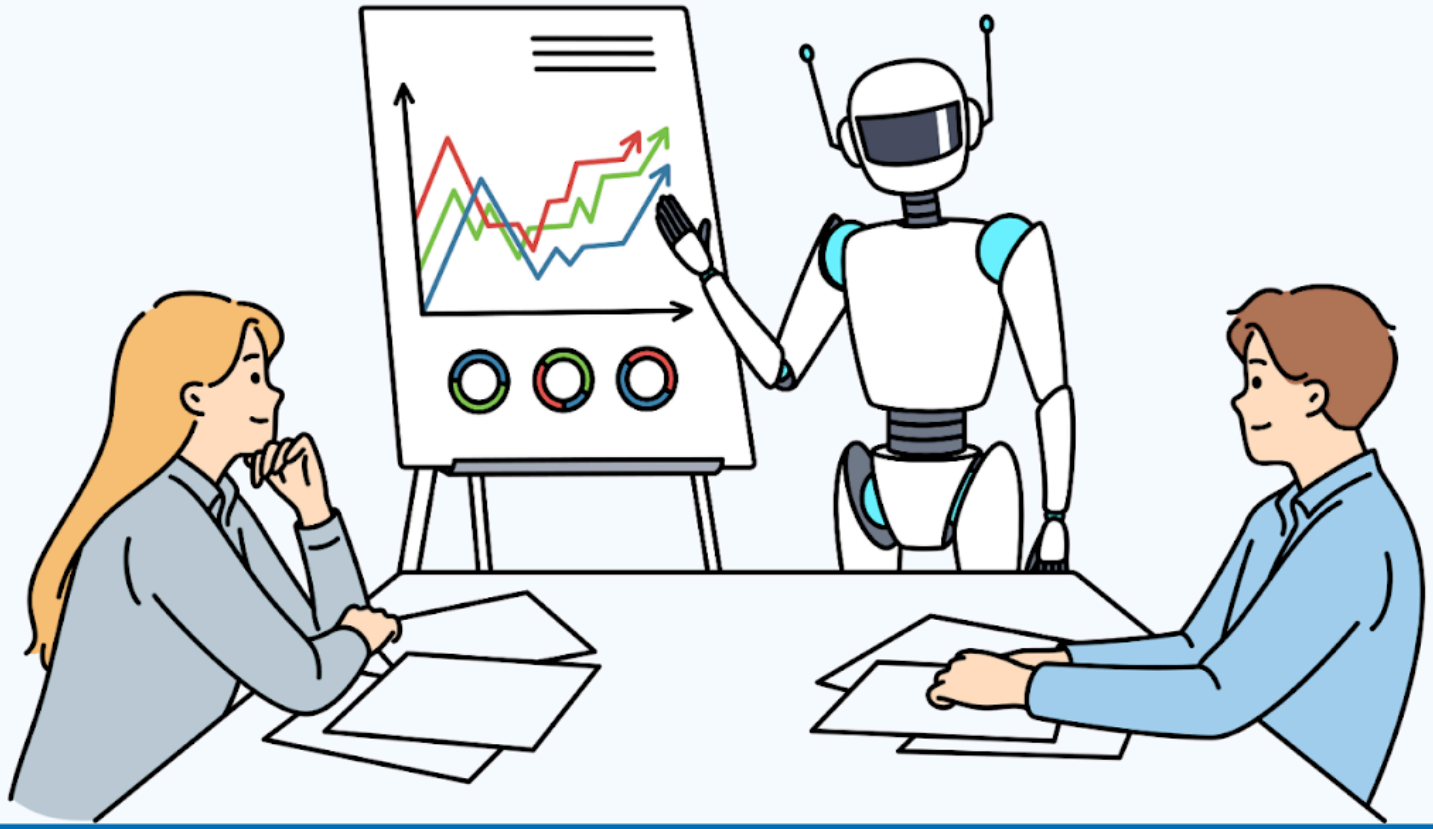
InTech press

ARTIFICIAL INTELLIGENCE

Proyek Keren dengan AI: Bikin Aplikasi Sendiri Mulai dari Ide Sampai Jadi!



UNTUK SMP KELAS IX SEMESTER 1

Devi Fatwanti
Onno W. Purbo

Buku AI SMP Kelas 9 Semester 1

Proyek Keren dengan AI: Bikin Aplikasi Sendiri Mulai dari Ide Sampai Jadi!

Devi Fatwanti
Onno W. Purbo

Institut Teknologi Tangerang Selatan (ITTS)
2025

Daftar Isi

Daftar Isi **3**
Lisensi & Catatan Karya **7**
Disclaimer **7**
Kata Pengantar **8**
BAB 1: Mengenal Teachable Machine dan Dasar-dasar AI 9
Tujuan Pembelajaran 9
Peta Konsep 9
Apersepsi 10
Penjelasan Konsep (Teori) 10
Apa Itu Teachable Machine? 10
Dasar-Dasar Machine Learning (Pembelajaran Mesin) 11
Membuat Model Deteksi Suara 11
Membuat Model Gambar dan Pose Tubuh 12
Perbandingan Model Suara dan Gambar 13
Ilustrasi: Aplikasi Deteksi Suara/Wajah dalam Dunia Nyata 13
Contoh Kasus atau Ilustrasi 14
Contoh Soal dan Pembahasan 16
Aktivitas Siswa 18
Fakta Menarik / Fun Facts 20
Tips Belajar atau Tips Cepat Menghafal 20
Rangkuman 21
Latihan Soal 22
Tugas Proyek 25
BAB 2: Jadi Kreator Digital! Hosting Script dengan GitHub **28**
Tujuan Pembelajaran 28
Peta Konsep 28
Apersepsi 29
Penjelasan Konsep (teori) 29
Webviewer di AppInventor 29
Apa Itu GitHub dan Mengapa Penting? 32
Langkah-Langkah Membuat Akun GitHub 32
Cara Membuat Project Baru (Repository) 33
Membuat Script index.html di Repository Github 33
(Alternatif) Membuat Script Sederhana untuk Diunggah 34
Uji Coba dengan App Inventor: WebViewer 35
Kesimpulan 36
Contoh Kasus atau Ilustrasi 37
Contoh Soal dan Pembahasan 39
Fakta Menarik / Fun Facts 41
Tips Belajar atau Tips Cepat Menghafal 42
Aktivitas Siswa 43
Rangkuman 45

Latihan Soal..... 46
Tugas Proyek..... 49
BAB 3: Aplikasi Pendeteksi Ekspresi: Membuat AI Sendiri dari Kamera HP!.....52
Tujuan Pembelajaran..... 52
Peta Konsep.....52
Apersepsi..... 53
Penjelasan Konsep (teori).....53
Buat Model di Teachable Machine..... 53
Buat Halaman Web di GitHub..... 53
Contoh isi index.html (script GitHub):.....53
Upload ke GitHub Pages.....54
Test GitHub Page..... 55
Buat App di MIT App Inventor..... 56
Hasil..... 56
Catatan untuk Siswa SMP.....56
Contoh Kasus atau Ilustrasi..... 56
Contoh Soal dan Pembahasan..... 58
Fakta Menarik / Fun Facts.....60
Tips Belajar atau Tips Cepat Menghafal..... 61
Aktivitas Siswa..... 62
Rangkuman.....64
Latihan Soal..... 64
Tugas Proyek..... 67
BAB 4: Memahami Akurasi dan Kualitas Data AI..... 71
Tujuan Pembelajaran..... 71
Peta Konsep.....71
Apersepsi..... 72
Penjelasan Konsep (teori).....72
Apa itu Akurasi Model AI?.....72
Mengapa Akurasi Penting?..... 72
Debugging Model AI (Cari & Perbaiki Kesalahan!)..... 73
Mengoptimalkan Kinerja Aplikasi AI.....73
Ilustrasi: Menggunakan Software untuk Memecahkan Masalah AI.....74
Contoh Kasus Ilustrasi Sederhana.....74
Contoh Soal dan Pembahasan..... 76
Fun Facts..... 78
Tips Belajar dan Cepat Menghafal.....78
Aktivitas Siswa..... 78
Rangkuman.....80
Latihan Soal..... 81
Tugas Proyek..... 84
BAB 5: Merancang Aplikasi Ala Kreator Muda: Belajar dari Masalah, AI, hingga UI/UX!..... 87
Tujuan Pembelajaran..... 87
Peta Konsep.....87

Apersepsi..... 88
Penjelasan Konsep (teori)..... 88
Identifikasi Masalah dan Kebutuhan Pengguna..... 88
Menentukan Fitur AI dalam Aplikasi..... 89
Merancang Tampilan dan Pengalaman Pengguna (UI & UX)..... 90
Ilustrasi..... 90
Contoh Kasus dan Ilustrasi..... 91
Contoh Soal dan Pembahasan..... 92
Fun Facts..... 94
Aktivitas Siswa..... 95
Tips Belajar dan Menghafal..... 97
Rangkuman..... 98
Latihan Soal..... 98
Tugas Proyek..... 101
BAB 6: Gunakan ChatGPT untuk Cari Ide Aplikasi..... 104
Tujuan Pembelajaran..... 104
Peta Konsep..... 104
Apersepsi..... 104
Penjelasan Konsep (Teori)..... 105
Mengenal ChatGPT sebagai Asisten Ide..... 105
Teknik Dasar Prompt Engineering untuk Ide Aplikasi..... 106
Contoh Soal & Pembahasan Menarik..... 107
Brainstorming & Pengembangan Ide Aplikasi..... 107
Ilustrasi..... 107
Contoh Kasus atau Ilustrasi..... 108
Contoh Soal & Pembahasan..... 109
Fun Facts..... 112
Tips Belajar..... 112
Aktivitas Siswa..... 113
Rangkuman..... 115
Latihan Soal..... 116
Tugas Proyek..... 118
BAB 7: Presentasi & Evaluasi Aplikasi..... 1
Tujuan Pembelajaran..... 1
Peta Konsep..... 1
Apersepsi..... 1
Penjelasan Konsep (Teori)..... 1
Mengapa Presentasi Itu Penting?..... 1
Elemen-Elemen Presentasi yang Efektif..... 1
Mendemonstrasikan Aplikasi dan Sesi Tanya Jawab..... 1
Menerima dan Memberikan Feedback Konstruktif..... 1
Konsep Iterasi: Perbaikan Terus-Menerus..... 1
Contoh Soal & Tantangan:..... 1
Ilustrasi & Perjuangan Mengatasinya..... 1

Contoh Kasus / dan Ilustrasi..... 1
Contoh Soal dan Pembahasan..... 1
Fun Facts..... 1
Tips Belajar.....1
Aktivitas Siswa..... 1
Rangkuman.....1
Latihan Soal..... 1
Tugas Proyek..... 1
BAB 8: Menulis Deskripsi Aplikasi dengan Bantuan AI.....1
Tujuan Pembelajaran..... 1
Peta Konsep.....1
Apersepsi..... 1
Penjelasan Konsep (Teori)..... 1
Pentingnya Deskripsi Aplikasi yang Menarik.....1
Apa Itu App Store Optimization (ASO)?..... 1
Memanfaatkan Kecerdasan Buatan (AI) untuk Menyusun Deskripsi..... 1
Menyempurnakan Deskripsi dengan Sentuhan Manusia.....1
Kesimpulan.....1
Ilustrasi.....1
Contoh Kasus dan Ilustrasi..... 1
Contoh Soal dan Pembahasan..... 1
Fun Facts..... 1
Tip Belajar..... 1
Aktivitas Siswa..... 1
Rangkuman.....1
Latihan Soal..... 1
Tugas Proyek..... 1
BAB: Mengenal Domain .id..... 1
Apa Itu Domain?.....1
Apa Itu .id?..... 1
Keunggulan domain .id.....1
Siapa Itu PANDI?..... 1
Kenapa Kamu Harus Tahu Ini?..... 1
Lebih dalam tentang PANDI.....1
Tata Kelola Internet..... 1
Institut Teknologi Tangerang Selatan (ITTS) dan Penulis..... 1

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

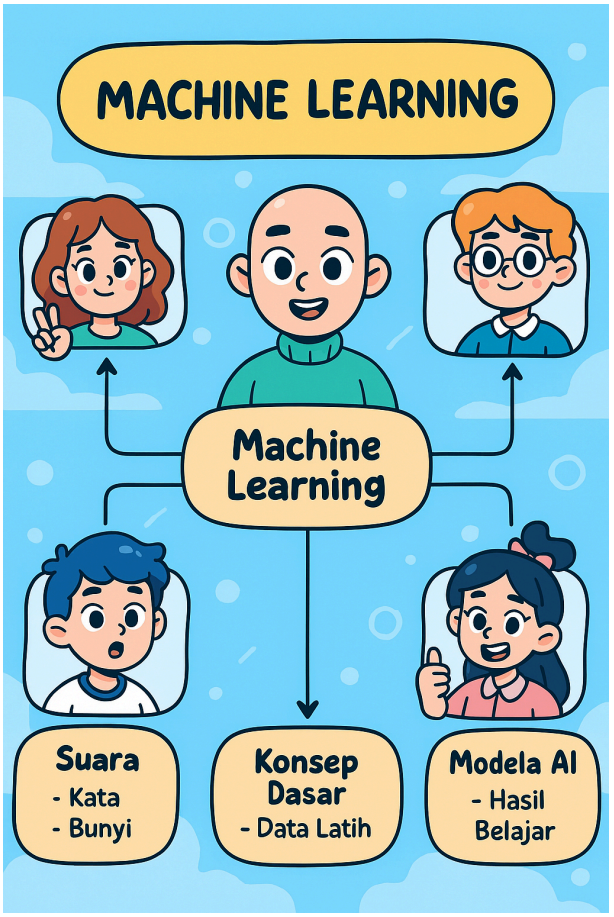
BAB 1: Mengenal Teachable Machine dan Dasar-dasar AI

Tujuan Pembelajaran

Setelah mengikuti pembelajaran ini, siswa dapat:

1. Menjelaskan apa itu Teachable Machine dan bagaimana cara kerjanya.
2. Mengenal konsep dasar machine learning: data latih, model AI, dan prediksi.
3. Membuat model AI sederhana menggunakan gambar, suara, dan gerakan tubuh.
4. Menggunakan Teachable Machine untuk membuat aplikasi interaktif.
5. Membedakan model suara dan model gambar/pose berdasarkan input dan cara pelatihannya.

Peta Konsep



Apersepsi

Bayangkan kamu bisa membuat komputer mengenali suara kamu saat bilang "Ya" atau "Tidak", atau bisa tahu apakah tanganmu sedang diangkat atau tidak — hanya dari foto atau suara! Seru, kan?

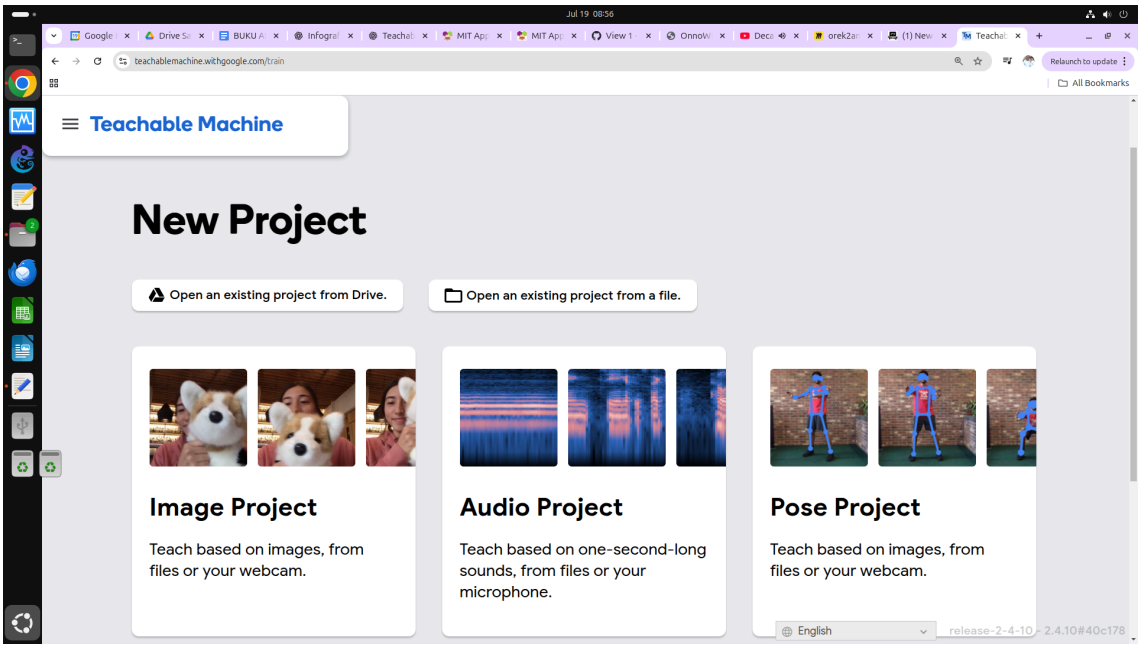
Nah, lewat **Teachable Machine**, kamu bisa melatih AI seperti itu **tanpa perlu coding sama sekali**. Tinggal rekam, foto, latih, dan tes — seperti main game, tapi sambil belajar teknologi masa depan!

Kamu akan jadi "guru" untuk mengajari komputer agar bisa **mengenali suara, gambar, bahkan gerakan tubuh**. Siap jadi pelatih AI hebat?

Penjelasan Konsep (Teori)

Apa Itu Teachable Machine?

Teachable Machine adalah platform web gratis dari Google yang memungkinkan siapa saja—termasuk kamu—untuk **membuat model kecerdasan buatan (AI)** hanya dengan beberapa klik. Hebatnya lagi, kamu **tidak perlu bisa coding** untuk menggunakannya!



Dengan Teachable Machine, kamu bisa "mengajarkan" komputer untuk mengenali **gambar**, **suara**, atau bahkan **gerakan tubuh**, cukup dengan memberikan contoh-contohnya.

Dasar-Dasar Machine Learning (Pembelajaran Mesin)

Sebelum kita mulai bermain dengan Teachable Machine, yuk kenali dulu konsep penting dalam Machine Learning:

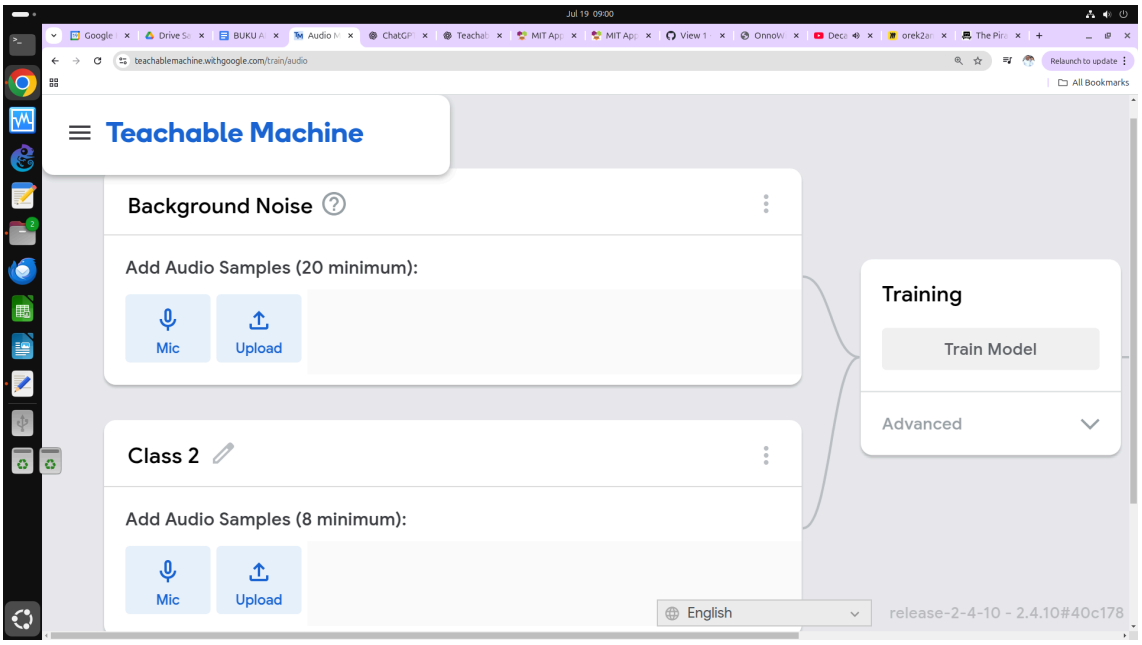
1. **Training Data (Data Latihan)**
   Data berupa gambar, suara, atau pose tubuh yang kita kumpulkan sebagai contoh untuk "mengajari" komputer.
2. **Model AI**
   Ini adalah hasil dari proses pelatihan. Model adalah otak buatan yang sudah belajar dari data yang kita berikan.
3. **Prediksi**
   Setelah model jadi, kita bisa mengujinya. Model akan mencoba "menebak" input baru berdasarkan apa yang sudah dipelajari sebelumnya.

Contoh Seru:
Agar AI bisa membedakan **kucing dan anjing**, kita harus memberinya banyak foto kucing **dan** anjing. Semakin bervariasi datanya, semakin pintar modelnya!

Membuat Model Deteksi Suara

Apa Itu Audio Model?

Audio model adalah jenis model AI yang dilatih menggunakan suara. Di Teachable Machine, kamu bisa **merekam suara langsung** dan melatih AI agar bisa mengenali kata atau bunyi tertentu.



Langkah-Langkah Membuatnya:

1. **Buat Proyek Baru** di Teachable Machine (pilih "Audio Project").

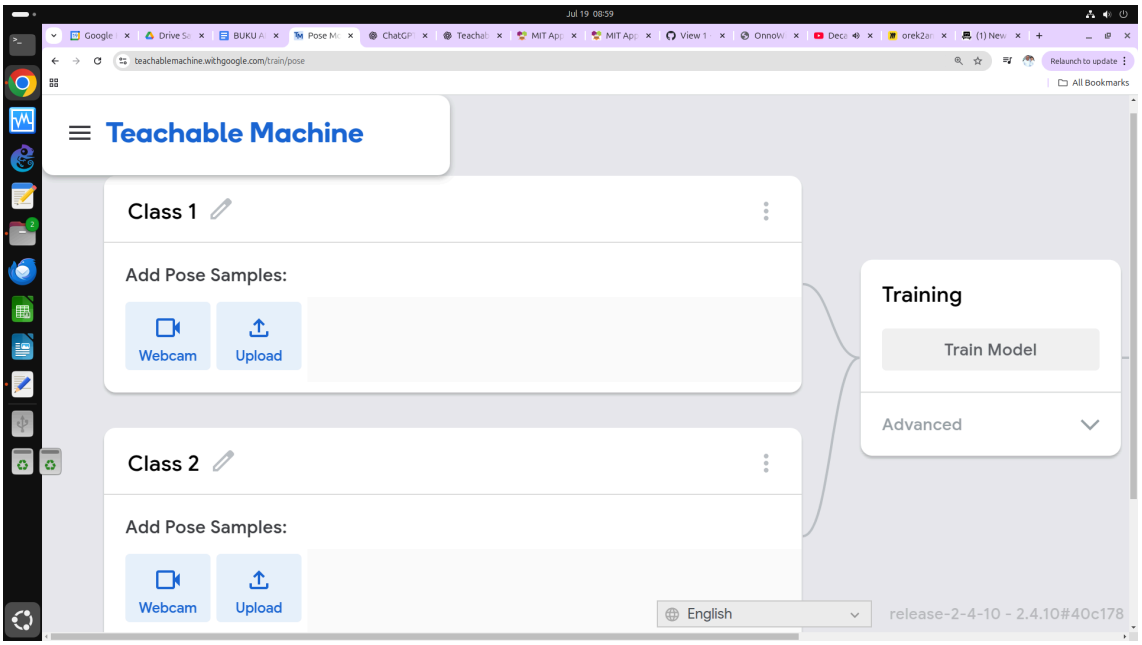
2. **Kumpulkan Data Suara** – Misalnya: rekam kata “Ya” dan “Tidak” dalam berbagai intonasi dan kondisi.
3. **Latih Model** – Biarkan Teachable Machine mempelajari pola dari suara-suara yang kamu rekam.
4. **Ekspor Model** – Simpan hasilnya dalam format **TensorFlow.js**, agar bisa digunakan dalam aplikasi.

Contoh Ilustrasi:
Buat aplikasi yang bisa berhitung saat kamu berkata "plus" atau "minus".
Suara-suara ini akan dilatih agar AI mengenali mana yang menambah dan mana yang mengurangi.

Membuat Model Gambar dan Pose Tubuh

Apa Itu Image/Pose Model?

Model jenis ini dilatih dengan **gambar** atau **pose tubuh**. Misalnya, kamu bisa melatih AI untuk mengenali apakah seseorang sedang **mengangkat tangan** atau tidak.



Langkah-Langkah Membuatnya:

1. **Mulai Proyek Gambar/Pose** di Teachable Machine.
   Ambil Foto dengan berbagai posisi atau ekspresi (misalnya: tangan di atas, tangan di bawah).
2. **Latih Model** – Teachable Machine akan mempelajari pola dari gambar-gambar itu.
3. **Ekspor Model** – Siap digunakan untuk membuat aplikasi pendeteksi gambar.

Tips:
Pastikan gambar yang kamu gunakan bervariasi:

- Sudut pandang berbeda
- Cahaya yang berbeda
- Latar belakang beragam
- Ekspresi wajah yang beragam

Contoh Ilustrasi:
Latih AI agar bisa mengenali **tangan diangkat** dan **tangan diturunkan**. Kamu tinggal foto dengan dua pose itu, dan biarkan Teachable Machine belajar membedakannya.

Perbandingan Model Suara dan Gambar

Aspek	Model Suara	Model Gambar/Pose
Input yang Dilatih	Rekaman suara	Gambar atau pose tubuh
Variasi yang Dibutuhkan	Nada, intonasi, kondisi suara	Sudut pandang, cahaya, ekspresi
Proses Pelatihan	Merekam → Latih → Uji	Memotret → Latih → Uji
Output Model	TensorFlow.js (untuk suara)	TensorFlow.js (untuk gambar/pose)

Dengan memahami teori dan langkah-langkah di atas, kamu siap membuat aplikasi AI buatanmu sendiri! Ayo tantang dirimu dan mulai bereksperimen dengan Teachable Machine

Ilustrasi: Aplikasi Deteksi Suara/Wajah dalam Dunia Nyata

Bayangkan kamu sedang bermain game di HP dan cukup berkata "lompat" maka karakter langsung melompat! Atau kamu membuka aplikasi tanpa menyentuh layar, hanya dengan **tersenyum ke kamera**. Wah, keren banget kan?

Semua itu bisa terjadi karena bantuan teknologi **AI (Artificial Intelligence)** yang belajar mengenali suara dan wajah. Dengan bantuan **Teachable Machine dari Google**, kalian bisa **melatih model AI sendiri** hanya dengan **klik dan rekam**, tanpa perlu bisa coding.

Lalu, bagaimana kalau kita buat aplikasi sederhana yang bisa **menggerakkan gambar ke kiri dan kanan hanya dengan suara**? Atau yang bisa **mendeteksi apakah tangan kalian sedang diangkat atau tidak**? Yuk kita coba!

Contoh Kasus atau Ilustrasi

Kasus 1: “Penjaga Gerbang Rahasia” – Suara Sebagai Kunci

Ilustrasi:
Bayangkan kamu punya *pintu rahasia* di game atau aplikasi buatanmu. Tapi... pintu itu cuma bisa dibuka kalau kamu berkata "buka" dengan suaramu. Kalau bilang "tutup", pintunya kembali terkunci. Hebat, ya?

Nah, di sini kamu akan berperan seperti pembuat sandi rahasia! Dengan **Teachable Machine**, kamu latih AI supaya mengenali dua kata: “buka” dan “tutup”. Model AI ini akan menjadi kunci suara digital yang hanya terbuka kalau kamu bicara dengan kata yang tepat.

Konsep yang dipelajari:

- Melatih AI suara
- Input: suara unik
- Output: aksi berdasarkan perintah suara

Kasus 2: “Cermin Ajaib” – Wajahmu Memunculkan Reaksi

Ilustrasi:
Kamu menghadap kamera dan tersenyum... tiba-tiba muncul gambar matahari cerah! Tapi kalau kamu cemberut, eh… muncul awan mendung.

Itulah kekuatan AI pengenal wajah! Kamu melatih **model gambar** untuk membedakan ekspresi wajahmu: senyum dan cemberut. Seolah-olah kamu sedang berbicara *tanpa kata*, dan AI membaca perasaan dari ekspresimu!

Konsep yang dipelajari:

- Melatih AI visual (gambar)
- Input: ekspresi wajah
- Output: respons visual berdasarkan ekspresi

Kasus 3: “Robot Taat Perintah” – Kontrol Gerakan Lewat Suara

Ilustrasi:
Bayangkan kamu punya robot kecil di layar aplikasi. Kalau kamu bilang “kiri”, dia meluncur ke kiri. Bilang “kanan”? Dia pindah ke kanan.

Kamu melatih **model suara** dengan dua kata: “kiri” dan “kanan”. AI akan mendengar dan merespons, seperti punya asisten robot pribadi yang patuh!

Konsep yang dipelajari:

- Input suara → tindakan arah
- Model mendeteksi arah hanya dari kata

- Kombinasi AI + logika gerak

Kasus 4: “Detektif Gerak Tubuh” – Tebak Pose, Tebak Aksi

Ilustrasi:
Kamu jongkok, AI langsung bilang: "Itu jongkok!". Kamu berdiri dan melompat, AI langsung tahu perbedaannya.

Kamu melatih **model pose tubuh** agar bisa mengenali berbagai posisi tubuhmu. Ini seperti kamu menciptakan game yang bisa dimainkan hanya dengan gerakan tubuh. Mau jadi ninja digital?

Konsep yang dipelajari:

- Gambar sebagai input pose
- AI mengenali postur/pose manusia
- Cocok untuk game berbasis kamera

Kasus 5: “Kalkulator Bicara” – Hitung Angka Cuma Pakai Suara

Ilustrasi:
Bayangkan kamu bilang “plus” dan angka bertambah. Bilang “minus”, angka berkurang. Tidak perlu tombol!

Kamu melatih AI dengan dua kata: “plus” dan “minus”, lalu menghubungkannya dengan angka di aplikasi. Hasilnya? Kamu punya kalkulator yang bisa mendengar!

Konsep yang dipelajari:

- Model suara → logika perhitungan
- Interaksi real-time dengan angka
- Kombinasi AI + logika matematika

Kasus 6: “Lampu Lalu Lintas Pintar” – Suara Jadi Rambu

Ilustrasi:
Bayangkan kamu sedang membuat simulasi jalan raya mini. Ada mobil di layar. Kalau kamu bilang “jalan”, mobilnya melaju. Tapi saat kamu bilang “stop”, langsung berhenti.

Kamu melatih AI untuk mengenali kata “jalan” dan “stop”. Lalu, kamu membuat AI menjadi polisi lalu lintas digital—yang mendengar, bukan meniup peluit!

Konsep yang dipelajari:

- AI untuk simulasi interaktif
- Suara → kontrol logika stop/start
- Belajar tanggung jawab melalui simulasi

Masing-masing kasus ini memperkenalkan **konsep AI secara menyenangkan dan aplikatif**, sekaligus **menantang siswa SMP untuk berpikir kreatif**. Mereka bisa *merasakan langsung* bagaimana teknologi AI bekerja dari suara, gambar, atau gerakan. Dengan pendekatan berbasis cerita, siswa tidak hanya belajar, tapi juga berimajinasi dan terinspirasi.

Contoh Soal dan Pembahasan

Soal 1: Bayangkan Kamu Jadi Pelatih AI!

Kamu ingin membuat AI yang bisa mengenali suara "Halo" dan "Bye". Apa langkah-langkah yang harus kamu lakukan di Teachable Machine agar model tersebut bisa mengenali suara kamu dengan akurat?

Pembahasan:
Langkah-langkahnya:

1. Buka Teachable Machine dan buat proyek baru, pilih **Audio Project**.
2. Rekam suara "Halo" dan "Bye" dengan berbagai intonasi (misalnya pelan, cepat, saat kamu flu, saat kamu semangat).
3. Klik **Train Model** agar AI bisa belajar dari suara-suaramu.
4. Uji model dengan suara baru.
5. Terakhir, **ekspor model** dalam format **TensorFlow.js** untuk digunakan di aplikasi seperti App Inventor.

Tips: Kenapa rekaman harus bervariasi? Karena AI belajar dari data! Semakin banyak variasi, semakin pintar prediksi AI-nya.

Soal 2: Siapakah Aku? Gambar atau Suara?

Aku adalah model AI yang dilatih dengan suara manusia. Aku bisa mengenali kata “stop” dan “jalan” hanya dari audio. Aku bukan gambar, aku bukan pose. Siapakah aku?

A. Model Pose
B. Model Gambar
C. Model Suara
D. Model Teks

Jawaban: C. Model Suara
Pembahasan:
Model suara dilatih dengan **rekaman audio**. Teachable Machine akan mempelajari pola-pola suara yang kamu berikan, seperti nada, intonasi, dan kecepatan bicara.

Soal 3: Kenapa Harus Bervariasi?

Mengapa sangat penting menggunakan **berbagai gaya dan kondisi suara atau gambar** saat melatih AI?

Pembahasan:
Agar AI **tidak bingung saat menghadapi data baru**!
Contoh: Jika kamu hanya merekam kata “Halo” satu kali, AI akan sulit mengenali ketika kamu mengucapkannya dengan nada berbeda.
Sama halnya dengan gambar — kalau hanya satu sudut wajah, AI bisa gagal mengenali saat kamu menoleh atau berubah pencahayaan. Jadi, variasi = **kunci kecerdasan model!**

Soal 4: Detektif Pose Tubuh

Dira membuat model AI yang bisa mengenali gerakan tubuh: berdiri, jongkok, dan melompat. Dia menggunakan kamera untuk mengambil gambar dari setiap pose.
Model AI yang Dira buat termasuk jenis model:

A. Teks
B. Audio
C. Pose/Gambar
D. Angka

Jawaban: C. Pose/Gambar
Pembahasan:
Model Pose atau Gambar dilatih dari **foto tubuh atau ekspresi**. Dalam kasus Dira, dia mengambil pose-pose yang berbeda dan memberikan label seperti "berdiri", "jongkok", dan "lompat" agar AI bisa membedakannya.

Soal 5: Tantangan Pemrogram Suara!

Kamu sudah punya model AI suara dengan dua label: "plus" dan "minus". Jelaskan bagaimana kamu bisa membuat aplikasi kalkulator yang berubah sesuai suara menggunakan App Inventor!

Pembahasan:
Langkah logis:

1. Hubungkan **model suara** ke App Inventor.

   Gunakan blok **IF** dalam pemrograman visual:

   ```
   IF suara == "plus" THEN
    nilai = nilai + 1
   ELSE IF suara == "minus" THEN
    nilai = nilai - 1
   ```

2. Tampilkan nilai tersebut di layar.
   Voila! Kamu baru saja membuat **kalkulator suara tanpa tombol**. AI + logika sederhana bisa jadi alat canggih!

Soal 6: Siapa Lebih Sensitif?

Dari segi input dan pelatihan, mana yang lebih sensitif terhadap **intonasi suara**, dan mana yang lebih sensitif terhadap **sudut pandang kamera**?

Pembahasan:

- **Model Suara** sensitif terhadap **intonasi, kecepatan bicara, dan kebisingan**.
- **Model Gambar/Pose** lebih sensitif terhadap **cahaya, sudut kamera, dan latar belakang**.

Artinya, saat kamu melatih model:

- Ucapkan kata-kata dengan gaya berbeda (untuk suara).
- Ambil gambar dari berbagai posisi dan ekspresi (untuk pose/gambar).

Aktivitas Siswa

Aktivitas 1: Jadi Pelatih AI Suaramu Sendiri!

Tujuan: Melatih AI agar bisa mengenali suara "ya" dan "tidak"

Langkah Seru:

1. Buka Teachable Machine dan buat **Audio Project**.
2. Rekam suara “ya” dan “tidak” dalam berbagai versi: pelan, keras, ceria, lelah.
3. Klik **Train Model** dan lihat bagaimana AI belajar dari suaramu.
4. Coba ucapkan “ya” atau “tidak” dan lihat apakah AI bisa menebaknya!

Tantangan: Tambahkan suara "mungkin" — bisa tidak AI membedakannya?

Aktivitas 2: Wajahmu, Kode Ekspresi Ajaib!

Tujuan: Melatih AI mengenali ekspresi senyum dan cemberut

Langkah Seru:

1. Buat proyek **Image Project** di Teachable Machine.
2. Ambil foto saat kamu **tersenyum** dan **cemberut** dalam berbagai sudut dan cahaya.
3. Latih model dan uji dengan ekspresi nyata di depan kamera.
4. Hubungkan ke aplikasi sederhana yang menampilkan emoji sesuai ekspresimu.

Tantangan: Tambahkan ekspresi baru seperti “kaget” atau “ngantuk” dan latih AI mengenalinya!

Aktivitas 3: Kalkulator Suara – Hitung Pakai Kata!

Tujuan: Membuat AI yang bisa menambah dan mengurangi angka berdasarkan suara

Langkah Seru:

1. Latih model suara untuk mengenali “plus” dan “minus”.
2. Uji model: apakah AI bisa membedakan kedua suara tersebut?
3. Gabungkan dengan App Inventor, lalu tampilkan angka di layar.
4. Coba perintah “plus” berkali-kali. Apakah angka terus bertambah?

Tantangan: Tambahkan kata “reset” untuk mengatur ulang angka ke nol.

Aktivitas 4: Pose Master – Gaya Tubuh Jadi Kendali!

Tujuan: Melatih AI mengenali gerakan tubuhmu seperti berdiri, jongkok, dan lompat

Langkah Seru:

1. Pilih **Pose Project** di Teachable Machine.
2. Ambil gambar kamu dalam posisi **berdiri**, **jongkok**, dan **lompat**.
3. Latih model dan uji: coba lakukan gerakan di depan kamera.
4. Tampilkan nama pose secara otomatis di layar saat AI mengenalnya.

Tantangan: Tambahkan pose baru seperti “miring ke kiri” dan lihat apakah AI bisa memahaminya.

Aktivitas 5: Navigasi Suara – Gerakkan Karakter Tanpa Tombol!

Tujuan: Mengendalikan arah karakter game hanya dengan suara

Langkah Seru:

1. Latih model suara untuk mengenali kata “kiri” dan “kanan”.
2. Uji AI dengan ucapkan perintah di berbagai nada.
3. Hubungkan ke karakter virtual di App Inventor.
4. Saat kamu bicara, karakter bergerak sesuai arah suara!

Tantangan: Tambahkan kata “lompat” dan lihat apakah AI bisa menangkap tiga perintah sekaligus.

Aktivitas 6: Polisi Lalu Lintas AI!

Tujuan: Membuat simulasi mobil yang jalan atau berhenti sesuai perintah suara

Langkah Seru:

1. Buat model suara untuk mengenali kata “jalan” dan “stop”.
2. Uji AI dengan berbagai gaya bicara.

3. Bangun simulasi mobil di App Inventor: gerakkan saat “jalan”, berhenti saat “stop”.
4. Coba bermain bersama temanmu dan buat simulasi lalu lintas sederhana!

Tantangan: Tambahkan suara klakson sebagai fitur tambahan!

Semua aktivitas ini tidak hanya melatih keterampilan teknologi, tetapi juga mengajak siswa menjadi **penemu AI mini** yang belajar lewat praktik nyata, tanpa harus mengerti coding rumit. Setiap langkah bisa dieksplorasi lebih lanjut di kelas dengan diskusi kelompok, eksperimen seru, dan refleksi hasil.

Fakta Menarik / Fun Facts

- Asisten suara di HP kalian (Siri, Google Assistant) adalah contoh nyata AI pendeteksi suara yang sangat canggih! Teknologi ini terus berkembang pesat.

