

RESUME PENGEMBANGAN DESAIN WEB

MESI ANANDA PUTRI || 20230140035 || CLASS A

1. HTTP (HyperText Transfer Protocol)

HTTP adalah protokol atau aturan yang menjadi fondasi komunikasi data di World Wide Web (WWW).

- Model: Bekerja dengan model Client-Server. Browser Anda (client) mengirim permintaan (request) ke server web, dan server mengirimkan kembali respons (response) yang berisi data seperti halaman HTML atau gambar.
- Metode Umum:
 - GET: Meminta data dari server (misalnya, saat membuka halaman web).
 - POST: Mengirim data ke server untuk diproses (misalnya, saat mengisi form registrasi).
- Kode Status: Pesan singkat dari server tentang hasil permintaan, contohnya 200 OK (sukses), 404 Not Found (halaman tidak ditemukan), dan 500 Internal Server Error (server bermasalah).

2. HTML (HyperText Markup Language)

HTML adalah bahasa markup standar untuk membuat struktur dan konten halaman web.

- Dasar: Terdiri dari elemen yang ditandai oleh tag, seperti <h1> untuk judul utama, <p> untuk paragraf, dan <a> untuk tautan/link. Setiap elemen memberi tahu browser cara menampilkan konten.
- Form: Digunakan untuk mengumpulkan input dari pengguna.
 - Elemen <form> membungkus semua input.
 - Atribut action menentukan ke mana data akan dikirim, dan method (biasanya POST) menentukan cara pengirimannya.
 - Input utama: <input type="text">, <input type="password">, <input type="submit">, <textarea>, dan <select>.

3. CSS (Cascading Style Sheets)

CSS digunakan untuk mendesain tampilan dan layout halaman web yang dibuat dengan HTML.

- Dasar: Bekerja dengan selektor untuk memilih elemen HTML (berdasarkan tag, class, atau ID) dan menerapkan properti gaya seperti color, font-size, dan background-color.
- Layout: CSS modern mengandalkan dua model utama untuk mengatur tata letak halaman:
 - Flexbox: Ideal untuk mengatur layout dalam satu dimensi (baris atau kolom). Sangat fleksibel untuk perataan dan distribusi ruang antar item.
 - Grid: Dirancang untuk layout dua dimensi (baris dan kolom secara bersamaan), memungkinkan pembuatan tata letak yang kompleks dan responsif dengan lebih mudah.

4. JavaScript (JS)

JavaScript adalah bahasa pemrograman yang membuat halaman web menjadi interaktif dan dinamis.

- Dasar: Meliputi variabel (let, const), tipe data, operator, dan struktur kontrol (seperti if...else dan for).
- Manipulasi DOM (Document Object Model): Kemampuan inti JS untuk mengubah konten HTML dan gaya CSS secara *real-time* tanpa perlu memuat ulang halaman.
- OOP (Object-Oriented Programming): Paradigma untuk mengorganisir kode menggunakan objek. Konsep utamanya adalah class (cetak biru) dan object (hasil jadi dari class), yang memiliki properties (data) dan methods (fungsi).
- Standard Library: Kumpulan objek dan fungsi bawaan yang sangat berguna, seperti Math (untuk operasi matematika), Date (untuk bekerja dengan tanggal), JSON (untuk parsing data), dan metode array canggih (.map(), .filter(), .reduce()).

5. Bootstrap 5.3

Bootstrap adalah framework CSS open-source yang sangat populer untuk mengembangkan situs web yang responsif dan *mobile-first* dengan cepat.

- Fitur Utama:
 - Sistem Grid Responsif: Mudah membuat layout berbasis kolom (.container, .row, .col) yang otomatis menyesuaikan diri di berbagai ukuran layar.

- **Komponen Siap Pakai:** Menyediakan banyak sekali komponen yang sudah didesain, seperti *Navbar*, *Buttons*, *Cards*, *Modals*, dan *Forms*.
- **Utility Classes:** Kelas-kelas praktis untuk mengatur *margin*, *padding*, warna, teks, dan lainnya langsung di HTML (misalnya, `.p-3`, `.mt-2`, `.text-center`).
- **Plugin JavaScript:** Komponen interaktif seperti *Carousel* dan *Dropdown* didukung oleh JavaScript, dan di versi 5 ke atas, tidak lagi bergantung pada jQuery.

6. Studi Kasus: PHP & MySQL dengan Bootstrap

Menggabungkan teknologi ini memungkinkan pembuatan aplikasi web dinamis, seperti sistem Login dan Registrasi.

- **Tujuan:** Membuat sistem di mana pengguna dapat mendaftar, masuk, dan mengakses halaman yang dilindungi.
- **Alur Kerja Umum (berdasarkan tutorial Petani Kode):**
 1. **Database (MySQL):** Membuat tabel `users` untuk menyimpan data seperti `username`, `email`, dan `password`.
 2. **Back-End (PHP):**
 - Menulis skrip untuk terhubung ke database MySQL.
 - Memproses data dari form HTML (`$_POST`).
 - **Logika Registrasi:** Memvalidasi input dan menyimpan pengguna baru ke database. Penting: Password di-hash menggunakan `password_hash()` demi keamanan.
 - **Logika Login:** Mengambil data pengguna dari database dan memverifikasi password menggunakan `password_verify()`.
 - Mengelola sesi (`$_SESSION`) untuk mengingat status login pengguna saat berpindah halaman.
 3. **Front-End (HTML & Bootstrap):**
 - Membuat tampilan form login dan registrasi yang rapi dan responsif menggunakan komponen form dari Bootstrap.
 - Menampilkan pesan (sukses atau error) yang dikirim dari skrip PHP.