

Nama: Annisa Dian Amarta

NIM: 20230140149

Kelas: C

CSS

CSS (Cascading Style Sheets) digunakan untuk mengatur tampilan dan tata letak konten HTML. Tiga cara menambahkan CSS ke HTML adalah *inline*, *internal*, dan *external*. CSS terdiri dari *selector* dan *property-value*, dan bisa menggunakan tag, class (.), atau ID (#) sebagai selector. Tipe selector lainnya termasuk combinator dan attribute selectors. Pengaturan teks meliputi warna, ukuran, jarak, dan dekorasi. CSS juga memungkinkan penggunaan font khusus (seperti Google Fonts), serta mengatur background menggunakan warna, gambar, atau gradien. Konsep *Box Model* penting untuk memahami struktur element, yang terdiri dari content, padding, border, dan margin. CSS Layout digunakan untuk mengatur tata letak halaman web agar menarik dan responsif. Layout standar disebut *Normal Flow*, namun sering kali kita perlu menggunakan properti seperti display, flexbox, dan grid untuk penataan lebih kompleks. Flexbox cocok untuk layout satu dimensi, sementara Grid lebih sesuai untuk layout dua dimensi. Properti penting termasuk justify-content, align-items, dan gap. CSS juga mendukung *positioning* (seperti absolute, relative, fixed, sticky) dan *media queries* untuk membuat tampilan adaptif terhadap berbagai ukuran layar. Tambahan seperti z-index, float, dan fraction unit memperkuat kontrol layout lebih lanjut.

HTML

HTML (HyperText Markup Language) adalah bahasa dasar untuk membuat halaman web. HTML membentuk struktur konten web dengan menggunakan tag seperti <p> (paragraf), <h1> hingga <h6> (heading), <a> (link), (gambar), dan masih banyak lagi. HTML juga mendukung tampilan media seperti gambar, audio, video, serta layout menggunakan <div> dan . HTML5 menambahkan elemen *semantic* seperti <header>, <article>, dan <footer> agar struktur dokumen lebih jelas. Web ditampilkan melalui Web Browser dan dijalankan melalui Web Server. Kita bisa membuat dan menampilkan HTML secara lokal dengan *Live Server* atau meng-hosting-nya melalui layanan seperti Niagahoster. HTML Form memungkinkan pengguna mengirim data ke server melalui tag <form>, yang berisi berbagai elemen input seperti <input>, <textarea>, <select>, dan <button>. Jenis input yang tersedia meliputi teks, password, email, tanggal, warna, file, checkbox, radio, dan lainnya. Properti penting pada form mencakup action, method, dan enctype. Input bisa diproses menggunakan bahasa server-side seperti PHP atau Node.js. Fitur tambahan seperti label (<label>), grouping dengan <fieldset> dan <legend>, serta elemen interaktif seperti <datalist> dan autocomplete, meningkatkan kenyamanan dan aksesibilitas pengguna.

JAVASCRIPT

JavaScript (JS) adalah bahasa pemrograman utama untuk membuat halaman web menjadi dinamis dan interaktif. JS dijalankan di browser (frontend), dan juga bisa di server (backend) melalui Node.js, menjadikannya bahasa *fullstack*. JavaScript bersifat dinamis dan mengikuti standar ECMAScript.

1. Dasar JavaScript

- Tipe Data: number, string, boolean, null, undefined, array, dan object.

- Variabel: Gunakan let untuk variabel bisa berubah, const untuk nilai tetap.
- Operator: Aritmatika (+, -, *, /), perbandingan (==, ===), logika (&&, ||, !).
- Fungsi: Dapat dibuat dengan function biasa, atau arrow function (=>).
- Kontrol Program: if-else, switch, for, while, do while, break, continue.
- Array & Object: Digunakan untuk menyimpan dan mengelola kumpulan data.
- Fitur lain: Template string, konversi data (parseInt, parseFloat), dan console (log, error, table).

2. Object-Oriented Programming (OOP)

- Object: Struktur data berisi property (data) dan method (fungsi).
- Class: Blueprint untuk membuat object; dibuat dengan class dan constructor.
- Inheritance: Pewarisan dari class induk menggunakan extends, pemanggilan dengan super().
- Method & Property: Bisa bersifat public, private (dengan #), atau static.
- Prototype: Mekanisme warisan sebelum ES6, masih mendasari cara kerja JS modern.
- Fitur OOP Lainnya: Getter, setter, instanceof, dan error handling (try-catch-finally).

3. Standard Library

- Number: Number.isNaN(), toString(), MAX_VALUE, dll.
- String: toLowerCase(), toUpperCase(), includes(), slice().
- Array: push(), pop(), map(), filter(), reduce(), find().
- Object: Object.keys(), values(), assign(), freeze(), seal().
- JSON: JSON.stringify() untuk konversi object ke string, JSON.parse() sebaliknya.
- Date: new Date(), getTime(), now(), parse(), toISOString().
- Math: Math.round(), random(), pow(), floor(), PI.
- Map & Set: Struktur data modern, mendukung iterable dan data unik.
- RegExp: Digunakan untuk pencocokan pola dalam string.

HTTP

HTTP (Hypertext Transfer Protocol) adalah protokol komunikasi antara client (biasanya browser) dan server di web. HTTP digunakan untuk mengirim dan menerima data seperti HTML, CSS, JavaScript, gambar, dan lainnya. HTTP bersifat stateless, artinya setiap permintaan tidak bergantung pada permintaan sebelumnya. Untuk menyimpan sesi seperti login, digunakan HTTP Cookie, yang memungkinkan server mengenali user di setiap request.

HTTP terdiri dari:

- HTTP Request: dikirim oleh client, berisi URL, method (GET, POST, PUT, DELETE, dll), header, dan body (opsional).

- HTTP Response: dikirim oleh server, berisi status (seperti 200 OK, 404 Not Found), header, dan body (data yang diminta).

Bagian penting lainnya:

- URL: alamat sumber daya web, terdiri dari schema, domain, path, query, dan anchor.
- Header: metadata dalam request/response (misalnya Content-Type, Authorization).
- Status Code: menunjukkan hasil request (misal: 2xx sukses, 4xx kesalahan client, 5xx kesalahan server).
- HTTPS: versi aman dari HTTP yang menggunakan enkripsi SSL/TLS.
- Caching: memungkinkan client menyimpan data agar tidak perlu mengunduh ulang dari server.

HTTP juga mendukung fitur lanjutan seperti redirect, WebSocket, dan REST API.