

## Rangkuman Materi Pengembangan Web: Dari Dasar hingga Fungsional

Dokumen ini merangkum konsep-konsep fundamental dalam pengembangan web, mulai dari protokol komunikasi, struktur, gaya, interaktivitas, hingga pengembangan sisi server dengan basis data.

### Halaman 1: Fondasi Web

#### 1. HTTP (Hypertext Transfer Protocol)

HTTP adalah **protokol** atau aturan yang menjadi dasar komunikasi data di World Wide Web (WWW). Ia bekerja dengan model permintaan-respons (request-response) antara klien (biasanya peramban web/browser) dan server.

- **Cara Kerja:**
  1. **Klien Mengirim Permintaan (Request):** Browser meminta sumber daya (seperti halaman HTML, gambar, atau data) dari server. Contohnya: `GET /index.html HTTP/1.1.`
  2. **Server Merespons (Response):** Server memproses permintaan dan mengirimkan kembali sumber daya yang diminta beserta *status code*.
- **Metode HTTP Umum:**
  - `GET`: Meminta data dari server.
  - `POST`: Mengirim data ke server untuk dibuat (misalnya, saat mengisi formulir).
  - `PUT`: Memperbarui data yang ada di server.
  - `DELETE`: Menghapus data di server.
- **Kode Status (Status Codes):**
  - **2xx (Sukses):** `200 OK` (permintaan berhasil).
  - **3xx (Redirection):** `301 Moved Permanently` (sumber daya dipindahkan).
  - **4xx (Client Error):** `404 Not Found` (sumber daya tidak ditemukan).
  - **5xx (Server Error):** `500 Internal Server Error` (terjadi masalah di server).

#### 2. HTML (Hypertext Markup Language): Dasar

HTML adalah bahasa markah standar untuk **membuat struktur** halaman web. Ia mendefinisikan "kerangka" konten menggunakan elemen-elemen yang direpresentasikan oleh *tag*.

- **Struktur Dasar Dokumen HTML:**

HTML

```

<!DOCTYPE html>
<html>
<head>
  <title>Judul Halaman</title>
</head>
<body>
  <h1>Judul Utama</h1>
  <p>Ini adalah sebuah paragraf.</p>
  <a href="https://www.google.com">Ini adalah tautan</a>
  
</body>
</html>

```

- **Elemen Penting:**

- `<!DOCTYPE html>`: Mendefinisikan tipe dokumen.
- `<html>`: Elemen akar dari halaman.
- `<head>`: Berisi informasi meta tentang halaman (judul, karakter set, dll.).
- `<body>`: Berisi konten yang terlihat oleh pengguna.
- `<h1>` hingga `<h6>`: Judul (Heading).
- `<p>`: Paragraf.
- `<a>`: Tautan (Anchor).
- `<img>`: Gambar.
- `<div>`: Pembungkus generik untuk mengelompokkan elemen.
- `<span>`: Pembungkus sebaris (inline) untuk teks.

### 3. HTML: Form

Formulir HTML digunakan untuk **mengumpulkan input dari pengguna**. Data yang diisi kemudian dikirim ke server untuk diproses.

- **Elemen Utama Form:**

- `<form>`: Elemen pembungkus yang mendefinisikan formulir. Atribut penting:
  - `action`: URL tujuan tempat data akan dikirim.
  - `method`: Metode HTTP yang digunakan (GET atau POST).

- **Elemen Input:**

- `<input>`: Elemen paling serbaguna. Atribut `type` menentukan jenisnya:
    - `type="text"`: Kotak teks satu baris.
    - `type="password"`: Kotak teks untuk kata sandi.
    - `type="email"`: Untuk input email (dengan validasi dasar).
    - `type="submit"`: Tombol untuk mengirim formulir.
    - `type="radio"`: Pilihan tunggal.
    - `type="checkbox"`: Pilihan ganda.
  - `<textarea>`: Area teks multi-baris.
  - `<select>`: Daftar pilihan (dropdown).
  - `<label>`: Menambahkan label deskriptif untuk elemen input, meningkatkan aksesibilitas.
-

## Halaman 2: Gaya dan Interaktivitas

### 4. CSS (Cascading Style Sheets): Dasar & Layout

CSS digunakan untuk **mengatur tampilan dan tata letak** halaman web. Ia memisahkan presentasi (tampilan) dari struktur (HTML), membuat kode lebih rapi dan mudah dikelola.

