

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB**



Disusun Oleh :
Martonius Epta Sidabutar
NIM : 202312051

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI TEKNOLOGI BONTANG
TAHUN 2025**

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh. Puji dan syukur penulis panjatkan kehadirat Allah SWT, yang telah melimpahkan rahmat dan Hidayah-Nya lah sehingga laporan praktikum sistem basis data ini dapat terselesaikan, tetapi masih sangat jauh dari kesempurnaan.

Penulisan laporan praktikum ini dimaksudkan sebagai tolak ukur Saya dalam mencapai pemahaman beserta pembahasan mengenai Program dari praktikum Pemrograman Dasar kali ini, disamping itu penulis berharap dalam penulisan laporan praktikum ini dapat bermanfaat bagi penulis, pembaca, khususnya mahasiswa Sekolah Tinggi Teknologi Bontang dan masyarakat umumnya.

Bontang, 6 Juni 2025



Martonius Epta Sidabutar

DAFTAR ISI

[Daftar Isi](#)

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
LEMBAR ASISTENSI	v
Modul 1. Pemrograman WEB HTML	1
A. Tujuan	1
B. Alat dan Bahan	1
C. Landasan Teori	1
D. Kegiatan Praktikum	3
Modul 2. Pengenalan CSS (Cascading Style Sheet).....	29
A. Tujuan.....	29
B. Teori Dasar CSS	29
C. Pratikum.....	29
D. Tugas Praktikum.....	49
Modul 3. Pengenalan CSS (Cascading Style Sheet).....	55
A. Tujuan.....	55
B. Teori Dasar Bootsrap.....	55
C. Pratikum.....	55
D. Tugas Praktikum.....	86
Modul 4. Javascript	95
A. Tujuan.....	95
B. Teori Dasar Bootsrap.....	95
C. Pratikum.....	95
D. Tugas Akhir Praktikum: Mini E-Comerce Product Page	128
Modul 5. Dasar-dasar PHP	132
A. Tujuan	132
B. Teori Dasar PHP	132

C. Praktikum.....	132
Analisis Singkat Kode PHP	145
Kelebihan Kode:	145
Kekurangan:	146
Kesimpulan:.....	146
D. Tugas Praktikum.....	160
Modul 6. Pemrograman Web dengan PHP dan MySQL.....	166
A. Tujuan.....	166
B. Teori Dasar PHP dan MySQL.....	166
C. Pratikum.....	166
D. Tugas Praktikum.....	178

LEMBAR ASISTENSI

Modul 1. Pemrograman WEB HTML

A. Tujuan

Setelah mengikuti praktikum, Mahasiswa diharapkan mampu:

1. Setelah mengikuti praktikum ini mahasiswa mampu memahami struktur dokumen html
2. Mahasiswa mampu membuat web sederhana dengan menggunakan HTML.

B. Alat dan Bahan

Peralatan dan bahan yang dibutuhkan untuk praktikum ini adalah sebagai berikut:

1. Hardware (Laptop HP)
2. Software:
 - a. Notepad++
 - b. Web Brower (Google Chrome)

C. Landasan Teori

HTML (HyperText Markup Language) adalah bahasa markup standar yang digunakan untuk membuat dan menyusun halaman web di internet. HTML bukan bahasa pemrograman melainkan bahasa markup yang menggunakan tag-tag untuk memberi struktur dan makna pada konten web.

Mengapa HTML Bukan Bahasa Pemrograman?

1. HTML tidak memiliki logika pemrograman seperti kondisi (if-else), perulangan (for, while), atau fungsi.
2. HTML hanya digunakan untuk menyusun dan menampilkan konten pada halaman web.

Apa Peran HTML dalam Pengembangan Web?

1. HTML mendefinisikan struktur halaman web seperti teks, gambar, tabel, dan formulir.
2. Digunakan bersama dengan CSS (untuk styling) dan JavaScript (untuk interaktivitas).

Sejarah HTML

1. Awal Mula HTML (1989–1995)

- a. 1989 – Tim Berners-Lee seorang fisikawan di CERN yang mengusulkan sistem berbasis internet untuk berbagi dokumen.
 - b. 1990 – Berners-Lee menciptakan prototipe pertama HTML dan browser WorldWideWeb di komputer NeXT.
 - c. 1991 – HTML pertama kali diperkenalkan ke public yang memungkinkan penggunaan hyperlinked text.
 - d. 1993 – Mosaic Browser memperkenalkan tag , merevolusi tampilan visual web.
 - e. 1994 – W3C (World Wide Web Consortium) didirikan untuk standarisasi teknologi web.
 - f. 1995 – HTML 2.0 dirilis sebagai standar resmi pertama.
2. Perkembangan HTML dan Persaingan Browser (1995–2000)
 - a. 1995 – Internet Explorer pertama kali diperkenalkan yang memicu persaingan dengan Netscape.
 - b. 1997 – HTML 3.2 dirilis dengan menambahkan dukungan untuk tabel, Apple Java, dan kontrol teks.
 - c. 1999 – HTML 4.01 menjadi standar utama yang mengintegrasikan CSS, fitur aksesibilitas, dan internasionalisasi.
 3. Era XML dan XHTML (2000–2006)
 - a. 2000 – XHTML 1.0 diperkenalkan sebagai versi HTML yang lebih ketat dan berbasis XML.
 - b. 2004 – Kelompok WHATWG dibentuk untuk mengembangkan HTML5 yang menentang keterbatasan XHTML.
 4. Revolusi HTML5 (2008–Sekarang)
 - a. 2008 – Draft pertama HTML5 dirilis yang menekankan pada multimedia, semantik, dan API.
 - b. 2014 – HTML5 menjadi rekomendasi resmi W3C dengan membawa fitur seperti <video> dan <audio>.
 - c. 2016–2017 – HTML5.1 dan HTML5.2 memperkenalkan peningkatan seperti penyimpanan offline dan aksesibilitas.
 - d. 2019 – W3C dan WHATWG menjadikan HTML sebagai "living standard" yang terus diperbarui.
 5. Fitur Modern HTML

- a. Elemen Semantik seperti <header>, <article>, <nav> untuk SEO dan aksesibilitas.
- b. Dukungan Multimedia, <video> dan <audio> tanpa perlu plugin tambahan.
- c. API Canggih yaitu Geolocation, drag-and-drop, dan Web Workers untuk aplikasi dinamis.
- d. Desain Mobile-First seperti gambar responsif dan meta viewport untuk tampilan optimal di perangkat mobile.

Struktur Dasar HTML

Dokumen HTML terdiri dari elemen-elemen yang ditandai dengan tag, yang biasanya terdiri dari tag pembuka dan penutup. Struktur dasar dokumen HTML meliputi:

- 1. <!DOCTYPE html> : Mendefinisikan tipe dokumen sebagai HTML5.
- 2. <html> : Elemen root yang membungkus seluruh isi halaman.
- 3. <head> : Bagian yang berisi metadata, judul halaman, dan link ke sumber daya eksternal.
- 4. <body> : Bagian yang berisi konten utama yang akan ditampilkan di browser.

Manfaat HTML

- 1. Membuat struktur halaman web yang terorganisir.
- 2. Memungkinkan penyisipan teks, gambar, video, dan elemen multimedia lainnya.
- 3. Dasar dari semua halaman web yang dapat diakses melalui browser.
- 4. Memudahkan pengembangan web dengan berbagai teknologi pendukung seperti CSS dan JavaScript.

D. Kegiatan Praktikum

Kegiatan praktikum Pemrograman Web

1. Pengenalan HTML dan Struktur dasar Dokumen

Penjelasan

HTML adalah bahasa markup yang menggunakan tag untuk membuat elemen elemen pada halaman web. Setiap elemen memiliki fungsi tertentu seperti heading, paragraf, gambar, dan lain-lain

Strukur Dasar HTML

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <title>Judul Halaman</title>
5. </head>
6. <body>
7.   <h1>Halo, Saya Abadi Nugroho!</h1>
8.   <p>Ini adalah paragraf pertama saya.</p>
9. </body>
10. </html>
```

Latihan Pratikum

- Buatlah halaman HTML sederhana dengan judul “Halaman Pertama Saya”.
- Tambahkan Heading `<h1>` dan paragraph `<p>` sesuai contoh diatas.
- Berikan *capture* (*screenshoot/tangkapan layar*) dan analisis dari hasil pratikum anda di laporan.

Jawab

Berikut HTMLnya:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Modul pratikum pertama</title>
  </head>
  <body>
    <h1>Halo, Saya Martonius Epta Sidabutar!</h1>
    <h2>Ini Heading h2</h2>
    <h3>Ini Heading h3</h3>
    <p>Ini adalah paragraf pertama Saya.</p>
  </body>
</html>
```

Berikut hasil html dibuka dengan web browser

Halo, Saya Martonius Epta Sidabutaa

Ini Heading h2

Ini Heading h3

Ini adalah paragraf pertama Saya.

Analisis: Halaman ini sederhana dan mudah dipahami, cocok untuk pemula yang baru belajar HTML.

2. Elemen HTML Dasar

Penjelasan

Elemen dasar HTML meliputi heading (<h1> sampai <h6>), paragraf (<p>), link (<a>), gambar (), dan daftar (, ,).

Contoh Kode

```
1. <h2>Heading Level 2</h2>
2. <p>Ini adalah paragraf dengan <a href="https://www.stitek.ac.id">link ke website stitek</a></p>
3. 
4. <ul>
5.   <li>Item pertama</li>
6.   <li>Item kedua</li>
7. </ul>
```

Latihan Pratikum

- Buat halaman dengan heading, paragraph yang mengandung link, gambar dengan atribut alt, dan daftar tidak berurutan.
- Berikan *capture* (screenshot/tangkapan layar) dan analisis dari hasil praktikum anda di laporan.

Jawab:

Kode HTML:

```
<!DOCTYPE html>
<html>
  <head>
```

```

<title>Halaman Praktikum HTML</title>
</head>
<body>
    <h2>Heading Level 2</h2>
    <p>Ini adalah paragraf dengan <a href="https://www.stitek.ac.id">link ke website stitek</a></p>
    

    <ul>
        <li>Item pertama</li>
        <li>Item kedua</li>
    </ul>
</body>
</html>

```

Berikut tampilannya:



Berikut jika link dibuka:



Analisis: Laporan dapat mencakup hal-hal berikut:

- Heading, paragraf, gambar, dan daftar tampil dengan baik sesuai fungsinya.
- Link dapat diklik dan mengarah ke situs yang benar.
- Gambar ditampilkan dengan deskripsi alternatif yang berguna.

3. ATRIBUT HTML

Penjelasan

Atribut memberikan informasi tambahan pada elemen HTML, seperti href pada tag <a>, src dan alt pada , serta title untuk tooltip

Contoh code

```
1. <a href="https://www.stitek.ac.id" title="Pergi ke stitek">STITEK</a>
2. 
```

Latihan Praktikum

- Buat paragraf dengan berbagai format teks menggunakan tag di atas.
- Berikan *capture* (*screenshoot/tangkapan layar*) dan analisis dari hasil praktikum anda di laporan.

Jawab:

Berikut kode HTML:

```
<!DOCTYPE html>
<html>
<head>
    <title>Halaman Praktikum HTML dengan Atribut Title</title>
</head>
```

```

<body>
    <h2>Heading Level 2</h2>
    <p>Ini adalah paragraf dengan <a href="https://www.stitek.ac.id"
        title="Pergi ke stitek">link ke website stitek</a></p>
    

    <ul>
        <li>Item pertama</li>
        <li>Item kedua</li>
    </ul>
</body>
</html>

```

Berikut jika kode HTML dijalankan di web browser:



Analisis: Laporan dapat mencakup hal-hal berikut:

- Atribut title berhasil ditambahkan pada link dan gambar.
- Tooltip yang muncul saat kursor diarahkan ke elemen (link atau gambar) sudah berfungsi dengan baik.

4. Format Teks

Penjelasan

Tag untuk memformat teks seperti **** (bold), *<i>* (italic), <u> (underline), ****,

dan .

Contoh Kode

```
1. <p>Ini adalah teks <b>tebal</b>, <i>miring</i>, dan <u>garis bawah</u>.</p>
2. <p><strong>Penting!</strong> Ini adalah teks penting.</p>
```

Latihan Praktikum

- Buat paragraf dengan berbagai format teks menggunakan tag di atas.
- Berikan *capture* (*screenshot/tangkapan layar*) dan analisis dari hasil praktikum anda di laporan.

Berikut kode HTML:

```
<!DOCTYPE html>

<html>
  <head>
    <title>Halaman Praktikum Format Teks</title>
  </head>
  <body>
    <h2>Format Teks dalam HTML</h2>
    <p>Ini adalah teks <b>tebal</b>, <i>miring</i>, dan <u>garis bawah</u>.</p>
    <p><strong>Penting!</strong> Ini adalah teks yang ditekankan pentingnya.</p>
    <p>Ini adalah teks yang <em>ditonjolkan</em> dengan tag <em>.</p>
  </body>
</html>
```

Berikut hasil kode HTML dijalankan di web browser:

Format Teks dalam HTML

Ini adalah teks **tebal**, *miring*, dan garis bawah.

Penting! Ini adalah teks yang ditekankan pentingnya.

Ini adalah teks yang *ditonjolkan* dengan tag .

Analisis: Laporan dapat mencakup hal-hal berikut:

- Semua format teks (tebal, miring, garis bawah, penting, dan tonjolan) diterapkan dengan benar.
- Teks tampil sesuai dengan format yang diinginkan.

5. Link dan Gambar

Penjelasan

Link dibuat dengan tak <a> dan gambar dengan tak . Atribut penting adalah href untuk link dan src serta alt untuk gambar

Contoh Kode

```
1. <a href="https://www.stitek.ac.id">Kampus Stitek</a>
2. 
```

Latihan Praktikum

- Buat halaman dengan beberapa link dan gambar yang berfungsi.
- Berikan *capture* (*screenshoot/tangkapan layar*) dan analisis dari hasil praktikum anda di laporan.

Jawab:

Berikut code HTML:

```
<!DOCTYPE html>
<html>
<head>
    <title>Halaman Link dan Gambar</title>
</head>
<body>
    <h2>Halaman dengan Link dan Gambar</h2>
```

```

<!-- Link ke Kampus STITEK -->
<p>Ini adalah link ke <a href="https://www.stitek.ac.id"
target="_blank">Kampus Stitek</a>.</p>

<!-- Gambar dengan deskripsi -->
<p></p>

<!-- Link ke Gambar -->
<p>Ini adalah link yang akan membuka gambar: <a href="D:\Marton\Kuliah S1 Teknik Informatika\Logo\LOGO_STITEK2.PNG" target="_blank">Lihat Foto Stitek</a>.</p>

</body>
</html>

```

Berikut Kode HTML dijalankan di web browser:

Halaman dengan Link dan Gambar

Ini adalah link ke [Kampus Stitek](#).



