



Hari 1 - Pengenalan Web Development



Tujuan Pembelajaran

- Memahami konsep dasar Web Development secara menyeluruh, termasuk sejarah dan evolusinya
 - Mengetahui perbedaan serta peran Frontend, Backend, dan Database secara mendalam
 - Memahami alur komunikasi antara client dan server (request-response) beserta contoh nyata
 - Mengenal tools, framework, dan workflow modern dalam pengembangan web
-

1. Apa itu Web Development?

Web Development adalah proses membangun dan mengembangkan aplikasi berbasis web yang dapat diakses melalui internet atau intranet. Web development tidak hanya soal membuat website statis, tapi juga aplikasi interaktif seperti e-commerce, media sosial, dashboard, dsb.

Sejarah Singkat

- **1990-an:** Website statis (HTML saja)
- **2000-an:** Muncul CSS & JavaScript, website mulai interaktif
- **2010-an:** Framework modern (React, Angular, Vue), API, SPA
- **Sekarang:** Progressive Web App, SSR, dll.

Tiga Pilar Utama:

- **Frontend (Client-side):** Bagian yang dilihat dan digunakan oleh pengguna. Fokus pada UI/UX, responsivitas, dan interaksi.
- **Backend (Server-side):** Bagian yang memproses logika aplikasi, autentikasi, otorisasi, dan komunikasi dengan database. Menyediakan API untuk frontend.
- **Database:** Tempat penyimpanan data seperti informasi pengguna, produk, transaksi, dsb. Bisa relasional (PostgreSQL, MySQL) atau non-relasional (MongoDB, Redis).

Analogi:

Bayangkan aplikasi web seperti restoran:

- **Frontend** = Ruang makan & menu (apa yang dilihat pelanggan)
 - **Backend** = Dapur & pelayan (memproses pesanan)
 - **Database** = Gudang bahan makanan (tempat data disimpan)
-

2. Perbedaan Frontend vs Backend vs Database

Aspek	Frontend	Backend	Database
Peran	Tampilan & interaksi user	Logika aplikasi & API	Menyimpan & mengelola data

Aspek	Frontend	Backend	Database
Bahasa Umum	HTML, CSS, JavaScript	JavaScript (Node.js), Python, Go, Java	SQL, NoSQL
Contoh Tools	React.js, Tailwind CSS, Next.js, Vue.js	Express.js, JWT, Bcrypt, REST, GraphQL	PostgreSQL, MySQL, MongoDB, Redis
Akses	Browser	Server	Server
Testing	Jest, Cypress, Playwright	Mocha, Supertest, Postman	pgAdmin, DBeaver
Deployment	Vercel, Netlify	Heroku, Railway, Render	Supabase, AWS RDS

Studi Kasus Sederhana

Misal kamu ingin membangun aplikasi "Catatan Harian":

- **Frontend:** Form input catatan, daftar catatan, tombol hapus/edit
- **Backend:** API untuk simpan, ambil, hapus, edit catatan
- **Database:** Tabel/collection untuk menyimpan data catatan

3. Contoh Alur Request-Response

Ilustrasi alur komunikasi sederhana saat user login:

1. **User** mengetikkan email & password di halaman login (Frontend)
2. Data dikirim ke **Server (Backend)** via HTTP Request (POST)
3. Server memverifikasi ke **Database**, apakah data user valid
4. Jika valid, server mengirim **response** (token atau data user)
5. Frontend menerima response dan menampilkan dashboard ke user

Diagram Sederhana:



Penjelasan Lebih Lanjut

- **Request:** Permintaan dari client ke server (misal: login, ambil data, simpan data)
- **Response:** Balasan dari server ke client (misal: data user, error message, token)
- **HTTP Method:** GET (ambil data), POST (kirim data), PUT/PATCH (update data), DELETE (hapus data)
- **Status Code:** 200 (OK), 201 (Created), 400 (Bad Request), 401 (Unauthorized), 404 (Not Found), 500 (Server Error)

Studi Kasus: Menambah Catatan

1. User isi form catatan baru (Frontend)
2. Frontend kirim POST request ke endpoint `/api/notes`
3. Backend validasi & simpan ke database
4. Backend kirim response sukses/gagal
5. Frontend update tampilan sesuai response

4. Workflow Modern Web Development

- **Version Control:** Git, GitHub, GitLab
- **Deployment:** CI/CD, Vercel, Netlify, Heroku
- **Testing:** Unit test, integration test, end-to-end test
- **Collaboration:** Pull Request, Code Review, Issue Tracker
- **Best Practice:** DRY, KISS, SOLID, YAGNI

Tugas Mandiri Hari Ini (8 Jam)

- Buat mindmap atau diagram alur request user ke server (gunakan tools seperti Whimsical, Miro, atau kertas)
- Quiz istilah: frontend, backend, database, request, response, HTTP method, status code
- Riset dan catat 3 contoh framework/library dari tiap bagian (FE, BE, DB) beserta keunggulannya
- Buat studi kasus sederhana: Deskripsikan alur request-response untuk fitur "Tambah Catatan" (boleh dalam bentuk diagram atau narasi)
- (Opsional) Coba install Node.js & VSCode di laptop masing-masing, dokumentasikan langkahnya

Tips untuk Santri

- Jangan ragu bertanya jika ada istilah yang belum paham
- Coba eksplorasi tools modern, tapi pahami dulu dasarnya
- Fokus pada konsep, bukan sekadar hapal syntax
- Dokumentasikan setiap proses belajar, ini akan sangat membantu ke depannya