langkah-Langkah Menantang:

- Jalankan aplikasi seolah-olah kamu adalah pengguna baru.
- Catat hal-hal yang terasa membingungkan, lambat, atau tidak berfungsi dengan benar.
- Coba perbaiki minimal 2 hal, misalnya:
    - Tombol bisa ditekan terlalu cepat → Tambahkan jeda atau batasi jumlah klik.
    - Error muncul tanpa penjelasan → Tambahkan notifikasi atau pesan kesalahan.

Tujuannya: Melatih kepekaan mendeteksi dan memperbaiki masalah teknis seperti seorang developer profesional.

Aktivitas 2: Tambahkan Fitur Penyimpanan yang Cerdas

Deskripsi Aktivitas:
Pernahkah kamu merasa kecewa karena data hilang setelah aplikasi ditutup? Kini saatnya kamu belajar menyimpan data dengan fitur **TinyDB**!

Langkah-Langkah Menantang:

- Pilih aplikasi yang cocok ditambah fitur penyimpanan, misalnya:
    - Game → Simpan skor tertinggi
    - Kalkulator → Simpan riwayat hitungan terakhir
    - Aplikasi cuaca → Simpan preferensi tampilan (mode terang/gelap)
- Tambahkan komponen TinyDB dan uji apakah data tetap ada setelah aplikasi ditutup.

Tujuannya: Membuat aplikasi yang lebih pintar dan "ingat" apa yang dilakukan sebelumnya!

Aktivitas 3: Rancang Aplikasi Daftar Belanja Dinamis

Deskripsi Aktivitas:
Bayangkan kamu sedang belanja dan ingin mencatat barang-barang yang ingin dibeli. Ciptakan aplikasi daftar belanja interaktif yang bisa kamu gunakan sendiri!

Langkah-Langkah Menantang:

- Buat input untuk menambahkan nama barang.
- Simpan data ke dalam **List**.
- Gunakan **ListPicker** untuk melihat dan menghapus barang yang sudah dibeli.

Tujuannya: Belajar cara menyimpan, menampilkan, dan menghapus data secara dinamis menggunakan struktur data List.

Aktivitas 4: Ubah Tampilan Aplikasi Jadi Lebih Menarik

Deskripsi Aktivitas:
Aplikasi yang keren bukan hanya pintar, tapi juga menarik dilihat! Yuk, ubah tampilan aplikasi buatanmu agar lebih enak digunakan dan menyenangkan bagi pengguna.

Langkah-Langkah Menantang:

- Tambahkan fitur **pengubah tema** (gelap/terang).
- Gunakan warna dan ikon yang konsisten.
- Sesuaikan ukuran tombol dan teks agar mudah dibaca.

Tujuannya: Meningkatkan **user experience** (pengalaman pengguna) dengan desain yang lebih profesional.

Aktivitas 5: Tambahkan Efek Suara atau Animasi Interaktif

Deskripsi Aktivitas:
Bikin aplikasi jadi lebih hidup! Efek suara dan animasi bisa membuat pengguna lebih terlibat dan senang.

Langkah-Langkah Menantang:

- Pilih satu momen penting di aplikasi (misalnya saat tombol ditekan atau saat level naik di game).
- Tambahkan suara pendek atau efek visual (seperti perubahan warna atau ikon bergerak).
- Uji efek tersebut – terlalu sering atau justru menyenangkan?

Tujuannya: Membuat aplikasi lebih **fun**, interaktif, dan berkesan!

Aktivitas 6: Uji Coba Aplikasi ke Temanmu dan Kumpulkan Masukan

Deskripsi Aktivitas:
Sekarang kamu sudah punya aplikasi keren. Tapi bagaimana menurut orang lain?

Langkah-Langkah Menantang:

- Minta 2–3 temanmu mencoba aplikasi buatanmu.
- Tanyakan bagian mana yang mudah digunakan dan mana yang membingungkan.

- Gunakan feedback itu untuk menyempurnakan aplikasimu.

Tujuannya: Melatih kemampuan berpikir dari sudut pandang pengguna dan memperbaiki aplikasi berdasarkan masukan nyata.

Semua aktivitas di atas bisa kamu pilih sesuai minat dan level tantangan yang kamu inginkan. Ingat, semakin sering kamu mencoba dan memperbaiki, semakin hebat kamu jadi programmer muda!

Rangkuman

Kamu pasti pernah merasa aplikasi buatanmu sudah keren karena bisa dijalankan tanpa error, tapi... tunggu dulu! Apakah benar-benar bebas masalah? Di bab ini, kamu akan belajar bahwa membuat aplikasi bukan hanya tentang "bisa jalan", tapi juga tentang seberapa **andal**, **mudah digunakan**, dan **tahan terhadap kesalahan**. Kamu akan menjelajahi dunia **debugging tingkat lanjut**, yaitu seni mencari dan memperbaiki bug seperti seorang *detektif kode*. Mulai dari fitur praktis seperti **"Do It"**, hingga cara menggunakan **pesan error yang ramah** agar pengguna tidak kebingungan, semua akan dibahas dengan gaya yang menyenangkan dan menantang!

Tak hanya itu, kamu juga akan belajar membuat aplikasi yang *cerdas* dan *pintar mengingat*, lho! Dengan bantuan fitur **TinyDB**, kamu bisa menyimpan data yang tetap ada meski aplikasi ditutup. Mau simpan skor tertinggi, riwayat kalkulasi, atau daftar belanja? Semua bisa! Kamu juga akan berkenalan dengan **struktur data List** dan **ListPicker**, supaya aplikasi buatanmu makin rapi dan dinamis. Plus, kamu akan mulai memahami dasar-dasar **AI (Artificial Intelligence)** – dari mengenali gambar sampai membaca tulisan – dan bagaimana menggabungkannya ke dalam aplikasi menggunakan alat seperti *Teachable Machine*.

Semua materi dikemas dengan contoh kasus nyata yang seru, seperti *kalkulator ngambek*, *catatan hilang misterius*, sampai *aplikasi pelupa*. Kamu diajak bukan cuma untuk membaca teori, tapi juga **menguji aplikasi buatan sendiri**, **memperbaiki bug**, dan **mengembangkan fitur baru**. Setiap aktivitas dan tantangan dirancang agar kamu bisa berpikir seperti programmer sungguhan. Siap jadi pahlawan di balik layar yang bisa membuat aplikasi pintar, ramah pengguna, dan bebas error? Yuk mulai petualanganmu!

Latihan Soal

A. Benar atau Salah (5 Soal)

1. Fitur "Do It" di App Inventor digunakan untuk langsung menjalankan aplikasi dari awal.

Jawaban: Salah

2. TinyDB bisa menyimpan data meskipun aplikasi ditutup.
   Jawaban: Benar

3. Pesan kesalahan yang baik adalah yang menggunakan kode teknis seperti "Error 404".
   Jawaban: Salah

4. Komponen ListPicker memungkinkan pengguna memilih data dari daftar.
   Jawaban: Benar

5. App Inventor tidak bisa digunakan untuk membuat aplikasi yang menggunakan AI.
   Jawaban: Salah

B. Pilihan Ganda (10 Soal)

1. Fungsi dari TinyDB adalah untuk:
   a. Mengatur tampilan
   b. Menyimpan data secara lokal
   c. Menghapus file
   d. Mengatur suara
   Jawaban: b

2. Saat tombol ditekan dan aplikasi crash, kita perlu melakukan:
   a. Animasi
   b. Debugging
   c. Instal ulang aplikasi
   d. Restart HP
   Jawaban: b

3. Komponen yang digunakan untuk menampilkan daftar pilihan:
   a. Label
   b. Image
   c. ListPicker
   d. Button
   Jawaban: c

4. Fitur "Do It" berguna untuk:
   a. Menyimpan file
   b. Menjalankan seluruh aplikasi
   c. Mengecek blok secara langsung
   d. Menghapus error
   Jawaban: c

5. Ketika koneksi internet gagal, aplikasi sebaiknya:
   a. Langsung ditutup
   b. Menampilkan pesan error
   c. Restart otomatis
   d. Menghapus data
   Jawaban: b

6. Yang termasuk strategi debugging adalah:
   a. Menambahkan musik

b. Membagi masalah ke bagian kecil
c. Menambah warna
d. Mengganti nama aplikasi
Jawaban: b

7. Untuk menyimpan data skor tertinggi dalam game, kita bisa menggunakan:
   a. Canvas
   b. Clock
   c. TinyDB
   d. WebViewer
   Jawaban: c

8. Komponen AI di App Inventor digunakan untuk:
   a. Memutar video
   b. Mengatur resolusi layar
   c. Mengenali gambar atau teks
   d. Membuat animasi
   Jawaban: c

9. Contoh pesan kesalahan yang baik:
   a. “Error!”
   b. “404 not found”
   c. “Gagal, coba lagi nanti ya!”
   d. “Crash detected”
   Jawaban: c

10. Untuk menampilkan daftar belanja yang bisa dipilih dan dihapus, digunakan:
    a. Clock dan Notifier
    b. List dan ListPicker
    c. Web dan Bluetooth
    d. Label dan Button
    Jawaban: b

C. Isian Singkat (5 Soal)

1. Proses mencari dan memperbaiki kesalahan dalam aplikasi disebut _____.
   Jawaban: debugging

2. TinyDB menyimpan data dalam format _____ dan value.
   Jawaban: tag

3. Fitur App Inventor yang memungkinkan kita melihat langsung nilai dari blok adalah _____.
   Jawaban: Do It

4. Komponen yang menampilkan pesan peringatan kepada pengguna adalah _____.
   Jawaban: Notifier

5. AI membutuhkan _____ agar bisa mengenali pola dengan baik.
   Jawaban: data latih

D. Soal Eksplorasi

Soal 1:

Jelaskan langkah-langkah yang bisa kamu lakukan jika aplikasi buatanmu crash setiap kali tombol "Simpan" ditekan. Sertakan contoh strategi debugging dan cara menggunakan fitur "Do It" untuk membantu menemukan masalah tersebut.

Soal 2:

Bayangkan kamu membuat aplikasi pendeteksi gambar menggunakan AI. Jelaskan bagaimana kamu melatih model AI sederhana (misalnya menggunakan Teachable Machine) dan cara mengintegrasikannya ke App Inventor agar bisa mengenali objek di gambar.

Soal 3:

Aplikasi catatan buatanmu bisa menyimpan dan menampilkan catatan menggunakan TinyDB dan List. Jelaskan bagaimana kamu mendesain alur kerja aplikasi tersebut dari pengguna mengetik, menyimpan, hingga data muncul kembali saat aplikasi dibuka ulang.

Tugas Proyek

Proyek 1: Aplikasi Kalkulator dengan Riwayat Perhitungan

Deskripsi Proyek:
Kembangkan aplikasi kalkulator yang bisa melakukan operasi dasar (tambah, kurang, kali, bagi), dan menampilkan **5 hasil perhitungan terakhir** dengan menyimpan data menggunakan **TinyDB**.

Langkah Saran:

- Merancang antarmuka dengan tombol angka, operasi, dan satu Label untuk menampilkan hasil.
- Tambahkan logika agar setiap hasil disimpan di TinyDB.
- Buat kondisi: jika riwayat lebih dari 5, hapus data paling lama.
- Tampilkan riwayat di Label atau ListView agar pengguna bisa melihat perhitungan terakhir.

Tantangan Tambahan: Tambahkan validasi agar pengguna tidak memasukkan karakter selain angka.

Proyek 2: Desain Aplikasi To-Do List

Deskripsi Proyek:
Desain aplikasi daftar tugas (To-Do List) di App Inventor yang memungkinkan pengguna **menambah, melihat, dan menandai tugas sebagai selesai**.

Langkah Saran:

- Gunakan TextBox untuk menulis tugas.
- Gunakan Button untuk menambahkan tugas ke dalam daftar.
- Tampilkan daftar dengan ListView.

- Gunakan CheckBox atau ubah warna tulisan untuk menandai tugas yang sudah selesai.
- Fokuskan proyek pada **desain antarmuka terlebih dahulu**, bukan logika program.

Tantangan Tambahan: Desain agar tampilan tetap menarik dan mudah digunakan.

Proyek 3: Game Kuis Sederhana dengan Skor Tertinggi

Deskripsi Proyek:
Buat game kuis menggunakan App Inventor dengan fitur penyimpanan **skor tertinggi**, dan tampilkan **pesan kesalahan yang ramah** jika pemain belum mengisi nama.

Langkah Saran:

- Buat kumpulan soal dan pilihan jawaban.
- Gunakan TextBox untuk nama pemain.
- Simpan skor tertinggi di TinyDB.
- Tambahkan pemeriksaan: jika nama kosong → tampilkan pesan seperti “Isi nama dulu ya, supaya skor kamu tercatat!”

Tantangan Tambahan: Tambahkan penghargaan seperti “Top Skorer Hari Ini” berdasarkan nilai tertinggi.

Proyek 4: Aplikasi Catatan Hobi atau Buku Favorit

Deskripsi Proyek:
Ciptakan aplikasi untuk mencatat **hobi, buku favorit, atau film kesukaan**. Data harus tetap tersimpan meskipun aplikasi ditutup, dengan bantuan TinyDB.

Langkah Saran:

- Gunakan TextBox untuk menulis item baru.
- Gunakan Button untuk menambah ke daftar.
- Tampilkan daftar dengan ListView.
- Simpan semua data ke TinyDB agar tidak hilang.

Tantangan Tambahan: Tambahkan fitur pencarian atau filter berdasarkan kategori.

Proyek 5: Aplikasi Riwayat Belajar Harian

Deskripsi Proyek:
Buat aplikasi yang mencatat **aktivitas belajar harian**, seperti berapa jam belajar atau mata pelajaran yang dipelajari.

Langkah Saran:

- Gunakan TextBox untuk menulis kegiatan belajar.

- Gunakan Button untuk menambahkan ke daftar.
- Tampilkan semua kegiatan belajar yang telah disimpan menggunakan ListPicker.
- Gunakan TinyDB agar data tidak hilang.

Tantangan Tambahan: Tambahkan grafik sederhana (bisa pakai label berurutan) untuk menunjukkan durasi belajar per hari.

Proyek 6: Uji dan Perbaiki Aplikasi Teman

Deskripsi Proyek:
Bekerja dalam kelompok untuk **mengulas aplikasi teman**, menemukan minimal 3 bug, dan memperbaikinya bersama.

Langkah Saran:

- Jalankan aplikasi teman dan catat kesalahan atau hal yang kurang bekerja.
- Diskusikan kemungkinan penyebabnya.
- Lakukan perbaikan bersama.
- Buat laporan singkat: apa saja bug-nya, bagaimana memperbaikinya, dan hasil akhirnya.

Tantangan Tambahan: Dokumentasikan proses debugging dalam bentuk video atau presentasi kelompok.

BAB 2: Proyek Kolaboratif: Tim Membangun Aplikasi

Tujuan Pembelajaran

Setelah mengikuti pembelajaran ini, peserta didik diharapkan mampu:

- Bekerja sama dalam tim untuk merancang dan membuat aplikasi mobile.
- Membagi peran dan tanggung jawab dalam tim secara efektif.
- Mengintegrasikan teknologi kecerdasan buatan (AI) dari Teachable Machine ke dalam aplikasi.
- Menggunakan App Inventor dan alat kolaborasi digital secara efisien.

Peta Konsep



Apersepsi

Pernahkah kalian membuat aplikasi sederhana seperti kalkulator, kuis, atau daftar tugas? Mungkin terasa mudah dan menyenangkan, ya. Tapi, bayangkan sekarang kalian ditantang untuk menciptakan sebuah aplikasi yang jauh lebih canggih—misalnya aplikasi yang bisa mengenali gambar lewat kamera, membedakan suara manusia, atau menyimpan banyak data dan digunakan banyak orang. Wah, pasti seru, tapi juga jauh lebih menantang!

Pertanyaannya, bisakah kalian mengerjakannya sendirian? Mulai dari desain, coding, pengujian, sampai pelatihan AI, semuanya dilakukan sendiri? Tentu tidak mudah. Di sinilah pentingnya kerja tim. Dalam sebuah tim, kalian bisa membagi tugas sesuai keahlian masing-masing—ada yang mendesain tampilan, ada yang ngoding, ada yang menguji, bahkan ada yang melatih AI! Dengan kerja sama yang baik, kalian bisa belajar satu sama lain, menyelesaikan masalah bersama, dan menciptakan aplikasi yang keren dan berkualitas. Melalui proyek ini, kalian akan membangun aplikasi mobile yang terhubung dengan teknologi AI, dan yang paling seru: kalian akan mengerjakannya bersama tim! Yuk, buktikan bahwa kerja tim bisa membawa kalian menciptakan sesuatu yang luar biasa!

Penjelasan Konsep (Teori)

Kerja Tim: Kunci Sukses dalam Membangun Aplikasi AI

Membangun aplikasi yang canggih, apalagi yang terintegrasi dengan teknologi AI (Artificial Intelligence), bukanlah pekerjaan satu orang saja. Seperti halnya membangun rumah, kita memerlukan tim dengan keahlian yang berbeda-beda. Nah, dalam dunia teknologi, kerja tim sangat penting untuk menyelesaikan proyek besar dan kompleks dengan sukses!

Berikut ini adalah konsep-konsep utama yang perlu kalian pahami dalam proyek kolaboratif membangun aplikasi.

Mengapa Harus Bekerja dalam Tim?

Bekerja bersama tim dalam membangun aplikasi bukan hanya soal mempercepat pekerjaan, tapi juga membuat prosesnya lebih seru dan hasilnya lebih keren! Yuk kita lihat alasannya:

- **Skalabilitas (Bisa Lebih Besar dan Rumit)**
  Kalau aplikasimu hanya sederhana, mungkin bisa dikerjakan sendiri. Tapi untuk aplikasi dengan banyak fitur dan kecerdasan buatan, kamu butuh tim. Ibarat bangun rumah besar, kamu perlu tukang listrik, tukang cat, arsitek, dan lainnya.
- **Bagi-bagi Keahlian**
  Setiap orang punya kelebihan. Ada yang jago desain, ada yang ahli coding, ada juga yang teliti saat menguji aplikasi. Semua bisa fokus di bidangnya, jadi aplikasi yang dihasilkan lebih rapi, bagus, dan berfungsi dengan baik.
- **Cepat Menyelesaikan Masalah**
  Kalau ada bug atau error, kamu tidak sendirian! Temanmu mungkin tahu solusinya atau bisa bantu cari jalan keluar bersama. Ingat, dua otak lebih baik dari satu.
- **Belajar dan Berkembang Bersama**
  Dalam tim, kalian bisa saling belajar. Misalnya kamu belajar desain dari temanmu, dan dia belajar logika program dari kamu. Kalian sama-sama naik level!

Siapa Saja yang Ada dalam Tim Proyek Aplikasi?

Supaya kerja lebih terarah, tiap anggota tim biasanya punya peran khusus. Ini dia peran-peran penting yang biasa ada:

- **Project Manager (Koordinator Tim)**
  Pemimpin proyek yang mengatur pembagian tugas, mengingatkan deadline, dan memastikan semua berjalan lancar.
- **UI/UX Designer**
  Bertugas membuat tampilan aplikasi yang menarik dan nyaman digunakan. Mereka juga membuat sketsa awal (mockup) sebelum aplikasi dibangun.
- **Programmer (Pengembang di App Inventor)**
  Menyusun logika aplikasi menggunakan blok-blok di App Inventor. Mereka juga menghubungkan aplikasi ke AI agar bisa mengenali gambar atau suara.
- **Data Scientist / AI Trainer**
  Melatih model AI dengan data seperti gambar atau suara, menggunakan tools seperti Teachable Machine. Lalu, mereka sambungkan model itu ke aplikasi.
- **Tester / Quality Assurance (QA)**
  Menguji aplikasi agar tidak ada bug, memastikan semua fitur berjalan baik, dan memberikan masukan untuk perbaikan.

Kunci Komunikasi Efektif dalam Tim

Tanpa komunikasi yang baik, kerja tim bisa berantakan. Makanya, kalian perlu:

- **Pertemuan Harian Singkat (Stand-up Meeting)**
  Setiap anggota menjawab 3 pertanyaan:
  - Apa yang saya kerjakan kemarin?
  - Apa yang akan saya kerjakan hari ini?
  - Apakah saya mengalami kendala?

- **Gunakan Alat Kolaborasi**
  - **Chat Grup (WhatsApp/Telegram)**: Untuk tanya jawab cepat dan diskusi ringan.
  - **Google Docs / Sheets**: Untuk membuat laporan bersama, mencatat tugas, atau membuat jadwal.

- **Memberi Saran dengan Sopan**
  Saat memberi masukan, gunakan kata-kata yang baik dan membangun. Contoh:
  ✅ “Warna tombolnya mungkin terlalu redup, coba ganti yang lebih cerah, yuk!”
  ❌ “Tombolnya jelek banget.”
  Saran yang baik membuat suasana tim tetap positif dan saling mendukung.

Cara Kerja Efisien di App Inventor (Teknik Modular dan Pengelolaan File)

App Inventor belum bisa digunakan secara bersamaan oleh banyak orang seperti Google Docs. Jadi, kita perlu strategi:

- **Pembagian File Proyek (.aia)**
  Setiap screen bisa dikerjakan oleh orang berbeda. Setelah selesai, digabungkan ke dalam proyek utama oleh koordinator.
- **Manajemen Aset yang Rapi**
  Aset seperti gambar, suara, atau model AI disimpan di folder bersama, misalnya di Google Drive. File diberi nama jelas agar mudah ditemukan.
- **Membangun Modul Terpisah**
  Bagi aplikasi menjadi modul-modul kecil, seperti:
    - Modul login
    - Modul kamera AI
    - Modul galeri
    - Modul informasi
- Tiap anggota mengerjakan modul tertentu. Setelah semua selesai, modul-modul ini digabungkan menjadi satu aplikasi utuh.

Kesimpulan: Bersama Kita Bisa!

Kerja tim dalam proyek membangun aplikasi bukan hanya tentang siapa yang paling jago. Ini tentang **bagaimana kita bisa saling bantu, belajar bersama, dan menciptakan sesuatu yang luar biasa**. Dengan komunikasi yang baik, pembagian tugas yang jelas, dan semangat kolaborasi, kalian bisa membangun aplikasi AI yang keren dan bermanfaat!

> Jadi... siap menjadi bagian dari tim hebat yang menciptakan aplikasi masa depan? 💪✨

Contoh Kasus atau Ilustrasi

Kasus 1: Detektif Daun – Aplikasi Pengenal Tanaman dari Foto Daun

> **Ilustrasi Kasus:**
> Bayangkan kamu dan timmu menjadi *detektif botani digital*. Kalian membuat aplikasi yang bisa menebak jenis tanaman hanya dengan memotret daunnya. Dari daun jambu sampai daun pepaya, aplikasi kalian akan “membaca” bentuk dan warna daun lalu menebaknya dengan bantuan AI.
>
> Tiap anggota tim punya peran seru: ada yang jadi pengatur strategi (Project Manager), ada yang mendesain tampilan keren aplikasi (UI/UX Designer), ada juga yang melatih AI supaya pintar membedakan daun (Data Scientist), hingga yang

membangun aplikasinya langsung (Programmer). Setiap pagi, kalian briefing bareng—ngobrolin kemajuan dan solusi kalau ada bug atau salah prediksi.

Tantangan:
AI kalian malah sering salah: daun mangga dikira daun jambu! Tim harus bekerja sama memperbaiki model dengan data yang lebih lengkap dan beragam.

Kasus 2: Si Pemilah Cerdas – Aplikasi Pendeteksi Sampah Organik & Anorganik

Ilustrasi Kasus:
Kalian membuat aplikasi untuk membantu orang memilah sampah hanya dengan mengambil fotonya. Kulit pisang? Itu organik. Plastik kresek? Anorganik. Tapi kadang, AI kalian bingung—mirip seperti manusia saat ragu antara botol kaca dan kaleng.

Di sini, kalian menjadi tim penyelamat lingkungan digital. Ada yang melatih AI dengan ratusan gambar sampah, ada yang membuat aplikasi dengan tombol hijau untuk “organik” dan abu-abu untuk “anorganik”, dan tentu saja ada yang memastikan aplikasi berjalan cepat di semua HP.

Tantangan:
Foto plastik putih sering dikira organik! Kalian harus kreatif: tambah data baru, ubah desain UI supaya hasil muncul lebih cepat, dan berdiskusi rutin agar semua anggota tetap semangat.

Kasus 3: Pendeteksi Suasana Hati – Aplikasi Pembaca Ekspresi Wajah

Ilustrasi Kasus:
Pernah lihat kamera yang bisa “membaca” emosi orang? Nah, kalian membuat aplikasi seperti itu! Saat seseorang tersenyum, muncul emoji senyum dan tulisan “Kamu terlihat senang!” Kalau ekspresi cemberut? Emoji sedih pun muncul.

Proyek ini terasa seperti campuran antara teknologi dan psikologi. AI kalian dilatih dengan gambar ekspresi marah, senang, dan netral. Desain aplikasinya harus simpel dan fun, supaya anak-anak juga bisa pakai.

Tantangan:
AI kalian sering salah kira orang yang cemberut sebagai marah. UI juga terlalu ramai sehingga aplikasi jadi lambat. Tim harus menyederhanakan desain dan memperbaiki model dengan lebih banyak contoh wajah dari berbagai latar belakang.

Kasus 4: Pendeteksi Buah Segar – Aplikasi Penilai Kematangan Buah

Ilustrasi Kasus:
Bayangkan kalian jadi tim ahli pasar digital. Kalian membuat aplikasi untuk membantu orang mengetahui apakah buah (seperti pisang atau mangga) sudah matang atau belum, cukup dengan memotretnya.

Model AI dilatih untuk membedakan warna, bintik, dan tekstur buah. Programmer membuat aplikasi agar bisa langsung membaca foto dari kamera HP. Sementara UI/UX Designer mendesain tampilan dengan warna cerah dan ikon buah segar.

Tantangan:
AI sering bingung antara pisang yang hampir matang dengan yang masih mentah. Harus ditambahkan contoh dari buah di berbagai kondisi. Selain itu, UI-nya harus tetap ceria, tapi tidak berat dijalankan di HP sederhana.

Kasus 5: Aplikasi Pendeteksi Mood Suara – Mengenali Emosi dari Nada Bicara

Ilustrasi Kasus:
Kalian merancang aplikasi pintar yang bisa mengenali emosi dari suara. Ketika seseorang berbicara dengan nada tinggi dan cepat, aplikasi bisa menandai itu sebagai “marah”. Kalau lembut dan lambat? Mungkin sedang “sedih”.

Kalian harus merekam dan melabeli suara-suara itu, lalu melatih AI-nya agar peka terhadap perubahan nada. UI-nya juga harus bisa menampilkan animasi atau ikon yang menggambarkan suasana hati.

Tantangan:
AI sering gagal membedakan suara sedih dan netral karena nada bicaranya mirip. Selain itu, suara di ruang ramai membuat hasil deteksi kacau. Tim perlu bekerja sama mencari cara menyaring suara dan melatih ulang model.

Kasus 6: Aplikasi Pengingat Posisi Duduk – Mengingatkan Saat Membungkuk

Ilustrasi Kasus:
Kalian membuat aplikasi yang bisa mengingatkan pengguna jika duduk dengan posisi bungkuk terlalu lama—supaya tubuh tetap sehat! Menggunakan kamera depan laptop atau HP, aplikasi ini mendeteksi postur duduk pengguna secara otomatis.

Data Scientist melatih AI-nya dengan gambar posisi duduk yang benar dan yang salah. Programmer menghubungkannya ke notifikasi yang bisa muncul ketika postur buruk terdeteksi. UI-nya pun dibuat simpel: ikon tubuh, warna hijau untuk “bagus”, dan merah untuk “perlu diperbaiki”.

Tantangan:
AI sering salah deteksi karena pencahayaan di ruangan gelap. Tim harus menambah data dari berbagai kondisi cahaya dan membuat aplikasi tetap ringan agar tidak mengganggu aktivitas pengguna.

Kalau kamu seorang siswa SMP dan membaca semua kasus di atas, bayangkan: **mana yang paling kamu ingin coba?** Semua kasus mengajakmu untuk berpikir kritis, berkomunikasi, bekerjasama, dan berinovasi—dengan cara yang seru dan bermakna!

Contoh Soal dan Pembahasan

Soal:

Bayangkan kamu adalah bagian dari tim pengembang aplikasi "Deteksi Hewan Peliharaan". Bagaimana cara kerja sama antara Programmer dan Data Scientist agar model AI dari Teachable Machine bisa digunakan dengan baik dalam aplikasi MIT App Inventor? Jelaskan!

Pembahasan:
Programmer dan Data Scientist harus bekerja sama dengan saling memahami kebutuhan teknis masing-masing.

