

Nama : Ahmad Zaki Mahogra

NIM : 2023010144

Kelas : TI C

RESUME FINAL EXAM

1. HTTP (Hypertext Transfer Protocol)

HTTP adalah protokol dasar yang memungkinkan pertukaran data di World Wide Web. Ini adalah protokol client-server, di mana permintaan (request) dikirim oleh klien (misalnya, browser web) ke server, dan server merespons dengan data yang diminta (response). Metode HTTP yang umum meliputi GET (untuk mengambil data), POST (untuk mengirim data), PUT/PATCH (untuk memperbarui data), dan DELETE (untuk menghapus data). Meskipun HTTP secara inheren stateless (server tidak menyimpan informasi sesi antar permintaan), mekanisme seperti cookies digunakan untuk mempertahankan status sesi.

2. HTML (HyperText Markup Language)

HTML adalah bahasa markup standar untuk membuat dan menyusun konten halaman web. Ini menggunakan serangkaian "elemen" atau "tag" untuk membungkus berbagai bagian konten agar tampil atau berfungsi dengan cara tertentu.

- Dasar HTML:

- ☐ Setiap dokumen HTML dimulai dengan deklarasi `<!DOCTYPE html>` dan dibungkus oleh elemen `<html>`.
- ☐ Bagian `<head>` berisi metadata tentang halaman (tidak ditampilkan di browser, seperti judul halaman `<title>`, meta tag, tautan ke CSS).
- ☐ Bagian `<body>` berisi semua konten yang terlihat oleh pengguna (teks, gambar, video, dll.).

- Form HTML:

- ☐ Digunakan untuk mengumpulkan input dari pengguna. Didefinisikan menggunakan tag `<form>`.
- ☐ Berisi berbagai kontrol input seperti `<input type="text">` (teks), `<input type="password">` (kata sandi), `<input type="email">`, `<input type="submit">` (tombol kirim), `<textarea>` (teks panjang), `<select>` (daftar pilihan), dan `<button>`. Atribut seperti name, id, value, placeholder, dan action/method pada form sangat penting.

3. CSS (Cascading Style Sheets)

CSS adalah bahasa yang digunakan untuk mendeskripsikan gaya presentasi dokumen HTML (misalnya, warna, font, tata letak, spasi). Tujuan utamanya adalah memisahkan konten (HTML) dari presentasi (CSS), sehingga memudahkan pengelolaan dan perubahan desain.

Dasar CSS:

- Sintaks dasar CSS terdiri dari selector (menargetkan elemen HTML) dan blok deklarasi (properti dan nilai, misal: color: blue;).
- CSS dapat diimplementasikan secara inline (menggunakan atribut style), embedded (dalam tag <style> di <head>), atau eksternal (file .css terpisah yang ditautkan menggunakan <link>).
- Selector meliputi tipe elemen (p), class (.nama-class), ID (#nama-id), dan kombinasi lainnya.

Layout CSS:

- **Normal Flow:** Cara default browser menata elemen (block elemen menumpuk secara vertikal, inline elemen berbaris horizontal).
- **CSS Box Model:** Setiap elemen HTML diperlakukan sebagai kotak yang terdiri dari konten, padding, border, dan margin.
- **Float:** Digunakan untuk menggeser elemen ke kiri atau kanan, memungkinkan teks atau elemen lain mengalir di sekitarnya. Dulunya sering digunakan untuk tata letak multi-kolom, namun sekarang lebih banyak menggunakan Flexbox dan Grid.
- **Positioning:** Properti position (static, relative, absolute, fixed, sticky) memungkinkan kontrol lebih lanjut atas penempatan elemen di luar aliran normal dokumen.

4. JavaScript

JavaScript adalah bahasa pemrograman lintas-platform, berorientasi objek, yang digunakan untuk membuat halaman web menjadi interaktif. Ia berjalan di sisi klien (browser) namun juga dapat digunakan di sisi server (misalnya, dengan Node.js).

Dasar JavaScript:

- Memiliki tipe data (string, number, boolean, object, array, null, undefined), operator (aritmatika, perbandingan, logika), struktur kontrol (if/else, switch), dan loop (for, while).
- Mampu memanipulasi Document Object Model (DOM) untuk mengubah konten, struktur, dan gaya halaman secara dinamis.
- Dapat menangani event (kejadian) dari pengguna (klik, input keyboard, hover) untuk memicu fungsi atau tindakan tertentu.