- **Senyum Itu "Data"!**
  Buat AI, senyum itu bukan hanya ekspresi manis—tapi data visual. Semakin banyak foto senyum dan cemberut yang kamu latih, semakin "cerdas" AI dalam membedakannya.

- **AI Tidak Pintar Sendiri!**
  AI tidak bisa "tahu" apa-apa sejak awal. Mereka seperti bayi baru lahir yang belajar dari contoh-contoh yang kamu berikan. Jadi, kamulah gurunya!

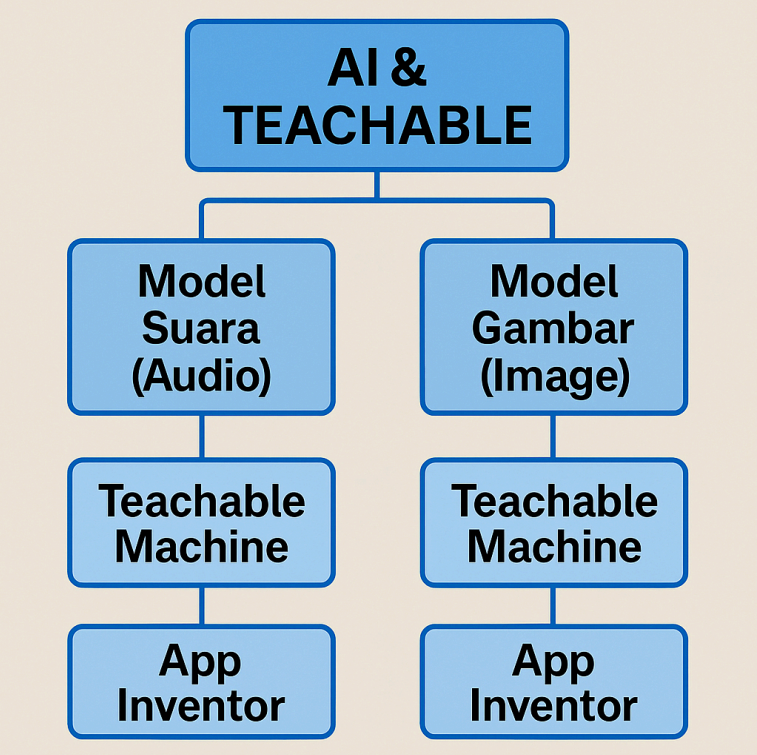
- **Tanpa Coding, Bisa AI!**
  Dulu, melatih AI butuh kode yang rumit. Tapi sekarang, dengan Teachable Machine, kamu hanya perlu klik, rekam, dan latih. Siapa bilang harus jago ngoding dulu?

Tips Belajar atau Tips Cepat Menghafal

Tips Mengingat:

- “Suara dan gambar bisa jadi input AI.”
- “Semakin banyak data, semakin pintar AI.”

Mind Mapping Singkat:



Cerita Singkat:

> Suatu hari, Dika ingin membuat aplikasi untuk menyapa gurunya. Dia merekam suara “Halo Bu” dalam 10 gaya berbeda: pelan, cepat, tertawa, serius, dan sebagainya. Setelah itu, AI-nya bisa menyapa gurunya dengan benar setiap kali Dika berbicara, bahkan saat flu! Itulah kekuatan data yang beragam.

Rangkuman

Bayangkan kamu bisa membuat komputer mengenali **suara, gambar, atau gerakan tubuhmu**—semudah merekam video atau mengambil foto, tanpa perlu menulis satu baris kode! Nah, semua itu bisa kamu lakukan lewat *Teachable Machine*, alat buatan Google yang memungkinkan kita membuat **model AI (Artificial Intelligence)** hanya dengan **klik, rekam, dan latih**. Dalam bab ini, kamu akan menjadi "pelatih AI", mengajari komputer agar bisa *mengenali ekspresi wajah, mendengar perintah suara, bahkan membaca pose tubuh*. Seru banget, kan? Seperti jadi pesulap teknologi!

Tapi tunggu dulu, biar nggak cuma seru-seruan, kamu juga perlu kenalan dulu dengan **dasar-dasar pembelajaran mesin (machine learning)**. Kamu akan belajar tentang **data latih**, yaitu contoh-contoh gambar atau suara yang kamu kumpulkan; **model AI**, yaitu hasil dari pelatihan yang jadi "otak" si komputer; dan **prediksi**, yaitu bagaimana AI menebak apa yang sedang terjadi berdasarkan data yang sudah dipelajari. Dari sini, kamu bisa

membandingkan cara kerja **model suara** dan **model gambar/pose**. Masing-masing punya cara melatih dan jenis input yang berbeda, tapi sama-sama asyik dan penuh tantangan!

Setelah itu, kamu akan diajak membuat berbagai **aplikasi interaktif** yang keren: dari **kalkulator suara**, **detektor ekspresi wajah**, sampai **game kontrol gerakan tubuh**. Kamu bisa bilang “jalan” supaya mobil bergerak, atau cukup senyum untuk memunculkan emoji lucu! Setiap proyek akan menantang kreativitasmu, sekaligus memperkenalkan bagaimana *AI bisa diajak kerjasama* dalam dunia nyata. Jadi, siapkah kamu menjadi pencipta teknologi masa depan—tanpa harus jago ngoding?

Latihan Soal

A. Benar atau Salah (5 Soal)

1. Teachable Machine hanya dapat digunakan oleh orang yang bisa coding.
   Jawaban: Salah

2. Data latihan adalah contoh-contoh yang digunakan untuk mengajari AI.
   Jawaban: Benar

3. Model AI akan belajar sendiri tanpa perlu diberikan data.
   Jawaban: Salah

4. AI bisa digunakan untuk mengenali suara maupun gambar.
   Jawaban: Benar

5. Model pose tubuh tidak membutuhkan variasi dalam pencahayaan.
   Jawaban: Salah

B. Pilihan Ganda (10 Soal)

6. Teachable Machine dikembangkan oleh:
   A. Microsoft
   B. Apple
   C. Google
   D. Intel
   Jawaban: C

7. Setelah melatih model di Teachable Machine, model akan:
   A. Merekam suara baru
   B. Menyimpan foto
   C. Membuat prediksi
   D. Menghapus data
   Jawaban: C

8. Apa yang harus dilakukan sebelum melatih model AI?
   A. Menebak hasilnya
   B. Menghapus semua data
   C. Mengumpulkan data latihan
   D. Menyusun paragraf
   Jawaban: C

9. Apa itu model AI?
   A. Data suara mentah
   B. Program belajar manual
   C. Hasil dari pelatihan data
   D. Format video
   Jawaban: C

10. Yang merupakan contoh penggunaan model suara adalah:
    A. Mengenali ekspresi wajah
    B. Menebak arah pandangan
    C. Mengendalikan karakter lewat suara
    D. Mengedit gambar
    Jawaban: C

11. Mengapa penting untuk menggunakan berbagai pose saat melatih model gambar?
    A. Supaya AI cepat bingung
    B. Supaya tampilannya lucu
    C. Supaya model mengenali variasi
    D. Supaya data lebih sedikit
    Jawaban: C

12. Apa yang terjadi jika data pelatihan terlalu sedikit?
    A. Model akan lebih pintar
    B. Prediksi jadi kurang akurat
    C. Latihan lebih seru
    D. Tidak ada pengaruhnya
    Jawaban: B

13. Fungsi utama format TensorFlow.js adalah:
    A. Menyimpan gambar
    B. Menyimpan lagu
    C. Menyimpan model AI untuk digunakan di web
    D. Membuat spreadsheet
    Jawaban: C

14. Untuk membuat model deteksi gerakan tubuh, kita harus:
    A. Menggunakan rekaman suara
    B. Menggunakan gambar tubuh dalam berbagai pose
    C. Menggunakan angka
    D. Mengedit video

Jawaban: B

15. Suara yang sama tetapi direkam dalam kondisi berbeda disebut:
    A. Data ganda
    B. Data suara mentah
    C. Variasi data
    D. Input dasar
    Jawaban: C

C. Isian Singkat (5 Soal)

16. Teachable Machine memungkinkan kita melatih model AI tanpa menulis _____.
    Jawaban: kode

17. Data yang digunakan untuk melatih AI disebut _____.
    Jawaban: data latihan

18. Hasil akhir dari pelatihan AI adalah sebuah _____.
    Jawaban: model

19. Gambar yang diambil dari berbagai sudut dan ekspresi disebut data yang _____.
    Jawaban: bervariasi

20. Untuk mengenali suara, kita harus membuat _____ suara terlebih dahulu.
    Jawaban: rekaman

D. Soal Eksplorasi (3 Soal)

Soal:

Bayangkan kamu ingin membuat aplikasi AI yang bisa mengenali dua perintah: "lompat" dan "berhenti". Jelaskan langkah-langkahmu dari awal hingga model bisa digunakan dalam aplikasi. Mulai dari memilih proyek suara di Teachable Machine, proses merekam suara yang bervariasi, melatih model, menguji hasilnya, dan mengunduh model dalam format yang sesuai.

Soal:

Kamu ingin AI mengenali ekspresi wajah temanmu: senyum dan cemberut. Rencanakan proses pelatihannya. Tuliskan bagaimana kamu mengumpulkan gambar senyum dan cemberut dari berbagai sudut dan pencahayaan, melatih model menggunakan gambar-gambar itu, lalu menguji hasilnya agar bisa dipakai dalam aplikasi yang menampilkan emoji sesuai ekspresi.

Soal:

Model AI kamu sudah bisa mengenali suara "kiri" dan "kanan". Sekarang kamu ingin membuat aplikasi dengan App Inventor agar karakter di layar bergerak mengikuti arah suara. Ceritakan komponen apa saja yang kamu perlukan di App Inventor, seperti tombol, label, dan fitur AI Extension, lalu gambarkan ide kode bloknya agar model suara bisa memberi perintah langsung ke objek di layar.

Tugas Proyek

Proyek 1: Komandan Suara – Kontrol Game Pakai Suara!

Deskripsi Proyek:
Siswa membuat model AI suara yang bisa mengenali kata “lompat” dan “berhenti” untuk mengendalikan karakter dalam game buatan sendiri.

Langkah-Langkah Pelaksanaan:

1. **Buka Teachable Machine** dan pilih *Audio Project*.
2. **Rekam suara**: ucapkan “lompat” dan “berhenti” sebanyak mungkin dengan berbagai nada.
3. **Latih model suara** hingga akurat.
4. **Ekspor model ke TensorFlow.js**.
5. **Gunakan App Inventor** untuk membuat game dan hubungkan kontrol suara ke karakter game.
6. Uji aplikasi: pastikan karakter *respon* terhadap perintah suara.

Nilai Plus: Tantangan ini mengasah kreativitas sekaligus membuat game seru yang dikendalikan tanpa tombol!

Proyek 2: Detektor Ekspresi – Wajahmu Bicara!

Deskripsi Proyek:
Siswa membuat model gambar yang bisa mengenali ekspresi wajah (senyum dan cemberut), lalu menampilkan emoji atau gambar sesuai ekspresi.

Langkah-Langkah Pelaksanaan:

1. **Pilih Image Project** di Teachable Machine.
2. **Ambil foto wajah** dengan ekspresi senyum dan cemberut, dari berbagai sudut dan pencahayaan.
3. **Latih model** hingga bisa mengenali ekspresi dengan baik.
4. **Ekspor model**.
5. **Bangun aplikasi di App Inventor**: ketika AI mendeteksi senyum → tampilkan gambar bunga; saat cemberut → tampilkan gambar awan.
6. Uji ekspresi langsung di depan kamera!

Nilai Plus: Siswa belajar bahwa ekspresi wajah juga bisa menjadi *data* untuk AI.

Proyek 3: Kalkulator Suara – Hitung Pakai Kata!

Deskripsi Proyek:
Siswa menciptakan aplikasi kalkulator sederhana yang merespons kata “plus” dan “minus” untuk menambah atau mengurangi angka di layar.

Langkah-Langkah Pelaksanaan:

1. **Pilih Audio Project** dan rekam suara “plus” dan “minus” dengan beragam intonasi.
2. **Latih model hingga akurat**.
3. **Ekspor model** dan hubungkan ke aplikasi kalkulator di App Inventor.
4. Gunakan **logika IF-ELSE**: jika suara “plus”, maka angka bertambah 1; jika “minus”, angka berkurang 1.
5. Coba: ucapkan “plus” → angka naik. Ucapkan “minus” → angka turun!

Nilai Plus: Belajar logika coding dan AI dalam satu proyek interaktif.

Proyek 4: Master Pose – AI Tahu Gerakanmu!

Deskripsi Proyek:
Siswa melatih model pose untuk mengenali tiga posisi tubuh: berdiri, jongkok, dan lompat.

Langkah-Langkah Pelaksanaan:

1. **Pilih Pose Project** di Teachable Machine.
2. **Ambil foto pose**: berdiri, jongkok, dan lompat dengan latar belakang dan sudut berbeda.
3. **Latih model pose**, lalu ekspor.
4. **Buat aplikasi** yang bisa menampilkan teks nama gerakan atau animasi sesuai gerakan tubuh.
5. Uji model langsung dengan webcam untuk mengecek akurasi prediksi.

Nilai Plus: Membantu siswa memahami bagaimana gerakan tubuh bisa menjadi *input* bagi AI.

Proyek 5: Rambu Suara – Atur Lalu Lintas dengan Mulutmu!

Deskripsi Proyek:
Siswa membuat simulasi mobil yang bisa jalan dan berhenti berdasarkan perintah suara: “jalan” dan “stop”.

Langkah-Langkah Pelaksanaan:

1. Pilih Audio Project, lalu rekam suara “jalan” dan “stop” dalam berbagai kondisi.
2. Latih model suara, lalu ekspor.
3. Bangun aplikasi mini di App Inventor dengan sprite mobil.

4. Buat logika: jika AI mendengar “jalan” → sprite bergerak; “stop” → sprite berhenti.
5. Tes simulasi: apakah mobil bergerak sesuai komando?

Nilai Plus: Menggabungkan konsep AI dengan simulasi dunia nyata (transportasi).

Proyek 6: Robot Navigator – Suara Jadi Setir!

Deskripsi Proyek:
Siswa menciptakan robot virtual di layar yang bisa belok kiri dan kanan sesuai perintah “kiri” dan “kanan”.

Langkah-Langkah Pelaksanaan:

1. **Latih model suara** di Teachable Machine: ucapkan “kiri” dan “kanan” berulang kali.
2. **Ekspor model**, lalu siapkan aplikasi dengan sprite robot.
3. Tambahkan **kontrol arah berbasis suara** di App Inventor.
4. Pastikan robot hanya belok jika mendengar perintah dengan jelas.

Nilai Plus: Membangun dasar kontrol berbasis suara yang relevan untuk dunia otomasi dan robotika.

Semua proyek ini tidak hanya mengasah logika dan teknologi, tetapi juga membangun rasa percaya diri siswa dalam menciptakan sesuatu yang *canggih* dan *bermakna*. Mereka akan menjadi pelatih AI sejati—tanpa perlu menulis kode, cukup dengan ide kreatif dan semangat eksplorasi!

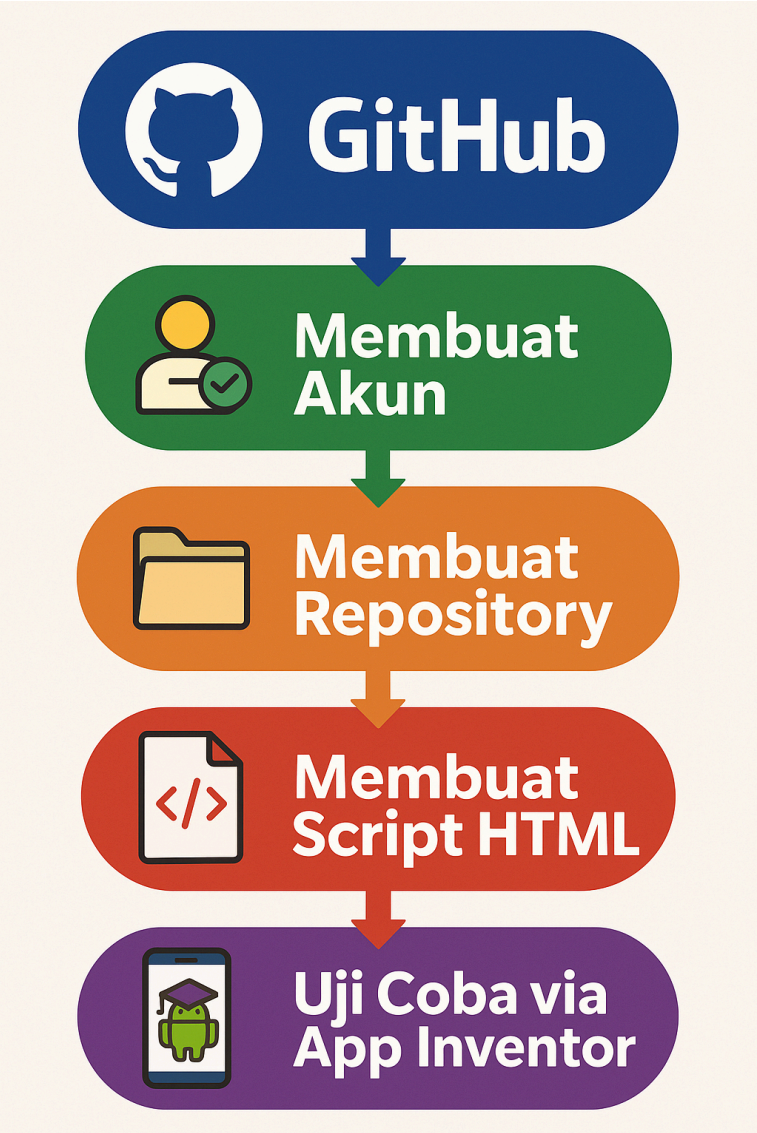
BAB 2: Jadi Kreator Digital! Hosting Script dengan GitHub

Tujuan Pembelajaran

Setelah mempelajari bab ini, kamu diharapkan dapat:

- Menjelaskan apa itu GitHub dan manfaatnya.
- Membuat akun GitHub sendiri.
- Membuat repository (proyek penyimpanan script).
- Membuat dan mengunggah script HTML sederhana.
- Menguji script menggunakan App Inventor dan WebViewer.

Peta Konsep



Apersepsi

Pernahkah kamu membuat proyek digital seperti game atau halaman web sederhana dan ingin menunjukkannya ke teman-temanmu? Atau pernah terpikir untuk menyimpan hasil karyamu di internet agar bisa dibuka kapan saja, dari mana saja?

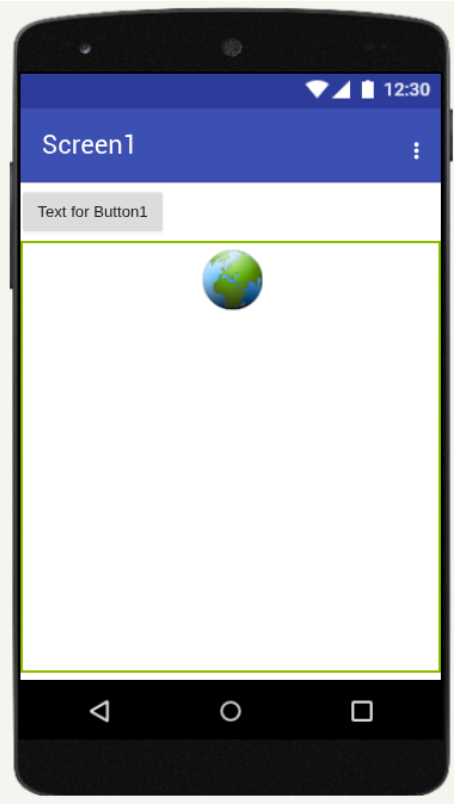
Nah, melalui bab ini kamu akan belajar menggunakan GitHub — alat keren yang dipakai oleh para programmer profesional di seluruh dunia. Kamu juga akan belajar menghubungkannya dengan App Inventor supaya bisa membuat aplikasi sendiri!

Siap jadi kreator digital sejati? Yuk mulai!

Penjelasan Konsep (teori)

Webviewer di AppInventor

WebViewer adalah komponen di App Inventor yang digunakan untuk menampilkan halaman web langsung di dalam aplikasi, seperti membuka Google tanpa harus keluar ke browser. Komponen ini sangat berguna untuk membuat aplikasi yang membutuhkan akses informasi dari internet, seperti portal berita, pencarian, atau tampilan website sekolah. Dengan WebViewer, siswa dapat belajar cara menghubungkan aplikasi dengan dunia online secara mudah dan praktis.



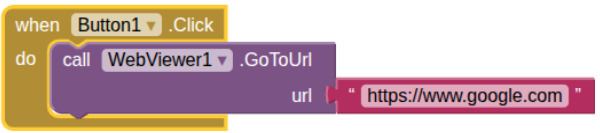
Langkah 1: Desain Tampilan Aplikasi

Pada tahap awal, buka **MIT App Inventor** dan buat sebuah proyek baru. Di tampilan **Designer**, tambahkan komponen berikut:

- **Button** dengan label default "Text for Button1".
- **WebViewer**, letakkan di bawah tombol. WebViewer ini akan menampilkan halaman web.
- Opsional: tambahkan gambar globe (ikon bumi) sebagai hiasan.

WebViewer disarankan diberi ukuran **Fill Parent** atau tinggi yang cukup agar halaman web terlihat jelas saat dijalankan.

Langkah 2: Pemrograman Blok

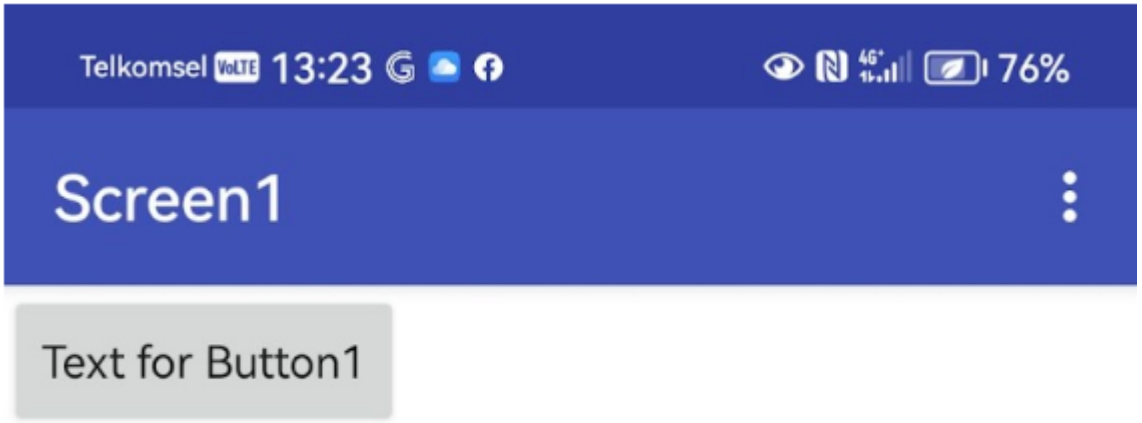


Masuk ke tab **Blocks**. Tambahkan blok berikut:

- Blok `when Button1.Click`
- Di dalamnya, tambahkan blok `call WebViewer1.GoToUrl`
- Isi URL dengan `"https://www.google.com"`

Artinya, ketika tombol ditekan, aplikasi akan membuka halaman Google melalui komponen WebViewer.

Langkah 3: Menjalankan Aplikasi

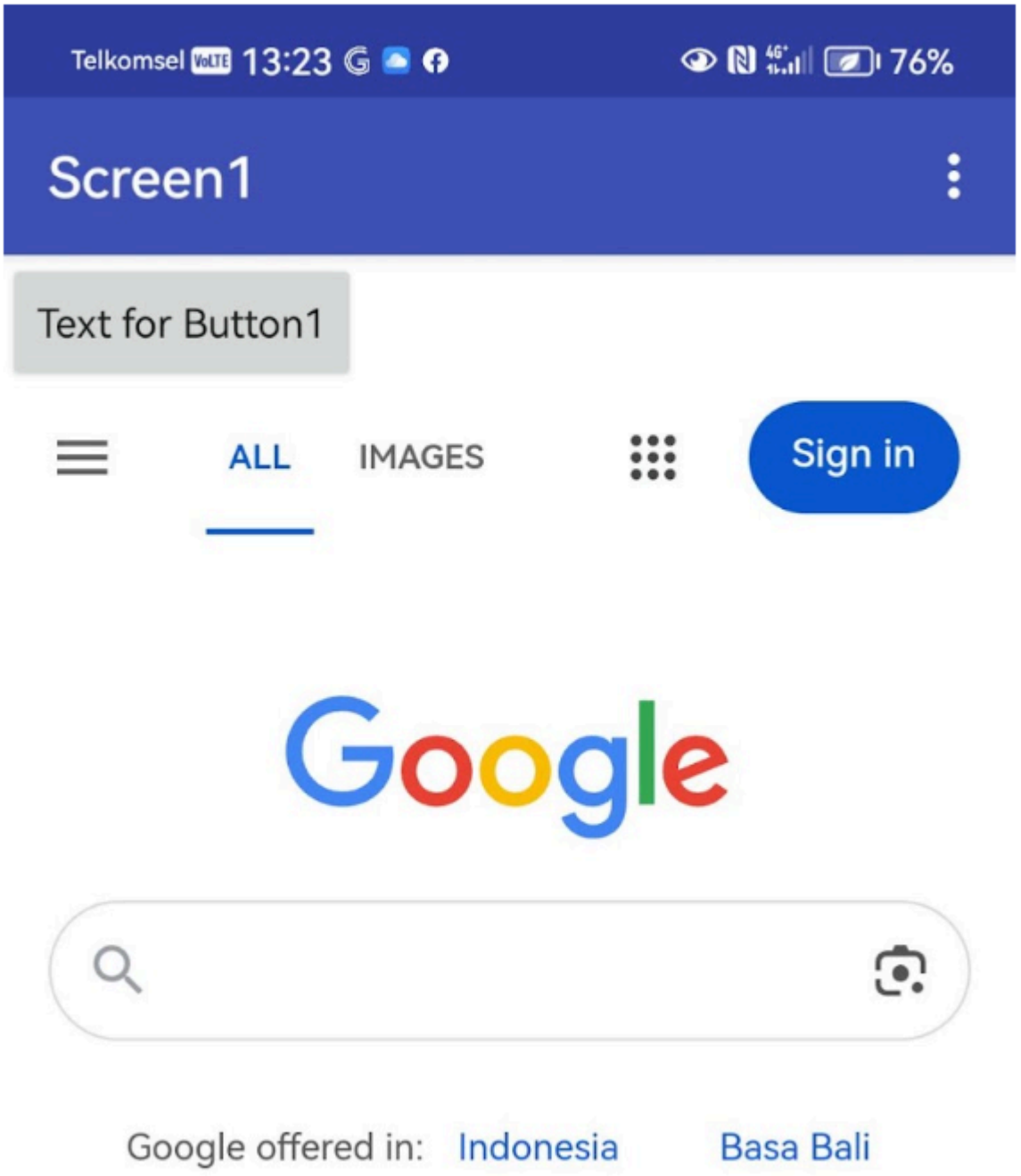


Setelah aplikasi dikompilasi atau diuji menggunakan companion App Inventor, tampilan awal akan menunjukkan:

- Sebuah tombol bertuliskan "Text for Button1"
- WebViewer masih kosong karena belum ada aksi yang dijalankan.

Langkah 4: Menampilkan Halaman Web

Setelah tombol ditekan, WebViewer akan otomatis membuka halaman Google sesuai perintah pada blok sebelumnya. Kini, pengguna dapat menjelajah Google langsung dari dalam aplikasi.



Tutorial ini memperlihatkan cara cepat dan mudah menggunakan WebViewer di App Inventor untuk menampilkan halaman web. Dengan hanya satu tombol dan satu blok pemrograman, kamu sudah bisa membuka situs seperti Google langsung dari aplikasimu.

Cocok untuk pemula yang ingin belajar interaksi aplikasi dengan web.

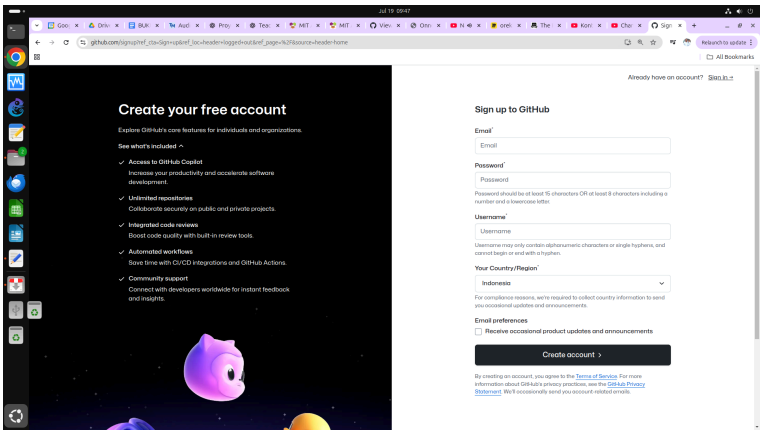
Apa Itu GitHub dan Mengapa Penting?

Bayangkan kamu punya sebuah karya, seperti cerita, gambar, atau program komputer, dan kamu ingin menyimpannya agar tidak hilang, bisa dibuka kapan saja, dan bahkan dibagikan ke temanmu. Nah, **GitHub adalah tempat penyimpanan online khusus untuk script atau kode program**. Banyak orang di seluruh dunia menggunakan GitHub untuk:

- Menyimpan kode mereka dengan aman di internet.
- Berbagi proyek mereka dengan orang lain.
- Bekerja sama membuat program secara tim, walau dari tempat yang berbeda.
- Menjadikan proyek mereka bisa diakses oleh aplikasi atau website lain.

GitHub **sangat penting** terutama bagi programmer dan pelajar karena **mudah digunakan dan bisa diakses dari mana saja**, bahkan dari HP sekalipun!

Langkah-Langkah Membuat Akun GitHub



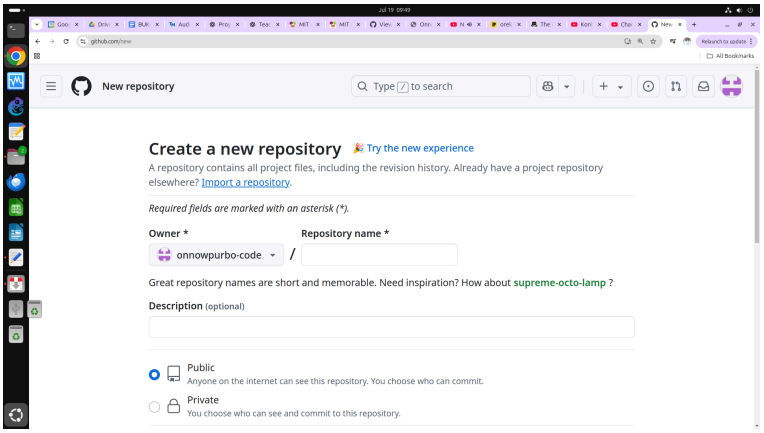
Untuk mulai menggunakan GitHub, kamu perlu punya akun terlebih dahulu. Tenang, caranya sangat mudah, bahkan bisa langsung menggunakan akun Google yang sudah kamu miliki!

Berikut langkah-langkahnya:

1. Buka situs https://github.com.
2. Klik tombol **Sign Up** di pojok kanan atas.
3. Pilih opsi **Continue with Google**.
4. Login dengan akun Google kamu seperti biasa.
5. Setelah berhasil login, kamu akan diminta memilih:
    - Nama pengguna (username),
    - Preferensi email (bisa diatur nanti),
    - Beberapa pengaturan akun awal.
6. Ikuti instruksi yang muncul di layar (biasanya singkat dan mudah).

7. Selesai! Kamu sekarang sudah punya akun GitHub yang terhubung dengan akun Google-mu.

Cara Membuat Project Baru (Repository)



Setelah punya akun, saatnya membuat "tempat" untuk menyimpan script-mu. Di GitHub, tempat ini disebut **repository**. Anggap saja seperti folder khusus di awan (cloud).

Langkah-langkahnya:

1. Masuk ke akun GitHub kamu.
2. Klik tombol **New Repository**.
3. Isi nama project kamu (contoh: `ScriptWebViewer`).
4. Tambahkan deskripsi singkat (opsional).
5. Pilih opsi **Public** (biar bisa diakses semua orang).
6. Klik **Create Repository**.

Sekarang kamu punya tempat untuk menyimpan script buatanmu!

Membuat Script index.html di Repository Github

Setelah repository dibuat, kita dapat mengisi file HTML untuk percobaan kita.

Langkah-langkahnya:

1. Masuk ke URL ScriptWebViewer, misalnya, https://github.com/onnowpurbo-code/ScriptWebViewer
2. Klik “creating a new file”
3. Isi nama file “index.html”
4. Isi dengan text HTML berikut

```
<!DOCTYPE html>
<html>
```

```
<head>
  <title>Hello World!</title>
</head>
<body>
  <h1>Selamat datang di halaman web pertamaku!</h1>
</body>
</html>
```

Klik “Commit Changes” setelah yakin tidak ada yang salah.

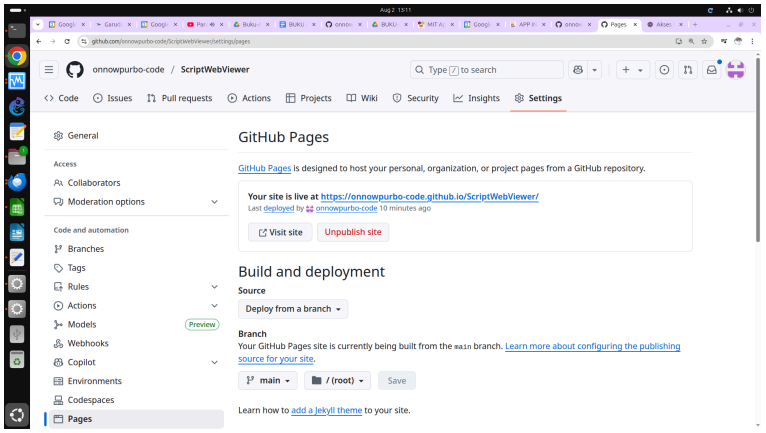
Aktifkan URL di Github:

Langkah-langkahnya:

1. Masuk ke repository di GitHub.
2. Klik tab **Settings**.
3. Cari bagian **"Pages"** (di sidebar atau dalam sub-menu "Code and automation").
4. Di bagian **"Source"**, pilih branch (misalnya `main`) dan folder (`/root` atau `/docs`) tempat file HTML kamu berada.
5. Klik **Save**.
6. Setelah itu GitHub akan menyediakan URL, seperti:

   https://onnowpurbo-code.github.io/ScriptWebViewer/

   Tampak pada screenshot Github Setting berikut:



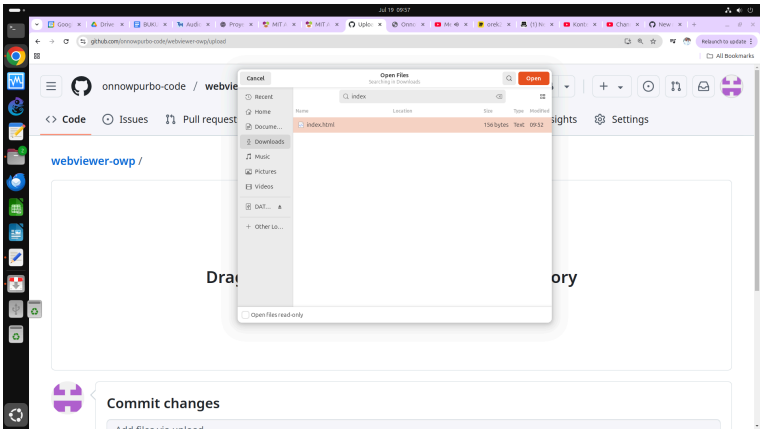
(Alternatif) Membuat Script Sederhana untuk Diunggah