Ini adalah link yang akan membuka gambar: [Lihat Foto Stitek](#).

Ketika klik link “Kampus Stitek”



Ketika klik link “Lihat Foto Stitek:



Analisis: Laporan dapat mencakup hal-hal berikut:

- Semua *link* berfungsi dengan baik, membuka halaman yang benar atau gambar.
- Gambar ditampilkan dengan benar dengan deskripsi alternatif yang muncul jika gambar tidak dapat dimuat.

6. Daftar (List)

Penejelasan

Daftar berurutan menggunakan , daftar tidak berurutan menggunakan , dan

daftar definisi menggunakan <dl>, <dt>, dan <dd>.

Contoh Kode

```

1. <ol>
2.   <li>Langkah pertama</li>
3.   <li>Langkah kedua</li>
4. </ol>
5.
6. <ul>
7.   <li>Buah</li>
8.   <li>Sayur</li>
9. </ul>
10.
11. <dl>
12.   <dt>HTML</dt>
13.   <dd>Bahasa markup untuk membuat halaman web.</dd>
14. </dl>

```

Latihan Praktikum

- Buat halaman dengan ketiga jenis daftar tersebut
- Berikan *capture* (*screenshoot/tangkapan layar*) dan analisis dari hasil praktikum anda di laporan.

Jawab:

Berikut kode HTML:

```

- <!DOCTYPE html>
- <html>
- <head>
-   <title>Halaman Daftar Berurutan, Tidak Berurutan, dan Definisi</title>
- </head>
- <body>
-   <h2>Daftar Berurutan, Tidak Berurutan, dan Definisi</h2>
-
-   <!-- Daftar Berurutan -->
-   <h3>Daftar Berurutan (Ordered List)</h3>
-   <ol>
-     <li>Langkah pertama</li>
-     <li>Langkah kedua</li>
-     <li>Langkah ketiga</li>
-   </ol>

```

```
-      - <!-- Daftar Tidak Berurutan -->
-      - <h3>Daftar Tidak Berurutan (Unordered List)</h3>
-      - <ul>
-          - <li>Buah</li>
-          - <li>Sayur</li>
-          - <li>Rempah</li>
-      - </ul>
-
-      - <!-- Daftar Definisi -->
-      - <h3>Daftar Definisi (Definition List)</h3>
-      - <dl>
-          - <dt>HTML</dt>
-          - <dd>Bahasa markup untuk membuat halaman web.</dd>
-          - <dt>CSS</dt>
-          - <dd>Bahasa untuk menggaya elemen HTML.</dd>
-          - <dt>JavaScript</dt>
-          - <dd>Bahasa pemrograman untuk membuat halaman web interaktif.</dd>
-      - </dl>
-      - </body>
-      - </html>
```

Berikut Kode HTML dibuka di web browser:

Daftar Berurutan, Tidak Berurutan, dan Definisi

Daftar Berurutan (Ordered List)

1. Langkah pertama
2. Langkah kedua
3. Langkah ketiga

Daftar Tidak Berurutan (Unordered List)

- Buah
- Sayur
- Rempah

Daftar Definisi (Definition List)

HTML

Bahasa markup untuk membuat halaman web.

CSS

Bahasa untuk menggaya elemen HTML.

JavaScript

Bahasa pemrograman untuk membuat halaman web interaktif.

Analisis: Laporan dapat mencakup hal-hal berikut:

- Semua jenis daftar (berurutan, tidak berurutan, dan definisi) diterapkan dengan benar.
- Daftar berurutan menampilkan nomor secara otomatis.
- Daftar tidak berurutan menggunakan tanda bullet.
- Daftar definisi menampilkan istilah dan deskripsi dengan format yang tepat.

7. Tabel HTML

Penejelasan

Tabel dibuat dengan tag `<table>`, baris dengan `<tr>`, header dengan `<th>`, dan data dengan `<td>`.

Contoh Kode

```

1. <table border="1">
2.   <tr>
3.     <th>Nama</th>
4.     <th>Umur</th>
5.   </tr>
6.   <tr>

7.     <td>Abadi</td>
8.     <td>20</td>
9.   </tr>
10.  <tr>
11.    <td>Nugroho</td>
12.    <td>22</td>
13.  </tr>
14. </table>

```

Latihan Pratikum

- Buat tabel data sederhana dengan minimal 3 baris dan 2 kolom.
- Berikan *capture* (*screenshot/tangkapan layar*) dan analisis dari hasil praktikum anda di laporan.

Jawab:

Berikut Kode HTML:

```

- <!DOCTYPE html>
- <html>
- <head>
-   <title>Tabel Data Sederhana</title>
- </head>
- <body>
-   <h2>Tabel Data Sederhana</h2>
-
-   <table border="1">
-     <tr>
-       <th>Nama</th>
-       <th>Umur</th>
-     </tr>
-     <tr>
-       <td>Abadi</td>
-       <td>20</td>
-     </tr>

```

```

-      <tr>
-          <td>Nugroho</td>
-          <td>22</td>
-      </tr>
-      <tr>
-          <td>Rina</td>
-          <td>25</td>
-      </tr>
-  </table>
-  </body>
- </html>

```

Berikut Kode HTML dijalankan di Web Browser

Tabel Data Sederhana

Nama	Umur
Abadi	20
Nugroho	22
Rina	25

Analisis: Laporan dapat mencakup hal-hal berikut:

- Tabel berhasil dibuat dengan benar, menampilkan header "Nama" dan "Umur".
- Data ditampilkan dalam tiga baris dengan dua kolom.
- Garis batas terlihat di sekitar tabel dan sel-selnya.

8. Formulir HTML

Penjelasan

Formulir digunakan untuk mengumpulkan data dari pengguna. Elemen penting :

<form>,

<input>, <textarea>, <select>, <button>.

Contoh Kode

```
1. <form action="/submit" method="post">
2.   <label for="nama">Nama:</label>
3.   <input type="text" id="nama" name="nama"><br><br>
4.
5.   <label for="email">Email:</label>
6.   <input type="email" id="email" name="email"><br><br>
7.
8.   <input type="submit" value="Kirim">
9. </form>
10.
```

Latihan Praktikum

- Buat Formulir pendaftaran dengan input teks, email, dan tombol submit.
- Berikan *capture* (*screenshot/tangkapan layar*) dan analisis dari hasil praktikum anda di laporan.

Jawab:

Berikut kode HTML:

```
<!DOCTYPE html>

<html>

<head>

<title>Formulir Pendaftaran</title>

</head>

<body>

<h2>Formulir Pendaftaran</h2>

<form action="/submit" method="post">

<label for="nama">Nama:</label>
```

```
<input type="text" id="nama" name="nama" required><br><br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required><br><br>

<label for="alamat">Alamat:</label>

<textarea      id="alamat"      name="alamat"      rows="4"      cols="50"
required></textarea><br><br>

<input type="submit" value="Kirim">

</form>

</body>

</html>
```

Berikut Kode HTML dijalankan di web Browser:

Formulir Pendaftaran

Nama:

Email:

Alamat:

Analisa:

- Formulir berhasil dibuat dengan input teks untuk nama, input email untuk email, dan area teks untuk alamat.
- Formulir dapat disubmit dengan tombol "Kirim" yang mengirimkan data formulir (meskipun pada aplikasi nyata, URL /submit harus dikonfigurasi di server).

9. Multimedia

Penjelasan

HTML5 mendukung penyisipan video dan audio dengan tag <video> dan <audio>.

Contoh

Kode

```
1. <video width="320" height="240" controls>
2.   <source src="movie.mp4" type="video/mp4">
3.   Browser Anda tidak mendukung tag video.

4. </video>
5.
6. <audio controls>
7.   <source src="audio.mp3" type="audio/mpeg">
8.   Browser Anda tidak mendukung tag audio.
9. </audio>
10.
```

Latihan Praktikum

- Buat halaman yang menampilkan video dan audio yang dapat diputar.
- Berikan *capture* (*screenshoot/tangkapan layar*) dan analisis dari hasil praktikum anda di laporan.

Jawab:

Kode HTML:

```
<!DOCTYPE html>
<html>
<head>
    <title>Halaman Video dan Audio</title>
</head>
<body>
    <h2>Video dan Audio yang Dapat Diputar</h2>

    <!-- Video -->
    <h3>Video</h3>
    <video width="640" height="360" controls>
        <source src="D:\Captures\sukses.mp4" type="video/mp4">
        Browser Anda tidak mendukung tag video.
    </video>
    <br><br>

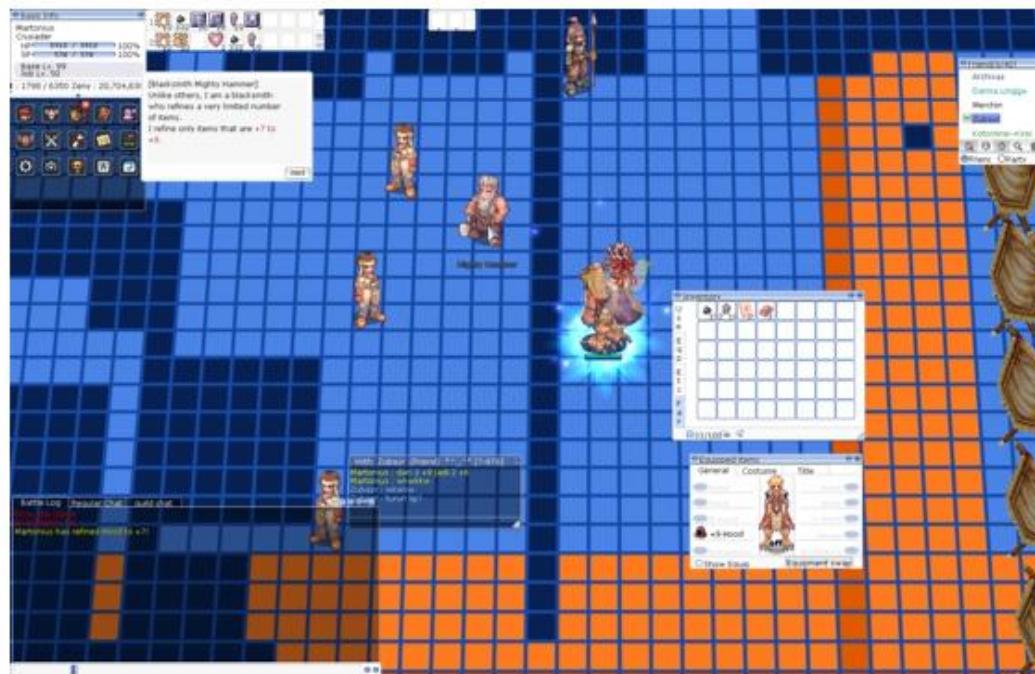
    <!-- Audio -->
    <h3>Audio</h3>
    <audio controls>
        <source src="C:\Users\Asus\Music\Suara 007.m4a"" type="audio/mpeg">
        Browser Anda tidak mendukung tag audio.
    </audio>

</body>
</html>
```

Kode HTML dibuka di web browser:

Video dan Audio yang Dapat Diputar

Video



Audio

|| 0:03 / 20:00

Analisa:

- Video dan audio ditampilkan dengan benar.
- Kontrol video dan audio berfungsi sebagaimana mestinya.

C. Tugas Pratikum

Buatlah sebuah halaman web HTML yang berisi **Formulir Pendaftaran Mahasiswa Baru** dengan ketentuan sebagai berikut:

1. Struktur Halaman
 - a. Gunakan struktur HTML5 standar (`<!DOCTYPE html>`, `<html>`, `<head>`, `<body>`).
 - b. Berikan judul halaman yang sesuai, misalnya "Formulir Pendaftaran Mahasiswa Baru".
 - c. Tambahkan heading utama `<h1>` yang berisi judul halaman.
2. Formulir Pendaftaran

Buat formulir dengan elemen-elemen berikut:

Elemen Formulir	Detail
Input teks	- Nama Lengkap (wajib diisi) - Email (wajib diisi dan bertipe email)
Radio button	Pilihan jenis kelamin : Pria atau Wanita (wajib dipilih)
Checkbox	Pilihan hobi (minimal 2 pilihan), misalnya Membaca, Olahraga, Musik, dll.
Dropdown (select)	Pilihan jurusan (minimal 3 pilihan), misalnya Teknik Informatika, Manajemen, Akuntansi
Textarea	Kolom untuk alamat lengkap (opsional)
Tombol submit	Tombol untuk mengirim data formulir

3. Tata Letak dan Keterangan
 - a. Setiap elemen form harus memiliki label yang jelas menggunakan tag `<label>` dan atribut `for` yang sesuai.
 - b. Gunakan `<fieldset>` dan `<legend>` untuk mengelompokkan bagian-bagian form yang relevan (misalnya data pribadi, hobi, jurusan).
 - c. Tambahkan placeholder pada input teks untuk memberikan contoh pengisian.
 - d. Gunakan atribut `required` pada input yang wajib diisi.
 - e. Tambahkan komentar HTML untuk menjelaskan bagian-bagian penting dalam kode.
4. Validasi dan Aksesibilitas
 - a. Pastikan form dapat divalidasi secara sederhana menggunakan atribut HTML (misalnya `required`, `type="email"`).
 - b. Gunakan atribut `name` yang sesuai untuk setiap input agar data dapat diproses dengan baik.
 - c. Pastikan form mudah dibaca dan diakses.
5. Tambahan
 - a. Tambahkan elemen multimedia seperti gambar logo kampus di atas formulir.
 - b. Tambahkan link ke halaman informasi kampus di bawah formulir.
 - c. Tambahkan style sederhana menggunakan atribut inline (misalnya warna latar belakang, margin).
 - d. Simpan kode di github anda masing-masing.
 - e. Buatlah **video** untuk menjelaskan script dari tugas yang anda buat, simpan di penyimpanan online misalnya youtube anda masing-masing dan sertakan link github dan video di laporan anda.

Jawab:

Berikut adalah contoh halaman HTML untuk **Formulir Pendaftaran Mahasiswa Baru** dengan memenuhi ketentuan yang diberikan. Saya juga akan memberikan penjelasan untuk setiap bagian penting dalam kode.

