

1. HTTP (Hypertext Transfer Protocol)

HTTP adalah protokol atau aturan yang mendasari komunikasi data di World Wide Web (WWW). Protokol ini bekerja dengan model client-server, di mana *client* (biasanya web browser) mengirimkan permintaan (*request*) ke *server*, dan server merespons dengan data (*response*).

Metode HTTP (HTTP Methods):

- GET: Meminta data dari server. Paling umum digunakan untuk mengambil halaman web. Data dikirim melalui URL.
- POST: Mengirim data ke server untuk diproses (misalnya, mengirimkan data formulir). Data dikirim dalam *body* permintaan, sehingga lebih aman untuk informasi sensitif.
- PUT: Memperbarui sumber daya yang ada di server.
- DELETE: Menghapus sumber daya di server.

2. HTML (Hypertext Markup Language)

HTML adalah bahasa markup standar untuk membuat struktur halaman web. HTML menggunakan "tag" untuk mendefinisikan elemen-elemen seperti judul, paragraf, gambar, dan tautan.

a. Dasar-Dasar HTML

Tag Umum:

- <h1> sampai <h6>: Tag untuk judul.
- <p>: Paragraf.
- <a>: Tautan (*anchor*), dengan atribut href untuk menentukan URL tujuan.
- : Gambar, dengan atribut src untuk sumber gambar dan alt untuk teks alternatif.
- <div>: Pembungkus generik (level blok) untuk mengelompokkan elemen.
- : Pembungkus generik (level inline) untuk mengelompokkan teks.
- , , : Untuk membuat daftar (unordered & ordered list).

b. Form HTML

Formulir digunakan untuk mengumpulkan input dari pengguna.

- **Elemen Utama:**
 - <form>: Pembungkus utama formulir. Atribut penting:
 - action: URL tujuan tempat data akan dikirim.
 - method: Metode HTTP yang digunakan (GET atau POST).

- `<input>`: Elemen input serbaguna. Atribut type menentukan jenisnya: text, password, email, number, checkbox, radio, submit.
- `<label>`: Label teks untuk sebuah elemen input.
- `<textarea>`: Untuk input teks multi-baris.
- `<button>`: Tombol yang dapat diklik.
- `<select>` dan `<option>`: Untuk membuat menu dropdown.

3. CSS (Cascading Style Sheets)

Konsep Layout Penting:

- **Box Model:** Setiap elemen dianggap kotak yang memiliki: content, padding (jarak dalam), border (garis tepi), margin (jarak luar).
- **Flexbox (display: flex):** Sistem layout modern untuk menata item dalam satu baris atau satu kolom. Sangat baik untuk alignment.
- **Grid (display: grid):** Sistem layout dua dimensi (baris dan kolom), cocok untuk tata letak halaman yang kompleks.

JavaScript (JS)

Bahasa pemrograman yang membuat halaman web menjadi **interaktif dan dinamis**. Berjalan di browser pengguna (sisi client).

- **Konsep Dasar:**
 - **Variabel:** let (bisa diubah) dan const (konstan) untuk menyimpan data.
 - **DOM Manipulation:** Kemampuan JS untuk mengubah konten dan style HTML secara langsung.
- **Standard Library:** Fungsi dan objek bawaan yang berguna, seperti:
 1. **Array Methods:** .map(), .filter(), .forEach() untuk memanipulasi array.
 2. **Math Object:** Math.random(), Math.floor().
 3. **Date Object:** Untuk bekerja dengan tanggal dan waktu.

4. Bootstrap

Framework CSS & JS siap pakai untuk membangun web yang **responsif** dan **mobile-first** dengan cepat.

- **Fungsi Utama:** Menyediakan kumpulan class CSS dan komponen JavaScript yang sudah jadi.
- **Cara Pakai:** Cukup hubungkan file CSS dan JS Bootstrap di HTML Anda.
- **Fitur Andalan:**
 - **Grid System:** Sistem tata letak 12 kolom (.container, .row, .col-md-6) untuk membuat layout yang otomatis menyesuaikan ukuran layar.
 - **Komponen:** Elemen UI yang sudah didesain, seperti Navbar, Card, Button, Modal (pop-up), dan Form.
 - **Utility Classes:** Class singkat untuk styling cepat, seperti m-3 (margin), p-3 (padding), text-center, bg-primary.

5. PHP (Hypertext Preprocessor)

Bahasa pemrograman yang berjalan di **sisi server**. Tugasnya adalah memproses data, berinteraksi dengan database, dan menghasilkan halaman HTML dinamis yang akan dikirim ke browser.

- **Fungsi Utama:**
 - Menghubungkan aplikasi web ke database.
 - Mengelola sesi pengguna (login/logout) menggunakan `$_SESSION`.

6. MySQL

Sistem manajemen **database** relasional (RDBMS) yang populer untuk menyimpan dan mengelola data aplikasi web secara terstruktur dalam bentuk tabel-tabel.

- **Konsep:** Data disimpan dalam tabel yang terdiri dari baris (records) dan kolom (fields).
- **Perintah Dasar SQL (digunakan di PHP):**
 - **SELECT:** Mengambil data.
 - **INSERT:** Menambahkan data baru.
 - **UPDATE:** Mengubah data yang sudah ada.
 - **DELETE:** Menghapus data.

Integrasi: PHP + MySQL (+ Bootstrap)

Studi Kasus: Alur Proses Login

Ini adalah contoh bagaimana semua teknologi bekerja sama.

1. **Tampilan (Front-End):**
 - Pengguna membuka `login.php`. Halaman ini dibuat dengan **HTML** dan dipercantik dengan **Bootstrap** untuk menampilkan formulir login yang rapi.
2. **Input Pengguna:**
 - Formulir mengirimkan data ini ke script **PHP** di server menggunakan metode **POST**.
3. **Proses di Server (Back-End):**
 - PHP terhubung ke database **MySQL**.
 - PHP menjalankan query **SQL SELECT** untuk mencari user berdasarkan username yang dikirim.
4. **Verifikasi & Keamanan:**
 - **PENTING:** PHP menggunakan fungsi `password_verify()` untuk membandingkan password yang diinput pengguna dengan *hash* yang tersimpan. Ini adalah cara aman untuk memverifikasi password.
5. **Manajemen Sesi:**
 - PHP kemudian mengarahkan (redirect) pengguna ke halaman dasbor yang terproteksi.
6. **Halaman Terproteksi:**
 - Setiap halaman yang butuh login akan memeriksa `$_SESSION` terlebih dahulu. Jika tidak ada sesi, pengguna akan dilempar kembali ke halaman login.