Jika kamu ingin membuat file index.html dari komputer sendiri. Kita dapat mengikuti langkah ini. Sebelum mengunggah ke GitHub, kamu perlu punya script sederhana dulu. Contohnya, kamu bisa membuat file bernama `index.html` dengan isi seperti ini:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World!</title>
  </head>
  <body>
    <h1>Selamat datang di halaman web pertamaku!</h1>
  </body>
</html>
```

File ini bisa kamu buat dengan Notepad, lalu disimpan sebagai `index.html`.

Upload Script ke GitHub



Cara mengunggah script:

1. Buka repository yang tadi kamu buat.
2. Klik **Add file** > **Upload files**.
3. Seret file `index.html` ke halaman.
4. Klik **Commit changes**.

Tada! Script kamu sekarang sudah tersimpan online.

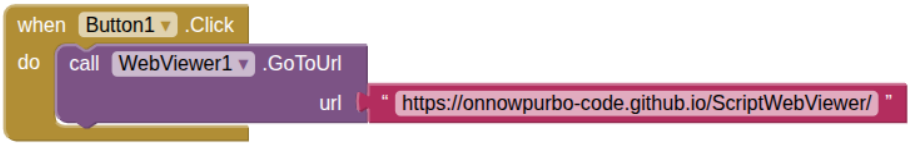
Uji Coba dengan App Inventor: WebViewer

Sekarang saatnya menguji script buatanmu lewat aplikasi! Gunakan **MIT App Inventor**, lalu buat aplikasi sederhana dengan **WebViewer**.

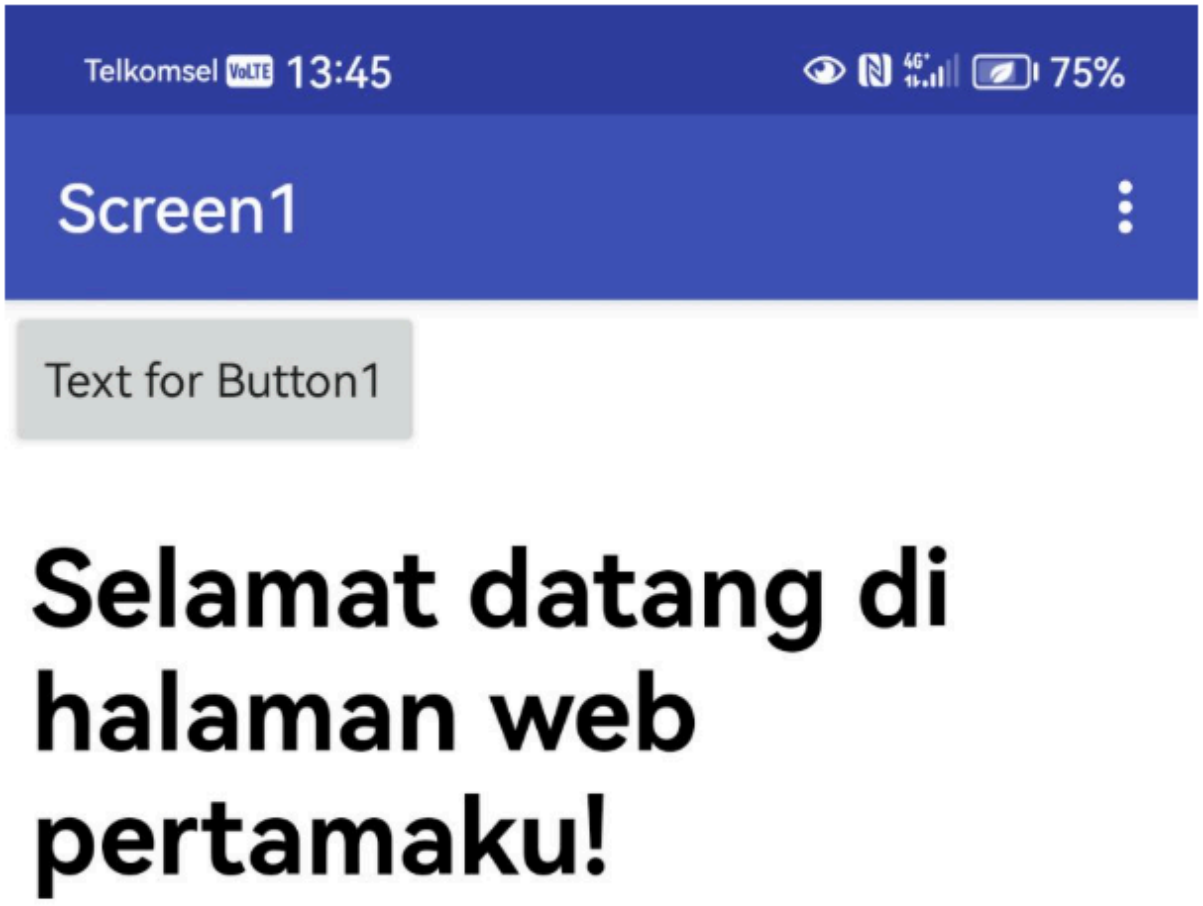
Langkah ringkas:

1. Buka https://ai2.appinventor.mit.edu.
2. Buat proyek baru.
3. Tambahkan komponen **WebViewer** ke layar aplikasi.
4. Pada pengaturan WebViewer, masukkan alamat GitHub Pages dari script kamu. Misalnya https://onnowpurbo-code.github.io/ScriptWebViewer/
5. Jalankan aplikasi. Jika berhasil, halaman web buatanmu akan tampil di layar!

6. Penampakan contoh source code app inventor bisa seperti berikut:



Screenshot layar HP setelah di jalankan di android adalah sebagai berikut



Kesimpulan

Dengan GitHub, kamu bisa:

- Menyimpan dan membagikan script-mu dengan mudah,
- Belajar bekerja seperti programmer profesional,
- Menguji hasil karyamu lewat aplikasi buatan sendiri.

Menarik, kan? Sekarang, tantang dirimu untuk membuat script yang lebih keren dan bereksperimen dengan App Inventor. Yuk, jadi kreator digital sejak dini! 💻🚀

Contoh Kasus atau Ilustrasi

Kasus 1: “Website Klub Rahasia”

Ilustrasi:
Bayangkan kamu dan teman-temanmu membentuk klub rahasia bernama *The Digital Explorers*. Kalian ingin membuat halaman web berisi kode rahasia, daftar anggota, dan tantangan mingguan.

Tantangannya:
Kamu perlu menyimpan halaman web klub itu agar bisa diakses dari mana saja. Tapi… kamu tidak mau filenya hilang atau rusak!

Solusi:
Gunakan GitHub untuk menyimpan file `index.html` berisi halaman informasi tentang klub. Dengan begitu, semua anggota bisa membuka halaman itu hanya dengan klik link!

Konsep yang Dipelajari:

- Membuat akun GitHub
- Membuat repository
- Mengunggah file HTML

Kasus 2: “Aplikasi Buatan Sendiri untuk Menampilkan Web Profilmu”

Ilustrasi:
Bayangkan kamu punya web tentang dirimu sendiri—hobimu, impianmu, dan video favoritmu. Tapi kamu ingin web itu bisa dibuka lewat aplikasi buatanmu sendiri.

Tantangannya:
Bisakah kamu membuat aplikasi Android sederhana yang menampilkan halaman HTML dari GitHub?

Solusi:
Gunakan App Inventor dan komponen WebViewer. Sambungkan ke GitHub Pages kamu. Boom! Aplikasi Android-mu bisa menampilkan halaman web buatanmu sendiri.

Konsep yang Dipelajari:

- WebViewer di App Inventor
- Menautkan HTML dari GitHub Pages

Kasus 3: “Komik Digital yang Bisa Dibuka di HP”

Ilustrasi:
Kamu hobi menggambar dan bikin komik. Kamu ingin komikmu bisa dibaca dari HP siapapun—bahkan tanpa install apa pun!

Tantangannya:
Bagaimana caranya membuat halaman komik digital dan membagikannya ke teman?

Solusi:
Scan atau gambar komik, tempelkan ke halaman HTML, lalu unggah ke GitHub. Buat link-nya jadi GitHub Pages. Tinggal buka, dan komikmu terbaca seperti e-book!

Konsep yang Dipelajari:

- HTML dengan gambar
- Hosting dengan GitHub Pages

Kasus 4: “Aplikasi Bicara yang Paham Suaramu”

Ilustrasi:
Bayangkan kamu membuat aplikasi pintar yang bisa merespons saat kamu bilang "halo" atau "bye". Aplikasi ini akan menampilkan pesan dan efek animasi. Seru banget, kan?

Tantangannya:
Bisa nggak ya bikin aplikasi Android yang bisa mendengar suara dan merespons?

Solusi:
Gunakan Teachable Machine untuk melatih suara “halo” dan “bye”. Sambungkan ke HTML, unggah ke GitHub, dan tampilkan di App Inventor pakai WebViewer.

Konsep yang Dipelajari:

- AI dasar dengan suara
- Integrasi GitHub dan App Inventor

Kasus 5: “Game Kendali Suara: Kiri–Kanan!”

Ilustrasi:
Kamu ingin membuat game sederhana di mana karakter bisa bergerak ke kiri atau ke kanan saat kamu mengucapkan perintah. Tanpa tombol—cukup dengan suaramu!

Tantangannya:
Apakah kamu bisa menggabungkan AI, HTML, dan App Inventor?

Solusi:
Melatih suara “kiri” dan “kanan” di Teachable Machine, buat tampilan game dengan HTML, upload ke GitHub, lalu mainkan lewat App Inventor.

Konsep yang Dipelajari:

- Coding AI interaktif
- Script HTML untuk game sederhana
- WebViewer untuk uji coba game

Kasus 6: “Buku Digital Pelajaran Favorit”

Ilustrasi:
Kamu suka menulis rangkuman pelajaran dengan gaya sendiri. Daripada hanya di buku, kamu ingin menjadikannya seperti e-book yang bisa dibaca online!

Tantangannya:
Bisa nggak ya membuat buku digital sendiri dan membaginya dengan teman?

Solusi:
Gunakan HTML untuk menulis bab per bab. Tambahkan gambar, link antar bab, lalu upload ke GitHub. Dengan GitHub Pages, buku digitalmu bisa dibaca siapa saja.

Konsep yang Dipelajari:

- HTML untuk navigasi bab
- GitHub sebagai perpustakaan digital
- Desain web edukatif

Kesimpulan

Setiap kasus ini:

- Mengajak siswa **berpikir kreatif**, seperti menyusun cerita atau membangun sesuatu.
- Menyisipkan **tantangan teknis yang menyenangkan**.
- Memperkenalkan **konsep pemrograman, AI, dan aplikasi** tanpa menakut-nakuti.
- Bisa digunakan sebagai bahan diskusi, simulasi, atau tantangan kelas.

Contoh Soal dan Pembahasan

Soal 1: Apa Itu GitHub?

Pertanyaan:
Bayangkan kamu membuat sebuah game sederhana dan ingin menyimpannya agar bisa dibuka kapan saja. Platform apakah yang cocok untuk menyimpan proyekmu secara online dan bisa dibagikan ke teman?

A. Microsoft Word
B. GitHub
C. Instagram
D. Canva

Pembahasan:
Jawaban yang benar adalah **B. GitHub**.
GitHub adalah platform penyimpanan kode program secara online. Dengan GitHub,

kamu bisa menyimpan, membagikan, dan bahkan menghubungkan proyekmu dengan aplikasi lain. Ini seperti “lemari digital” khusus untuk karya-karya digitalmu, terutama script dan kode.

Soal 2: Membuat Repository Pertamamu

Pertanyaan:
Setelah membuat akun GitHub, kamu ingin menyimpan file `index.html` milikmu. Apa langkah pertama yang harus kamu lakukan?

A. Menghapus akun GitHub
B. Mengunduh App Inventor
C. Membuat repository baru
D. Menambahkan efek animasi

Pembahasan:
Jawaban yang benar adalah **C. Membuat repository baru**.
Repository ibarat "folder penyimpanan online" di GitHub untuk menaruh script kamu. Langkah ini penting supaya GitHub tahu kamu sedang menyimpan proyek baru.

Soal 3: Memahami Isi Script HTML

Pertanyaan:
Perhatikan potongan script berikut ini:

```
<h1>Selamat datang di halaman web pertamaku!</h1>
```

Apa fungsi dari tag `<h1>` dalam HTML tersebut?

A. Menambahkan gambar
B. Membuat tombol
C. Menampilkan judul besar di halaman web
D. Menyembunyikan teks

Pembahasan:
Jawaban yang benar adalah **C. Menampilkan judul besar di halaman web**.
Tag `<h1>` digunakan untuk membuat judul utama (heading terbesar) dalam sebuah halaman HTML. Dalam kasus ini, teks "Selamat datang di halaman web pertamaku!" akan tampil besar dan jelas di browser.

Soal 4: Upload Script ke GitHub

Pertanyaan:
Kamu sudah membuat file `index.html`. Langkah berikutnya untuk menyimpannya di GitHub adalah…

A. Menyalin file ke flashdisk
B. Klik tombol "Add file" lalu pilih "Upload files"

C. Kirim lewat email ke GitHub
D. Cetak file dan simpan di laci

Pembahasan:
Jawaban yang benar adalah **B. Klik tombol "Add file" lalu pilih "Upload files"**.
Setelah kamu berada di repository milikmu, kamu bisa mengunggah file dengan klik **"Add file"**, lalu pilih **"Upload files"** dan unggah file HTML buatanmu.

Soal 5: Menguji Script dengan App Inventor

Pertanyaan:
Kamu ingin menampilkan halaman web dari GitHub di aplikasi buatanmu di App Inventor. Komponen manakah yang digunakan untuk ini?

A. Sound Recorder
B. WebViewer
C. Canvas
D. TinyDB

Pembahasan:
Jawaban yang benar adalah **B. WebViewer**.
Komponen **WebViewer** berfungsi untuk menampilkan halaman web dalam aplikasi. Kamu tinggal memasukkan alamat dari GitHub Pages tempat kamu menyimpan script HTML, dan halaman webmu akan langsung muncul di aplikasi!

Soal 6: Tahu atau Tidak?

Pertanyaan:
GitHub bisa digunakan langsung dari smartphone, tanpa perlu komputer.

A. Benar
B. Salah

Pembahasan:
Jawaban yang benar adalah **A. Benar**.
GitHub bisa diakses lewat browser HP. Kamu bisa melihat, mengedit, bahkan mengunggah file dari ponsel selama terhubung internet. Praktis banget, kan?

Fakta Menarik / Fun Facts

1. **GitHub tuh kayak Instagram-nya para programmer!**
   Bedanya, yang di-posting bukan foto, tapi kode-kode keren yang bisa jadi website, game, bahkan aplikasi!

2. **File `index.html` adalah bintang utamanya website!**
   Kalau website itu drama, `index.html` itu pemeran utamanya. Semua web dimulai

dari file ini!

3. **App Inventor diciptakan oleh MIT, kampus para ilmuwan jenius dunia!**
   Yup, kampus yang sama dengan penemu roket, AI, dan robot super canggih. Sekarang kamu bisa pakai buatan mereka dari rumah!

4. GitHub pertama kali dibuat untuk programmer, tapi sekarang digunakan **oleh seniman, ilmuwan, guru, hingga pelajar!**

5. Presiden Barack Obama pernah **unggah kode** pertamanya di GitHub!

6. Banyak aplikasi populer (seperti VS Code, React, atau TensorFlow) adalah proyek **open-source GitHub**.

7. **GitHub juga dipakai untuk bikin AI, robot, dan game-game populer!**
   Banyak proyek canggih di dunia ini dimulai dari… ya, dari sebuah repository kecil di GitHub!

8. **Script kamu bisa "terbang" ke awan dengan GitHub!**
   Karena GitHub berbasis cloud, semua script-mu bisa diakses dari mana saja — bahkan dari gunung (asal ada sinyal 😄).

Kalimat-kalimat ini bisa kamu tampilkan di slide, pojok buku siswa, atau diselipkan di sela-sela latihan soal untuk menjaga semangat dan rasa ingin tahu mereka tetap tinggi! Mau dibantu juga untuk membuat stiker digital atau infografis dari fun facts ini?

Tips Belajar atau Tips Cepat Menghafal

Gunakan metode **"3S + 1C"** berikut:

1. **Singkat**: Ringkas poin penting jadi kata kunci.
2. **Spasial**: Buat gambar/mind map agar otak mudah menyimpan info.
3. **Suara**: Ucapkan ulang dengan lantang (gunakan irama).
4. **Cerita (Connection)**: Hubungkan dengan kisah nyata sehari-hari.

Teknik Bonus:

- Gunakan **akronim/lafal lucu**. Misalnya: **"GUHUREP"** untuk GitHub → *GU* = Git, *HU* = Hub, *REP* = Repository.
- Terapkan **"Belajar Aktif"**: langsung praktik > hanya baca.
- **Ajarkan kembali ke teman**. Metode *Feynman Technique*.

Mind Mapping Jadi Kreator Digital!



Cerita

Bayangkan kamu **punya komik buatan sendiri**.

> Kamu ingin teman-temanmu bisa membacanya di mana saja, kapan saja.
> Jadi kamu unggah ke GitHub — ibarat **“perpustakaan online pribadi”**.
> Lalu kamu buat **aplikasi sederhana di HP** yang bisa menampilkan komikmu.
> Saat dibuka, komik tampil di layar HP lewat **WebViewer**!

Aktivitas Siswa

Aktivitas 1: Jelajahi Dunia GitHub!

> **Tujuan:** Mengenal GitHub sebagai tempat penyimpanan karya digital.
> **Langkah Seru:**

- Ajak siswa membuka situs https://github.com dan menjelajah tampilannya.
- Diskusikan: "Kalau kamu punya karya digital (game, web, aplikasi), bagaimana GitHub bisa membantumu menyimpannya dengan aman dan membagikannya ke teman?"
- Tunjukkan proyek-proyek publik yang menarik dari pengguna lain untuk memberi inspirasi.

- Tantang siswa: "Bayangkan kamu punya proyek digital. Apa nama dan tujuan proyekmu?"

Catatan: Ini bisa dikemas seperti petualangan digital pertama—menjelajahi markas para programmer dunia.

Aktivitas 2: Buat Identitas Digital!

Tujuan: Membuat akun GitHub sendiri.
Langkah Seru:

- Ajak siswa membuat akun GitHub menggunakan akun Google mereka.
- Dorong mereka memilih username yang keren tapi tetap formal (contoh: AryaDev2025).
- Sampaikan pentingnya keamanan dan tanggung jawab sebagai pengguna platform publik.
- Ajak mereka mengisi bio sederhana: siapa kamu dan apa yang ingin kamu buat?

Tips: Anggap ini seperti membuat paspor digital sebagai kreator masa depan!

Aktivitas 3: Bangun Proyek Pertamamu!

Tujuan: Membuat repository sebagai tempat menyimpan kode.
Langkah Seru:

- Ajak siswa membuat "folder online" di GitHub, beri nama yang unik dan mencerminkan isi proyek.
- Pandu mereka mengisi deskripsi singkat—ini melatih mereka menjelaskan ide secara ringkas.
- Dorong mereka memilih opsi *Public* agar bisa pamer karya mereka ke dunia!

Analogi: Repository = lemari khusus untuk karya digitalmu. Tunjukkan lemari digitalmu dengan bangga!

Aktivitas 4: Tulis Kode Pertamamu – Katakan Halo ke Dunia!

Tujuan: Membuat dan mengunggah file HTML sederhana ke GitHub.
Langkah Seru:

- Ajak siswa menulis kode HTML "Hello World!" di Notepad atau editor teks sederhana.
- Tunjukkan bahwa mereka sedang menciptakan *halaman web* milik sendiri.
- Upload ke repository melalui GitHub dengan fitur **Add file > Upload**.

Kreatif: Tantang siswa mengganti teks di halaman jadi sapaan khas mereka atau nama proyek impian.

Aktivitas 5: Uji Kreasimu di Aplikasi Sendiri!

Tujuan: Menghubungkan hasil script dengan App Inventor menggunakan WebViewer.
Langkah Seru:

- Ajak siswa membuka App Inventor dan membuat proyek baru.
- Tambahkan komponen **WebViewer** dan isikan alamat GitHub Pages mereka.
- Jalankan aplikasi: halaman HTML yang mereka buat kini muncul di layar seperti aplikasi sungguhan!

Tantangan: Siapa yang bisa menampilkan tampilan web paling menarik dengan gaya sendiri?

Aktivitas 6: Eksperimen dengan AI & Suara!

Tujuan: Menyambungkan model suara dari Teachable Machine ke App Inventor.
Langkah Seru:

- Tunjukkan cara melatih model dengan dua suara (misalnya “Halo” dan “Bye”).
- Ajak siswa menyambungkan model ke aplikasi di App Inventor, lalu beri respons sesuai suara.
- Biarkan siswa bereksperimen dengan suara unik: bisa perintah lucu, bahasa daerah, atau efek suara.

Fun & Techy: Ini kesempatan mereka berkreasi dengan teknologi AI tanpa coding rumit.

Jika aktivitas-aktivitas ini dilakukan bertahap dan dengan suasana eksploratif, siswa tidak hanya belajar teknis tapi juga merasa bangga dengan karya digital mereka. Siap jadi kreator digital sejati? Ayo mulai petualangan coding-mu!

Rangkuman

Pernahkah kamu ingin menunjukkan proyek digitalmu seperti game atau web buatan sendiri ke teman-teman, tapi bingung cara menyimpannya secara online? Nah, di bab ini kamu akan belajar menjadi **kreator digital** dengan bantuan **GitHub**, sebuah *platform* keren yang dipakai para programmer dunia. Di sini, kamu akan belajar cara membuat akun GitHub, menyimpan *script HTML* sederhana di *repository*, lalu mengujinya melalui aplikasi buatan kamu sendiri dengan bantuan **App Inventor**. Seru kan? Ayo kita mulai petualangan digitalmu!

Setelah akun GitHub siap, kamu akan membuat *repository*, yaitu semacam folder online untuk menyimpan karya kode kamu. Lalu, kamu akan belajar membuat file HTML dasar bernama **index.html** berisi halaman web sederhana. Script ini nantinya akan di-*upload* ke GitHub agar bisa diakses siapa saja melalui internet. Ini adalah langkah awal menuju dunia pemrograman yang sesungguhnya, dan kamu bisa mengakses proyekmu kapan saja—bahkan dari HP!

Tantangan makin seru ketika kamu menguji hasil karyamu lewat **App Inventor** menggunakan komponen *WebViewer*. Dengan memasukkan alamat dari halaman web GitHub-mu, aplikasi yang kamu buat bisa menampilkan halaman tersebut. Di akhir bab, kamu bahkan bisa mencoba membuat proyek suara dengan AI yang mengenali kata seperti “kiri” dan “kanan”. Ini bukan hanya belajar teknologi, tapi kamu juga sedang melatih kemampuan berpikir logis, kreatif, dan jadi pembuat aplikasi sejak dini. **Siap jadi kreator digital sejati?**

Latihan Soal

A. Benar atau Salah (5 Soal)

1. GitHub adalah tempat penyimpanan kode yang hanya bisa diakses oleh pemiliknya.
   Jawaban: Salah

2. Repository di GitHub bisa diatur agar bisa diakses publik.
   Jawaban: Benar

3. WebViewer di App Inventor digunakan untuk melihat tampilan halaman web dari URL tertentu.
   Jawaban: Benar

4. File HTML harus disimpan dengan ekstensi `.txt` sebelum diunggah ke GitHub.
   Jawaban: Salah

5. App Inventor dapat menampilkan halaman web yang disimpan di GitHub.
   Jawaban: Benar

B. Pilihan Ganda (10 Soal)

6. Apa fungsi utama GitHub?
   A. Membuat animasi
   B. Mengedit foto
   C. Menyimpan dan membagikan kode program
   D. Mengunduh aplikasi
   Jawaban: C

7. Di mana kamu bisa membuat repository baru di GitHub?
   A. Menu File
   B. Halaman utama akun
   C. Menu View
   D. Menu Tools
   Jawaban: B

8. Ekstensi file HTML yang benar adalah:
   A. .docx
   B. .html
   C. .jpg
   D. .exe
   Jawaban: B

9. Setelah membuat repository, langkah selanjutnya untuk menyimpan script adalah:
   A. Klik Upload Video
   B. Klik Add file > Upload files
   C. Klik Edit Gambar
   D. Klik Play
   Jawaban: B

10. Script HTML sederhana biasanya dimulai dengan tag:
    A. `<script>`
    B. `<html>`
    C. `<code>`
    D. `<img>`
    Jawaban: B

11. Fungsi dari tag `<h1>` dalam HTML adalah:
    A. Membuat daftar
    B. Menambahkan gambar
    C. Menampilkan judul besar
    D. Membuat tautan
    Jawaban: C

12. App Inventor bisa diakses melalui:
    A. www.github.com
    B. www.notepad.com
    C. ai2.appinventor.mit.edu
    D. www.htmlmaker.com
    Jawaban: C

13. Mengapa perlu memilih opsi *Public* saat membuat repository?
    A. Agar tampilannya lebih menarik
    B. Agar bisa dibuka semua orang
    C. Agar file bisa dihapus otomatis
    D. Agar file tidak terlihat
    Jawaban: B

14. Komponen apa yang digunakan untuk menampilkan halaman GitHub di App Inventor?
    A. Button
    B. Label
    C. WebViewer

D. Image
Jawaban: C

15. Salah satu manfaat menggunakan GitHub adalah:
    A. Mengubah file menjadi gambar
    B. Menghapus aplikasi
    C. Menyimpan script dengan aman di internet
    D. Menyembunyikan data
    Jawaban: C

C. Isian Singkat (5 Soal)

16. Situs untuk mengakses GitHub adalah _____.
    Jawaban: https://github.com

17. Nama tempat penyimpanan script di GitHub disebut _____.
    Jawaban: repository

18. Untuk membuat file HTML, kamu bisa menggunakan aplikasi _____.
    Jawaban: Notepad

19. Tag HTML untuk menampilkan teks besar sebagai judul adalah _____.
    Jawaban: <h1>

20. Setelah mengunggah file ke GitHub, kamu harus klik tombol _____ untuk menyimpan perubahan.
    Jawaban: Commit changes

D. Soal Explorasi (3 Soal)

Soal:

Jelaskan langkah-langkah mulai dari membuat script HTML sederhana hingga berhasil menampilkannya di aplikasi buatanmu menggunakan App Inventor. Tuliskan secara urut dan jelas!

Soal:

Kamu diminta membuat halaman web sederhana bertema "Profil Sekolah" dan mengunggahnya ke GitHub. Halaman itu akan ditampilkan di aplikasi Android lewat App Inventor. Apa saja isi HTML yang kamu tulis? Bagaimana kamu menyusun tampilannya agar menarik?

Soal:

Bayangkan kamu sedang membuat proyek digital untuk memperkenalkan makanan khas daerah. Kamu ingin menampilkan halaman web tentang makanan tersebut lewat App Inventor. Buatlah rencana mulai dari pembuatan repository, pembuatan

script HTML, hingga integrasi dengan WebViewer.

Tugas Proyek

Proyek 1: Web Pribadi Pertamaku

Deskripsi:
Buatlah halaman web sederhana yang berisi perkenalan diri dan hobimu, lalu unggah ke GitHub agar bisa diakses teman-temanmu dari internet.

Langkah Saran:

1. Buat akun GitHub.
2. Buat file `index.html` di Notepad dengan isi biodata singkat, hobi, dan cita-cita.
3. Simpan dan upload file tersebut ke repository GitHub.
4. Aktifkan GitHub Pages untuk menampilkan website secara online.
5. Bagikan link web ke teman dan guru!

Tantangan tambahan: Tambahkan gambar atau tautan ke video favoritmu.

Proyek 2: Aplikasi Pemutar Web Menggunakan WebViewer

Deskripsi:
Bangun aplikasi sederhana di App Inventor yang menampilkan halaman web yang kamu buat di GitHub.

Langkah Saran:

1. Gunakan proyek dari Proyek 1 sebagai halaman yang akan ditampilkan.
2. Masuk ke MIT App Inventor dan buat proyek baru.
3. Tambahkan komponen WebViewer dan atur URL-nya ke alamat GitHub Pages kamu.
4. Jalankan aplikasi di emulator atau HP.

Tantangan tambahan: Tambahkan tombol untuk berpindah halaman jika kamu menambahkan lebih dari satu HTML.

Proyek 3: Aplikasi Perintah Suara “Halo” dan “Bye”

Deskripsi:
Latih model AI untuk mengenali suara “halo” dan “bye”, lalu tampilkan teks sapaan otomatis di layar aplikasi.

Langkah Saran:

1. Latih model suara di Teachable Machine.

2. Ekspor model sebagai TensorFlow.js.
3. Gunakan App Inventor dan WebViewer untuk menghubungkan model suara.
4. Tambahkan blok logika:
    - Jika "halo", tampilkan “Selamat Datang”.
    - Jika "bye", tampilkan “Sampai Jumpa”.

Tantangan tambahan: Tambahkan efek suara atau animasi saat perintah dikenali.

Proyek 4: Game Interaktif Suara “Kiri” dan “Kanan”

Deskripsi:
Buat game sederhana di App Inventor yang dikendalikan oleh suara menggunakan model AI dari Teachable Machine.

Langkah Saran:

1. Latih model untuk dua perintah: “kiri” dan “kanan”.
2. Buat sprite (gambar karakter) di App Inventor.
3. Gunakan logika:
    - Jika suara “kiri”, geser sprite ke kiri.
    - Jika suara “kanan”, geser sprite ke kanan.

Tantangan tambahan: Tambahkan rintangan dan skor untuk membuatnya jadi mini-game!

Proyek 5: Buku Digital Mini di GitHub

Deskripsi:
Tulis cerita pendek atau catatan pelajaranmu di HTML, lalu unggah ke GitHub agar bisa dibaca online.

Langkah Saran:

1. Tulis cerita atau ringkasan pelajaran di Notepad dengan tag HTML.
2. Tambahkan judul, paragraf, dan jika bisa, gambar ilustrasi.
3. Unggah ke GitHub dan tampilkan dengan GitHub Pages.

Tantangan tambahan: Tambahkan tombol navigasi antar bab menggunakan link internal (`<a href="#bab2">`).

Proyek 6: Uji Coba Sensor AI dengan WebViewer

Deskripsi:
Eksperimen dengan berbagai model AI buatanmu (misalnya deteksi wajah, suara, atau gambar) dan tampilkan hasilnya di App Inventor.

Langkah Saran:

1. Pilih jenis model yang ingin kamu latih di Teachable Machine.

2. Ekspor dan upload model ke GitHub.
3. Buat halaman HTML yang memuat model AI tersebut.
4. Gunakan App Inventor untuk menampilkan halaman tersebut via WebViewer.
5. Uji coba AI-mu di HP!

Tantangan tambahan: Ubah tampilan atau fungsi aplikasi sesuai jenis model.

Semua proyek ini dirancang agar kamu **berpikir kreatif**, **berani mencoba teknologi baru**, dan merasa bangga saat karyamu **bisa dilihat secara online**.
Yuk, mulai salah satu proyek dan jadilah **kreator digital sejak SMP!**

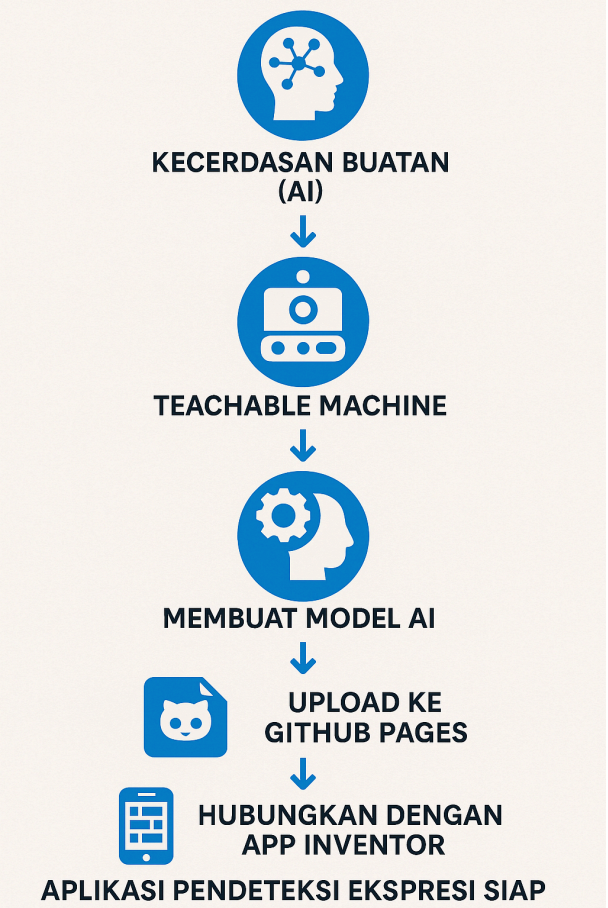
BAB 3: Aplikasi Pendeteksi Ekspresi: Membuat AI Sendiri dari Kamera HP!

Tujuan Pembelajaran

Di akhir pembelajaran ini, siswa diharapkan mampu:

1. Menjelaskan konsep dasar kecerdasan buatan (AI) yang dapat mengenali ekspresi wajah.
2. Membuat model AI sederhana menggunakan kamera HP dengan bantuan *Teachable Machine*.
3. Menghubungkan model AI ke halaman web yang dibuat sendiri.
4. Menampilkan AI tersebut di aplikasi Android melalui *MIT App Inventor*.
5. Menerapkan teknologi ini untuk mengenali ekspresi wajah atau benda di sekitar.

Peta Konsep



Apersepsi

Pernahkah kamu melihat kamera HP bisa mengenali wajahmu sedang tersenyum atau serius?
Nah, sekarang kamu bisa membuat AI sederhana seperti itu sendiri! Cukup dengan laptop, kamera HP, dan koneksi internet, kamu bisa melatih sebuah “otak kecil” yang bisa mengenali ekspresi wajahmu secara otomatis.
Seru, kan? Yuk kita coba jadi ilmuwan muda yang menciptakan teknologi sendiri! 🚀

Penjelasan Konsep (teori)

Buat Model di Teachable Machine

1. Buka: https://teachablemachine.withgoogle.com
2. Pilih **"Image Project"** → **Standard Image Model**.
3. Klik **"Add a class"** dan beri nama (misalnya: "Tersenyum", "Serius").
4. Klik **"Webcam"** dan ambil contoh gambar dari Android webcam.
5. Tambahkan beberapa gambar ke masing-masing kelas.
6. Klik **"Train Model"**.
7. Setelah selesai, klik **"Export Model"** → **"TensorFlow.js"** → klik **"Upload (hosted)"**.
8. Salin/copy **"Your sharable link"**.

Buat Halaman Web di GitHub

Kita butuh file HTML yang memuat model ini agar bisa ditampilkan di App Inventor.

Contoh isi `index.html` (script GitHub):

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Teachable Machine Webcam</title>
</head>
<body>
  <h1>Deteksi Wajah</h1>
  <button type="button" onclick="init()">Mulai Kamera</button>
  <div><canvas id="canvas"></canvas></div>
  <div id="label-container"></div>

  <script
src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@0.8/dist/teacha
blemachine-image.min.js"></script>
  <script type="text/javascript">
    const URL = "PASTE-LINK-MODEL-DI-SINI/";
    let model, webcam, labelContainer, maxPredictions;
```

```
    async function init() {
        const modelURL = URL + "model.json";
        const metadataURL = URL + "metadata.json";

        model = await tmImage.load(modelURL, metadataURL);
        maxPredictions = model.getTotalClasses();

        const flip = true;
        webcam = new tmImage.Webcam(200, 200, flip);
        await webcam.setup();
        await webcam.play();
        window.requestAnimationFrame(loop);

        document.getElementById("canvas").appendChild(webcam.canvas);
        labelContainer = document.getElementById("label-container");
        for (let i = 0; i < maxPredictions; i++) {
            labelContainer.appendChild(document.createElement("div"));
        }
    }

    async function loop() {
        webcam.update();
        await predict();
        window.requestAnimationFrame(loop);
    }

    async function predict() {
        const prediction = await model.predict(webcam.canvas);
        for (let i = 0; i < maxPredictions; i++) {
            const classPrediction = prediction[i].className + ": " +
(prediction[i].probability * 100).toFixed(2) + "%";
            labelContainer.childNodes[i].innerHTML = classPrediction;
        }
    }
  </script>
</body>
</html>
```

Ganti `PASTE-LINK-MODEL-DI-SINI/` dengan URL dari model Teachable Machine.

Upload ke GitHub Pages

1. Buat akun GitHub.
2. Buat repository baru, misalnya `teachable-web`.
3. Upload `index.html` ke repository tersebut.

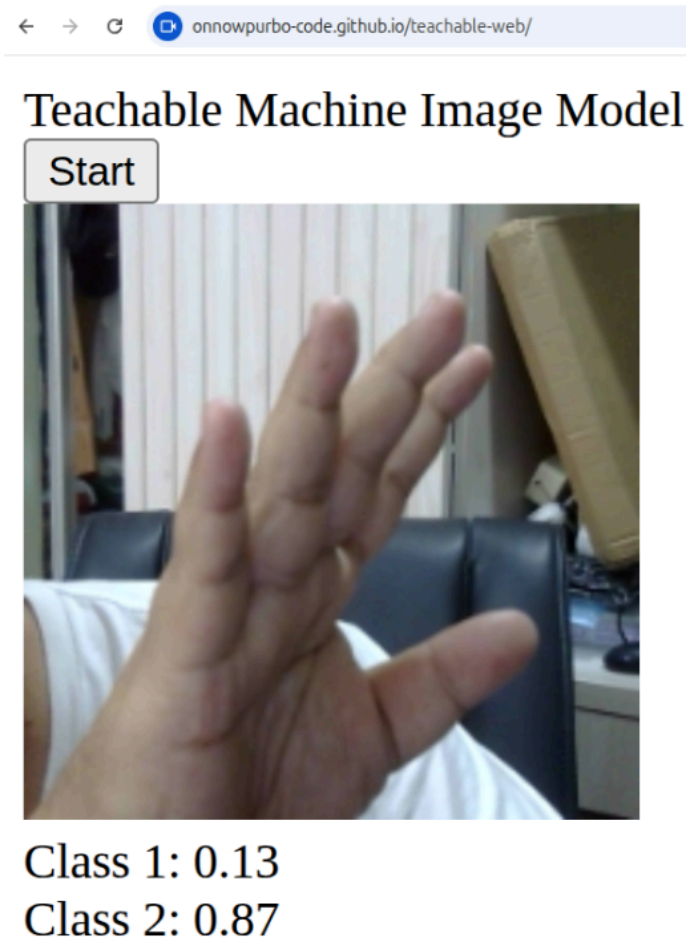
4. Aktifkan GitHub Pages: `Settings → Pages → Source → main /root → Save`.
5. Catat URL GitHub Pages, misalnya: `https://namamu.github.io/teachable-web/`

Test GitHub Page

Test terlebih dulu dengan mengakses URL di Github, misalnya:

https://namamu.github.io/teachable-web/

Harusnya akan keluar gambar berikut



Gambar di atas adalah tampilan dari sebuah program di GitHub yang menggunakan **Google Teachable Machine** untuk mengenali gambar lewat kamera. Setelah tombol “Start” ditekan, program memproses gambar tangan yang terlihat di layar dan membandingkannya dengan model yang sudah dilatih sebelumnya. Hasilnya ditampilkan dalam dua angka: **Class 1: 0.13** (13%) dan **Class 2: 0.87** (87%). Artinya, program sangat yakin — dengan tingkat keyakinan 87% — bahwa gambar tersebut termasuk ke dalam **Class 2**. Ini menunjukkan model dapat mengklasifikasikan gambar dengan presisi tinggi sesuai data latihnya.

Langkah selanjutnya adalah membuat aplikasi App Inventor Webviewer agar bisa mengakses halaman Github.

Buat App di MIT App Inventor

1. Buka: https://ai2.appinventor.mit.edu
2. Tambahkan komponen:
    - **WebViewer**
    - **Button**
3. Atur `WebViewer.Url` ke URL GitHub Pages kamu.
4. Gunakan blok:

```
when Button1.Click do
 set WebViewer1.HomeUrl to "https://namamu.github.io/teachable-web/"
```



5. Bangun (build) aplikasi dan instal di HP.

Hasil

Saat tombol ditekan, halaman web dari GitHub (yang menjalankan model) terbuka, dan langsung menggunakan **kamera Android** untuk mengenali wajah/objek sesuai yang dilatih di Teachable Machine.

Kadang kala gagal, jika izin untuk akses ke Kamera tidak diberikan. Jika itu terjadi, sebaiknya akses menggunakan Web biasa di browser saja.

Catatan untuk Siswa SMP

- Gunakan objek mudah dikenali: ekspresi wajah, warna, atau benda seperti sendok/sisir.
- Gunakan bahasa sederhana di App Inventor dan desain minimal.
- Jelaskan bahwa “model” itu seperti "otak" kecil yang belajar dari contoh.

Contoh Kasus atau Ilustrasi

Kasus 1: Detektif Ekspresi — Siapa yang Lagi Bohong?

Ilustrasi:
Bayangkan kamu jadi detektif di kelas! Temanmu bilang dia *nggak ngantuk*, tapi ekspresinya seperti orang mau tidur. Kamu buat model AI di Teachable Machine dengan dua kelas: **"Ngantuk"** dan **"Terjaga"**, lalu latih dengan ekspresi wajah

teman-teman.
Saat diuji, AI langsung mendeteksi: "Ngantuk 95%" — ketahuan deh, bohongnya!

Tujuan tersembunyi: Melatih AI untuk membaca ekspresi wajah sebagai "sinyal diam-diam" dari perasaan seseorang.

Kasus 2: Warna Favoritku? Tanyakan ke Kamera!

Ilustrasi:
Kamu sedang main tebak warna dengan temanmu. Kamu melatih model AI untuk mengenali warna **merah, biru, dan hijau**.
Setelah itu, temanmu tunjukkan benda berwarna, dan kamera langsung menebak: "Hijau 99%"!

Keseruan: Seolah-olah kamu punya teman robot yang bisa menebak warna lebih cepat daripada manusia.

Kasus 3: Wajah Bahagia vs Wajah Serius

Ilustrasi:
Kamu penasaran, berapa kali kamu tersenyum dalam sehari?
Buat model dengan dua kelas: **"Tersenyum"** dan **"Serius"**. Latih dengan wajahmu dan teman-teman saat tersenyum dan tidak.
Setiap kali kamu buka kamera HP lewat aplikasi, AI akan menunjukkan apakah kamu sedang bahagia atau serius.

Pelajaran seru: Menyadari emosi sendiri lewat teknologi, dan belajar bagaimana komputer bisa memahami ekspresi.

Kasus 4: Robot Juri Lomba Fashion Show

Ilustrasi:
Bayangkan kamu juri dalam lomba pakaian adat. Kamu latih AI untuk mengenali **baju batik**, **seragam sekolah**, dan **kaos santai**.
Saat peserta tampil, kamera akan menampilkan prediksi: “Batik 89%” atau “Seragam 97%” — seolah jadi juri digital!

Nilai tambah: Belajar mengenali pola dan ciri visual secara otomatis menggunakan kamera.

Kasus 5: Kamera Anti Barang Tertukar

Ilustrasi:
Kamu dan temanmu sering tertukar sisir atau sendok saat piknik. Kamu melatih AI mengenali **sisir** dan **sendok** dari kameramu.
Saat AI aktif, ia langsung memberi label: “Sisir 100%” — nggak bakal tertukar lagi!

Manfaat nyata: Membuat AI jadi “pengingat visual” yang bisa bantu kehidupan sehari-hari.