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Formulir Pendaftaran Mahasiswa Baru</title>
</head>
<body style="font-family: Arial, sans-serif; background-color: #f0f0f0; margin: 0; padding: 20px;">

    <!-- Gambar Logo Kampus -->
    

    <h1 style="text-align: center; color: #2c3e50;">Formulir Pendaftaran Mahasiswa Baru</h1>

    <!-- Formulir Pendaftaran -->
    <form action="/submit" method="post" style="background-color: white; padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);">

        <!-- Bagian Data Pribadi -->
        <fieldset style="border: 1px solid #ddd; padding: 10px; margin-bottom: 20px;">
            <legend style="font-weight: bold;">Data Pribadi</legend>

            <!-- Nama Lengkap -->
            <label for="nama" style="display: block; margin-bottom: 5px;">Nama Lengkap:</label>
```

```

<input type="text" id="nama" name="nama" placeholder="Masukkan Nama Lengkap" required style="width: 100%; padding: 8px; margin-bottom: 10px;">

<!-- Email -->
<label for="email" style="display: block; margin-bottom: 5px;">Email:</label>
<input type="email" id="email" name="email" placeholder="Masukkan Email" required style="width: 100%; padding: 8px; margin-bottom: 10px;">

<!-- Jenis Kelamin -->
<label>Jenis Kelamin:</label><br>
<input type="radio" id="pria" name="jenis_kelamin" value="Pria" required>
<label for="pria">Pria</label>
<input type="radio" id="wanita" name="jenis_kelamin" value="Wanita" required>
<label for="wanita">Wanita</label>
</fieldset>

<!-- Bagian Hobi -->
<fieldset style="border: 1px solid #ddd; padding: 10px; margin-bottom: 20px;">
    <legend style="font-weight: bold;">Hobi</legend>
    <label for="hobi" style="display: block; margin-bottom: 5px;">Pilih Hobi (minimal 2):</label>
    <input type="checkbox" id="membaca" name="hobi" value="Membaca">
    <label for="membaca">Membaca</label><br>
    <input type="checkbox" id="olahraga" name="hobi" value="Olahraga">
    <label for="olahraga">Olahraga</label><br>
    <input type="checkbox" id="musik" name="hobi" value="Musik">
    <label for="musik">Musik</label><br>
    <input type="checkbox" id="melukis" name="hobi" value="Melukis">
    <label for="melukis">Melukis</label>
</fieldset>

<!-- Bagian Jurusan -->

```

```

<fieldset style="border: 1px solid #ddd; padding: 10px; margin-bottom: 20px;">
    <legend style="font-weight: bold;">Pilihan Jurusan</legend>
    <label for="jurusan" style="display: block; margin-bottom: 5px;">Pilih Jurusan:</label>
        <select id="jurusan" name="jurusan" required style="width: 100%; padding: 8px; margin-bottom: 10px;">
            <option value="Teknik Informatika">Teknik Informatika</option>
            <option value="Manajemen">Manajemen</option>
            <option value="Akuntansi">Akuntansi</option>
        </select>
    </fieldset>

    <!-- Bagian Alamat -->
    <fieldset style="border: 1px solid #ddd; padding: 10px; margin-bottom: 20px;">
        <legend style="font-weight: bold;">Alamat</legend>
        <label for="alamat" style="display: block; margin-bottom: 5px;">Alamat Lengkap:</label>
            <textarea id="alamat" name="alamat" placeholder="Masukkan alamat lengkap" style="width: 100%; padding: 8px; height: 100px;"></textarea>
    </fieldset>

    <!-- Tombol Submit -->
    <button type="submit" style="background-color: #4CAF50; color: white; padding: 10px 20px; border: none; border-radius: 5px; cursor: pointer;">
        Kirim Pendaftaran
    </button>
</form>

    <!-- Link ke Halaman Informasi Kampus -->
    <p style="text-align: center; margin-top: 20px;">
        <a href="https://www.kampus.com/informasi" style="color: #3498db;">Klik di sini untuk informasi lebih lanjut tentang kampus</a>
    </p>

```

```
</body>  
</html>
```

Hasil Kode HTML dijalankan di Web Browser

The screenshot shows a web page titled "Formulir Pendaftaran Mahasiswa Baru". At the top right is the STITEK logo with the text "BERGASAH TRIGA TEKNOLOGI BONTANG". The form is divided into several sections:

- Data Pribadi**: Fields for "Nama Lengkap" (with placeholder "Masukkan Nama Lengkap") and "Email" (with placeholder "Masukkan Email"). Below these are gender options: "Jenis Kelamin" with radio buttons for "Pria" and "Wanita".
- Hobi**: A section titled "Pilih Hobi (minimal 2)" with checkboxes for "Membaca", "Olahraga", "Musik", and "Melukis".
- Pilihan Jurusan**: A dropdown menu labeled "Pilih Jurusan" with the option "Teknik Informatika" selected.
- Alamat**: A text area labeled "Alamat Lengkap" with placeholder "Masukkan alamat lengkap".

Penjelasan Bagian-bagian Penting:

1. Struktur Halaman:

- Menggunakan struktur HTML5 standar dengan elemen seperti <html>, <head>, dan <body>.
- Halaman diberi judul "Formulir Pendaftaran Mahasiswa Baru" dan heading utama <h1> di bagian atas.

2. Formulir Pendaftaran:

- Formulir dibagi menjadi beberapa bagian menggunakan <fieldset> dan <legend> untuk kejelasan.
- Setiap elemen input seperti teks, radio button, checkbox, dan dropdown disertai label yang sesuai untuk aksesibilitas.
- Elemen required digunakan untuk memastikan input yang penting wajib diisi.
- Placeholder ditambahkan pada input teks untuk memberikan contoh pengisian.

3. Tata Letak dan Gaya:

- Menggunakan inline CSS di dalam atribut style untuk memberikan warna latar belakang, margin, dan padding pada elemen.

- Logo kampus ditambahkan di atas formulir untuk mempercantik halaman.
- Formulir memiliki latar belakang putih dengan bayangan ringan untuk efek desain yang lebih menarik.

4. Validasi dan Aksesibilitas:

- Menggunakan atribut required untuk validasi input yang wajib diisi.
- Elemen label digunakan untuk memberikan keterangan yang jelas pada setiap input.

5. Tambahan:

- Tautan ke halaman informasi kampus ditambahkan di bawah formulir.
- Anda dapat menambahkan gambar logo kampus dengan mengganti URL gambar pada elemen .
- Simpan file ini ke GitHub dan buat video penjelasan sesuai dengan instruksi.

Modul 2. Pengenalan CSS (Cascading Style Sheet)

A. Tujuan

Setelah mengikuti praktikum, Mahasiswa diharapkan mampu:

1. Mampu memahami dasar-dasar CSS.
2. Mampu mengenal lebih jauh properti CSS.
3. Dapat membuat program sederhana dengan menggunakan CSS

B. Teori Dasar CSS

CSS (Cascading Style Sheets) adalah bahasa yang digunakan untuk mengatur tampilan dan layout halaman web. CSS memungkinkan kita mengubah warna, ukuran, posisi, dan gaya elemen HTML agar halaman web menjadi menarik dan mudah dibaca. CSS digunakan untuk mengatur tampilan elemen HTML di berbagai media seperti yang tertampil pada layar komputer atau perangkat lainnya. CSS memungkinkan pengembang web untuk mengontrol tata letak banyak halaman sekaligus sehingga menghemat waktu dan usaha dalam desain web.

Pentingnya menggunakan CSS

1. **Memisahkan** Konten dan Desain : HTML digunakan untuk menyusun konten sementara CSS menentukan tampilannya.
2. **Efisiensi** : Dengan satu file CSS tampilan seluruh situs web dapat diubah tanpa harus mengedit setiap halaman HTML.
3. **Konsistensi** : Memastikan tampilan yang seragam di seluruh halaman web.
4. **Responsivitas** : CSS memungkinkan desain yang dapat menyesuaikan dengan berbagai ukuran layar.

C. Pratikum

1. Penulisan CSS (Cascading Style Sheets) Penjelasan Dalam peneterapannya CSS (Cascading Style Sheets) bisa digunakan dengan bahasa HTML . Contoh Code File index.htm

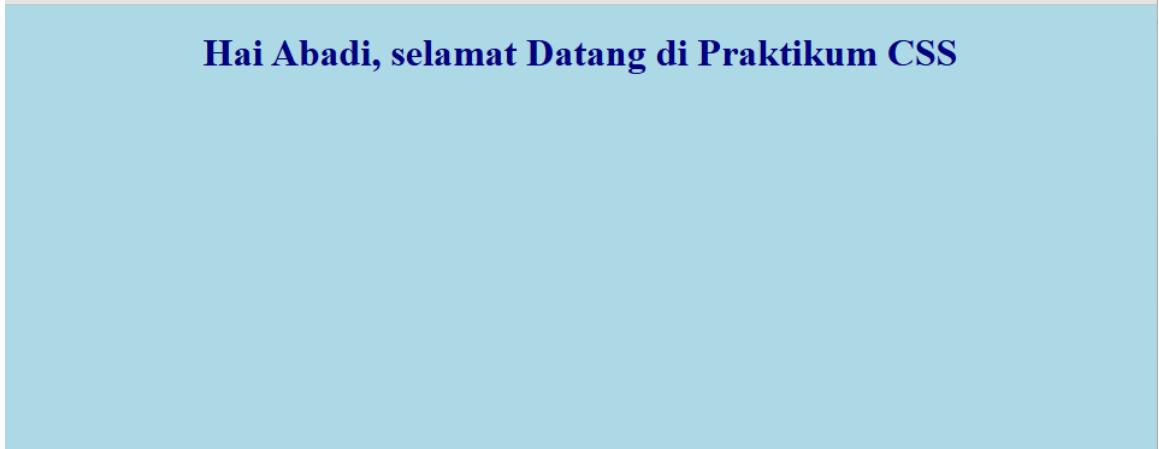
```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <style>
5.     body {
6.       background-color: lightblue;
7.     }
8.     h1 {
9.       color: navy;
10.      text-align: center;
11.    }
12.   </style>
13. </head>
14. <body>
15.   <h1>Hai Abadi, selamat Datang di Praktikum CSS</h1>
16. </body>
17. </html>
```

Latihan Pratikum

- Silakan pratikan contoh kode di atas
- Latihan selanjutnya buat halaman HTML dengan latar belakang berwarna kuning muda. Tambahkan judul dengan warna merah dan teks rata Tengah.
- Berikan *capture*(printscreen/tangkapan layer) ada **2 capture** yaitu dari contoh code dan Latihan kemudian analisis dari hasil pratikum anda di laporan.

Jawab:

Berikut jika kode atas dipraktekkan:



Hai Abadi, selamat Datang di Praktikum CSS

Untuk mengubah latar belakang berwarna kuning dan judul dengan warna merah dan teks rata Tengah, yaitu kode html+css jadi berikut:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: yellow; /* latar belakang kuning */
  text-align: center; /* teks di tengah */
```

```
}

h1 {
    color: red;          /* judul berwarna merah */
}

</style>
</head>
<body>
<h1>Hai Abadi, selamat Datang di Praktikum CSS</h1>
</body>
</html>
```

Jika dipraktikan:



Hai Abadi, selamat Datang di Praktikum CSS

Analisis:

- Versi **sebelum** lebih kalem, cocok untuk tampilan formal atau edukatif.
- Versi **sesudah** lebih **menarik dan kuat secara visual**, cocok untuk tampilan yang ingin cepat **menarik perhatian** (misalnya promosi, pengumuman, atau praktikum yang menyenangkan).
- **Struktur CSS** juga jadi lebih rapi dan efisien dengan text-align diatur langsung di body.

2. Sintaks dan Cara menyisipkan CSS

Penjelasan

CSS terdiri dari selector dan deklarasi. Selector memilih elemen HTML yang ingin diatur sedangkan deklarasi berisi properti dan nilainya.

Ada tiga cara menyisipkan CSS:

- a. Inline CSS (langsung di tag HTML)
- b. Internal CSS (di dalam tag <style> di bagian <head>)
- c. External CSS (file terpisah dengan ekstensi .css)

Contoh Kode

Inline CSS:

File inline_css.htm

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
5. <h1 style="color:blue;text-align:center;">This is a heading</h1>
6. <p style="color:red;">This is a paragraph.</p>
7.
8. </body>
9. </html>
```

Internal CSS:

File internal_css.htm

```
1. <!DOCTYPE html>
2. <html>
3. <head>

4. <style>
5. body {
6.   background-color: linen;
7. }
8.
9. h1 {
10.   color: maroon;
11.   margin-left: 40px;
12. }
13. </style>
14. </head>
15. <body>
16.
17. <h1>This is a heading</h1>
18. <p>This is a paragraph.</p>
19.
20. </body>
21. </html>
```

External CSS:

File external_css.htm

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <link rel="stylesheet" href="mystyle.css">
5. </head>
6. <body>
7.
8. <h1>This is a heading</h1>
9. <p>This is a paragraph.</p>
10.
11. </body>
12. </html>
```

File mystyle.css

```
1. body {
2.   background-color: lightblue;
3. }
4.
5. h1 {
6.   color: navy;
7.   margin-left: 20px;
8. }
```

Latihan pratikum

- Silakan pratikkan code diatas
- Buat file HTML dan file CSS eksternal. Hubungkan file CSS ke HTML dan ubah warna teks paragraph menjadi ungu
- Berikan *capture*(*print screen/tangkapan layar*) Ada **2 capture** yaitu dari contoh code dan Latihan kemudian analisis dari hasil pratikum anda di laporan

Pratik code di contoh soal:

Inline css:

This is a heading

This is a paragraph.

Internal css:

This is a heading

This is a paragraph.

Eksternal CSS:

This is a heading

This is a paragraph.

Buat file HTML dan File CSS Eksternal. Hubungkan file css ke HTML dan ubah warna paragraph menjadi ungu:

Dengan mengubah code di mystyle.css dengan tambahan p{color:purple;}

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: navy;  
    margin-left: 20px;  
}  
  
p {  
    color: purple;  
}
```

Berikut jika diperlakukan:

This is a heading

This is a paragraph.

Analisis Setelah Perubahan

Setelah menambahkan selector p dan menetapkan color: purple;:

- Paragraf tampil dengan warna ungu sesuai instruksi.
- Ini menunjukkan bahwa file CSS eksternal berhasil dihubungkan dan diterapkan dengan benar ke file HTML.
- Penggunaan CSS eksternal memungkinkan styling konsisten dan lebih mudah dikelola dibandingkan inline style.

3. Selector CSS

Penjelasan

Selector digunakan untuk memilih elemen HTML yang akan diberi gaya.

Contoh selector:

- a. Selector elemen : p, h1
- b. Selector kelas : kelas
- c. Selector IDE : #id,

Contoh Kode

File selector_css.htm

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5.   p { color: red; }
6.   .highlight { background-color: yellow; }
7.   #judul { font-size: 24px; }
8. </style>
9. </head>
10. <body>
11.
12. <p>Paragraf biasa</p>
13. <p class="highlight">Paragraf dengan highlight</p>
14. <h1 id="judul">Judul Halaman</h1>
15.
16. </body>
17. </html>
```

Latihan Pratikum

- Silakan praktikkan contoh *code* di atas.
- Buat beberapa paragraf dengan kelas berbeda dan atur warna latar belakangnya. Buat elemen dengan ID dan ubah ukuran *font*-nya.
- Berikan *capture* (*print screen* /tangkapan layar) Ada 2 *capture* yaitu dari contoh *code* dan latihan kemudian analisis dari hasil praktikum Anda di laporan.