- **Cara Menambahkan CSS:**
  1. **Inline:** Menggunakan atribut `style` langsung di tag HTML (tidak direkomendasikan).
  2. **Internal:** Menggunakan tag `<style>` di dalam `<head>` dokumen HTML.
  3. **Eksternal:** Menulis kode CSS di file terpisah (misal, `style.css`) dan menautkannya menggunakan tag `<link>` di `<head>`. (Praktik terbaik).
- **Sintaks Dasar:**

CSS

```
selektor {  
    properti: nilai;  
}  
  
/* Contoh */  
p {  
    color: blue;  
    font-size: 16px;  
}
```

- **Selektor Umum:**
  - **Elemen:** `p`, `h1`, `div`
  - **Class:** `.nama-kelas` (dapat digunakan berulang kali).
  - **ID:** `#id-unik` (hanya boleh ada satu di halaman).
- **Properti Layout Penting:**
  - **Box Model:** Setiap elemen HTML dianggap sebagai kotak dengan properti: `margin` (jarak luar), `border` (garis tepi), `padding` (jarak dalam), dan `content`.
  - **Display:** Mengatur bagaimana elemen ditampilkan. Nilai umum:
    - `block`: Mengambil seluruh lebar yang tersedia.
    - `inline`: Hanya mengambil lebar sesuai kontennya.
    - `flex` (Flexbox): Model layout satu dimensi untuk mengatur item dalam baris atau kolom. Sangat kuat untuk alignment dan distribusi ruang.
    - `grid`: Model layout dua dimensi (baris dan kolom), ideal untuk layout halaman yang kompleks.
  - **Position:** Mengatur posisi elemen (`static`, `relative`, `absolute`, `fixed`).

### 5. JavaScript: Dasar & OOP

JavaScript adalah bahasa pemrograman yang **menjadikan halaman web interaktif dan dinamis**. Kode JavaScript dieksekusi di sisi klien (browser).

- **Dasar-dasar:**

- **Variabel:** `let` (bisa diubah), `const` (konstan).
- **Typing Data:** `String`, `Number`, `Boolean`, `Object`, `Array`.
- **Operator:** Aritmatika (+, -, \*, /), Perbandingan (==, ===, !=, >), Logika (&&, ||, !).
- **Struktur Kontrol:** `if...else`, `switch`, `for`, `while`.
- **Fungsi:** Blok kode yang dapat digunakan kembali.

JavaScript

```
function sapa(nama) {  
    return "Halo, " + nama;  
}
```

- **Manipulasi DOM (Document Object Model):** JavaScript dapat mengubah konten dan struktur HTML secara dinamis.

JavaScript

```
// Mengubah teks elemen dengan id 'judul'  
document.getElementById('judul').innerHTML = "Judul Baru";
```

- **OOP (Object-Oriented Programming) di JavaScript:** JavaScript menggunakan *prototypes*, tetapi juga mendukung sintaks kelas (`class`) yang lebih familiar.
  - **Objek:** Kumpulan pasangan *key-value*.
  - **Class:** "Cetakan" untuk membuat objek.

JavaScript

```
class Mobil {  
    constructor(merk) {  
        this.merk = merk;  
    }  
  
    nyalakan() {  
        console.log(this.merk + " menyala!");  
    }  
}  
  
let avanza = new Mobil("Toyota Avanza");  
avanza.nyalakan(); // Output: Toyota Avanza menyala!
```

- **Standard Library:** JavaScript memiliki pustaka fungsi bawaan untuk berbagai keperluan, seperti:
  - **Math:** Untuk operasi matematika (`Math.random()`, `Math.floor()`).
  - **Date:** Untuk bekerja dengan tanggal dan waktu.
  - **JSON:** Untuk parsing dan mengubah data JSON (`JSON.parse()`, `JSON.stringify()`).
  - **Metode Array:** `forEach()`, `map()`, `filter()`, `reduce()`.

---

## Halaman 3: Framework dan Pengembangan Backend

### 6. Bootstrap

Bootstrap adalah **framework CSS, JS, dan HTML front-end yang paling populer**. Tujuannya adalah untuk mempercepat dan mempermudah pengembangan antarmuka web yang responsif (*mobile-first*).