Kasus 6: Alarm Wajah Marah di Rumah

Ilustrasi:
Kamu ingin tahu kapan orang tua sedang serius atau marah supaya tahu kapan harus diam.
Kamu latih AI untuk mengenali wajah dengan ekspresi **“Marah”** dan **“Netral”**. Saat orang tua lewat di depan kamera HP, AI akan menampilkan status wajah.
Kamu langsung tahu: "Oops, mending diam dulu deh!" 😅

Menariknya: Kamera bisa menjadi pembaca situasi rumah jika digunakan secara bijak.

Contoh Soal dan Pembahasan

Soal:

Apa itu "model" dalam AI, dan bagaimana cara kerjanya di proyek kita?

A. Komputer super besar yang menyimpan semua gambar
B. File HTML biasa yang membuka kamera
C. "Otak mini" buatan yang belajar dari contoh gambar
D. Kamera yang bisa mengenali wajah

Pembahasan:
Yang dimaksud "model" adalah file yang berisi kemampuan AI untuk mengenali gambar berdasarkan *latihan* sebelumnya. Kita membuat model ini di **Teachable Machine** dengan cara memberikan contoh ekspresi, seperti *tersenyum* atau *serius*. Model ini seperti **"otak kecil" AI**—jawaban **C** adalah benar!

Soal:

Saat kamu membuat model di Teachable Machine, apa langkah penting yang harus kamu lakukan sebelum "Export Model"?

A. Memasang App Inventor
B. Mengambil gambar di setiap kelas ekspresi
C. Membuat akun GitHub
D. Menulis kode HTML

Pembahasan:
Sebelum kamu bisa mengekspor model, kamu harus **mengisi setiap kelas dengan contoh gambar**. Misalnya, kelas "Tersenyum" dan "Serius" masing-masing harus punya gambar yang cukup. Tanpa itu, AI tidak akan tahu perbedaannya. Jawaban **B** adalah yang paling tepat!

Soal:

Kenapa kita perlu mengganti tulisan "PASTE-LINK-MODEL-DI-SINI" di kode HTML?

A. Agar halaman bisa terbuka lebih cepat
B. Supaya HTML bisa mendeteksi warna kamera
C. Untuk menghubungkan AI yang sudah dilatih ke halaman web
D. Karena itu hanya dekorasi

Pembahasan:
Tulisan "PASTE-LINK-MODEL-DI-SINI" adalah **placeholder**. Kita harus menggantinya dengan **URL model yang sudah dilatih** dari Teachable Machine agar skrip AI tahu model mana yang harus dipakai. Tanpa ini, AI tidak bisa memuat “otak” kita! Jawaban **C** adalah benar.

Soal:

Fungsi tombol “Mulai Kamera” di halaman HTML adalah untuk…

A. Menyimpan data ekspresi ke Google
B. Menyalakan kamera dan mulai mendeteksi ekspresi
C. Membuka halaman GitHub
D. Menginstal aplikasi ke HP

Pembahasan:
Saat tombol "Mulai Kamera" ditekan, fungsi `init()` dijalankan. Fungsi ini akan **menyalakan kamera HP**, lalu memuat model, dan mulai membaca ekspresi kita. Artinya, tombol ini adalah **pemicu awal pengenalan wajah**. Jawaban **B** benar!

Soal:

Mengapa kita menggunakan GitHub Pages dalam proyek ini?

A. Agar bisa membuat akun Zoom
B. Untuk menyimpan model di Android
C. Agar halaman web kita bisa diakses dari HP dan dihubungkan ke App Inventor
D. Supaya bisa belajar coding HTML

Pembahasan:
GitHub Pages memungkinkan kita **menyimpan dan menayangkan halaman web secara gratis**. Web ini akan dibuka lewat App Inventor, jadi halaman AI kita bisa dijalankan langsung dari aplikasi buatan kita. Jawaban **C** adalah yang paling sesuai.

Soal:

Kamu ingin membuat AI yang mengenali benda, bukan ekspresi wajah. Apa yang harus kamu lakukan berbeda?

A. Mengubah kode HTML sepenuhnya
B. Memilih jenis model lain di Teachable Machine
C. Menyimpan file di App Store
D. Menghapus script kamera

Pembahasan:
Jika kamu ingin mengenali **benda** (misalnya: sendok vs. sisir), kamu tetap bisa menggunakan **Image Project** di Teachable Machine. Hanya saja, kamu perlu memberikan **contoh gambar yang sesuai dengan benda tersebut**, bukan ekspresi wajah. Jawaban **B** tepat karena kamu mungkin memilih jenis proyek atau data yang berbeda, tapi langkah umumnya tetap serupa.

Fakta Menarik / Fun Facts

1. **Kamu bisa bikin AI sendiri hanya dari HP dan webcam!** Nggak perlu jadi ilmuwan dulu, cukup senyum di depan kamera dan ajari AI-nya.

2. **Kamera HP kamu bukan cuma buat selfie**, tapi juga bisa jadi "mata" buat mengenali ekspresi wajah, warna, atau benda favoritmu!

3. **Model AI itu kayak otak kecil digital!** Ia belajar dari contoh, seperti kamu belajar dari latihan soal.

4. **Kamu bisa bikin website AI sendiri dan diakses dari mana aja** — bahkan teman kamu di kota lain bisa coba model AI buatanmu lewat GitHub Pages.

5. **AI bisa belajar membedakan wajah kamu saat tersenyum, serius, bahkan melotot!** Cuma butuh beberapa foto contoh aja, lho.

6. **Teachable Machine dibuat oleh Google!** Jadi, kamu sedang belajar teknologi dari perusahaan raksasa dunia.

7. **Gabungin AI + Web + App Inventor = Aplikasi Keren buatan kamu sendiri!** Bisa deteksi ekspresi langsung dari aplikasi Android.

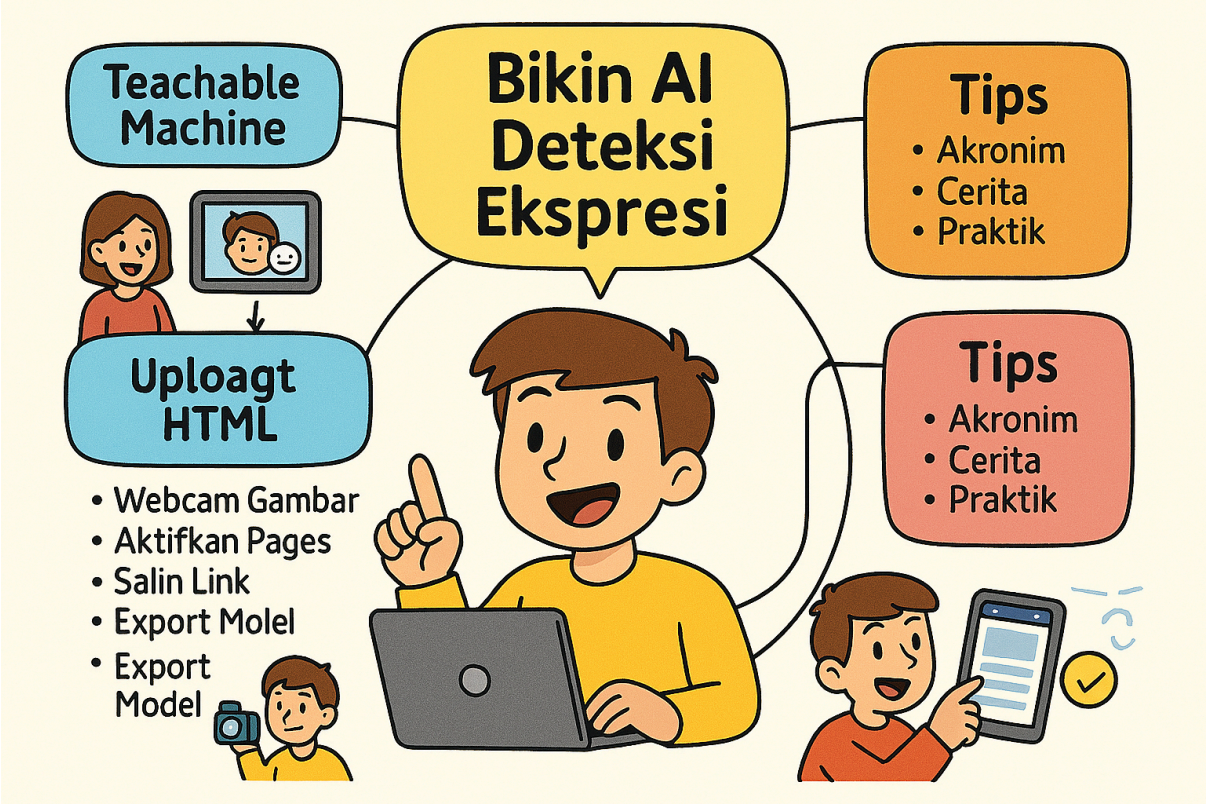
8. **Tanpa coding ribet, kamu sudah membuat program AI.** Hebatnya, kamu juga jadi desainer dan developer sekaligus!

Tips Belajar atau Tips Cepat Menghafal

Tips Menghafal Materi:

1. **Gunakan Mind Mapping** (Peta Konsep Visual)
2. **Ubah Menjadi Cerita / Narasi**
3. **Gunakan Akronim / Singkatan Lucu**
4. **Belajar Sambil Praktik**
5. **Ajak Teman Diskusi atau Ajar Orang Lain**

Mind Mapping: "Bikin AI Sendiri dari Kamera HP!"



Cerita Ringan

Judul: *"Tari Ekspresi Bersama Si Teachable"*

Bayangin kamu punya robot kecil lucu bernama *Tari* yang bisa nebak ekspresi wajah kamu—lagi senyum atau serius. Tapi awalnya dia belum jago, jadi kamu harus sekolahin dia dulu di tempat keren bernama **Teachable Machine**. Di sana, kamu foto wajahmu sendiri: kadang senyum, kadang serius (kayak foto KTP tapi lebih seru!).

Setelah Tari belajar, kamu simpan “otaknya” dalam kotak ajaib yang disebut **Model**, lalu kamu upload ke rak online bernama **GitHub Pages**. Terakhir, kamu bikin aplikasi sendiri pakai **MIT App Inventor**. Begitu kamu pencet tombol "Mulai", kamera nyala dan... *Tari*

langsung komentar! “Wah, kamu lagi senyum 95% nih!” Seru kan? Belajar teknologi bisa sekreatif itu!

AkrOnIM Lucu untuk Mengingat Prosesnya:

T - G - A - B

- **T**: Teachable Machine
- **G**: GitHub Pages
- **A**: App Inventor
- **B**: Build & Jalankan!

Aktivitas Siswa

Aktivitas 1: Jelajahi Dunia AI dari Kamera HP!

Tujuan: Mengenalkan konsep dasar Artificial Intelligence (AI) melalui kamera dan ekspresi wajah.

Langkah-Langkah:

1. Diskusikan bersama guru: Apa itu AI? Apa itu “model pembelajaran mesin”?
2. Tonton video pendek tentang AI yang bisa mengenali wajah.
3. Ajak siswa berdiskusi: “Bagaimana jika kamu bisa melatih komputer mengenali senyumanmu?”

Tantangan: Ajak temanmu membuat daftar benda atau ekspresi wajah apa saja yang bisa dikenali oleh AI.

Aktivitas 2: Latih AI-mu Sendiri di Teachable Machine!

Tujuan: Siswa membuat model AI sederhana yang mengenali ekspresi wajah (misalnya: senyum vs serius).

Langkah-Langkah:

1. Buka Teachable Machine.
2. Pilih *Image Project → Standard Image Model*.
3. Tambahkan dua kelas: “Senyum” dan “Serius”.
4. Gunakan webcam HP untuk mengambil 10–15 gambar ekspresi wajah.
5. Klik **Train Model** untuk “melatih otak AI”.

Tips Seru: Coba gerakkan kepala ke kiri/kanan saat ambil gambar. AI suka variasi!

Aktivitas 3: Uji Coba Model AI Kamu di Web!

Tujuan: Menjalankan model AI lewat halaman web secara langsung.

Langkah-Langkah:

1. Setelah *train model*, klik **Export Model** → pilih *TensorFlow.js*.
2. Klik *Upload (hosted)* lalu salin link yang muncul.
3. Diskusikan bersama guru: Kenapa perlu “hosting” model AI?

Fakta Menarik: Model AI kamu sekarang bisa digunakan siapa saja di internet, lho!

Aktivitas 4: Buat Halaman Web Deteksi Ekspresi!

Tujuan: Membuat halaman web sederhana yang memuat model AI.

Langkah-Langkah:

1. Salin kode HTML yang sudah disiapkan.
2. Ganti bagian `PASTE-LINK-MODEL-DI-SINI` dengan link model kamu.
3. Simpan file sebagai `index.html`.

Tantangan: Tambahkan teks lucu seperti “Saya melihat kamu tersenyum!” di halamanmu.

Aktivitas 5: Upload Projek ke GitHub Pages

Tujuan: Menampilkan halaman web AI kamu agar bisa diakses dari HP.

Langkah-Langkah:

1. Buat akun GitHub dan repository baru, misalnya: `wajahku-ai`.
2. Upload `index.html` ke repository tersebut.
3. Aktifkan GitHub Pages dari menu Settings → Pages.

Tips Jitu: Simpan URL GitHub Pages kamu. Nanti kita butuh untuk ditampilkan di aplikasi Android!

Aktivitas 6: Bangun Aplikasi Android Tanpa Coding di MIT App Inventor!

Tujuan: Menyambungkan halaman web AI ke aplikasi Android.

Langkah-Langkah:

1. Buka MIT App Inventor.
2. Tambahkan komponen: `Button` dan `WebViewer`.
3. Saat tombol ditekan, tampilkan URL GitHub Pages kamu.
4. Bangun (Build) aplikasinya, lalu instal di HP masing-masing.

Tantangan Tambahan: Desain aplikasi dengan ikon dan warna favoritmu!

Aktivitas-aktivitas di atas bukan sekadar membuat aplikasi, tapi juga **melatih logika, kreativitas, dan rasa ingin tahu**. Cocok untuk siswa SMP yang ingin mengenal teknologi masa depan dengan cara yang fun dan menantang!

Rangkuman

Pernah nggak sih kamu kepikiran bikin aplikasi canggih yang bisa mengenali ekspresi wajah kamu sendiri? Di bab ini, kamu akan diajak bikin aplikasi pendeteksi ekspresi wajah hanya dengan **kamera HP** dan alat-alat seru yang tersedia gratis! Pertama-tama, kamu akan bermain-main di *Teachable Machine* dari Google. Di sana kamu bisa melatih model AI sendiri dengan memberikan contoh gambar wajah, misalnya wajah **"tersenyum"** atau **"serius"**, lalu tinggal klik **"Train Model"** dan boom! AI kamu sudah belajar membedakan ekspresi!

Setelah AI kamu jadi, saatnya membuatnya tampil keren di sebuah halaman web! Kita akan menggunakan *GitHub Pages* untuk memajang model kamu. Cukup buat file HTML yang memuat script dari *Teachable Machine*, tempelkan link model kamu, lalu aktifkan halaman web-nya. Hasilnya? Kamu punya web yang bisa langsung mengenali wajah kamu melalui kamera! Bayangkan serunya: AI buatan kamu sendiri bisa tahu kapan kamu tersenyum atau memasang wajah serius.

Nah, supaya makin greget, kita akan **hubungkan halaman web ini ke aplikasi Android buatanmu sendiri** di *MIT App Inventor*. Cukup tambahkan komponen *WebViewer* dan satu tombol, lalu arahkan ke halaman web buatanmu tadi. Begitu tombol ditekan, kamera langsung aktif dan model AI bekerja! Ini tantangan seru: **bisakah kamu bikin aplikasi sederhana dengan AI buatan sendiri?** Jangan takut mencoba dan eksplorasi — karena ini bukan cuma belajar teknologi, tapi juga **berkreasi dan bermain** dengan ide-ide kamu!

Latihan Soal

A. Benar atau Salah (5 Soal)

1. Teachable Machine dapat digunakan untuk melatih model pendeteksi ekspresi wajah menggunakan gambar dari webcam.
   Benar

2. Model dari Teachable Machine tidak bisa diekspor ke format lain selain gambar.
   Salah

3. URL GitHub Pages harus dimasukkan ke dalam App Inventor agar aplikasi bisa menampilkan halaman web.

Benar

4. Kita tidak bisa menggunakan webcam untuk mengumpulkan data di Teachable Machine.
   Salah

5. Model AI yang dibuat akan secara otomatis bisa mendeteksi semua jenis objek tanpa dilatih.
   Salah

B. Pilihan Ganda (10 Soal)

6. Langkah pertama untuk membuat model AI di Teachable Machine adalah...
   A. Membuat akun GitHub
   B. Membuka App Inventor
   C. Memilih "Image Project"
   D. Menyimpan file HTML
   Jawaban: C

7. Komponen apa yang digunakan di App Inventor untuk membuka halaman web model AI?
   A. Camera
   B. WebViewer
   C. ImageSprite
   D. Clock
   Jawaban: B

8. Fungsi dari tombol “Train Model” pada Teachable Machine adalah...
   A. Menyimpan gambar
   B. Mengatur kamera
   C. Melatih model dengan gambar yang dikumpulkan
   D. Mengekspor file HTML
   Jawaban: C

9. File HTML dari model harus diunggah ke...
   A. Microsoft Word
   B. Dropbox
   C. GitHub Pages
   D. Google Drive
   Jawaban: C

10. Script `<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@0.8/dist/teachablemachine-image.min.js"></script>` berfungsi untuk...
    A. Membuka kamera
    B. Menampilkan label ekspresi
    C. Memuat pustaka Teachable Machine

D. Menyimpan metadata model
Jawaban: C

11. Fungsi `predict()` dalam kode JavaScript digunakan untuk...
    A. Menyalakan kamera
    B. Menebak ekspresi dari gambar kamera
    C. Membuat halaman web
    D. Mengatur ukuran gambar
    Jawaban: B

12. URL model dari Teachable Machine harus dimasukkan ke dalam...
    A. Tag `<head>`
    B. Variabel `URL` di script JavaScript
    C. Tombol di HTML
    D. Metadata GitHub
    Jawaban: B

13. App Inventor digunakan untuk...
    A. Melatih model AI
    B. Membuat aplikasi Android
    C. Mengedit HTML
    D. Menyimpan gambar ekspresi
    Jawaban: B

14. Apa nama format model yang diekspor dari Teachable Machine untuk digunakan di web?
    A. Python
    B. TensorFlow.js
    C. C++
    D. HTML
    Jawaban: B

15. Setelah model dimuat di aplikasi, saat tombol ditekan...
    A. Kamera akan mengambil screenshot
    B. Aplikasi akan crash
    C. Halaman GitHub Pages akan terbuka dan mengenali ekspresi
    D. Gambar akan dikirim ke Google Drive
    Jawaban: C

C. Isian Singkat (5 Soal)

16. Untuk mulai membuat model AI di Teachable Machine, kita harus memilih jenis proyek yaitu ____.
    Jawaban: Image Project

17. Komponen yang digunakan untuk membuka halaman web dalam App Inventor adalah ____.

Jawaban: WebViewer

18. Model AI dari Teachable Machine harus dilatih dengan beberapa contoh gambar untuk setiap ____.
Jawaban: kelas

19. File utama yang harus dibuat dan diunggah ke GitHub untuk menampilkan model adalah ____.
Jawaban: index.html

20. Di dalam HTML, kita perlu mengganti tulisan "PASTE-LINK-MODEL-DI-SINI/" dengan ____.
Jawaban: URL dari model Teachable Machine

D. Soal Explorasi (3 Soal)

Soal:

Jelaskan bagaimana kamu bisa membuat model AI yang bisa membedakan antara wajah tersenyum dan wajah serius menggunakan kamera HP. Sebutkan langkah-langkah utamanya dari awal sampai aplikasinya bisa dijalankan di HP.

Soal:

Bayangkan kamu ingin membuat model yang bisa mengenali dua benda: sendok dan sisir. Jelaskan bagaimana kamu akan mengumpulkan data, melatih model di Teachable Machine, dan memastikan modelmu cukup akurat sebelum digunakan di App Inventor.

Soal:

Kamu telah berhasil mengunggah halaman web ke GitHub Pages, tetapi saat membuka aplikasi di HP, halaman tidak muncul. Sebutkan 3 kemungkinan penyebab masalah tersebut dan bagaimana cara memperbaikinya.

Tugas Proyek

Proyek 1: AI Pendeteksi Senyuman Menggunakan Kamera HP

Deskripsi:
Siswa membuat aplikasi yang bisa mendeteksi apakah seseorang sedang **tersenyum atau tidak** dengan memanfaatkan kamera HP dan teknologi AI dari Teachable Machine.

Langkah-Langkah:

1. Akses Teachable Machine.
2. Buat dua kelas: "Tersenyum" dan "Tidak Tersenyum".
3. Ambil foto wajah langsung dari webcam HP.

4. Latih model dan ekspor dalam format TensorFlow.js.
5. Buat halaman web (index.html) untuk menjalankan model.
6. Unggah ke GitHub Pages dan hubungkan dengan App Inventor.
7. Bangun aplikasi yang menampilkan prediksi ekspresi wajah secara langsung.

Tujuan Pembelajaran:
Mengenal cara kerja model AI sederhana dan keterkaitan antara kamera, data, dan pemrosesan AI.

Proyek 2: Robot AI Mini—Deteksi Objek Sehari-Hari

Deskripsi:
Proyek seru untuk membuat AI yang bisa mengenali benda-benda seperti **sendok, sisir, atau botol** menggunakan kamera HP.

Langkah-Langkah:

1. Buat kelas pada Teachable Machine seperti "Sendok", "Sisir", "Botol".
2. Kumpulkan data gambar dari webcam.
3. Latih dan ekspor model.
4. Gunakan HTML dan GitHub Pages untuk membuat antarmuka.
5. Buat aplikasi App Inventor untuk menampilkan model.
6. Uji apakah AI dapat mengenali benda dengan benar.

Saran Tambahan:
Pilih benda yang mudah dikenali dan tidak terlalu mirip agar hasil prediksi lebih akurat.

Proyek 3: AI Emosi Detektor—Bisa Tebak Perasaan Kamu!

Deskripsi:
Siswa membuat sistem AI yang bisa mengenali **ekspresi emosi**, seperti marah, senang, sedih, atau netral.

Langkah-Langkah:

1. Tambahkan 3–4 kelas emosi di Teachable Machine.
2. Ambil banyak contoh ekspresi untuk setiap emosi.
3. Lanjutkan ke pelatihan model dan ekspor.
4. Buat tampilan web dan aplikasi HP seperti proyek sebelumnya.
5. Tambahkan suara atau teks motivasi saat emosi dikenali.

Tantangan Seru:
Uji aplikasi ini ke teman dan lihat apakah AI bisa menebak dengan benar!

Proyek 4: Eksperimen AI Warna—Deteksi Warna Favoritmu

Deskripsi:
Aplikasi AI yang bisa mengenali warna tertentu seperti **merah, biru, hijau** hanya dari kamera.

Langkah-Langkah:

1. Ambil foto benda berwarna dan klasifikasikan sesuai warnanya.
2. Latih AI untuk membedakan warna.
3. Gunakan antarmuka web dari GitHub.
4. Buat aplikasi HP yang bisa menebak warna yang ditunjukkan ke kamera.

Tips:
Gunakan benda polos (misal kertas warna) agar hasil lebih akurat.

Proyek 5: Game Ekspresi AI—Ayo Tiru Wajah Ini!

Deskripsi:
Buat game sederhana: siswa menirukan ekspresi wajah, dan AI akan memeriksa apakah benar!

Langkah-Langkah:

1. Siapkan kelas ekspresi (senyum, bingung, serius).
2. Latih AI dengan data ekspresi kamu sendiri.
3. Tambahkan fungsi skor atau feedback di App Inventor.
4. Setiap kali prediksi benar, pemain dapat poin.

Tujuan:
Melatih ekspresi wajah sekaligus bermain dengan teknologi AI!

Proyek 6: Studio AI Mini—Gabungkan Semua dalam 1 Aplikasi

Deskripsi:
Proyek lanjutan untuk siswa yang ingin tantangan lebih: gabungkan **deteksi objek, warna, dan ekspresi** dalam satu aplikasi AI.

Langkah-Langkah:

1. Latih beberapa model berbeda di Teachable Machine.
2. Buat beberapa halaman web untuk tiap model.
3. Buat aplikasi App Inventor dengan tombol berbeda untuk setiap fungsi AI.
4. Latih pengguna untuk menggunakan aplikasi sesuai kebutuhan.

Tantangan:
Membuat aplikasi yang memiliki banyak fitur tapi tetap mudah digunakan.

Catatan Tambahan untuk Guru dan Siswa

- Gunakan bahasa sederhana dalam App Inventor.
- Gunakan desain antarmuka yang tidak membingungkan.
- Pastikan koneksi internet stabil saat mengakses GitHub Pages.
- Sampaikan bahwa *model AI* ibarat “otak digital” yang belajar dari contoh.

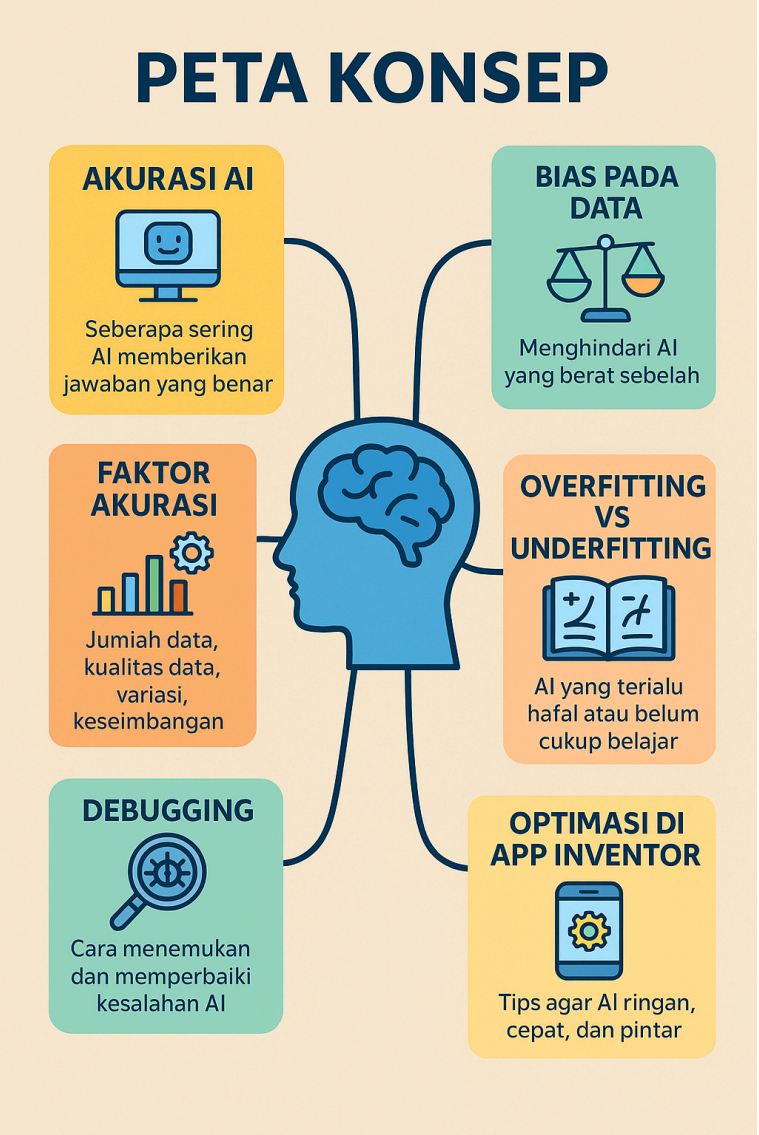
BAB 4: Memahami Akurasi dan Kualitas Data AI

Tujuan Pembelajaran

Setelah menyelesaikan bab ini:

- Siswa mampu memahami apa itu akurasi dalam AI.
- Siswa dapat mengenali penyebab kesalahan AI dalam mengenali data.
- Siswa bisa memperbaiki dan meningkatkan kinerja model AI.
- Siswa mampu menghubungkan AI ke App Inventor dan mengoptimalkannya.

Peta Konsep



Apersepsi

Pada bab ini, kita akan menjelajahi dunia AI (Artificial Intelligence) lebih dalam dengan fokus pada **akurasi dan kualitas data**. Kamu akan belajar bagaimana AI bisa membuat prediksi yang tepat atau malah keliru, serta apa saja penyebabnya. Kita akan memahami bahwa akurasi AI sangat dipengaruhi oleh **jumlah dan kualitas data**, variasi data yang digunakan, serta seimbang atau tidaknya data tersebut. Bahkan, kamu akan menemukan bahwa jika data yang digunakan berat sebelah (bias), maka AI bisa keliru dalam mengambil keputusan! Seolah-olah AI jadi "berpihak" tanpa sadar. Seru, kan? Ini penting banget supaya kamu bisa membangun model AI yang adil dan cerdas.

Tapi tidak cukup hanya membuat AI, kamu juga akan belajar jadi "detektif teknologi" dengan memahami **debugging** — yaitu cara mencari dan memperbaiki kesalahan saat AI tidak berjalan sesuai harapan. Selain itu, kamu juga akan belajar bagaimana menghindari **overfitting** (AI terlalu hafal data latihan) dan **underfitting** (AI belum cukup belajar), serta bagaimana menghubungkan AI ke dalam **App Inventor** agar aplikasi ringan, cepat, dan responsif. Semua ini akan membuatmu tidak hanya sebagai pengguna AI, tapi juga sebagai **pengembang muda** yang tahu cara berpikir kritis, memecahkan masalah, dan menciptakan solusi berbasis teknologi. Siap tantang diri kamu jadi pembuat AI masa depan? Let’s go!

Penjelasan Konsep (teori)

Akurasi dalam dunia kecerdasan buatan (AI) sangat penting! Ibaratnya seperti kemampuan seorang teman menebak dengan tepat apa yang kita pikirkan—semakin sering benar, berarti dia makin akurat!

Apa itu Akurasi Model AI?

Akurasi adalah ukuran seberapa sering model AI memberikan **jawaban yang benar**. Misalnya, jika kamu membuat AI yang bisa mengenali gambar kucing dan 8 dari 10 kali jawabannya benar, berarti akurasinya 80%.

Mengapa Akurasi Penting?

Karena kita ingin AI **membantu**, bukan malah membuat kesalahan. Bayangkan jika AI penerjemah salah terus menerjemahkan—bisa bikin salah paham, kan?

Faktor-Faktor yang Mempengaruhi Akurasi:

1. **Jumlah Data** – Semakin banyak contoh, AI makin pintar.
2. **Kualitas Data** – Gambar yang buram atau suara yang berisik bisa membuat AI bingung.
3. **Variasi Data** – AI harus melihat **banyak jenis contoh**, seperti gambar dari berbagai sudut, warna, atau pencahayaan.

4. **Keseimbangan Data** – Jika AI hanya banyak belajar tentang “anjing” dan sedikit tentang “kucing”, ia akan cenderung mengira semuanya anjing!

Waspadai Bias!

Bias artinya model terlalu condong ke salah satu sisi. Misalnya, jika semua data wajah yang digunakan hanya dari satu ras, maka AI bisa gagal mengenali wajah dari ras lain. Penting untuk punya data yang **beragam dan adil**.

Overfitting vs Underfitting — Penjelasan Simpel:

- **Overfitting**: AI terlalu “hafal” soal latihan dan tidak bisa menjawab soal baru. Ibarat murid yang hanya bisa menjawab soal latihan tapi bingung saat ujian.
- **Underfitting**: AI belum cukup belajar, jadi sering salah. Seperti murid yang kurang belajar, jadi banyak tebakan saat ujian.

Debugging Model AI (Cari & Perbaiki Kesalahan!)

Kalau AI yang kita buat tidak bekerja seperti yang diharapkan, saatnya kita jadi **detektif teknologi**! Proses ini disebut **debugging**, yaitu mencari tahu kesalahan dan memperbaikinya.

Teknik Dasar Debugging di Teachable Machine:

1. **Perhatikan hasil prediksi**: Apakah AI sering salah? Misalnya, mengira “kucing” itu “anjing”?
2. **Periksa data pelatihan**: Apakah gambarnya cukup jelas? Apakah ada label yang salah?
3. **Lihat variasinya**: Apakah semua gambar terlalu mirip? Tambahkan variasi!

Debugging di App Inventor:

- Pastikan **data input** (misalnya dari kamera atau mikrofon) masuk dengan benar.
- Gunakan **Label atau Notifier** untuk menampilkan hasil prediksi.
- Lihat apakah hasilnya masuk akal. Jika tidak, cek lagi apakah AI dan aplikasinya sudah “terhubung” dengan baik.

Mengoptimalkan Kinerja Aplikasi AI

Mau aplikasimu cepat, ringan, dan pintar? Ini dia tipsnya!

Tips Optimasi AI:

- Gunakan **model kecil** agar aplikasi tidak berat di HP.
- Atur **frekuensi pengambilan data** (contohnya, jangan terlalu sering ambil gambar kalau tidak perlu).

- Selalu minta **masukan dari pengguna** (feedback) agar bisa tahu bagian mana yang perlu diperbaiki.

Dengan memahami konsep-konsep ini, kamu tidak hanya bisa membuat AI yang pintar, tapi juga tahu **cara mengembangkan dan memperbaikinya**. Siap jadi ahli AI masa depan? 🌟

Ilustrasi: Menggunakan Software untuk Memecahkan Masalah AI

Bayangkan kamu membuat aplikasi pendeteksi buah dengan **kamera**. Kamu menggunakan **Teachable Machine** untuk melatih AI agar bisa mengenali apel, pisang, dan jeruk. Tapi saat dites, AI sering mengira apel sebagai jeruk. Bingung? Jangan panik!

Dengan bantuan **Preview** di Teachable Machine, kamu melihat bahwa semua gambar apel yang kamu latih diambil dari atas, sedangkan gambar jeruk dari samping. Aha! Mungkin AI mengira bentuk bundar dari atas itu jeruk! Kamu lalu menambahkan **gambar apel dari berbagai sudut** dan mencoba lagi. Ternyata hasilnya membaik!

Kamu lanjutkan ke **App Inventor**, memasang hasil AI dan menampilkan hasil prediksi di **Label**. Sekarang, aplikasi kamu bisa membedakan apel dan jeruk lebih akurat.

Itulah kekuatan dari **debugging + data yang bervariasi**. AI jadi makin pintar!

Contoh Kasus Ilustrasi Sederhana

Kasus 1: Si Kamera Ajaib Salah Tebak

> Bayangkan kamu membuat aplikasi AI yang bisa mengenali apakah makanan di piringmu adalah "nasi goreng" atau "mie goreng". Tapi... ternyata AI kamu sering salah menebak! Bahkan nasi goreng yang jelas-jelas pedas disebutnya mie goreng.
>
> **Ilustrasi:** Kamera mengambil gambar yang gelap atau buram, jadi AI bingung membedakan mie dan nasi.
>
> **Pelajaran:** AI bisa salah jika **kualitas datanya buruk**, seperti gambar gelap atau tidak fokus. Jadi, pastikan datamu jelas!

Kasus 2: Robot Penerjemah yang Bikin Salah Paham

> Kamu membuat chatbot AI yang bisa menerjemahkan bahasa Indonesia ke bahasa Inggris. Tapi waktu temanmu bilang “saya lapar”, AI malah menerjemahkannya jadi “I am happy”.

Ilustrasi: AI belajar dari data yang salah label—misalnya kata “lapar” pernah dilabeli sebagai “happy”.

Pelajaran: AI butuh **data yang benar dan tidak bias**. Kalau banyak label salah, maka hasilnya juga salah.

Kasus 3: Detektif Suara yang Tersesat

Kamu bikin aplikasi pendeteksi suara untuk membedakan antara “tepuk tangan” dan “suara tawa”. Tapi saat temanmu tertawa pelan, AI malah menyangka itu suara angin.

Ilustrasi: Semua suara tawa dalam data latihan terlalu keras, jadi AI tidak kenal suara tawa yang pelan.

Pelajaran: Variasi data itu penting! AI harus belajar dari berbagai jenis contoh, bukan yang itu-itu saja.

Kasus 4: AI yang Selalu Tebak "Anjing"

Di proyek AI-mu, kamu ingin mengenali hewan peliharaan: kucing dan anjing. Tapi anehnya, AI kamu selalu bilang "anjing", bahkan saat kamu tunjukkan foto kucing lucu!

Ilustrasi: Ternyata di data latihan, kamu punya 100 foto anjing tapi hanya 10 foto kucing.

Pelajaran: Ini karena **data tidak seimbang**. AI lebih sering melihat anjing, jadi pikir semua adalah anjing!

Kasus 5: Murid yang Cuma Hafal Latihan

Kamu bangga banget! AI-mu bisa mengenali semua gambar latihan dengan sempurna. Tapi saat kamu beri gambar baru, AI langsung bingung. Kenapa?

Ilustrasi: AI seperti murid yang hanya hafal contoh latihan tapi gagal ujian sungguhan.

Pelajaran: Ini namanya **overfitting**. AI terlalu hafal, jadi tidak fleksibel. Kita harus beri data yang bervariasi agar AI bisa “berpikir”.

Kasus 6: Debugging ala Detektif Digital

AI-mu tidak jalan seperti rencana. Saat digunakan di App Inventor, hasilnya aneh. AI bilang semua benda adalah “sendok” meski kamu tunjukkan garpu.

Ilustrasi: Seperti jadi detektif, kamu harus cek: apakah kamera mengirim gambar yang benar? Apakah AI sudah tersambung dengan baik?

Pelajaran: Debugging adalah proses penting untuk mencari dan memperbaiki kesalahan. Jadi, jangan langsung menyerah—ayo jadi detektif teknologi!

Jika kamu bisa memahami keenam kasus seru ini, kamu sedang melangkah menjadi pembuat AI yang **cerdas, kreatif, dan siap tantangan**!

Contoh Soal dan Pembahasan

Soal 1: Detektif Akurasi AI

Kamu membuat model AI yang bisa mengenali gambar buah. Dari 10 gambar apel, AI berhasil mengenali 7 dengan benar. Berapa persen akurasi AI-mu? Apakah akurasi ini sudah bagus?

Pembahasan:
Akurasi dihitung dengan rumus:

Akurasi = (jumlah prediksi benar / total prediksi) x 100%

Jadi,

Akurasi = (7 / 10) x 100% = 70%

Angka 70% tergolong **cukup baik**, tapi masih bisa ditingkatkan. Kita bisa meningkatkan akurasi dengan menambahkan **data yang lebih banyak dan berkualitas**.

Soal 2: Misteri Gambar Buram

AI milik Adit sering salah mengenali gambar. Ternyata, sebagian besar gambar latihannya buram atau gelap. Menurutmu, apa yang harus Adit lakukan agar akurasi model AI-nya meningkat?

Pembahasan:
Masalah utama ada di **kualitas data**. Solusi yang bisa dilakukan Adit adalah:

- Mengganti gambar buram dengan yang lebih **jelas dan terang**.
- Mengambil gambar dari **berbagai sudut dan kondisi cahaya**.
- Memastikan setiap gambar memiliki **label yang benar**.

Data yang baik = prediksi yang lebih akurat!

Soal 3: Anjing atau Kucing?

Sebuah model AI dibuat untuk mengenali anjing dan kucing. Tapi hasilnya, AI lebih sering menebak semua gambar sebagai “anjing”. Padahal, ada banyak gambar kucing juga. Apa kemungkinan penyebabnya?

Pembahasan:
AI mungkin mengalami **ketidakseimbangan data**. Jika data pelatihan lebih banyak gambar anjing dibandingkan kucing, AI jadi cenderung memilih “anjing” terus-menerus.

Solusinya:

- Tambahkan **jumlah gambar kucing**.
- Pastikan jumlah data **seimbang** untuk setiap kategori.