OOP (Object-Oriented Programming) JavaScript:

- JavaScript mendukung paradigma OOP melalui model berbasis prototipe (prototype-based) dan juga sintaks class (diperkenalkan di ES6/ES2015) yang merupakan sintaks gula di atas prototipe.
- Konsep utama meliputi: objek (kumpulan properti dan metode), enkapsulasi, pewarisan (melalui prototipe atau extends dengan class), dan polimorfisme.

5. Bootstrap

Bootstrap adalah *framework* CSS *open-source* yang paling populer untuk mengembangkan proyek web responsif dan mobile-first dengan cepat. Ia menyediakan koleksi komponen HTML, CSS, dan JavaScript yang siap pakai.

Tujuan : mempercepat pengembangan *frontend* dengan menyediakan *boilerplate* dan komponen yang sudah teruji dan konsisten.

Cara Cepat Memulai: Dapat langsung digunakan dengan menautkan file CSS dan JS Bootstrap melalui CDN (Content Delivery Network) di file HTML Anda.

Pengaturan Global Penting:

- Memerlukan HTML5 doctype.
- Penggunaan `<meta name="viewport" content="width=device-width, initial-scale=1">` untuk memastikan *rendering* responsif.
- Mengatur box-sizing: border-box secara global untuk perhitungan ukuran elemen yang lebih intuitif.
- **Reboot:** Kumpulan perubahan CSS spesifik elemen dalam satu file untuk memberikan dasar yang konsisten antar browser dan menyediakan *reset* yang beropini untuk elemen HTML umum.

6. PHP, Bootstrap, MySQL (Sistem Login/Register)

Kombinasi PHP, Bootstrap, dan MySQL umum digunakan untuk membangun aplikasi web dinamis, seperti sistem login dan registrasi.

- **Peran Masing-masing:**
 - **PHP:** Bertindak sebagai bahasa pemrograman *server-side* yang memproses data yang dikirim dari form, berinteraksi dengan database MySQL, dan mengelola sesi pengguna.
 - **MySQL:** Berfungsi sebagai database untuk menyimpan data pengguna (username, password terenkripsi, email, dll.) dan informasi lain yang diperlukan.
 - **Bootstrap:** Digunakan untuk mendesain *user interface* (UI) dari form login dan registrasi, memastikan tampilannya menarik dan responsif di berbagai perangkat.
- **Alur Kerja Sistem Login/Registrasi:**
 1. **Persiapan Lingkungan:** Instalasi server web (Apache/Nginx), PHP, dan MySQL.
 2. **Struktur Database:** Buat database (misalnya, *pesbuk*) dan tabel pengguna (*users*) di MySQL dengan kolom seperti *id*, *username*, *password*, *email*.
 3. **Koneksi Database:** PHP terhubung ke MySQL (misalnya, menggunakan MySQLi atau PDO).
 4. **Registrasi Pengguna:**
 - Pengguna mengisi form registrasi (didesain dengan Bootstrap).
 - Data dikirim ke skrip PHP.
 - PHP memvalidasi dan memfilter input.
 - **Kata Sandi Di-hash:** Kata sandi dienkripsi (misalnya, menggunakan `password_hash()`) sebelum disimpan di database untuk keamanan.
 - Data pengguna disimpan ke tabel *users* di MySQL.
 5. **Login Pengguna:**
 - Pengguna mengisi form login (didesain dengan Bootstrap).
 - Data dikirim ke skrip PHP.
 - PHP mengambil data pengguna dari database berdasarkan *username/email* yang diberikan.
 - PHP memverifikasi kata sandi yang dimasukkan dengan hash yang disimpan di database menggunakan `password_verify()`.
 - Jika cocok, sesi pengguna dimulai (`session_start()`) dan pengguna dialihkan ke halaman yang dilindungi (misalnya, halaman profil).
 6. **Manajemen Sesi:** Variabel sesi (`$_SESSION`) digunakan di PHP untuk melacak status login pengguna di seluruh halaman.
 7. **Logout:** Sesi pengguna dihancurkan (`session_destroy()`) saat pengguna melakukan logout.