Jawaban latihan pratikum:

Jika contoh kode di atas dipraktikan:

Paragraf biasa

Paragraf dengan highlight

Judul Halaman

Pembuatan paragraf dengan kelas berbeda dan atur warna latar belakang, buat elemen dengan IDE dan ubah ukuran font

Hasilnya:

Paragraf pertama dengan latar belakang biru muda dan ukuran font 18px.

Paragraf kedua dengan latar belakang hijau muda dan ukuran font 20px.

Paragraf ketiga dengan latar belakang merah muda dan ukuran font 22px.

Dengan kode HTML:

```
<!DOCTYPE html>
<html>
<head>
<style>
.class1 {
    background-color: lightblue;
    font-size: 18px;}
.class2 {
    background-color: lightgreen;
    font-size: 20px;}
.class3 {
    background-color: lightcoral;
    font-size: 22px;}
</style>
</head>
<body>
<p class="class1">Paragraf pertama dengan latar belakang biru muda dan ukuran font 18px.</p>
<p class="class2">Paragraf kedua dengan latar belakang hijau muda dan ukuran font 20px.</p>
<p class="class3">Paragraf ketiga dengan latar belakang merah muda dan ukuran font 22px.</p>

</body>
</html>
```

Analisa:

Kesimpulan:

- Kode 1 lebih sederhana, dengan penggunaan kelas dan ID yang terfokus pada elemen penting, dan sangat efektif dalam mengatur elemen spesifik seperti judul dan highlight.
- Kode 2 lebih cocok untuk memberikan tampilan yang bervariasi pada beberapa elemen sejenis (seperti paragraf) dan menawarkan fleksibilitas lebih dalam penataan.

4. Properti Warna dan Background

Penjelasan

CSS menyediakan properti untuk mengatur warna teks dan latar belakang, seperti:

1. color : warna teks
2. background-color : warna latar belakang
3. Background-image : gambar latar belakang

Contoh kode

File propertiwarna_css.html

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. p {
6.   color: white;
7.   background-color: black;
8. }
9.
10. body {
11.   background-image: url('gambar.jpg');
12.   background-repeat: no-repeat;
13.   background-size: cover;
14. }
15.
16.
17. </style>
18. </head>
19. <body>
20.
21. <p>Nama saya abadi nugroho</p>
22.
23. </body>
24. </html>
```

Latihan Pratikum

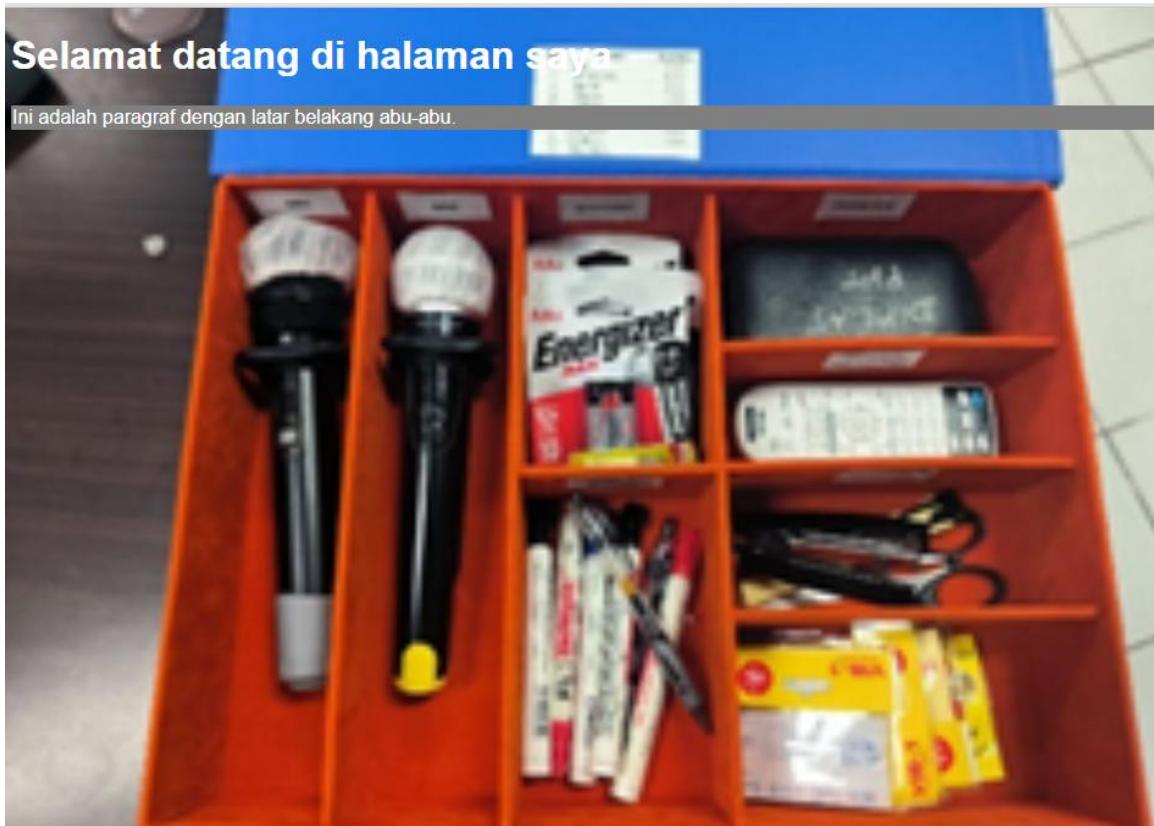
- Silakan praktikkan contoh kode di atas
- Buat halaman dengan latar belakang gambar dan teks berwarna putih. Ubah warna latar belakang paragraf menjadi abu-abu.
- Berikan *capture* (*print screen/tangkapan layar*) ada 2 *capture* yaitu dari contoh code dan latihan kemudian analisis dari hasil prktikum Anda di laporan.

Jawab:

Jika kode di atas dipraktikkan maka akan menghasilkan berikut:

```
Nama saya abadi nugroho
```

Buat halaman dengan latar belakang gambar dan teks berwarna putih. Ubah warna latar belakang paragraf menjadi abu-abu. Berikut:



Dengan kode html:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url('file:///C:/Users/Asus/OneDrive/Pictures/Lampiran8.png');
    background-repeat: no-repeat;
    background-size: cover;
    color: white; /* Text color is white */
    font-family: Arial, sans-serif;
}

p {
    background-color: gray; /* Gray background for paragraph */
    padding: 1px;
    border-radius: 1px;
}
</style>
```

```
</head>
<body>

<h1>Selamat datang di halaman saya</h1>
<p>Ini adalah paragraf dengan latar belakang abu-abu.</p>

</body>
</html>
```

Analisa:

Kedua kode di atas memiliki fungsi serupa, yaitu menambahkan gambar latar belakang dan mengatur gaya teks dan paragraf. Perbedaan utama adalah penggunaan path gambar lokal pada kode pertama, yang membatasi akses pada komputer tertentu, sedangkan kode kedua menggunakan gambar dengan path relatif, yang lebih fleksibel untuk digunakan dalam proyek berbasis web.

6. Box Model (Margin, Border, Padding, dan Content)

Penjelasan

Setiap Elemen HTML dianggap sebagai kotak yang terdiri dari:

- a. Content : isi elemen
- b. Padding : ruang di dalam kotak yaitu antara isi dengan border
- c. Border : Garis tepi kotak
- d. Margin : ruang di lluar border yaitu memisahkan elemen dengan elemen lain

Contoh kode

File boxmodel_css.htm

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    margin: 20px;
    padding: 10px;
    border: 2px solid black;
    width: 200px;
}
</style>
</head>
<body>

<div>Nama saya adalah abadi nugroho, saya sedang melakukan praktikum css.</div>

</body>
</html>
```

Latihan Praktikum

- Silakan Praktikkan contoh code di atas.

```
Nama saya adalah abadi  
nugroho, saya sedang  
melakukan praktikum css.
```

Jawab:

- Buat kotak dengan margin 30px, padding 15px, dan border merah tebal. Ubah ukuran kotak menjadi 300px lebar

Jawab:

Berikut hasil codenya

```
Ini adalah kotak dengan margin 30px, padding  
15px, dan border merah tebal.
```

Berikut codenya:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div {  
    margin: 30px;  
    padding: 15px;  
    border: 5px solid red; /* Border merah tebal */  
    width: 300px; /* Lebar kotak 300px */  
}  
</style>  
</head>  
<body>  
  
    <div>Ini adalah kotak dengan margin 30px, padding 15px, dan border merah  
    tebal.</div>  
  
</body>  
</html>
```

- Analisa:

Kode pertama lebih cocok untuk desain yang lebih sederhana dan kompak, dengan ukuran kotak yang lebih kecil.

Kode kedua lebih cocok untuk menonjolkan elemen atau untuk penggunaan yang membutuhkan ruang lebih besar di dalam kotak dan border yang lebih tebal untuk menarik perhatian.

7. Layout dengan Flexbox

Penjelasan

Flexbox adalah metode *layout* modern yang memudahkan pengaturan posisi elemen dalam satu baris atau kolom

Contoh Kode

File flexbox_css.html

```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
    display: flex;
    justify-content: space-around;
    align-items: center;
}

.item {
    background-color: lightcoral;
    padding: 20px;
    margin: 10px;
}
</style>
</head>
<body>

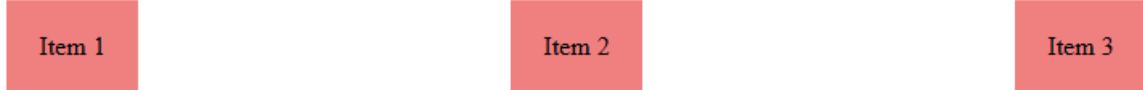
<div class="container">
<div class="item">Item 1</div>
<div class="item">Item 2</div>
<div class="item">Item 3</div>
</div>

</body>
</html>
```

Latihan Praktikum

- Silakan praktikkan contoh code di atas.

Jawab:



Item 1

Item 2

Item 3

- Buat container flex dengan 3 item. Atur agar item terbesar merata dan sejajar di tengah.

Jawab:



Item 1 (besar)

Item 2 (tetap)

Item 3 (besar)

Dengan code:

```
<!DOCTYPE html>
```

```

<html>
<head>
<style>
    .container {
        display: flex;
        justify-content: center; /* Menyelaraskan semua item di tengah */
        align-items: center; /* Menyelaraskan item secara vertikal di tengah */
        height: 100vh; /* Memberikan tinggi penuh untuk menampilkan hasil secara jelas */
    }

    .item {
        background-color: lightcoral;
        padding: 20px;
        margin: 10px;
    }

    .item1 {
        flex-grow: 1; /* Item pertama akan berkembang untuk mengisi ruang yang tersedia */
    }
    .item2 {
        flex-grow: 0; /* Item kedua memiliki ukuran tetap */
    }
    .item3 {
        flex-grow: 1; /* Item ketiga akan berkembang untuk mengisi ruang yang tersedia */
    }
</style>
</head>
<body>
    <div class="container">
        <div class="item item1">Item 1 (besar)</div>
        <div class="item item2">Item 2 (tetap)</div>
        <div class="item item3">Item 3 (besar)</div>
    </div>
</body>
</html>

```

Analisa dari kedua kode di atas:

- **Kode pertama** lebih cocok untuk desain yang lebih sederhana, di mana item harus diposisikan secara merata tanpa perlu mengubah ukuran.
- **Kode kedua** lebih cocok untuk desain yang membutuhkan pengaturan ukuran fleksibel pada item, seperti menonjolkan item terbesar di tengah atau memberi ukuran tetap pada item lainnya.

8. CSS Grid Layout

Penjelasan

CSS Grid adalah metode layout dua dimensi yang sangat powerful untuk mengatur elemen-elemen dalam baris dan kolom. Dengan Grid kita bisa membuat layout yang kompleks dengan mudah dan fleksibel.