- **Data Scientist** melatih model AI menggunakan gambar (misalnya kucing, anjing, kelinci) di Teachable Machine, lalu mengekspor modelnya ke format yang cocok (seperti TFLite atau model web).
- Ia juga membuat **dokumentasi** berisi label (nama-nama hewan), ukuran gambar yang tepat (misalnya 224x224 piksel), dan format input.
- **Programmer** menggunakan dokumentasi itu untuk **mengintegrasikan model ke App Inventor**, menggunakan extension yang mendukung AI.
- Hasil prediksi (misalnya "Anjing") akan disimpan di TinyDB agar bisa ditampilkan kembali.

Dengan komunikasi yang lancar dan saling membantu, aplikasi dapat berjalan mulus!

Soal:

Tim sedang mengembangkan aplikasi AI. Siapa saja anggota tim dan apa tugas utamanya dalam proyek ini? (Sebutkan Project Manager, UI/UX Designer, Programmer, Data Scientist, dan Tester)

Pembahasan:

- **Project Manager**: Menyusun jadwal kerja, membagi tugas, dan memastikan tim tetap selaras dengan tujuan proyek.
- **UI/UX Designer**: Mendesain tampilan aplikasi agar menarik dan mudah digunakan, serta sesuai dengan kemampuan App Inventor.
- **Programmer**: Membangun logika aplikasi dan menghubungkan model AI ke aplikasi menggunakan blok kode dan TinyDB.
- **Data Scientist**: Mengumpulkan data (gambar), melatih model AI, dan menyediakan model serta dokumentasi untuk programmer.
- **Tester**: Mencoba semua fitur aplikasi, melaporkan bug, dan memastikan model AI serta UI bekerja dengan baik.

Soal:

Saat mendesain aplikasi, UI/UX Designer ingin meletakkan tombol deteksi gambar di tengah layar agar mudah dijangkau. Namun Programmer kesulitan

menyambungkannya ke logika program. Bagaimana sebaiknya tim menyelesaikan perbedaan ini?

Pembahasan:
Perbedaan pendapat sebaiknya diselesaikan dengan:

1. **Diskusi terbuka** – mendengarkan alasan dari sisi desain dan teknis.
2. **Uji coba antarmuka** – coba tampilkan dua versi desain ke pengguna, lihat mana yang paling nyaman.
3. **Kompromi** – cari posisi tombol yang tetap menarik secara visual tapi juga mudah diatur secara teknis.

Prinsipnya, keputusan harus **berpusat pada kenyamanan pengguna**, sambil tetap realistis bagi teknis tim.

Soal:

Model AI yang sudah dilatih dan diekspor oleh Data Scientist ternyata tidak berfungsi saat digunakan di App Inventor. Aplikasi tidak bisa mendeteksi gambar. Apa yang harus dilakukan tim?

Pembahasan:
Langkah yang bisa dilakukan tim antara lain:

1. **Periksa dokumentasi model** – Pastikan ukuran dan format gambar yang dimasukkan sesuai dengan yang dibutuhkan model.
2. **Cek integrasi teknis di App Inventor** – Pastikan model dihubungkan dengan benar (URL tepat, blok kode sesuai).
3. **Uji coba langkah demi langkah** – Misalnya tes dulu apakah gambar bisa dikirim ke model sebelum cek hasil prediksi.
4. **Kolaborasi ulang** – Programmer bisa minta bantuan Data Scientist untuk cek apakah model masih berfungsi di platform web.

Kolaborasi dan komunikasi tetap menjadi kunci suksesnya integrasi.

Soal:

Apa manfaat dari dokumentasi yang dibuat oleh Data Scientist, dan mengapa dokumentasi itu penting bagi Programmer saat membuat aplikasi AI?

Pembahasan:
Dokumentasi dari Data Scientist berisi informasi penting, seperti:

- Label atau kelas yang dikenali oleh model (misalnya: kucing, anjing, kelinci).
- Format input (ukuran gambar, jenis file).

Bagi **Programmer**, dokumentasi ini sangat berguna agar mereka tahu bagaimana cara **mengirim data ke model AI dengan benar** dan **menampilkan hasil prediksi dengan tepat** di aplikasi. Tanpa dokumentasi ini, integrasi bisa jadi gagal atau membingungkan.

Fakta Menarik / Fun Facts

1. **Aplikasi Hebat, Bukan dari Orang Hebat Saja!**
   Tahukah kamu? Aplikasi keren seperti Snapchat dan Spotify lahir dari ide-ide mahasiswa biasa yang kerja bareng teman kampusnya. Jadi bukan harus jenius dulu, tapi yang penting bisa kerja sama dan punya ide yang out of the box!

2. **AI Bisa Dilatih Seperti Peliharaan!**
   Model AI di Teachable Machine bisa “belajar” mengenali hal-hal seperti daun, suara, atau ekspresi wajah—mirip seperti kamu melatih anjing duduk atau lompat! Kamu kasih contoh beberapa kali, lalu dia mulai paham. Tapi jangan lupa: AI juga bisa "lupa" kalau datanya kurang bagus!

3. **Desainer Aplikasi = Seniman Digital**
   Peran UI/UX Designer itu seperti seniman di dunia teknologi. Mereka menciptakan tampilan aplikasi yang bukan cuma cantik, tapi juga nyaman digunakan. Kalau aplikasi bikin orang bingung, mungkin desainnya perlu dirombak, bukan kodenya!

4. **"Stand-up Meeting" Itu Bukan Lawakan!**
   Dalam dunia developer profesional, ada yang namanya daily stand-up meeting. Tapi bukan buat ngelawak! Ini pertemuan cepat (biasanya sambil berdiri) di pagi hari untuk cerita: apa yang dikerjakan kemarin, hari ini, dan apa kesulitannya. Jadi tim tetap selaras dan fokus!

5. **Cuma Satu yang Ngoding? Gak Asik!**
   Kalau cuma satu orang yang ngoding, yang lain ngapain dong? 😅 Dalam proyek nyata, semua anggota tim punya peran penting: mulai dari desain, pelatihan AI, testing, sampai dokumentasi. Proyek jadi asik kalau semua aktif dan saling dukung!

6. **TinyDB: Si Laci Ajaib Aplikasi!**
   TinyDB itu seperti laci kecil dalam aplikasimu. Bisa dipakai buat nyimpen skor tertinggi game, nama pengguna, atau pilihan tema. Dan yang paling keren: isinya tetap tersimpan walau aplikasi ditutup!

7. **AI Bukan Sihir, Tapi Cuma Statistik!**
   Banyak yang mikir AI itu kayak sihir, padahal dasarnya cuma matematika dan statistik. Dia hanya menebak berdasarkan pola dari data yang sudah kamu kasih. Jadi, kalau datamu ngaco, hasilnya juga bisa ngawur!

8. **Minecraft Dulu Dibuat Sendirian, Tapi…**
   Awalnya Minecraft dibuat oleh satu orang, Markus Persson. Tapi setelah makin sukses, dia bentuk tim! Karena semakin besar aplikasinya, makin banyak orang yang dibutuhkan untuk mengelolanya. Bahkan raksasa seperti Google atau Apple pun kerja tim terus-menerus.

9. **Gagal Itu Teman Dekat Programmer**
   Setiap programmer pasti pernah bikin aplikasi yang gagal jalan. Tapi justru dari

situlah mereka belajar. Bahkan di perusahaan teknologi besar, tester memang disuruh cari error! Jadi gagal bukan malu-maluin, tapi bagian dari proses belajar.

Tips Belajar atau Tips Cepat Menghafal

1. **Merubah materi jadi cerita nyata / ilustrasi**
   Contoh: Anggap kamu jadi *detektif digital* yang bikin aplikasi pelacak daun. Biar lebih seru dan gampang diingat!

2. **Gunakan Mind Mapping / Peta Konsep**
   Visualisasi membantu otak menghubungkan ide lebih cepat dibanding teks panjang.

3. **Pakai singkatan / jembatan keledai**
   Misal: P.U.P.D.T – untuk 5 peran penting dalam tim: **P**M, **U**X, **P**rogrammer, **D**ata Scientist, **T**ester

4. **Ajarin kembali ke teman**
   Kalau kamu bisa menjelaskan ke orang lain, artinya kamu sudah menguasai.

5. **Belajar pakai warna & gambar**
   Otak manusia lebih cepat mengingat visual. Gunakan warna berbeda untuk setiap cabang mind map.

Mind Mapping Proyek Kolaboratif



Contoh Memudahkan Ingatan

Analogi Rumah

Bangun aplikasi	= bangun rumah
Tukang batu	= Programmer
Arsitek	= UI/UX Designer
Surveyor tanah	= Data Scientist
Pengawas bangunan	= Tester
Mandor proyek	= Project Manager

Aktivitas Siswa

Aktivitas 1: Bentuk Tim Hebatmu! (Kolaborasi & Penjelajahan Ide)

Tujuan: Menumbuhkan kerja sama dan imajinasi dalam tim.

Langkah-Langkah:

- Bentuk tim berisi 3–5 orang yang siap seru-seruan bareng membangun aplikasi AI.
- Duduk bersama dan lakukan sesi brainstorming: “Kira-kira aplikasi AI apa yang bisa kita buat dan berguna?”
- Buat papan ide (boleh digital atau kertas) berisi usulan anggota tim. Beri nilai bintang untuk ide yang paling memungkinkan dan keren untuk dibuat.

Tantangan Seru: Coba temukan ide aplikasi yang *unik*, belum banyak dibuat, tapi bisa dilatih dengan Teachable Machine. Bisa tentang tumbuhan, suara binatang, hingga ekspresi wajah!

Aktivitas 2: Pilih Peranmu, Tunjukkan Gaya Mainmu! (Pemetaan Kekuatan Tim)

Tujuan: Menyadari kekuatan masing-masing dalam sebuah proyek nyata.

Langkah-Langkah:

- Setiap anggota tim menyebutkan minat dan kemampuan mereka. Misalnya: “Aku suka desain!” atau “Aku tertarik coding!”.
- Cocokkan dengan 5 peran utama: **Project Manager**, **UI/UX Designer**, **Programmer**, **Data Scientist**, **Tester**.
- Buat papan peran tim dan tempelkan nama anggota dengan peran masing-masing. Gunakan warna atau ikon untuk memperjelas.

Tips: Tidak apa-apa kalau masih pemula! Yang penting kamu tertarik belajar dan siap saling bantu.

Aktivitas 3: Rancang Proyekmu seperti Arsitek Aplikasi! (Pembuatan Rencana Proyek Mini)

Tujuan: Melatih perencanaan kerja kolaboratif dalam format yang ringan.

Langkah-Langkah:

- Bersama tim, tentukan: nama aplikasi, tujuan utamanya, fitur yang ingin dibuat (misalnya kamera AI, tombol deteksi, tampilan hasil).
- Buat “Peta Rencana Proyek” dalam bentuk tabel, infografis, atau presentasi mini. Sebutkan pembagian tugas dan target waktu pengerjaan (boleh mingguan).

Tantangan Menantang: Pastikan rencana kalian realistis dan bisa diselesaikan dalam waktu yang disepakati. Gunakan kalender sederhana agar mudah dipantau.

Aktivitas 4: Simulasi Harian ala Developer Profesional (Mini Stand-up Meeting)

Tujuan: Membiasakan komunikasi efektif dalam kerja tim digital.

Langkah-Langkah:

- Setiap hari kerja tim (misalnya seminggu 2 kali), lakukan “pertemuan cepat” berdurasi 5–10 menit.
- Masing-masing anggota menjawab tiga pertanyaan:
    1. Apa yang sudah aku kerjakan?
    2. Apa yang akan aku kerjakan?
    3. Apa kendala yang aku hadapi?

Saran: Lakukan sambil berdiri atau pakai timer agar tidak terlalu lama. Bisa lewat grup chat jika daring.

Aktivitas 5: Bangun Aplikasi Modular Secara Bertahap (Kolaborasi Asinkron)

Tujuan: Mengelola proyek besar menjadi bagian-bagian kecil yang bisa dikerjakan serempak.

Langkah-Langkah:

- Bagi aplikasi menjadi beberapa **modul**: misalnya halaman utama, halaman kamera, halaman hasil deteksi, dll.
- Setiap anggota bertanggung jawab atas satu atau dua modul. Pastikan tidak mengerjakan **screen yang sama** dalam App Inventor.
- Gunakan folder bersama untuk menyimpan aset seperti gambar, suara, dan model AI. Nama file harus jelas dan konsisten.

Tantangan: Melatih koordinasi dengan memastikan setiap perubahan di dokumentasi dan disimpan rapi.

Aktivitas 6: Tes, Koreksi, dan Perbaiki Bareng! (Simulasi QA & Debugging)

Tujuan: Mengasah kemampuan menguji dan memperbaiki aplikasi sebagai tim.

Langkah-Langkah:

- Setelah aplikasi awal selesai dibuat, semua anggota mencoba aplikasinya di HP masing-masing.
- Buat daftar bug/kesalahan/fungsi yang tidak berjalan. Diskusikan penyebab dan siapa yang akan memperbaikinya.
- Gunakan format tabel atau catatan bersama: *Bug - Lokasi - Solusi - Status*.

Fun Point: Jadikan ini seperti “permainan detektif digital”! Siapa paling cepat menemukan bug?

Dengan mengikuti aktivitas-aktivitas ini, kalian tidak hanya belajar membuat aplikasi berbasis AI, tapi juga belajar menjadi **tim yang solid, kreatif, dan profesional muda di dunia digital**. Ayo jadikan proyek ini pengalaman belajar tak terlupakan!

Rangkuman

Dalam membuat aplikasi canggih berbasis *Artificial Intelligence* (AI), kita tidak bisa bekerja sendirian. **Kerja sama tim sangat penting**, karena setiap anggota bisa berkontribusi sesuai keahlian mereka. Ada yang jago desain (UI/UX Designer), ada yang andal dalam logika dan coding (Programmer), hingga yang teliti dalam menguji aplikasi (Tester). Dengan saling berbagi peran dan tugas, proyek pun bisa berjalan lebih cepat, lebih rapi, dan hasilnya lebih memuaskan. Seperti membangun rumah, semakin besar proyeknya, semakin banyak tangan yang dibutuhkan!

Tapi kerja tim tidak cuma soal tugas—**komunikasi yang efektif juga kunci suksesnya**. Setiap anggota perlu saling memberi kabar, rutin berdiskusi, dan menyampaikan ide atau masukan dengan cara yang sopan dan membangun. Gunakan alat kolaborasi seperti *Google Docs* untuk menulis bersama, atau *chat group* untuk koordinasi cepat. Jangan lupa, tantangan pasti muncul—entah itu kesalahan AI, desain yang membingungkan, atau jadwal yang molor. Di sinilah kekompakan dan kemampuan problem solving tim diuji!

Untuk menyelesaikan proyek seperti aplikasi pengenal daun atau pendeteksi suasana hati, **kerja modular sangat disarankan**. Bagi proyek menjadi beberapa bagian kecil, lalu kerjakan secara paralel. Simpan data dan aset secara rapi di folder bersama, dan pastikan hanya satu orang yang menggabungkan semua bagian di App Inventor. Dengan strategi ini, kalian bisa membangun aplikasi AI yang keren sambil belajar bekerja seperti developer profesional. Seru, menantang, dan pastinya penuh pengalaman tak terlupakan!

Latihan Soal

A. Benar atau Salah (5 Soal)

1. Dalam proyek aplikasi AI, hanya programmer yang bertanggung jawab atas hasil akhir.
   Salah

2. Model AI dari Teachable Machine bisa digunakan dalam App Inventor.
   Benar

3. Pembagian tugas dalam tim tidak diperlukan selama semua orang bisa ngoding.
   Salah

4. UI/UX Designer bertugas membuat tampilan dan pengalaman pengguna aplikasi.
   Benar

5. TinyDB digunakan untuk menyimpan data sementara yang akan hilang saat aplikasi ditutup.
   Salah

B. Pilihan Ganda (10 Soal)

1. Siapa yang bertanggung jawab melatih model AI dalam tim proyek?
   A. Programmer
   B. Tester
   C. Project Manager
   D. Data Scientist
   Jawaban: D

2. Fungsi utama TinyDB adalah...
   A. Mengirim data ke internet
   B. Menyimpan data secara lokal di perangkat
   C. Melatih AI
   D. Membuat grafik
   Jawaban: B

3. Apa manfaat kerja tim dalam pengembangan aplikasi?
   A. Semua tugas dilakukan sendiri
   B. Proyek lebih lambat tapi lebih rapi
   C. Pembagian tugas sesuai keahlian
   D. Hanya satu orang yang mengerjakan
   Jawaban: C

4. Yang dimaksud *stand-up meeting* adalah...
   A. Pertemuan panjang membahas teori AI
   B. Diskusi cepat membahas progres kerja
   C. Latihan berdiri bersama
   D. Evaluasi aplikasi oleh pengguna
   Jawaban: B

5. Apa yang tidak dilakukan oleh UI/UX Designer?
   A. Mendesain tampilan aplikasi
   B. Menulis logika program
   C. Membuat mockup aplikasi
   D. Mengatur kenyamanan pengguna
   Jawaban: B

6. Model AI yang dilatih di Teachable Machine bisa mengenali...
   A. Bahasa pemrograman
   B. Warna, gambar, suara
   C. Blok kode

D. Data base
Jawaban: B

7. Salah satu contoh peran Project Manager adalah...
   A. Membuat efek suara
   B. Menyusun jadwal kerja tim
   C. Menyimpan data dengan TinyDB
   D. Melatih model AI
   Jawaban: B

8. Jika programmer mengalami kesulitan dalam integrasi AI, siapa yang sebaiknya dihubungi?
   A. Tester
   B. UI/UX Designer
   C. Project Manager
   D. Data Scientist
   Jawaban: D

9. Komunikasi dalam tim bisa dilakukan melalui...
   A. Kode rahasia
   B. Stand-up meeting dan chat grup
   C. Sinyal tangan
   D. Memo kertas saja
   Jawaban: B

10. Supaya semua bagian aplikasi bisa digabung, apa strategi terbaiknya?
    A. Bekerja di screen yang sama
    B. Saling menyimpan proyek secara sembarangan
    C. Kerjakan bagian berbeda lalu gabungkan secara modular
    D. Tidak usah koordinasi
    Jawaban: C

C. Isian Singkat (5 Soal)

1. Orang yang bertugas menguji aplikasi dan mencari bug disebut...
   Tester

2. Platform yang digunakan untuk melatih model AI dalam proyek ini adalah...
   Teachable Machine

3. File proyek App Inventor disimpan dalam format...
   .aia

4. Peran yang membuat tampilan aplikasi agar mudah digunakan disebut...
   UI/UX Designer

5. Dalam tim proyek, orang yang memimpin dan membagi tugas disebut...
   Project Manager

D. Soal Eksplorasi (3 Soal Essay Singkat)

1. Bayangkan kamu adalah bagian dari tim pembuat aplikasi pendeteksi ekspresi wajah. Jika AI kalian sering salah menebak ekspresi, strategi apa yang bisa kamu

dan tim lakukan agar model jadi lebih akurat? Jelaskan langkah-langkahnya.

2. Kamu dan timmu membuat aplikasi pendeteksi kematangan buah. Namun hasil deteksi tidak konsisten karena pencahayaan berbeda. Bagaimana kamu akan memperbaiki hal ini? Apa yang akan dilakukan oleh masing-masing peran dalam tim untuk mengatasi masalah ini?

3. Suatu tim membuat aplikasi pemilah sampah otomatis, tapi anggota tim jarang berkomunikasi sehingga banyak pekerjaan yang tumpang tindih. Buatlah rencana komunikasi tim yang efektif agar proyek bisa berjalan lebih lancar, efisien, dan tidak membingungkan.

Tugas Proyek

Proyek 1: “Detektif Daun” – Aplikasi Pengenal Tanaman Berdasarkan Gambar Daun

Deskripsi:
Tim membuat aplikasi yang bisa mengenali jenis tanaman hanya dari foto daunnya. Cocok untuk pecinta alam dan pelajar IPA!

Langkah-langkah Pelaksanaan:

1. **Data Scientist** mengumpulkan dan melabeli gambar berbagai jenis daun (misalnya daun mangga, pepaya, jambu).
2. **UI/UX Designer** membuat tampilan aplikasi dengan ikon tanaman dan hasil deteksi yang ramah pengguna.
3. **Programmer** menyambungkan model AI dari Teachable Machine ke aplikasi melalui MIT App Inventor.
4. **Tester** mencoba foto daun yang berbeda dan mencatat jika ada prediksi yang salah.
5. **Project Manager** memastikan semua data dan fitur digabung dengan rapi dalam satu aplikasi utuh.

Tantangan:
AI sering salah membedakan daun yang mirip. Tim harus memperluas data latih dan melakukan pengujian berkala.

Proyek 2: “Si Pemilah Cerdas” – Aplikasi Pemilah Sampah Otomatis

Deskripsi:
Aplikasi ini membantu pengguna memilah sampah organik dan anorganik cukup dengan memfoto sampahnya.

Langkah-langkah Pelaksanaan:

1. **Data Scientist** melatih AI dengan ratusan foto sampah (kulit pisang, botol plastik, kertas).
2. **UI/UX Designer** menciptakan antarmuka dengan dua warna tombol: hijau (organik), abu-abu (anorganik).
3. **Programmer** membuat logika deteksi dan menampilkan hasil klasifikasi secara otomatis.
4. **Tester** melakukan uji coba dengan berbagai jenis sampah dan pencahayaan.
5. **Project Manager** membuat jadwal evaluasi dan pengumpulan ulang data jika hasil masih tidak akurat.

Tantangan:
AI bisa tertipu oleh warna plastik putih. Solusinya: tambah data lebih beragam dan pertajam algoritma deteksi.

Proyek 3: “MoodCam” – Aplikasi Pembaca Ekspresi Wajah

Deskripsi:
Aplikasi ini bisa membaca ekspresi wajah pengguna lalu menampilkan emoji dan pesan motivasi yang sesuai.

Langkah-langkah Pelaksanaan:

1. **Data Scientist** mengumpulkan gambar ekspresi wajah (senang, marah, netral) dan melatih AI.
2. **UI/UX Designer** membuat desain lucu dan ekspresif yang cocok untuk anak-anak hingga remaja.
3. **Programmer** menambahkan fitur kamera dan menampilkan hasil deteksi secara real-time.
4. **Tester** menguji aplikasi dengan berbagai ekspresi dan mencatat respons AI.
5. **Project Manager** mengevaluasi performa dan memandu tim dalam penambahan fitur, misalnya emoji bergerak.

Tantangan:
AI bisa bingung antara wajah “sedih” dan “lelah”. Tim perlu memperbaiki desain dan kumpulkan data dari lebih banyak wajah.

Proyek 4: “Buah Cerdas” – Aplikasi Pendeteksi Kematangan Buah

Deskripsi:
Membantu pembeli buah menilai apakah pisang, mangga, atau jeruk sudah matang dengan cara memfotonya.

Langkah-langkah Pelaksanaan:

1. **Data Scientist** melatih AI dengan gambar buah dalam berbagai tahap kematangan.
2. **UI/UX Designer** membuat antarmuka cerah dengan indikator “Matang” / “Belum Matang”.
3. **Programmer** membuat fitur kamera dan klasifikasi hasil deteksi buah.

4. **Tester** menguji dengan kondisi pencahayaan berbeda dan buah dari berbagai pasar.
5. **Project Manager** memantau perkembangan dan menyatukan seluruh modul aplikasi.

Tantangan:
Pencahayaan mempengaruhi akurasi. Solusinya: tambah data dari berbagai kondisi dan berikan tips pencahayaan di dalam aplikasi.

Proyek 5: “Mood Voice” – Aplikasi Pendeteksi Emosi dari Suara

Deskripsi:
Aplikasi ini bisa mengenali emosi (senang, marah, sedih) dari cara seseorang berbicara.

Langkah-langkah Pelaksanaan:

1. **Data Scientist** merekam suara dengan emosi berbeda dan melatih AI di Teachable Machine.
2. **UI/UX Designer** menambahkan ikon suara, efek gelombang, dan warna sesuai mood.
3. **Programmer** menyambungkan input suara dengan model AI untuk menganalisis nada bicara.
4. **Tester** mencobanya dalam lingkungan tenang dan bising untuk membandingkan hasil.
5. **Project Manager** mengatur sesi evaluasi dan mendorong tim melakukan pembaruan data jika perlu.

Tantangan:
Suara latar yang bising membuat deteksi kacau. Gunakan filter suara sederhana atau minta pengguna berbicara di tempat tenang.

Proyek 6: “Posture Alert” – Aplikasi Pengingat Posisi Duduk

Deskripsi:
Aplikasi ini membantu pengguna agar tidak duduk bungkuk terlalu lama, demi kesehatan tulang belakang.

Langkah-langkah Pelaksanaan:

1. **Data Scientist** melatih AI dengan gambar postur duduk baik dan buruk.
2. **UI/UX Designer** mendesain tampilan minimalis dengan ikon tubuh dan lampu indikator.
3. **Programmer** mengaktifkan kamera dan notifikasi otomatis jika postur buruk terdeteksi.
4. **Tester** mencoba aplikasi dalam berbagai posisi dan pencahayaan.
5. **Project Manager** mencatat hasil pengujian dan menyarankan pembaruan model AI.

Tantangan:
Ruangan gelap membuat AI salah deteksi. Tim perlu mengatur standar pencahayaan atau memberi peringatan di aplikasi.

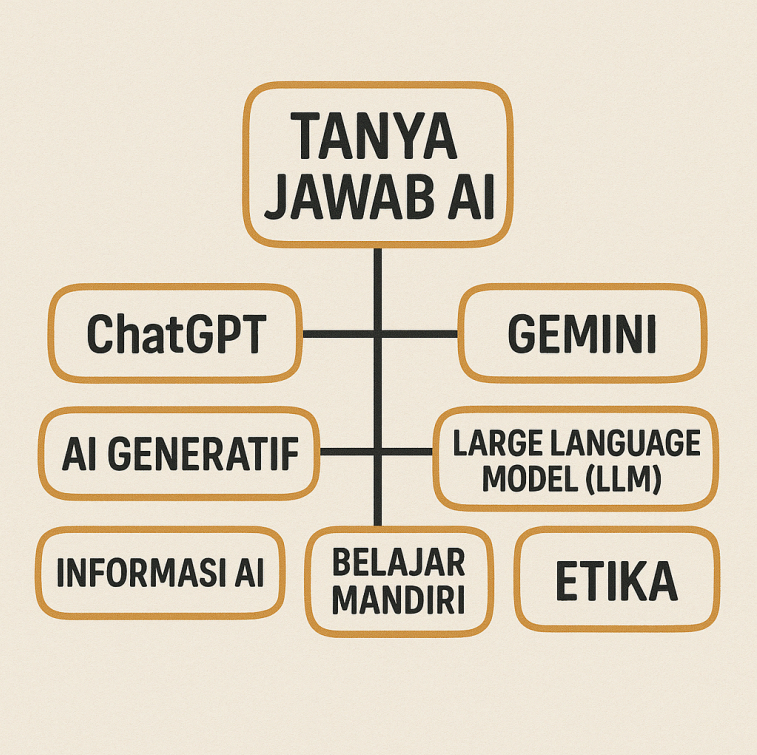
Seluruh proyek di atas menantang siswa untuk **berpikir kritis**, **bekerja sama**, dan **membangun aplikasi nyata** dengan bantuan AI. Dengan membagi peran secara adil dan menjalankan proyek secara modular, tim SMP bisa menghasilkan karya digital yang profesional dan membanggakan.

BAB 3: Tanya Jawab AI dengan ChatGPT & Gemini

Tujuan Pembelajaran

Peserta didik mampu menggunakan AI generatif (ChatGPT, Gemini) untuk mencari informasi dan memahami konsep AI yang lebih luas, serta memverifikasi informasi.

Peta Konsep



Apersepsi

Pernah nggak, kalian bingung mau tanya sesuatu, tapi nggak tahu ke siapa?
Misalnya soal PR, resep makanan, atau cara main game.
Biasanya kalian cari jawabannya di mana?

Mungkin di Google, tanya ke Siri, atau ngobrol sama ChatGPT dan Gemini.
Nah, ChatGPT dan Gemini itu contoh dari **AI** — kecerdasan buatan!

Mereka bisa bantu kita cari info dengan cepat, seperti ngobrol sama orang beneran.
Tapi... apakah semua jawaban mereka selalu benar?
Belum tentu! Kita harus **pintar bertanya**, dan juga **kritis mengecek jawabannya**.

Di bab ini, kita akan belajar:

- Apa itu AI generatif seperti ChatGPT dan Gemini,
- Cara memakainya dengan tepat,
- Dan bagaimana mengecek kebenaran informasi dari mereka.