Soal 4: Overfitting atau Underfitting?

Bayu membuat model AI yang sangat bagus di data latihan, tapi saat digunakan di aplikasi nyata, hasil prediksinya buruk. Apakah model AI-nya overfitting atau underfitting? Jelaskan!

Pembahasan:
Itu adalah **overfitting**.

Artinya, model terlalu “hafal” data latihan, tapi tidak bisa menangani data baru. Ibarat murid yang hanya belajar dari contoh soal, tapi bingung saat ujian.

Cara mengatasi:

- Tambahkan data yang **bervariasi**.
- Kurangi pelatihan yang terlalu lama agar AI tidak terlalu “terpaku” pada data latihan.

Soal 5: Jadi Ahli Debugging!

Nina membuat aplikasi AI di App Inventor untuk mengenali suara. Tapi hasil prediksi AI-nya tidak muncul di layar. Sebagai “detektif AI”, langkah apa saja yang bisa Nina lakukan untuk debugging?

Pembahasan:
Langkah-langkah debugging yang bisa dilakukan Nina:

1. **Cek input suara** — Apakah mikrofon aktif dan data masuk?
2. **Gunakan Label atau Notifier** — Agar hasil prediksi bisa ditampilkan.
3. **Cek koneksi model AI** — Pastikan model sudah terhubung dengan benar di App Inventor.
4. **Uji coba dengan suara yang jelas** — Hindari suara bising saat merekam.

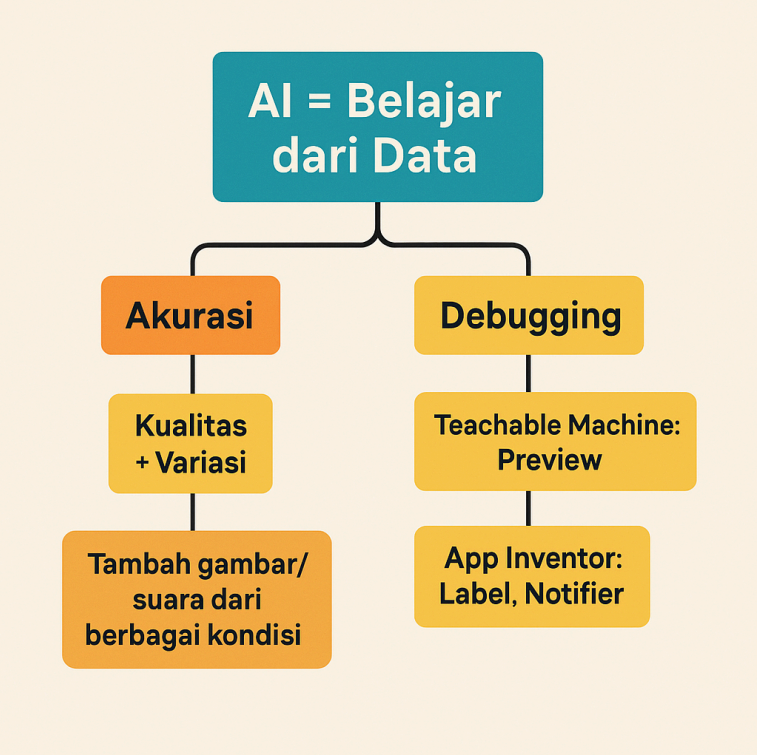
Dengan debugging, Nina bisa tahu bagian mana yang error dan memperbaikinya. Seperti jadi detektif digital!

Fun Facts

- **AI bisa belajar dari suara kamu!** Tapi AI bisa bingung kalau kamu sedang flu 😄
- **Kualitas data lebih penting daripada jumlah.** Seribu gambar buram kalah dengan seratus gambar jelas!
- **AI tidak bisa "berpikir",** tapi bisa mengenali pola seperti ekspresi wajah atau suara.
- **Teachable Machine dibuat oleh Google** dan bisa dipakai gratis untuk belajar AI.

Tips Belajar dan Cepat Menghafal

Mind Mapping



Contoh Cerita Lucu

Bayangkan AI itu seperti anak kecil belajar mengenal buah. Kalau cuma dikasih apel merah dari atas, dia akan kira semua buah bulat dari atas itu apel. Makanya perlu lihat apel hijau, dari samping, bahkan yang agak busuk. Baru deh AI ngerti semua jenis apel.

Aktivitas Siswa

Aktivitas 1: Eksperimen "AI Tebak Gambar": Uji Akurasi Model!

Tujuan: Siswa mengamati seberapa akurat model AI mengenali gambar.

Langkah Aktivitas:

- Gunakan **Teachable Machine** untuk membuat model AI sederhana (misalnya mengenali kucing vs anjing).
- Uji dengan gambar dari berbagai sudut, warna, dan pencahayaan.
- Catat berapa kali AI benar dan salah → hitung akurasi secara manual.
- Diskusikan: Apa yang membuat AI sering salah? Gambar buram? Pencahayaan?

Tantangan: Tambahkan satu kelas baru (misalnya "kelinci"). Apakah akurasinya tetap stabil?

Aktivitas 2: Tantangan "Data Seimbang atau Bias?"

Tujuan: Siswa menyadari pentingnya keberagaman dan keseimbangan data.

Langkah Aktivitas:

- Siapkan dua set data: satu seimbang (misalnya 10 gambar kucing & 10 anjing), satu tidak seimbang (misalnya 15 anjing & 5 kucing).
- Latih dua model AI menggunakan masing-masing set data.
- Bandingkan hasilnya saat diuji.
- Ajak siswa berdiskusi: Apa yang terjadi ketika data tidak seimbang?

Refleksi: “Kalau AI-nya hanya lihat anjing terus, kira-kira bisa adil tidak ya?”

Aktivitas 3: Simulasi "Detektif Debugging AI"

Tujuan: Melatih siswa memeriksa kesalahan model AI dan memperbaikinya.

Langkah Aktivitas:

- Buat model AI dengan sengaja memasukkan beberapa data salah label (misalnya gambar kucing diberi label anjing).
- Jalankan model dan minta siswa mencari kejanggalan hasil.
- Dorong mereka untuk mengusulkan perbaikan (misalnya memperbaiki label, menambah variasi gambar).
- Diskusi: Apa yang paling berpengaruh pada perbaikan akurasi?

Tantangan: Siapa paling cepat menemukan dan memperbaiki kesalahan?

Aktivitas 4: Lokakarya Mini: “Optimasi Aplikasi AI-mu!”

Tujuan: Siswa memahami pentingnya efisiensi dalam aplikasi AI.

Langkah Aktivitas:

- Gunakan App Inventor untuk membuat aplikasi prediksi sederhana menggunakan model AI dari Teachable Machine.
- Tes aplikasi di HP masing-masing: apakah lambat? Apakah prediksi muncul dengan cepat?
- Minta siswa mencoba mengubah ukuran model atau menyesuaikan frekuensi prediksi.
- Catat perubahan performa dan bahas hasilnya bersama.

Challenge: Siapa yang bisa membuat aplikasi tercepat dan tetap akurat?

Aktivitas 5: Simulasi "Overfit vs Underfit": Latih AI-mu Secukupnya!

Tujuan: Memahami perbedaan overfitting dan underfitting secara praktis.

Langkah Aktivitas:

- Latih satu model dengan sangat sedikit data → amati hasilnya (underfitting).
- Latih satu model dengan data yang terlalu mirip dan banyak diulang → amati hasilnya (overfitting).
- Uji kedua model dengan data baru. Diskusikan: Mana yang lebih siap menghadapi dunia nyata?

Insight: AI yang pintar bukan yang hafal, tapi yang bisa beradaptasi!

Aktivitas 6: Forum Kelas: "AI Gagal? Yuk, Evaluasi Bareng!"

Tujuan: Siswa membangun pola pikir evaluatif dan kolaboratif.

Langkah Aktivitas:

- Setiap kelompok menampilkan model AI mereka dan menunjukkan satu masalah yang mereka temukan.
- Kelompok lain memberikan masukan solusi.
- Buat “papan solusi kelas” berisi ide-ide untuk memperbaiki dan mengoptimalkan model AI.

Nilai plus: Belajar dari kegagalan itu kunci jadi ahli AI masa depan!

Rangkuman

ernah nggak kamu bertanya-tanya, kenapa *AI* kadang bisa salah tebak? Nah, di bab ini kamu akan diajak jadi “detektif AI” dan mempelajari bagaimana cara mengecek dan meningkatkan **akurasi** model AI kamu. *Akurasi* itu penting banget—ibarat teman yang selalu bisa menebak pikiran dengan tepat! Tapi supaya AI bisa jadi pintar dan bisa diandalkan, kamu perlu tahu apa saja yang mempengaruhi kualitas prediksinya, seperti **jumlah data**, **kualitas gambar atau suara**, dan **keragaman contoh**. Kalau semua datanya cuma tentang anjing, jangan heran kalau AI-nya mikir semua hewan adalah anjing!

Tapi tunggu dulu, ada hal yang lebih seru lagi—kadang AI bisa mengalami *bias*! Artinya, AI terlalu fokus pada satu jenis data saja, dan itu bisa bikin hasilnya nggak adil. Makanya penting banget punya data yang **beragam dan seimbang**. Kamu juga akan kenalan dengan istilah keren seperti *overfitting* (AI terlalu hafal latihan, bingung saat ujian) dan *underfitting* (AI belum cukup belajar, jadi suka ngawur). Nah, kalau hasil AI-mu belum memuaskan, kamu bisa lakukan yang namanya **debugging**—semacam investigasi untuk mencari tahu di mana letak kesalahannya. Seru kan?

Setelah paham kesalahan, waktunya kamu jadi *developer AI sejati*! Di sinilah kamu belajar mengoptimalkan aplikasi AI buatanmu di *App Inventor*. Supaya aplikasinya nggak berat dan tetap cerdas, kamu bisa pakai **model kecil**, atur **frekuensi pengambilan data**, dan jangan lupa minta **feedback pengguna**. Dengan bekal ini, kamu nggak cuma jadi pengguna AI, tapi **pencipta AI** yang tahu cara membangun, memperbaiki, dan menyempurnakan teknologi masa depan. Siap jadi ahli AI muda?

Latihan Soal

A. Benar atau Salah (5 Soal)

1. AI akan selalu memberikan hasil yang benar jika sudah dilatih dengan banyak data.
   Jawaban: Salah

2. Salah satu cara memperbaiki model AI yang sering salah adalah dengan menambah variasi data pelatihan.
   Jawaban: Benar

3. Overfitting berarti model AI sudah memahami data dengan sangat baik dan pasti akurat.
   Jawaban: Salah

4. Kualitas data pelatihan lebih penting daripada jumlah data.
   Jawaban: Benar

5. Debugging hanya dilakukan di App Inventor, tidak perlu dilakukan di Teachable Machine.
   Jawaban: Salah

B. Pilihan Ganda (10 Soal)

1. Apa yang menyebabkan model AI menjadi tidak akurat?
   A. Baterai habis
   B. Data pelatihan yang tidak bervariasi
   C. Internet lambat
   D. Jumlah tombol di aplikasi terlalu banyak
   Jawaban: B

2. Apa arti dari overfitting dalam model AI?
   A. Model tidak belajar apa-apa
   B. Model hanya cocok pada data pelatihan dan sulit mengenali data baru
   C. Model bisa memprediksi segalanya
   D. Model tidak bisa digunakan
   Jawaban: B

3. Ketika hasil AI sering salah, langkah pertama yang bisa kita lakukan adalah:
   A. Menghapus aplikasi
   B. Menambah tombol
   C. Memeriksa data pelatihan
   D. Mematikan komputer
   Jawaban: C

4. Apa yang dimaksud dengan bias dalam data AI?
   A. Data hilang
   B. Data berwarna-warni
   C. Data tidak seimbang atau tidak mewakili semua kondisi
   D. Data berasal dari luar negeri
   Jawaban: C

5. Apa fungsi fitur “Preview” di Teachable Machine?
   A. Mengubah warna background
   B. Melihat hasil prediksi secara langsung
   C. Membuat suara
   D. Mengedit foto
   Jawaban: B

6. Jika AI tidak bisa mengenali suara dengan baik, kemungkinan karena:
   A. Suara terlalu keras
   B. Mikrofon rusak
   C. Data suara pelatihan terlalu mirip
   D. Terlalu banyak tombol
   Jawaban: C

7. Kenapa kita perlu variasi data dalam pelatihan AI?
   A. Supaya data lebih ringan
   B. Supaya aplikasi terlihat keren
   C. Supaya model bisa mengenali berbagai kondisi
   D. Supaya model lebih lambat
   Jawaban: C

8. Saat integrasi model AI ke App Inventor, bagaimana cara menampilkan hasil prediksi?
   A. Menggunakan tombol Save
   B. Menggunakan Label atau Notifier
   C. Menggunakan Bluetooth
   D. Menggunakan tombol power

Jawaban: B

9. Model yang tidak cukup belajar dari data disebut:
   A. Overload
   B. Overtraining
   C. Overfitting
   D. Underfitting
   Jawaban: D

10. Apa manfaat user feedback pada aplikasi AI?
    A. Membuat aplikasi jadi mahal
    B. Mengurangi ukuran aplikasi
    C. Membantu memperbaiki dan meningkatkan akurasi model
    D. Mengubah warna tema
    Jawaban: C

C. Isian Singkat (5 Soal)

1. Istilah untuk model AI yang hanya hafal data pelatihan disebut _____.
   Jawaban: Overfitting

2. Salah satu penyebab kesalahan prediksi AI adalah kualitas _____ yang buruk.
   Jawaban: data

3. Fitur yang digunakan untuk melihat hasil prediksi model di Teachable Machine adalah _____.
   Jawaban: Preview

4. Salah satu teknik debugging di App Inventor adalah dengan menampilkan hasil AI di _____ atau Notifier.
   Jawaban: Label

5. Jika data pelatihan hanya berisi gambar kucing putih, model AI bisa mengalami _____.
   Jawaban: bias

D. Soal Explorasi

Soal:

Jelaskan langkah-langkah yang kamu lakukan jika model AI buatanmu selalu salah dalam membedakan antara suara "ya" dan "tidak". Ceritakan bagaimana kamu mengecek ulang data, dan apa yang akan kamu tambahkan atau ubah untuk memperbaikinya.

Soal:

Bayangkan kamu membuat aplikasi pendeteksi buah menggunakan kamera. Setelah diintegrasikan dengan App Inventor, model AI-nya salah mengenali apel sebagai

jeruk. Jelaskan bagaimana kamu menganalisis dan memperbaiki masalah ini. Apakah kamu perlu mengganti data, menambah gambar, atau mengubah posisi kamera?

Soal:

Kamu membuat aplikasi pendeteksi ekspresi wajah (senyum, sedih, marah), tetapi akurasi modelnya rendah. Tuliskan rencana lengkapmu untuk meningkatkan akurasinya, mulai dari mengecek ulang data pelatihan, menambahkan lebih banyak ekspresi wajah, hingga mengecek bagaimana input kamera ditangkap oleh App Inventor.

Tugas Proyek

Proyek 1: Detektif Data — Cari Kesalahan di AI!

Tujuan: Siswa belajar mengenali kesalahan pada model AI (debugging) dan memperbaikinya.

Langkah-langkah:

1. Gunakan Teachable Machine untuk membuat model sederhana (misalnya: mengenali ekspresi wajah atau benda).
2. Uji model dengan beberapa data baru dan catat prediksi yang salah.
3. Evaluasi: Apakah ada gambar yang buram? Apakah labelnya tepat?
4. Perbaiki model dengan menambahkan lebih banyak data yang bervariasi dan berkualitas.
5. Ulangi proses pelatihan dan bandingkan hasilnya.

Pesan penting: Siswa diajak menjadi “detektif AI” yang teliti dan kritis, sekaligus menyenangkan.

Proyek 2: Adu Akurasi! Model Siapa yang Lebih Pintar?

Tujuan: Mengajak siswa berkompetisi sehat membangun model AI dengan akurasi terbaik.

Langkah-langkah:

1. Buat kelompok kecil dan tantang mereka membuat model AI di Teachable Machine (contoh: mengenali jenis buah).
2. Semua kelompok menguji model mereka dengan data yang belum pernah digunakan sebelumnya.
3. Bandingkan akurasi tiap model.
4. Diskusikan: Apa yang membuat suatu model lebih akurat dari yang lain?
5. Sempurnakan model berdasarkan hasil diskusi.

Pesan penting: Siswa belajar dari kesalahan dan saling memberi masukan—belajar kolaboratif!

Proyek 3: Cek Bias AI — Adil atau Tidak?

Tujuan: Memahami pentingnya data yang beragam dan adil agar AI tidak bias.

Langkah-langkah:

1. Buat model AI sederhana yang mengenali wajah (atau gambar lain).
2. Latih model hanya dengan satu jenis data (misalnya: wajah laki-laki).
3. Uji model dengan data dari kelompok lain (misalnya: wajah perempuan).
4. Amati apakah model bisa mengenali dengan baik.
5. Ulangi pelatihan dengan data yang lebih beragam dan bandingkan hasilnya.

Pesan penting: AI harus adil untuk semua. Ini latihan membangun empati lewat teknologi!

Proyek 4: Overfitting vs Underfitting — Drama Ujian AI

Tujuan: Memahami konsep overfitting dan underfitting dengan eksperimen langsung.

Langkah-langkah:

1. Buat dua model AI:
    - Model A: Dilatih hanya dengan sedikit data.
    - Model B: Dilatih dengan terlalu banyak data yang sangat mirip.
2. Uji kedua model dengan data baru.
3. Diskusikan hasilnya:
    - Model A banyak salah = underfitting.
    - Model B bagus di data latihan tapi salah di data baru = overfitting.
4. Latih ulang model dengan data yang seimbang dan bervariasi.

Pesan penting: AI juga butuh latihan yang pas—tidak kurang, tidak berlebihan!

Proyek 5: Optimasi Aplikasi AI di App Inventor

Tujuan: Mengembangkan aplikasi ringan dan akurat dengan AI terintegrasi.

Langkah-langkah:

1. Integrasikan model Teachable Machine ke App Inventor (misalnya: deteksi gerakan tangan).
2. Tambahkan Label untuk menampilkan hasil prediksi.
3. Uji aplikasi dan amati kecepatan serta akurasinya.
4. Kurangi ukuran model jika terlalu lambat.
5. Tambahkan fitur feedback dari pengguna (misalnya: “Apakah ini sudah benar?”).

Pesan penting: Siswa jadi tahu bahwa AI di dunia nyata harus efisien dan user-friendly!

Proyek 6: Evaluasi Akurasi — Jadi Peneliti AI Mini!

Tujuan: Mengajarkan siswa cara mengukur dan membaca akurasi model AI secara ilmiah.

Langkah-langkah:

1. Siswa membuat tabel prediksi: benar vs salah (confusion matrix sederhana).
2. Hitung akurasi: jumlah prediksi benar dibagi jumlah total prediksi.
3. Bandingkan beberapa versi model untuk melihat peningkatan akurasi.
4. Presentasikan hasilnya ke kelas.

Pesan penting: Belajar AI bukan hanya soal membuat, tapi juga soal mengukur dan menyempurnakan.

Semua proyek di atas bisa dipilih sesuai tingkat pemahaman siswa, dan bisa dikombinasikan dalam satu rangkaian pembelajaran. Proyek-proyek ini mendorong siswa untuk berpikir kritis, kreatif, dan bertanggung jawab dalam membuat teknologi yang lebih cerdas dan adil.

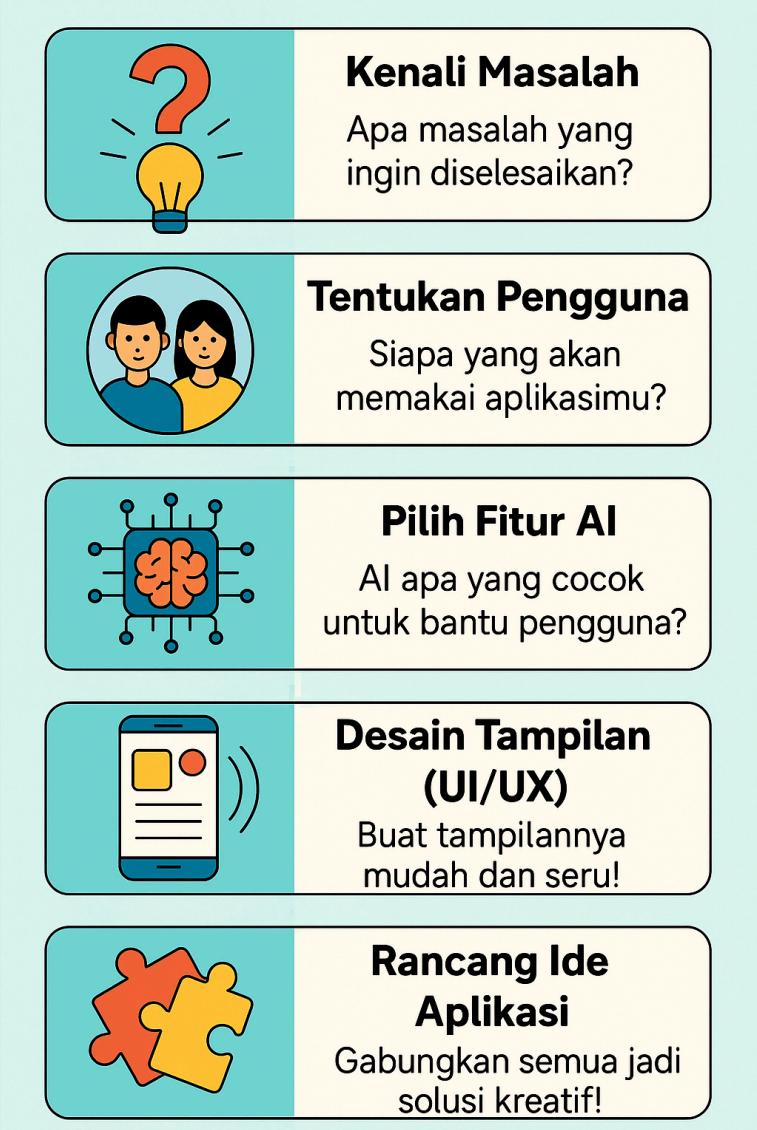
BAB 5: Merancang Aplikasi Ala Kreator Muda: Belajar dari Masalah, AI, hingga UI/UX!

Tujuan Pembelajaran

Setelah menyelesaikan bab ini:

- Siswa dapat **merancang aplikasi sederhana** dengan **fitur AI** untuk **menyelesaikan masalah sehari-hari** di bidang **kesehatan atau pendidikan**.
- Siswa mampu **mengembangkan ide, mendesain tampilan**, dan **menentukan fitur AI** yang tepat dalam aplikasinya.

Peta Konsep



Apersepsi

Pernahkah kamu membayangkan punya aplikasi pintar yang bisa mengingatkan minum air saat belajar, melatih pengucapan bahasa Inggris, atau memberi semangat saat kamu olahraga? Seru, kan? Nah, sekarang saatnya kamu menjadi **kreator muda** yang bukan hanya memakai aplikasi, tapi juga bisa **membuat sendiri aplikasi keren** dengan bantuan **kecerdasan buatan (AI)**. Siap menerima tantangan dan menciptakan solusi dari ide-ide cemerlangmu? Yuk, kita mulai petualangan seru ini!

Penjelasan Konsep (teori)

Merancang aplikasi bukan hanya tentang coding atau teknologi. Proses ini dimulai dari memahami masalah nyata di sekitar kita dan mencoba menemukan solusi kreatif, terutama dengan bantuan kecerdasan buatan (AI). Nah, yuk kita pelajari langkah-langkah pentingnya satu per satu!



Identifikasi Masalah dan Kebutuhan Pengguna

Sebelum membuat aplikasi, kita harus tahu dulu **masalah apa yang ingin diselesaikan**. Setelah itu, kita perlu mencari tahu **siapa yang mengalami masalah tersebut** dan **apa yang mereka butuhkan** agar masalah itu bisa teratasi.

Langkah-langkah penting:

- **Temukan masalah nyata.** Misalnya, masalah di bidang kesehatan (seperti lupa minum obat) atau edukasi (sulit belajar matematika).

- **Tentukan target pengguna.** Siapa yang akan memakai aplikasi ini? Anak-anak? Orang tua? Pelajar?
- **Analisis kebutuhan pengguna.** Apa yang dibutuhkan pengguna agar masalah bisa terselesaikan dengan baik?
- **Lakukan observasi atau wawancara sederhana.** Tanyakan langsung ke calon pengguna agar kita paham dari sudut pandang mereka.

Contoh kasus:

Masalah: Sering lupa minum obat
Target pengguna: Orang dewasa yang sibuk
Kebutuhan: Aplikasi pengingat minum obat yang interaktif dan tidak membosankan

Menentukan Fitur AI dalam Aplikasi

Setelah tahu masalah dan kebutuhan pengguna, saatnya berpikir kreatif: **fitur AI apa yang bisa digunakan untuk membantu?** AI atau kecerdasan buatan bisa membuat aplikasi jadi lebih pintar, responsif, dan membantu pengguna secara otomatis.

Contoh fitur AI yang bisa digunakan:

- **Di bidang kesehatan:**
  - Deteksi pose tubuh untuk koreksi gerakan olahraga
  - Pengenalan suara untuk mencatat konsumsi obat
- **Di bidang edukasi:**
  - Deteksi ekspresi wajah saat belajar (apakah bingung, senang, atau frustrasi)
  - Pengenalan tulisan tangan untuk mengenali jawaban siswa

Brainstorming ide:

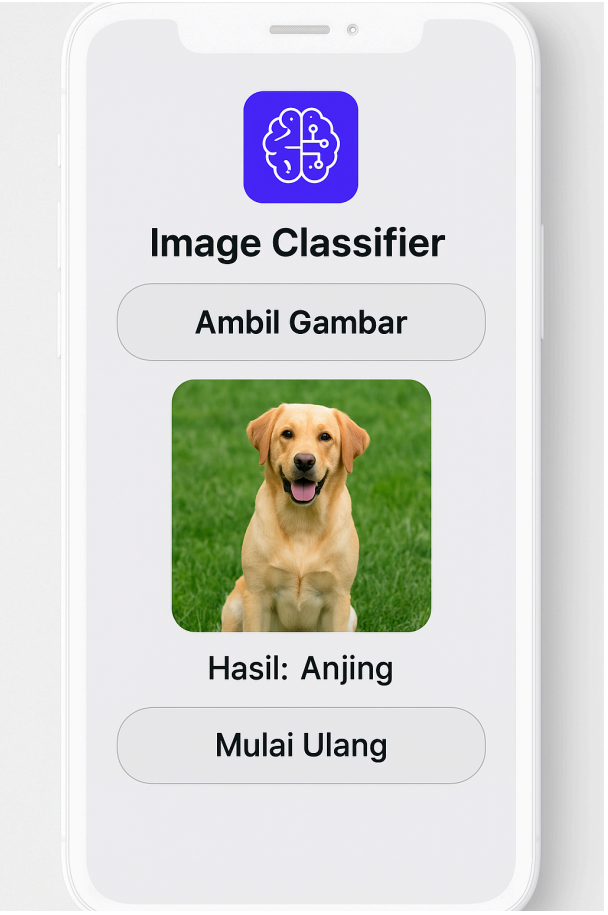
Lanjut dari kasus sebelumnya (pengingat obat), kita bisa menambahkan fitur AI seperti:

- Aplikasi bisa **mengenali suara** saat pengguna bilang “sudah minum obat”
- Kamera bisa **mendeteksi gerakan memegang botol obat** sebagai tanda bahwa obat sudah diminum

Tantangan seru:

Coba pikirkan!
Jika kamu membuat aplikasi belajar matematika, fitur AI apa yang bisa ditambahkan?
Jawaban bisa berupa: pengenalan angka yang ditulis tangan, mendeteksi ekspresi frustasi saat menjawab soal sulit, dll.

Merancang Tampilan dan Pengalaman Pengguna (UI & UX)



Setelah tahu apa saja fiturnya, kita lanjut ke bagian desain! Aplikasi yang bagus bukan hanya pintar, tapi juga **mudah digunakan dan enak dilihat**.

Apa itu UI dan UX?

- **UI (User Interface):** Tampilan aplikasi yang dilihat pengguna, seperti tombol, ikon, warna, dan teks.
- **UX (User Experience):** Pengalaman pengguna saat memakai aplikasi. Apakah mudah? Menyenangkan? Atau malah bikin bingung?

Tips membuat desain yang keren:

- Gunakan **desain yang simpel dan ramah pengguna**
- Buat **sketsa awal (wireframe)** di kertas atau alat digital
- Bayangkan: **Bagaimana pengguna berpindah dari satu tampilan ke tampilan lain?** Apakah mereka bisa mengerti dengan mudah?

Dengan memahami tiga konsep dasar di atas, kamu sudah siap untuk mulai **merancang aplikasi berbasis AI yang seru dan bermanfaat!**
Selamat berkreasi dan jadilah problem solver masa depan! 💡✨

Ilustrasi

Bayangkan seorang siswa bernama Dina. Dina sering lupa minum air saat belajar online di rumah. Ia juga kadang merasa bosan saat belajar bahasa Inggris karena tidak tahu apakah dia sudah mengucapkan kata-katanya dengan benar. Suatu hari, Dina mengikuti pelajaran tentang **AI (Artificial Intelligence)** dan belajar bahwa AI bisa membantu mengingatkan dia untuk minum air secara otomatis, atau bahkan memperbaiki pengucapannya dalam bahasa Inggris.

Dina memutuskan untuk membuat **aplikasi sederhana** yang bisa berbicara dan mengingatkan waktu minum air, serta memberi latihan pengucapan bahasa Inggris yang menyenangkan. Ia pun mulai menggambar desain aplikasinya dan berdiskusi dengan

teman-teman. Dari situ, Dina belajar bahwa **teknologi bisa jadi sahabat** untuk membantu kehidupan sehari-hari.

Contoh Kasus dan Ilustrasi

Kasus 1: Misi Rahasia Si Penjelajah Waktu - Lupa Minum Air!

Ilustrasi:
Bayangkan kamu adalah seorang penjelajah waktu muda bernama Rafa. Saat sedang menjelajah masa depan, kamu sering lupa minum air karena terlalu fokus menjelajah planet asing. Tubuhmu menjadi cepat lelah. Kamu pun menemukan ide membuat **aplikasi AI yang bisa mengenali ekspresi wajah hausmu dan mengingatkan minum air secara otomatis**.

Tantangannya:
Bagaimana cara aplikasi itu tahu kamu haus? Bisa nggak wajah atau gerakan tanganmu ke arah botol dikenali oleh AI? Yuk, pikirkan fitur AI yang bisa digunakan!

Kasus 2: Operasi Matematika - Si Jenius Pemalas

Ilustrasi:
Arga adalah siswa jenius, tapi... ia malas menulis! Saat ulangan matematika, ia lebih suka menggambar angka besar-besar di buku catatan. Bayangkan kamu membuat **aplikasi AI yang bisa membaca tulisan tangannya** lalu **memberikan penjelasan soal matematika secara otomatis**, bahkan saat dia menulis di kertas!

Tantangannya:
AI harus bisa mengenali angka yang ditulis tangan, dan tahu mana yang salah hitung. Gimana ya caranya?

Kasus 3: Misteri Si Pembaca Malam - Anak yang Tak Bisa Membaca Lancar

Ilustrasi:
Di desa kecil, tinggal seorang anak bernama Zia yang suka membaca cerita horor... tapi ia kesulitan membaca kata-kata panjang. Kamu ingin membantunya dengan membuat **aplikasi pembaca teks otomatis**. Saat Zia membaca dan salah mengucapkan kata, aplikasi akan **mendengarkan suaranya dan membetulkan dengan suara AI yang ramah**.

Tantangannya:
AI harus bisa mendengar suara Zia dengan jelas, dan tahu mana pengucapan yang benar. Apa saja fitur yang harus kamu masukkan?

Kasus 4: Pelatih Virtual Sang Juara Kelas

Ilustrasi:
Bayangkan kamu adalah pelatih pribadi untuk tim voli SMP. Tapi kamu tidak bisa hadir langsung. Kamu membuat aplikasi **dengan AI yang bisa mendeteksi gerakan tubuh pemain** dan memberi semangat otomatis seperti, “Ayo, posisi tanganmu kurang tinggi!” atau “Keren! Smash-mu mantap!”

Tantangannya:
Apa fitur AI yang bisa membaca gerakan tubuh? Bagaimana agar aplikasi ini menyenangkan seperti pelatih sungguhan?

Kasus 5: Si Detektif Bahasa Inggris

Ilustrasi:
Alya ingin jago bahasa Inggris, tapi setiap belajar pronunciation (pengucapan), dia bingung apakah sudah benar atau belum. Ia berharap ada aplikasi AI yang bisa **mendengarkan ucapannya, mengecek kebenaran, lalu memperbaiki pengucapannya dengan suara asli** seperti guru native speaker.

Tantangannya:
Bagaimana caranya AI mengenali suara Alya dan tahu kapan dia salah mengucapkan kata? Apa yang bisa ditampilkan di layar agar belajarnya seru?

Kasus 6: Sekolah Anti Stres - Pemindai Wajah Bahagia

Ilustrasi:
Di SMP Ceria, guru ingin tahu apakah siswa merasa senang atau stres saat pelajaran. Kamu diminta membuat **aplikasi AI yang bisa membaca ekspresi wajah siswa melalui kamera**, lalu memberikan saran ke guru: “Siswa tampak bingung, mungkin perlu istirahat sejenak.”

Tantangannya:
AI harus bisa membedakan ekspresi wajah. Bagaimana supaya aplikasi ini tidak menyeramkan dan tetap menjaga privasi siswa?

Setiap kasus di atas mengandung:

- **Masalah nyata** (kesehatan atau edukasi)
- **Target pengguna yang relevan** (siswa, anak-anak, remaja)
- **Fitur AI sebagai solusi**
- **Sentuhan cerita yang menyenangkan dan menantang** untuk membangkitkan rasa ingin tahu siswa SMP.

Contoh Soal dan Pembahasan

Soal 1: Yuk, Jadi Detektif Masalah!

Pertanyaan:
Bayangkan kamu melihat seorang teman yang sering kesulitan belajar matematika karena bingung memahami soal cerita. Masalah apa yang sebenarnya sedang dihadapi? Jika kamu ingin membantu lewat aplikasi AI, fitur apa yang bisa kamu tambahkan?

Pembahasan:
Masalah yang dihadapi temanmu adalah **kesulitan memahami soal cerita matematika**. Aplikasi bisa menggunakan **fitur AI seperti pengenalan teks dan penjelasan otomatis** yang mengubah soal cerita menjadi bentuk matematika yang lebih mudah dipahami. Bahkan, AI bisa membaca ekspresi bingung dan menawarkan bantuan tambahan.

Soal 2: Tebak UI/UX!

Pertanyaan:
Kamu ingin membuat aplikasi pengingat minum air. Menurutmu, apa 3 elemen UI (User Interface) yang penting ditampilkan di layar utama agar pengguna merasa nyaman dan terbantu?

Pembahasan:
Contoh tiga elemen UI yang penting:

1. **Tombol "Saya sudah minum!"** – agar pengguna bisa mencatat kapan mereka minum.
2. **Jam/waktu pengingat berikutnya** – agar tahu kapan harus minum lagi.
3. **Gambar atau ikon lucu (misalnya karakter botol air)** – membuat aplikasi lebih menarik.

Semua ini membantu UX (User Experience) yang menyenangkan dan mudah digunakan.

Soal 3: Wawancara Mini, Ide Maksimal!

Pertanyaan:
Kamu mewawancarai seorang anggota keluarga dan mengetahui bahwa ia sering lupa minum obat. Rancang ide aplikasi AI sederhana untuk membantunya. Sebutkan satu fitur AI yang cocok!

Pembahasan:
Masalah: **Lupa minum obat.**
Solusi: Aplikasi AI dengan fitur **pengenalan suara** yang bisa bertanya, "Apakah sudah minum obat?" dan mendeteksi jawaban pengguna. Bisa juga pakai **kamera untuk mengenali gerakan** mengambil botol obat. Sederhana tapi sangat membantu!

Soal 4: Mana Yang Benar, Nih?

Pertanyaan:
Pilihlah pernyataan yang **benar** mengenai proses merancang aplikasi:
A. Langkah pertama adalah membuat tampilan yang keren.
B. Menentukan nama aplikasi adalah hal utama.
C. Harus tahu dulu masalah yang ingin dipecahkan.
D. Langsung saja coding agar tidak membuang waktu.

Pembahasan:
Jawaban: **C. Harus tahu dulu masalah yang ingin dipecahkan.**
Merancang aplikasi dimulai dari **mengidentifikasi masalah dan kebutuhan pengguna**, bukan dari tampilan atau nama. Tanpa tahu masalah, aplikasi bisa jadi keren tapi tidak berguna.

Soal 5: Teka-Teki Fitur AI

Pertanyaan:
Kamu menemukan masalah: Anak-anak usia 7–9 tahun sulit membaca dengan lancar. Tantangan untukmu: Sebutkan dua fitur AI yang bisa membantu mereka belajar membaca lebih baik.

Pembahasan:
Dua fitur AI yang cocok adalah:

1. **Pembaca teks otomatis (text-to-speech)** – agar anak bisa mendengar cara baca yang benar.
2. **Pengenalan suara** – agar aplikasi bisa menilai apakah anak sudah membaca dengan benar dan memberikan koreksi.

Dengan begitu, anak-anak merasa seperti sedang belajar bersama guru virtual.

Jika kamu suka tantangan ini, bayangkan nanti kamu bisa **membuat aplikasi sendiri** untuk membantu orang-orang di sekitarmu. Karena di era teknologi ini, kamu bukan hanya pengguna — kamu juga bisa jadi **pencipta solusi digital!**

Fun Facts

1. **AI Bisa Jadi Pelatih Olahraga Virtual!**
   Ada aplikasi yang bisa melihat gerakan tubuhmu dan memberi semangat seperti pelatih beneran. “Ayo, tinggal satu kali lagi!”

2. **AI Bisa Menilai Cara Kamu Ngomong Bahasa Inggris!**
   Kalau kamu salah ucap kata "education", AI bisa langsung kasih tahu cara pengucapan yang benar. Gak perlu guru 24 jam!

3. **Ada Aplikasi yang Bisa Mengingatkan Kamu Minum Air, Lho!**
   Bahkan bisa pakai kamera untuk mengenali saat kamu pegang gelas air dan bilang, “Good job!”

4. **Tulis Angka di Kertas, AI Bisa Bacain Jawabannya!**
   AI bisa mengenali tulisan tangan kamu saat ngerjain soal matematika. Asal gak jelek banget, ya!

5. **AI Bisa Tahu Kalau Kamu Lagi Bingung!**
   Beberapa aplikasi belajar bisa mengenali ekspresi wajahmu. Kalau kamu terlihat frustasi, mereka kasih soal yang lebih gampang dulu.

6. **UI Itu Bukan Singkatan dari “Unik dan Imut”!**
   Tapi User Interface—bagian tampilan aplikasi, seperti tombol dan warna. Bikin aplikasi jadi cantik dan mudah dipakai.

7. **Kamu Gak Perlu Jadi Ahli Coding untuk Rancang Aplikasi!**
   Cukup mulai dari ide sederhana dan sketsa di kertas. Kreativitas lebih penting dari hafalan kode!

8. **Pernah Kepikiran AI Bisa "Melihat"?**
   Dengan teknologi *computer vision*, AI bisa mendeteksi apakah kamu sedang olahraga, membaca, atau bengong.

9. **Setiap Aplikasi Hebat Berawal dari Masalah Sepele!**
   Sering lupa minum air? Susah hafal pelajaran? Justru dari masalah sehari-hari seperti itu, ide brilian muncul!