- **Fitur Utama:**
  - **Sistem Grid (Grid System):** Fitur paling kuat dari Bootstrap. Membagi halaman menjadi 12 kolom, memungkinkan pembuatan layout yang kompleks dan responsif dengan mudah menggunakan kelas seperti `.container`, `.row`, dan `.col-md-6`.
  - **Komponen Siap Pakai:** Menyediakan berbagai komponen yang sudah dirancang dan fungsional, seperti:
    - Tombol (`.btn`)
    - Navigasi (`.navbar`)
    - Kartu (`.card`)
    - Modal (`.modal`)
    - Formulir (`.form-control`)
    - Carousel
  - **Utilitas (Utilities):** Kelas-kelas bantuan untuk melakukan tugas umum dengan cepat, seperti mengatur margin (`.m-1`), padding (`.p-1`), warna teks (`.text-primary`), dan lainnya.
- **Cara Menggunakan:** Cukup tambahkan file CSS dan JS Bootstrap ke dalam proyek Anda melalui CDN (Content Delivery Network) atau dengan mengunduhnya secara lokal.

#### HTML

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
```

### 7. PHP & MySQL: Studi Kasus Sistem Login & Registrasi

Kombinasi PHP dan MySQL adalah salah satu tumpukan teknologi paling umum untuk pengembangan web sisi server (*backend*).

- **PHP (Hypertext Preprocessor):**
  - Bahasa skrip yang berjalan di server.

- Tugas utamanya adalah memproses data formulir, berinteraksi dengan basis data, mengelola sesi pengguna, dan menghasilkan konten HTML secara dinamis sebelum dikirim ke browser.
- **MySQL:**
  - Sistem manajemen basis data relasional (RDBMS).
  - Digunakan untuk menyimpan, mengelola, dan mengambil data secara terstruktur dalam bentuk tabel.
- **Alur Kerja Sistem Login & Registrasi (berdasarkan referensi Petani Kode):**

#### 1. Struktur Proyek:

- `index.php`: Halaman utama setelah login.
- `login.php`: Halaman dengan formulir login.
- `register.php`: Halaman dengan formulir registrasi.
- `config.php`: File untuk koneksi ke database MySQL.
- `style.css`: File CSS (bisa digantikan/ditambah dengan Bootstrap).

#### 2. Proses Registrasi:

- Pengguna mengisi formulir registrasi (username, email, password) di `register.php`.
- Data dikirim ke server menggunakan metode `POST`.
- Skrip PHP menerima data tersebut.
- **Penting:** Kata sandi di-*hash* menggunakan fungsi seperti `password_hash()` untuk keamanan sebelum disimpan.
- Skrip PHP menjalankan query SQL `INSERT` untuk menyimpan data pengguna baru ke dalam tabel `users` di database MySQL.

#### 3. Proses Login:

- Pengguna memasukkan username dan password di `login.php`.
- Data dikirim ke server via `POST`.
- Skrip PHP melakukan query SQL `SELECT` untuk mencari pengguna berdasarkan username yang dimasukkan.
- Jika pengguna ditemukan, skrip menggunakan `password_verify()` untuk membandingkan password yang dimasukkan dengan *hash* yang tersimpan di database.
- Jika verifikasi berhasil, PHP akan membuat **sesi (session)** untuk pengguna tersebut (misalnya, `$_SESSION['user_id'] = ...`). Sesi ini berfungsi sebagai penanda bahwa pengguna telah berhasil login.
- Pengguna diarahkan (`redirect`) ke halaman `index.php`.

#### 4. Otentikasi Halaman:

- Di setiap halaman yang memerlukan login (seperti `index.php`), skrip PHP akan memeriksa apakah sesi pengguna ada dan valid. Jika tidak, pengguna akan diarahkan kembali ke halaman login.

#### 5. Logout:

- Proses logout akan menghancurkan sesi (`session_destroy()`) dan mengarahkan pengguna kembali ke halaman login.

Dengan mengintegrasikan **Bootstrap** pada sistem ini, formulir, tombol, dan keseluruhan tata letak halaman (login, register, dashboard) dapat dibuat lebih menarik dan responsif dengan cepat tanpa perlu menulis banyak kode CSS dari awal.