Konsep Utama CSS Grid:

- a. Grid Container: Elemen induk yang diatur sebagai grid dengan properti display: grid;
- b. Grid Items: Elemen anak di dalam grid container yang akan diatur posisinya
- c. Grid Lines: Garis-garis pembatas baris dan kolom
- d. Grid Tracks: Baris dan kolom yang dibentuk oleh grid lines
- e. Grid Areas: Area yang dibentuk oleh beberapa grid cells

Contoh Kode

File grid_css.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Grid CSS Example</title>
<style>
.grid-container {
    display: grid;
    grid-template-columns: 100px 200px 100px;
    grid-template-rows: 50px 100px;
    gap: 10px;
    background-color: #f2f2f2;
    padding: 10px;
}

.grid-item {
    background-color: #4CAF50;
    color: white;
    text-align: center;
    padding: 20px;
    font-size: 18px;
}
</style>
</head>
<body>

<h2>SEKOLAH TINGGI TEKNOLOGI BONTANG</h2>
<p>PROGRAM STUDI TEKNIK INFORMATIKA</p>
<p><strong>Praktikum:</strong> Pemrograman Web</p>
<p><strong>Modul 2:</strong> Pengenalan CSS (Cascading Style Sheet)</p>
<p><strong>Tgl Update:</strong> 02-06-2025</p>

<div class="grid-container">
<div class="grid-item">1</div>
<div class="grid-item">2</div>
<div class="grid-item">3</div>
<div class="grid-item">4</div>
<div class="grid-item">5</div>
<div class="grid-item">6</div>
</div>
```

```
</body>  
</html>
```

Latihan Praktikum

- Silakan praktikkan contoh kode diatas. Berikut:

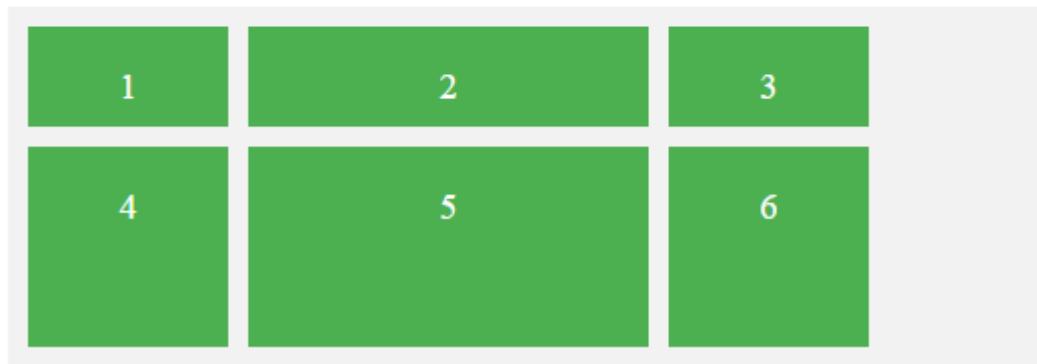
SEKOLAH TINGGI TEKNOLOGI BONTANG

PROGRAM STUDI TEKNIK INFORMATIKA

Praktikum: Pemrograman Web

Modul 2: Pengenalan CSS (Cascading Style Sheet)

Tgl Update: 02-06-2025



- Buatlah grid container dengan 3 kolom dan 2 baris. Atur jarak antar grid item sebesar 15 px. Buat 6 grid *item* dengan warna latar berbeda. Cobalah mengatur salah satu grid item agar menempati dua kolom sekaligus.

Berikut Code:

```
<!DOCTYPE html>  
<html>  
<head>  
    <meta charset="UTF-8">  
    <title>Grid Layout 3x2</title>  
    <style>  
        .grid-container {  
            display: grid;  
            grid-template-columns: repeat(3, 1fr); /* 3 kolom sama lebar */  
            grid-template-rows: repeat(2, 100px); /* 2 baris tinggi 100px */  
            gap: 15px;  
            padding: 20px;  
            background-color: #f5f5f5;  
        }  
  
        .grid-item {  
            display: flex;  
            align-items: center;  
            justify-content: center;  
            color: white;  
            font-size: 18px;  
        }  
    </style>  
</head>  
<body>  
    <div class="grid-container">  
        <div class="grid-item">1</div>  
        <div class="grid-item">2</div>  
        <div class="grid-item">3</div>  
        <div class="grid-item">4</div>  
        <div class="grid-item">5</div>  
        <div class="grid-item">6</div>  
    </div>  
</body>  
</html>
```

```

        font-weight: bold;
        border-radius: 8px;
    }

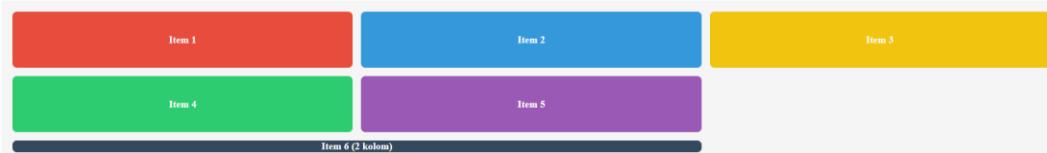
    .item1 { background-color: #e74c3c; }
    .item2 { background-color: #3498db; }
    .item3 { background-color: #f1c40f; }
    .item4 { background-color: #2ecc71; }
    .item5 { background-color: #9b59b6; }
    .item6 {
        background-color: #34495e;
        grid-column: span 2; /* Menempati 2 kolom secara horizontal */
    }
</style>
</head>
<body>

<div class="grid-container">
    <div class="grid-item item1">Item 1</div>
    <div class="grid-item item2">Item 2</div>
    <div class="grid-item item3">Item 3</div>
    <div class="grid-item item4">Item 4</div>
    <div class="grid-item item5">Item 5</div>
    <div class="grid-item item6">Item 6 (2 kolom)</div>
</div>

</body>
</html>

```

Berikut jika code di atas dijalankan di web browser:



- Analisa:

Kesimpulan Perbandingan

Aspek	Kode 1	Kode 2
Tipe Grid	Fixed size (px)	Flexible (fr unit)
Jumlah Item	6	6
Variasi Warna	Tidak ada (semua hijau)	Ada (setiap item beda warna)
Gap antar item	10px	15px (lebih lega)
Item menempati 2 kolom	✗ Tidak ada	✓ Ya, .item6 pakai grid-column: span 2
Estetika visual	Sederhana dan kaku	Modern dan dinamis

Aspek	Kode 1	Kode 2
Cocok untuk	Pembelajaran dasar CSS Grid	Implementasi layout nyata (dashboard, card layout)

9. Responsive Desain (Desain Responsif)

Penjelasan

Responsive Design adalah teknik membuat tampilan web yang dapat menyesuaikan diri dengan berbagai ukuran layar perangkat, seperti desktop, tablet, dan smartphone. Tujuannya agar website tetap nyaman dilihat dan digunakan di semua perangkat.

Cara utama membuat desain responsif:

- a. Menggunakan media queries untuk mengubah gaya CSS berdasarkan ukuran layar.
- b. Menggunakan layout fleksibel seperti Flexbox dan Grid.
- c. Menggunakan satuan relatif seperti %, em, dan rem daripada satuan tetap seperti px.

Media Queries

Media queries memungkinkan kita menulis aturan CSS yang hanya berlaku pada kondisi tertentu, misalnya ukuran layar tertentu.

Contoh Kode

Kode dibawah ini tidak perlu di praktikkan, cukup dipahami.

```
/* Gaya default untuk desktop */
body {
    background-color: lightblue;
    font-size: 18px;
}

/* Jika layar maksimal 600px (misal smartphone) */
@media only screen and (max-width: 600px) {
    body {
        background-color: lightcoral;
        font-size: 14px;
    }
}
```

File responsive_css.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Responsive Grid Example</title>
    <style>
        .grid-container {
            display: grid;
            grid-template-columns: repeat(3, 1fr);
```

```

        gap: 10px;
        padding: 10px;
    }

    .grid-item {
        background-color: #2196F3;
        color: white;
        padding: 20px;
        font-size: 20px;
        text-align: center;
    }

    /* Responsive untuk layar kecil */
    @media only screen and (max-width: 600px) {
        .grid-container {
            grid-template-columns: 1fr; /* 1 kolom */
        }
    }

```

</style>

</head>

<body>

<div class="grid-container">

<div class="grid-item">Item 1</div>

<div class="grid-item">Item 2</div>

<div class="grid-item">Item 3</div>

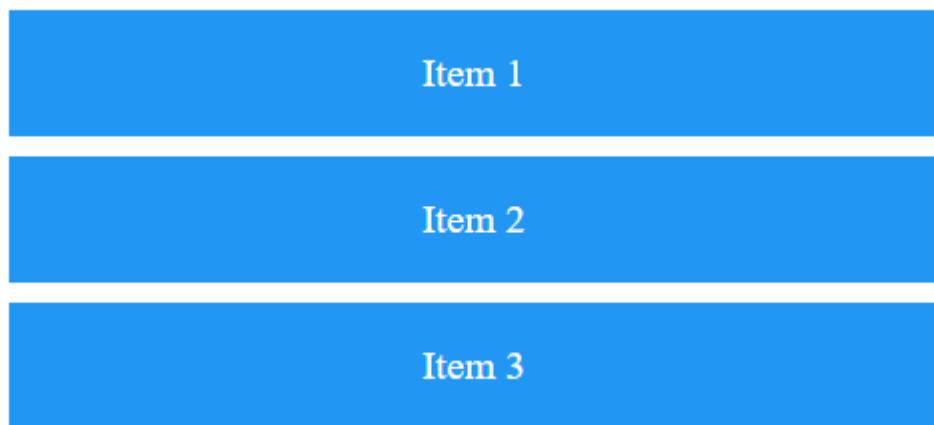
</div>

</body>

</html>

Laporan Praktikum

- Mempraktikan contoh kode di atas:
Berikut jika lebar layar terlalu kecil:



Berikut jika lebar layar mencukupi:



- Buat halaman dengan grid 3 kolom untuk layar desktop. Gunakan media query agar pada layar dengan lebar maksimal 600px, grid berubah menjadi 1 kolom. Ubah warna latar belakang halaman sesuai ukuran layar (misal

biru untuk desktop, merah untuk smartphone). Coba tambahkan teks yang ukurannya berubah sesuai ukuran layar.

Berikut jawaban jika tampilan smartphone:

Ini adalah teks yang ukurannya menyesuaikan layar

Kolom 1

Kolom 2

Kolom 3

Berikut tampilan desktop:

Ini adalah teks yang ukurannya menyesuaikan layar

Kolom 1

Kolom 2

Kolom 3

Dengan kode:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Responsive Grid Layout</title>
<style>
body {
    background-color: lightblue;
    font-size: 18px;
    margin: 0;
    font-family: Arial, sans-serif;
}
```

```

.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 15px;
  padding: 20px;
}

.grid-item {
  background-color: #4CAF50;
  color: white;
  padding: 30px;
  text-align: center;
  border-radius: 8px;
  font-weight: bold;
}

.responsive-text {
  text-align: center;
  padding: 20px;
  font-size: 24px;
}

@media only screen and (max-width: 600px) {
  body {
    background-color: lightcoral;
    font-size: 14px;
  }
}

.container {
  grid-template-columns: 1fr;
}

.responsive-text {
  font-size: 18px;
}
}

</style>
</head>
<body>

<div class="responsive-text">
  Ini adalah teks yang ukurannya menyesuaikan layar
</div>

<div class="container">
  <div class="grid-item">Kolom 1</div>
  <div class="grid-item">Kolom 2</div>
  <div class="grid-item">Kolom 3</div>
</div>

</body>
</html>

```

Analisa:

Kode ini sangat baik sebagai latihan atau proyek awal untuk memahami prinsip **CSS Grid** dan **media queries**. Ia mencerminkan praktik modern dalam membuat halaman yang bisa beradaptasi otomatis dengan ukuran layar, memperhatikan **kenyamanan visual**, dan memiliki struktur yang bisa dengan mudah dikembangkan ke versi yang lebih kompleks (misalnya menambahkan gambar, tombol, navigasi, dsb).

D. Tugas Praktikum

Buatlah sebuah halaman web sederhana dengan ketentuan sebagai berikut:

1. Tampilan Umum

- a. Background halaman berwarna biru muda (#add8e6).
- b. Gunakan font Arial atau sans-serif untuk seluruh halaman.
- c. Halaman memiliki judul utama (`<h1>`) dengan teks "Praktikum CSS" yang berwarna putih dan berukuran besar (sekitar 40px).
- d. Tambahkan sebuah paragraf (`<p>`) di bawah judul dengan latar belakang abu-abu muda (#d3d3d3), teks berwarna hitam, dan padding yang cukup agar nyaman dibaca.

2. Layout Kotak (Box)

- a. Buat sebuah container yang berisi tiga kotak (`div`) dengan kelas `.box1`, `.box2`, dan `.box3`.
- b. Setiap kotak memiliki border hitam tebal 2px, padding 10px, dan tinggi 150px.
- c. Warna latar belakang kotak:
 - 1) `.box1`: merah muda (#ff9999)
 - 2) `.box2`: hijau muda (#99ff99)
 - 3) `.box3`: biru muda (#9999ff)
- d. Container menggunakan **CSS Grid** dengan 3 kolom yang sama besar dan jarak antar kotak 20px.
- e. Container dibatasi maksimal lebar 800px dan diposisikan di tengah halaman.

3. Responsive Design

- a. Ketika lebar layar kurang dari atau sama dengan 768px (tablet), container harus berubah menjadi 2 kolom.
- b. Ketika lebar layar kurang dari atau sama dengan 480px (smartphone), container berubah menjadi 1 kolom.
- c. Pada tampilan smartphone, ukuran font judul diperkecil menjadi 28px, tinggi kotak menjadi 120px, dan padding paragraf dikurangi agar proporsional.

4. Tambahan

- a. Berikan border-radius pada kotak dan paragraf agar sudutnya membulat.
- b. Pastikan seluruh elemen menggunakan `box-sizing` yang tepat agar padding dan border tidak merusak ukuran elemen.
- c. Simpan kode di github anda masing-masing.
- d. Buatlah video untuk menjelaskan script dari tugas yang anda buat, simpan di penyimpanan online misalnya youtube anda masing-masing dan sertakan link github dan video di laporan anda.

Link vidio dapat didownload pada link berikut: [Link](#)

Berikut hasilnya:

The screenshot shows a web page with a light blue header containing the title "Praktikum CSS". Below the header is a grey box containing text about CSS grid layout. At the bottom of the page are three rectangular boxes labeled "Kotak 1", "Kotak 2", and "Kotak 3", each in a different color (red, green, and blue respectively).

Berikut jika dijalankan di web browser dengan lebar kecil (smartphone):

Praktikum CSS

Ini adalah paragraf pembuka yang menjelaskan tentang penggunaan CSS dalam pembuatan layout responsif menggunakan grid. Desain ini dibuat agar nyaman dibaca di berbagai perangkat mulai dari desktop, tablet, hingga smartphone.

Kotak 1

Kotak 2

Kotak 3



Praktikum CSS

Ini adalah paragraf pembuka yang menjelaskan tentang penggunaan CSS dalam pembuatan layout responsif menggunakan grid. Desain ini dibuat agar nyaman dibaca di berbagai perangkat mulai dari desktop, tablet, hingga smartphone.

Kotak 1

Kotak 2

Kotak 3



Dengan kode:

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Praktikum CSS</title>
    <style>
        /* Box sizing untuk semua elemen */
        * {
            box-sizing: border-box;
        }

        body {
            background-color: #add8e6; /* biru muda */
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 20px;
        }

        h1 {
            color: white;
            font-size: 40px;
            text-align: center;
            margin-bottom: 10px;
        }

        p {
            background-color: #d3d3d3; /* abu-abu muda */
            color: black;
            padding: 20px;
            border-radius: 10px;
            max-width: 800px;
            margin: 0 auto 30px auto;
            text-align: justify;
        }

        .container {
            display: grid;
            grid-template-columns: repeat(3, 1fr);
            gap: 20px;
            max-width: 800px;
            margin: 0 auto;
        }

        .box1, .box2, .box3 {
            border: 2px solid black;
            padding: 10px;
            height: 150px;
            border-radius: 10px;
        }

        .box1 {
            background-color: #ff9999;
        }
    </style>

```

```

}

.box2 {
    background-color: #99ff99;
}

.box3 {
    background-color: #9999ff;
}

/* Tablet (≤768px): 2 kolom */
@media only screen and (max-width: 768px) {
    .container {
        grid-template-columns: repeat(2, 1fr);
    }
}

/* Smartphone (≤480px): 1 kolom */
@media only screen and (max-width: 480px) {
    h1 {
        font-size: 28px;
    }

    p {
        padding: 10px;
    }

    .container {
        grid-template-columns: 1fr;
    }
}

.box1, .box2, .box3 {
    height: 120px;
}

```

</style>

</head>

<body>

<h1>Praktikum CSS</h1>

<p>

Ini adalah paragraf pembuka yang menjelaskan tentang penggunaan CSS dalam pembuatan layout responsif menggunakan grid. Desain ini dibuat agar nyaman dibaca di berbagai perangkat mulai dari desktop, tablet, hingga smartphone.

