

HTML: Struktur Dasar Halaman Web

HTML (Hypertext Markup Language) adalah tulang punggung dari setiap halaman web. Ia mendefinisikan struktur dan konten dari sebuah dokumen web menggunakan elemen-elemen yang disebut **tag**. Struktur dasar dokumen HTML selalu diawali dengan tag `<html>` yang membungkus dua bagian utama: `<head>` dan `<body>`.

- **<head>**: Bagian ini berisi metadata atau informasi tentang dokumen yang tidak ditampilkan secara langsung di halaman. Ini termasuk judul halaman yang didefinisikan dengan tag `<title>`, tautan ke file eksternal seperti CSS, dan informasi meta lainnya.
- **<body>**: Bagian ini berisi semua konten yang terlihat oleh pengguna, seperti teks, gambar, video, dan elemen interaktif lainnya.

Untuk memformat konten di dalam `<body>`, HTML menyediakan berbagai tag, di antaranya:

- **Teks**: `<h1>` hingga `<h6>` untuk judul dengan berbagai tingkatan, `<p>` untuk paragraf, `` untuk teks tebal, `<i>` untuk teks miring, `
` untuk pindah baris, dan `<hr>` untuk garis horizontal.
- **Tautan dan Gambar**: Tag `<a>` (anchor) dengan atribut `href` digunakan untuk membuat hyperlink ke halaman lain. Tag `` dengan atribut `src` (sumber) dan `alt` (teks alternatif) digunakan untuk menyisipkan gambar.
- **Daftar (List)**: Daftar terurut dibuat dengan `` (ordered list) dan daftar tak terurut dengan `` (unordered list). Setiap item di dalam daftar dibungkus oleh tag `` (list item).
- **Tabel**: Untuk menampilkan data terstruktur, digunakan tag `<table>`. Di dalamnya terdapat `<tr>` (table row) untuk setiap baris, `<th>` (table header) untuk sel judul, dan `<td>` (table data) untuk sel data.
- **Formulir**: Formulir interaktif dibuat dengan tag `<form>`. Di dalamnya bisa terdapat berbagai elemen input seperti `<input>` (untuk teks, password, checkbox, radio button), `<textarea>` (untuk teks panjang), `<select>` (untuk dropdown), dan `<button>` atau `<input type="submit">` untuk mengirimkan data.

CSS: Menata Tampilan dan Layout Halaman

CSS (Cascading Style Sheets) adalah bahasa yang digunakan untuk mengontrol presentasi visual dari elemen-elemen HTML. Dengan CSS, Anda bisa mengatur warna, font, spasi, layout, dan banyak lagi, memastikan

tampilan yang konsisten di berbagai perangkat dan browser.

Cara Penerapan CSS:

1. **Eksternal**: Menulis CSS di file `.css` terpisah dan menautkannya ke HTML menggunakan tag `<link>`. Ini adalah metode yang paling direkomendasikan.
2. **Internal**: Menulis CSS di dalam tag `<style>` yang ditempatkan di bagian `<head>` dokumen HTML.
3. **Inline**: Menerapkan gaya langsung ke elemen HTML menggunakan atribut `style`. Metode ini kurang disarankan karena sulit dikelola.

Konsep Inti CSS:

- **Selektor**: Digunakan untuk menargetkan elemen HTML yang ingin diubah gayanya. Selektor dasar meliputi selektor *tipe* (misalnya `p`), *class* (misalnya `.nama-class`), dan *ID* (misalnya `#id-unik`).
- **Box Model**: Setiap elemen HTML dapat dianggap sebagai sebuah kotak yang terdiri dari empat lapisan: *Content* (konten), *Padding* (ruang di sekitar konten), *Border* (garis di luar padding), dan *Margin* (ruang di luar border).
- **Layout**: CSS menyediakan teknologi canggih untuk mengatur tata letak halaman:
 - **Flexbox**: Sistem layout satu dimensi yang ideal untuk mengatur item dalam baris atau kolom. Sangat efisien untuk menyelaraskan dan mendistribusikan ruang di antara item, bahkan saat ukurannya dinamis. Properti utamanya adalah `display: flex`.
 - **Grid Layout**: Sistem layout dua dimensi (baris dan kolom) yang kuat untuk membuat tata letak halaman yang kompleks. Memberikan kontrol penuh atas penempatan elemen dalam sebuah grid. Properti utamanya adalah `display: grid`.
- **Positioning**: Memungkinkan penempatan elemen di luar alur normal dokumen. Nilai properti `position` mencakup `static` (default), `relative`, `absolute`, `fixed` (selalu menempel di layar), dan `sticky` (kombinasi `relative` dan `fixed`).
- **Responsive Design**: Menggunakan **Media Queries** (`@media`), CSS dapat menerapkan aturan gaya yang berbeda berdasarkan