Yuk, kita eksplorasi bersama dan jadi pengguna AI yang cerdas dan kreatif!
Siap tantangan baru? 🚀✨

Penjelasan Konsep (Teori)

Di era digital ini, teknologi semakin canggih dan pintar. Salah satu teknologi yang sering kita dengar adalah **AI (Artificial Intelligence)** atau dalam bahasa Indonesia disebut **Kecerdasan Buatan**. Nah, ada jenis AI yang keren banget, namanya **AI Generatif**. Apa itu?

Pengertian AI Generatif

AI Generatif adalah jenis kecerdasan buatan yang tidak hanya bisa mencari atau menyalin informasi yang sudah ada, tapi juga bisa **menciptakan sesuatu yang baru**! Misalnya, kamu minta dibuatkan cerita tentang "kucing yang bisa terbang", AI bisa langsung membuat cerita yang belum pernah ada sebelumnya.

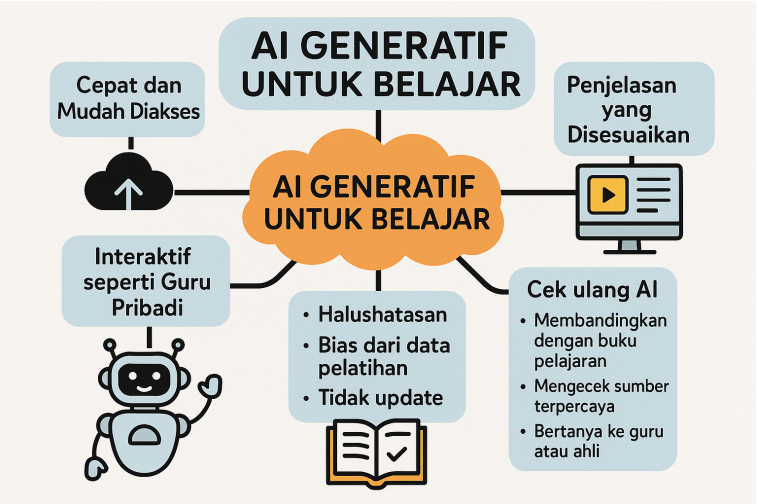
Dua contoh terkenal dari AI generatif adalah:

- **ChatGPT**
- **Gemini**

Keduanya termasuk dalam kelompok teknologi canggih yang disebut **Large Language Model (LLM)**. Model ini belajar dari miliaran kata dan kalimat yang dikumpulkan dari buku, artikel, dan internet. Karena itu, mereka sangat hebat dalam memahami bahasa dan menjawab berbagai pertanyaan dengan baik.

AI Generatif untuk Belajar: Teman Pintar yang Siap Bantu!

Belajar tidak harus membosankan! Dengan bantuan AI generatif seperti ChatGPT dan Gemini, proses belajar bisa jadi lebih menyenangkan dan efektif.



Berikut beberapa alasan mengapa AI sangat membantu dalam belajar:

1. **Cepat dan Mudah Diakses**
   Bisa digunakan kapanpun

dan dimanapun asalkan ada internet.

2. **Penjelasan yang Disesuaikan**
   Mau penjelasan singkat atau mendalam? AI bisa menyesuaikan dengan kebutuhan kita.
3. **Interaktif seperti Guru Pribadi**
   Kita bisa bertanya, mengulang, bahkan minta dijelaskan dengan gaya yang lebih sederhana.

Namun, AI juga punya **keterbatasan**, seperti:

- Kadang memberikan **jawaban yang tidak akurat (halusinasi)**.
- Bisa **terpengaruh bias** dari data pelatihan.
- Tidak selalu **update dengan informasi terbaru**.

Karena itu, penting banget untuk **mengecek ulang jawaban dari AI** dengan cara:

- Membandingkan dengan buku pelajaran
- Mengecek sumber terpercaya
- Bertanya ke guru atau ahli

Cara Bertanya yang Efektif ke AI (Dasar Prompt Engineering)

Agar AI memberikan jawaban yang **tepat dan berguna**, kita perlu tahu cara membuat pertanyaan yang baik. Yuk, simak tipsnya!

Tips Bertanya ke AI

1. **Pertanyaan Harus Jelas dan Spesifik**
   ❌ "Jelaskan Neural Network."
   ✅ "Jelaskan Neural Network untuk anak SMP kelas 9 dengan contoh sederhana."

2. **Berikan Konteks**
   Misalnya:
   - "Jelaskan untuk tugas sekolah."
   - "Buat penjelasan untuk siswa kelas 8."

3. **Tentukan Format Jawaban**
   Kita bisa minta:
   - Dalam bentuk **daftar poin**
   - **Tabel perbandingan**
   - **Dialog sederhana**
   - **Ringkasan satu paragraf**



4. **Lakukan Iterasi (Perbaikan Pertanyaan)**
   Jika jawaban pertama kurang pas, kita bisa:

- Minta versi lebih sederhana
- Tambahkan instruksi
- Ulangi dengan pertanyaan yang lebih terarah

Contoh:

- “Tolong jelaskan lebih mudah.”
- “Coba buat versi untuk anak SD.”
- “Bisa dijelaskan dalam tabel?”

Contoh Penggunaan AI Generatif dalam Kehidupan Sehari-Hari

Ternyata AI tidak hanya untuk belajar, loh! Berikut ini beberapa contoh penggunaan AI dalam aktivitas harian:

Kegiatan	Contoh Permintaan ke AI
Ringkasan Berita	“Ringkas berita ini jadi 5 poin.”
Menulis Email	“Buatkan email izin tidak masuk sekolah karena sakit.”
Merencanakan Liburan	“Buatkan itinerary liburan 3 hari ke Bali.”
Brainstorming Ide	“Beri ide konten media sosial tentang kebersihan.”

Kesimpulan

AI Generatif seperti ChatGPT dan Gemini adalah **alat yang luar biasa** untuk belajar dan kehidupan sehari-hari. Tapi ingat, kita harus **menggunakannya dengan cerdas dan kritis**. Dengan bertanya dengan baik, mengecek ulang informasi, dan memahami batasannya, kita bisa menjadikan AI sebagai **teman belajar yang menyenangkan dan penuh tantangan**!

Contoh Kasus atau Ilustrasi

Kasus 1: Si Detektif AI: Menyelidiki Apa Itu Kecerdasan Buatan

Situasi:
Raka, siswa kelas 8, penasaran dengan istilah *Artificial Intelligence (AI)* yang sering muncul di berita dan media sosial.

Ilustrasi:
Raka berperan sebagai detektif muda yang sedang menyelidiki makhluk "AI" misterius. Ia mengetik pertanyaan di ChatGPT:

> “Apa itu Artificial Intelligence dan apa contohnya?”

AI menjawab dengan penjelasan yang menarik. Namun Raka tidak berhenti di situ! Ia memeriksa jawaban dari ensiklopedia digital dan buku pelajaran. Dari sana, ia membandingkan informasi dan mencatat hal-hal yang cocok atau berbeda.

Kesimpulan:
AI bisa memberi informasi awal, tapi seperti detektif sejati, kita perlu mencari bukti tambahan agar yakin.

Kasus 2: Si Ahli Bahasa: Menaklukkan Istilah Sulit

Situasi:
Nadia kesulitan memahami konsep “Algoritma” di pelajaran Informatika.

Ilustrasi:
Bayangkan Nadia seperti sedang bermain teka-teki rumit. Ia meminta bantuan AI:

> “Tolong jelaskan algoritma dengan cara yang mudah, seperti menjelaskan resep memasak.”

AI menjawab dengan membandingkan algoritma dengan langkah-langkah membuat mie instan: mulai dari merebus air sampai menyajikan. Nadia jadi lebih paham, dan bahkan minta contoh algoritma dalam kehidupan sehari-hari.

Kesimpulan:
AI bisa menjelaskan konsep rumit dengan cara seru dan sederhana, seperti teman yang jago bercerita.

Kasus 3: Si Penemu Ide: Menciptakan Proyek Canggih dari Nol

Situasi:
Kelas Dika diminta membuat proyek teknologi bertema “AI dalam Kehidupan Sehari-hari”.

Ilustrasi:
Dika membuka ChatGPT dan bertanya:

> “Apa ide proyek AI sederhana untuk anak SMP yang bisa dibuat pakai HP?”

AI menyarankan:

- Aplikasi chatbot ramah untuk menjawab pertanyaan sekolah
- Aplikasi mengenali suara dan memberi respon lucu
- Aplikasi rekomendasi buku bacaan sesuai minat

Dika langsung semangat memilih satu ide dan menyusun langkah-langkah pembuatannya.

Kesimpulan:
AI bisa menjadi rekan brainstorming yang tak pernah kehabisan ide.

Kasus 4: Si Juru Latih Presentasi: Menyusun Slide yang Keren

Situasi:
Laras akan presentasi tentang “AI dan Etika” di depan kelas, tapi bingung harus mulai dari mana.

Ilustrasi:
Laras mengetik di Gemini:

> “Bantu aku membuat outline presentasi tentang AI dan etika untuk SMP.”

AI menyusun poin-poin menarik, lengkap dengan contoh dan urutan logis. Laras bahkan minta AI bantu membuat slide pembuka yang catchy dan pertanyaan untuk audiens.

Kesimpulan:
AI bisa menjadi pelatih presentasi pribadi yang siap membantu dari awal sampai akhir.

Kasus 5: Si Editor Hebat: Mengecek Tulisan Secara Cerdas

Situasi:
Rian menulis artikel untuk majalah sekolah tentang “AI di Masa Depan”, tapi tidak yakin apakah bahasanya sudah bagus.

Ilustrasi:
Rian mengetik tulisannya di ChatGPT dan meminta:

> “Tolong periksa dan perbaiki tulisan ini supaya lebih jelas dan menarik.”

AI memberikan saran perbaikan: ada yang perlu disingkat, ada kata-kata yang bisa diganti agar lebih kuat. Rian belajar cara menulis lebih baik dari hasil koreksi tersebut.

Kesimpulan:
AI bisa menjadi editor virtual yang membantu kita menulis lebih rapi dan efektif.

Kasus 6: Si Penjelajah Masa Depan: Membayangkan Dunia AI

Situasi:
Guru meminta semua siswa membayangkan dunia 10 tahun ke depan jika AI sudah ada di semua bidang.

Ilustrasi:
Salsa bertanya ke AI:

> “Ceritakan skenario tahun 2035 di mana AI digunakan di sekolah, rumah, dan jalan raya.”

AI menjawab dengan kisah menarik: ada robot guru di sekolah, rumah pintar yang tahu suasana hati penghuninya, dan mobil tanpa sopir di jalan. Salsa terinspirasi membuat cerita fiksi ilmiah dari situ.

Kesimpulan:
AI bisa menjadi mesin waktu imajinatif yang membantu kita menjelajah masa depan dengan ide-ide kreatif.

Contoh Soal dan Pembahasan

Soal 1: Mengenal Deep Learning Lewat AI

Pernah dengar istilah *Deep Learning*? Yuk, tantang dirimu!
Cobalah tanyakan kepada ChatGPT atau Gemini:
"Apa itu Deep Learning?"
Lalu, bandingkan penjelasan AI dengan penjelasan dari buku pelajaran atau artikel internet.
Apa saja perbedaan penting yang kamu temukan?

Pembahasan:
Penjelasan dari AI sering lebih singkat, praktis, dan memakai bahasa santai, cocok untuk pelajar. Namun, penjelasan seperti ini kadang kurang lengkap atau tidak menyebutkan sumber. Sebaliknya, buku atau artikel edukatif biasanya disusun oleh ahli, dilengkapi referensi ilmiah, dan menjelaskan lebih mendalam. Maka, membandingkan dua sumber bisa membantumu memahami konsep secara lebih utuh dan melatih kemampuan berpikir kritis.

Soal 2: Cerdas Membuat Pertanyaan ke AI

Kamu penasaran: “Bagaimana sih AI digunakan dalam dunia kedokteran?”
Coba pikirkan, seperti apa bentuk pertanyaan yang tepat supaya ChatGPT atau Gemini bisa memberikan jawaban yang jelas dan lengkap?

Pembahasan:
Pertanyaan yang baik itu **spesifik dan fokus**, misalnya:

> “Berikan 3 contoh penggunaan AI dalam kedokteran, seperti diagnosis penyakit, robot bedah, atau pengembangan obat.”
> Dengan begitu, AI tahu kamu menginginkan **jawaban yang terstruktur**, jelas, dan sesuai kebutuhanmu. Ini melatihmu untuk berpikir kritis sebelum bertanya.

Soal 3: Meringkas Materi IPA dengan Bantuan AI

Bayangkan kamu dapat tugas IPA untuk meringkas bab "Sistem Saraf Manusia". Bagaimana kamu bisa memanfaatkan AI seperti ChatGPT atau Gemini agar ringkasanmu jadi lebih mudah dipahami?

Pembahasan:
Kamu bisa salin teks dari buku, lalu beri perintah seperti:

> “Tolong ringkas teks ini dengan bahasa sederhana untuk siswa SMP.”
> AI akan meringkas poin-poin penting. Tapi ingat, kamu **tetap perlu membaca ulang ringkasannya** agar sesuai dengan isi asli dan tidak ada bagian penting yang terlewat. Ini membantu kamu belajar menyaring informasi dengan lebih cerdas.

Soal 4: Menyaring Informasi AI Tentang Energi Terbarukan

Kamu bertanya ke AI:

> “Apa manfaat energi terbarukan bagi lingkungan?”
> Jawaban yang diberikan cukup bagus. Tapi, bagaimana cara memastikan bahwa jawaban tersebut benar dan tidak menyesatkan?

Pembahasan:
Langkah terbaik adalah **memverifikasi** jawaban tersebut dengan sumber lain, seperti buku pelajaran, situs Kementerian Lingkungan Hidup, atau website edukatif seperti Britannica atau National Geographic. Tujuannya adalah memastikan informasi dari AI **sesuai fakta dan bisa dipercaya**. Ini adalah bagian penting dari belajar mandiri yang bijak.

Soal 5: Kapan Harus Percaya Jawaban AI?

AI seperti ChatGPT bisa menjelaskan banyak hal, tapi tidak selalu benar. Menurutmu, kapan kita bisa mempercayai jawaban dari AI, dan kapan kita perlu lebih hati-hati?

Pembahasan:
AI sangat berguna untuk memahami hal baru dengan cepat, terutama dalam bahasa yang mudah dimengerti. Namun, AI tidak selalu menyebutkan sumber, bisa saja memberikan informasi yang tidak lengkap atau kurang akurat. Maka, **percaya boleh, tapi jangan langsung menelan mentah-mentah**. Selalu cek ulang informasi penting, apalagi untuk tugas sekolah atau topik ilmiah.

Fakta Menarik / Fun Facts

Fun Fact:

> Kalau otak serangga seperti lebah punya sekitar **1 juta neuron**, ChatGPT punya **triliunan "neuron digital"!** Tapi jangan tertipu, meskipun besar, AI tetap gak punya perasaan seperti lebah yang bisa takut atau marah 😄.

Fun Fact:

> Cara AI belajar itu kayak main tebak-tebakan! Misalnya:
> "Aku makan nasi dan…"
> AI akan coba menebak: "ayam?", "tempe?", atau "kerupuk?"
> Seru banget, kan? Belajarnya mirip kita saat baca cerita misteri

Fun Fact:

> Nama "Gemini" artinya **kembar** dalam bahasa Latin! Kenapa? Karena AI ini bisa **multibahasa dan multimodal** – kayak dua saudara kembar yang jago banyak hal: bisa baca, dengar, lihat, bahkan menulis dalam berbagai bahasa

Fun Fact:

> AI seperti ChatGPT bisa terdengar **ramah, sedih, atau lucu**, tapi itu semua cuma akting! Ia **tidak benar-benar merasakan emosi**, hanya meniru gaya manusia. Jadi, kalau AI bilang "aku bangga padamu", itu cuma gaya bahasa, bukan perasaan 🥲

Fun Fact:

> Meskipun pintar, AI **bisa salah!** Kadang-kadang dia malah **ngarang jawaban**. Jadi, selalu cek ulang informasi dari AI, ya! Pintar pakai AI itu penting, tapi lebih penting lagi pintar **menyaringnya**!

Kalimat-kalimat ini bisa dipakai untuk presentasi, poster, atau materi edukatif yang seru di kelas. Mau dibuat versi kuis atau meme juga bisa!

Tips Belajar atau Tips Cepat Menghafal

Tips: AI itu seperti Guru, Kamu Editornya

> Jawaban AI bisa membantu, tapi tetap harus dicek. Misalnya:

- Tanya: "Apa itu Deep Learning?"
- Dapat jawaban AI? Bandingkan dengan buku atau situs edukatif.
- Temukan perbedaan, buat kesimpulan sendiri.

Tips: Tanya, Periksa, Simpulkan

Misal:

- Tanya: “Jelaskan AI dengan bahasa sederhana.”
- Periksa: Apakah masuk akal dan benar?
- Simpulkan: “AI itu seperti otak digital yang bisa belajar dari teks.”

Tips: Prompt yang Baik = Jawaban Berkualitas

Misal:

- Jangan: “Apa itu AI?”
- Lebih baik: “Jelaskan AI untuk anak SMP dengan contoh sehari-hari.”
- Tambahkan konteks, format, tujuan

Tips: Belajar bareng AI itu Diskusi, bukan Sekali Tanya

seperti teman belajar:

- Belum paham? Tanya lagi: “Bisa lebih sederhana?”
- Butuh format? “Tolong buat dalam bentuk tabel.”
- Diskusi = Pemahaman Lebih Dalam

Tips: Catat Poin Penting, Bukan Semua

Contoh hasil diskusi dengan AI:

Machine Learning = Mesin belajar dari data, seperti manusia belajar dari pengalaman.
Tambahkan warna, gambar, atau simbol supaya cepat ingat.

Lima (5) langkah belajar dengan AI

1. **Tanya dengan jelas** (spesifik, pakai konteks)
2. **Dapatkan jawaban AI**
3. **Cek ulang kebenaran info**
4. **Ulangi kalau belum paham**
5. **Catat dengan bahasamu sendiri**

Teknik Menghafal Tambahan (Berpadu dengan AI)

- **Ulangi informasi** 3x dalam waktu berbeda (spaced repetition)
- **Jelaskan kembali ke teman** seolah-olah kamu gurunya
- **Gunakan analogi** (contoh: AI = otak digital, Prompt = remote control)
- **Buat mind map** atau sketsa visual agar lebih mudah diingat
- **Fokus 1 topik tiap sesi belajar, hindari multitasking**

Aktivitas Siswa

Aktivitas 1: “Tanya AI, Tapi Cerdas!”

Tujuan: Melatih kemampuan bertanya yang spesifik dan efektif ke ChatGPT/Gemini.

Langkah-langkah:

- Coba tanyakan sesuatu ke AI yang benar-benar kamu ingin tahu, misalnya: “Apa bedanya otak manusia dan AI?”
- Setelah mendapat jawaban, **evaluasi**: Apakah jawabannya jelas? Bisa dimengerti?
- Coba ubah pertanyaannya jadi lebih spesifik atau tambahkan konteks. Misalnya: “Jelaskan bedanya otak manusia dan AI untuk anak kelas 8 dalam 5 poin.”
- Bandingkan kedua jawaban, mana yang lebih cocok untukmu?
- Rasakan sendiri bahwa kualitas pertanyaan = kualitas jawaban.

Tips Seru: Anggap kamu sedang menjadi “detektif AI” yang ingin menggali info dengan cara bertanya paling cerdas!

Aktivitas 2: “Benar atau Berhalusinasi?”

Tujuan: Mengasah kemampuan memverifikasi informasi dari AI.

Langkah-langkah:

- Pilih topik menarik, misalnya: “Kapan AI pertama kali dibuat?”
- Tanyakan ke ChatGPT/Gemini.
- Setelah itu, **bandingkan jawabannya dengan buku pelajaran, Wikipedia, atau situs resmi.**
- Tandai informasi yang sama, yang berbeda, atau yang diragukan.
- Diskusikan bersama teman: *Apakah AI selalu benar?*

Tantangan seru: Bisa tidakkamu menemukan 1 “halusinasi AI”? (Jawaban meyakinkan tapi ternyata salah?)

Aktivitas 3: “Prompt-ku, Gayaku!”

Tujuan: Melatih kreativitas dalam membuat prompt yang unik dan bervariasi.

Langkah-langkah:

- Buat 1 pertanyaan umum, contohnya: “Jelaskan sistem saraf manusia.”
- Ubah jadi 3 versi berbeda:
    - “Jelaskan sistem saraf seperti kamu sedang bercerita ke anak kecil.”
    - “Tampilkan penjelasan dalam bentuk tabel ringkas.”
    - “Jelaskan seperti dialog antara dua siswa SMP.”

- Lihat perbedaan gaya jawabannya, dan tentukan mana yang paling kamu suka!

Seru-seruan: Buat tantangan antarkelompok: *Siapa yang bisa membuat prompt paling kreatif dan hasil paling keren?*

Aktivitas 4: “AI Jadi Teman Proyek”

Tujuan: Menggunakan AI untuk eksplorasi ide kreatif atau proyek mini.

Langkah-langkah:

- Bayangkan kamu diminta membuat proyek mini sains atau sosial.
- Minta ide dari AI, misalnya: “Berikan 3 ide proyek tentang lingkungan untuk siswa SMP.”
- Pilih salah satu ide yang menarik, lalu minta AI bantu menyusun langkah-langkah awalnya.
- Diskusikan apakah ide itu bisa dilaksanakan. Tambahkan ide dari dirimu sendiri!

Simulasi keren: AI adalah “asisten brainstorming”, tapi keputusan dan kreativitas tetap kamu yang pegang!

Aktivitas 5: “Ringkasan Kilat Bareng AI”

Tujuan: Menggunakan AI untuk membantu merangkum materi pelajaran.

Langkah-langkah:

- Ambil teks dari buku pelajaran (misalnya bab IPA: Sistem Saraf).
- Salin ke AI dengan perintah seperti: “Ringkas bab ini dalam 5 poin utama, mudah dimengerti oleh anak SMP.”
- Bandingkan hasilnya dengan ringkasan buatanmu sendiri atau buatan teman.
- Diskusikan: Apakah AI membantumu memahami materi? Apakah semua poin penting sudah disebut?

Challenge kreatif: Gabungkan ringkasan AI + versi kamu jadi satu “super ringkasan” terbaik!

Aktivitas 6: “AI dan Dunia Nyata”

Tujuan: Menghubungkan AI dengan kehidupan sehari-hari.

Langkah-langkah:

- Tanyakan ke AI: “Apa contoh penggunaan AI di rumah atau sekolah?”
- Pilih 2 contoh yang pernah kamu alami (misalnya: YouTube rekomendasi video, Google Translate).
- Bahas: Bagaimana cara kerja AI itu? Apa manfaat dan resikonya?

- Ceritakan pengalamanmu menggunakan fitur tersebut dalam kehidupan nyata.

Menarik: Bayangkan hidupmu sehari saja tanpa teknologi AI — apa yang berubah?

Penutup Aktivitas

Selama eksplorasi ini, anggaplah kamu sedang:

- Belajar *bukan hanya dari AI*, tapi juga *dengan AI*
- Bukan sekadar menerima jawaban, tapi aktif *menyaring, membandingkan, dan mengolah* informasi
- Menjadi generasi digital yang bukan cuma “hebat bertanya”, tapi juga “bijak berpikir kritis”

Kalau kamu siap tantangan, aktivitas-aktivitas ini akan membuat proses belajarmu jadi jauh lebih **seru, bermakna, dan up-to-date!**

Rangkuman

Di zaman serba digital ini, kita semakin sering bertemu dengan *chatbot* canggih seperti **ChatGPT dan Gemini**, dua contoh *AI generatif* yang bisa membantu menjawab pertanyaan, membuat ringkasan pelajaran, atau bahkan memberi ide proyek. Mereka bisa kita ajak ngobrol seolah seperti teman pintar yang tahu segalanya. Tapi hati-hati—meskipun terdengar meyakinkan, jawaban dari AI **tidak selalu benar**. Di bab ini, kita akan **belajar menggunakan AI dengan cerdas**, mulai dari mengenal apa itu *AI generatif*, memahami cara kerjanya, sampai tahu cara memeriksa kebenaran informasinya. Seru, bukan?

Supaya AI bisa memberikan jawaban yang tepat dan membantu, kita harus tahu **cara bertanya yang efektif**. Ini disebut *prompt engineering* dasar. Mulai dari membuat pertanyaan yang **spesifik**, memberi **konteks**, hingga meminta jawaban dalam **format tertentu** seperti tabel atau daftar. Bahkan kalau jawabannya belum pas, kita bisa **mengulang dan memperbaiki pertanyaan**—seperti sedang berdiskusi dengan guru sabar yang tak pernah lelah. Dan serunya lagi, AI ini bisa bantu kita bikin *ringkasan materi pelajaran*, cari ide tugas, atau menjelaskan topik sulit dengan analogi yang mudah. Tapi ingat, setelah mendapat jawaban, kita harus **memverifikasi** dengan sumber lain—seperti buku, situs terpercaya, atau bertanya ke guru.

Penggunaan AI seperti ChatGPT dan Gemini tak hanya untuk belajar, tapi juga bisa jadi **alat bantu kehidupan sehari-hari**. Mulai dari menulis pesan formal, membuat *itinerary* liburan, hingga *brainstorming* ide kreatif. Di bab ini, kamu juga akan diajak mencoba berbagai aktivitas menantang, seperti mencari *halusinasi AI* (jawaban salah tapi terdengar meyakinkan!), membuat *prompt* kreatif, atau membandingkan ringkasan buatan AI dan buatan sendiri. Intinya, AI bukan untuk menggantikan cara belajar kita, tapi untuk **mendukung dan menambah semangat eksplorasi**. Jadilah pelajar yang bukan hanya pintar bertanya, tapi juga **bijak dalam berpikir kritis**!

Latihan Soal

A. Benar atau Salah (5 Soal)

Tulislah “Benar” atau “Salah”

1. ChatGPT dan Gemini termasuk dalam jenis AI generatif.
   Jawaban: Benar

2. Jawaban dari AI seperti ChatGPT dan Gemini selalu benar karena diambil dari sumber terpercaya.
   Jawaban: Salah

3. AI seperti ChatGPT dapat membuat ringkasan teks panjang menjadi poin-poin penting.
   Jawaban: Benar

4. Kita tidak perlu mengecek kembali informasi dari AI jika terdengar meyakinkan.
   Jawaban: Salah

5. ChatGPT dan Gemini bisa membantu menjelaskan topik pelajaran dengan cara yang disesuaikan dengan permintaan.
   Jawaban: Benar

B. Pilihan Ganda (10 Soal)

Pilih jawaban yang paling tepat.

6. Apa itu AI generatif seperti ChatGPT dan Gemini?
   a. AI yang hanya menyalin data dari buku
   b. AI yang bisa menciptakan teks, gambar, atau konten baru
   c. AI yang hanya menjawab soal matematika
   d. AI yang hanya digunakan untuk bermain game
   Jawaban: b

7. Fungsi utama dari prompt yang jelas saat bertanya ke AI adalah...
   a. Membuat AI menjadi lebih cepat
   b. Membuat pertanyaan terlihat bagus
   c. Agar jawaban AI lebih relevan dan sesuai kebutuhan
   d. Agar AI tidak menjawab
   Jawaban: c

8. Yang dimaksud dengan “halusinasi AI” adalah...
   a. AI yang bisa bermimpi
   b. AI yang mendengar suara aneh
   c. AI memberikan jawaban yang terdengar benar tapi salah
   d. AI diam saat diberi pertanyaan

Jawaban: c

9. ChatGPT dan Gemini termasuk ke dalam jenis AI yang disebut...
   a. Jaringan nirkabel
   b. Chatbot biasa
   c. Model Bahasa Besar (LLM)
   d. Mesin pencari otomatis
   Jawaban: c

10. Mengapa penting memverifikasi jawaban dari AI?
    a. Karena AI sering malas menjawab
    b. Karena AI belum tentu paham bahasa manusia
    c. Karena AI bisa salah atau bias
    d. Karena AI tidak bisa membaca
    Jawaban: c

11. Contoh penggunaan AI dalam kehidupan sehari-hari adalah...
    a. Membuat mesin waktu
    b. Memasak otomatis
    c. Rekomendasi video YouTube
    d. Menghitung jarak matahari
    Jawaban: c

12. Salah satu cara agar AI memberikan jawaban yang mudah dipahami adalah...
    a. Menggunakan bahasa asing
    b. Menyuruh AI berpikir keras
    c. Meminta AI menjelaskan dengan analogi
    d. Meminta AI dalam bentuk sandi
    Jawaban: c

13. Contoh prompt yang baik untuk AI adalah...
    a. "Apa itu?"
    b. "Ceritakan."
    c. "Jelaskan perbedaan otak manusia dan AI dalam 3 poin."
    d. "AI pintar nggak?"
    Jawaban: c

14. Salah satu kelebihan AI seperti ChatGPT dan Gemini adalah...
    a. Hanya bisa digunakan saat malam
    b. Bisa diakses kapan saja dengan internet
    c. Harus diinstal dulu di komputer
    d. Tidak bisa menjawab pertanyaan sains
    Jawaban: b

15. Jika kamu tidak puas dengan jawaban dari AI, kamu sebaiknya...
    a. Marah-marah
    b. Menutup aplikasinya
    c. Ulangi atau perbaiki pertanyaanmu

d. Mengganti ke AI lain
Jawaban: c

C. Isian Singkat (5 Soal)

Jawablah secara singkat.