10. **Bikin Aplikasi Itu Seperti Bermain LEGO**
    Kamu bisa rakit dari fitur-fitur sederhana (seperti pengingat, suara, deteksi wajah) jadi aplikasi keren buatanmu sendiri!

Aktivitas Siswa

Aktivitas 1: Detektif Masalah Sehari-hari

Judul Aktivitas: Temukan Masalah, Ciptakan Solusi!

Langkah-Langkah:

1. Ajak siswa mengamati kegiatan sehari-hari di rumah, sekolah, atau lingkungan sekitar.
2. Tantang mereka untuk menemukan **1 masalah nyata** yang dialami orang-orang di sekitar mereka, terutama di bidang **kesehatan atau edukasi**.
3. Diskusikan bersama: *Siapa yang mengalami masalah itu? Apa yang membuatnya sulit?*
4. Minta mereka menuliskan ide awal aplikasi yang bisa menjadi solusi sederhana.

Tujuan: Melatih kepekaan sosial dan berpikir kritis sebagai dasar dalam perancangan aplikasi.

Aktivitas 2: AI Bukan Cuma Robot

Judul Aktivitas: Kenali AI, Temukan Keajaibannya!

Langkah-Langkah:

1. Perkenalkan berbagai contoh fitur **AI yang sudah ada** dalam kehidupan sehari-hari (misal: Google Translate, Siri, kamera yang bisa mengenali wajah).
2. Bahas bersama: *Bagaimana AI bisa membantu orang belajar atau menjaga kesehatan?*
3. Tantang siswa untuk **memilih satu fitur AI** dan mengaitkannya dengan masalah yang mereka temukan pada Aktivitas 1.
4. Siswa membuat catatan: *"AI bisa membantu masalah ini karena…"*

Tujuan: Membangun pemahaman tentang kekuatan teknologi secara nyata dan aplikatif.

Aktivitas 3: Desainer dalam 15 Menit

Judul Aktivitas: Tantangan Wireframe Kilat

Langkah-Langkah:

1. Sediakan kertas kosong atau alat digital sederhana.
2. Berikan tantangan: *Dalam 15 menit, gambarlah tampilan utama aplikasi kalian!* Sertakan tombol, ikon, dan tulisan.
3. Dorong mereka untuk membuat desain **ramah pengguna** dan **nyaman dipakai siswa seusia mereka**.
4. Boleh presentasi singkat (1 menit) untuk menjelaskan desainnya ke teman.

Tujuan: Mengembangkan kreativitas visual dan logika tata letak antarmuka (UI/UX).

Aktivitas 4: Main Peran Si Aplikasi Pintar

Judul Aktivitas: Simulasi Aplikasi AI

Langkah-Langkah:

1. Buat pasangan: satu siswa sebagai *pengguna*, satu lagi sebagai *aplikasi*.
2. Pengguna menyampaikan masalah atau kebutuhan (misal: "Saya mau belajar pengucapan kata 'education'").
3. "Aplikasi" menjawab dan merespons dengan fitur AI yang telah dipelajari.
4. Bergantian peran agar semua mengalami kedua sisi.

Tujuan: Melatih pemahaman logika fitur AI secara menyenangkan dan aplikatif.

Aktivitas 5: Peta Ide Kelas

Judul Aktivitas: Papan Ide Digital

Langkah-Langkah:

1. Siswa menuangkan ide aplikasinya ke dalam bentuk **post-it digital atau kertas warna**.
2. Setiap ide harus menjawab 3 hal:
    - Masalah yang ingin dipecahkan
    - Siapa penggunanya
    - Fitur AI yang dipakai
3. Tempelkan semua ide di papan kelas/online board.
4. Ajak siswa **berkeliling membaca ide teman** dan beri 1 komentar atau saran.

Tujuan: Menumbuhkan rasa kolaboratif, menghargai ide orang lain, serta memperluas wawasan.

Aktivitas 6: Logika Pintar dengan Pseudocode

Judul Aktivitas: Kalau Jam 2, Ingatkan Minum Air!

Langkah-Langkah:

1. Ajak siswa menulis **logika sederhana** aplikasi dengan pseudocode.
2. Berikan contoh:

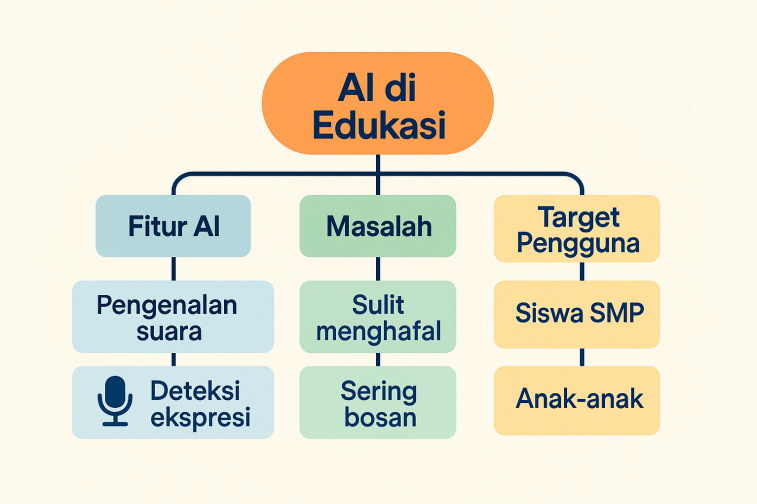
```
IF waktu == jam_pengingat THEN
   tampilkan "Sudah minum air belum?"
END IF
```

3. Tantang mereka membuat pseudocode sendiri sesuai ide aplikasi masing-masing.

Tujuan: Melatih cara berpikir logis ala pemrogram tanpa harus menulis kode sungguhan.

Tips Belajar dan Menghafal

Mind Mapping



Cerita Mudah Diingat

Bayangkan kamu punya **asisten pribadi** yang selalu menolongmu:

- Ia berkata: “Ayo minum air, sudah jam 2 siang!”
- Ia berkata: “Coba ulangi pengucapan ‘education’”
  Dengan AI, kamu seakan-akan punya sahabat digital yang membantu belajar dan menjaga kesehatanmu.

Rangkuman

Merancang aplikasi dengan teknologi AI bukan hanya untuk orang dewasa atau programmer profesional. Siswa SMP pun bisa mulai dari **mengidentifikasi masalah sederhana**, seperti lupa minum air atau sulit belajar matematika. Langkah pentingnya adalah **mengamati**, **berempati**, dan **berimajinasi** tentang solusi cerdas.

Dengan menambahkan fitur AI seperti **pengenalan suara**, **deteksi ekspresi**, atau **deteksi gerakan**, aplikasi yang kita buat bisa menjadi **lebih hidup dan berguna**. Jangan lupa untuk memperhatikan desain tampilan (UI) dan kenyamanan pengguna (UX) agar aplikasi tidak hanya pintar, tapi juga menyenangkan digunakan!

Latihan Soal

A. Benar atau Salah (5 Soal)

Tulislah "Benar" atau "Salah" pada pernyataan berikut:

1. AI hanya bisa digunakan untuk membuat game, tidak untuk membantu di bidang kesehatan.
   Salah

2. Salah satu proses awal merancang aplikasi adalah memahami kebutuhan pengguna.
   Benar

3. Deteksi wajah termasuk salah satu contoh penggunaan AI di bidang edukasi.
   Benar

4. UI adalah singkatan dari User Internet.
   Salah

5. Aplikasi yang baik harus sulit digunakan agar tidak disalahgunakan.
   Salah

B. Pilihan Ganda (10 Soal)

Pilihlah jawaban yang paling tepat!

6. Apa langkah pertama dalam merancang aplikasi AI?
   A. Membuat logo aplikasi
   B. Menentukan warna tampilan
   C. Mengidentifikasi masalah
   D. Menyusun laporan
   Jawaban: C

7. Siapa target pengguna dari aplikasi yang mengingatkan minum obat?
   A. Anak-anak
   B. Orang dewasa yang sibuk
   C. Mahasiswa teknologi
   D. Dokter
   Jawaban: B

8. Contoh fitur AI di bidang edukasi adalah...
   A. Musik latar aplikasi
   B. Deteksi ekspresi wajah saat belajar
   C. Warna tombol aplikasi
   D. Notifikasi dari media sosial
   Jawaban: B

9. UX dalam aplikasi merujuk pada...
   A. Jenis tulisan
   B. Pengalaman pengguna saat menggunakan aplikasi
   C. Ukuran memori aplikasi
   D. Bahasa pemrograman
   Jawaban: B

10. Mengapa penting memahami kebutuhan pengguna?
    A. Agar aplikasi lebih mahal
    B. Agar pengguna tertipu
    C. Agar aplikasi tepat sasaran

D. Agar kode menjadi lebih singkat
Jawaban: C

11. Contoh penggunaan AI di bidang kesehatan adalah...
    A. Fitur curhat
    B. Deteksi pose saat olahraga
    C. Edit video
    D. Alarm jam
    Jawaban: B

12. Fungsi dari wireframe adalah...
    A. Membuat program
    B. Mendesain logo
    C. Membuat sketsa tampilan aplikasi
    D. Memasang server
    Jawaban: C

13. Yang termasuk bagian dari UI adalah...
    A. Emosi pengguna
    B. Kenyamanan pengguna
    C. Tombol dan teks di layar
    D. Kecepatan internet
    Jawaban: C

14. Kenapa AI bisa membantu dalam edukasi?
    A. Bisa menggantikan guru
    B. Bisa membuat materi jadi interaktif dan adaptif
    C. Bisa membuat aplikasi lebih mahal
    D. Bisa membaca pikiran siswa
    Jawaban: B

15. Aplikasi belajar bahasa Inggris dengan AI bisa membantu pengguna dengan cara...
    A. Menghapus semua kosa kata sulit
    B. Memutar lagu bahasa Inggris
    C. Memberikan latihan dengan pengenalan suara
    D. Menyembunyikan jawaban
    Jawaban: C

C. Isian Singkat (5 Soal)

Jawablah pertanyaan berikut dengan singkat!

16. Singkatan dari UI adalah ___.
    Jawaban: User Interface

17. Fitur AI yang mengenali suara disebut ___.
    Jawaban: Pengenalan suara

18. Sebelum merancang aplikasi, kita harus mengetahui ___ yang ingin dipecahkan.
    Jawaban: masalah

19. Sketsa awal desain aplikasi disebut ___.
    Jawaban: wireframe

20. AI bisa digunakan untuk mendeteksi ___ wajah saat seseorang belajar.
    Jawaban: ekspresi

D. Soal Eksplorasi

Jawablah pertanyaan berikut ini dalam bentuk uraian pendek (3–5 kalimat per soal):

Soal:

Bayangkan kamu membuat aplikasi untuk membantu anak-anak belajar membaca. Masalah apa yang ingin kamu selesaikan, dan fitur AI apa yang akan kamu tambahkan? Jelaskan alasanmu.

Soal:

Buatlah sketsa kasar (dalam pikiranmu atau di kertas) tampilan utama aplikasi yang mengingatkan orang untuk minum air. Sebutkan minimal 3 elemen UI yang kamu masukkan dan alasannya.

Soal:

Amati orang di sekitarmu yang mengalami kesulitan belajar. Jelaskan satu masalah yang mereka hadapi dan ide aplikasi AI sederhana yang bisa membantu mereka. Fitur AI apa yang menurutmu paling cocok?

Tugas Proyek

Proyek 1: "Sahabat Belajar AI" – Merancang Aplikasi Edukasi dengan Fitur Cerdas

Tujuan: Membuat desain awal aplikasi edukasi yang membantu siswa memahami pelajaran, seperti Matematika atau Bahasa Inggris.

Langkah-langkah:

1. Pilih satu pelajaran yang dirasa sulit oleh teman-teman di kelas.
2. Identifikasi masalah utamanya (misalnya: bingung saat mengerjakan soal).
3. Tentukan fitur AI yang bisa membantu (misal: deteksi tulisan tangan atau ekspresi bingung).
4. Gambar wireframe (tampilan awal) aplikasi dengan minimal 3 elemen UI (tombol, kolom input, gambar).
5. Presentasikan desainmu dan minta umpan balik dari teman.

Catatan Fun: Tambahkan fitur suara yang bisa menyemangati pengguna saat mereka berhasil!

Proyek 2: "Dokter Pintar di Saku" – Aplikasi AI untuk Kesehatan Harian

Tujuan: Mendesain aplikasi yang bisa membantu pengguna menjaga kebiasaan sehat, seperti minum air atau berolahraga.

Langkah-langkah:

1. Pilih satu kebiasaan sehat yang sering dilupakan (misalnya: minum air setiap 2 jam).
2. Bayangkan siapa yang paling butuh aplikasi ini (misalnya: siswa yang sibuk sekolah online).
3. Tentukan fitur AI-nya: pengingat suara, deteksi wajah saat memegang gelas, atau deteksi pose olahraga.
4. Buat desain wireframe yang ramah pengguna.
5. Simulasikan penggunaannya bersama teman sebagai “pengguna” dan “aplikasi”.

Tantangan Seru: Gunakan minimal **2 fitur AI** yang berbeda!

Proyek 3: "AI dari Sekitar" – Menemukan Masalah Nyata di Sekolah

Tujuan: Menciptakan ide aplikasi berdasarkan masalah sehari-hari yang ditemukan di lingkungan sekolah.

Langkah-langkah:

1. Lakukan observasi cepat atau wawancara singkat dengan teman, guru, atau petugas sekolah.
2. Catat satu masalah yang sering terjadi (contoh: sulit memahami instruksi guru).
3. Brainstorm solusi digital menggunakan AI (misalnya: AI penerjemah suara guru ke teks).
4. Gambar wireframe aplikasi.
5. Buat deskripsi singkat: untuk siapa, apa manfaatnya, dan fitur AI-nya.

Tantangan Kreatif: Diskusikan ide kalian di kelompok dan buat papan ide bersama!

Proyek 4: "Tebak Fitur AI!" – Permainan Serius tapi Seru

Tujuan: Melatih pemahaman tentang fitur-fitur AI melalui permainan kelompok.

Langkah-langkah:

1. Buat daftar berisi 3 masalah sehari-hari (misal: lupa jadwal pelajaran, malas olahraga, bingung saat belajar).
2. Tukar daftar masalah dengan teman sekelompok.

3. Temanmu harus menebak fitur AI apa yang paling cocok untuk menyelesaikan masalah itu.
4. Diskusikan jawabannya: mana yang paling realistis dan inovatif?
5. Pilih satu ide terbaik, lalu buat wireframe sederhananya bersama-sama.

Pesan Menantang: Jangan takut berpikir out of the box, bahkan ide lucu bisa jadi cemerlang!

Proyek 5: "Suara Teman, Ide Hebat" – Wawancara Mini dan Desain Aplikasi

Tujuan: Melatih empati dan kreativitas dengan mendesain aplikasi berdasarkan kebutuhan orang lain.

Langkah-langkah:

1. Wawancarailah satu orang di rumah atau sekolah (bisa teman, kakak, atau orang tua).
2. Tanyakan: apa masalah harian yang mereka alami dan ingin ada teknologi yang membantu?
3. Catat jawabannya, lalu pikirkan fitur AI yang bisa jadi solusinya.
4. Rancang sketsa aplikasi yang dapat digunakan oleh orang tersebut.
5. Ceritakan ide aplikasimu kepada mereka—apakah sesuai harapan?

Nilai Plus: Ini bisa melatih kemampuan komunikasi, berpikir kritis, dan rasa peduli!

Proyek 6: "Aplikasi Impian Dina" – Cerita Jadi Desain

Tujuan: Mengubah cerita inspiratif menjadi desain nyata.

Langkah-langkah:

1. Bacalah kembali kisah tentang Dina yang belajar dari AI.
2. Gunakan imajinasi untuk mengembangkan ide Dina menjadi aplikasi nyata.
3. Tambahkan fitur AI seperti pengingat minum, pelatihan pengucapan bahasa Inggris, dan deteksi bosan.
4. Buat sketsa UI sederhana dan tulis deskripsi singkatnya.
5. Bandingkan dengan hasil teman. Siapa yang punya desain paling menarik?

Pesan Seru: Dari cerita sederhana bisa lahir teknologi luar biasa!

BAB 6: Gunakan ChatGPT untuk Cari Ide Aplikasi

Tujuan Pembelajaran

- Siswa dapat menggunakan ChatGPT untuk mencari ide kreatif.
- Siswa mampu mengembangkan ide menjadi konsep aplikasi sederhana.
- Siswa memahami cara membuat perintah (prompt) yang tepat agar ChatGPT memberikan jawaban terbaik.

Peta Konsep



Apersepsi

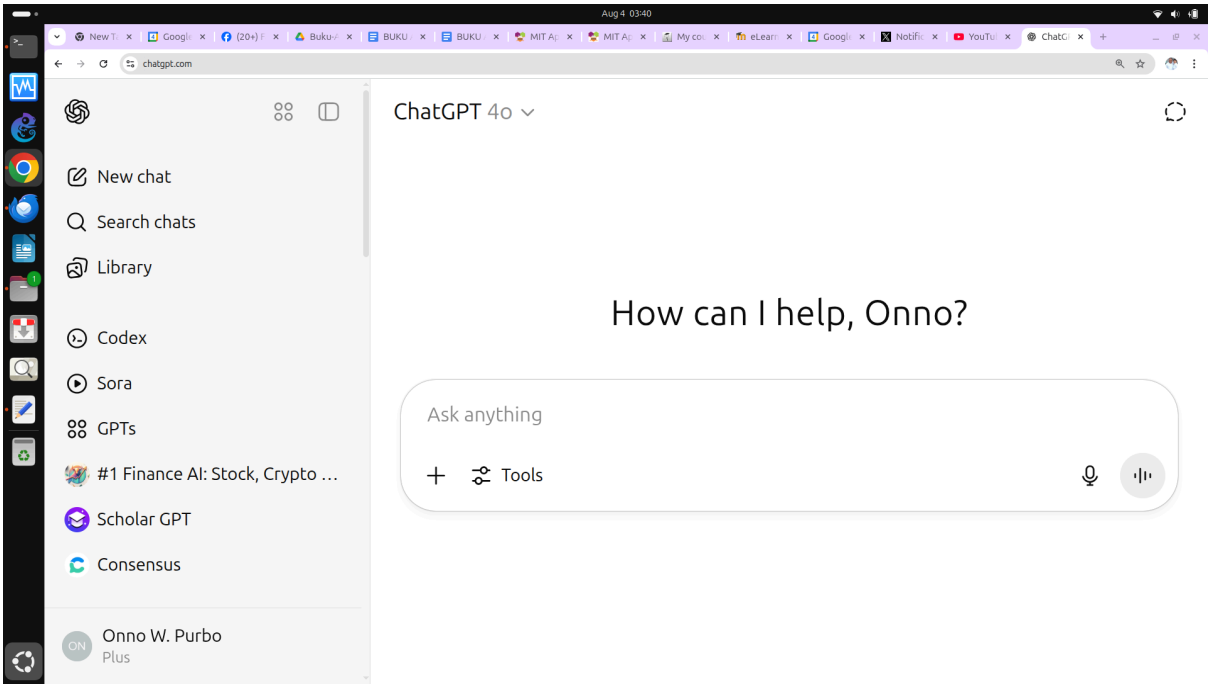
Pernah bingung cari ide untuk tugas atau proyek?
Bagaimana kalau ada "teman ngobrol" yang pintar, bisa bantu kasih ide, dan siap 24 jam?
Nah, sekarang kamu akan kenalan dengan ChatGPT, si asisten AI super kreatif!
Ayo cari tahu bagaimana cara memintanya bantu bikin ide aplikasi keren buatanmu sendiri!

Penjelasan Konsep (Teori)

Mengenal ChatGPT sebagai Asisten Ide

Di zaman sekarang, teknologi berkembang sangat cepat, termasuk kecerdasan buatan (AI). Salah satu contoh AI yang sangat menarik dan bermanfaat adalah **ChatGPT**. Tapi, apa sebenarnya ChatGPT itu?

Apa itu ChatGPT?



ChatGPT adalah sebuah program komputer berbasis AI yang bisa *mengerti* dan *menanggapi* tulisan kita. Kita bisa mengajukan pertanyaan, meminta saran, bahkan minta bantuan membuat ide kreatif. ChatGPT bukan manusia, tapi kemampuannya dalam menggunakan bahasa sangat mirip seperti manusia sungguhan!

Beberapa kemampuan ChatGPT antara lain:

- Menjawab pertanyaan dengan cepat dan masuk akal.
- Memberikan ide untuk proyek, tulisan, hingga resep makanan.
- Membantu merancang aplikasi, membuat cerita, dan banyak lagi!

Bagaimana ChatGPT bisa sepintar itu?

ChatGPT dilatih dengan membaca **jutaan halaman teks** dari berbagai sumber. Ia belajar mengenali pola bahasa, memahami konteks, dan merespon seperti manusia. Proses pelatihannya disebut **machine learning**. Dengan pelatihan ini, ChatGPT jadi bisa "berbicara" dan "berpikir" seperti teman ngobrol yang pintar.

Kamu bisa melihat antarmuka (tampilan layar) ChatGPT yang sederhana, seperti kotak chat biasa — tapi isinya luar biasa!

Contoh sederhana penggunaannya:

Misalnya, kamu punya beberapa bahan makanan di rumah: telur, mie, dan sayur. Lalu kamu bertanya ke ChatGPT:
"Apa saja makanan enak yang bisa aku masak dengan bahan ini?"
Dalam hitungan detik, kamu akan mendapat beberapa ide menarik. Mudah, kan?

Teknik Dasar Prompt Engineering untuk Ide Aplikasi

Untuk mendapatkan hasil terbaik dari ChatGPT, kita perlu tahu cara **memberi instruksi yang jelas**. Nah, teknik ini disebut **Prompt Engineering**.

Apa itu *Prompt Engineering*?

Prompt Engineering adalah **seni memberi perintah atau pertanyaan yang tepat** kepada AI supaya jawabannya sesuai harapan. Prompt yang baik ibarat kunci yang membuka jawaban terbaik dari ChatGPT.

Elemen Penting dalam Prompt yang Efektif:

1. **Konteks** – Jelaskan dulu situasinya. Misalnya, siapa yang akan memakai aplikasi tersebut?
2. **Instruksi Spesifik** – Hindari perintah yang terlalu umum. Tambahkan detail yang penting.
3. **Batasan atau Kriteria** – Misalnya, hanya ide yang bisa digunakan oleh anak SMP.
4. **Contoh (opsional)** – Kalau perlu, beri contoh agar AI lebih mengerti maksudmu.

Tips Menarik:

Mulailah dengan pertanyaan **umum**, lalu lanjutkan ke pertanyaan **spesifik**. Seperti membuka percakapan — semakin akrab, semakin dalam!

Ilustrasi Perbandingan:

- **Prompt Umum:** "Berikan ide aplikasi untuk mengurangi sampah."
- **Prompt Spesifik:** "Berikan 5 ide aplikasi untuk mengurangi sampah di sekolah, cocok untuk anak SMP."

Tentu saja, prompt yang spesifik biasanya menghasilkan jawaban yang lebih relevan dan kreatif!

Contoh Soal & Pembahasan Menarik

Soal:
"Kamu diminta membuat fitur baru untuk aplikasi game sepak bola. Coba minta bantuan ChatGPT."

Diskusi Prompt:

- Prompt 1 (kurang spesifik): *"Berikan ide fitur baru untuk game sepak bola."*
- Prompt 2 (lebih spesifik): *"Berikan 3 ide fitur baru untuk game sepak bola mobile, yang menarik untuk siswa SMP dan membuat permainan lebih seru."*

Kesimpulan:
Semakin jelas permintaan kita, semakin tepat jawaban ChatGPT. Jadi, latih dirimu untuk membuat prompt yang jelas dan menantang!

Brainstorming & Pengembangan Ide Aplikasi

Setelah mendapatkan ide dasar dari ChatGPT, kamu bisa **mengembangkan ide itu lebih jauh**!

Hal-hal yang bisa kamu minta dari ChatGPT:

- Memperluas fitur-fitur dari ide awal.
- Menentukan siapa target penggunanya.
- Membuat nama aplikasi yang keren dan mudah diingat.
- Menjelaskan manfaat aplikasi dan cara kerjanya.

Tips Penting:

Berinteraksilah bolak-balik! ChatGPT akan semakin "mengerti" arah pikiranmu jika kamu bertanya dan memberi tanggapan secara bertahap. Seperti berdiskusi dengan teman brainstorming.

Kesimpulan Seru:
Dengan belajar memahami konsep ChatGPT dan teknik Prompt Engineering, kamu bisa menjadi penemu ide keren, bahkan di usia SMP! Siapa tahu, ide aplikasimu suatu hari bisa jadi nyata!

Ilustrasi

Bayangkan kamu adalah seorang siswa bernama Raka. Suatu hari, Raka diminta oleh guru TIK untuk membuat ide aplikasi. Tapi Raka bingung. “Aku nggak pintar bikin ide, semuanya sudah pernah ada,” pikirnya. Ia hampir menyerah. Tapi temannya, Lala, mengenalkan Raka pada ChatGPT.

“Coba tanya AI ini, kayak kamu ngobrol sama orang. Misalnya: ‘Berikan ide aplikasi belajar yang cocok untuk anak SMP.’” Raka pun mencobanya.

Ternyata ChatGPT menjawab dengan cepat: “Aplikasi belajar berbasis game dengan fitur kuis dan ranking antar siswa.” Raka takjub. Ia pun terus bertanya lagi untuk mengembangkan ide tersebut, bahkan sampai aplikasi itu punya nama, fitur utama, dan target penggunanya. Raka akhirnya berhasil membuat ide yang keren—berkat bantuannya dari AI dan rasa penasarannya sendiri.

Contoh Kasus atau Ilustrasi

Kasus 1: “Rafa Bingung Buat Proyek, Lalu Bertemu ‘Teman Pintar’”

Ilustrasi:
Rafa mendapat tugas dari gurunya: buat ide aplikasi yang bermanfaat untuk lingkungan. Tapi dia benar-benar buntu. Saat itu, ia teringat pernah dengar tentang ChatGPT. Ia pun mengetik, “Aplikasi sederhana yang bisa membantu mengurangi sampah di sekolah untuk siswa SMP.”

Tantangan:
Bisakah kamu membantu Rafa memperbaiki prompt-nya agar lebih spesifik dan menarik? Coba bayangkan, kamu juga lagi duduk di meja belajar Rafa!

Kasus 2: “Nina Cari Ide Aplikasi untuk Jualan Teman”

Ilustrasi:
Temannya Nina jago bikin kue, tapi bingung gimana cara jualan ke teman-teman sekolah. Nina penasaran: bisa nggak ya bikin aplikasi kecil buat bantu jualan kue? Ia bertanya ke ChatGPT: “Bantu buat aplikasi jualan makanan buatan anak SMP.”

Tantangan:
Menurut kamu, apa saja informasi yang perlu Nina tambahkan ke prompt-nya supaya ChatGPT bisa kasih jawaban lebih tepat? Misalnya: siapa pengguna, fiturnya, atau tujuannya.

Kasus 3: “Farel Ingin Aplikasi Anti-Lupa PR”

Ilustrasi:
Farel sering lupa bawa PR. Ia ingin buat aplikasi pengingat khusus untuk pelajar SMP. Ia pun mengetik ke ChatGPT: “Tolong bantu buat aplikasi pengingat PR untuk anak sekolah.”

Tantangan:
Kira-kira, bagaimana cara membuat permintaan yang lebih spesifik? Bisa nggak kamu bantu Farel dengan ide fitur-fitur aplikasi yang bikin dia nggak malas pakai aplikasinya?

Kasus 4: “Tiara Jadi Detektif Prompt”

Ilustrasi:
Tiara iseng membandingkan dua temannya yang meminta bantuan ke ChatGPT. Yang satu bilang: “Beri ide aplikasi sekolah.” Yang lain bilang: “Berikan 3 ide aplikasi belajar matematika yang cocok untuk anak SMP dan bisa dimainkan saat istirahat.”

Tantangan:
Menurut kamu, mana yang lebih bagus? Kenapa? Yuk jadi detektif prompt seperti Tiara: analisa, bandingkan, lalu simpulkan!

Kasus 5: “Dino Ingin Bikin Game yang Tidak Biasa”

Ilustrasi:
Dino senang main game. Tapi dia bosan dengan game yang itu-itu saja. Ia ingin bikin ide game sepak bola yang punya fitur unik. Ia bertanya ke ChatGPT, tapi jawabannya kurang seru. Dino sadar, mungkin prompt-nya terlalu umum.

Tantangan:
Coba bantu Dino buat prompt yang lebih keren. Tambahkan hal-hal yang membuat game-nya berbeda: apakah ada avatar khusus? Mode turnamen antar kelas? Misi rahasia?

Kasus 6: “Zahra dan Proyek Aplikasi Kesehatan Remaja”

Ilustrasi:
Zahra ingin membuat aplikasi tentang kesehatan mental untuk remaja, tapi ia bingung mulai dari mana. Ia menulis: “Ide aplikasi tentang kesehatan remaja.” Jawaban ChatGPT terlalu luas.

Tantangan:
Coba bantu Zahra menyusun prompt dengan struktur: *siapa pengguna? masalah apa yang dihadapi? fitur apa yang diinginkan?* Bayangkan kamu adalah teman diskusi Zahra!

Setiap kasus ini akan melatih kamu untuk berpikir kreatif, mengasah kemampuan memberi instruksi, dan membiasakan diri berdiskusi dengan AI. Yuk, jadi pemecah masalah sambil bersenang-senang dengan teknologi!

Contoh Soal & Pembahasan

Soal 1: Kenalan Dulu Yuk!

Pertanyaan:
Bayangkan kamu belum pernah menggunakan ChatGPT. Coba jelaskan dengan

bahasamu sendiri: *Apa itu ChatGPT dan apa saja yang bisa dilakukannya untuk membantumu sebagai pelajar SMP?*

Pembahasan:
Jawaban ideal:

> ChatGPT adalah program pintar yang bisa diajak ngobrol lewat tulisan. Ia bisa bantu menjawab pertanyaan, kasih ide, bantu bikin cerita, bahkan bantu cari ide aplikasi atau tugas sekolah. ChatGPT sangat berguna kalau kita lagi butuh inspirasi atau ide baru.

Tujuan soal ini: Mengajak siswa memahami fungsi dasar ChatGPT dan manfaatnya dengan bahasa mereka sendiri.

Soal 2: Tantangan Ide Kilat!

Pertanyaan:
Kamu hanya punya waktu 30 detik untuk mengetik prompt ke ChatGPT. Mana di antara dua prompt berikut yang kemungkinan besar memberi jawaban lebih baik? Jelaskan alasannya.

- A: "Kasih ide aplikasi"
- B: "Kasih 3 ide aplikasi seru untuk siswa SMP yang suka hewan peliharaan"

Pembahasan:
Jawaban ideal:

> Prompt B lebih baik karena lebih jelas dan spesifik. Kita tahu siapa penggunanya (siswa SMP) dan topiknya (hewan peliharaan), jadi ChatGPT bisa memberikan jawaban yang lebih sesuai dan bermanfaat.

Tujuan soal ini: Melatih logika membuat *prompt efektif* dan mengenalkan prinsip dasar *Prompt Engineering*.

Soal 3: Bedah Prompt!

Pertanyaan:
Berikut sebuah prompt:

> *“Buatkan ide aplikasi edukasi tentang sejarah Indonesia untuk anak SMP, dilengkapi fitur kuis dan game.”*

Menurutmu, bagian mana dari prompt ini yang:

1. Menjelaskan **konteks**?
2. Memberi **instruksi spesifik**?
3. Menyebutkan **fitur atau batasan**?

Pembahasan:

1. Konteks: *"aplikasi edukasi tentang sejarah Indonesia untuk anak SMP"*
2. Instruksi spesifik: *"buatkan ide aplikasi edukasi"*
3. Fitur/batasan: *"dilengkapi fitur kuis dan game"*

Tujuan soal ini: Membiasakan siswa mengenali *struktur prompt* yang efektif.

Soal 4: Coba Tanya ChatGPT!

Pertanyaan:
Kamu sedang membuat aplikasi kebersihan sekolah. Pilih prompt yang paling efektif untuk meminta bantuan ide dari ChatGPT dan jelaskan alasannya.

- A: “Bantu dong ide aplikasi.”
- B: “Beri 3 ide aplikasi kebersihan sekolah yang cocok dipakai siswa SMP dan mudah dipakai di HP.”

Pembahasan:
Jawaban:

> Prompt B paling efektif karena menjelaskan topik, target pengguna, dan perangkatnya. Ini membuat ChatGPT bisa memberi jawaban yang lebih pas dan langsung bisa dikembangkan.

Tujuan soal ini: Mengajak siswa berpikir kritis tentang kualitas *perintah ke AI*.

Soal 5: Pengembangan Ide!

Pertanyaan:
Kamu mendapatkan ide awal dari ChatGPT: *“Aplikasi pemantau tanaman sekolah.”*
Tantanganmu: kembangkan ide itu jadi aplikasi utuh dengan menjawab 3 pertanyaan ini:

1. Siapa penggunanya?
2. Apa saja fitur aplikasinya?
3. Apa nama aplikasinya?

Pembahasan:
Contoh jawaban kreatif:

1. Pengguna: Siswa SMP pecinta lingkungan.
2. Fitur: Pencatatan pertumbuhan tanaman, alarm penyiraman, lomba antar kelas.
3. Nama: GreenBuddy.

Tujuan soal ini: Melatih siswa mengembangkan ide dasar menjadi *konsep aplikasi sederhana*.

Fun Facts

1. **ChatGPT itu seperti "teman ngobrol super pintar" yang bisa bantu kamu cari ide 24 jam non-stop!** Gak pernah tidur, gak pernah lelah!

2. **ChatGPT belajar dari jutaan halaman di internet** — kayak kamu baca semua buku di perpustakaan dalam semalam! 😱

3. **Prompt yang jelas = jawaban yang cerdas!** Kalau kamu pintar bikin pertanyaan, ChatGPT bisa jadi mesin ide paling jenius!

4. **Mau bikin aplikasi anti-gabut? ChatGPT bisa bantu cari ide aplikasi khusus buat siswa SMP.** Asal kasih tahu dia “maunya seperti apa.”

5. **Kamu bisa minta ChatGPT bantu buat fitur game, nama aplikasi, sampai manfaatnya — lengkap seperti proposal mini!**

6. **Cuma punya telur, mie, dan sayur? ChatGPT bisa bantu kamu jadi ‘MasterChef dadakan’!** Bayangkan kalau dia bantu kamu bikin aplikasi masak untuk anak kos 😆

7. **Teknik Prompt Engineering itu seperti kode rahasia** untuk membuka "lemari ide" ChatGPT yang penuh kejutan!

8. **Dengan satu kalimat yang tepat, kamu bisa bikin ChatGPT bantu rancang game sepak bola khusus buat siswa SMP.** Seru banget, kan?

9. **Kamu bisa diskusi bolak-balik dengan ChatGPT — kayak punya tim brainstorming pribadi!**

10. **Siapa sangka, dari ngobrol santai dengan AI, kamu bisa punya ide aplikasi yang suatu hari bisa viral!**

Kalimat-kalimat ini bisa disisipkan sebagai jeda ringan dalam materi, slide presentasi, atau kuis interaktif.

Tips Belajar

Tips Cepat Menghafal:

- Buat mind map dari satu ide utama.
- Gunakan warna berbeda dan gambar lucu!
- Hubungkan ide-ide dengan pengalaman pribadi.

Mind Map: "Aplikasi Pengingat Tugas"



Cerita Pengingat:
Bayangkan kamu punya robot kecil bernama “TugasKuy”. Setiap kali kamu lupa, dia muncul dan bilang, “Eh, tugas matematika belum dikerjakan lho!” Cerita lucu semacam ini bisa bantu kamu mengingat fungsi aplikasi dengan mudah.

Aktivitas Siswa

Aktivitas 1: Eksplorasi Seru: Kenalan dengan ChatGPT

Langkah-langkah:

- Ajak siswa membuka ChatGPT (atau versi simulasi jika tidak ada akses).
- Minta mereka mencoba menyapa dan bertanya hal-hal ringan, seperti "Apa kabar hari ini?" atau "Ceritakan tentang dinosaurus lucu!"
- Diskusikan bareng: “Menurutmu, ChatGPT itu pintar atau lucu?”, “Apa hal paling mengejutkan yang kamu temukan?”

Tujuan: Mengenalkan siswa pada ChatGPT sebagai "teman ngobrol pintar", dengan cara santai dan menyenangkan.

Aktivitas 2: Tantangan Ide Kreatif: Tebak-Tebakan Aplikasi

Langkah-langkah:

- Fasilitator memberikan contoh prompt: “Buat ide aplikasi yang bisa membantu pelajar bangun pagi.”
- Siswa mencoba membuat prompt versi mereka sendiri dan meminta ide dari ChatGPT.
- Tukar hasil dengan teman dan tebak: "Aplikasi siapa yang cocok untuk sekolah kita?"

Tujuan: Melatih keberanian eksplorasi dan kreativitas melalui percakapan interaktif dengan AI.

Aktivitas 3: Uji Ketajaman: Perbaiki Prompt-ku!

Langkah-langkah:

- Guru menampilkan dua versi prompt (umum dan spesifik), lalu berdiskusi: “Menurut kalian, mana yang lebih jelas?”
- Siswa diminta memperbaiki prompt umum menjadi lebih tajam dan menarik.
- Ajak siswa mencoba hasil prompt mereka langsung di ChatGPT dan bandingkan jawabannya!

Tujuan: Melatih kemampuan membuat instruksi yang jelas dan terstruktur (prompt engineering).

Aktivitas 4: Game Seru: Bikin Aplikasi Paling Unik

Langkah-langkah:

- Bagikan tema umum (misal: "lingkungan", "hobi", atau "belajar").
- Siswa bekerja dalam kelompok kecil, mengobrol dengan ChatGPT untuk mencari ide aplikasi paling unik.
- Presentasi singkat: “Inilah aplikasi aneh tapi keren dari kelompok kami!”

Tujuan: Melatih kolaborasi, berpikir kreatif, dan komunikasi ide.

Aktivitas 5: Simulasi Brainstorming: Obrolan Kaya Gagasan

Langkah-langkah:

- Siswa memilih satu ide aplikasi yang mereka suka.
- Ajak mereka berdiskusi dengan ChatGPT untuk mengembangkan:
    - Nama aplikasi
    - Fitur-fitur utama
    - Target pengguna dan manfaat
- Bagikan hasil brainstorming dalam bentuk peta konsep sederhana.

Tujuan: Melatih logika pengembangan ide, keterampilan bertanya, dan pemetaan konsep.

Aktivitas 6: Refleksi Kritis: Seberapa Baik Prompt-ku?

Langkah-langkah:

- Siswa melihat kembali prompt yang pernah mereka buat.
- Diskusi bersama: “Apa yang bisa diperbaiki?”, “Bagaimana membuatnya lebih spesifik atau lebih menarik?”
- Tantang diri mereka membuat *"prompt versi upgrade"*!

Tujuan: Melatih refleksi diri, evaluasi karya, dan penyempurnaan komunikasi dengan AI.

Aktivitas-aktivitas ini dirancang untuk mengajak siswa SMP **aktif berpikir, berdiskusi, dan mengeksplorasi teknologi AI secara menyenangkan**. Jika kamu ingin, saya juga bisa bantu buatkan lembar aktivitas atau panduan pengajaran dari ide-ide ini!

Rangkuman

Pernah merasa buntu saat harus cari ide tugas atau proyek? Tenang, sekarang kamu punya “teman ngobrol” baru yang siap bantu kapan saja: **ChatGPT**, si asisten AI super pintar! Di bab ini, kamu akan belajar cara memanfaatkan *ChatGPT* untuk mencari **ide aplikasi kreatif**, seperti aplikasi daur ulang, game edukatif, atau bahkan alat bantu belajar. Serunya, kamu bisa minta *ChatGPT* menjawab pertanyaan, memberi saran, bahkan bantu mikir nama keren untuk aplikasimu. Bayangkan kamu punya *partner brainstorming* yang gak pernah capek — seru banget, kan?