</p>

```

<div class="container">
    <div class="box1">Kotak 1</div>
    <div class="box2">Kotak 2</div>
    <div class="box3">Kotak 3</div>
</div>
</body>

```

</html>

Modul 3. Pengenalan CSS (Cascading Style Sheet)

A. Tujuan

Setelah mengikuti praktikum, Mahasiswa diharapkan mampu:

1. Mampu memahami bootstrap sebagai framework CSS.
2. Dapat membuat program dengan framework CSS.

B. Teori Dasar Bootstrap

Bootstrap adalah framework CSS *open-source* yang sangat populer untuk membangun website dan aplikasi web yang responsif. Bootstrap 5 adalah versi terbaru yang membawa banyak perbaikan dan fitur baru dibanding versi sebelumnya

Keunggulan Bootstrap 5

1. Mudah digunakan kelas-kelas CSS siap pakai
2. Sistem grid yang fleksibel untuk layout responsif
3. Banyak komponen UI siap pakai (button, navbar, card, modal, dll)
4. Dukungan utilities untuk styling cepat
5. Tidak lagi bergantung pada jQuery (lebih ringan dan modern).

C. Praktikum

1. Instalasi dan Setup Bootstrap

Menggunakan CDN (Content Delivery Network)

Penulisan kode di dalam tag <head> pada file HTML.

```
1. <!-- Latest compiled and minified CSS -->
2. <link
   href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
   rel="stylesheet">
3.
4. <!-- Latest compiled JavaScript -->
5. <script
   src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></s
cript>
6.
```

Contoh Code

File setup_bootstrap.html

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Latihan Framework Bootstrap</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link
   href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
   rel="stylesheet">
8.   <script
   src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></s
cript>
9. </head>
10. <body>

11. <div class="container">
12.   <h1>Hai, nama saya abadi nugroho</h1>
13.   <p>ini bagian dari .container class. lho</p>
14.   <p>.container class membuat kotak yang lebarnya bisa responsive.</p>
15.   <p>Silahkan coba besak kecilkan tampilan broser anda kalau ndak percaya.</p>
16. </div>
17. </body>
18. </html>
```

Latihan Praktikum:

- Silakan praktikan contoh kode di atas.

Jawab:

Saat tampilan browser kecil:

Hai, nama saya Abadi Nugroho

Ini bagian dari .container class. Lho

.container class membuat kotak yang lebarnya bisa responsif.

Silakan coba besar kecilkan tampilan browser Anda kalau nggak percaya.

Saat tampilan browser besar:

Hai, nama saya Abadi Nugroho

Ini bagian dari .container class. Lho

.container class membuat kotak yang lebarnya bisa responsif.

Silakan coba besar kecilkan tampilan browser Anda kalau nggak percaya.

2. Sistem grid di Bootstrap 5

Penjelasan

Bootstrap menggunakan sistem grid berbasis flexbox yang membagi halaman menjadi 12 kolom. Anda dapat mengatur layout dengan mudah menggunakan kelas grid.

Bootstrap menyediakan kelas grid responsif berdasarkan ukuran layar:

Kelas	Ukuran Layar	Contoh penggunaan
.col-sm-	≥ 576px (small devices)	<div class="col-sm-6">
.col-md-	≥ 768px (medium devices)	<div class="col-md-4">
.col-lg-	≥ 992px (large devices)	<div class="col-lg-3">
.col-xl-	≥ 1200px (extra large devices)	<div class="col-xl-2">

Contoh Kode

Grid 3 Kolom

File grid_bootstrap.html

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Latihan Grid 3 Kolom</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
8.   <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></s
cript>
9. </head>
10. <body>
11. <div class="container">
12.   <div class="row">
13.     <div class="col">Kolom 1</div>
14.     <div class="col">Kolom 2</div>
15.     <div class="col">Kolom 3</div>
16.   </div>
17. </div>
18. </body>
19. </html>
```

.container : pembungkus utama yang memberikan padding dan mengatur lebar maksimal.

.row : baris yang berisi kolom

.col : kolom yang secara otomatis membagi ruang secara merata.

Latihan Praktikum

- Silakan praktikan contoh kode diatas

Jawab

Kolom 1	Kolom 2	Kolom 3
---------	---------	---------

- Buatlah halaman web yang memiliki 3 kolom menggunakan grid bootstrap. Setiap kolom berisi teks berbeda dan memiliki background warna berbeda.

Berikut kodenya

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Halaman Web 3 Kolom</title>
    <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.mi
n.css" rel="stylesheet">
    <script
        src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bund
le.min.js"></script>
    <style>
        .col-1 {
            background-color: #f8d7da; /* Background merah muda */
            padding: 20px;
            text-align: center;
        }
        .col-2 {
            background-color: #d4edda; /* Background hijau muda */
            padding: 20px;
            text-align: center;
        }
        .col-3 {
            background-color: #cce5ff; /* Background biru muda */
            padding: 20px;
            text-align: center;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="row">
```

```

<div class="col col-1">
    <h3>Kolom 1</h3>
    <p>Ini adalah kolom pertama dengan background merah muda.</p>
</div>
<div class="col col-2">
    <h3>Kolom 2</h3>
    <p>Ini adalah kolom kedua dengan background hijau muda.</p>
</div>
<div class="col col-3">
    <h3>Kolom 3</h3>
    <p>Ini adalah kolom ketiga dengan background biru muda.</p>
</div>
</div>
</body>
</html>

```

Berikut Hasilnya jika dijalankan di web browser

Kolom 1	Kolom 2	Kolom 3
1 Ini adalah kolom pertama dengan background merah muda.	Ini adalah kolom kedua dengan background hijau muda.	Ini adalah kolom ketiga dengan background biru muda.

- **Analisa:**

Secara keseluruhan, kode ini membuat halaman web responsif dengan tiga kolom menggunakan Bootstrap grid system. Desain ini akan berfungsi baik di perangkat dengan ukuran layar besar maupun kecil, dan memberikan pengalaman pengguna

yang baik dengan warna latar belakang yang berbeda untuk masing-masing kolom.

3. Typography dan Utilities

Penjelasan

Bostrap menyediakan kelas untuk mengatur teks dan utilitas styling cepat.

Contoh Kode

File typo_bootsrap.html

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <meta name="viewport" content="width=device-width, initial-scale=1">
5.   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
6.   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
7. </head>
8. <body>
9. <div class="container mt-3">
10.   <h2>Typography</h2>
11.   <p>Ini merupakan contoh tulisan paragraf normal.</p>
12.   <h1>h1 Bootstrap heading</h1>

13.   <h2>h2 Bootstrap heading</h2>
14.   <h3>h3 Bootstrap heading</h3>
15.   <h4>h4 Bootstrap heading</h4>
16.   <h5>h5 Bootstrap heading</h5>
17.   <h6>h6 Bootstrap heading</h6>
18.   <p class="text-lowercase">Lowercased text.</p>
19.   <p class="text-uppercase">Uppercased text.</p>
20.   <p class="text-capitalize">Capitalized text.</p>
21.   <p class="lead">This paragraph stands out.</p>
22.   <p class="text-start">Left-aligned text.</p>
23.   <p class="text-end">Right-aligned text.</p>
24.   <p class="text-center">Center-aligned text.</p>
25.   <p>Gunakan <kbd>ctrl + c</kbd> untuk copy dokumen.</p>
26.   <p>Kampus <abbr title="Sekolah Tinggi Teknologi Bontang">STITEK</abbr> mempunyai program studi Teknik Informatika.</p>
27.   <p>Ini contoh penulisan dengan style code: <code>span</code>, <code>section</code>, and <code>div</code>.</p>
28.   <ul class="list-unstyled">
29.     <li>Coffee</li>
30.     <li>Tea
31.       <ul>
32.         <li>Black tea</li>
33.         <li>Green tea</li>
34.       </ul>
35.     </li>
36.     <li>Milk</li>
37.   </ul>
38.   <blockquote class="blockquote">
39.     <p>Ini merupakan contoh penulisan dengan style quote.</p>
40.     <footer class="blockquote-footer">From Abadi Nugroho</footer>
41.   </blockquote>
42. </div>
43. </body>
44. </html>
45.

```

Text colors

File text_colors_bootstrap.html

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap font color</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8.   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
9. </head>
10. <body>
11.
12. <div class="container mt-3">
13.   <h2>Warna pada font</h2>
14.   <p>Use the contextual classes to provide "meaning through colors":</p>
15.   <p class="text-muted">This text is muted.</p>
16.   <p class="text-primary">This text is important.</p>
17.   <p class="text-success">This text indicates success.</p>
18.   <p class="text-info">This text represents some information.</p>
19.   <p class="text-warning">This text represents a warning.</p>
20.   <p class="text-danger">This text represents danger.</p>
21.   <p class="text-secondary">Secondary text.</p>
22.   <p class="text-dark">This text is dark grey.</p>
23.   <p class="text-body">Default body color (often black).</p>
24.   <p class="text-light">This text is light grey (on white background).</p>
25.   <p class="text-white">This text is white (on white background).</p>
26. </div>
27. </body>
28. </html>

```

Background Colors

File bg_colors_bootstrap.html

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8.   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
9. </head>
10. <body>
11. <div class="container mt-3">
12.   <h2>Macam warna background</h2>
13.   <div class="bg-primary p-3"></div>
14.   <div class="bg-success p-3"></div>
15.   <div class="bg-info p-3"></div>
16.   <div class="bg-warning p-3"></div>
17.   <div class="bg-danger p-3"></div>
18.   <div class="bg-secondary p-3"></div>
19.   <div class="bg-dark p-3"></div>
20.   <div class="bg-light p-3"></div>
21. </div>
22. </body>
23. </html>

```

Latihan Praktikum

- Silakan praktikkan contoh kode diatas

Jawab:

Berikut kode pertama jika dijalankan di web browser:

Typography

Ini merupakan contoh tulisan paragraf normal.

h1 Bootstrap heading

h2 Bootstrap heading

h3 Bootstrap heading

h4 Bootstrap heading

h5 Bootstrap heading

h6 Bootstrap heading

lowercased text.

UPPERCASED TEXT.

Capitalized Text.

This paragraph stands out.

Left-aligned text.

Right-aligned text.

Center-aligned text.

Gunakan **ctrl + c** untuk copy dokumen.

Kampus STITEK mempunyai program studi Teknik Informatika.

Ini contoh penulisan dengan style code: `span`, `section`, and `div`.

Coffee

Tea

- Black tea
- Green tea

Milk

Ini merupakan contoh penulisan dengan style quote.

— From Abadi Nugroho

Berikut Kode kedua jika dijalankan di web browser ;

Warna pada font

Use the contextual classes to provide "meaning through colors":

This text is muted.

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

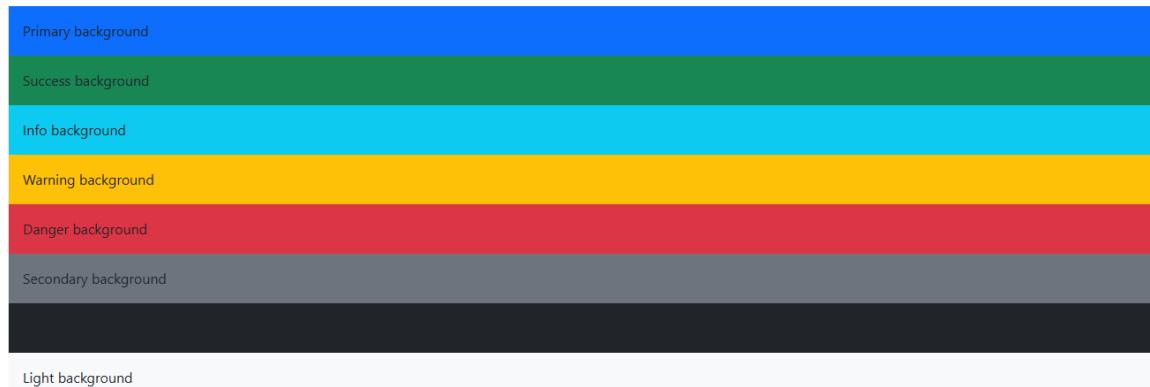
Secondary text.

This text is dark grey.

Default body color (often black).

Berikut kode ketiga dijalankan di web browser:

Macam warna background



4. Tables

Penjelasan

Pada bootstrap juga tersedia berbagai macam desain table. Antara lain sebagai berikut:

.table

.table-striped
.table-bordered
.table-hover
.table-dark
.table-borderless
.table-sm
.table-sm
.table-responsive

Contoh Kode

File basic_table_bootsrap.html

```

1. <!DOCTYPE html>
2. <html lang="en">
3.   <head>
4.     <title>Bootstrap Basic Table</title>
5.     <meta charset="utf-8">
6.     <meta name="viewport" content="width=device-width, initial-scale=1">
7.     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8.     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
9.   </head>
10.  <body>
11.    <div class="container mt-3">
12.      <h2>Basic Table</h2>
13.      <p>Ini tampilan tabel dasar pada bootstrap:</p>
14.      <table class="table">
15.        <thead>
16.          <tr>
17.            <th>Nama Depan</th>
18.            <th>Nama Belakang</th>
19.            <th>Email</th>
20.          </tr>
21.        </thead>
22.        <tbody>
23.          <tr>
24.            <td>Abadi</td>
25.            <td>Nugroho</td>
26.            <td>abadi.nugroho.mail@gmail.com</td>
27.          </tr>
28.          <tr>
29.            <td>Alvin</td>
30.            <td>Stitek</td>
31.            <td>alvin@stitek.ac.id</td>
32.          </tr>
33.          <tr>
34.            <td>Adeel</td>
35.            <td>Stitek</td>
36.            <td>adeel@stitek.ac.id</td>
37.          </tr>
38.        </tbody>
39.      </table>
40.    </div>
41.  </body>
42. </html>

```

Latihan Praktikum

- Silakan praktikkan contoh kode di atas.