16. Nama lain dari jenis AI seperti ChatGPT dan Gemini yang dapat membuat teks baru adalah _____.
Jawaban: AI generatif

17. Untuk mendapatkan jawaban yang sesuai, kita perlu memberikan pertanyaan yang _____ dan _____.
Jawaban: jelas, spesifik

18. AI seperti ChatGPT bisa menjelaskan ulang sebuah topik jika kita meminta dengan cara _____.
Jawaban: iterasi (mengulang/menyempurnakan pertanyaan)

19. Jika AI memberikan jawaban yang meragukan, kita bisa memverifikasinya melalui _____ atau bertanya kepada _____.
Jawaban: sumber tepercaya, guru/ahli

20. Salah satu alasan pentingnya menyaring informasi dari AI adalah untuk menghindari _____ informasi.
Jawaban: salah

D. Soal Eksplorasi (3 Soal Tantangan)

Tulis masing-masing jawaban dalam paragraf pendek.

Soal:

Buatlah satu pertanyaan yang buruk dan satu pertanyaan yang baik untuk ditujukan ke AI tentang topik “Perubahan Iklim”. Jelaskan mengapa pertanyaan yang baik lebih efektif dalam menghasilkan jawaban berkualitas.
Jawaban:
Pertanyaan buruk: "Perubahan iklim itu apa?"
Pertanyaan baik: "Jelaskan dampak perubahan iklim terhadap kehidupan manusia dan berikan 3 contoh nyata."
Pertanyaan yang buruk terlalu umum dan tidak memberikan arah yang jelas kepada AI. Sementara itu, pertanyaan yang baik bersifat spesifik, fokus pada topik dampak, dan meminta jumlah contoh. Dengan begitu, AI akan memberikan jawaban yang lebih relevan, terstruktur, dan sesuai kebutuhan pengguna.

Soal:

Coba bayangkan kamu diminta membuat proyek sains sederhana dengan bantuan ChatGPT atau Gemini. Tuliskan ide proyeknya, bagaimana kamu meminta bantuan

AI, dan bagaimana kamu akan memverifikasi bahwa idenya layak dan benar.
Jawaban:
Saya membuat proyek sains tentang "Mini Alat Pengukur Suhu Berbasis AI". Saya bertanya kepada ChatGPT: "Berikan ide proyek sains SMP yang menggunakan AI sederhana untuk mengukur suhu lingkungan." ChatGPT memberi ide menggunakan sensor suhu dan pemrosesan data suhu untuk prediksi cuaca sederhana. Saya lalu memverifikasi ide ini dengan mencari referensi dari situs edukasi seperti National Geographic dan buku IPA kelas 9. Setelah membandingkan, saya memastikan bahwa proyek ini aman, layak dilakukan, dan sesuai dengan kurikulum.

Soal:

Kamu membaca artikel AI terbaru dari situs berita. Setelah itu, kamu juga meminta ChatGPT membuat ringkasannya. Tuliskan langkah-langkah apa saja yang akan kamu lakukan untuk memastikan bahwa ringkasan AI tersebut akurat, tidak melewatkan informasi penting, dan mudah dipahami oleh teman sekelasmu.
Jawaban:
Langkah pertama, saya membaca artikel aslinya secara lengkap untuk memahami isi dan poin pentingnya. Kedua, saya membandingkan isi artikel dengan ringkasan dari ChatGPT, mencocokkan apakah semua informasi utama seperti fakta, tanggal, dan nama disebutkan. Ketiga, saya menilai gaya bahasa AI—apakah cukup sederhana dan cocok untuk teman sekelas. Jika ada bagian yang hilang, saya tambahkan sendiri dalam bahasa yang mudah dipahami. Terakhir, saya bertanya ke guru untuk memastikan isi ringkasan sudah sesuai dengan informasi yang benar.

Tugas Proyek

Proyek 1: "AI Fact-Checkers" – Detektif Informasi Zaman Digital

Tujuan: Melatih kemampuan berpikir kritis dan memverifikasi informasi dari AI.
Deskripsi: Siswa berperan sebagai “detektif digital” yang menyelidiki apakah jawaban AI itu benar atau menyesatkan (hallucination).
Langkah Pelaksanaan:

1. Pilih topik tertentu (misalnya: sejarah teknologi, fakta ilmiah, tokoh dunia).
2. Ajukan pertanyaan ke ChatGPT/Gemini.
3. Salin jawabannya.
4. Bandingkan dengan informasi dari buku pelajaran, Wikipedia, atau situs terpercaya.
5. Tandai bagian yang benar, keliru, atau meragukan.
6. Presentasikan hasil penyelidikanmu ke kelas.

Tantangan bonus: Temukan 1 contoh "halusinasi AI" dan jelaskan kenapa bisa terjadi.

Proyek 2: "Prompt Challenge!" – Adu Kreativitas Bikin Pertanyaan

Tujuan: Mengembangkan keterampilan membuat prompt (pertanyaan) yang efektif dan bervariasi.
Deskripsi: Siswa akan merancang berbagai bentuk prompt kreatif untuk topik yang sama, lalu mengevaluasi hasilnya.
Langkah Pelaksanaan:

1. Pilih satu topik (misal: "Perubahan Iklim").
2. Buat 3 jenis prompt berbeda:
    - Prompt biasa (pertanyaan umum).
    - Prompt dengan format tertentu (tabel, daftar, cerita).
    - Prompt dengan gaya komunikasi tertentu (dialog, analogi, untuk anak kecil, dll).
3. Masukkan prompt ke AI.
4. Bandingkan hasilnya. Apa bedanya gaya penjelasan?
5. Tentukan prompt mana yang paling informatif dan menarik.

Mini kompetisi: Kelompok mana yang dapat membuat *prompt* paling kreatif dan hasil AI terbaik?

Proyek 3: "Bikin Proyek Bareng AI" – Ide dari Mesin, Eksekusi dari Manusia

Tujuan: Memanfaatkan AI sebagai alat brainstorming untuk proyek mini berbasis tema (sains, sosial, teknologi).
Deskripsi: Gunakan AI sebagai mitra dalam merancang proyek kreatif.
Langkah Pelaksanaan:

1. Pilih tema proyek (misalnya: lingkungan, energi, teknologi sederhana).
2. Tanyakan ke AI: "Berikan 3 ide proyek kreatif tentang [tema] untuk siswa SMP."
3. Pilih salah satu ide yang menarik.
4. Minta AI bantu membuat langkah-langkah pelaksanaan proyek tersebut.
5. Tambahkan ide dari diri sendiri atau kelompokmu.
6. Rancang poster atau presentasi proyek.

Catatan: AI memberi inspirasi, tapi kreativitas dan realisasi tetap dari kamu!

Proyek 4: “AI Ringkas Materi, Aku Evaluatornya”

Tujuan: Melatih kemampuan menyaring dan mengevaluasi ringkasan dari AI.
Deskripsi: Siswa menggunakan AI untuk merangkum bab pelajaran, lalu membandingkannya dengan versi buatan sendiri.
Langkah Pelaksanaan:

1. Ambil satu bab pelajaran (misalnya IPA – Sistem Saraf).
2. Minta AI membuat ringkasan 5 poin utama.
3. Buat ringkasan versi kamu sendiri tanpa melihat jawaban AI.
4. Bandingkan hasilnya:

    - Apa saja yang sama?
    - Apa yang berbeda?
    - Apakah ada informasi penting yang terlewat?
5. Gabungkan kedua ringkasan menjadi *super ringkasan* yang lengkap dan mudah dipahami.

Tugas akhir: Sajikan ringkasan gabungan sebagai infografik atau mind map.

Proyek 5: "AI & Aku: Cerita Masa Depan"

Tujuan: Mendorong siswa berpikir kritis dan kreatif tentang masa depan AI.
Deskripsi: Buat esai atau video pendek berisi prediksi dan opini pribadi tentang peran AI dalam kehidupan mendatang.
Langkah Pelaksanaan:

1. Buat 5 pertanyaan reflektif, misalnya:
    - Apa dampak positif AI untuk pendidikan?
    - Bagaimana kalau AI menggantikan banyak pekerjaan?
2. Tanyakan ke ChatGPT/Gemini.
3. Rangkum jawabannya dalam poin-poin.
4. Tambahkan pandangan pribadimu: Apakah kamu setuju? Kenapa?
5. Tuangkan hasilnya dalam bentuk esai, komik pendek, atau video.

Opsional: Upload video presentasimu ke platform sekolah atau media sosial dengan izin guru.

Proyek 6: "AI dan Kehidupan Sehari-hari: Observasi & Cerita"

Tujuan: Menyadari peran AI yang tersembunyi dalam aktivitas harian.
Deskripsi: Siswa akan mengamati dan mendokumentasikan penggunaan AI dalam hidup mereka.
Langkah Pelaksanaan:

1. Buat daftar aktivitas sehari-hari (nonton YouTube, Google Maps, Translate, dll).
2. Tanyakan ke AI: "Apakah ini menggunakan teknologi AI?"
3. Catat penjelasannya.
4. Pilih 2–3 aktivitas dan ceritakan:
    - Bagaimana AI bekerja di baliknya?
    - Apa manfaat dan resikonya?
5. Sajikan hasil dalam bentuk slide, poster, atau vlog pendek.

Pilihan gaya: Bisa disajikan seperti *mini dokumenter* tentang “AI di Sekitarku”.

Proyek-proyek ini bukan sekadar tugas—tetapi petualangan belajar di era teknologi. Siswa diajak **berpikir kritis**, **berkreasi**, **berkolaborasi**, dan **beradaptasi dengan AI**, sambil tetap menjaga nilai-nilai logika dan kebenaran.

BAB 4: Diskusi Etika Penggunaan AI di Masyarakat

Tujuan Pembelajaran

Peserta didik mampu menganalisis dampak etika dari penggunaan AI dalam skala sosial yang lebih luas dan mengembangkan pemikiran kritis terhadap isu-isu etika AI.

Peta Konsep



Apersepsi

Kalian pasti sering bertemu AI di kehidupan sehari-hari, tanpa disadari. Misalnya:

- Saat buka TikTok, muncul video yang kalian suka.
- Kamera di bandara bisa mengenali wajah penumpang.
- Chat otomatis yang bantu kita memilih kata saat kirim pesan.

Itu semua adalah hasil kerja AI (kecerdasan buatan).
Tapi... apakah AI **selalu baik**?

Coba pikirkan:

- Bagaimana kalau data pribadi kita dipakai tanpa izin?
- Bagaimana jika AI membuat keputusan yang tidak adil?
- Bagaimana jika AI justru dipakai untuk merugikan orang lain?

Di bab ini, kita akan **menantang diri** untuk berpikir lebih kritis:

- Apa saja dampak etika dari AI?
- Bagaimana agar AI digunakan secara adil dan aman?
- Dan bagaimana kita sebagai generasi muda bisa bersikap bijak?

Yuk, kita jelajahi bersama!
Belajar etika AI bukan hanya soal teknologi—tapi soal masa depan kita semua.
Siap berpikir? Siap berdiskusi? Let's go! 🚀

Penjelasan Konsep (Teori)

Apa Itu Etika AI?

Etika AI adalah cabang ilmu yang mempelajari bagaimana kecerdasan buatan (AI) sebaiknya dirancang dan digunakan agar tidak merugikan siapa pun. Tujuan utamanya adalah memastikan AI bisa digunakan secara adil, aman, dan bertanggung jawab.

> Bayangkan AI seperti robot pembantu pintar. Ia bisa membantu kita banyak hal, tapi... apakah kita sudah yakin ia bertindak adil untuk semua orang?

Beberapa pertanyaan penting yang perlu kita pikirkan:

- Apakah AI memperlakukan semua orang dengan adil?
- Apakah data pribadi kita digunakan dengan izin?
- Apakah AI bebas dari diskriminasi terhadap ras, gender, atau kelompok tertentu?

Jika etika tidak diterapkan, AI bisa menyebabkan **masalah serius** yang merugikan masyarakat. Yuk, kita bahas satu per satu!

Isu-Isu Etika dalam Penggunaan AI

1. Privasi dan Penggunaan Data Pribadi

> AI sering mengumpulkan data pribadi: lokasi, kebiasaan, bahkan wajah kita! Jika tidak dijaga dengan baik, data itu bisa:
>
> - Dipakai tanpa izin
> - Dijual ke pihak lain
> - Digunakan untuk mengawasi kita tanpa kita tahu
>
> Contoh nyata: Kamera pengenal wajah dipasang di tempat umum tanpa persetujuan. Meskipun tujuannya demi keamanan, ini bisa dianggap melanggar privasi.

2. Bias dan Diskriminasi dalam AI

AI belajar dari data yang diberikan manusia. Jika data itu tidak adil, AI juga bisa menjadi tidak adil.

Contohnya:

- Sistem rekrutmen AI lebih sering memilih pelamar dari kelompok tertentu.
- AI pengenal wajah lebih akurat mengenali ras tertentu, sehingga bisa menyebabkan salah tangkap!

> Tantangannya: Bagaimana agar AI tidak ikut membawa "ketidakadilan manusia"?

3. Siapa yang Bertanggung Jawab?

Kalau AI melakukan kesalahan, siapa yang harus disalahkan?

> Bayangkan mobil tanpa sopir menabrak orang. Siapa yang bertanggung jawab? Pembuat mobil, pemilik, atau pembuat programnya?

Ini adalah **masalah besar** dalam dunia AI. Banyak negara belum punya aturan jelas soal ini.

4. AI dan Dunia Kerja

AI bisa menggantikan pekerjaan manusia, terutama yang berulang.

Contohnya:

- Chatbot menggantikan customer service
- Robot industri menggantikan pekerja pabrik

Akibatnya, orang-orang yang tidak memiliki keterampilan teknologi bisa kehilangan pekerjaan.

> Tantangan kita: Bagaimana agar AI membantu manusia, bukan menggantikannya?

5. Transparansi (Explainability)

AI sering bekerja seperti "kotak hitam": kita tahu hasilnya, tapi tidak tahu **kenapa** hasil itu muncul.

Contoh:

- Pengajuan pinjaman ditolak oleh AI tanpa penjelasan.
- AI menuduh seseorang karena wajahnya mirip pelaku kejahatan—padahal salah!

Kita butuh AI yang bisa menjelaskan *mengapa* ia mengambil keputusan, agar kita bisa mengoreksi jika ada kesalahan.

6. Peraturan dan Kebijakan tentang AI

Teknologi AI seperti pisau: bisa untuk memasak, bisa juga melukai.

Oleh karena itu, diperlukan **aturan dan kebijakan yang jelas**. Tujuannya:

- Melindungi hak asasi manusia
- Menjaga keamanan data pribadi
- Mencegah diskriminasi

Contoh: **GDPR di Eropa** memberi hak kepada warga untuk mengetahui bagaimana data mereka digunakan oleh AI, bahkan bisa meminta data itu dihapus.

Selain itu, organisasi seperti PBB dan Uni Eropa sedang membuat aturan ketat untuk penggunaan AI di bidang berisiko tinggi, seperti senjata otomatis dan sistem hukum.

Kesimpulan: Mengapa Kita Harus Peduli?

AI adalah teknologi hebat, tapi seperti kekuatan besar lainnya, **harus digunakan dengan bijak**.

Sebagai generasi muda, kalian adalah calon pemimpin masa depan. Maka penting untuk:

- Mengerti manfaat dan risikonya
- Bertanya dan berpikir kritis
- Membantu menciptakan masa depan teknologi yang **adil, aman, dan bertanggung jawab**

Jika kamu jadi ilmuwan atau pembuat teknologi AI di masa depan, kamu mau AI-mu digunakan untuk apa?

🧠 Mari berpikir. 💬 Mari berdiskusi. 🌍 Mari bertindak.

Contoh Kasus atau Ilustrasi

Kasus 1: Si Cerdas yang Menyebalkan – AI Penilai Tugas

Bayangkan kamu mengumpulkan tugas esai Bahasa Indonesia, dan hasilnya dinilai oleh sistem AI. Kamu sudah yakin tulisanmu kreatif dan orisinal. Tapi… nilaimu jelek!

Ilustrasi:
AI menilai tulisanmu "tidak sesuai pola", padahal kamu pakai gaya menulis naratif yang unik. Kamu protes ke gurumu, tapi jawabannya, “AI-nya sudah menilai seperti itu.”

Tantangan berpikir:

- Haruskah AI dipakai untuk menilai semua tugas?
- Bisakah kreativitas dinilai oleh mesin?
- Siapa yang bertanggung jawab jika nilai tidak adil?

Kasus 2: Teman yang Terlalu Tahu – AI di Media Sosial

Setiap kali kamu buka TikTok atau YouTube, selalu muncul konten yang kamu sukai. Seru sih... tapi lama-lama kamu sadar, semua informasi yang kamu lihat cuma dari satu sisi.

Ilustrasi:
Kamu mulai merasa semua orang sependapat denganmu, padahal sebenarnya dunia itu luas. Kamu masuk ke “gelembung informasi”.

Tantangan berpikir:

- Apakah AI seharusnya hanya memberi konten yang kamu sukai?
- Bagaimana caranya keluar dari filter bubble?
- Apa bahaya kalau kita hanya melihat satu sisi?

Kasus 3: Nama yang “Kurang Keren” – AI di Dunia Kerja

Seorang kakak kelasmu melamar kerja ke perusahaan besar. Tapi dari ratusan pelamar, hanya beberapa dipanggil. Anehnya, semua yang lolos punya nama "kebarat-baratan".

Ilustrasi:
AI yang digunakan perusahaan dilatih dari data masa lalu, yang ternyata lebih sering memilih nama-nama seperti “Daniel” atau “Michelle”. Nama “Rizky” dan “Siti”? Terabaikan.

Tantangan berpikir:

- Apakah AI adil dalam menyeleksi pelamar?
- Haruskah sistemnya diubah?
- Siapa yang harus bertanggung jawab atas diskriminasi ini?

Kasus 4: Kamera Pintar di Sekolah

Di sekolahmu dipasang kamera baru yang katanya bisa mendeteksi wajah dan emosi siswa. Tapi tidak ada pemberitahuan sebelumnya.

Ilustrasi:
Kamu jadi merasa diawasi terus. Bahkan ketika sedih atau lelah, kamera menandainya sebagai "tidak fokus belajar".

Tantangan berpikir:

- Apakah pantas memasang kamera seperti itu tanpa izin?
- Apakah data wajah siswa aman?
- Bolehkah kamu menolak dipantau?

Kasus 5: Mobil Pintar, Masalah Serius

Bayangkan kamu sedang menyeberang jalan, dan sebuah mobil tanpa sopir (mobil AI) menabrak tembok karena menghindarimu. Tidak ada yang terluka, tapi mobilnya rusak parah.

Ilustrasi:
Perusahaan menyalahkan pemilik mobil karena "tidak memperbarui sistemnya". Pemilik bilang, "Saya cuma penumpang!" Lalu siapa yang salah?

Tantangan berpikir:

- Siapa yang bertanggung jawab saat AI membuat keputusan buruk?
- Haruskah AI memiliki "aturan moral"?
- Apakah teknologinya sudah siap digunakan di dunia nyata?

Kasus 6: Si Pembuat Lagu Misterius

Temanmu menunjukkan lagu keren yang katanya dia buat. Tapi ternyata, itu dari aplikasi AI yang bisa menciptakan lagu secara otomatis.

Ilustrasi:
Lagu itu viral. Tapi siapa yang berhak atas lagu itu? Temanmu? Aplikasinya? Atau pembuat kodenya?

Tantangan berpikir:

- Apakah hasil karya AI bisa dianggap karya manusia?
- Siapa pemilik hak cipta dalam kasus ini?
- Apakah kreativitas manusia akan tergeser?

Contoh Soal dan Pembahasan

Soal 1: Misteri Nilai dari AI

Situasi:
Tugas esaimu dinilai oleh sistem AI dan hasilnya ternyata rendah, padahal kamu merasa sudah menulis dengan kreatif.

Pertanyaan:
Apakah adil jika AI digunakan untuk menilai tugas siswa? Sebutkan satu keuntungan dan satu risiko etis dari penggunaan AI dalam sistem pendidikan.

Pembahasan:

- **Keuntungan:** AI bisa memberikan penilaian secara objektif berdasarkan kriteria tertentu dan menghindari favoritisme.
- **Risiko Etis:** AI belum tentu bisa memahami kreativitas atau konteks tulisan siswa yang unik, sehingga bisa memberi nilai yang tidak sesuai.

Pesan moral: Mesin pintar belum tentu paham perasaan atau niatmu saat menulis. Jangan biarkan kreativitasmu dikalahkan oleh kode!

Soal 2: Mobil Pintar Menabrak!

Situasi:
Sebuah mobil otonom menabrak trotoar untuk menghindari pejalan kaki. Tidak ada korban jiwa, tapi mobil rusak parah.

Pertanyaan:
Siapa yang seharusnya bertanggung jawab dalam kasus ini? Jelaskan alasanmu!

Pembahasan:
Beberapa kemungkinan

- **Perusahaan pembuat mobil:** Mereka merancang sistem AI, jadi tanggung jawab utama ada di mereka.
- **Pemilik mobil:** Jika tidak memperbarui sistem seperti yang diwajibkan, bisa ikut bertanggung jawab.
- **Pemerintah:** Jika belum ada regulasi jelas, berarti sistem hukum juga perlu disalahkan.

Pesan moral: Di dunia AI, pertanyaan "siapa yang salah?" bisa punya banyak jawaban!

Soal 3: Teman yang Terlalu Tahu

Situasi:
AI di media sosial selalu memberimu video atau berita yang kamu sukai, tapi akhirnya kamu hanya melihat satu sisi dari suatu isu.

Pertanyaan:
Apa bahaya dari fenomena “filter bubble”? Bagaimana cara kita bisa keluar dari jebakan ini?

Pembahasan:

- **Bahaya:** Kita jadi terkurung dalam pandangan sempit, sulit melihat sudut pandang lain. Ini bisa membuat kita cepat percaya hoaks atau intoleran.
- **Solusi:** Coba cari konten dari sumber berbeda, aktif berdiskusi, dan jangan cuma mengandalkan rekomendasi AI.

Pesan moral: AI pintar, tapi kamu harus lebih pintar dengan mencari kebenaran dari berbagai sisi!

Soal 4: AI dan Nama Pelamar

Situasi:
Seorang pelamar kerja ditolak karena sistem AI lebih sering memilih nama “barat” yang muncul di data lama.

Pertanyaan:
Apakah sistem AI seperti itu adil? Bagaimana seharusnya sistem rekrutmen AI diatur agar lebih etis?

Pembahasan:

- **Tidak adil:** Karena AI mewarisi bias dari data masa lalu, bisa terjadi diskriminasi tanpa disadari.
- **Solusi:** Data pelatihan harus diperiksa dan diperbaiki, melibatkan tim etika, dan tambahkan sistem pengawasan manusia.

Pesan moral: Masa lalu tak selalu adil. Tugas kita adalah menciptakan masa depan yang lebih adil—termasuk untuk AI.

Soal 5: Kamera Pengawas di Sekolah

Situasi:
Sekolahmu memasang kamera pengenal wajah tanpa pemberitahuan. Beberapa siswa merasa tidak nyaman.

Pertanyaan:
Apakah etis menggunakan teknologi pengenal wajah tanpa izin? Sebutkan satu risiko privasi dan satu langkah bijak yang bisa diambil.

Pembahasan:

- **Risiko:** Data wajah bisa disalahgunakan atau disimpan tanpa sepengetahuan siswa.
- **Langkah bijak:** Sekolah harus transparan, memberi sosialisasi, dan menawarkan pilihan bagi siswa untuk setuju atau tidak.

Pesan moral: Teknologi boleh canggih, tapi manusia tetap punya hak untuk tahu dan memilih.

Soal 6: Lagu dari Mesin

Situasi:
Temanmu membuat lagu keren... tapi ternyata dari aplikasi AI! Lagu itu viral!

Pertanyaan:
Siapa yang pantas mendapat penghargaan? Temanmu, aplikasinya, atau pembuat AI?

Pembahasan:

- Jawaban terbuka, bisa diperdebatkan:
    - **Temanmu:** Karena dia memilih, menyusun, dan menyebarkannya.
    - **Pembuat AI:** Karena mereka menciptakan sistem yang menghasilkan lagu.
    - **Solusi tengah:** Hak cipta dibagi atau dijelaskan secara transparan sesuai kebijakan yang disepakati.

Pesan moral: AI bisa mencipta, tapi apakah ia bisa "berhak cipta"? Di sinilah etika mulai beraksi!

Fakta Menarik / Fun Facts

1. **Tahukah kamu?** AI bisa belajar mengenali wajah dan emosi manusia hanya dari ekspresi wajah! Tapi, AI ini nggak bisa merasa sedih atau senang seperti manusia, lho—hanya "menebak" dari data.

2. Ketika kamu buka TikTok dan muncul video yang "pas banget", itu bukan sulap—itu kerja AI yang tahu apa yang kamu suka... tapi juga tahu *terlalu banyak* kalau tidak dijaga privasinya! 😅

3. **AI bisa menilai ribuan CV dalam hitungan detik**, tapi kalau datanya bias, bisa-bisa yang dipanggil kerja cuma orang dengan nama “keren” tertentu saja. Waduh, diskriminasi digital, nih!

4. **Mobil tanpa sopir (autonomous car) sudah nyata**! Tapi, kalau mobil AI itu menabrak, siapa yang harus ditilang? Mobilnya? Programernya? Atau pemiliknya? Hayo, siapa yang tanggung jawab?

5. **AI bisa bantu polisi mencari pelaku kejahatan**, tapi di beberapa kasus, AI malah menangkap orang yang salah karena wajahnya mirip. Ngeri, kan, kalau AI salah kenal wajah?

6. **Ada pekerjaan baru yang muncul di era AI: AI Ethicist!** Mereka adalah “penjaga moral” teknologi, memastikan AI tidak merugikan manusia. Siapa tahu, itu jadi pekerjaan impianmu kelak!

7. **Uni Eropa punya aturan khusus untuk AI bernama "AI Act"**, kayak undang-undang buat robot biar nggak seenaknya pakai data orang! Keren ya, robot juga harus taat aturan!

8. **Di universitas top dunia seperti MIT dan Oxford, etika AI sudah jadi mata kuliah serius!** Jadi, selain jago ngoding, sekarang anak IT juga harus jago mikir etis!

9. **AI bisa bantu dokter membuat diagnosis medis**, tapi bayangkan kalau AI salah baca data—bisa bikin salah obat! Itulah kenapa harus tetap ada pengawasan manusia.

10. **AI bisa bikin lukisan, nulis puisi, bahkan bikin lagu!** Tapi pertanyaannya: siapa yang harus dapat penghargaan? Mesinnya, atau orang yang memprogramnya?

Kalimat-kalimat ini bisa digunakan sebagai pembuka diskusi, selingan, atau kuis cepat di kelas.

Tips Belajar atau Tips Cepat Menghafal

Gunakan Mind Mapping

Peta pikiran membantu otak mengingat struktur dan hubungan antar konsep secara visual. Cocok untuk pelajar visual dan pemahaman holistik.

Merubah Materi Jadi Cerita / Kasus Nyata

Mengaitkan materi dengan contoh nyata seperti TikTok, rekrutmen kerja, atau mobil otonom membuat pelajaran lebih hidup dan membekas di memori.

Gunakan Akronim

Contoh akronim untuk mengingat isu utama Etika AI:
"PRiBaTaRe" =

- **P**rivasi
- **Ri**siko Bias
- **Ba**tas Tanggung Jawab
- **Transparansi**
- **Re**gulasi

Teknik Tanya Jawab (Q&A) Pribadi

Buat pertanyaan seolah kamu yang jadi AI:

- “Kalau aku salah pilih pelamar kerja, siapa yang disalahkan?”
- “Apakah aku netral? Atau aku belajar dari data yang berat sebelah?”