Supaya hasilnya maksimal, kamu juga perlu belajar teknik memberi perintah yang tepat ke AI, yang disebut **Prompt Engineering**. Dengan membuat **prompt** (perintah) yang jelas dan spesifik, kamu bisa mendapatkan jawaban yang lebih akurat dan menarik. Misalnya, daripada hanya bilang “beri ide aplikasi”, lebih baik kamu tambahkan detail: siapa penggunanya, tujuannya, dan batasannya. Prompt seperti “Berikan 3 ide aplikasi daur ulang yang cocok untuk siswa SMP” jauh lebih ampuh. Prompt yang baik itu ibarat “kunci rahasia” untuk membuka isi kepala ChatGPT!

Setelah dapat ide dari ChatGPT, jangan berhenti di situ! Kamu bisa mengembangkan ide tersebut jadi konsep aplikasi yang keren — dengan fitur tambahan, target pengguna yang jelas, nama aplikasi yang catchy, hingga penjelasan cara kerjanya. Diskusi dua arah dengan *ChatGPT* membuatmu berpikir lebih dalam, layaknya tim kreatif sungguhan. Dengan latihan membuat prompt yang seru dan eksplorasi ide yang berani, kamu bisa jadi pencipta aplikasi masa depan, meskipun masih SMP. **Siap tantang dirimu? Yuk mulai!**

Latihan Soal

A. Benar atau Salah (5 Soal)

1. ChatGPT bisa membantu kita mencari ide aplikasi.
   Jawaban: Benar

2. Prompt yang panjang pasti selalu lebih baik daripada yang pendek.
   Jawaban: Salah

3. ChatGPT dapat memberikan jawaban berdasarkan informasi yang pernah dipelajarinya.
   Jawaban: Benar

4. ChatGPT hanya bisa menjawab soal Matematika, tidak bisa memberi ide aplikasi.
   Jawaban: Salah

5. Semakin spesifik prompt yang kita berikan, semakin baik hasil dari ChatGPT.
   Jawaban: Benar

B. Pilihan Ganda (10 Soal)

6. Apa itu ChatGPT?
   A. Aplikasi game
   B. Asisten AI yang bisa menghasilkan teks
   C. Mesin pencetak gambar
   D. Kamera pintar
   Jawaban: B

7. Apa tujuan dari prompt engineering?
   A. Mengedit gambar AI
   B. Membuat teks jadi puisi
   C. Memberi instruksi jelas ke AI
   D. Menghitung angka secara cepat
   Jawaban: C

8. Berikut ini adalah contoh *prompt* yang baik, kecuali:
   A. “Buatkan 5 ide aplikasi belajar untuk anak SMP”
   B. “Kasih saya ide aplikasi”
   C. “Berikan contoh aplikasi yang ramah lingkungan untuk sekolah”
   D. “Bantu saya cari ide aplikasi olahraga untuk remaja”
   Jawaban: B

9. Jika kamu ingin ChatGPT memberi ide aplikasi khusus untuk sekolah, maka kamu harus...
   A. Memberi pertanyaan umum
   B. Menyuruh teman menuliskannya

C. Menggunakan prompt yang spesifik
D. Tidak perlu menulis apapun
Jawaban: C

10. Dalam membuat ide aplikasi, ChatGPT bisa membantu dalam hal berikut, kecuali:
A. Menentukan warna seragam sekolah
B. Memberi nama aplikasi
C. Menentukan fitur aplikasi
D. Menentukan target pengguna
Jawaban: A

11. Mengapa penting untuk berdiskusi beberapa kali dengan ChatGPT saat mengembangkan ide?
A. Agar hasil selalu lucu
B. Agar ide makin matang dan spesifik
C. Supaya bisa main game
D. Supaya cepat selesai
Jawaban: B

12. Ketika kamu memberi batasan dalam prompt (misalnya: hanya untuk SMP), maka...
A. ChatGPT bingung
B. Jawaban jadi lebih fokus
C. Jawaban jadi membosankan
D. Jawaban selalu sama
Jawaban: B

13. Berikut ini adalah manfaat ChatGPT dalam membuat ide aplikasi:
A. Menjawab soal ulangan
B. Menyusun laporan keuangan
C. Membantu brainstorming
D. Menghitung kecepatan mobil
Jawaban: C

14. Jika kamu ingin membuat aplikasi yang mengurangi sampah di sekolah, kamu bisa bertanya ke ChatGPT dengan prompt:
A. “Tolong buatkan puisi tentang sampah”
B. “Apa itu sampah?”
C. “Berikan 5 ide aplikasi untuk mengurangi sampah di sekolah”
D. “Siapa yang buang sampah sembarangan?”
Jawaban: C

15. Fungsi dari elemen *konteks* dalam prompt adalah...
A. Menjelaskan suasana hati
B. Menyebutkan lokasi pengguna
C. Memberi latar belakang yang membantu ChatGPT memahami masalah
D. Mempercepat loading ChatGPT
Jawaban: C

C. Isian Singkat (5 Soal)

16. ChatGPT adalah contoh dari teknologi _____.
    Jawaban: AI generatif

17. Seni membuat pertanyaan yang baik dan tepat untuk AI disebut _____.
    Jawaban: prompt engineering

18. Fitur, target pengguna, dan nama aplikasi bisa dibantu oleh _____.
    Jawaban: ChatGPT

19. Prompt yang terlalu umum akan menghasilkan jawaban yang kurang _____.
    Jawaban: spesifik

20. Kita bisa menggunakan ChatGPT sebagai teman _____ ide.
    Jawaban: brainstorming

D. Soal Eksplorasi (3 Soal)

Soal:

Bayangkan kamu diminta membuat aplikasi untuk membantu siswa yang sering lupa jadwal pelajaran. Tuliskan **prompt lengkap** yang bisa kamu berikan ke ChatGPT agar menghasilkan ide aplikasi yang sesuai. Pastikan kamu menyebutkan siapa pengguna, tujuannya, dan fitur yang diinginkan.

Soal:

Kamu dan kelompokmu punya ide aplikasi untuk mengurangi stres pada remaja. Gunakan ChatGPT untuk mengembangkan **3 fitur utama** yang bisa membuat aplikasi ini menarik dan bermanfaat. Tuliskan fitur-fitur yang disarankan ChatGPT dan alasan kamu memilihnya.

Soal:

Coba cari ide aplikasi sosial dari ChatGPT yang bisa membantu komunitas lokal di lingkungan tempat tinggalmu. Tulis **hasil percakapanmu dengan ChatGPT,** dan sebutkan bagaimana kamu akan mengembangkan ide itu menjadi aplikasi nyata. Gunakan minimal satu prompt eksploratif dan satu prompt spesifik.

Tugas Proyek

Proyek 1: Ide Aplikasi Ramah Lingkungan di Sekolah

Deskripsi:
Gunakan ChatGPT untuk menemukan dan mengembangkan ide aplikasi sederhana yang dapat membantu mengurangi sampah atau menjaga kebersihan sekolah.

Langkah-Langkah:

1. Tanyakan pada ChatGPT: *"Berikan 3 ide aplikasi yang bisa membantu siswa SMP menjaga kebersihan lingkungan sekolah."*
2. Pilih satu ide yang paling menarik.
3. Diskusikan fitur utama yang dibutuhkan.
4. Buat sketsa tampilan aplikasi (misalnya di kertas atau pakai PowerPoint/Canva).
5. Presentasikan ide dan cara kerja aplikasi di depan kelas.

Catatan Seru:
Kamu juga bisa minta ChatGPT bantu cari nama aplikasinya yang catchy, misalnya: “EcoBuddy” atau “SekolahHijau”.

Proyek 2: Game Edukasi Karya Sendiri

Deskripsi:
Minta bantuan ChatGPT untuk mencari ide game edukasi yang seru untuk anak SMP, lalu kembangkan konsep dasarnya.

Langkah-Langkah:

1. Prompt awal: *"Beri 3 ide game edukasi mobile untuk anak SMP yang menyenangkan dan membantu belajar pelajaran sekolah."*
2. Pilih satu ide (misalnya game matematika petualangan atau kuis sains).
3. Kembangkan level-level gamenya dengan bantuan ChatGPT.
4. Buat storyboard: urutan tampilan dari awal sampai akhir permainan.
5. Simulasikan cara mainnya melalui presentasi atau mockup.

Tantangan Bonus:
Tanyakan ChatGPT fitur tambahan untuk membuat gamenya makin menantang!

Proyek 3: Aplikasi Pengingat Tugas Sekolah

Deskripsi:
Bersama ChatGPT, ciptakan ide aplikasi pengingat tugas dan ulangan agar siswa lebih tertib dan tidak lupa.

Langkah-Langkah:

1. Tulis prompt: *"Ide aplikasi pengingat tugas dan ulangan untuk siswa SMP yang mudah digunakan."*
2. Tentukan fitur penting: pengingat harian, kalender tugas, catatan deadline.
3. Buat wireframe sederhana (gambar alur tampilan aplikasi).
4. Diskusikan bagaimana aplikasinya bisa membantu kebiasaan belajar.
5. Presentasikan konsep kepada teman-teman, siapa tahu mereka juga butuh!

Tips Tambahan:
Minta ChatGPT bantu membuat slogan aplikasi yang singkat dan mudah diingat.

Proyek 4: Ide Aplikasi Hobi dan Minat Pribadi

Deskripsi:
Gunakan ChatGPT untuk mencari ide aplikasi berdasarkan hobi atau minatmu (misalnya menggambar, membaca, musik, dll.).

Langkah-Langkah:

1. Prompt: *"Berikan 3 ide aplikasi kreatif untuk anak SMP yang hobi [sebutkan hobimu]."*
2. Pilih ide yang paling sesuai dengan minatmu.
3. Kembangkan fitur-fitur yang relevan dan menyenangkan.
4. Buat deskripsi aplikasi, siapa penggunanya, dan apa manfaatnya.
5. Boleh tambahkan desain logo dengan bantuan aplikasi desain sederhana.

Kejutan Seru:
Minta ChatGPT buatkan tagline dan deskripsi aplikasi seperti di Play Store!

Proyek 5: Ide Aplikasi Anti Stres untuk Pelajar

Deskripsi:
Dengan ChatGPT, cari dan rancang ide aplikasi sederhana yang bisa membantu mengurangi stres pelajar—misalnya meditasi ringan, tips motivasi, atau permainan santai.

Langkah-Langkah:

1. Prompt: *"Berikan ide aplikasi sederhana untuk membantu anak SMP mengurangi stres atau kecemasan belajar."*
2. Tentukan apakah aplikasinya berbasis audio, teks, atau interaktif.
3. Buat daftar fitur seperti "kata motivasi harian", "waktu jeda", atau "game ringan".
4. Diskusikan siapa yang akan memakai aplikasi dan kapan digunakan.
5. Susun proposal mini: tujuan, pengguna, manfaat, dan tampilan aplikasi.

Tips Fun:
Kamu juga bisa minta ChatGPT menuliskan “kata-kata penyemangat” untuk ditampilkan dalam aplikasimu!

Proyek 6: Tantangan Prompt Kreatif Mingguan

Deskripsi:
Buat proyek tantangan di mana kamu dan teman-teman membuat prompt ChatGPT yang paling kreatif untuk menghasilkan ide aplikasi luar biasa setiap minggu.

Langkah-Langkah:

1. Setiap minggu, tentukan tema (misalnya: teknologi, lingkungan, hiburan, sekolah).
2. Masing-masing siswa membuat 1 prompt menarik dan berbagi hasilnya.

3. Diskusikan ide-ide aplikasi dari hasil prompt yang berbeda.
4. Voting: pilih ide paling kreatif dan aplikatif.
5. Dokumentasikan semua ide ke dalam buku proyek kelas.

Bonus Menantang:
Siapa yang bisa membuat *prompt terbaik* minggu ini?

BAB 7: Presentasi & Evaluasi Aplikasi

Tujuan Pembelajaran

Setelah mengikuti pembelajaran ini, kamu dapat:

1. Menjelaskan mengapa presentasi itu penting.
2. Menyusun presentasi dengan struktur yang jelas dan menarik.
3. Menampilkan aplikasi atau ide secara percaya diri melalui demo.
4. Berkomunikasi secara verbal dan nonverbal dengan baik saat presentasi.
5. Memberikan dan menerima feedback secara sopan dan membangun.
6. Memahami konsep *iterasi* atau perbaikan terus-menerus dalam proyek.

Peta Konsep



Apersepsi

Pernah punya ide keren tapi bingung menjelaskannya ke teman atau guru?
Nah, bayangkan kamu membuat aplikasi pintar pengingat minum air. Hebat, kan? Tapi... kalau kamu tidak bisa menjelaskannya dengan baik, siapa yang mau pakai?

Di sinilah kamu belajar jadi **pembicara muda yang keren dan percaya diri!**
Dengan latihan presentasi, kamu tidak hanya menyampaikan ide, tapi juga **menyentuh pikiran dan hati audiens**.

Yuk, tantang dirimu!

> "Siapkan proyekmu, latih presentasimu, dan buktikan kamu siap jadi inovator muda masa depan!"

Penjelasan Konsep (Teori)

Mengapa Presentasi Itu Penting?

Dalam dunia nyata, ide yang hebat saja belum cukup. Kita juga perlu bisa **menjelaskan dan meyakinkan orang lain** tentang ide kita. Di sinilah pentingnya sebuah **presentasi yang jelas, menarik, dan terstruktur dengan baik**. Presentasi adalah jembatan antara **proyek yang sudah kita buat** dan **pemahaman orang lain terhadap proyek tersebut**.

> Misalnya, kamu sudah membuat aplikasi keren untuk mengingatkan orang minum air putih. Tapi kalau kamu tidak bisa menjelaskannya dengan baik, orang lain bisa saja bingung atau tidak tertarik.

Elemen-Elemen Presentasi yang Efektif

Agar presentasimu berhasil, perhatikan struktur dan elemen-elemen berikut ini:

Struktur Presentasi yang Disarankan

Bayangkan presentasi seperti bercerita. Cerita yang baik punya awal, tengah, dan akhir. Maka, susunlah presentasimu seperti ini:

1. **Pendahuluan**: Perkenalkan dirimu dan tujuan presentasi. Bangun rasa penasaran audiens.
2. **Masalah**: Jelaskan masalah nyata yang ingin kamu selesaikan.
3. **Solusi**: Tunjukkan ide atau produk sebagai solusi dari masalah tersebut.
4. **Fitur Utama**: Jelaskan fitur-fitur penting dan menarik dari aplikasi atau proyekmu.

5. **Demo Langsung**: Tampilkan cara kerja aplikasi secara nyata.
6. **Kesimpulan**: Ringkas semua poin utama dan sampaikan harapan atau ajakan.

Visual yang Menarik

Gunakan gambar, diagram, atau slide yang membantu audiens memahami apa yang kamu jelaskan. Ingat: visual bukan hiasan, tapi alat bantu komunikasi!

Komunikasi Verbal dan Nonverbal yang Efektif

- Kontak mata: Tunjukkan kamu percaya diri dan menghargai audiens.
- Intonasi suara: Jangan datar! Mainkan nada untuk menekankan poin penting.
- Bahasa tubuh: Gerakan tangan dan ekspresi wajah bisa memperkuat pesan.



Mendemonstrasikan Aplikasi dan Sesi Tanya Jawab

Bagaimana Cara Melakukan Demo Aplikasi yang Sukses?

Demo aplikasi adalah momen emas! Di sinilah kamu menunjukkan bahwa aplikasi kamu **benar-benar bekerja**, bukan hanya ide di atas kertas.

Tips penting:

- **Pastikan aplikasi bisa berjalan lancar** sebelum presentasi.
- **Latih demo-nya**, seolah kamu sedang tampil sungguhan.
- **Gunakan narasi** saat demo: “Sekarang saya klik tombol ini, lalu akan muncul notifikasi...”

Sesi Tanya Jawab: Siap Menjawab!

Setelah demo, biasanya akan ada sesi tanya jawab. Ini kesempatan untuk menunjukkan bahwa kamu **menguasai proyekmu**.

Cara Mempersiapkan Diri:

- Coba **bayangkan pertanyaan** yang mungkin ditanyakan audiens.
- Siapkan **jawaban singkat, jelas, dan tidak bertele-tele**.

- Jika tidak tahu jawabannya, jujur saja, dan katakan kamu akan mencari tahu. Itu lebih baik daripada mengarang.

Menerima dan Memberikan Feedback Konstruktif

Apa Itu Feedback dan Kenapa Penting?

Feedback adalah masukan atau tanggapan dari orang lain terhadap proyek kita. Ini sangat penting agar kita bisa terus memperbaiki dan menyempurnakan aplikasi atau ide kita.

> Bayangkan kamu membuat kue. Rasanya enak menurutmu, tapi temanmu bilang terlalu manis. Kalau kamu mendengarkan feedback itu, kamu bisa membuat kue yang lebih pas untuk semua orang.



Cara Menerima Feedback dengan Baik:

- **Mendengarkan tanpa menyela.**
- Jangan langsung membela diri, cobalah pahami maksudnya.
- Catat poin penting dari feedback yang kamu terima.

Cara Memberikan Feedback yang Membangun:

- Fokus pada **apa yang bisa diperbaiki**, bukan hanya mengkritik.
- Gunakan bahasa yang sopan dan jelas.
- Coba gunakan format ini:
  “Saya suka bagian ini…, tapi mungkin bisa lebih baik kalau…”

Konsep Iterasi: Perbaikan Terus-Menerus

Dalam dunia teknologi dan kreativitas, produk **jarang langsung sempurna** di versi pertama. Di sinilah pentingnya konsep **iterasi**, yaitu proses memperbaiki produk berdasarkan feedback yang diterima.

Iterasi berarti:

- Menerima masukan.
- Memperbaiki produk.
- Mencoba lagi.
- Menerima masukan baru.
- Dan seterusnya...

Contoh Soal & Tantangan:

Pertanyaan:
Saat kamu mempresentasikan sebuah aplikasi, apa saja hal yang perlu kamu jelaskan agar orang lain mengerti?

Pembahasan:

1. Tujuan aplikasi (masalah apa yang ingin diselesaikan).
2. Fitur utama (apa yang membuat aplikasi berguna atau unik).
3. Cara kerja aplikasi secara singkat (termasuk jika ada teknologi seperti AI).
4. Hasil atau manfaat yang bisa dirasakan pengguna.

Jika kamu memahami dan menerapkan semua hal diatas, maka kamu siap menjadi **presenter muda yang percaya diri, komunikatif, dan inovatif!**

Siap tantang dirimu? Coba siapkan presentasi proyekmu dan latih demo-nya sekarang juga!

Ilustrasi & Perjuangan Mengatasinya

Bayangkan kamu dan timmu sudah membuat aplikasi keren untuk membantu teman-teman belajar matematika. Tapi, saat presentasi di depan kelas… kamu gugup, lupa apa yang harus dijelaskan, dan aplikasimu malah error! Rasanya malu dan kecewa. Tapi tenang, ini adalah proses belajar.

Dina, siswi kelas 8, juga pernah mengalaminya. Saat pertama kali presentasi, suaranya pelan, dan ia tidak percaya diri. Namun setelah latihan berulang, menonton contoh presentasi di YouTube, dan meminta masukan dari guru dan teman, akhirnya ia bisa tampil percaya diri. Presentasinya menjadi inspirasi bagi teman-temannya. Artinya? Semua orang bisa jadi hebat kalau mau berlatih!

Contoh Kasus / dan Ilustrasi

Kasus 1: Aplikasi Pengingat Minum Air – Tapi Temanmu Malah Bingung!

Ilustrasi:
Bayangkan kamu membuat aplikasi keren bernama "Minum Yuk!" yang akan mengingatkan orang untuk minum air putih setiap dua jam. Tapi saat kamu menjelaskan aplikasimu di depan kelas, teman-temanmu malah saling melirik, bingung, bahkan ada yang menguap.
Tantangannya:
Apa yang kurang? Mungkin kamu terlalu cepat bicara, tidak menunjukkan demo, atau tidak menyampaikan kenapa aplikasi ini penting.

Pelajaran: Tanpa penjelasan yang jelas dan menarik, ide hebat pun bisa tidak dipahami.

Kasus 2: “Kucing Cerdas” Tapi Tidak Menjawab Pertanyaan

Ilustrasi:
Kamu dan timmu membuat chatbot bernama “Kucing Cerdas” untuk membantu orang belajar IPA. Saat demo, aplikasi berjalan lancar, tapi saat sesi tanya jawab, kamu kebingungan saat ditanya:
"Kenapa chatbot-nya pakai gambar kucing, bukan robot?"
Kamu tidak siap, wajahmu panik, dan hanya menjawab, “Ya… iseng aja sih.”
Tantangannya:
Apa yang harus dilakukan jika tidak tahu jawaban?

Pelajaran: Persiapan tanya-jawab itu bagian penting dari presentasi. Lebih baik jujur daripada asal jawab.

Kasus 3: Tim Kamu Punya Aplikasi Keren, Tapi Slide-Nya Membosankan

Ilustrasi:
Kelompokmu membuat aplikasi “Tugas Ringan” untuk mengatur jadwal belajar. Tapi ketika presentasi, slide-nya hanya penuh teks. Tidak ada gambar, diagram, atau ilustrasi. Audiens jadi kehilangan minat, padahal aplikasi keren!
Tantangannya:
Bagaimana caranya membuat visual yang menarik dan membantu?

Pelajaran: Visual bukan sekadar hiasan — mereka adalah alat bantu komunikasi!

Kasus 4: “Suara Monoton” Mengubur Ide Brilian

Ilustrasi:
Kamu punya solusi pintar: alat pendeteksi banjir otomatis. Tapi saat presentasi, suaramu sangat datar, seperti robot. Tidak ada penekanan atau antusiasme.
“Ini… alat pendeteksi banjir… lalu… air naik… sensor nyala…”
Audiens jadi tidak semangat meski idenya luar biasa.

Tantangannya:
Bagaimana menghidupkan suasana saat bicara?

Pelajaran: Intonasi, ekspresi, dan kontak mata adalah senjata utama presenter!

Kasus 5: Demo Aplikasi Gagal Total – Apa yang Harus Dilakukan?

Ilustrasi:
Kamu siap tampil, semua latihan sudah dilakukan. Tapi tiba-tiba, saat kamu klik tombol “Mulai”, aplikasi tidak jalan! Teman-temanmu menahan tawa.
Tantangannya:
Apa yang sebaiknya dilakukan? Panik atau tetap tenang dan jelaskan apa yang *seharusnya* terjadi?

Pelajaran: Siapkan *plan B*, dan tetap percaya diri meski ada gangguan teknis.

Kasus 6: Feedback Pedas Bikin Kesal – Tapi Ternyata Bermanfaat!

Ilustrasi:
Setelah presentasi, salah satu teman berkata, “Menurutku aplikasimu terlalu biasa, mirip aplikasi lain.”
Awalnya kamu kesal. Tapi setelah dipikir-pikir, kamu jadi mencari ide tambahan agar lebih unik, dan versi berikutnya jadi jauh lebih keren!
Tantangannya:
Bagaimana caramu merespons feedback?

Pelajaran: Feedback itu bukan serangan — tapi bahan bakar untuk berkembang!

Kalau kamu bisa belajar dari keenam kasus ini, kamu bukan hanya bisa membuat aplikasi hebat, tapi juga **menjadi presenter muda yang siap bersinar!**
Siapkah kamu menghadapi tantangan berikutnya?

Contoh Soal dan Pembahasan

Soal 1: Struktur Presentasi

Pertanyaan:
Kamu sedang membuat presentasi tentang aplikasi pengingat minum air putih. Dari urutan berikut, manakah struktur presentasi yang paling efektif?

A. Demo aplikasi – Perkenalan – Fitur – Masalah – Kesimpulan
B. Masalah – Perkenalan – Solusi – Fitur – Kesimpulan
C. Perkenalan – Masalah – Solusi – Fitur – Demo – Kesimpulan
D. Perkenalan – Solusi – Demo – Masalah – Fitur – Kesimpulan

Pembahasan:
Struktur presentasi yang baik menyerupai alur cerita: ada pembuka, isi, dan penutup. Urutan yang benar yaitu:

1. **Perkenalan**
2. **Masalah**
3. **Solusi**
4. **Fitur**
5. **Demo**
6. **Kesimpulan**

Jadi, **jawaban yang benar adalah C.**

Soal 2: Visual yang Efektif

Pertanyaan:
Kamu membuat slide presentasi. Dari pilihan berikut, manakah yang *bukan* tujuan penggunaan visual dalam presentasi?

A. Membuat presentasi tampak keren
B. Membantu audiens memahami informasi
C. Menjelaskan hal yang sulit dengan gambar
D. Menarik perhatian ke poin penting

Pembahasan:
Visual dalam presentasi **bukan hanya hiasan**, melainkan alat bantu komunikasi. Jadi, jika tujuanmu **hanya membuatnya keren**, itu bukan alasan utama.
Jawaban: **A.**

Soal 3: Komunikasi yang Meyakinkan

Pertanyaan:
Kamu gugup saat presentasi dan melihat layar terus-menerus. Apa yang seharusnya kamu lakukan agar terlihat lebih percaya diri?

A. Membaca teks presentasi cepat-cepat
B. Menjaga kontak mata dengan audiens
C. Berdiri diam tanpa bergerak
D. Bicara pelan agar tidak salah

Pembahasan:
Salah satu cara menunjukkan kepercayaan diri adalah **menjaga kontak mata**. Ini membuat audiens merasa diperhatikan dan membangun kepercayaan.
Jawaban: **B.**

Soal 4: Tantangan Demo Aplikasi

Pertanyaan:
Sebelum melakukan demo aplikasi di depan kelas, kamu sebaiknya melakukan hal berikut, *kecuali*:

A. Mencoba aplikasi berulang kali
B. Mempersiapkan narasi demo
C. Meninggalkan demo untuk spontanitas
D. Latihan seolah presentasi sungguhan

Pembahasan:
Melakukan demo tanpa latihan bukan ide bagus. Kamu perlu **berlatih dan mempersiapkan narasi**, bukan hanya mengandalkan spontanitas.
Jawaban: **C.**

Soal 5: Feedback dan Iterasi

Pertanyaan:
Setelah presentasi, temanmu memberi saran agar tampil lebih bersemangat. Bagaimana sikap terbaikmu?

A. Langsung membela diri karena kamu sudah berusaha
B. Mengabaikannya karena itu hanya pendapat
C. Menerima feedback dan mencoba memperbaiki di latihan berikutnya
D. Marah karena kamu merasa kritiknya tidak sopan

Pembahasan:
Feedback adalah cara agar kita bisa terus **berkembang dan berlatih menjadi lebih baik.** Menerimanya dengan terbuka dan melakukan **iterasi (perbaikan)** adalah kunci sukses.
Jawaban: **C.**

Soal 6: Ayo Tunjukkan Siapa Kamu!

Pertanyaan Tantangan:
Bayangkan kamu membuat aplikasi pengingat belajar. Coba sebutkan **empat hal penting** yang akan kamu sampaikan saat presentasi aplikasimu.

Pembahasan Singkat:

1. Tujuan aplikasi (masalah apa yang diselesaikan).
2. Fitur utama (fitur-fitur unggulan).
3. Cara kerja aplikasi secara singkat.
4. Manfaat atau hasil bagi pengguna.

Jika kamu bisa menjawab soal-soal ini, artinya kamu siap menjadi **presenter muda yang jago, percaya diri, dan inovatif!** Mau coba tantangan berikutnya? Yuk, latihan demo aplikasimu sekarang juga!

Fun Facts

- Steve Jobs (pendiri Apple) berlatih ratusan kali sebelum presentasi produk!
- 93% komunikasi saat presentasi berasal dari **intonasi dan bahasa tubuh**, bukan kata-kata!
- Banyak aplikasi sukses lahir dari ide sederhana, tapi dipresentasikan dengan cara yang **kreatif dan jelas**.
- Presentasi pertama Mark Zuckerberg tentang Facebook… sangat sederhana dan dilakukan di kelas kecil!

Tips Belajar

Menyusun Presentasi

Gunakan Mind Mapping untuk menyusun isi presentasi:



Cerita Nyata:
Bayangkan kamu sedang menceritakan kisah kepada temanmu. Ubah isi presentasi jadi cerita. Ini akan lebih mudah diingat dan menarik untuk disampaikan!

Gunakan singkatan atau akronim, misalnya:
"PMSFDK" (Pendahuluan, Masalah, Solusi, Fitur, Demo, Kesimpulan)

Aktivitas Siswa

Aktivitas 1: “Cerita di Balik Inovasi”

Tujuan: Melatih siswa menyusun struktur presentasi secara menarik.

Langkah-langkah:

- Ajak siswa membayangkan mereka sedang membuat presentasi seperti bercerita: ada awal (pendahuluan), konflik (masalah), solusi, hingga penutup.
- Tantang mereka untuk menyusun alur cerita dari proyek mereka (misalnya aplikasi buatan mereka) dalam urutan logis dan menyenangkan.
- Dorong mereka menyampaikan presentasi seperti sedang menceritakan kisah yang ingin didengar orang banyak.

Aktivitas 2: “Presenter Hebat, Bukan Sekadar Bicara”

Tujuan: Mengembangkan kemampuan komunikasi verbal dan nonverbal saat presentasi.

Langkah-langkah:

- Melatih siswa untuk melakukan kontak mata, memainkan intonasi suara, serta menggunakan gerakan tangan saat berbicara.
- Ciptakan simulasi: ajak mereka berdiri di depan teman sekelas dan mempresentasikan satu ide sederhana, lalu diskusikan kekuatan dan hal yang bisa ditingkatkan dari cara mereka berbicara.
- Beri tantangan: Siapa bisa membuat audiens paling fokus hanya dengan ekspresi dan suara?

Aktivitas 3: “Demo Day Mini Show”

Tujuan: Membiasakan siswa mendemonstrasikan produk atau proyek digital secara percaya diri.

Langkah-langkah:

- Buat latihan simulasi demo, di mana siswa berpura-pura memperkenalkan dan menunjukkan cara kerja aplikasinya.
- Ajak mereka berlatih berbicara sambil mengarahkan perhatian audiens pada fitur di layar.
- Tantang mereka untuk tidak hanya menunjukkan, tapi juga menjelaskan “mengapa fitur itu penting.”

Aktivitas 4: “Jagoan Tanya Jawab”

Tujuan: Melatih kesiapan siswa dalam menjawab pertanyaan audiens dengan percaya diri dan jujur.

Langkah-langkah:

- Buat sesi permainan peran: satu kelompok menjadi presenter, kelompok lain menjadi “tim penasaran” yang mengajukan pertanyaan.
- Dorong siswa untuk menjawab singkat, jelas, dan tidak panik.
- Diskusikan bersama: bagaimana cara menjawab jika tidak tahu? Jawaban jujur pun bisa tetap terlihat profesional.

Aktivitas 5: “Feedback Challenge”

Tujuan: Membangun budaya memberi dan menerima masukan secara konstruktif.

Langkah-langkah:

- Ajak siswa memberikan masukan terhadap presentasi teman sekelas dengan format positif: “Saya suka bagian..., mungkin akan lebih baik jika…”
- Latih mereka menerima feedback tanpa membela diri, cukup dengan mendengarkan, mencatat, dan mengucapkan terima kasih.
- Beri apresiasi bagi siswa yang berhasil mengubah feedback menjadi perbaikan nyata.

Aktivitas 6: “Level Up dengan Iterasi”

Tujuan: Menanamkan pemahaman bahwa kesempurnaan lahir dari proses berulang dan penyempurnaan.

Langkah-langkah:

- Tunjukkan bahwa proyek hebat biasanya melewati banyak revisi.
- Tantang siswa untuk memperbaiki versi pertama presentasi atau aplikasi mereka berdasarkan masukan yang diterima.
- Rayakan setiap versi baru sebagai "Level Up", seperti naik level dalam permainan.

Keenam aktivitas ini dirancang agar siswa tidak hanya *tahu teori presentasi*, tetapi juga **mengalaminya secara langsung, menyenangkan, dan penuh tantangan**. Siap menjadi presenter muda yang keren dan percaya diri? Yuk, mulai dari aktivitas pertama!

Rangkuman

Kenapa sih presentasi itu penting banget? Gini ya, punya ide keren aja belum cukup. Kamu juga harus bisa *menjual* ide itu ke orang lain. Presentasi yang baik bikin ide kamu jadi hidup dan dimengerti. Misalnya, kamu bikin aplikasi pengingat minum air, tapi kalau gak bisa menjelaskannya dengan jelas dan menarik, orang-orang bisa aja cuek. Nah, biar presentasimu keren, kamu butuh struktur seperti bercerita: mulai dari **pendahuluan yang bikin penasaran**, **masalah yang nyata**, **solusi kreatif**, lalu jelaskan **fitur utama aplikasi**,

lakukan **demo langsung**, dan akhiri dengan **kesimpulan yang mengajak**. Jangan lupa, tampilan visual seperti gambar dan slide itu bukan hiasan, tapi **alat bantu komunikasi**!

Sekarang bayangin kamu lagi tampil di depan orang banyak. Supaya tampil percaya diri, jaga **kontak mata**, mainkan **intonasi suara**, dan gunakan **bahasa tubuh yang ekspresif**. Saat demo, pastikan aplikasi lancar jaya! Latih demo-nya kayak kamu lagi tampil di TV, menggunakan narasi biar audiens paham: “Sekarang saya klik tombol ini, lalu muncul notifikasi…” Setelah itu, siap-siap deh menghadapi sesi tanya jawab. Jangan takut! **Persiapkan jawaban singkat dan jelas**, dan kalau belum tahu jawabannya, gak apa-apa bilang jujur, yang penting kamu mau cari tahu.

Terakhir, kamu harus terbuka dengan **feedback** alias masukan. Jangan langsung defensif kalau ada yang kasih kritik. Dengerin dulu, catat poin penting, lalu gunakan untuk memperbaiki proyekmu. Saat ngasih feedback ke teman juga, pakai cara yang sopan dan membangun, misalnya: “Aku suka bagian ini, tapi mungkin bisa lebih baik kalau…” Di dunia teknologi, *iterasi* atau perbaikan terus-menerus itu hal biasa. Kamu akan terus belajar, memperbaiki, dan mencoba lagi. Yuk, tantang dirimu sekarang juga! Siapkan presentasimu, latih demo-nya, dan jadi presenter muda yang **percaya diri, komunikatif, dan inovatif**!

Latihan Soal

A. Benar atau Salah (5 Soal)

1. Presentasi yang baik tidak membutuhkan struktur yang jelas.
   Jawaban: Salah

2. Kontak mata penting saat berbicara di depan umum.
   Jawaban: Benar

3. Demo aplikasi tidak perlu dipersiapkan sebelum presentasi.
   Jawaban: Salah

4. Memberikan feedback harus dilakukan dengan cara yang sopan dan membangun.
   Jawaban: Benar

5. Tujuan presentasi adalah agar orang lain memahami dan tertarik pada aplikasi kita.
   Jawaban: Benar

B. Pilihan Ganda (10 Soal)

6. Apa tujuan utama dari presentasi aplikasi?
   A. Menjual aplikasi ke pasar luar negeri
   B. Menunjukkan hasil kerja dan meyakinkan audiens
   C. Melatih berbicara di depan kamera
   D. Menghindari tanya jawab

Jawaban: B

7. Salah satu cara membuat presentasi menarik adalah...
   A. Membaca semua teks di slide
   B. Menghindari berbicara dengan intonasi
   C. Menggunakan visual yang mendukung
   D. Duduk saat berbicara
   Jawaban: C

8. Apa yang sebaiknya dilakukan saat demo aplikasi?
   A. Menyembunyikan fitur yang belum selesai
   B. Menjelaskan cara kerja aplikasi
   C. Membuat slide sebanyak mungkin
   D. Diam saat aplikasi berjalan
   Jawaban: B

9. Hal berikut ini termasuk dalam elemen struktur presentasi, kecuali...
   A. Pendahuluan
   B. Demo
   C. Kritik tajam
   D. Kesimpulan
   Jawaban: C

10. Apa yang dimaksud dengan feedback konstruktif?
    A. Feedback yang membuat orang kesal
    B. Feedback tanpa solusi
    C. Umpan balik yang membangun dan berguna
    D. Jawaban otomatis dari aplikasi
    Jawaban: C

11. Apa yang sebaiknya dilakukan saat menerima kritik dari audiens?
    A. Langsung membantah
    B. Diam saja
    C. Mendengarkan dengan terbuka
    D. Menyela pembicara
    Jawaban: C

12. Bahasa tubuh yang baik saat presentasi ditunjukkan dengan...
    A. Membelakangi audiens
    B. Menatap layar saja
    C. Berdiri tegak dan gerakan tangan yang wajar
    D. Menghindari gerakan apa pun
    Jawaban: C

13. Apa manfaat dari sesi tanya jawab dalam presentasi?
    A. Untuk menguji audiens
    B. Untuk mengisi waktu
    C. Untuk memberikan pemahaman lebih dan menunjukkan kesiapan

D. Untuk membuat orang bingung
Jawaban: C

14. Mengapa feedback dari teman dan guru penting?
    A. Karena harus
    B. Karena bisa jadi hiburan
    C. Karena membantu memperbaiki aplikasi
    D. Karena membuat kita populer
    Jawaban: C

15. Apa yang dimaksud dengan iterasi dalam pengembangan aplikasi?
    A. Membuat aplikasi dari awal lagi
    B. Proses memperbaiki aplikasi berdasarkan feedback
    C. Menghapus semua fitur aplikasi
    D. Menyimpan aplikasi di komputer
    Jawaban: B

C. Isian Singkat (5 Soal)

16. Kontak mata, intonasi, dan bahasa tubuh adalah bagian dari _____ saat presentasi.
    Jawaban: komunikasi verbal

17. Slide presentasi yang baik sebaiknya berisi _____ dan tidak terlalu penuh tulisan.
    Jawaban: visual pendukung

18. Dalam presentasi, sesi untuk menunjukkan cara kerja aplikasi disebut _____.
    Jawaban: demo

19. Feedback yang fokus pada solusi disebut _____ feedback.
    Jawaban: konstruktif

20. Setelah menerima masukan, kita melakukan _____ untuk memperbaiki aplikasi.
    Jawaban: revisi

D. Soal Explorasi (3 Soal)

Soal:

Kamu baru saja melihat presentasi dari kelompok lain tentang aplikasi mereka. Tulis 3 pertanyaan yang kamu ingin ajukan kepada mereka. Jelaskan mengapa pertanyaan-pertanyaan itu penting dan bagaimana jawabannya bisa membantu mereka meningkatkan aplikasinya.

Soal:

Buatlah rencana presentasi untuk aplikasi kelompokmu. Rencana ini harus memuat: pembukaan, masalah yang diangkat, solusi yang ditawarkan, fitur utama aplikasi, demo, dan kesimpulan. Jelaskan siapa yang akan menyampaikan bagian mana dan bagaimana kalian akan melatih penyampaiannya agar menarik.

Soal:

Setelah presentasi aplikasimu, kamu menerima 2 kritik dari temanmu: "fiturnya kurang jelas" dan "desain warnanya terlalu gelap." Tuliskan bagaimana kamu akan menanggapi dua kritik ini secara positif. Lalu buat daftar perubahan atau revisi yang akan kamu lakukan sebelum presentasi berikutnya.

Tugas Proyek

Proyek 1: Aplikasi Mini – “Ingat Minum!”

Deskripsi:
Buatlah aplikasi sederhana yang mengingatkan pengguna untuk minum air putih secara berkala.