Jawab:

Basic Table

Ini tampilan tabel dasar pada bootstrap:

Nama Depan	Nama Belakang	Email
Abadi	Nugroho	abadi.nugroho.mail@gmail.com
Alvin	Stitek	alvin@stitek.ac.id
Adeel	Stitek	adeel@stitek.ac.id

- Silakan coba style yang lain .table, .table-striped, .table-bordered, .table-hover, .table-dark, .table-borderless, .table-sm, .table-responsive.

Berikut kodenya:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Table Styles</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.mi
n.css" rel="stylesheet">
  <script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bund
le.min.js"></script>
</head>
<body>
  <div class="container mt-3">
    <h2>Different Bootstrap Table Styles</h2>

    <h3>.table</h3>
    <table class="table">
      <thead>
        <tr>
          <th>Nama Depan</th>
          <th>Nama Belakang</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Abadi</td>
```

```

<td>Nugroho</td>
<td>abadi.nugroho.mail@gmail.com</td>
</tr>
<tr>
<td>Alvin</td>
<td>Stitek</td>
<td>alvin@stitek.ac.id</td>
</tr>
</tbody>
</table>

<h3>.table-striped</h3>
<table class="table table-striped">
<thead>
<tr>
<th>Nama Depan</th>
<th>Nama Belakang</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abadi</td>
<td>Nugroho</td>
<td>abadi.nugroho.mail@gmail.com</td>
</tr>
<tr>
<td>Alvin</td>
<td>Stitek</td>
<td>alvin@stitek.ac.id</td>
</tr>
</tbody>
</table>

```

```

<h3>.table-bordered</h3>
<table class="table table-bordered">
<thead>
<tr>
<th>Nama Depan</th>
<th>Nama Belakang</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abadi</td>
<td>Nugroho</td>
<td>abadi.nugroho.mail@gmail.com</td>
</tr>
<tr>
<td>Alvin</td>
<td>Stitek</td>
<td>alvin@stitek.ac.id</td>
</tr>
</tbody>
</table>

<h3>.table-hover</h3>
<table class="table table-hover">
<thead>
<tr>
<th>Nama Depan</th>
<th>Nama Belakang</th>
<th>Email</th>
</tr>
</thead>

```

```

<tbody>
<tr>
<td>Abadi</td>
<td>Nugroho</td>
<td>abadi.nugroho.mail@gmail.com</td>
</tr>
<tr>
<td>Alvin</td>
<td>Stitek</td>
<td>alvin@stitek.ac.id</td>
</tr>
</tbody>
</table>

<h3>.table-dark</h3>
<table class="table table-dark">
<thead>
<tr>
<th>Nama Depan</th>
<th>Nama Belakang</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abadi</td>
<td>Nugroho</td>
<td>abadi.nugroho.mail@gmail.com</td>
</tr>
<tr>
<td>Alvin</td>
<td>Stitek</td>
<td>alvin@stitek.ac.id</td>

```

```

        </tr>
    </tbody>
</table>

<h3>.table-borderless</h3>
<table class="table table-borderless">
    <thead>
        <tr>
            <th>Nama Depan</th>
            <th>Nama Belakang</th>
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Abadi</td>
            <td>Nugroho</td>
            <td>abadi.nugroho.mail@gmail.com</td>
        </tr>
        <tr>
            <td>Alvin</td>
            <td>Stitek</td>
            <td>alvin@stitek.ac.id</td>
        </tr>
    </tbody>
</table>

<h3>.table-sm</h3>
<table class="table table-sm">
    <thead>
        <tr>
            <th>Nama Depan</th>
            <th>Nama Belakang</th>

```

```

<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abadi</td>
<td>Nugroho</td>
<td>abadi.nugroho.mail@gmail.com</td>
</tr>
<tr>
<td>Alvin</td>
<td>Stitek</td>
<td>alvin@stitek.ac.id</td>
</tr>
</tbody>
</table>

<h3>.table-responsive</h3>
<div class="table-responsive">
<table class="table">
<thead>
<tr>
<th>Nama Depan</th>
<th>Nama Belakang</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abadi</td>
<td>Nugroho</td>
<td>abadi.nugroho.mail@gmail.com</td>
</tr>

```

```
<tr>
    <td>Alvin</td>
    <td>Stitek</td>
    <td>alvin@stitek.ac.id</td>
</tr>
</tbody>
</table>
</div>

</div>
</body>
</html>
```

Berikut hasilnya jika dijalankan di web browser:

Different Bootstrap Table Styles

.table

Nama Depan	Nama Belakang	Email
Abadi	Nugroho	abadi.nugroho.mail@gmail.com
Alvin	Stitek	alvin@stitek.ac.id

.table-striped

Nama Depan	Nama Belakang	Email
Abadi	Nugroho	abadi.nugroho.mail@gmail.com
Alvin	Stitek	alvin@stitek.ac.id

.table-bordered

Nama Depan	Nama Belakang	Email
Abadi	Nugroho	abadi.nugroho.mail@gmail.com
Alvin	Stitek	alvin@stitek.ac.id

.table-hover

Nama Depan	Nama Belakang	Email
Abadi	Nugroho	abadi.nugroho.mail@gmail.com
Alvin	Stitek	alvin@stitek.ac.id

.table-dark

Nama Depan	Nama Belakang	Email
Abadi	Nugroho	abadi.nugroho.mail@gmail.com

.table-borderless

Nama Depan	Nama Belakang	Email
Abadi	Nugroho	abadi.nugroho.mail@gmail.com
Alvin	Stitek	alvin@stitek.ac.id

.table-sm

Nama Depan	Nama Belakang	Email
Abadi	Nugroho	abadi.nugroho.mail@gmail.com
Alvin	Stitek	alvin@stitek.ac.id

.table-responsive

Nama Depan	Nama Belakang	Email
Abadi	Nugroho	abadi.nugroho.mail@gmail.com
Alvin	Stitek	alvin@stitek.ac.id

- Analisa hasil praktikum :

Kode ini menunjukkan berbagai fitur yang disediakan oleh Bootstrap untuk mendesain tabel dengan cara yang fleksibel dan responsif. Dengan menerapkan berbagai kelas Bootstrap seperti .table, .table-striped, .table-bordered, dll., pengguna dapat dengan mudah membuat tabel yang lebih menarik dan fungsional sesuai dengan kebutuhan tampilan atau interaksi yang diinginkan. Penggunaan .table-responsive juga memastikan bahwa tabel tetap responsif di berbagai perangkat, meningkatkan pengalaman pengguna di berbagai ukuran layar.

5. Komponen Bootstrap 5

Penjelasan

Ada berbagai macam komponen yang bisa kita gunakan :

Contoh Kode

Button

File buuton_bootstrap.html

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8.   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
9. </head>
10. <body>
11. <div class="container mt-3">
12.   <h2>Button Styles</h2>
13.   <button type="button" class="btn">Basic</button>
14.   <button type="button" class="btn btn-primary">Primary</button>
15.   <button type="button" class="btn btn-secondary">Secondary</button>
16.   <button type="button" class="btn btn-success">Success</button>
17.   <button type="button" class="btn btn-info">Info</button>
18.   <button type="button" class="btn btn-warning">Warning</button>
19.   <button type="button" class="btn btn-danger">Danger</button>
20.   <button type="button" class="btn btn-dark">Dark</button>
21.   <button type="button" class="btn btn-light">Light</button>
22.   <button type="button" class="btn btn-link">Link</button>
23. </div>
24. </body>
25. </html>
```

Navbars

File navbar_bootstrap.html

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8.   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
9. </head>
10. <body>
11. <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
12.   <div class="container-fluid">
13.     <a class="navbar-brand" href="javascript:void(0)">Logo</a>
14.     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#mynavbar">
15.       <span class="navbar-toggler-icon"></span>
16.     </button>
17.     <div class="collapse navbar-collapse" id="mynavbar">
18.       <ul class="navbar-nav me-auto">
19.         <li class="nav-item">
20.           <a class="nav-link" href="javascript:void(0)">Home</a>
21.         </li>
22.         <li class="nav-item">
23.           <a class="nav-link" href="javascript:void(0)">Profil</a>
24.         </li>
25.         <li class="nav-item">
26.           <a class="nav-link" href="javascript:void(0)">About Us</a>
27.         </li>
28.         <li class="nav-item dropdown">
29.           <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown">Dropdown</a>
30.             <ul class="dropdown-menu">
31.               <li><a class="dropdown-item" href="#">Link</a></li>
32.               <li><a class="dropdown-item" href="#">Another link</a></li>
33.               <li><a class="dropdown-item" href="#">A third link</a></li>
34.             </ul>
35.           </li>
36.         </ul>
37.         <form class="d-flex">
38.           <input class="form-control me-2" type="text" placeholder="Search">
39.           <button class="btn btn-primary" type="button">Search</button>
40.         </form>
41.       </div>
42.     </div>
43.   </nav>
44.   <div class="container-fluid mt-3">
45.     <h3>Navbar Forms</h3>
46.     <p>You can also include forms inside the navigation bar.</p>
47.   </div>
48. </body>
49. </html>
```

Modal

File modal_bootstrap.html

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Modal</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8.   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
9. </head>
10. <body>
11.   <div class="container mt-3">
12.     <h3>Fading Modal Example</h3>
13.     <p>Click on the button to open the modal.</p>
14.     <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#myModal">
15.       Open modal
16.     </button>
17.   </div>
18.   <!-- The Modal -->
19.   <div class="modal fade" id="myModal">
20.     <div class="modal-dialog">
21.       <div class="modal-content">
22.
23.         <!-- Modal Header -->
24.         <div class="modal-header">
25.           <h4 class="modal-title">Modal Heading</h4>
26.           <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
27.         </div>
28.
29.         <!-- Modal body -->
30.         <div class="modal-body">
31.           Modal body..
32.         </div>
33.
34.         <!-- Modal footer -->
35.         <div class="modal-footer">
36.           <button type="button" class="btn btn-danger" data-bs-dismiss="modal">Close</button>
37.         </div>
38.
39.       </div>
40.     </div>
41.   </body>
42. </html>
```

Pagination

File pagination_bootstrap.html

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap pagination</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8.   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
9. </head>
10. <body>
11. <div class="container mt-3">
12.   <h2>Pagination</h2>
13.   <p>To create a basic pagination, add the .pagination class to an ul element. Then add the .page-item to each li element and a .page-link class to each link inside li:</p>
14.   <ul class="pagination">
15.     <li class="page-item"><a class="page-link" href="#">Previous</a></li>
16.     <li class="page-item"><a class="page-link" href="#">1</a></li>
17.     <li class="page-item"><a class="page-link" href="#">2</a></li>
18.     <li class="page-item"><a class="page-link" href="#">3</a></li>
19.     <li class="page-item"><a class="page-link" href="#">Next</a></li>
20.   </ul>
21. </div>
22.
23. </body>
24. </html>
```

Card

File card_bootstrap.html

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8.   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
9. </head>
10. <body>
11.   <div class="container mt-3">
12.     <h2>Card Image</h2>
13.     <p>Image at the top (card-img-top):</p>
14.     <div class="card" style="width:400px">
15.       
16.       <div class="card-body">
17.         <h4 class="card-title">John Doe</h4>
18.         <p class="card-text">Some example text some example text. John Doe is an architect and engineer</p>
19.         <a href="#" class="btn btn-primary">See Profile</a>
20.       </div>
21.     </div>
22.     <br>
23.     <p>Image at the bottom (card-img-bottom):</p>
24.     <div class="card" style="width:400px">
25.       <div class="card-body">
26.         <h4 class="card-title">Jane Doe</h4>
27.         <p class="card-text">Some example text some example text. Jane Doe is an architect and engineer</p>
28.         <a href="#" class="btn btn-primary">See Profile</a>
29.       </div>
30.       
31.     </div>
32.   </div>
33. </body>
34. </html>
```

- Silakan praktikkan contoh kode di atas.

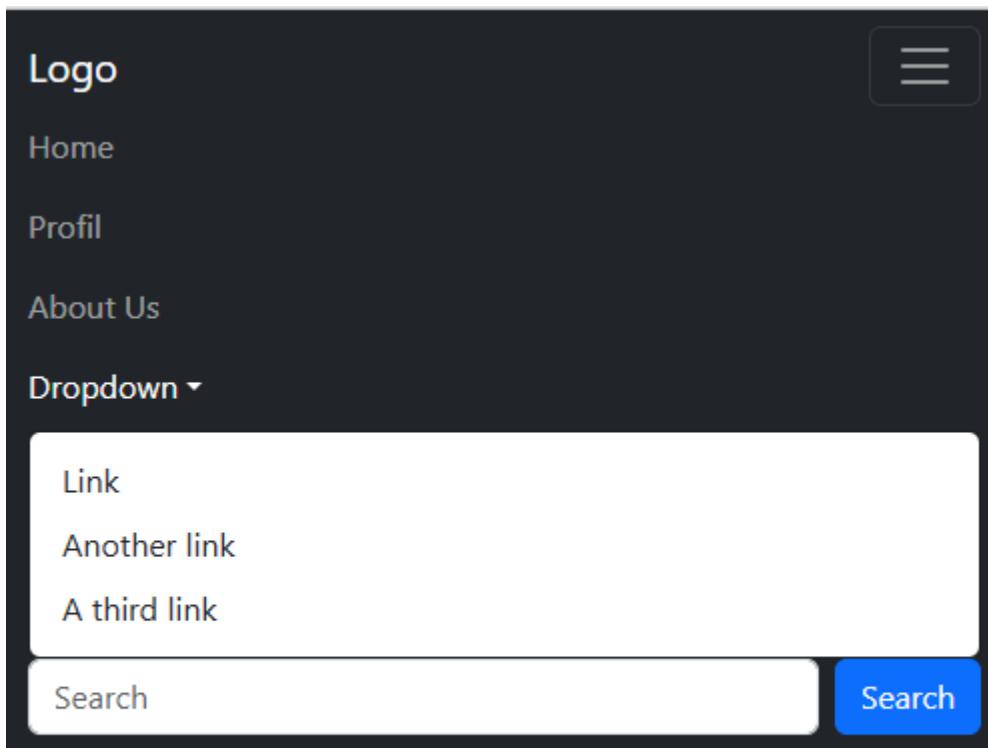
Button:

Button Styles

Basic Primary Secondary Success Info Warning Danger Dark Light [Link](#)

Navbar:





Navbar Forms

You can also include forms inside the navigation bar.

Modal:

1. 2. 3. 4. 5. 6. 7. 8.