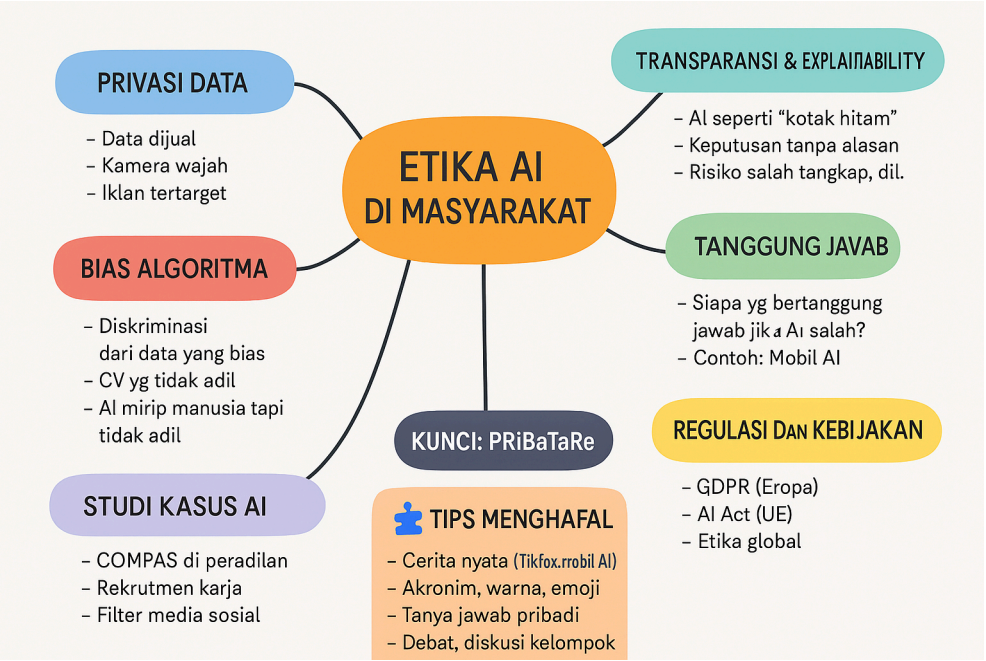
Gunakan Warna & Simbol

Gunakan **ikon**, **emoji**, dan **warna berbeda** di catatanmu untuk membedakan konsep.

Diskusi Kelompok & Debat

Berdiskusi akan memaksa otak untuk mengolah dan menyampaikan kembali pemahaman.

Mind Mapping Etika AI di Masyarakat



Contoh Kalimat Penguat Memori

- "AI tahu apa yang kamu suka... tapi belum tentu tahu apa yang adil."
- "Kalau AI bikin salah, siapa yang tanggung jawab? Bukan cuma mesinnya!"
- "Data masa lalu bisa jadi cermin masa depan... tapi bisa juga memperpanjang ketidakadilan."
- "AI cepat, tapi belum tentu tepat—apalagi soal etika!"

Aktivitas Siswa

Aktivitas 1: Drama Mini: "Ketika AI Membuat Keputusan"

Deskripsi Aktivitas:
Siswa membuat dan memainkan drama singkat tentang situasi sehari-hari yang melibatkan AI, seperti:

- AI menolak lamaran kerja karena nama pelamar.
- Mobil otonom menabrak orang.
- AI salah mengenali wajah seseorang di sekolah.

Langkah Pelaksanaan:

- Bagi siswa menjadi kelompok 4–6 orang.
- Minta setiap kelompok memilih satu isu etika AI.
- Mereka menyusun skenario berdurasi 3–5 menit, lalu memainkannya di depan kelas.
- Setelah tampil, kelompok lain bisa memberikan pertanyaan atau tanggapan etis.

Tujuan: Mengasah empati, pemikiran kritis, dan memahami dilema AI secara nyata dan seru.

Aktivitas 2: Game Kartu “Etis atau Tidak Etis?”

Deskripsi Aktivitas:
Siswa bermain game kartu yang berisi situasi penggunaan AI. Mereka harus menentukan apakah situasi itu **etis** atau **tidak etis**, dan **alasan logisnya**.

Langkah Pelaksanaan:

- Guru menyiapkan kartu dengan skenario pendek (misal: "AI menyarankan video radikal secara terus-menerus").
- Siswa bermain secara berkelompok, menarik satu kartu dan berdiskusi 2 menit sebelum menyatakan sikap.
- Kelompok lain bisa memberikan tanggapan atau tantangan terhadap argumen mereka.

Tujuan: Menumbuhkan daya analisis dan kemampuan debat secara menyenangkan.

Aktivitas 3: Tantangan Kreatif: Desain Poster “AI Bertanggung Jawab”

Deskripsi Aktivitas:
Siswa membuat **poster edukatif atau kampanye** berisi prinsip-prinsip etika AI yang menurut mereka penting bagi masa depan.

Langkah Pelaksanaan:

- Setiap siswa atau kelompok mendesain poster dengan tema “AI Bertanggung Jawab”.
- Poster bisa menggambarkan larangan, anjuran, atau solusi kreatif terhadap masalah AI.
- Hasilnya dipajang di kelas atau sekolah untuk menginspirasi yang lain.

Tujuan: Menumbuhkan tanggung jawab sosial dan kemampuan menyampaikan pesan etika dengan cara visual dan menarik.

Aktivitas 4: Diskusi Jalan Tengah: “Kalau Saya Jadi AI…”

Deskripsi Aktivitas:
Siswa berpura-pura sebagai AI yang harus mengambil keputusan sulit, lalu mereka berdiskusi mencari **jalan tengah yang adil**.

Langkah Pelaksanaan:

- Guru membacakan situasi etis (misalnya: "Seorang AI harus memilih siapa yang mendapat beasiswa").
- Siswa diminta memposisikan diri sebagai AI, lalu menjelaskan keputusan dan alasan mereka.
- Setelah itu, seluruh kelas mencoba mencari solusi kompromi yang lebih adil.

Tujuan: Membangun logika etis dan empati dalam mengambil keputusan, bukan sekadar benar-salah.

Aktivitas 5: Eksplorasi Mandiri: Jejak AI di Sekitar Kita

Deskripsi Aktivitas:
Siswa melakukan observasi atau mini riset tentang penggunaan AI di lingkungan sekitar mereka (rumah, sekolah, media sosial, dll).

Langkah Pelaksanaan:

- Ajak siswa mencatat selama 1–2 hari penggunaan AI yang mereka temui: rekomendasi YouTube, iklan personal, voice assistant, dsb.
- Di kelas, mereka saling berbagi temuan dan berdiskusi: “Apakah penggunaannya bijak?”, “Apakah ada resiko etikanya?”
- Bisa ditutup dengan membuat daftar “Tips Aman Berinteraksi dengan AI”.

Tujuan: Membuka mata siswa terhadap realitas AI dalam kehidupan mereka dan mendorong literasi digital yang etis.

Aktivitas 6: Simulasi Sidang Etika AI: “Siapa yang Bertanggung Jawab?”

Deskripsi Aktivitas:
Siswa memainkan simulasi pengadilan etika, di mana mereka harus menyelidiki siapa yang bertanggung jawab atas sebuah kesalahan AI.

Langkah Pelaksanaan:

- Guru memberikan kasus (misal: mobil otonom menabrak pejalan kaki).
- Siswa dibagi menjadi peran: pembuat AI, pemilik mobil, regulator, korban, pengacara.
- Masing-masing menyampaikan argumennya, dan "hakim kelas" memutuskan siapa yang paling bertanggung jawab.

Tujuan: Meningkatkan pemahaman tentang konsep tanggung jawab dalam dunia AI yang kompleks.

Rangkuman

Kita hidup di zaman ketika **kecerdasan buatan (AI)** sudah menjadi bagian dari kehidupan sehari-hari—dari video rekomendasi di TikTok hingga kamera pengenal wajah di bandara. Tapi, pernahkah kamu berpikir, apakah semua itu selalu membawa kebaikan? Bab ini mengajak kalian menyelami dunia **etika penggunaan AI**, yaitu bagaimana kita memastikan teknologi ini digunakan dengan cara yang **adil, aman, dan bertanggung jawab**. Kita akan belajar bahwa AI bukan sekadar mesin pintar, tapi bisa juga membawa masalah kalau tidak dikendalikan, seperti *pelanggaran privasi*, *bias algoritma*, atau *keputusan tanpa penjelasan*. Maka dari itu, kita harus belajar menjadi pengguna teknologi yang **kritis dan bijak**.

Bayangkan AI menolak lamaran kerja hanya karena data lama yang tidak adil, atau mobil tanpa sopir yang menabrak orang, tapi tidak ada yang tahu siapa yang harus bertanggung jawab. Inilah tantangan etika yang dibahas: bagaimana *AI* harus memperlakukan semua orang **tanpa diskriminasi**, dan siapa yang harus bertanggung jawab jika AI melakukan kesalahan. Kita juga akan membahas pentingnya **transparansi (explainability)**—AI harus bisa menjelaskan keputusannya, bukan hanya bilang "karena saya begitu diprogram." Selain itu, kalian juga akan mengenal **aturan dan kebijakan global**, seperti *GDPR* dan *AI Act*, yang dibuat agar AI tidak digunakan sembarangan. Semua ini menunjukkan bahwa *teknologi canggih harus tetap punya aturan main*.

Melalui berbagai **studi kasus menarik**, kalian akan diajak untuk berdiskusi: apakah adil jika hakim dibantu AI saat menjatuhkan hukuman? Atau jika media sosial hanya menampilkan konten yang sesuai minatmu, tapi membuatmu terjebak di *filter bubble*? Diskusi dan aktivitas seru seperti drama, debat, dan simulasi sidang akan membantu kalian melihat bahwa AI bukan hanya soal teknis, tapi juga soal **nilai, keadilan, dan kemanusiaan**. Jadi, yuk berpikir kritis dan berani bertanya: *"Apakah AI ini benar-benar adil? Siapa yang diuntungkan? Dan siapa yang mungkin dirugikan?"*

Latihan Soal

A. Benar atau Salah (5 Soal + Jawaban)

1. **AI dapat membuat keputusan yang benar secara etis tanpa campur tangan manusia.**
   Jawaban: Salah

2. **Penggunaan AI dalam sistem pengenal wajah selalu akurat dan bebas diskriminasi.**
   Jawaban: Salah

3. **Data yang digunakan untuk melatih AI harus bebas dari bias agar hasilnya adil.**
   Jawaban: Benar

4. **AI bisa mengakses data pribadi seseorang tanpa perlu izin.**
   Jawaban: Salah

5. **Transparansi dalam keputusan AI penting untuk mencegah kesalahan fatal.**
   Jawaban: Benar

B. Pilihan Ganda (10 Soal + Jawaban)

6. **Apa tujuan utama dari etika dalam pengembangan AI?**
   A. Membuat AI lebih cepat
   B. Mengurangi biaya produksi
   C. Menjamin AI digunakan secara adil dan tidak merugikan
   D. Menghilangkan semua bentuk teknologi lama
   Jawaban: C

7. **Risiko utama dari AI yang tidak mempertimbangkan etika adalah...**
   A. Biaya operasional tinggi
   B. Koneksi internet lambat
   C. Pelanggaran privasi dan diskriminasi
   D. Terlalu banyak data tersimpan
   Jawaban: C

8. **Sistem AI yang tidak menjelaskan cara kerjanya disebut...**
   A. Smart machine
   B. Open box
   C. Closed source
   D. Black box
   Jawaban: D

9. **Apa yang dimaksud dengan bias algoritma?**
   A. Kesalahan teknis dalam sistem AI
   B. Pola pengambilan data yang acak
   C. Ketidakadilan dalam hasil AI karena data yang tidak netral
   D. AI yang terlalu banyak belajar
   Jawaban: C

10. **Jika AI salah menilai seseorang dalam sistem hukum, maka yang perlu dipertanyakan adalah...**
    A. Harga AI
    B. Etika dan keadilan dari sistem tersebut
    C. Kecepatan proses hukum
    D. Komputer yang digunakan
    Jawaban: B

11. **Mengapa penting ada regulasi AI?**
    A. Agar AI lebih laku di pasar
    B. Untuk mencegah pencurian listrik
    C. Untuk melindungi hak asasi manusia dan keamanan data
    D. Supaya robot bisa bekerja di semua tempat
    Jawaban: C

12. **Filter bubble di media sosial bisa menyebabkan...**
    A. Kecepatan koneksi internet
    B. Keragaman pendapat
    C. Keterbatasan sudut pandang dan informasi
    D. Semakin banyak konten viral
    Jawaban: C

13. **GDPR di Eropa adalah contoh dari...**
    A. Aplikasi media sosial
    B. Sistem AI pengenal suara
    C. Aturan perlindungan data pribadi
    D. Platform belajar daring
    Jawaban: C

14. **Salah satu masalah dari AI dalam rekrutmen kerja adalah...**
    A. Tidak bisa membaca tulisan tangan
    B. Bisa menilai berdasarkan data yang bias
    C. Menolak semua pelamar
    D. Terlalu banyak wawancara dilakukan
    Jawaban: B

15. **AI yang digunakan tanpa penjelasan atau tanggung jawab dapat menyebabkan...**
    A. Efisiensi maksimal
    B. Penurunan biaya
    C. Ketidakpercayaan masyarakat
    D. Semangat belajar meningkat
    Jawaban: C

C. Isian Singkat (5 Soal + Jawaban)

16. **Sebutkan satu contoh risiko etika jika data pribadi digunakan tanpa izin!**
    Jawaban: Data bisa disalahgunakan untuk iklan tertarget atau dijual ke pihak ketiga tanpa sepengetahuan pengguna.

17. **Apa istilah untuk sistem AI yang tidak bisa menjelaskan alasannya dalam mengambil keputusan?**
    Jawaban: Black box

18. **Siapa yang seharusnya bertanggung jawab jika AI menimbulkan kerugian?**
    Jawaban: Pihak pengembang, perusahaan pengguna, atau regulator tergantung

konteks.

19. **Tuliskan satu akibat dari penggunaan AI dalam dunia kerja!**
    Jawaban: Beberapa pekerjaan manusia tergantikan oleh mesin atau chatbot, sehingga terjadi pengurangan tenaga kerja.

20. **Mengapa bias dalam data pelatihan AI bisa berdampak negatif pada hasil keputusan?**
    Jawaban: Karena AI akan membuat keputusan yang tidak adil atau diskriminatif berdasarkan data yang tidak seimbang.

D. Soal Explorasi (3 Soal + Contoh Jawaban)

Soal:

Situasi: Kamu menjadi tim pengembang AI rekrutmen dan menemukan sistem menolak pelamar dari daerah tertentu.
Jawaban:
Langkah pertama, saya akan mengevaluasi ulang data pelatihan AI untuk melihat apakah memang terdapat bias geografis. Lalu, saya akan melibatkan tim etika atau pihak independen untuk mengaudit sistem secara menyeluruh. Kami juga perlu menambahkan data dari berbagai wilayah agar sistem menjadi lebih inklusif dan adil. Keterlibatan pihak luar penting untuk menjaga transparansi dan kepercayaan publik terhadap sistem ini.

Soal:

Situasi: AI digunakan untuk menilai tugas siswa, tapi hasilnya dirasa tidak adil.
Jawaban:
Saya akan mengusulkan agar penilaian AI dilengkapi dengan tinjauan guru, terutama untuk tugas esai atau jawaban kreatif. AI dapat membantu dalam penilaian teknis, tetapi konteks dan ekspresi siswa tetap perlu dipertimbangkan oleh manusia. Selain itu, pihak sekolah harus menjelaskan cara kerja sistem AI dan memberi kesempatan siswa untuk menanggapi jika merasa nilai tidak sesuai.

Soal:

Situasi: Kamera pengenal wajah dipasang di sekolah tanpa pemberitahuan.
Jawaban:
Kepada Yth. Kepala Sekolah,
Saya merasa tidak nyaman dengan pemasangan kamera pengenal wajah di lingkungan sekolah tanpa adanya sosialisasi terlebih dahulu. Sebagai siswa, saya ingin tahu untuk apa data wajah kami digunakan, siapa yang mengelola, dan apakah ada jaminan data tidak disalahgunakan. Saya meminta agar sekolah menghormati hak privasi siswa dan melibatkan kami dalam diskusi tentang kebijakan ini, termasuk memberikan pilihan untuk menolak jika tidak setuju. Terima kasih atas perhatiannya.

Tugas Proyek

Proyek 1: Simulasi Sidang Etika “AI Menabrak Siapa?”

Tema: Tanggung jawab dan keadilan dalam keputusan AI.
Deskripsi:
Siswa akan mensimulasikan sidang untuk kasus mobil otonom yang menabrak orang atau properti. Mereka akan memerankan pembuat AI, pemilik mobil, regulator, korban, dan pengacara.
Langkah-langkah:

1. Guru membagikan skenario kecelakaan AI.
2. Siswa dibagi menjadi tim berdasarkan peran.
3. Setiap tim menyampaikan argumen dan bukti.
4. Diadakan sidang mini untuk menentukan siapa yang paling bertanggung jawab.

Tujuan: Melatih logika, empati, dan pemahaman tentang kompleksitas tanggung jawab AI.

Proyek 2: Poster Kampanye “AI Bertanggung Jawab”

Tema: Literasi etika dan komunikasi visual.
Deskripsi:
Siswa membuat poster edukatif bertema tanggung jawab dalam penggunaan AI, seperti: "AI Jangan Diskriminatif!" atau "Privasi Adalah Hak Kami."
Langkah-langkah:

1. Setiap siswa atau kelompok memilih satu isu etika AI (privasi, bias, transparansi, dll).
2. Mereka mencari data dan contoh kasus nyata.
3. Mendesain poster secara kreatif dan informatif.
4. Poster dipajang di kelas/sekolah.

Tujuan: Mendorong literasi etika dan kemampuan menyampaikan gagasan dalam bentuk visual yang menarik.

Proyek 3: Investigasi Mini: “AI di Sekitarku”

Tema: Eksplorasi mandiri & observasi etis.
Deskripsi:
Siswa diminta mencatat penggunaan AI yang mereka temui dalam kehidupan sehari-hari selama 2–3 hari (media sosial, toko online, asisten virtual, dll).
Langkah-langkah:

1. Siswa membuat jurnal sederhana berisi:
    - Di mana AI digunakan?
    - Apa manfaatnya?

    - Apa potensi resikonya?
2. Presentasi hasil di kelas.
3. Diskusi: Apakah penggunaannya bijak? Etis?

Tujuan: Menghubungkan teori dengan kenyataan, melatih kepekaan etis terhadap teknologi di sekitar mereka.

Proyek 4: Drama Mini: “Kalau AI yang Menilai”

Tema: Kreativitas dan empati terhadap keputusan AI.
Deskripsi:
Siswa membuat drama pendek berdasarkan isu etika AI, seperti AI yang salah memberi nilai atau diskriminatif dalam rekrutmen kerja.
Langkah-langkah:

1. Siswa dibagi dalam kelompok kecil (4–6 orang).
2. Pilih satu kasus nyata dari materi (misal: AI menolak lamaran karena nama).
3. Tulis naskah pendek (3–5 menit).
4. Tampilkan di depan kelas, dilanjutkan dengan diskusi kelas.

Tujuan: Mengasah daya kritis dan pemahaman etika melalui ekspresi kreatif.

Proyek 5: Desain Infografik “PRiBaTaRe”

Tema: Visualisasi konsep etika AI.
Deskripsi:
Siswa merancang infografik tentang 5 isu utama etika AI berdasarkan akronim **PRiBaTaRe** (Privasi, Risiko Bias, Tanggung Jawab, Transparansi, Regulasi).
Langkah-langkah:

1. Guru menjelaskan konsep PRiBaTaRe.
2. Siswa memilih satu atau dua aspek untuk dibuat infografik.
3. Infografik bisa dibuat digital atau manual.
4. Presentasi singkat di kelas.

Tujuan: Mempermudah siswa mengingat dan menjelaskan konsep etika AI secara visual dan sistematis.

Proyek 6: Tantangan Menulis Opini “Apakah AI Adil?”

Tema: Berpikir kritis dan ekspresi ide pribadi.
Deskripsi:
Siswa menulis artikel pendek (250–400 kata) mengenai satu isu etika AI yang paling menarik bagi mereka.
Langkah-langkah:

1. Pilih isu (misal: bias dalam rekrutmen, kamera pengenal wajah, dll).
2. Riset kasus nyata.
3. Tulis artikel dengan struktur: deskripsi isu – dampak – solusi menurutmu.

4. Artikel bisa dibaca bergilir di kelas.

Tujuan: Melatih argumen, opini, dan empati sosial dalam bentuk tertulis.

BAB 5: Dokumentasi Proyek dengan ChatGPT

Tujuan Pembelajaran

Peserta didik mampu membuat dokumentasi proyek aplikasi AI yang baik dan komprehensif dengan bantuan tools AI generatif.

Peta Konsep



Apersepsi

Kalian pasti sudah bekerja keras membuat aplikasi yang keren, kan? Mungkin kalian sampai harus menghabiskan waktu berhari-hari menulis kode, merancang tampilan, dan menguji fitur-fiturnya. Tapi coba pikirkan sejenak, jika teman kalian ingin menggunakan atau mengembangkan aplikasi yang kalian buat, apakah mereka langsung mengerti cara kerjanya?

Pernahkah kalian berpikir, cukupkah hanya dengan memberikan file kodingan saja? Atau apakah penting menjelaskan cara kerja aplikasi kalian? Dokumentasi adalah kunci penting untuk memastikan aplikasi kalian bisa digunakan dan dikembangkan lebih lanjut oleh orang lain di masa depan. Jadi, apakah kalian siap untuk belajar cara membuat dokumentasi proyek yang baik dan lengkap dengan bantuan AI?

Penjelasan Konsep (Teori)

Bayangkan kamu sedang membuat aplikasi keren—seperti alat absensi digital atau game edukatif—lalu beberapa bulan kemudian kamu ingin memperbaikinya, tapi lupa cara kerjanya. Atau, temanmu mau bantu mengembangkan aplikasi itu, tapi dia bingung harus mulai dari mana. Nah, disinilah **dokumentasi proyek** berperan penting! Dokumentasi ibarat **peta dan buku panduan** dari aplikasi yang kamu buat.

Yuk, kita bahas kenapa dokumentasi sangat penting dan jenis-jenisnya!

Mengapa Dokumentasi itu Penting?



1. **Sebagai Alat Komunikasi**
    - Dokumentasi membantu menjelaskan bagaimana aplikasi bekerja.
    - Orang lain bisa tahu fitur apa saja yang ada dan mengapa aplikasi dibuat dengan cara tertentu.
    - Ini penting untuk teman tim, guru, atau siapa pun yang ingin memahami aplikasimu dengan cepat.
2. **Untuk Perawatan dan Pengembangan**
    - Suatu saat, aplikasi bisa butuh update, perbaikan bug, atau tambahan fitur baru.
    - Dokumentasi membuat proses ini lebih mudah karena semua catatan teknis sudah tersedia.
    - Jadi, kamu tidak perlu mengingat-ingat semuanya dari awal!
3. **Mempermudah Kerja Tim**
    - Kalau kamu bekerja dalam tim, dokumentasi jadi seperti “bahasa bersama”.
    - Semua anggota tim bisa cepat memahami alur aplikasi tanpa harus tanya terus-menerus.
4. **Meningkatkan Nilai Portofolio**
    - Dokumentasi membuat hasil kerjamu terlihat profesional.
    - Ini bisa jadi nilai tambah saat kamu mendaftar lomba, magang, atau sekolah lanjutan.

Jenis-Jenis Dokumentasi Proyek Aplikasi

Dokumentasi itu tidak hanya satu jenis, lho! Berikut beberapa jenis penting yang biasa digunakan dalam proyek aplikasi:

1. **User Manual (Panduan Pengguna)**
    - Berisi langkah-langkah menggunakan aplikasi, seperti instalasi, login, dan penggunaan fitur.
    - Cocok untuk pengguna biasa agar mereka tidak bingung saat memakai aplikasi.
2. **Developer Documentation (Panduan Pengembang)**
    - Ditujukan untuk programmer yang ingin memahami kode aplikasi.
    - Biasanya mencakup struktur folder, fungsi penting, dan alur logika.
3. **Feature List (Daftar Fitur)**
    - Daftar fitur utama dengan penjelasan singkat, contohnya:
        - *Login dengan Wajah* → Pengguna masuk tanpa password.
        - *Riwayat Transaksi* → Menampilkan data transaksi sebelumnya.
4. **Project Overview (Gambaran Umum Proyek)**
    - Menjelaskan tujuan, latar belakang, masalah yang diselesaikan, siapa penggunanya, dan manfaat nyata aplikasimu.

Peran AI (seperti ChatGPT) dalam Membantu Dokumentasi

Zaman sekarang, kamu bisa minta bantuan AI untuk membuat dokumentasi jadi lebih cepat dan rapi. Tapi ingat, manusia tetap harus memeriksa dan menyempurnakannya. Ini beberapa hal keren yang bisa dilakukan AI:



- **Membuat Draf Awal**
  Cukup beri informasi singkat, AI bisa bantu menyusun draft dokumentasi.
- **Menjelaskan Kode**
  AI bisa menjelaskan bagian kode seperti perulangan atau kondisi *if-else* secara sederhana.
- **Menyusun Struktur Dokumentasi**
  AI dapat membantu membuat kerangka dokumentasi dari awal sampai akhir, agar urut dan rapi.
- **Mengecek dan Merapikan Teks**
  AI bisa menjadi editor otomatis: memeriksa tata bahasa, menyederhanakan kalimat, dan memperbaiki struktur tulisan.

Strategi Kerja Sama Manusia dan AI (Human-in-the-Loop)

Meskipun AI sangat membantu, kamu tetap harus:

- Mengecek isi dokumen yang dibuat AI.
- Memastikan tidak ada kesalahan data atau penjelasan.
- Menyesuaikan gaya penulisan agar cocok untuk tugas atau pembaca yang dituju.

Kesimpulan:
Dokumentasi bukan sekadar pelengkap, tapi bagian penting dari setiap proyek aplikasi. Dengan dokumentasi yang baik, aplikasi kamu bisa lebih mudah digunakan, dikembangkan, dan bahkan meningkatkan nilai kerja kamu di mata orang lain! Dan serunya, kamu bisa belajar sambil dibantu AI—keren, kan?

Contoh Kasus atau Ilustrasi

Kasus 1: AI Bantu Merapikan Dokumentasi Aplikasi “Kalkulator BMI”

Ilustrasi:
Kamu dan tim membuat aplikasi sederhana bernama "Kalkulator BMI" yang menghitung apakah berat badan seseorang tergolong normal, kurus, atau berlebih.

Kamu lalu mencoba menulis dokumentasi sendiri. Tapi hasilnya terasa agak membingungkan dan berbelit. Nah, kamu coba masukkan draf penjelasanmu ke ChatGPT dan bilang:

> “Tolong bantu perbaiki penjelasan tentang aplikasi Kalkulator BMI agar lebih rapi dan mudah dipahami siswa SMP.”

Hasilnya? Kalimatmu jadi lebih jelas dan runtut, tanpa mengubah inti dari aplikasi buatanmu. Ini contoh nyata bahwa AI bisa jadi editor yang keren!

Pesan penting: AI bisa bantu merapikan kata-kata, tapi tetap kamu yang mengontrol isi sebenarnya.

Kasus 2: Saat AI Salah Tebak Fitur Aplikasi

Ilustrasi:
Bayangkan kamu membuat aplikasi "Pencatat Jadwal Belajar", yang bisa menyimpan jadwal dan memberi pengingat.

Saat kamu minta bantuan AI menulis daftar fitur, AI menambahkan:

> “Fitur sinkronisasi otomatis dengan Google Calendar.”

Padahal aplikasi belum sejauh itu! Kamu langsung sadar dan memperbaikinya, menghapus fitur yang belum ada.

Pesan penting: AI bisa bantu menyusun, tapi kamu tetap harus **review** hasilnya — jangan langsung percaya 100%.

Kasus 3: Memberi Perintah Terlalu Umum ke AI

Ilustrasi:
Kamu bilang ke ChatGPT:

> “Buatkan dokumentasi aplikasi saya.”

AI lalu menebak-nebak dan memberi hasil yang kurang nyambung.

Lalu kamu coba lagi:

> “Saya membuat aplikasi pengingat minum air setiap 2 jam menggunakan App Inventor. Tolong bantu buat dokumentasinya.”

Tiba-tiba AI menghasilkan daftar fitur, panduan penggunaan, bahkan saran desain!

Pesan penting: Makin jelas dan lengkap kamu menjelaskan ke AI, makin bagus dan akurat hasilnya.

Kasus 4: Menjelaskan Logika Blok App Inventor lewat AI

Ilustrasi:
Dalam aplikasi “Kuis Matematika”, kamu pakai blok:

```
set ScoreLabel.Text to get global score
```

Kamu bingung gimana jelasin blok ini ke temanmu yang baru belajar coding.

Lalu kamu tanya ke ChatGPT:

> “Jelaskan fungsi blok ini dengan bahasa SMP.”

