

Nama : Fahrezi Ahmad Syahyana

Nim : 20230140039_A

1. HTTP (Hypertext Transfer Protocol)

- **HTTP** adalah protokol komunikasi antara **client (browser)** dan **server**.
 - HTTP bekerja menggunakan metode seperti:
 - GET → untuk **mengambil data** (misal membuka halaman)
 - POST → untuk **mengirim data** (misal login/form)
 - Setiap permintaan HTTP menghasilkan **respons** dari server yang berisi:
 - Status (200 OK, 404 Not Found, 500 Error)
 - Header (info tentang data)
 - Body (isi data yang diminta)
-

2. HTML (HyperText Markup Language)

◆ Dasar HTML:

- HTML adalah **struktur dasar** halaman web.
- Elemen umum:

```
html
SalinEdit
<html>
  <head> <title>Judul</title> </head>
  <body>
    <h1>Halo Dunia</h1>
    <p>Paragraf</p>
  </body>
</html>
```

◆ HTML Form:

Digunakan untuk mengambil input dari pengguna.

```
html
SalinEdit
<form action="proses.php" method="POST">
  <input type="text" name="username">
  <input type="password" name="password">
  <input type="submit" value="Login">
</form>
```

3. CSS (Cascading Style Sheets)

◆ Dasar CSS:

- CSS mengatur **tampilan/style** HTML.
- Dapat ditulis di:
 - Tag `<style>`, file `.css`, atau inline.

```
css
SalinEdit
body {
  background-color: #f0f0f0;
  color: #333;
}
```

◆ Layout CSS:

- Properti layout penting:
 - `display: flex;` → untuk menyusun elemen sejajar
 - `grid, position, margin, padding`
- Contoh:

```
css
SalinEdit
.container {
  display: flex;
  justify-content: space-between;
}
```

4. JavaScript (JS)

◆ Dasar JS:

- Bahasa pemrograman **klien-side** (berjalan di browser).

```
js
SalinEdit
let nama = "Akbar";
console.log("Halo, " + nama);
```

◆ OOP JS:

Object-Oriented Programming dalam JS:

```
js
SalinEdit
class Mahasiswa {
  constructor(nama) {
    this.nama = nama;
  }
  sapa() {
    return "Halo " + this.nama;
  }
}
```

```
let m1 = new Mahasiswa("Rina");
console.log(m1.sapa());
```

◆ **Standard Library JS:**

- Fungsi umum: `alert()`, `prompt()`, `Math`, `Date`, `JSON.parse()`, dll.
 - DOM Manipulation: `document.getElementById()`, `addEventListener()`
-

5. Bootstrap 5.3

([Link Referensi](#))

◆ **Bootstrap Dasar:**

- Framework CSS untuk membuat tampilan responsif dan cepat.
- Cukup import:

```
html
SalinEdit
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.c
ss" rel="stylesheet">
```

◆ **Contoh Komponen:**

```
html
SalinEdit
<button class="btn btn-primary">Klik Saya</button>
<div class="alert alert-success">Berhasil!</div>
```

◆ **Layout Bootstrap:**

- Sistem grid 12 kolom:

```
html
SalinEdit
<div class="container">
  <div class="row">
    <div class="col-md-6">Kolom 1</div>
    <div class="col-md-6">Kolom 2</div>
  </div>
</div>
```

6. PHP + Bootstrap + MySQL

([Referensi: Petani Kode Login Register](#))

◆ **PHP Dasar:**

- PHP adalah bahasa skrip **server-side**.
- Digunakan untuk proses data form, interaksi database, autentikasi.

```
php
SalinEdit
<?php
$nama = $_POST['nama'];
echo "Halo $nama";
?>
```

◆ Koneksi ke MySQL:

```
php
SalinEdit
$koneksi = mysqli_connect("localhost", "root", "", "db_user");
```

◆ Sistem Login/Register (Ringkasan Alur):

1. **Register:** user isi form → data disimpan di database (**INSERT**)
2. **Login:** user isi username/password → dicek dengan database (**SELECT**)
3. **Jika cocok** → redirect ke halaman dashboard
4. Menggunakan session untuk menyimpan status login:

```
php
SalinEdit
session_start();
$_SESSION['username'] = $username;
```

◆ Bootstrap Digunakan untuk:

- Form yang responsif
- Alert/error
- Layout halaman login

Penutup

Dengan memadukan:

- **HTML + CSS + JS** → tampilan dan interaktivitas di browser
- **PHP + MySQL** → pengolahan data dan penyimpanan di server
- **Bootstrap** → mempercepat pembuatan tampilan
- **HTTP** → sebagai dasar komunikasi client-server