

Muhammad Anan Paradigma Alvarais – 20230140049 – Resume PDW

1. HTTP (Hypertext Transfer Protocol)

HTTP adalah protokol untuk komunikasi antara klien (browser) dan server.

- **Model:** Client-Server. Klien meminta (request), server menanggapi (response).
- **Sifat:** *Stateless*. Setiap permintaan adalah independen. Sesi (seperti login) dikelola menggunakan *cookies*.

Metode Request Umum: | Metode | Kegunaan | | :--- | :--- | | **GET** | Meminta atau mengambil data dari server. | | **POST** | Mengirim data ke server untuk membuat sumber daya baru. | | **PUT** | Memperbarui data yang sudah ada di server. | | **DELETE** | Menghapus data di server. |

Kategori Kode Status: | Kode | Makna | Contoh | | :--- | :--- | :--- | | **2xx** | Sukses | 200 OK | | **3xx** | Pengalihan (Redirect) | 301 Moved Permanently | | **4xx** | Kesalahan Klien | 404 Not Found, 403 Forbidden | | **5xx** | Kesalahan Server | 500 Internal Server Error |

2. HTML (Hypertext Markup Language)

HTML adalah kerangka atau struktur dari sebuah halaman web.

Elemen Formulir (Forms): Digunakan untuk mengumpulkan input dari pengguna.

- **<form>**: Elemen pembungkus.
 - **action**: URL tujuan pengiriman data.
 - **method**: Metode HTTP (GET atau POST).
- **<input>**: Elemen input utama.
 - **type**: Jenis input (text, password, submit).
 - **name**: Nama variabel untuk data yang dikirim.
- **<label>**: Label deskriptif untuk sebuah input.
- **<textarea>**: Untuk input teks yang panjang.

3. CSS (Cascading Style Sheets)

CSS digunakan untuk mengatur tampilan visual dan tata letak (layout) halaman HTML.

Konsep Layout:

- **Box Model**: Setiap elemen HTML adalah sebuah kotak yang terdiri dari content, padding, border, dan margin.
- **Flexbox**: Ideal untuk mengatur tata letak item dalam satu dimensi (baris atau kolom). Sangat baik untuk penyelarasan dan distribusi ruang.
- **Grid**: Ideal untuk mengatur tata letak item dalam dua dimensi (baris dan kolom), cocok untuk layout halaman yang kompleks.

4. JavaScript

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

Dasar-dasar:

- **Variabel:** Gunakan `let` untuk nilai yang bisa diubah dan `const` untuk nilai tetap.
- **Manipulasi DOM:** JavaScript dapat mengubah konten dan gaya HTML secara langsung.
- `// Mengubah teks dari elemen dengan id="pesan"`
- `document.getElementById("pesan").innerHTML = "Halo Dunia!";`

OOP (Object-Oriented Programming) dengan Class: Sintaks `class` (sejak ES6) menyediakan cara yang lebih jelas untuk membuat objek dan menangani pewarisan.

- **class:** Cetak biru untuk membuat objek.
- **constructor:** Metode khusus untuk membuat dan menginisialisasi objek.
- **extends:** Digunakan untuk membuat kelas anak yang mewarisi dari kelas induk.

Standard Library (Objek Bawaan):

- **Array:** Metode seperti `.map()`, `.filter()` untuk memanipulasi array.
- **Math:** Fungsi matematika seperti `Math.random()`.
- **Date:** Untuk bekerja dengan tanggal dan waktu.
- **JSON:** `JSON.parse()` dan `JSON.stringify()` untuk mengelola data JSON.

5. Bootstrap

Bootstrap adalah framework CSS untuk membangun situs web yang responsif dan *mobile-first* dengan cepat.

Fitur Utama:

- **Grid System:** Sistem tata letak 12 kolom berbasis Flexbox untuk membuat layout yang fleksibel.
- **Components:** Kumpulan komponen UI seperti *Modals*, *Navbars*, *Carousels*, dan *Alerts*.
- **Utilities:** Kelas-kelas bantuan untuk mengatur spasi, warna, perataan, dll. (contoh: `mt-3` untuk `margin-top`, `text-center`).

6. PHP & MySQL

Kombinasi kuat untuk pengembangan backend web dinamis.

- **PHP (Hypertext Preprocessor):** Bahasa skrip yang berjalan di server untuk memproses data, mengelola sesi, dan berinteraksi dengan database.
- **MySQL:** Sistem manajemen basis data relasional (RDBMS) untuk menyimpan, mengelola, dan mengambil data secara terstruktur.

Penggabungan dari semua jika diimplementasikan:

1. Struktur Database (MySQL) Buat tabel `users` dengan kolom esensial:

- id (INT, PRIMARY KEY, AUTO_INCREMENT)
- username (VARCHAR)
- email (VARCHAR, UNIQUE)
- password (VARCHAR) - **Sangat penting:** kolom ini akan menyimpan *hash* dari password, bukan password aslinya.

2. Alur Registrasi

1. **Formulir (HTML & Bootstrap):** Pengguna mengisi data di halaman register.php.
2. **Proses (PHP):**
 - Server menerima data melalui \$_POST.
 - **Validasi** input untuk memastikan data lengkap dan valid.
 - **Hashing Password:** Amankan password dengan password_hash().
 - \$hash = password_hash(\$_POST['password'], PASSWORD_DEFAULT);
 - **Simpan ke Database:** Jalankan query INSERT untuk menyimpan data pengguna (termasuk password yang sudah di-hash) ke tabel users.

3. Alur Login

1. **Formulir (HTML & Bootstrap):** Pengguna memasukkan email dan password di login.php.
2. **Proses (PHP):**
 - Ambil data pengguna dari database berdasarkan email yang dimasukkan.
 - **Verifikasi Password:** Bandingkan password yang diinput dengan hash di database menggunakan password_verify().
 - if (password_verify(\$_POST['password'], \$user['password'])) {
 - // Password cocok, login berhasil
 - **Buat Sesi:** Jika verifikasi berhasil, mulai sesi dengan session_start() dan simpan data pengguna (misalnya ID) di \$_SESSION untuk menjaga status login.

4. Halaman Terproteksi Pada setiap halaman yang memerlukan login (misalnya dashboard.php), letakkan kode ini di bagian paling atas untuk memeriksa sesi:

```
session_start();

if (!isset($_SESSION['user_id'])) {
    header("Location: login.php"); // Alihkan ke halaman login jika belum login
    exit();
}
```