AI menjawab:

> “Blok ini memerintahkan aplikasi untuk menampilkan nilai skor ke layar. Jadi kalau kamu dapat poin, nilainya langsung muncul di layar.”

Temanmu pun langsung mengangguk paham.

Pesan penting: AI bisa bantu menjelaskan teknis ke dalam bahasa yang lebih mudah — cocok buat kerja tim!

Kasus 5: Kolaborasi AI dalam Dokumentasi “Aplikasi Deteksi Ekspresi Wajah”

Ilustrasi:
Timmu membuat aplikasi yang mendeteksi apakah seseorang sedang senang atau sedih lewat kamera.

Kamu butuh menulis dokumentasi awal. Dengan bantuan AI, kamu minta:

> “Buatkan dokumentasi awal untuk aplikasi deteksi ekspresi wajah berbasis kamera dan model Teachable Machine.”

AI langsung bantu membuat:

- Deskripsi aplikasi
- Daftar fitur (misalnya: real-time detection, notifikasi suara)
- Cara pakai aplikasi

Lalu kamu tinggal menyesuaikan sesuai versi final aplikasimu.

Pesan penting: AI bisa jadi *starter kit*, tapi kamu tetap penyusunnya!

Kasus 6: Dokumentasi Bisa Jadi Senjata Rahasia di Masa Depan

Ilustrasi:
Setelah proyek selesai, kamu simpan dokumentasinya lengkap: judul, fitur, alur penggunaan, cara kerja AI.

Dua bulan kemudian, gurumu minta kamu presentasi ulang proyek. Untung kamu masih punya dokumentasi!

Lebih keren lagi, kamu upload ke GitHub dan cantumkan di CV. Orang-orang bisa langsung paham isi aplikasi, tanpa harus lihat kode.

Pesan penting: Dokumentasi bukan cuma untuk sekarang, tapi juga buat masa depan. Dan AI bisa jadi teman menulis terbaikmu.

Contoh Soal dan Pembahasan

Soal 1: Siapa Bosnya? AI atau Kamu?

Kamu menggunakan AI seperti ChatGPT untuk membantu menulis dokumentasi aplikasi yang kamu buat. Apakah kamu bisa langsung menyalin semua hasil dari AI tanpa memeriksanya? Jelaskan alasannya.

Pembahasan:
Tidak bisa! Walaupun AI sangat membantu, AI **bukan pengganti otak kita sendiri**. AI bisa salah mengerti, menebak informasi yang belum ada, atau memakai istilah yang kurang tepat. Misalnya, AI bisa menambahkan fitur “notifikasi email” padahal aplikasi belum punya fitur itu.
Kita harus tetap memeriksa fakta, kalimat, dan istilah teknis dalam hasil AI. AI adalah asisten, **tapi keputusan akhir tetap di tangan kita.**

Soal 2: Tantangan Prompt yang Jelas

Kamu ingin ChatGPT membantumu menulis dokumentasi aplikasi “Pengingat Minum Air”. Pilihlah kalimat permintaan (prompt) yang lebih efektif untuk mendapatkan hasil yang bagus:

A. "Tolong buatkan dokumentasi aplikasi saya."
B. "Saya membuat aplikasi pengingat minum air setiap 2 jam dengan notifikasi. Bisa bantu buat dokumentasi awalnya?"

Pembahasan:
Jawaban: **B.**
Permintaan yang lebih jelas dan spesifik akan memberi hasil yang lebih relevan dan akurat dari AI.
Kalimat A terlalu umum — AI akan menebak-nebak.
Kalimat B memberi **konteks fitur** (“notifikasi setiap 2 jam”), sehingga hasilnya bisa mencakup deskripsi, fitur, dan panduan penggunaan yang lebih tepat.

Soal 3: Detektif Dokumentasi

Dalam dokumentasi aplikasi yang dibantu AI, kamu menemukan bagian ini:

> “Fitur: sinkronisasi otomatis dengan Google Calendar.”

Padahal aplikasi belum bisa melakukan itu. Apa yang harus kamu lakukan? Mengapa?

Pembahasan:
Kamu harus **menghapus atau mengedit bagian tersebut**.
AI bisa saja salah menebak fitur, terutama jika prompt kamu kurang jelas. **Kamu adalah orang yang paling tahu tentang aplikasi buatanmu** — jadi kamu bertanggung jawab untuk memastikan dokumentasi mencerminkan kenyataan. Dokumentasi yang jujur dan akurat lebih berguna dan bisa dipercaya.

Soal 4: Bahasa AI ke Bahasa SMP

Kamu membuat aplikasi “Kuis Matematika” dengan blok:
`set ScoreLabel.Text to get global score`
Temanmu bingung membaca itu. Apa prompt terbaik yang bisa kamu berikan ke ChatGPT agar bisa menjelaskan blok tersebut dalam bahasa yang mudah dimengerti?

Pembahasan:
Prompt yang bisa kamu pakai:

> “Jelaskan fungsi blok `set ScoreLabel.Text to get global score` dengan bahasa yang cocok untuk siswa SMP.”

AI akan menjelaskan bahwa **blok itu berfungsi untuk menampilkan skor ke layar**, jadi pemain tahu berapa poin yang mereka dapat.

Gunakan AI untuk menyederhanakan istilah teknis dan **membantu kerja tim kamu jadi lebih kompak.**

Soal 5: Menyusun Dokumentasi Sederhana

Kamu membuat aplikasi “Pencatat Skor Permainan” dan ingin menyusun dokumentasinya. Bagian apa saja yang **wajib kamu masukkan** ke dalam dokumentasi awal, agar orang lain bisa memahami dan menggunakan aplikasimu dengan mudah?

Pembahasan:
Setidaknya, kamu perlu mencantumkan:

1. **Gambaran Umum Proyek** – apa tujuan aplikasi ini? (misalnya: mencatat skor Tim A dan Tim B secara real-time)
2. **Daftar Fitur** – misalnya: tombol tambah skor, tombol reset, label skor
3. **Cara Menggunakan Aplikasi** – langkah-langkah penggunaannya secara runtut dan mudah dipahami

Dokumentasi yang baik membuat proyekmu **mudah digunakan, dikembangkan, dan dipahami siapapun** — bahkan oleh dirimu sendiri beberapa bulan ke depan!

Soal 6: AI Bisa Jadi Editor, Tapi…

Kamu menulis kalimat:

> “Aplikasi ini buat bantu orang tahu ekspresi wajah mereka kayak senang atau sedih.”

Kamu ingin kalimat ini terdengar lebih profesional dalam dokumentasi. Apa yang bisa kamu lakukan dengan bantuan AI?

Pembahasan:
Gunakan prompt:

> Perbaiki bahasa dan buat kalimat ini lebih profesional.”

AI akan membantumu merapikan kalimat menjadi, misalnya:

> “Aplikasi ini membantu pengguna mengenali ekspresi wajah mereka, seperti senang atau sedih, secara otomatis.”

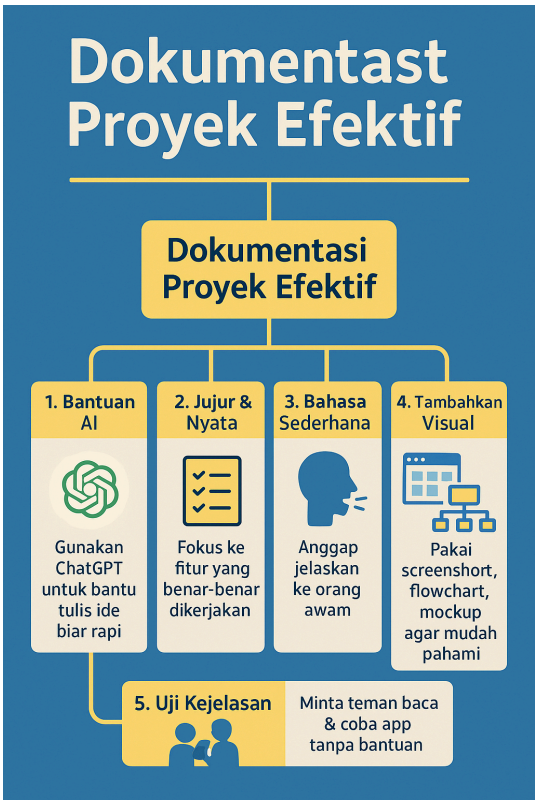
Dengan begitu, kamu belajar **menyusun kalimat yang jelas, formal, dan tetap komunikatif**. AI membantumu berkembang sebagai penulis teknis!

Fakta Menarik / Fun Facts

- Pernah baca kode sendiri tapi merasa seperti lagi baca tulisan alien? 😳 Tenang, itu normal banget! Makanya banyak programmer rajin nulis komentar biar nggak nyasar di masa depan!
- Komentar dalam kode itu seperti “pesan rahasia dari masa lalu” – ditulis oleh kamu hari ini, buat membantu kamu versi beberapa minggu kemudian
- Kode tanpa dokumentasi = jalan-jalan ke hutan tanpa kompas! Bahkan programmer jago bisa tersesat kalau tidak ada petunjuknya!
- Ada profesi keren bernama **Technical Writer**. Kerjanya? Bikin dokumentasi teknis biar programmer lain nggak pusing! Cocok banget buat kamu yang suka nulis dan teknologi!
- Proyek open source terkenal seperti React dan Linux bisa sukses besar bukan cuma karena kodenya keren, tapi karena dokumentasinya top markotop!
- Dokumentasi bagus bisa jadi **nilai plus** saat kamu lamar kerja nanti. Bisa bikin HR langsung ngerti proyekmu tanpa harus ngoding dulu!
- Dokumentasi = kunci rahasia agar proyek bisa dipakai dan dikembangkan orang lain. Tanpa itu, kodenya bakal jadi misteri abadi!

Tips Belajar atau Tips Cepat Menghafal

Mind Mapping Dokumentasi Proyek



Gunakan Singkatan Mnemonik

Coba ingat dengan singkatan:

"AI JUBeL VAU"
👉 *AI JUrnaL Bisa Lebih Visual dan AUtentik*

Huruf	Makna
A	**AI** bantu nulis ide dan penjelasan
J	**Jujur** dalam mencatat fitur yang benar-benar ada
B	**Bahasa** sederhana untuk orang awam
V	**Visual** bantu pemahaman (gambar, flowchart)
U	**Uji** kejelasan dengan orang lain

Contoh Narasi

"Saya membuat aplikasi To-Do List. Untuk menjelaskan fiturnya, saya dibantu oleh ChatGPT agar tulisannya lebih runtut dan tidak berbelit. Saya hanya menuliskan fitur yang sudah saya buat: tambah, ubah, dan hapus tugas. Supaya orang awam mudah paham, saya tidak menggunakan istilah teknis seperti *CRUD*. Saya juga menambahkan gambar tampilan aplikasi dan flowchart alur tugas. Setelah selesai, saya minta teman yang belum pernah lihat aplikasinya untuk mencoba hanya dari dokumentasi ini — dan dia bisa menjalankannya dengan lancar."

Aktivitas Siswa

Aktivitas 1: Tantangan "Menjelaskan Proyekku ke Orang Asing!"

Tujuan: Siswa berlatih membuat *Gambaran Umum Proyek* yang jelas dan menarik.

Langkah-langkah:

1. Bayangkan proyek aplikasi kalian akan diperkenalkan ke orang yang belum pernah melihatnya — bisa teman sekelas lain, guru lain, atau bahkan investor!
2. Diskusikan dalam kelompok: Apa tujuan aplikasi ini? Masalah apa yang coba diselesaikan? Siapa yang akan menggunakannya?
3. Gunakan ChatGPT untuk membantu membuat versi awal penjelasannya.

Prompt contoh: *"Bantu buat deskripsi proyek aplikasi absensi yang kami buat, dengan tujuan dan manfaatnya dalam 3 paragraf sederhana."*

4. Revisi hasilnya agar sesuai dengan kenyataan proyekmu, dan bacakan ke kelompok lain. Apakah mereka langsung paham?

Aktivitas 2: Misi "Pamer Fitur Keren!"

Tujuan: Siswa mengidentifikasi fitur utama aplikasi dan menjelaskannya secara jelas dan ringkas.

Langkah-langkah:

1. Buat daftar fitur utama dalam aplikasi kalian — tidak perlu semua, cukup yang paling berguna atau menarik.
2. Jelaskan tiap fitur dalam 1-2 kalimat. Gunakan bahasa awam, hindari istilah teknis.
3. Minta bantuan AI jika butuh menyusun kalimat yang lebih rapi.

   Prompt contoh: *"Bantu susun daftar fitur aplikasi to-do list kami agar mudah dipahami siswa SMP."*

4. Buat versi visualnya (misalnya infografik atau slide digital) agar fitur-fitur itu terlihat makin menarik!

Aktivitas 3: Simulasi "Jadi Pengguna Baru!"

Tujuan: Siswa menyusun panduan penggunaan aplikasi dari sudut pandang pengguna pertama kali.

Langkah-langkah:

1. Bayangkan kalian baru pertama kali membuka aplikasi kalian. Apa langkah pertama yang harus dilakukan? Lalu langkah kedua?
2. Tulis langkah-langkah menggunakan gaya instruksi: 1, 2, 3… dan pastikan jelas.
3. Uji coba: tukar panduan antar kelompok. Apakah kelompok lain bisa mengikuti dan berhasil memakai aplikasi hanya berdasarkan panduan tersebut?
4. Jika butuh bantuan memperbaiki bahasa, minta saran AI.

   Prompt contoh: *"Tolong buatkan panduan penggunaan sederhana untuk aplikasi deteksi ekspresi wajah yang kami buat."*

Aktivitas 4: Tantangan "Bahasa AI, Bahasa Kita!"

Tujuan: Siswa belajar menyunting dan menyempurnakan hasil dari AI agar sesuai dengan konteks proyek mereka.

Langkah-langkah:

1. Berikan informasi proyek ke AI dan lihat hasil draf dokumentasi awal yang dibuat.
2. Baca bersama kelompok. Apa saja yang akurat? Apa yang terasa salah atau terlalu ‘ngarang’?
3. Tandai bagian yang perlu dikoreksi, lalu diskusikan bagaimana memperbaikinya agar lebih sesuai dengan aplikasi kalian.
4. Ajak AI lagi untuk membantu menyempurnakan kalimat.

    Prompt contoh: *"Perbaiki kalimat ini agar lebih profesional: ‘Aplikasi ini buat bantu orang tahu ekspresi wajah mereka kayak senang atau sedih.’"*

Aktivitas 5: Misi Kolaborasi “Dokumentasi Super Tim”

Tujuan: Siswa mengembangkan kemampuan kolaboratif dengan me-review dokumentasi kelompok lain.

Langkah-langkah:

1. Tukarkan dokumentasi awal antar kelompok.
2. Baca dokumentasi kelompok lain sebagai “pengguna luar”.
3. Berikan komentar membangun: Apa yang sudah jelas? Apa yang masih membingungkan? Apakah kamu bisa membayangkan aplikasi tersebut hanya dari dokumentasinya?
4. Buat catatan revisi, lalu masing-masing kelompok memperbaiki dokumentasi berdasarkan masukan tersebut.

Aktivitas 6: Tantangan “AI = Asisten Interaktif”

Tujuan: Siswa belajar memaksimalkan peran AI untuk menyusun struktur dokumentasi yang lengkap dan rapi.

Langkah-langkah:

1. Diskusikan: bagian apa saja yang perlu ada dalam dokumentasi (Project Overview, Fitur, Cara Penggunaan, dll.).
2. Gunakan ChatGPT untuk menyusun kerangka dokumentasi yang sistematis.

    Prompt contoh: *"Tolong bantu buat kerangka lengkap dokumentasi aplikasi pengingat minum air berbasis notifikasi."*

3. Gunakan kerangka itu sebagai panduan menulis dokumentasi akhir proyek kalian.
4. Tambahkan elemen visual seperti flowchart, screenshot aplikasi, atau mindmap.

Aktivitas-aktivitas ini didesain untuk:

- **menantang logika dan empati siswa** (melihat proyek dari sudut pandang orang lain),
- **melatih komunikasi teknis dengan bantuan AI**, dan
- **membentuk kebiasaan berpikir dokumentatif dan kolaboratif**.

Rangkuman

Kalian sudah capek-capek bikin aplikasi keren—ngetik kode berhari-hari, mengatur tampilan, sampai uji coba terus-menerus. Tapi, coba bayangin kalau ada teman yang pengin pakai atau kembangkan aplikasi kalian, apakah mereka langsung ngerti cara kerjanya? Nah, disinilah pentingnya **dokumentasi proyek**! Dokumentasi itu ibarat **peta dan petunjuk ajaib** yang bikin aplikasi kalian lebih mudah dipahami, dipakai, dan dikembangkan. Tanpa dokumentasi, bisa-bisa aplikasi kalian malah jadi misteri! Dan kabar baiknya: sekarang kalian bisa minta bantuan *AI* seperti **ChatGPT** buat bantu bikin dokumentasi yang rapi, jelas, dan profesional.

Dokumentasi bukan cuma soal nulis panjang-panjang lho, tapi tentang **komunikasi, kolaborasi, dan pemeliharaan**. Misalnya, dokumentasi bisa jelaskan fitur-fitur utama, alur logika, atau bahkan alasan di balik pemilihan desain aplikasi. Saat aplikasi butuh perbaikan atau ditambah fitur baru, dokumentasi bikin semuanya lebih gampang. Bahkan kalau proyek kalian dikerjakan bareng teman-teman, dokumentasi bisa bikin kerja tim lebih lancar. Dan jangan lupakan satu hal penting: dokumentasi yang bagus bisa jadi **nilai plus untuk portofolio kalian**, apalagi kalau mau daftar magang, lomba, atau lanjut sekolah!

Yang paling seru, kalian bisa minta *AI* bantu bikin kerangka awal dokumentasi, menjelaskan kode rumit, atau menyusun daftar fitur secara sistematis. Tapi ingat: **AI cuma asisten, kalian tetap bosnya!** Kalian harus pastikan semua isi dokumentasi akurat dan sesuai dengan aplikasi sebenarnya. Gunakan *prompt* yang jelas, contoh: "Saya membuat aplikasi pengingat minum air setiap 2 jam dengan notifikasi. Bisa bantu buat dokumentasinya?" Bukan cuma lebih cepat, dokumentasi kalian juga bakal terlihat profesional dan siap digunakan siapapun — dari teman sekelas sampai calon investor. Yuk, tantang diri kalian untuk bikin dokumentasi yang nggak cuma informatif, tapi juga seru dan mudah dipahami!

Latihan Soal

A. Benar atau Salah (5 Soal)

1. Dokumentasi hanya diperlukan ketika aplikasi sudah selesai dibuat.
   Jawaban: Salah

2. Dokumentasi teknis ditujukan bagi pengguna umum agar mudah memahami fitur aplikasi.

Jawaban: Salah

3. ChatGPT bisa membantu membuat draf awal dokumentasi proyek aplikasi.
   Jawaban: Benar

4. Dokumentasi yang baik dapat membantu proses debugging di masa depan.
   Jawaban: Benar

5. Dalam dokumentasi, tidak perlu mencantumkan alasan pemilihan teknologi yang digunakan.
   Jawaban: Salah

B. Pilihan Ganda (10 Soal)

1. **Apa tujuan utama dari dokumentasi proyek?**
   A. Menyembunyikan logika aplikasi
   B. Menyulitkan orang lain memodifikasi kode
   C. Menjelaskan cara kerja aplikasi
   D. Mengganti tugas programmer
   Jawaban: C

2. **Berikut ini yang termasuk ke dalam dokumentasi pengguna adalah…**
   A. Penjabaran struktur kode
   B. Panduan instalasi dan login
   C. Alur logika program
   D. Penjelasan API
   Jawaban: B

3. **Dokumentasi berguna untuk mempermudah:**
   A. Penilaian oleh investor dan pengembang
   B. Menyembunyikan fitur utama
   C. Menghindari tanggung jawab
   D. Membuat kode jadi lebih cepat
   Jawaban: A

4. **Siapa yang paling bertanggung jawab memeriksa keakuratan dokumentasi?**
   A. AI
   B. Pengguna
   C. Pembimbing
   D. Pembuat aplikasi
   Jawaban: D

5. **Saat bekerja dalam tim, dokumentasi berguna untuk…**
   A. Menurunkan komunikasi antar anggota
   B. Menyatukan pemahaman
   C. Menyembunyikan alur kerja
   D. Menghindari dokumentasi

Jawaban: B

6. **Dalam dokumentasi teknis, sebaiknya memuat…**
   A. Daftar login pengguna
   B. Langkah-langkah debugging
   C. Penjelasan struktur folder dan fungsi
   D. Cuplikan video promosi
   Jawaban: C

7. **Apa manfaat mencantumkan versi teknologi dalam dokumentasi?**
   A. Untuk mempercepat loading aplikasi
   B. Agar dokumentasi terlihat panjang
   C. Supaya orang lain tahu spesifikasi sistem
   D. Untuk meningkatkan rating aplikasi
   Jawaban: C

8. **Dokumentasi proyek sebaiknya dimulai dengan…**
   A. Troubleshooting
   B. Daftar pustaka
   C. Gambaran Umum Proyek
   D. Kode program
   Jawaban: C

9. **Jika AI menambahkan fitur yang belum ada di aplikasi, kita harus…**
   A. Menambahkan saja di aplikasinya
   B. Mengabaikannya
   C. Menyesuaikan prompt dan mengedit hasilnya
   D. Langsung publikasikan
   Jawaban: C

10. **Kelebihan menggunakan AI dalam dokumentasi adalah…**
    A. Membuat presentasi
    B. Menghapus fitur yang tidak penting
    C. Membantu menyusun struktur dan merapikan kalimat
    D. Mengubah kode program
    Jawaban: C

C. Isian Singkat (5 Soal)

1. Dokumentasi proyek membantu _____ dan pengembangan aplikasi di masa depan.
   Jawaban: pemeliharaan

2. _____ adalah bagian dokumentasi yang ditujukan untuk programmer.
   Jawaban: Dokumentasi teknis

3. Daftar fitur dalam dokumentasi menjelaskan _____ dari aplikasi.
   Jawaban: kemampuan

4. AI seperti ChatGPT dapat membantu menyusun _____ dokumentasi yang rapi**.**
   Jawaban: struktur

5. Kita tetap harus melakukan _____ terhadap hasil dokumentasi dari AI.
   Jawaban: review

D. Soal Explorasi (3 Soal Paragraf Pendek)

1. **Susunlah gambaran umum proyek** untuk sebuah aplikasi “Pemantau Cuaca Sederhana” yang menggunakan API cuaca. Jelaskan tujuan aplikasi, masalah yang ingin diselesaikan, dan siapa penggunanya.
   Jawaban:
   Aplikasi "Pemantau Cuaca Sederhana" dibuat untuk membantu pengguna mengetahui kondisi cuaca harian secara real-time menggunakan data dari API cuaca terpercaya. Aplikasi ini menyelesaikan masalah sulitnya akses informasi cuaca yang akurat dan cepat, terutama bagi pengguna yang membutuhkan data cuaca untuk kegiatan sehari-hari seperti petani, pelajar, atau pekerja lapangan. Target pengguna aplikasi ini adalah masyarakat umum yang membutuhkan informasi cuaca harian dengan antarmuka yang sederhana dan mudah dipahami.

2. **Bayangkan kamu menggunakan ChatGPT untuk membuat dokumentasi aplikasi “Kalkulator BMI”.** Tuliskan permintaan (prompt) lengkap dan efektif yang akan kamu berikan ke AI agar hasil dokumentasinya sesuai kebutuhanmu.
   Jawaban:
   “Saya membuat aplikasi Kalkulator BMI dengan antarmuka sederhana menggunakan App Inventor. Aplikasi ini menghitung status berat badan berdasarkan input tinggi dan berat. Fitur utama: input tinggi badan, input berat badan, tombol hitung, dan hasil status BMI. Tolong bantu buatkan dokumentasi awalnya dengan gaya bahasa yang mudah dipahami pelajar SMP.”

3. **Kamu menemukan kalimat berikut dalam hasil dokumentasi AI:**
   “Aplikasi ini memiliki fitur pencadangan data otomatis ke cloud.”
   Padahal aplikasi belum memiliki fitur ini. Apa tindakan yang harus kamu lakukan, dan mengapa penting untuk melakukannya?
   Jawaban:
   Saya harus menghapus atau memperbaiki kalimat tersebut agar sesuai dengan fitur yang sebenarnya ada di aplikasi. Penting untuk melakukannya agar dokumentasi tetap akurat dan tidak menyesatkan pengguna atau pihak lain yang membaca. Dokumentasi yang tidak sesuai dapat menimbulkan ekspektasi yang salah dan merugikan reputasi pengembang.

Latihan soal ini dirancang untuk menguji pemahaman siswa secara menyeluruh — dari konsep dasar, penerapan praktis, hingga refleksi kritis terhadap penggunaan AI dalam dokumentasi proyek.

Tugas Proyek

Proyek 1: “Dokumentasi Aplikasi Absensi Digital”

Deskripsi Proyek:
Siswa membuat aplikasi absensi digital sederhana, lalu menyusun dokumentasi lengkapnya menggunakan bantuan AI seperti ChatGPT.

Langkah-langkah:

- Buat aplikasi absensi (misalnya dengan App Inventor, Python, atau Scratch).
- Identifikasi fitur utamanya: login, absen masuk/keluar, rekap kehadiran.
- Gunakan AI untuk menyusun gambaran umum, daftar fitur, dan panduan penggunaan.
- Tambahkan screenshot dan flowchart sederhana alur kerja aplikasi.
- Simulasikan presentasi ke “pengguna baru” (teman sekelas lain).

Tantangan seru:
Sampaikan semua informasi dalam bahasa yang mudah dimengerti oleh siswa lain, seolah kamu adalah “technical writer” profesional!

Proyek 2: “Misi Dokumentasi Kuis Matematika”

Deskripsi Proyek:
Kelompok membuat aplikasi kuis matematika dan mendokumentasikan fitur, logika skor, serta panduan penggunaannya.

Langkah-langkah:

- Buat kuis dengan fitur skor otomatis dan pembahasan jawaban.
- Tulis dokumentasi teknis: jelaskan logika blok, skor, dan aturan kuis.
- Gunakan ChatGPT untuk menjelaskan kode/blok dengan bahasa sederhana.
- Uji dokumentasi dengan meminta kelompok lain memainkan kuis hanya dari dokumen.

Tantangan seru:
Bisakah kelompok lain mendapatkan nilai tinggi hanya dengan membaca dokumentasi kalian?

Proyek 3: “Detektif Dokumentasi Proyek Teman”

Deskripsi Proyek:
Siswa bertukar dokumentasi dengan kelompok lain, lalu menilai kejelasan dan ketepatannya seperti seorang reviewer profesional.

Langkah-langkah:

- Tulis dokumentasi proyek kalian (gunakan bantuan AI jika perlu).
- Tukar dokumentasi dengan kelompok lain dan beri masukan jujur: mana yang membingungkan, mana yang keren?
- Koreksi dan perbaiki berdasarkan masukan teman.
- Buat versi final dokumentasi yang siap dipresentasikan.

Tantangan seru:
Jadilah tim "reviewer teknis" — kamu harus bisa memahami dan menjalankan aplikasi teman hanya dari dokumentasinya!

Proyek 4: "Dokumentasi Aplikasi Pengingat Minum Air"

Deskripsi Proyek:
Siswa menyusun dokumentasi aplikasi yang mengingatkan pengguna untuk minum air secara berkala — lalu mengubahnya menjadi panduan visual.

Langkah-langkah:

- Buat aplikasi sederhana berbasis notifikasi atau timer.
- Susun dokumentasi awal: deskripsi proyek, fitur, langkah penggunaan.
- Gunakan AI untuk merapikan kalimat dan menyusun FAQ.
- Tambahkan elemen visual: infografik fitur, flowchart waktu pengingat.

Tantangan seru:
Buat dokumentasi seolah-olah akan dibaca oleh seseorang yang baru pertama kali belajar teknologi!

Proyek 5: "Narasi Profesional dengan Bantuan AI"

Deskripsi Proyek:
Siswa menuliskan narasi proyek dengan gaya profesional, dibantu AI — mulai dari tujuan hingga manfaat aplikasinya.

Langkah-langkah:

- Buat ringkasan proyek secara singkat.
- Gunakan prompt ke ChatGPT seperti:
  "Bantu buat narasi proyek pencatat tugas harian kami dalam 3 paragraf profesional."
- Bandingkan hasil AI dengan narasi buatan sendiri.
- Revisi bersama untuk mendapatkan versi terbaik.
- Presentasikan narasi proyek sebagai pembukaan dokumentasi.

Tantangan seru:
Bisakah kamu membuat dokumentasi sekelas startup profesional hanya dari ide sederhana?