Langkah-langkah:

1. Riset tentang pentingnya minum air dan kebutuhan harian tubuh.
2. Gunakan platform sederhana seperti MIT App Inventor atau Scratch.
3. Tambahkan fitur notifikasi dan visual lucu agar menarik.
4. Siapkan presentasi: Jelaskan masalah dehidrasi, solusi aplikasimu, dan demo aplikasi.
5. Latihan menjawab pertanyaan: Apa yang membuat aplikasi unik dibanding alarm biasa?

Nilai Tantangan:
Gabungan kreativitas desain, teknologi, dan kemampuan berbicara di depan umum.

Proyek 2: Poster Interaktif “Selamatkan Lingkungan!”

Deskripsi:
Desain poster digital yang interaktif tentang masalah lingkungan lokal (sampah, polusi udara, atau air).

Langkah-langkah:

1. Pilih isu lingkungan yang dekat dengan kehidupan sehari-hari.
2. Gunakan Canva atau PowerPoint untuk membuat poster digital dengan link, video, atau tombol interaktif.
3. Buat presentasi seperti cerita: mulai dari masalah, data pendukung, lalu solusi sederhana dari kamu.
4. Ajak teman berdiskusi dan beri sesi tanya jawab.

Nilai Tantangan:
Melatih empati terhadap isu nyata dan menyampaikan pesan dengan media visual yang kuat.

Proyek 3: Rancang Game Edukatif Mini

Deskripsi:
Buat game sederhana untuk membantu siswa lain belajar topik sulit, seperti pecahan atau sejarah.

Langkah-langkah:

1. Tentukan topik yang ingin diajarkan (misal: “Game Tebak Tokoh Pahlawan”).
2. Gunakan Scratch atau aplikasi pembuat kuis seperti Kahoot atau Quizizz.
3. Buat aturan main dan visual menarik.
4. Latih presentasi: jelaskan bagaimana game ini bisa membantu belajar.
5. Lakukan demo langsung dengan teman bermain game buatanmu.

Nilai Tantangan:
Menggabungkan teknologi, kreativitas, dan pemahaman materi pelajaran.

Proyek 4: “Ide Bisnis Mini” untuk Teman Sekolah

Deskripsi:
Simulasikan proyek bisnis kecil, misalnya menjual stiker digital, bookmark, atau jasa bantu tugas ringan.

Langkah-langkah:

1. Tentukan produk/jasa yang akan ditawarkan.
2. Buat presentasi seperti pitch bisnis: masalah, solusi, target pasar, keunikan produk, dan cara kerja.
3. Melatih intonasi dan ekspresi saat berbicara seperti pengusaha muda.
4. Tambahkan visual: gambar produk, tabel harga, atau animasi.
5. Minta teman memberi feedback dan gunakan untuk iterasi ide.

Nilai Tantangan:
Melatih kreativitas, kemampuan menjual ide, dan berpikir strategis.

Proyek 5: Video “How To” – Tutorial Singkat dari Kamu

Deskripsi:
Buat video tutorial 2–3 menit menjelaskan cara melakukan sesuatu, seperti membuat origami, coding dasar, atau trik belajar cepat.

Langkah-langkah:

1. Pilih keahlian sederhana yang kamu kuasai.

2. Rancang skrip dengan struktur presentasi: pembukaan – langkah-langkah – penutup.
3. Rekam video dengan narasi jelas dan visual pendukung.
4. Tampilkan di depan kelas dan minta masukan dari teman.
5. Lakukan revisi dan tampilkan versi perbaikannya (iterasi).

Nilai Tantangan:
Menumbuhkan rasa percaya diri, kemampuan komunikasi, dan kesadaran terhadap kebutuhan audiens.

Proyek 6: Talkshow Mini “Ide Masa Depanku”

Deskripsi:
Buat simulasi talkshow sederhana bersama 1–2 teman, membahas ide masa depan seperti teknologi, profesi impian, atau perubahan dunia sekolah.

Langkah-langkah:

1. Tentukan tema dan susun alur pertanyaan.
2. Buat naskah ringan namun berbobot.
3. Latihan komunikasi verbal dan nonverbal: intonasi, ekspresi wajah, kontak mata.
4. Rekam atau tampilkan langsung di kelas.
5. Minta feedback dari guru dan teman.

Nilai Tantangan:
Meningkatkan kerja sama, public speaking, dan refleksi terhadap masa depan.

BAB 8: Menulis Deskripsi Aplikasi dengan Bantuan AI

Tujuan Pembelajaran

Siswa dapat **menulis deskripsi aplikasi yang menarik dan persuasif** dengan bantuan **AI**, serta **menyempurnakannya dengan kreativitas sendiri**.

Peta Konsep



Apersepsi

“Pernah nggak kamu unduh aplikasi gara-gara deskripsinya keren?”
“Menurutmu, kenapa deskripsi itu penting buat menarik perhatian orang di Play Store atau App Store?”

Penjelasan Konsep (Teori)

Pentingnya Deskripsi Aplikasi yang Menarik

Saat kamu membuka Play Store atau App Store, apa hal pertama yang kamu lihat dari sebuah aplikasi? Selain ikon dan rating bintang, tentu **deskripsi aplikasi** jadi penentu penting. Deskripsi ini seperti "salesman digital" yang bertugas menjelaskan mengapa aplikasi tersebut layak diunduh.

Tanpa deskripsi yang menarik, aplikasi bisa saja terlewatkan oleh calon pengguna, meskipun sebenarnya bagus. Karena itulah, belajar menulis deskripsi aplikasi dengan baik itu **penting banget**, apalagi jika kamu bercita-cita membuat aplikasi sendiri suatu hari nanti.

Fungsi Deskripsi Aplikasi:

- Menjelaskan **apa fungsi utama aplikasi** secara singkat dan jelas.
- Menarik perhatian agar orang ingin mencoba (unduh) aplikasi.
- Menampilkan **keunggulan aplikasi** dibandingkan yang lain.
  Memberikan informasi penting seperti fitur, manfaat, dan cara penggunaan.

Elemen-Elemen Deskripsi Aplikasi yang Baik:

1. **Judul Menarik**
   - Gunakan kata-kata yang bikin penasaran atau langsung to the point.
     Contoh: *Kalkulator Pintar: Cepat, Ringan, Akurat!*

2. **Paragraf Pembuka yang Menggoda**
   - Cukup 1-2 kalimat, tapi harus membuat orang ingin lanjut membaca.

3. **Daftar Fitur Utama**
   - Gunakan bullet point untuk mempermudah pembaca.
     Contoh:
     - Hitung cepat dalam hitungan detik
     - Desain simpel dan elegan
     - Bisa dipakai offline

4. **Manfaat untuk Pengguna**
   - Jelaskan keuntungan nyata yang akan mereka rasakan.

5. **Call to Action (Ajakan untuk Bertindak)**

    - Kalimat penutup yang mendorong pengguna untuk mengunduh, misalnya:
      Yuk, coba sekarang dan rasakan kemudahannya!

Apa Itu App Store Optimization (ASO)?

Sama seperti SEO (Search Engine Optimization) yang membuat website lebih mudah ditemukan di Google, **ASO** atau *App Store Optimization* adalah teknik untuk membuat aplikasi kita lebih mudah ditemukan di toko aplikasi.

Beberapa trik ASO:

- Gunakan **kata kunci (keywords)** yang sering dicari orang.
- Tampilkan **rating tinggi dan review positif**.
- Pakai **ikon yang menarik** dan deskripsi yang optimal.

Dengan ASO yang tepat, peluang aplikasi kamu diunduh jadi **lebih besar!**

Memanfaatkan Kecerdasan Buatan (AI) untuk Menyusun Deskripsi

Menulis deskripsi bukan hal yang mudah. Tapi jangan khawatir! Sekarang kita bisa minta bantuan **AI seperti ChatGPT atau Gemini** untuk menyusun draf deskripsi. Ini seperti punya asisten pribadi yang super pintar dan cepat mengetik.

Langkah-Langkah Menggunakan AI:

1. **Berikan Prompt yang Jelas**
    - Jelaskan jenis aplikasi, siapa penggunanya, dan fitur utama.
      Contoh prompt:
      Buatkan deskripsi menarik untuk aplikasi kalkulator sederhana yang cepat, ramah pengguna, dan bisa digunakan offline.

2. **Minta Saran Kata Kunci (Keywords)**
    - Supaya deskripsimu mudah ditemukan di toko aplikasi.

3. **Tinjau Hasil AI**
    - Lihat apakah draf dari AI sudah sesuai dengan keinginanmu. Kalau belum, bisa disesuaikan.

Menyempurnakan Deskripsi dengan Sentuhan Manusia

AI memang hebat, tapi tetap saja **manusia punya rasa dan gaya yang unik**. Setelah AI membuat draft, tugas kita adalah menyempurnakan hasilnya agar terasa lebih hidup dan menarik.

Sentuhan Akhir yang Perlu Dilakukan:

- **Edit gaya bahasa** agar terdengar lebih alami dan tidak kaku.
- **Periksa fakta**: pastikan semua informasi benar.

- **Tambahkan keunikan aplikasi**: apa yang membuatnya berbeda dari yang lain?
- **Gunakan bahasa yang sesuai target pengguna**: anak-anak, remaja, orang tua, dll.

Kesimpulan

Menulis deskripsi aplikasi itu seperti membuat iklan kecil—pendek, padat, tapi harus memikat. Dengan bantuan AI, prosesnya jadi lebih cepat, tapi kamu tetap perlu mengolahnya agar sesuai gaya kamu sendiri. Ini kombinasi sempurna antara teknologi dan kreativitas.

Jika kamu pernah berpikir membuat aplikasi sendiri, yuk mulai dari belajarnya dulu: **tulis deskripsinya seperti seorang profesional!**

Ilustrasi

Bayangkan kamu membuat aplikasi sederhana bernama **BelajarKu** — aplikasi pengingat belajar harian. Kamu sudah bekerja keras membangun aplikasinya, tapi... saat kamu unggah ke Play Store, ternyata **tidak banyak yang mengunduh**.

Kamu heran: "Padahal aplikasinya berguna banget!" Tapi ternyata... **deskripsinya membosankan dan tidak menjelaskan manfaat aplikasi dengan baik**. Di sinilah peran AI dan kreativitas kamu sangat penting.

Kamu mencoba ChatGPT, memberi tahu tentang aplikasi dan fiturnya. Hasilnya? Deskripsi awal langsung jadi! Tapi masih terasa kaku. Setelah kamu edit, tambahkan gaya khas kamu, dan menyisipkan kata-kata keren seperti “ingat belajar tanpa stres!”, unduhan pun mulai naik!

Contoh Kasus dan Ilustrasi

Kasus 1: Dua Aplikasi, Satu Tujuan — Mana yang Kamu Pilih?

Ilustrasi:
Bayangkan kamu sedang mencari aplikasi kalkulator di Play Store. Ada dua pilihan:

- **KalkuLite:** "Aplikasi kalkulator sederhana. Bisa hitung tambah, kurang, kali, dan bagi."
- **SmartKalk Pro:** "Hitung cepat tanpa ribet! Kalkulator pintar dengan desain elegan, hasil akurat secepat kilat!"

Tantangan untuk dipikirkan:
Menurut kalian, mana yang lebih menarik? Apakah kalian pernah memilih aplikasi hanya karena deskripsinya terdengar keren dan meyakinkan?

Kasus 2: Aplikasi Anti Malas — Yuk Bikin Deskripsinya!

Ilustrasi:
Sebuah aplikasi dibuat khusus untuk remaja yang suka menunda-nunda belajar. Namanya “ByeBye Malas!”. Fungsinya: mengatur jadwal belajar, kasih alarm motivasi, dan ada suara penyemangat dari karakter animasi.

Pertanyaan pemantik:
Kalau kalian jadi pembuatnya, gimana caranya bikin orang lain tertarik untuk download? Kata-kata seperti apa yang harus muncul di deskripsinya?

Kasus 3: Adu Hebat antara Manusia dan AI

Ilustrasi:
Bayangkan kamu dan temanmu sama-sama membuat deskripsi untuk aplikasi resep masakan. Tapi temanmu dibantu AI seperti ChatGPT. Kalian sama-sama menulis, lalu membandingkan hasilnya.
Deskripsi mana yang lebih menarik? Apakah AI bisa mengalahkan kreativitas manusia?

Diskusi seru:
Kapan kita bisa mengandalkan AI, dan kapan perlu sentuhan manusia?

Kasus 4: ASO — Senjata Rahasia Developer!

Ilustrasi:
Seorang developer muda gagal mendapatkan banyak unduhan karena judul aplikasinya terlalu umum: “Aplikasi Catatan Harian”. Tapi saat ia mengganti menjadi “DiaryKu – Catatan Harian Estetik & Rahasia”, unduhan langsung meningkat.

Pertanyaan menantang:
Kenapa judul dan deskripsi bisa begitu berpengaruh? Apa rahasianya? Yuk, kita kenalan dengan teknik App Store Optimization!

Kasus 5: Aplikasi yang Salah Deskripsi, Salah Target

Ilustrasi:
Sebuah aplikasi belajar matematika SD malah pakai deskripsi yang terlalu teknis, seperti: “Menampilkan parameter integral dan statistik kuantitatif”. Hasilnya? Anak-anak bingung, dan tidak ada yang mau download.

Refleksi:
Bagaimana cara menyusun deskripsi yang sesuai dengan target pengguna? Kalau audiensnya anak SD, remaja, atau orang tua, gaya bahasanya harus beda, kan?

Kasus 6: Editan Manusia vs Draf Mentah AI

Ilustrasi:
AI berhasil membuat deskripsi awal untuk aplikasi “SleepBuddy” — alat bantu tidur untuk remaja. Tapi hasilnya agak kaku. Misalnya:
“SleepBuddy adalah solusi tidur remaja. Dilengkapi dengan musik relaksasi dan pelacak pola tidur.”

Setelah diedit manusia, hasilnya jadi:
“Sulit tidur? SleepBuddy siap jadi teman malam kamu! Dengan musik relaksasi dan pelacak tidur pintar, tidur nyenyak bukan lagi mimpi.”

Diskusi eksploratif:
Apa bedanya? Bagaimana editan manusia bisa menambahkan “perasaan” dan gaya yang lebih hidup?

Contoh Soal dan Pembahasan

Soal 1: "Kenapa Harus Tertarik?"

Pertanyaan:
Bayangkan kamu menemukan dua aplikasi kalkulator di Play Store. Yang satu punya deskripsi seperti ini:

> "Kalkulator standar. Bisa tambah, kurang, kali, bagi."

Yang satunya lagi seperti ini:

> "Kalkulator Pintar – Cepat, Akurat, dan Mudah Digunakan! Cocok untuk pelajar, lengkap dengan fitur hitung akar dan persen!"

Pertanyaannya:
Menurutmu, mengapa deskripsi kedua lebih menarik? Sebutkan **2 alasan** yang membuat deskripsi aplikasi penting dalam menarik perhatian pengguna!

Pembahasan:
Deskripsi kedua lebih menarik karena:

1. **Bahasa yang menggoda** – menggunakan kata-kata seperti “pintar”, “cepat”, dan “mudah” membuat pengguna penasaran.
2. **Menunjukkan manfaat** – menyebutkan “cocok untuk pelajar” dan fitur seperti “akar dan persen” langsung menjelaskan kelebihan aplikasinya.

Soal 2: "Bongkar Isi Deskripsi!"

Pertanyaan:
Sebuah deskripsi aplikasi berbunyi:

> “Belajar Bahasa Inggris jadi lebih seru! Gunakan aplikasi ini untuk latihan listening, speaking, dan grammar. Dapatkan skor dan naik level!”

Coba identifikasi **3 elemen penting** dari deskripsi tersebut berdasarkan teori menulis deskripsi aplikasi!

Pembahasan:
Elemen penting yang bisa ditemukan:

1. **Deskripsi singkat yang menggoda** – “jadi lebih seru!”
2. **Fitur utama** – latihan listening, speaking, dan grammar.
3. **Call to action tersembunyi** – “Dapatkan skor dan naik level” mendorong pengguna untuk mencoba.

Soal 3: "Si Jenius AI"

Pertanyaan:
Kamu ingin membuat aplikasi belajar matematika bernama **"Math Magic"**. Tugasmu adalah **memberikan prompt** ke AI seperti ChatGPT agar ia bisa membantu menulis deskripsi aplikasimu.
Apa saja informasi minimal yang harus kamu berikan ke AI supaya hasilnya bagus?

Pembahasan:
Agar AI bisa membantu secara maksimal, kamu harus memberikan:

1. **Jenis aplikasi** – misalnya, aplikasi belajar matematika interaktif.
2. **Target pengguna** – seperti: siswa SMP.
3. **Fitur utama** – latihan soal, video penjelasan, game edukasi.
4. **Manfaat** – bantu memahami konsep matematika dengan cara menyenangkan.

Soal 4: "Uji Keunggulan!"

Pertanyaan:
Kamu sudah membuat aplikasi bernama **“Daily Habits”** untuk membentuk kebiasaan baik. Minta bantuan AI untuk menuliskan **3 keunggulan utama (unique selling points)** dari aplikasi ini. Lalu, dari hasil tersebut, pilih satu dan ubahlah gaya bahasanya agar lebih manusiawi dan persuasif!

Pembahasan:
Misalnya AI menghasilkan:

- Pengingat harian otomatis
- Statistik perkembangan mingguan
- Antarmuka sederhana dan ramah pengguna

Contoh penyempurnaan:

AI: “Pengingat harian otomatis”
Kamu ubah: “Tidak perlu takut lupa! Aplikasi ini akan mengingatkanmu setiap hari untuk tetap konsisten.”

Di sini kamu menambahkan **emosi dan kedekatan**, hasil gabungan AI dan sentuhan manusia.

Soal 5: "Tantangan Kata Kunci!"

Pertanyaan:
Aplikasi bernama **"FotoKilau"** adalah editor foto sederhana untuk remaja.
Tantangannya: **Mintalah AI untuk memberikan 5 kata kunci (keywords)** yang cocok agar aplikasi ini mudah ditemukan di Play Store. Setelah itu, pilih 2 kata kunci dan jelaskan mengapa mereka cocok.

Pembahasan:
Kata kunci yang mungkin diberikan AI:

- edit foto
- filter aesthetic
- editor pemula
- efek keren
- aplikasi remaja

Dua kata kunci yang cocok:

1. **filter aesthetic** – banyak dicari oleh remaja yang suka foto stylish.
2. **editor pemula** – menggambarkan bahwa aplikasi ini mudah digunakan.

Soal 6: "Sentuhan Akhir!"

Pertanyaan:
Kamu sudah mendapatkan draf deskripsi aplikasi dari AI. Namun, rasanya terlalu kaku dan seperti robot.
Apa **3 hal** yang bisa kamu lakukan agar deskripsi ini terasa lebih alami dan menarik untuk dibaca oleh manusia?

Pembahasan:
Langkah-langkah yang bisa kamu lakukan:

1. **Gunakan bahasa yang akrab** – bukan sekadar formal, tapi komunikatif.
2. **Tambahkan gaya khas atau humor ringan** – agar terasa lebih personal.
3. **Periksa akurasi dan kesesuaian** – jangan sampai AI menambahkan fitur yang tidak ada!

Fun Facts

- **Lebih dari 80%** pengguna memutuskan mengunduh aplikasi hanya karena **tertarik pada deskripsinya!**
- **“Unduh sekarang”** adalah salah satu kalimat **call to action** paling sering dipakai di seluruh dunia.
- Nama aplikasi yang mengandung kata **“cepat”, “mudah”,** atau **“gratis”** cenderung lebih banyak di klik.
- Beberapa perusahaan besar juga **menggunakan AI** untuk menulis deskripsi aplikasi mereka!

Tip Belajar

Mind Mapping

Gunakan mind mapping untuk mengingat struktur deskripsi aplikasi:



Contoh Cerita Pengingat

Bayangkan kamu adalah seorang "pemandu aplikasi" yang harus menarik pengunjung di taman teknologi. Setiap kali kamu menjelaskan aplikasi, kamu harus bisa membuat mereka berkata: “Wah, keren banget!”

Aktivitas Siswa

Aktivitas 1: Diskusi Seru: Aplikasi Favorit dan Deskripsinya!

Tujuan: Menumbuhkan kesadaran akan pentingnya deskripsi aplikasi yang menarik.
Langkah-langkah:

- Ajak siswa membuka Play Store atau App Store di perangkat masing-masing (atau lihat di layar guru).
- Minta mereka memilih satu aplikasi favorit dan membaca deskripsinya.
- Ajak berdiskusi: Apa yang membuat mereka tertarik? Kata-kata apa yang menurut mereka paling menggoda?
- Guru bisa menampilkan dua deskripsi aplikasi (satu menarik, satu membosankan) dan mengajak siswa membandingkan secara interaktif.

Aktivitas 2: Jelajahi Unsur Deskripsi Aplikasi yang Keren

Tujuan: Mengenal elemen-elemen penting dalam deskripsi aplikasi.
Langkah-langkah:

- Guru memberi contoh nyata deskripsi aplikasi dan menyorot unsur seperti: judul, deskripsi singkat, fitur, manfaat, dan ajakan.
- Siswa bekerja berpasangan untuk menandai elemen-elemen tersebut pada deskripsi aplikasi lain yang diberikan.
- Tantangan: Temukan satu unsur yang paling menonjol dan satu yang perlu ditingkatkan, lalu sampaikan alasannya di depan kelas.

Aktivitas 3: Ayo Berlatih Meminta Bantuan AI!

Tujuan: Mengenalkan cara memberi prompt yang efektif kepada AI untuk membuat draf deskripsi.
Langkah-langkah:

- Guru memberi demonstrasi cara meminta bantuan ChatGPT atau Gemini (misalnya: “Tolong buatkan deskripsi aplikasi kalkulator cepat dan ramah pengguna untuk anak sekolah.”).
- Siswa mencoba membuat prompt mereka sendiri (misalnya untuk aplikasi karya mereka di proyek sebelumnya).
- Tunjukkan hasil respon AI, dan diskusikan apakah sudah sesuai atau perlu diperbaiki.

Aktivitas 4: Berburu Kata Kunci: Misi Rahasia dari AI

Tujuan: Mengenal pentingnya kata kunci (keywords) dalam deskripsi.
Langkah-langkah:

- Guru menjelaskan bahwa kata kunci membuat aplikasi mudah ditemukan (ASO sederhana).
- Siswa memberi prompt kepada AI: “Apa saja kata kunci penting untuk aplikasi belajar matematika untuk SMP?”
- Tantangan: Siswa memilih 3 kata kunci terbaik dan menuliskannya dalam kalimat promosi yang menarik!

Aktivitas 5: Tantangan Kreatif: Deskripsi AI + Sentuhan Manusia

Tujuan: Melatih kemampuan revisi dan menambahkan gaya personal dalam teks.
Langkah-langkah:

- Gunakan deskripsi yang sudah dibuat AI pada aktivitas sebelumnya.
- Siswa diminta menyuntingnya agar lebih manusiawi, lucu, atau khas gaya anak muda, tanpa menghilangkan informasi penting.
- Bisa dilakukan berpasangan agar saling memberi masukan.
- Hasil revisi dibacakan dan dipilih versi paling menarik oleh teman-teman.

Aktivitas 6: Battle Deskripsi Aplikasi!

Tujuan: Menguji kreativitas dalam membuat deskripsi yang menjual.
Langkah-langkah:

- Siswa dibagi kelompok dan diberi “aplikasi imajiner” (misal: Aplikasi jam alarm yang bisa bernyanyi!).
- Setiap kelompok membuat deskripsi versi AI dan versi revisi manusia.
- Presentasi dan voting kelas: Mana deskripsi yang paling bikin ingin mendownload?

Aktivitas ini dirancang untuk mengajak siswa berpikir kritis, berani bereksperimen dengan teknologi, serta menyadari bahwa menulis bisa menjadi kegiatan kreatif dan menyenangkan. Jika ingin saya bantu siapkan lembar kerja atau panduan aktivitasnya, saya siap bantu!

Rangkuman

Pernah nggak sih kamu mengunduh aplikasi hanya karena baca deskripsinya yang keren banget? Nah, di dunia digital seperti sekarang, **deskripsi aplikasi yang menarik itu sangat penting**! Di *Play Store* atau *App Store*, orang akan membaca deskripsi sebelum memutuskan mengunduh. Jadi, kamu perlu tahu bagaimana menyusun deskripsi yang bikin orang penasaran. Mulai dari **judul yang catchy**, **penjelasan singkat yang menggoda**, hingga **fitur-fitur utama** dan manfaatnya bagi pengguna. Di sini kamu juga akan kenalan dengan *App Store Optimization (ASO)*—cara supaya aplikasi kamu gampang ditemukan orang.

Tenang, kamu nggak perlu jago nulis dulu untuk bikin deskripsi yang keren. Sekarang kamu bisa pakai **AI seperti *ChatGPT* atau *Gemini*** buat bantu menyusun draf awal! Kamu tinggal

kasih *prompt* yang jelas—misalnya jenis aplikasinya apa, siapa yang bakal pakai, dan fitur utamanya. AI akan bantu menyusun deskripsi dan bahkan kasih **kata kunci penting** (*keywords*) supaya aplikasimu lebih mudah ditemukan. Seru, kan? Kamu bisa coba minta AI bikin deskripsi aplikasi kalkulator yang *cepat, akurat, dan ramah pengguna*, dan lihat bedanya sama deskripsi yang biasa-biasa aja.

Tapi jangan lupa, **AI itu cuma alat bantu**. Supaya deskripsinya lebih *manusiawi* dan menggugah, kamu tetap perlu kasih **sentuhan pribadi**. Di sinilah kamu belajar bagaimana menyempurnakan draf dari AI: mengedit, menyesuaikan gaya bahasa, dan menambahkan **keunikan aplikasi kamu**. Kamu juga bisa minta AI bantu menulis 3 *unique selling points* dari aplikasi yang kamu buat di proyek sebelumnya. Tantangannya, bagaimana kamu bisa bikin deskripsi yang nggak cuma informatif, tapi juga penuh *personality*? Yuk, kita eksplorasi bersama!

Latihan Soal

A. Benar atau Salah (5 Soal)

1. Deskripsi aplikasi yang baik bisa menarik lebih banyak pengguna.
   Jawaban: Benar

2. AI dapat digunakan untuk menulis deskripsi aplikasi secara otomatis.
   Jawaban: Benar

3. Deskripsi aplikasi sebaiknya panjang dan penuh istilah teknis.
   Jawaban: Salah

4. App Store Optimization (ASO) membantu aplikasi kita mudah ditemukan di toko aplikasi.
   Jawaban: Benar

5. Hasil tulisan AI tidak perlu diedit lagi oleh manusia.
   Jawaban: Salah

B. Pilihan Ganda (10 Soal)

6. Apa fungsi utama dari deskripsi aplikasi?
   A. Membuat aplikasi lebih mahal
   B. Menarik perhatian dan menjelaskan manfaat aplikasi
   C. Menyembunyikan fitur yang belum jadi
   D. Menghindari ulasan negatif
   Jawaban: B

7. Salah satu elemen penting dalam deskripsi aplikasi adalah...
   A. Riwayat hidup pembuat aplikasi
   B. Daftar makanan favorit pengguna

C. Daftar fitur utama aplikasi
D. Waktu pembuatan aplikasi
Jawaban: C

8. Apa yang dimaksud dengan "call to action"?
A. Panggilan darurat
B. Ajak pengguna untuk bertindak, seperti "Unduh sekarang!"
C. Nama kontak pengguna
D. Alarm aplikasi
Jawaban: B

9. Mengapa kita perlu menyempurnakan deskripsi yang dibuat AI?
A. Karena AI tidak boleh digunakan
B. Supaya lebih cepat selesai
C. Agar terdengar lebih alami dan sesuai kebutuhan
D. Agar terlihat lebih keren
Jawaban: C

10. Keyword atau kata kunci digunakan dalam deskripsi untuk...
A. Menyembunyikan fitur
B. Membuat deskripsi panjang
C. Membantu aplikasi muncul dalam pencarian
D. Menghindari komentar
Jawaban: C

11. Judul aplikasi yang menarik biasanya...
A. Membingungkan
B. Mengandung nama hewan
C. Mudah diingat dan menggambarkan fungsi
D. Berisi angka saja
Jawaban: C

12. Apa itu ASO (App Store Optimization)?
A. Sistem keamanan aplikasi
B. Cara agar aplikasi mudah ditemukan di toko aplikasi
C. Format gambar aplikasi
D. Waktu update aplikasi
Jawaban: B

13. Saat memberikan prompt ke AI untuk menulis deskripsi, kita sebaiknya menyebutkan...
A. Nama guru
B. Target pengguna dan fitur utama
C. Judul film
D. Ukuran aplikasi
Jawaban: B

14. Contoh kalimat "call to action" yang tepat adalah...
    A. “Silakan abaikan aplikasi ini”
    B. “Klik di sini untuk mencobanya sekarang!”
    C. “Saya sedang bingung”
    D. “Tunggu dulu sampai besok”
    Jawaban: B

15. Saat mengedit hasil dari AI, kita perlu memastikan bahwa...
    A. Semua kata asing dihapus
    B. Teks dibuat dalam bentuk puisi
    C. Informasi akurat dan mudah dipahami
    D. Nama aplikasi diganti
    Jawaban: C

C. Isian Singkat (5 Soal)

16. _____ membantu aplikasi muncul dalam hasil pencarian di toko aplikasi.
    Jawaban: Kata kunci

17. Deskripsi yang menarik harus menyebutkan _____ yang dimiliki aplikasi.
    Jawaban: fitur utama

18. Kalimat seperti "Unduh sekarang dan rasakan bedanya!" disebut _____.
    Jawaban: call to action

19. Untuk membuat draf deskripsi, kita bisa menggunakan bantuan _____.
    Jawaban: AI (seperti ChatGPT)

20. Revisi manusia penting untuk memastikan deskripsi terdengar _____.
    Jawaban: alami

D. Soal Eksplorasi (3 Soal)

Soal:

Bayangkan kamu membuat aplikasi pengingat belajar untuk siswa SMP. Gunakan AI (misalnya ChatGPT) untuk membuat deskripsi awal aplikasi tersebut. Tulis prompt yang kamu berikan, hasil dari AI, dan kemudian tulis versi editan kamu sendiri. Jelaskan apa yang kamu ubah dan mengapa kamu merasa versi kamu lebih baik.

Soal:

Pilih satu aplikasi yang kamu sukai dari Play Store atau App Store. Salin deskripsi singkatnya, lalu analisis bagian-bagian berikut: judul, pembuka, fitur utama, manfaat, dan call to action. Apa yang membuat deskripsi itu menarik? Tuliskan dua kata kunci yang menurutmu penting dari deskripsi tersebut.

Soal:

Kamu dan timmu membuat aplikasi sederhana untuk membantu orang memilih menu sehat. Sekarang, kalian harus menulis deskripsi yang menarik untuk Play Store. Buatlah:

- Judul aplikasi
- Deskripsi singkat
- Tiga fitur utama
- Dua manfaat utama
- Satu kalimat call to action

Tulis versi awal (boleh dari AI), lalu tulis versi hasil revisi kelompok kalian. Jelaskan perbedaan gaya bahasa atau isi antara kedua versi.

Tugas Proyek

Proyek 1: Bandingkan dan Nilai Deskripsi Aplikasi

Tujuan: Melatih siswa mengenali elemen deskripsi aplikasi yang menarik.

Langkah-langkah:

1. Carilah dua deskripsi aplikasi yang mirip (misalnya, dua aplikasi kalkulator) di Play Store atau App Store.
2. Bacalah kedua deskripsi dan identifikasi:
    - Judul aplikasi
    - Fitur-fitur utama
    - Gaya bahasa
    - Call to action
3. Buat tabel perbandingan dan berikan penilaian dari 1–5 pada masing-masing elemen.
4. Diskusikan: Mengapa salah satu lebih menarik daripada yang lain?

Bonus Tantangan: Presentasikan hasil analisismu dalam format infografis.

Proyek 2: Buat Deskripsi Aplikasi Impianmu dengan Bantuan AI

Tujuan: Melatih siswa menulis deskripsi aplikasi secara kreatif dan memanfaatkan AI.

Langkah-langkah:

1. Bayangkan dan gambarkan ide aplikasi yang kamu ciptakan.
2. Gunakan AI seperti ChatGPT atau Gemini, lalu beri perintah seperti:
    - “Tolong buatkan deskripsi untuk aplikasi belajar matematika untuk siswa SMP yang seru dan menyenangkan.”
3. Salin hasilnya dan analisis bersama teman kelompokmu: apakah sudah menarik?

4. Lakukan editing dan revisi untuk membuatnya lebih personal dan meyakinkan.

Proyek 3: Tantangan "3 Keunggulan Aplikasiku!"

Tujuan: Mengidentifikasi *unique selling points* aplikasi.

Langkah-langkah:

1. Ambil ide aplikasi dari tugas atau proyek sebelumnya.
2. Minta AI menyebutkan 3 keunggulan utama aplikasi tersebut.
3. Diskusikan bersama: Apakah ketiga poin tersebut sudah mencerminkan kelebihan aplikasi kalian?
4. Ubah atau tambahkan poin jika diperlukan, lalu susun ulang agar lebih persuasif.

Catatan: Gunakan kata-kata yang singkat, menarik, dan mudah diingat.

Proyek 4: Uji Kreativitas—Revisi Deskripsi Biasa Jadi Luar Biasa

Tujuan: Melatih kemampuan menulis ulang dan meningkatkan gaya bahasa.

Langkah-langkah:

1. Ambil contoh deskripsi aplikasi yang dianggap “membosankan”.
2. Analisis bagian mana yang kurang menarik.
3. Gunakan AI untuk memberi saran perbaikan.
4. Gabungkan saran dari AI dan ide kalian sendiri untuk membuat versi baru yang lebih menarik dan komunikatif.

Tantangan Tambahan: Buat versi formal dan versi lucu, lalu minta teman memilih mana yang lebih efektif.

Proyek 5: Simulasi Tim ASO—Optimalkan Deskripsimu!

Tujuan: Memperkenalkan dasar-dasar *App Store Optimization*.

Langkah-langkah:

1. Pilih satu aplikasi (buatan sendiri atau dari toko aplikasi).
2. Teliti kata kunci (keywords) yang relevan menggunakan AI atau Google Trends.
3. Buat ulang deskripsi aplikasi menggunakan kata-kata kunci yang kamu temukan.
4. Jelaskan mengapa kata-kata tersebut dipilih dan bagaimana bisa membantu aplikasi ditemukan lebih mudah.

Bonus: Uji hasil deskripsi dengan teman—apakah mereka tertarik mengunduhnya?

Proyek 6: Kolaborasi Deskripsi Tim Developer & Copywriter

Tujuan: Meningkatkan kerja sama dan kreativitas dalam tim.

Langkah-langkah:

1. Bentuk kelompok kecil: 1 orang sebagai “pengembang aplikasi”, 1 sebagai “copywriter”, 1 sebagai “editor”.
2. Buat skenario bahwa kalian akan meluncurkan aplikasi baru.
3. Siapkan data aplikasi: fungsi, target pengguna, fitur utama.
4. Gunakan AI sebagai asisten menulis.
5. Tulis, edit, dan sempurnakan deskripsi secara tim, lalu presentasikan ke kelas.

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:

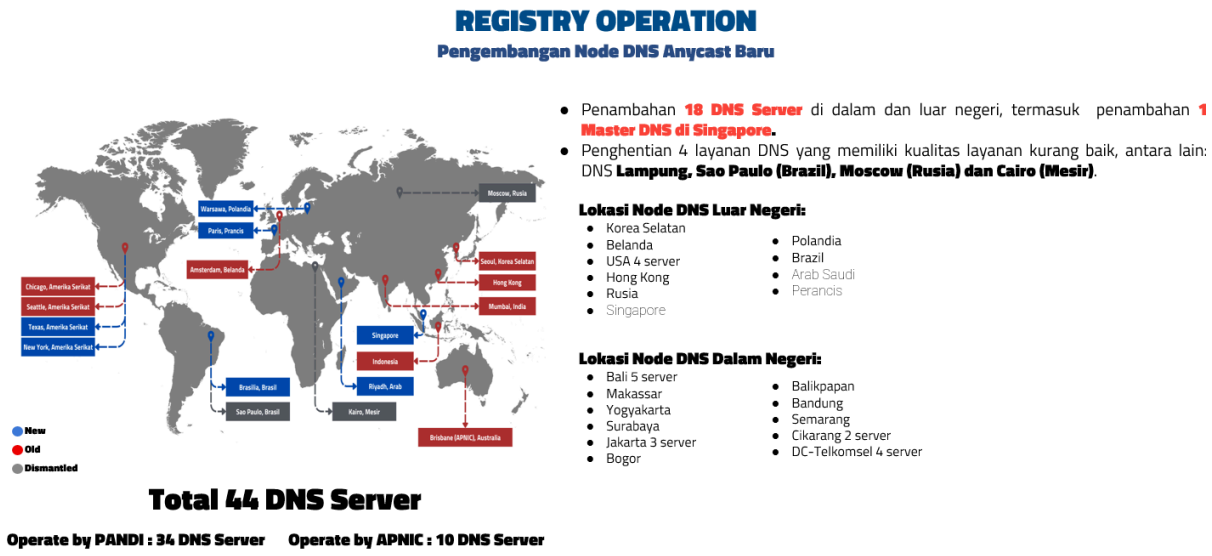
- .jp → Jepang
- .us → Amerika Serikat
- .id → Indonesia 🇮🇩

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.

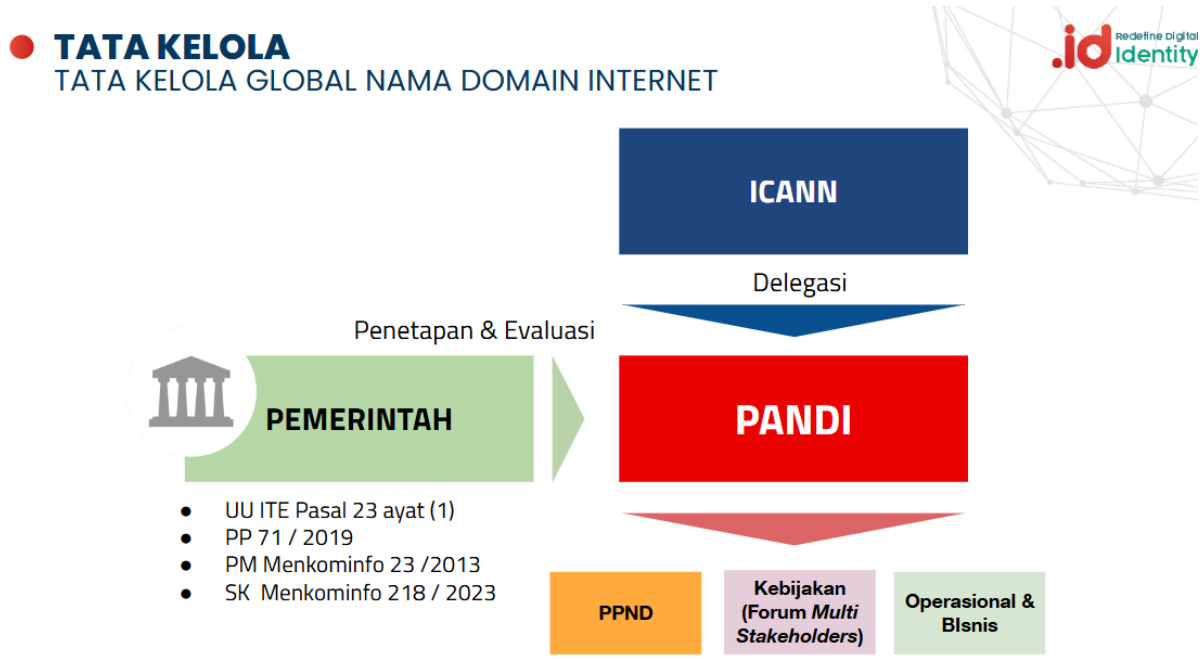
Tugas dan Kewajiban PANDI

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