karakteristik perangkat, seperti lebar layar. Ini memungkinkan halaman web beradaptasi dan terlihat bagus di desktop, tablet, maupun smartphone.

JavaScript: Menghidupkan Halaman Web

JavaScript adalah bahasa pemrograman yang berjalan di sisi klien (browser) untuk membuat halaman web menjadi interaktif dan dinamis. Jika HTML adalah kerangka dan CSS adalah kulitnya, maka JavaScript adalah otaknya.

Fungsi Utama JavaScript:

- **Manipulasi DOM (Document Object Model):** JavaScript dapat memilih, mengubah, menambah, atau menghapus elemen HTML dan CSS secara dinamis. Ini memungkinkan perubahan konten halaman tanpa perlu memuat ulang.
- **Event Handling:** Merespons interaksi pengguna seperti klik (click), pengarah kursor (hover), atau pengiriman formulir (submit) untuk memicu aksi tertentu.
- **Validasi Formulir:** Memeriksa input pengguna secara langsung di browser sebelum data dikirim ke server, memberikan umpan balik instan dan mengurangi beban server.
- **Asynchronous Operations (AJAX):** Mengambil atau mengirim data ke server di latar belakang tanpa mengganggu interaksi pengguna dengan halaman. Fitur modern seperti **Promise** dan sintaks `async/await` mempermudah pengelolaan operasi ini.
- **Animasi:** Membuat efek visual dan transisi yang halus dengan memanipulasi properti CSS dari waktu ke waktu.

Struktur Dasar JavaScript:

- **Variabel:** Wadah untuk menyimpan data, dideklarasikan menggunakan `var`, `let`, atau `const`.
- **Tipe Data:** Termasuk string (teks), number (angka), boolean (`true/false`), object, dan lainnya.
- **Struktur Kontrol:** Seperti `if...else` untuk kondisional dan `for` atau `while` untuk perulangan.
- **Fungsi:** Blok kode yang dapat digunakan kembali untuk melakukan tugas tertentu.

ECMAScript (ES): Adalah standar yang menjadi dasar JavaScript. Versi seperti ES6 (ECMAScript 2015)

memperkenalkan banyak fitur modern yang membuat kode lebih bersih dan efisien, seperti *arrow functions*, *classes*, dan *modules*.

Protokol HTTP & Model OSI

Komunikasi di web didasari oleh protokol, dan yang paling fundamental adalah HTTP.

- **HTTP (Hypertext Transfer Protocol):** Protokol aplikasi yang mengatur komunikasi antara klien (seperti browser) dan server. HTTP bersifat *stateless*, artinya setiap permintaan dianggap independen. Ia menggunakan metode seperti:
 - GET: Meminta data dari server.
 - POST: Mengirim data baru ke server.
 - PUT: Memperbarui data yang ada di server.
 - DELETE: Menghapus data dari server.
- **HTTPS (HTTP Secure):** Versi aman dari HTTP yang menggunakan enkripsi TLS/SSL untuk melindungi data yang ditransfer antara klien dan server, menjaga privasi dan integritas data.
- **Model OSI (Open Systems Interconnection):** Kerangka konseptual yang membagi fungsi jaringan menjadi tujuh lapisan, mulai dari fisik hingga aplikasi. HTTP berada di **Lapisan 7 (Application)**, lapisan yang paling dekat dengan pengguna. Lapisan lainnya mencakup **Transport** (misalnya TCP) yang memastikan pengiriman data yang andal, dan **Network** (misalnya IP) yang menangani pengalamatan dan routing.