Proyek 6: “Kalkulator BMI: Dokumentasi Interaktif”

Deskripsi Proyek:
Siswa membuat dokumentasi proyek “Kalkulator BMI” dan menggabungkan fitur AI sebagai penjelas interaktif.

Langkah-langkah:

- Buat aplikasi kalkulator BMI (bisa dengan App Inventor atau Scratch).
- Minta AI menjelaskan fungsi setiap blok.
- Tulis dokumentasi berisi: deskripsi, fitur, panduan input, logika perhitungan.
- Tambahkan bagian “Bahasa AI ke Bahasa Kita” — yaitu penjelasan kode dalam versi siswa SMP.

Tantangan seru:
Buat dokumentasi yang bisa menjawab pertanyaan teknis seperti: “Kenapa saya mendapat status overweight?”

Catatan Penutup

Setiap proyek ini dirancang agar siswa:

- **Melatih berpikir sistematis dan empatik**
- **Mengasah kemampuan teknis dan komunikasi**
- **Belajar menjadi penulis dokumentasi yang profesional — dengan bantuan AI**

BAB 6: Presentasi Proyek Akhir

Tujuan Pembelajaran

Setelah mengikuti pembelajaran ini, peserta didik diharapkan mampu:

- Menyampaikan presentasi proyek akhir aplikasi AI dengan percaya diri dan profesional.
- Menjelaskan peran masing-masing anggota dalam tim.
- Merefleksikan proses kerja sama, tantangan, dan hasil yang telah dicapai.

Peta Konsep



Apersepsi

Selamat! Kalian telah menyelesaikan perjalanan panjang dalam mengembangkan aplikasi AI bersama tim. Sekarang saatnya untuk menunjukkan hasil kerja keras kalian! Lewat presentasi proyek akhir ini, kalian akan memamerkan:

- Apa yang sudah dipelajari.
- Apa yang sudah berhasil dibuat.
- Bagaimana kalian bekerja sebagai tim.

Ingat, ide yang hebat bukan hanya soal teknologi, tetapi juga bagaimana kita menyampaikannya dengan percaya diri!

Penjelasan Konsep (Teori)

Presentasi proyek akhir bukan hanya sekedar kegiatan formal yang dilakukan di akhir proyek, melainkan juga kesempatan untuk menunjukkan hasil kerja keras dan pembelajaran yang telah dilakukan. Ini adalah momen di mana kalian dapat berbagi pengalaman dan memberi inspirasi kepada orang lain. Presentasi yang baik tidak hanya memperkenalkan produk yang telah dibuat, tetapi juga menggambarkan proses dan tantangan yang dihadapi selama pengembangan. Berikut adalah tujuan dan elemen penting yang perlu diperhatikan dalam presentasi proyek akhir.

Tujuan Presentasi Proyek Akhir



1. **Showcase (Memamerkan Hasil Kerja)**
   Ini adalah saat yang tepat untuk memamerkan hasil kerja tim kalian. Tunjukkan aplikasi yang telah kalian buat, fitur-fitur unggulannya, tampilan, dan segala hal yang membedakan proyek kalian dari yang lain. Misalnya, kalian bisa menampilkan demo langsung aplikasi atau memberikan penjelasan singkat tentang fungsionalitas unggulan yang dimiliki.

2. **Edukasi (Memberi Penjelasan yang Mudah Dimengerti)**
   Jelaskan bagaimana aplikasi kalian bekerja dan apa manfaatnya bagi pengguna. Bayangkan kalian sedang menjelaskan aplikasi kalian kepada teman yang belum tahu apa-apa tentang aplikasi tersebut. Penjelasannya harus sederhana, mudah dipahami, dan menarik.
   Contoh:
   “Aplikasi ini memudahkan guru dalam mencatat absensi siswa. Dengan hanya menekan tombol ‘Absen Masuk’, data langsung tersimpan otomatis.”

3. **Refleksi (Menceritakan Proses dan Pembelajaran)**
   Ceritakan perjalanan tim kalian dalam mengembangkan aplikasi ini, termasuk tantangan yang dihadapi dan bagaimana cara tim mengatasinya. Misalnya, kalian bisa menceritakan saat mengalami kesulitan dalam pengkodean atau saat fitur

aplikasi tidak berjalan sesuai rencana. Proses ini adalah bagian yang penting karena menunjukkan bagaimana tim kalian berkembang dan belajar.
Contoh:
“Awalnya kami kesulitan menghubungkan aplikasi ke database MySQL. Tapi, setelah berdiskusi dan mencoba berbagai cara, akhirnya kami berhasil. Kami belajar banyak tentang pentingnya komunikasi dalam tim dan kesabaran saat debugging.”

4. **Portofolio (Memperlihatkan Karya untuk Masa Depan)**
   Presentasi ini juga bisa menjadi bagian dari portofolio pribadi. Portofolio adalah kumpulan karya terbaik yang bisa kalian tunjukkan saat melamar kuliah, magang, atau pekerjaan. Semakin jelas dan menarik cara kalian menjelaskan proyek ini, semakin besar pula peluang untuk dihargai orang lain.
   Contoh:
   “Proyek ini menunjukkan kemampuan saya dalam menggunakan Python, PyQt, dan MySQL dalam pengembangan aplikasi.”

Struktur Presentasi yang Efektif

Agar presentasi berjalan lancar dan menarik, penting untuk mengikuti struktur yang jelas. Berikut adalah panduan umum untuk membagi waktu dan isi presentasi proyek akhir:

- **Pembukaan (±10%)**
  Di bagian ini, sapa audiens dan perkenalkan tim kalian. Jelaskan secara singkat tujuan dan masalah yang ingin diselesaikan oleh aplikasi kalian.
  Contoh:
  “Halo! Kami adalah tim TechSavvy, terdiri dari Ana, Budi, dan Citra. Kami ingin memperkenalkan aplikasi SmartAttendance yang dapat membantu guru mencatat absensi siswa secara digital, lebih cepat dan akurat.”



- **Penjelasan Proyek (±30%)**
  Di sini, jelaskan aplikasi kalian dengan lebih mendalam, tetapi tetap menggunakan bahasa yang mudah dipahami. Jangan lupa untuk menyebutkan fitur utama dan manfaat aplikasinya.
  Poin-Poin yang Dapat Dijelaskan:
  - **Deskripsi Aplikasi:** Nama aplikasi dan fitur utama yang dimiliki.

    - **Penggunaan AI:** Penjelasan singkat tentang bagaimana teknologi AI digunakan dalam aplikasi.
    - P**roses Pengembangan:** Bagaimana ide muncul, perancangan tampilan, bahasa pemrograman yang digunakan, dan pengujian aplikasi.

  Contoh:
  “Kami menciptakan aplikasi SmartAttendance untuk mengatasi masalah absensi manual di sekolah. Fitur utamanya meliputi absen masuk/keluar, melihat riwayat absensi, dan fitur pengenalan suara dengan menggunakan Teachable Machine.”

- **Demonstrasi Langsung (±40%)**
  Tunjukkan aplikasi kalian langsung kepada audiens. Tampilkan cara login, navigasi menu, hingga penggunaan fitur utamanya. Fokus pada fitur yang paling menarik, seperti pengenalan wajah atau suara.
  Tujuan:
  Setelah demo, audiens harus merasa bahwa aplikasi ini benar-benar berfungsi dengan baik dan bermanfaat.

- **Pembagian Peran & Kontribusi Tim (±10%)**
  Jelaskan secara singkat peran masing-masing anggota tim dalam pengembangan aplikasi. Ini penting untuk menunjukkan bahwa proyek ini adalah hasil kerja sama tim.
  Contoh:
  “Saya bertanggung jawab atas desain tampilan aplikasi, Budi mengurus pengintegrasian AI, dan Citra bertugas dalam pengujian dan dokumentasi.”

- **Tantangan & Solusi (±5%)**
  Di bagian ini, ceritakan tantangan yang dihadapi selama pengembangan aplikasi dan bagaimana cara tim mengatasinya. Jangan lupa untuk menunjukkan pembelajaran yang kalian dapatkan dari proses tersebut.
  Contoh:
  “Kami sempat mengalami kesulitan dalam menghubungkan aplikasi dengan database, tetapi berkat kerja sama tim dan riset, akhirnya kami berhasil.”

- **Pengembangan Selanjutnya (±5%)**
  Sebutkan ide atau fitur baru yang ingin ditambahkan ke aplikasi di masa depan. Tunjukkan bahwa tim kalian berpikir jauh ke depan dan ingin terus mengembangkan aplikasi.
  Contoh:
  “Kami berencana menambahkan fitur suara interaktif dan meningkatkan kemampuan AI dalam mengenali ekspresi wajah siswa.”

- **Penutupan & Tanya Jawab**
  Akhiri presentasi dengan ringkasan singkat tentang aplikasi, terima kasih kepada pembimbing dan audiens, serta harapan kalian untuk masa depan aplikasi tersebut. Buka sesi tanya jawab untuk memberikan kesempatan bagi audiens bertanya atau memberi masukan.

Tips untuk Presentasi yang Menarik

1. **Visual Aid**
   Gunakan visual seperti screenshot aplikasi atau diagram alur kerja untuk membuat presentasi lebih menarik dan mudah dipahami. Hindari slide yang penuh dengan teks.
2. **Bahasa Tubuh & Kontak Mata**
   Berdirilah dengan percaya diri, tatap audiens, menggunakan gerakan tangan untuk menekankan poin-poin penting.
3. **Latihan**
   Latihan presentasi dengan tim, pastikan setiap anggota tahu perannya, dan waktu presentasi sesuai alokasi.
4. **Manajemen Waktu**
   Atur waktu presentasi dengan baik. Jangan terlalu lama di satu bagian dan pastikan sesi tanya jawab juga tidak terlewatkan.



Dengan mengikuti struktur yang jelas dan tips di atas, kalian dapat melakukan presentasi yang tidak hanya informatif, tetapi juga menarik dan menyenangkan!

Contoh Kasus atau Ilustrasi

Kasus 1: “Kacau Saat Demo, Tapi Tampil Percaya Diri”

Ilustrasi:
Kamu dan tim sudah menyiapkan aplikasi “Smart Parking Finder” yang bisa menunjukkan tempat parkir kosong di sekolah menggunakan AI. Tapi saat presentasi dimulai, *aplikasinya nge-lag* dan tidak menampilkan peta seperti biasa.

Tantangannya:
Apa yang harus kalian lakukan saat demo gagal tapi tetap ingin membuat audiens terkesan?

Tujuan Pembelajaran:
Menunjukkan pentingnya *rencana cadangan*, kemampuan improvisasi, dan tetap tenang di bawah tekanan. Mungkin kalian bisa berkata:

> “Sayangnya karena kendala jaringan, kami tidak bisa menunjukkan live demo. Tapi kami sudah siapkan video pendek yang merekam saat aplikasi ini berhasil digunakan minggu lalu!”

Kasus 2: “Anggota Tim Diam Saja”

Ilustrasi:
Tim kalian membuat aplikasi “FoodRescue” untuk menyelamatkan makanan sisa kantin agar bisa disumbangkan. Tapi saat presentasi, hanya satu orang yang bicara, sementara anggota tim lain hanya berdiri tanpa berkata apa-apa.

Tantangannya:
Bagaimana cara menunjukkan bahwa proyek ini adalah hasil kerja tim?

Tujuan Pembelajaran:
Menguatkan pentingnya *pembagian peran yang jelas dan tampil sebagai tim*. Coba sampaikan:

> “Saya, Rani, mendesain tampilan aplikasinya. Arya yang menghubungkan database, dan Dimas mengetes fitur reminder. Kami akan bergantian menjelaskan sesuai peran masing-masing.”

Kasus 3: “Slide Penuh Teks, Audiens Bingung”

Ilustrasi:
Tim kalian mempresentasikan aplikasi “ClassBuddy” untuk membantu siswa mengatur jadwal belajar. Tapi saat slide ditampilkan, isinya seperti buku pelajaran — padat, banyak tulisan, dan kecil-kecil!

Tantangannya:
Bagaimana membuat presentasi visual yang menarik, tidak membosankan?

Tujuan Pembelajaran:
Belajar *membuat slide efektif* yang mendukung, bukan membingungkan. Coba lakukan ini:

> “Daripada tulis semua isi di slide, kami hanya tampilkan poin penting, lalu kami ceritakan detailnya secara langsung. Ini membuat audiens lebih fokus ke penjelasan kami.”

Kasus 4: “Guru Bertanya Teknis, Siswa Bingung”

Ilustrasi:
Kalian membuat aplikasi “SpeakEasy” yang mengenali emosi lewat suara. Saat sesi tanya jawab, guru bertanya:

> “Bagaimana aplikasi kalian memproses data suara menjadi klasifikasi emosi?”

Tantangannya:
Bagaimana menjawab pertanyaan teknis dengan bahasa yang sederhana?

Tujuan Pembelajaran:
Mengembangkan kemampuan menjelaskan hal teknis secara *sederhana tapi tepat*. Contoh jawaban:

> "Kami melatih AI menggunakan suara yang sudah dikategorikan: marah, senang, sedih. Aplikasi mengenali nada dan energi suara, lalu membandingkannya dengan data yang sudah diajarkan sebelumnya."

Kasus 5: "Presentasi Biasa Saja, Tapi Aplikasi Hebat"

Ilustrasi:
Aplikasinya keren — bisa mendeteksi kualitas udara dari sensor sederhana, tapi saat presentasi, penyampaian terdengar datar, tidak semangat, dan hampir tidak ada kontak mata.

Tantangannya:
Apa yang bisa dilakukan agar presentasi terlihat *hidup dan menginspirasi*?

Tujuan Pembelajaran:
Memahami pentingnya *bahasa tubuh, ekspresi wajah, dan antusiasme*. Berlatih:

> "Kami sempat kaget saat data sensor menunjukkan kualitas udara di kantin lebih buruk dari lapangan sekolah. Dari situ, kami sadar pentingnya mengedukasi siswa tentang ventilasi dan udara sehat."

Kasus 6: "Waktu Habis, Bagian Penting Belum Tersampaikan"

Ilustrasi:
Saat presentasi aplikasi "LangLearn" yang membantu siswa belajar bahasa asing dengan flashcard AI, kalian terlalu lama menjelaskan latar belakang. Akibatnya, bagian demo hanya sebentar, padahal itu bagian paling menarik.

Tantangannya:
Bagaimana mengatur waktu presentasi agar semua bagian penting tersampaikan?

Tujuan Pembelajaran:
Melatih manajemen waktu saat presentasi. Gunakan pembagian seperti:

- 1 menit: Perkenalan tim dan masalah yang ingin diselesaikan
- 2 menit: Penjelasan fitur utama
- 3 menit: Demo aplikasi
- 1 menit: Refleksi & penutup

> "Karena waktu terbatas, kami langsung perlihatkan fitur utama: 'Scan & Translate'. Kami akan peragakan bagaimana aplikasi ini bisa menerjemahkan tulisan dari buku ke bahasa Jepang."

Setiap kasus di atas bisa dijadikan simulasi atau diskusi kelas:

“Kalau kamu ada di kasus ini, apa yang akan kamu lakukan?”

Cara ini *tidak terasa seperti tugas*, tapi tetap mengasah kreativitas, empati, kolaborasi, dan komunikasi siswa SMP dengan cara yang seru dan menantang.

Contoh Soal dan Pembahasan

Soal:

"Apa saja isi dari bagian pembukaan dalam presentasi proyek akhir? Mengapa bagian ini penting dan tidak boleh dilewatkan?"

Pembahasan:
Bagian pembukaan mencakup:

- Perkenalan tim dan anggota
- Nama dan tujuan aplikasi
- Masalah yang ingin diselesaikan

Penting karena:

- Memberi kesan pertama kepada audiens
- Membuat audiens paham latar belakang proyek
- Menjelaskan motivasi tim secara singkat dan menarik

Tips fun:
Bayangkan kalian sedang membuka acara TV favorit kalian — harus seru, jelas, dan bikin orang penasaran!

Soal:

"Kalau kamu jadi presenter dalam tim, mana yang paling penting untuk kamu tunjukkan saat demo aplikasi? Jelaskan alasannya!"

Pembahasan:
Fitur utama aplikasi adalah yang paling penting untuk ditunjukkan, misalnya:

- Fitur absensi otomatis
- Pengenalan suara atau wajah
- Proses input–output secara langsung

Kenapa?

- Audiens ingin bukti nyata bahwa aplikasi bekerja
- Demonstrasi langsung membantu pemahaman lebih cepat dan menimbulkan kesan “Wow!”

Contoh jawaban:
“Kalau saya, akan langsung menunjukkan fitur pengenalan suara. Karena ini yang paling beda dari absen biasa, dan memperlihatkan keunggulan AI yang kami pakai.”

Soal:

"Ceritakan bagaimana pembagian peran di tim kalian! Mengapa tidak semua orang mengerjakan hal yang sama?"

Pembahasan:
Jawaban ideal menjelaskan tugas tiap anggota, contohnya:

- “Alya membuat desain tampilan”
- “Rafi mengatur logika program”
- “Nabila melatih AI”
- “Doni menyiapkan presentasi dan dokumentasi”

Pentingnya pembagian peran:

- Efisien dan tepat sasaran
- Sesuai kemampuan anggota
- Melatih kerja sama dan tanggung jawab

Tantangan seru:
Simulasi kelompok: "Kalau hanya dua orang di timmu, bagaimana kalian membagi tugas dengan adil?"

Soal:

"Saat demo aplikasi tiba-tiba tidak bisa berjalan karena gangguan teknis. Apa yang bisa kalian lakukan agar presentasi tetap lancar dan menarik?"

Pembahasan:
Solusi yang tepat:

- Siapkan video demo cadangan
- Tunjukkan screenshot dan jelaskan alur penggunaannya
- Gunakan cerita saat uji coba sebelumnya

Contoh jawaban:
“Kami menyiapkan video pendek demo fitur utama aplikasi saat masih berjalan dengan baik, jadi walaupun live demo gagal, audiens tetap bisa melihat fungsinya.”

Catatan penting:
Jangan panik. Tunjukkan sikap tenang dan tetap profesional — itu juga nilai plus!

Soal:

"Kalau aplikasi kalian ingin dikembangkan lebih lanjut, fitur baru apa yang ingin kalian tambahkan? Jelaskan manfaatnya bagi pengguna!"

Pembahasan:
Jawaban harus kreatif dan relevan. Contoh:

- Fitur suara penanda sukses: “Selamat datang, Raka!”
- Statistik grafik kehadiran bulanan
- Terhubung ke WhatsApp guru atau orang tua

Manfaat:

- Menambah kenyamanan dan pengalaman pengguna
- Memperluas kegunaan aplikasi
- Menunjukkan semangat belajar dan inovasi tim

Soal:

"Kenapa menurutmu Steve Jobs sangat terkenal dalam presentasi produk Apple-nya? Apa yang bisa kamu pelajari dari gayanya?"

Pembahasan:
Steve Jobs dikenal karena:

- Cara menyampaikan ide teknologi secara sederhana dan memukau
- Antusiasme dan ekspresi wajahnya
- Penggunaan demonstrasi nyata, bukan cuma penjelasan teknis

Pelajaran:

- Sampaikan dengan percaya diri
- Gunakan cerita dan contoh
- Tunjukkan energi positif saat berbicara

Fakta Menarik / Fun Facts

Fun Facts Seputar Presentasi dan Teknologi

1. **Tahukah kamu?** Steve Jobs pernah berlatih **lebih dari 200 kali** sebelum presentasi peluncuran iPhone pertama. Jadi, kalau kamu deg-degan, itu wajar! Bahkan orang hebat pun butuh latihan!
2. **Otak manusia 22 kali lebih cepat** memproses gambar dibanding teks. Itu sebabnya, **slide penuh tulisan bikin ngantuk**, tapi satu gambar bisa langsung “nendang”!
3. Saat kamu presentasi dengan **senyuman**, otak audiens akan otomatis lebih tertarik mendengarkan. Jadi, senyum itu bukan cuma sopan, tapi juga strategi!

4. Presentasi yang baik itu seperti **cerita film**: ada pembuka yang bikin penasaran, isi yang seru, dan penutup yang bikin “wah!”
5. Riset menunjukkan, **manusia lebih mudah mengingat cerita** daripada angka. Makanya, **ceritakan proses bikin aplikasi kalian**—bukan cuma jelaskan fiturnya!

Fun Facts Seputar AI dan Aplikasi

6. Aplikasi yang kamu buat dengan **Teachable Machine** mungkin terlihat sederhana, tapi teknologinya sama seperti yang dipakai untuk **deteksi wajah di ponsel pintar!**
7. Kamu bisa bikin AI hanya dengan **rekaman suara atau foto**—kamu baru saja belajar hal yang dilakukan ilmuwan di Google dan Meta!
8. AI seperti yang kalian pakai juga digunakan untuk membuat **mobil bisa nyetir sendiri** atau **asisten virtual seperti Siri dan Google Assistant**. Keren banget, kan?
9. Setiap kali kamu membuat fitur baru di aplikasimu, kamu sedang berpikir seperti seorang **problem solver dunia nyata**—bukan cuma “main coding”!
10. Banyak startup besar seperti **Instagram dan WhatsApp** dulunya juga dimulai dari **tim kecil seperti kalian**—yang penting adalah idenya dan semangatnya!

Fun Facts Tentang Kerja Tim dan Proyek

11. Tim terbaik bukan yang paling jago semua, tapi yang **tahu cara bekerja sama**. Bahkan superhero pun perlu tim: lihat saja Avengers!
12. Saat kamu menyampaikan hasil proyekmu, kamu bukan cuma presentasi… kamu sedang belajar **cara meyakinkan dunia tentang ide brilian!**
13. Presentasi pertama mungkin grogi, tapi lama-lama kamu bisa ketagihan. Banyak orang sukses bilang: “**Panggung presentasi adalah tempat ide berubah jadi kenyataan**.”
14. Gagal saat demo? Jangan panik! Banyak aplikasi hebat lahir dari **kesalahan pertama yang memalukan**, lalu diperbaiki jadi luar biasa.

Kalimat-kalimat ini bisa disisipkan saat pembukaan, transisi antar bagian, atau sebagai “penyegar” di tengah presentasi

Tips Belajar atau Tips Cepat Menghafal

Mind Mapping Presentasi Proyek Akhir



Tips Cepat Belajar

Gunakan Akronim

Contoh untuk struktur presentasi:
"PE-DI-TI-PEN"
Pembukaan – Deskripsi – Demo – Tim – Penutup

Latihan dengan Cerita

Ubah isi menjadi alur cerita:“Kami melihat guru capek absen manual → muncullah ide bikin aplikasi SmartAttendance → kami desain UI → kami latih AI untuk pengenalan suara → kami uji → kami bagi tugas → sekarang kami tampilkan ke kalian!”

Kata Kunci Tiap Bagian

- **Pembukaan**: Nama, Tim, Masalah
- **Isi Proyek**: Fitur, Fungsi, AI
- **Demo**: Fitur unggulan LIVE
- **Tim**: Tugas per orang
- **Tantangan**: Kesulitan dan Solusi
- **Penutup**: Ucapan & Harapan

Tulis di sticky notes atau flashcard.

Teknik Gladi Resik

Belajar 25 menit (fokus pada satu bagian mind map) → Istirahat 5 menit → Ulang 4 kali. Setelah itu, uji diri sendiri: “Coba jelaskan bagian DEMO tanpa lihat catatan!”

Presentasi = Cerita + Visual

Bayangkan presentasimu adalah **cerita film**:

- Awal: Siapa kamu dan kenapa proyek ini ada
- Tengah: Bagaimana kamu bikin aplikasi ini?
- Akhir: Apa hasilnya, dan apa rencana ke depan?

Gunakan gambar, screenshot, dan video pendek.

Contoh Narasi Singkat

“Halo! Kami tim SmartDev, terdiri dari Nino, Dela, dan Raka. Kami membuat aplikasi SmartAttendance untuk membantu guru mencatat absen siswa secara digital. Aplikasi ini punya fitur absen masuk, laporan bulanan, dan pengenalan suara pakai AI. Awalnya kami kesulitan menghubungkan suara ke fitur AI, tapi setelah latihan dan diskusi, berhasil juga! Saya bikin UI-nya, Dela integrasi AI, dan Raka buat dokumentasi. Semoga aplikasi kami bisa dikembangkan lebih lanjut. Terima kasih!”

Aktivitas Siswa

Aktivitas 1: “Buka Ceritamu!” (Simulasi Pembukaan Presentasi yang Kuat)

Tujuan: Melatih siswa membuka presentasi dengan percaya diri, jelas, dan menarik.

Langkah Aktivitas:

- Setiap tim menyusun pembukaan presentasi seperti sedang memperkenalkan acara favorit mereka.
- Gunakan 3 komponen wajib: **(1) Perkenalan Tim**, **(2) Nama & Tujuan Proyek**, **(3) Masalah yang Diselesaikan**.
- Melatih gaya bicara dan ekspresi wajah agar pembukaan terasa *energik dan mengundang penasaran*.

Serunya:
Boleh menambahkan gaya pembukaan seperti di acara TV atau YouTube channel, tapi tetap sopan dan relevan.

Aktivitas 2: “Ayo, Ceritakan Proyekmu!” (Storytelling Proyek ala Tim Developer)

Tujuan: Mengubah penjelasan teknis menjadi cerita menarik yang mudah dipahami siapa pun.

Langkah Aktivitas:

- Uraikan cerita proyek dari awal ide hingga pengujian, seperti kisah petualangan.
- Gunakan pola: *Masalah → Ide → Desain → Coding → Uji Coba → Pelajaran yang Didapat*.
- Tambahkan “twist” atau tantangan nyata yang kalian hadapi.

Serunya:
Boleh menggunakan *alat bantu visual* seperti papan gambar, sketsa UI, atau bahkan membuat *alur cerita (comic strip)* tentang proyek kalian.

Aktivitas 3: “Coba, Tunjukkan Live-nya!” (Latihan Demo Aplikasi Interaktif)

Tujuan: Membiasakan siswa menunjukkan aplikasi secara langsung dan memikat audiens.

Langkah Aktivitas:

- Setiap tim melakukan latihan demonstrasi aplikasi dari awal login hingga fitur utama.
- Fokus pada fitur AI atau keunggulan unik: bisa suara, wajah, otomatisasi, dll.
- Siapkan **video cadangan** jika demo live gagal.

Serunya:
Buat sesi tantangan: “Coba fitur ini hanya dengan suara!”, atau “Siapa cepat klik, dia menang!” untuk melibatkan audiens saat demo.

Aktivitas 4: “Kami Satu Tim Hebat” (Showcase Kolaborasi Tim)

Tujuan: Menunjukkan bahwa setiap anggota tim punya peran penting dan saling melengkapi.

Langkah Aktivitas:

- Masing-masing anggota menjelaskan perannya dengan kalimat singkat dan jelas.
- Ceritakan proses kerja sama: apakah ada perbedaan pendapat? Bagaimana menyatukan ide?
- Bisa dibuat sebagai *mini talk show* tim, di mana satu jadi moderator, yang lain menjawab.

Serunya:
Coba bikin sesi simulasi: “Jika hanya berdua di tim, bagaimana tugas dibagi?” – untuk mengasah strategi kolaborasi.

Aktivitas 5: “Tantangan & Transformasi” (Refleksi Tim ala Developer Sejati)

Tujuan: Melatih refleksi jujur tentang tantangan dan solusi yang dialami selama proyek.

Langkah Aktivitas:

- Setiap anggota menceritakan satu tantangan nyata (teknis atau komunikasi) dan bagaimana mengatasinya.
- Gunakan format: *“Awalnya kami… lalu kami mencoba… akhirnya kami berhasil dengan…”*
- Bisa dikemas sebagai *podcast kelas*, video dokumentasi singkat, atau sesi tanya-jawab santai.

Serunya:
Bagikan pengalaman “gagal saat uji coba tapi tetap semangat” — ini justru bikin audiens kagum dan terinspirasi.

Aktivitas 6: “Lanjut atau Hancur?” (Visi Masa Depan Proyek)

Tujuan: Mendorong siswa berpikir kreatif untuk pengembangan lanjutan aplikasi mereka.

Langkah Aktivitas:

- Setiap tim membuat papan ide “Kalau lanjut, fitur apa yang akan ditambah?”
- Tunjukkan manfaat fitur itu dan tantangan teknis yang mungkin dihadapi.
- Presentasikan sebagai *pitch singkat*, seolah-olah tim kalian ingin meyakinkan investor untuk mendukung aplikasi ini.

Serunya:
Boleh pakai gambar, mockup, atau suara efek — tunjukkan kreativitas maksimal!

Catatan Umum untuk Guru:

- Dorong siswa tampil ekspresif dan percaya diri — bukan sekadar menghafal teks.
- Jadikan setiap aktivitas sebagai ajang eksplorasi, bukan penilaian kaku.
- Beri ruang untuk apresiasi antar tim agar suasana kelas mendukung dan membangun rasa percaya diri.

