

## 1. Konsep dan cara kerja website beserta browser

**Website** = kumpulan halaman web yang saling terhubung dan dapat diakses melalui internet. Melibatkan penggunaan HTML, CSS, dan JavaScript untuk membangun tampilan dan fungsionalitas yang interaktif.

### Cara kerja website

- 1) User memasukkan URL (*Uniform Resource Locator*) ke dalam browser. URL => alamat dari website yang ingin diakses.
- 2) Browser mengirimkan permintaan HTTP (*Hypertext Transfer Protocol*) ke server yang berisi website tersebut.
- 3) Server menerima permintaan dan melakukan pemrosesan.
- 4) Server mengirimkannya ke browser dalam bentuk respons HTTP.
- 5) Browser menerima respons dan memuat data awal.
- 6) JavaScript dijalankan di sisi client untuk memproses, memanipulasi, dan merender data tersebut menjadi tampilan yang interaktif.
- 7) Browser mengirim ke user berupa display.
- 8) Selama interaksi pengguna dengan halaman web, JavaScript di browser dapat melakukan permintaan tambahan ke server.

### Perbedaan Client-side

- Lokasi pemrosesan: Kode dieksekusi di sisi klien atau perangkat pengguna (seperti browser web).
- Bahasa pemrograman: HTML, CSS, dan JavaScript.
- Tanggung jawab: Bertanggung jawab untuk tampilan, interaksi pengguna, dan pemrosesan data di perangkat pengguna.
- Contoh penggunaan: Validasi formulir, efek visual, animasi, validasi input di sisi klien sebelum mengirim ke server.

### Perbedaan Server-side

- Lokasi pemrosesan: Kode dieksekusi di sisi server (server web).
- Bahasa pemrograman: PHP, Python, Ruby, dan lain-lain.
- Tanggung jawab: Bertanggung jawab untuk pemrosesan permintaan dari klien, pengolahan logika bisnis, dan interaksi dengan basis data.
- Contoh penggunaan: Pemrosesan formulir, autentikasi pengguna, pengambilan data dari basis data, pemrosesan transaksi keuangan, pengiriman email.

## 2. Tag HTML untuk layouting

**<header>** – Menentukan header untuk dokumen atau bagian. **<nav>** – Menentukan wadah untuk tautan navigasi. **<section>** – Menentukan bagian dalam dokumen. **<article>** – Menentukan artikel.

**<aside>** – Menentukan konten selain konten (seperti bilah sisi/sidebar). **<footer>** – Menentukan footer untuk dokumen atau bagian. **<details>** – Menentukan detail

tambahan yang dapat dibuka dan ditutup pengguna sesuai permintaan. **<summary>** – Menentukan heading untuk elemen **<details>**.

### Tag HTML 5

**<!DOCTYPE html>** deklarasi dari halaman web. **<html>** element root dari html. **<head>** elemen yang berisi informasi dari meta-data. **<title>** elemen yang berisi judul dari halaman web **<body>** elemen yang akan menampilkan visualisasi di browser.

## 3. Koneksi CSS dalam HTML (modul 5)

- a. Inline – Syntax CSS di letakkan di dalam atribut style dalam elemen HTML.
- b. Internal – Syntax CSS di letakkan di dalam tag **<style>**, diletakkan pada bagian tag **<head>** pada HTML.
- c. Eksternal – menggunakan elemen **<link>**, file css dan html terpisah.

## 4. Kekurangan dan Kelebihan Bootstrap

**Bootstrap** adalah front-end framework (CSS) untuk memudahkan dan mempercepat proses pengembangan suatu aplikasi web.

**Kelebihan Bootstrap** : Ramah untuk Pemula, Grid System yang Canggih, Kompatibilitas dengan Web Browser Terbaru, Bersifat Open-Source, Fitur kustomisasi yang berlimpah, Rutin Diperbarui, Tersedia Dokumentasi Lengkap, Ketersediaan resource & dukungan, Meningkatkan konsistensi desain, Membuat website mobile-friendly, Menghemat waktu pembuatan website, Meminimalisir bug antar browser.

**Kekurangan Bootstrap** : Elemennya Banyak Digunakan Developer Website Lain, Bisa Memperlambat Website, Perlu proses pembelajaran.

## 5. Tipe Data pada Javascript (modul 7)

- 1) **Undefined** = Mewakili variabel yang belum diinisialisasi/tidak memiliki nilai. Dalam JavaScript, jika variabel dideklarasikan tetapi tidak diinisialisasi, nilainya akan menjadi 'undefined'.
- 2) **Numbers** = Mewakili angka baik bilangan bulat maupun pecahan. Contoh: '42', '3.14', '-10'
- 3) **Strings** = Mewakili urutan karakter. String diapit oleh tanda kutip. Contoh: 'Hello', 'World'.
- 4) **Boolean** = Mewakili nilai kebenaran. Hanya memiliki dua nilai, yaitu 'true' atau 'false'.
- 5) **Null** = Mewakili nilai null/tidak ada nilai. Dalam JavaScript, 'null' adalah tipe data yang hanya memiliki satu nilai yaitu 'null'.
- 6) **Arrays** = Digunakan untuk menyimpan dan mengelola kumpulan nilai dalam satu variabel. Array adalah jenis objek khusus yang menyimpan nilai-nilai dengan indeks numerik. Contoh: '[1, 2, 3, 4]'.
- 7) **Object** = Digunakan untuk menyimpan koleksi nilai sebagai pasangan nama-properti (key value pairs). Objek dapat berisi metode dan properti. Contoh: '{name: 'John', age: 25}'
- 8) **Function** = Digunakan untuk mendefinisikan blok kode yang dapat dipanggil dan dijalankan. Fungsi juga dapat menerima argumen dan mengembalikan nilai. Contoh:  

```
function add(a, b) {  
  return a + b;  
}
```