Alur Kerja HTTP:

1. **Permintaan (Request):** Browser mengirim permintaan ke server, yang berisi URL, metode HTTP, *header* (metadata tambahan), dan terkadang *body* (data).
2. **Respons (Response):** Server membalas dengan *status code* (misalnya 200 OK untuk sukses, 404 Not Found untuk tidak ditemukan), *header*, dan *body* (konten halaman).

PHP: Pemrograman Sisi Server

Sementara JavaScript berjalan di browser pengguna (sisi klien), PHP (Hypertext Preprocessor) adalah bahasa skrip yang berjalan di **sisi server**. PHP sangat populer untuk

pengembangan web karena kemampuannya memproses data dari formulir, berinteraksi dengan database, mengelola sesi pengguna, dan menghasilkan konten HTML secara dinamis sebelum dikirim ke browser.

Contoh Kasus Penggunaan PHP:

- **Sistem Login dan Registrasi:** PHP dapat memvalidasi data yang dikirim dari formulir login, memeriksa kredensial pengguna di database, dan membuat sesi untuk menjaga status login pengguna saat mereka bernavigasi di situs.
- **Menampilkan Konten Dinamis:** Mengambil data produk, artikel berita, atau postingan blog dari database dan menampilkannya sebagai halaman HTML.
- **Mengelola File:** Mengunggah file yang dikirim oleh pengguna, seperti gambar profil atau dokumen.

PHP disematkan langsung di dalam kode HTML menggunakan tag `<?php ... ?>`, membuatnya mudah diintegrasikan untuk menghasilkan halaman web yang dinamis.

MySQL: Sistem Manajemen Database

Hampir semua aplikasi web dinamis memerlukan tempat untuk menyimpan data secara permanen. **MySQL** adalah salah satu Sistem Manajemen Database Relasional (RDBMS) sumber terbuka yang paling banyak digunakan di dunia. Ia menggunakan **SQL (Structured Query Language)** untuk mengelola data.

Peran MySQL dalam Aplikasi Web:

- **Penyimpanan Data:** Menyimpan informasi pengguna (nama, email, password terenkripsi), konten situs (artikel, produk), dan data lainnya dalam tabel-tabel yang terstruktur.
- **Integritas Data:** Memastikan data yang disimpan konsisten dan akurat.
- **Pengambilan Data:** Bekerja sama dengan bahasa sisi server seperti PHP untuk menjalankan *query* (permintaan data) guna mengambil, menyaring, dan menyortir data yang akan ditampilkan kepada pengguna.

Kombinasi **PHP dan MySQL** adalah fondasi klasik untuk banyak aplikasi web, mulai dari blog sederhana hingga platform e-commerce yang kompleks.

Bootstrap adalah framework CSS, JavaScript, dan HTML front-end yang sangat populer. Tujuannya adalah untuk menyederhanakan dan mempercepat proses pembuatan antarmuka web yang responsif dan menarik secara visual. Daripada menulis semua kode CSS dan JavaScript dari nol, pengembang dapat memanfaatkan komponen siap pakai dari Bootstrap.

Fitur Utama Bootstrap:

- **Sistem Grid Responsif:** Sistem layout berbasis *flexbox* yang kuat dan mudah digunakan. Pengembang dapat dengan cepat membuat tata letak kolom yang secara otomatis menyesuaikan diri dengan ukuran layar perangkat (desktop, tablet, atau mobile).
- **Komponen Siap Pakai:** Menyediakan berbagai komponen UI yang sudah ditata gayanya dan fungsional, seperti: Tombol (Buttons), Navigasi (Navbar), Kartu (Cards), Formulir (Forms), Modal (jendela pop-up), Carousel (slider gambar)
- **Utilitas:** Kumpulan kelas CSS pembantu (misalnya, untuk margin, padding, warna, dan perataan teks) yang memungkinkan penyesuaian cepat tanpa perlu menulis CSS kustom.

Bootstrap: Framework CSS untuk Pengembangan Cepat