Rangkuman

Halo semua! Bayangkan kalian sedang jadi pembawa acara *tech show* seru. Nah, sekarang giliran tim kalian tampil! Mulailah dengan **memperkenalkan anggota tim**, lalu kenalkan **nama proyek dan tujuan aplikasi** kalian. Jangan lupa ceritakan **masalah yang ingin kalian selesaikan**—buat audiens langsung penasaran! Misalnya, “Kami dari tim NextGen Makers membuat *SmartAttendance*, aplikasi yang bikin absen siswa jadi otomatis, tanpa ribet!” Jadikan pembukaan kalian semenarik pembukaan vlog atau acara TV favorit kalian—**bikin semua mata tertuju pada kalian!**

Setelah audiens kenal dengan proyek kalian, saatnya masuk ke isi utama: **ceritakan perjalanan aplikasi kalian** seperti petualangan. Mulai dari ide awalnya muncul karena apa, desain aplikasinya seperti apa, lalu masuk ke bagian seru: **fitur utama**, seperti pengenalan suara pakai *Teachable Machine*, atau sistem login yang praktis. Tunjukkan juga **alat yang kalian gunakan**, seperti *MIT App Inventor*, dan bagaimana proses coding serta uji cobanya. Jangan takut untuk menyisipkan cerita “gagal tapi akhirnya berhasil”—itu justru bikin audiens kagum! Saat **demo aplikasi**, pastikan fitur unggulan tampil nyata dan lancar. Kalau pun ada kendala, siapkan *backup video* dan tetap percaya diri. Ini saatnya membuktikan: “Aplikasi kami bukan sekadar ide, tapi **bisa langsung dipakai**!”

Akhiri presentasi kalian dengan **refleksi dan rasa syukur**. Ceritakan apa yang kalian pelajari—misalnya, pentingnya kerja sama tim atau belajar dari kegagalan. Jelaskan juga **kontribusi tiap anggota tim** dengan bangga, seperti siapa yang membuat UI, siapa yang melatih AI, dan siapa yang mengatur presentasi. Lalu, buka sesi tanya jawab dengan percaya diri. Kalau ada pertanyaan sulit, cukup bilang, “Itu pertanyaan menarik, kami akan pelajari lebih lanjut.” Terakhir, beri penutup yang bikin audiens mikir, “Wow, ini tim keren banget!” Misalnya: “Kami berharap *SmartAttendance* bisa digunakan di sekolah-sekolah di seluruh Indonesia!” Jangan lupa: senyum, kontak mata, dan antusiasme kalian adalah senjata paling ampuh di panggung!

Latihan Soal

A. Benar atau Salah (5 Soal)

1. AI hanya bisa digunakan untuk pengenalan wajah dan tidak bisa mengenali suara.
   Salah

2. Teachable Machine adalah salah satu platform yang dapat digunakan untuk melatih model AI secara mudah.
   Benar

3. Proyek yang baik tidak perlu ada pembagian peran tim karena semua bisa dikerjakan bersama-sama.
   Salah

4. Dalam presentasi, menyebutkan tantangan dan solusi justru memperlihatkan kelemahan tim.
   Salah

5. Demo aplikasi secara langsung adalah salah satu bagian penting dalam presentasi proyek.
   Benar

B. Pilihan Ganda (10 Soal)

6. Apa tujuan dari bagian pembukaan dalam presentasi proyek?
   A. Menunjukkan hasil akhir
   B. Langsung mendemokan aplikasi
   C. Memperkenalkan tim dan proyek
   D. Menjelaskan kode program
   Jawaban: C

7. Apa fungsi utama dari penggunaan AI dalam aplikasi SmartAttendance?
   A. Mencetak laporan keuangan
   B. Membuat animasi
   C. Mendeteksi dan mencatat kehadiran
   D. Mengatur jadwal sekolah
   Jawaban: C

8. Yang termasuk platform untuk melatih AI dengan mudah bagi pemula adalah...
   A. Microsoft Word
   B. Adobe XD
   C. Teachable Machine
   D. Scratch
   Jawaban: C

9. Mengapa penting membagi peran dalam tim?
   A. Agar tugas lebih ringan dan efisien
   B. Supaya tidak ada yang merasa paling hebat
   C. Karena disuruh guru
   D. Agar terlihat sibuk saat presentasi
   Jawaban: A

10. Bagian mana dalam presentasi yang menunjukkan aplikasi bekerja secara nyata?
    A. Penutupan
    B. Pembukaan
    C. Demo Langsung
    D. Pengembangan Selanjutnya
    Jawaban: C

11. Dalam proses pengembangan aplikasi, tahapan pertama yang perlu dilakukan adalah...
    A. Membuat laporan

B. Menguji aplikasi
C. Menentukan ide dan masalah
D. Presentasi
Jawaban: C

12. Apa contoh fitur tambahan yang bisa ditambahkan untuk pengembangan aplikasi ke depan?
A. Tampilan error lebih banyak
B. Musik latar
C. Suara otomatis saat berhasil login
D. Tombol keluar disembunyikan
Jawaban: C

13. Salah satu keuntungan presentasi dengan ekspresi wajah dan antusiasme adalah...
A. Membuat audiens takut
B. Menambah poin nilai coding
C. Meningkatkan perhatian audiens
D. Menghindari lupa teks
Jawaban: C

14. Fungsi dari diagram alur kerja dalam slide presentasi adalah untuk...
A. Mempercantik tampilan
B. Menambah slide
C. Menjelaskan cara kerja aplikasi secara visual
D. Membingungkan audiens
Jawaban: C

15. Jika saat demo aplikasi mengalami crash, apa langkah terbaik?
A. Langsung keluar ruangan
B. Minta guru skip bagian ini
C. Tunjukkan video cadangan atau screenshot
D. Menyalahkan anggota tim
Jawaban: C

C. Isian Singkat (5 Soal)

16. Platform yang digunakan untuk melatih model AI berbasis suara dan gambar secara sederhana adalah ____.
Jawaban: Teachable Machine

17. Salah satu manfaat AI dalam aplikasi absensi adalah membuat proses absensi menjadi lebih ____ dan ____.
Jawaban: cepat, otomatis

18. Saat membagi tugas dalam tim, orang yang membuat tampilan aplikasi bertanggung jawab atas bagian ____.
Jawaban: desain UI

19. Dalam presentasi, bagian yang berisi rencana jangka panjang aplikasi disebut bagian ____.
    Jawaban: Pengembangan Selanjutnya

20. Salah satu strategi saat tidak bisa menjawab pertanyaan adalah dengan berkata ____.
    Jawaban: “Akan kami pelajari lebih lanjut.”

D. Soal Eksplorasi (3 Soal Esai)

Soal:

Bayangkan kamu adalah presenter utama dari tim mu. Saat demo aplikasi tiba-tiba tidak bisa dijalankan karena koneksi terputus. Bagaimana kamu menjelaskan situasi ini kepada audiens agar mereka tetap memahami fitur unggulan aplikasi kalian? Jelaskan langkah-langkah dan sikap yang kamu ambil.

Soal:

Tim kamu hanya terdiri dari dua orang. Aplikasi yang kalian buat cukup kompleks dan membutuhkan desain UI, integrasi AI, serta dokumentasi. Bagaimana kalian membagi peran secara adil dan efisien? Buatlah pembagian tugas dan jelaskan alasan di balik pembagian tersebut.

Soal:

Guru bertanya kepada timmu: “Apa bukti bahwa AI kalian benar-benar belajar dari data suara yang diberikan?” Jelaskan jawaban yang sederhana namun meyakinkan, sehingga bisa dipahami oleh semua orang di ruangan, termasuk yang belum paham tentang AI.

Tugas Proyek

Proyek 1: “SmartAttendance” – Absen Digital dengan Bantuan AI Suara

Deskripsi Proyek:
Membuat aplikasi absensi yang menggunakan pengenalan suara untuk mencatat kehadiran siswa secara otomatis.

Langkah-langkah:

- **Tentukan masalah:** Guru masih mencatat absensi manual? Itu lambat dan rawan salah!
- **Rancang solusi:** Aplikasi yang bisa mengenali suara nama siswa dan langsung mencatat kehadirannya.
- **Gunakan alat:** Gunakan *Teachable Machine* untuk melatih suara, hubungkan ke *App Inventor* untuk aksi mencatat.
- **Latih & uji coba:** Rekam suara siswa, uji fitur apakah bisa mengenali dengan tepat.

- **Demo seru:** Tunjukkan siswa cukup bilang, “Saya hadir!” dan sistem langsung mencatat!

Nilai tambah: Siswa belajar teknologi AI sambil menyelesaikan masalah nyata di sekolah!

Proyek 2: “ClassBuddy” – Pengatur Jadwal Belajar Pintar

Deskripsi Proyek:
Aplikasi yang membantu siswa mengatur jadwal belajar harian dengan alarm pintar dan pengingat motivasi.

Langkah-langkah:

- **Identifikasi masalah:** Banyak siswa kesulitan mengatur waktu belajar? Sering lupa PR?
- **Desain fitur:** Jadwal harian, notifikasi, dan kutipan motivasi otomatis.
- **Teknologi yang digunakan:** App Inventor untuk tampilan dan logika jadwal, AI untuk menyarankan waktu belajar ideal.
- **Personalisasi:** Aplikasi bisa menyesuaikan saran jadwal berdasarkan mata pelajaran yang sering diabaikan.
- **Tambahan fun:** Tambahkan suara penyemangat, seperti “Semangat belajar, kamu bisa ujian dengan mantap!”

Tujuan pembelajaran: Melatih siswa bertanggung jawab atas waktu sambil memanfaatkan teknologi canggih.

Proyek 3: “SpeakEasy” – Pendeteksi Emosi Lewat Suara

Deskripsi Proyek:
Membuat aplikasi yang bisa mengenali emosi seseorang hanya dari nada suaranya, misalnya marah, sedih, atau senang.

Langkah-langkah:

- **Cari ide awal:** Penting untuk tahu suasana hati teman, apalagi saat belajar daring atau bimbingan!
- **Pelatihan AI:** Gunakan *Teachable Machine* untuk melatih suara dengan label emosi (senang, sedih, netral).
- **Integrasi:** Gabungkan dengan App Inventor agar aplikasi bisa memberi respon berbeda tergantung emosinya.
- **Pengujian:** Coba gunakan suara berbeda—apakah aplikasi bisa mengenali?
- **Tampilan seru:** Tambahkan warna latar sesuai emosi (biru untuk sedih, kuning untuk senang, dll).

Pesan moral: Teknologi bisa bantu kita lebih peduli satu sama lain!

Proyek 4: “FoodRescue” – Aplikasi Penyelamat Makanan Sisa

Deskripsi Proyek:
Aplikasi yang mencatat makanan sisa dari kantin/sekolah dan menyarankan donasi ke panti sosial atau siswa yang membutuhkan.

Langkah-langkah:

- **Tentukan tujuannya:** Mengurangi pemborosan makanan, bantu sesama.
- **Rancang fitur:** Input jenis makanan sisa, waktu kadaluarsa, dan lokasi penjemputan.
- **Tambahkan fitur AI:** AI bisa menyarankan apakah makanan masih layak atau tidak, berdasarkan data waktu dan jenis.
- **Kerja sama tim:** Bagi peran: desainer UI, pengelola database, integrator AI, dokumentasi.
- **Demo yang menginspirasi:** Simulasikan kantin menyumbang makanan sisa ke siswa lain.

Dampak sosial: Menumbuhkan kepedulian sosial + memanfaatkan teknologi secara bijak.

Proyek 5: “LangLearn” – Penerjemah Flash Card Otomatis

Deskripsi Proyek:
Aplikasi belajar bahasa asing berbasis gambar dan suara. Cukup scan gambar, langsung muncul terjemahannya!

Langkah-langkah:

- **Identifikasi tantangan:** Banyak siswa ingin belajar bahasa, tapi sulit menghafal.
- **Buat solusi:** Aplikasi dengan kartu kosakata otomatis dari gambar atau suara.
- **AI-nya bagaimana?** Gunakan model pengenalan gambar atau suara untuk mengenali benda dan sebutannya.
- **Fitur unggulan:** Scan → Terjemah otomatis → Simpan flashcard → Ulang belajar dengan kuis!
- **Demo interaktif:** Tunjukkan fitur live scan, atau suara “apple” → muncul “apel (Indonesia)”.

Serunya: Gabungan AI + game kuis bikin belajar bahasa jadi asyik!

Proyek 6: “AirSense” – Detektor Udara Sekolah

Deskripsi Proyek:
Aplikasi yang terhubung dengan sensor sederhana untuk mendeteksi kualitas udara di ruang kelas.

Langkah-langkah:

- **Masalah utama:** Udara di kelas sering pengap dan membuat ngantuk.

- **Gunakan sensor:** Sensor kualitas udara sederhana (misal MQ135) terhubung ke microcontroller.
- **Data ke aplikasi:** Aplikasi menampilkan kualitas udara secara real-time: baik, sedang, buruk.
- **Visualisasi data:** Tambahkan grafik dan tips: “Buka jendela!”, “Saatnya menyiram tanaman!”
- **Ide pengembangan:** Tambahkan AI yang bisa menyarankan jadwal ventilasi otomatis.

Nilai tambah: Siswa jadi sadar pentingnya lingkungan sehat dan berpikir ilmiah.

Catatan Tambahan:

Semua proyek bisa dibuka dengan pola narasi:

> “Kami melihat masalah [nyata] di sekolah/keseharian kami, dan ingin menyelesaikannya dengan bantuan teknologi AI.”

Dan ditutup dengan:

> “Semoga proyek kami bisa terus berkembang dan benar-benar digunakan oleh siswa, guru, dan masyarakat sekitar.”

BAB: Mengenal Domain .id



Apa Itu Domain?

Bayangkan internet seperti **kota besar** yang punya **jutaan rumah**. Setiap rumah di kota itu punya alamat agar bisa dikunjungi. Nah, nama **domain** itu seperti **alamat kita di internet**, nanti bisa digunakan sebagai alamat website maupun alamat email kita. Dan kita bisa mempunyai alamat email sendiri misalnya halo@meita.id. Misalnya alamat website

Contohnya:

- pandi.id
- home.s.id
- meita.my.id

Alamat itu disebut **nama domain**, dan tanpa domain, kamu harus mengingat angka-angka ribet yang disebut **IP address**. Pasti gak seru, kan?

Apa Itu .id?

Sekarang coba perhatikan domain ini: www.sekolahku.id

Nah, yang bagian **".id"** itu disebut **ekstensi domain**. ".id" menunjukkan bahwa website itu **berasal dari Indonesia**. Setiap negara punya ekstensinya masing-masing, contohnya:

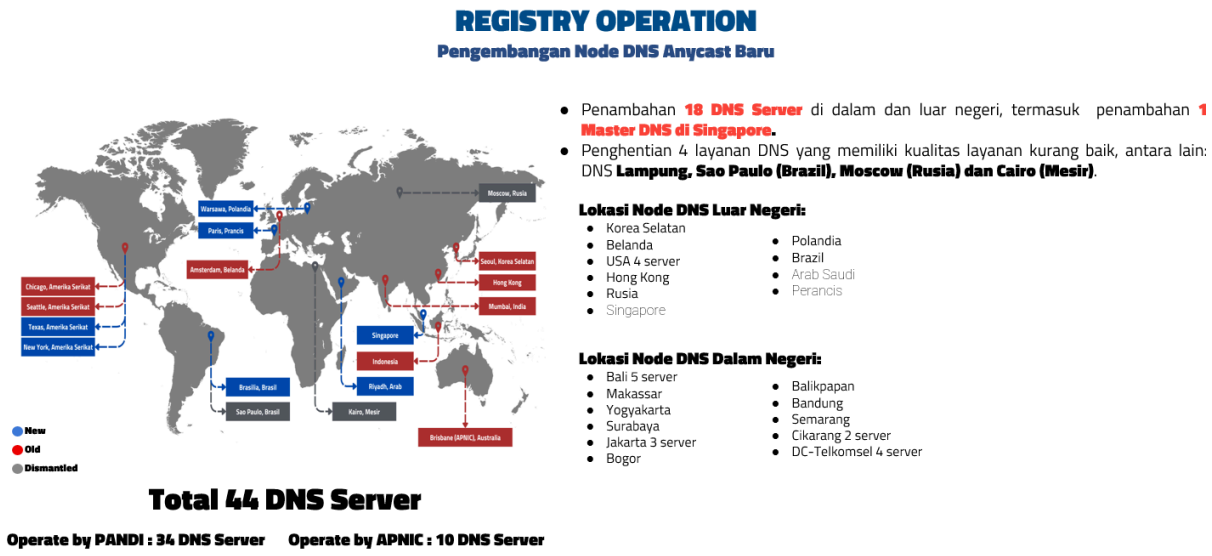
- .jp → Jepang
- .us → Amerika Serikat
- .id → Indonesia 🇮🇩

Menggunakan domain .id itu keren, karena:

- Menunjukkan kebanggaan sebagai orang Indonesia
- Membuat situsmu terlihat lebih terpercaya di mata orang Indonesia
- Cocok banget buat bisnis lokal, sekolah, komunitas, atau bahkan blog pribadi

Keunggulan domain .id

- Representasi Indonesia
  Menunjukan bahwa ini entitas ataupun personal dari Indonesia karena .id adalah kode negara Indonesia
- Sebagai identity
  Selain reprensentasi Indonesia .id juga bisa berarti sebagai identity atau identitas, sehingga sangat menarik untuk dijadikan sebuah alamat website, email maupun brand di internet



Siapa Itu PANDI?

PANDI adalah singkatan dari **Pengelola Nama Domain Internet Indonesia.**
Coba bayangkan **PANDI** seperti **penjaga gerbang dunia .id**. Mereka:

- Mengatur dan mengelola semua domain dengan **akhiran .id**
- Menjaga agar **domain .id aman, tertata, dan tidak disalahgunakan**
- Memberi kesempatan buat siapa saja — termasuk kamu — **punya domain sendiri!**

Misalnya kamu ingin punya website "tokohfavoritku.id", kamu bisa daftar lewat penyedia domain resmi, dan **PANDI** yang mengatur sistemnya agar website-mu bisa diakses semua orang.

Kenapa Kamu Harus Tahu Ini?

- Siapa tahu **nanti kamu bikin website** untuk tugas sekolah, bisnis kecil-kecilan, atau komunitas temanmu
- Supaya kamu melek teknologi, karena **dunia digital adalah masa depan**
- Karena kamu bisa bikin identitas online-mu sendiri dan itu sangat keren!

Kalau kamu bisa punya akun Instagram, kamu juga bisa punya domain sendiri. Bayangkan: www.namamu.id – tempat kamu membagikan karya, foto, tulisan, atau bahkan jualan!

Lebih dalam tentang PANDI

PANDI adalah Registry Nama Domain Indonesia yang berperan untuk mengoperasikan, memelihara dan mengelola Nama Domain Indonesia (.id)

Pengelola Nama Domain Internet Indonesia (PANDI) merupakan perkumpulan yang terdiri dari berbagai pemangku kepentingan, termasuk perwakilan dari pemerintah, operator industri internet, dan akademisi.

Didirikan pada tahun 2006, **PANDI** menerima Redelegasi dari **Internet Assigned Numbers Authority (IANA)** sebagai Registry .id pada tahun 2013.

Hingga 31 Desember 2024 jumlah domain .id yang terdaftar tumbuh menjadi yang terbesar di Asia Tenggara, dengan total 1.215.714 Nama Domain. Nama Domain .id sendiri bisa diartikan sebagai Indonesia, identitas, ide, dan lainnya. Sejalan dengan makna tersebut, **PANDI** juga memiliki kebijakan khusus dalam menyelesaikan perselisihan nama domain .id.

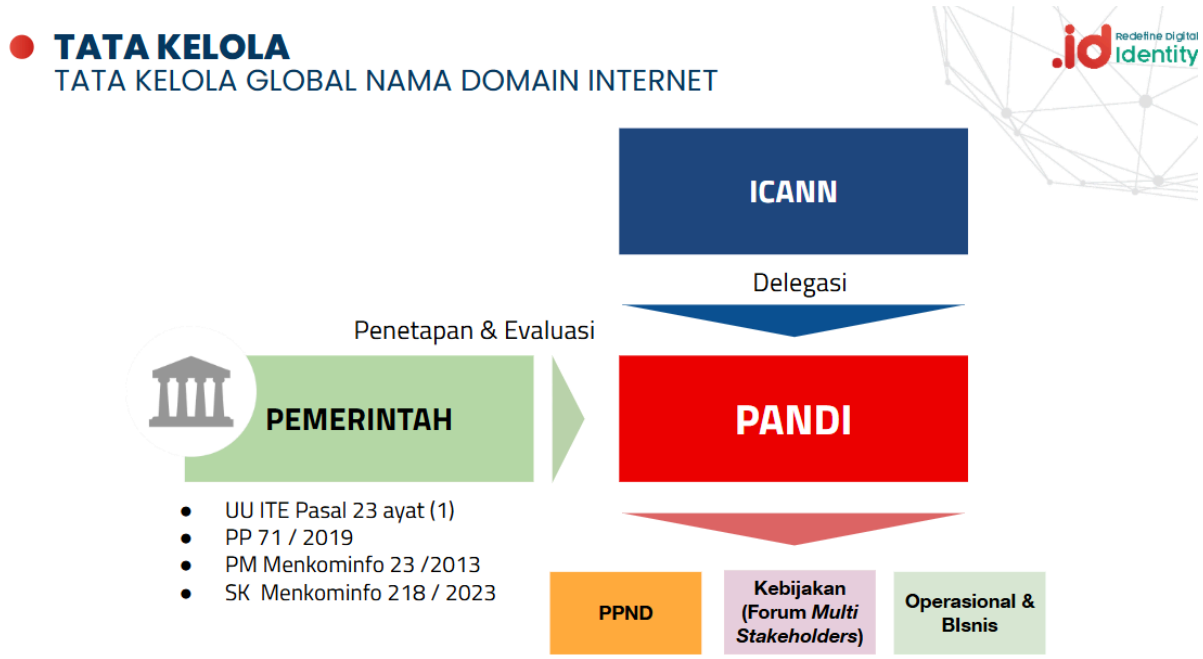
Tugas dan Kewajiban PANDI

Tugas

- Merumuskan kebijakan di bidang pengelolaan Nama Domain Tingkat Tinggi Indonesia
- Menyiapkan, mengoperasikan, dan memelihara infrastruktur yang dibutuhkan serta menyediakan sistem elektronik untuk pengelolaan Nama Domain Tingkat Tinggi Indonesia
- Menyelenggarakan pendaftaran Nama Domain Tingkat Tinggi Indonesia sesuai dengan ketentuan peraturan perundang-undangan, kepatutan yang berlaku dalam masyarakat, dan prinsip kehati-hatian
- Melaksanakan seleksi Registrar Nama Domain
- Memberikan peringatan kepada Registrar Nama Domain jika terindikasi melakukan pelanggaran
- Mencabut hak operasional Registrar Nama Domain jika terbukti melakukan pelanggaran, dan
- Melakukan pengawasan operasional dan teknis Registrar Nama Domain

Kewajiban

- Menjamin sistem elektronik Registri Nama Domain Indonesia beroperasi dengan baik, stabil, aman didukung dengan layanan yang dapat diandalkan
- Menempatkan pusat data dan pusat pemulihan bencana di wilayah Indonesia
- Melakukan pengawasan terhadap Registrar Nama Domain
- Memfasilitasi penyelesaian perselisihan Nama Domain Indonesia
- Melaporkan daftar Registrar Nama Domain kepada Menteri
- Mengikuti ketentuan pengelolaan Nama Domain internasional dan peraturan perundang-undangan
- Menyampaikan laporan berkala kepada Menteri sekurang-kurangnya 1 (satu) kali dalam setahun
- Membayar pungutan biaya Pengelolaan Nama Domain Indonesia



Tata Kelola Internet

PANDI berkomitmen penuh untuk terus berkontribusi aktif dalam tata kelola internet global dengan mendukung infrastruktur yang aman dan inovatif. Hal ini karena domain .id merupakan bagian penting dari ekosistem digital dunia yang inklusif dan berkelanjutan.

Di tahun 2024, PANDI menginisiasi berbagai wadah tata kelola internet nasional serta berpartisipasi aktif dalam forum tata kelola internet global, seperti:

- Forum Multi Stakeholder (FMS),
- PANDI Meeting 2024,
- ICANN 79 di Puerto Riko,
- ICANN 81 di Turki,
- APTLD 86 di Vietnam,
- APAC DNS Forum 2024 yang diselenggarakan oleh PANDI di Bali.

Selain APAC DNS Forum, PANDI juga menggelar program Indonesia Internet Governance Academy (IDIGA) yang bertujuan mencetak pemimpin muda di bidang tata kelola internet, baik di tingkat nasional maupun internasional.

Institut Teknologi Tangerang Selatan (ITTS) dan Penulis.

Para penulis adalah **dosen aktif di Institut Teknologi Tangerang Selatan (ITTS)**, sebuah **institusi pendidikan tinggi berbasis teknologi** yang berperan aktif dalam pengembangan **solusi digital** dan *kecerdasan buatan (AI)*. Tidak hanya mengajar, mereka juga merupakan **praktisi dan inovator** yang menghasilkan **karya-karya nyata di bidang teknologi mandiri**.

Berlokasi di **Komplek Komersial BSD, Serpong Utara, Tangerang Selatan**, **ITTS hadir sebagai pusat inovasi** yang mengintegrasikan **pembelajaran, penelitian, dan pengembangan teknologi secara menyeluruh**. Fokus utama ITTS mencakup bidang strategis seperti *AI, Machine Learning, Deep Learning*, jaringan, *cloud*, keamanan siber, pemrograman, sistem informasi, dan multimedia.

ITTS menawarkan lingkungan akademik yang dinamis dan kolaboratif. Setiap minggu, diselenggarakan berbagai *workshop*, *demo teknologi*, seminar, hingga *webinar*—mayoritas **gratis dan tersedia secara terbuka di YouTube**. Ini menciptakan **kultur belajar yang aktif dan relevan dengan perkembangan industri**.

Fasilitas ITTS menunjang kegiatan riset dan inovasi secara optimal. Lab komputer dilengkapi *GPU RTX 4060* untuk eksperimen *AI*, serta **laboratorium teknologi dengan puluhan server, perangkat *IoT*, sistem komunikasi radio, dan infrastruktur jaringan jarak jauh**. Dari ekosistem ini lahirlah **produk-produk mandiri** seperti **ChatGPT versi lokal** untuk institusi pendidikan dan **sistem perpustakaan digital yang dapat diakses tanpa koneksi internet**.

Daya saing ITTS juga tercermin dari karakter dosen dan mahasiswa yang tidak hanya akademik, tetapi juga **produktif secara praktis**. Mahasiswa didorong aktif menjadi **narasumber, penulis, dan peneliti** sejak dini. Banyak diantaranya telah menghasilkan **karya buku, artikel ilmiah**, serta **terlibat langsung dalam proyek-proyek teknologi**.

Kolaborasi strategis dengan berbagai mitra industri—seperti *IDCloudHost, Ubuntu, XecureIT, Dinas KOMINFO Tangerang Selatan*, berbagai *ISP*, dan komunitas IT nasional—menjadi **jembatan nyata antara kampus dan dunia profesional**. Hasilnya, **lebih dari 75% mahasiswa ITTS telah bekerja bahkan sebelum lulus**, dan **hampir seluruh lulusan langsung terserap industri**.

ITTS bukan hanya tempat belajar, tapi tempat mencipta. Dosen sebagai **penulis dan praktisi**, mahasiswa sebagai **peneliti dan kontributor**—semuanya berperan **membangun ekosistem teknologi yang mandiri dan berdaya saing tinggi**.

Institut Teknologi Tangerang Selatan (ITTS)
Komplek Komersial BSD
Jl. Raya Serpong No. Kav. 9, Lengkong Karya, Serpong Utara
Kota Tangerang Selatan, Banten 15331
📞 (+62) 877-7277-1775 📧 info@itts.ac.id 🌐 www.itts.ac.id

Tentang Penulis



Anas Nasrulloh, M.Kom., merupakan lulusan S2 dari Universitas Budi Luhur, Jakarta jurusan Rekayasa Komputer Terapan. Saat ini, Anas aktif sebagai Kepala Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) serta mengajar sebagai dosen tetap di Institut Teknologi Tangerang Selatan (ITTS).

Buku ini ditulis sebagai bentuk komitmen penulis untuk menghadirkan sumber belajar data science yang menyenangkan, aplikatif, dan mudah diakses oleh pelajar dan mahasiswa di Indonesia.

Email : anas@itts.ac.id