Pagination:

Pagination

To create a basic pagination, add the .pagination class to an ul element. Then add the .page-item to each li element and a .page-link class to each link inside li:

Previous 1 2 3 Next

Card:

Card Image

Image at the top (card-img-top):

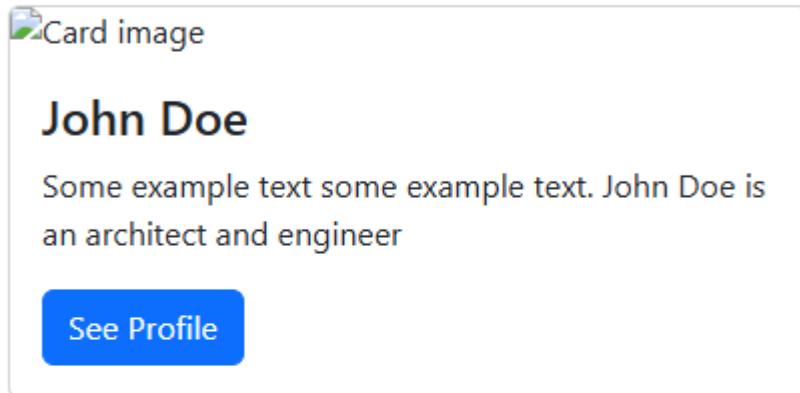
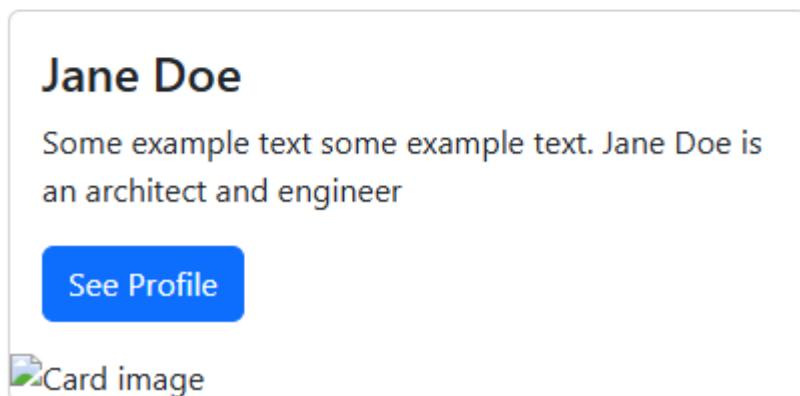


Image at the bottom (card-img-bottom):



- Pada bagian card silahkan pasang foto kalian masing-masing simpan dalam satu folder yang sama dengan file script card_bootstrap.htm

Berikut kodennya:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```

<link
  href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script
  src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
  <div class="container mt-3">
    <h2>Card Image</h2>
    <p>Image at the top (card-img-top):</p>
    <div class="card" style="width:400px">
      <!-- Gantilah 'foto_saya.png' dengan nama file foto Anda -->
      
      <div class="card-body">
        <h4 class="card-title">John Doe</h4>
        <p class="card-text">Some example text some example text. John Doe is an architect and engineer</p>
        <a href="#" class="btn btn-primary">See Profile</a>
      </div>
    </div>
    <br>
    <p>Image at the bottom (card-img-bottom):</p>
    <div class="card" style="width:400px">
      <div class="card-body">
        <h4 class="card-title">Jane Doe</h4>
        <p class="card-text">Some example text some example text. Jane Doe is an architect and engineer</p>
        <a href="#" class="btn btn-primary">See Profile</a>
      </div>
    </div>
    <!-- Gantilah 'foto_saya.png' dengan nama file foto Anda -->
  </div>
</body>

```

```
>  
</div>  
</div>  
</body>  
</html>
```

Dan hasil screenshot:

Card Image

Image at the top (card-img-top):



John Doe

Some example text some example text. John Doe is an architect and engineer

[See Profile](#)

Image at the bottom (card-img-bottom):

Jane Doe

Some example text some example text. Jane Doe is an architect and engineer

[See Profile](#)



- Analisa:

Bootstrap menyediakan komponen UI siap pakai seperti **button**, **navbar**, **modal**, **pagination**, dan **card** yang memungkinkan pembuatan halaman web responsif dengan desain modern. Setiap komponen menggunakan kelas CSS yang sudah ditentukan untuk memastikan konsistensi visual dan fungsionalitas, dengan kemampuan responsif yang kuat untuk memastikan pengalaman pengguna yang baik di berbagai perangkat.

D. Tugas Praktikum

Buatlah sebuah halaman web company profile dengan ketentuan sebagai berikut:

1. Struktur Halaman

Header

Berisi judul website dan navigasi utama (navbar) dengan beberapa menu (misal: Home, About, Services, Contact).

Navbar

Navbar harus responsif, artinya pada layar kecil berubah menjadi tombol hamburger yang dapat diklik untuk menampilkan menu.

Konten Utama

Tabel Informasi Perusahaan Buat tabel yang berisi data penting perusahaan, misalnya: Nama Perusahaan, Tahun Berdiri, Jumlah Karyawan, dan Lokasi Kantor.

Card Produk/Layanan Buat minimal 3 card yang menampilkan produk atau layanan perusahaan. Setiap card berisi gambar, judul, deskripsi singkat, dan tombol aksi (misal: "Selengkapnya").

Footer

Berisi informasi hak cipta, kontak, atau link sosial media.

2. Desain dan Styling

- Gunakan CSS atau framework seperti Bootstrap 5 untuk mempercantik tampilan.
- Pastikan halaman bersifat responsif dan dapat tampil baik di berbagai ukuran layar (desktop, tablet, dan smartphone).
- Gunakan warna, margin, padding, dan font yang konsisten dan menarik.

3. Interaktivitas

- Tombol pada card harus memiliki efek hover (misal perubahan warna).
- Navbar harus dapat berfungsi sebagai menu dropdown pada layar kecil (hamburger menu).

4. Tambahan

- Simpan kode di github anda masing-masing.
- Buatlah video untuk menjelaskan script dari tugas yang anda buat, simpan di penyimpanan online misalnya youtube anda masing-masing dan sertakan link github dan video di laporan anda.

5. Rubrik Penilaian

Aspek Penilaian	Keterangan	Bobot (%)
Struktur HTML	Header, navbar, konten utama, footer lengkap dan sesuai instruksi	10%
Penggunaan Tabel	Tabel informasi perusahaan dibuat dengan benar dan rapi	15%
Penggunaan Card	Card produk/layanan dibuat dengan gambar, teks, dan tombol yang sesuai	10%
Styling dan Responsivitas	Tampilan menarik, tombol interaktif, dan halaman responsif di berbagai perangkat	10%
Kerapihan Kode	Kode terstruktur, rapi, dan mudah dibaca	15%
Kreativitas	Penambahan fitur atau desain tambahan yang memperkaya tampilan	10%
Dokumentasi	Disertakan penjelasan dalam video dan disimpan di github	30%

Jawab:

Kode :

```
<!DOCTYPE html>
<html lang="id">

<head>
    <title>Company Profile - Pupuk Marton</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
<style>
  .card-img-top {
    height: 200px;
    object-fit: cover;
  }

  .card:hover {
    box-shadow: 0 0 15px rgba(0, 0, 0, 0.1);
    transform: translateY(-5px);
    transition: transform 0.3s, box-shadow 0.3s;
  }

  footer {
    background-color: #f8f9fa;
    padding: 20px;
    text-align: center;
  }

  .navbar-toggler-icon {
    background-color: black;
  }

  .navbar-nav li {
    margin-right: 20px;
  }

  /* Hover effect for table rows */
  .table-hover tbody tr:hover {
    background-color: #f1f1f1;
  }
</style>
</head>

<body>
  <!-- Header -->
  <header class="bg-dark text-white text-center py-3">
    <h1>Pupuk Marton</h1>
    <p>Perusahaan Pupuk Terpercaya dan Berkualitas</p>
  </header>

  <!-- Navbar -->
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">Pupuk Marton</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav"
        aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
        navigation">

```

```

<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
  <ul class="navbar-nav ms-auto">
    <li class="nav-item">
      <a class="nav-link active" href="#">Home</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">About</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Services</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Contact</a>
    </li>
  </ul>
</div>
</div>
</nav>

<!-- Konten Utama -->
<div class="container mt-5">
  <h2 class="text-center mb-4">Tabel Informasi Perusahaan</h2>

  <!-- Tabel Informasi Perusahaan -->
  <table class="table table-bordered">
    <thead>
      <tr>
        <th>Nama Perusahaan</th>
        <th>Tahun Berdiri</th>
        <th>Jumlah Karyawan</th>
        <th>Lokasi Kantor</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Pupuk Marton</td>
        <td>1980</td>
        <td>1.000</td>
        <td>Jakarta, Indonesia</td>
      </tr>
    </tbody>
  </table>

  <!-- Tabel Cabang Perusahaan (Hoverable) -->
  <h2 class="text-center my-4">Cabang Perusahaan Pupuk Marton</h2>
  <table class="table table-hover table-bordered">
    <thead>
      <tr>

```

```

<th>Nama Cabang</th>
<th>Lokasi</th>
<th>Tahun Berdiri</th>
<th>Jumlah Karyawan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pupuk Marton Surabaya</td>
<td>Surabaya, Jawa Timur</td>
<td>1995</td>
<td>250</td>
</tr>
<tr>
<td>Pupuk Marton Medan</td>
<td>Medan, Sumatera Utara</td>
<td>2000</td>
<td>180</td>
</tr>
<tr>
<td>Pupuk Marton Makassar</td>
<td>Makassar, Sulawesi Selatan</td>
<td>2010</td>
<td>150</td>
</tr>
</tbody>
</table>

<!-- Card Produk/Layanan -->
<h3 class="text-center my-4">Produk & Layanan Kami</h3>
<div class="row">
<div class="col-md-4">
<div class="card">

<div class="card-body">
<h5 class="card-title">Pupuk Urea</h5>
<p class="card-text">Pupuk Urea kami sangat efektif untuk meningkatkan hasil pertanian dan kesuburan tanah.</p>
<a href="#" class="btn btn-primary">Selengkapnya</a>
</div>
</div>
</div>
<div class="col-md-4">
<div class="card">

<div class="card-body">
<h5 class="card-title">Pupuk NPK</h5>
<p class="card-text">Pupuk NPK kami mengandung unsur hara lengkap untuk tanaman yang sehat dan subur.</p>
<a href="#" class="btn btn-primary">Selengkapnya</a>

```

```

        </div>
        </div>
        </div>
<div class="col-md-4">
<div class="card">

<div class="card-body">
<h5 class="card-title">Pupuk Organik</h5>
<p class="card-text">Pupuk Organik kami membantu meningkatkan kualitas tanah secara alami tanpa bahan kimia berbahaya.</p>
<a href="#" class="btn btn-primary">Selengkapnya</a>
</div>
</div>
</div>
</div>
</div>

<!-- Footer -->
<footer>
<p>&copy; 2025 Pupuk Marton - Perusahaan Pupuk Terpercaya</p>
<p>Kontak: info@pupukmarton.co.id | <a href="#">Facebook</a> | <a href="#">Twitter</a></p>
</footer>

<!-- Script Bootstrap -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
</body>

</html>

```

Tampilan jika dijalankan di web browser dengan tampilan lebar:

Pupuk Marton
Perusahaan Pupuk Terpercaya dan Berkualitas

Pupuk Marton

Home About Services Contact

Tabel Informasi Perusahaan

Nama Perusahaan	Tahun Berdiri	Jumlah Karyawan	Lokasi Kantor
Pupuk Marton	1980	1.000	Jakarta, Indonesia

Cabang Perusahaan Pupuk Marton

Nama Cabang	Lokasi	Tahun Berdiri	Jumlah Karyawan
Pupuk Marton Surabaya	Surabaya, Jawa Timur	1995	250
Pupuk Marton Medan	Medan, Sumatera Utara	2000	180
Pupuk Marton Makassar	Makassar, Sulawesi Selatan	2010	150

Produk & Layanan Kami



© 2025 Pupuk Marton - Perusahaan Pupuk Terpercaya
Kontak: info@pupukmarton.co.id | [Facebook](#) | [Twitter](#)

Tampilan di layar tidak lebar:

Pupuk Marton
Perusahaan Pupuk Terpercaya dan Berkualitas

Pupuk Marton

≡

Tabel Informasi Perusahaan

Nama Perusahaan	Tahun Berdiri	Jumlah Karyawan	Lokasi Kantor
Pupuk Marton	1980	1.000	Jakarta, Indonesia

Cabang Perusahaan Pupuk Marton

Nama Cabang	Lokasi	Tahun Berdiri	Jumlah Karyawan
Pupuk Marton Surabaya	Surabaya, Jawa Timur	1995	250
Pupuk Marton Medan	Medan, Sumatera Utara	2000	180
Pupuk Marton Makassar	Makassar, Sulawesi Selatan	2010	150

Produk & Layanan Kami



Pupuk Urea

Pupuk Urea kami sangat efektif untuk meningkatkan hasil pertanian dan kesuburan tanah.

[Selengkapnya](#)





Pupuk NPK

Pupuk NPK kami mengandung unsur hara lengkap untuk tanaman yang sehat dan subur.

[Selengkapnya](#)



Pupuk Organik

Pupuk Organik kami membantu meningkatkan kualitas tanah secara alami tanpa bahan kimia berbahaya.

[Selengkapnya](#)

© 2025 Pupuk Marton - Perusahaan Pupuk Terpercaya

Kontak: info@pupukmarton.co.id | [Facebook](#) | [Twitter](#)

Berikut link github video penjelasan video: [github](#)

Modul 4. Javascript

A. Tujuan

- Setelah mengikuti praktikum, Mahasiswa diharapkan mampu:
1. Mampu memahami konsep dasar dan manipulasi DOM dengan JavaScript.
 2. Mampu menangani interaksi pengguna menggunakan *event handling*.
 3. Dapat menerapkan logika percabangan dan perulangan untuk membuat halaman web dinamis.
 4. Mampu membuat validasi form dan kalkulasi sederhana secara *real-time*.
 5. Bisa bekerja dengan objek-objek bawaan JavaScript seperti Array, String, Math dan Date.
 6. Dapat memanipulasi elemen gambar secara dinamis.

B. Teori Dasar Bootsrap

JavaScript adalah bahasa pemrograman yang membuat halaman web menjadi hidup dan interaktif. Berbeda dengan HTML yang hanya menyusun struktur dan CSS yang mengatur tampilan, JavaScript dapat mengubah konten HTML dan CSS secara langsung di browser tanpa perlu memuat ulang halaman. Kemampuan ini didasarkan pada pilar-pilar utama seperti Manipulasi DOM, Event Handling, dan penerapan Logika Pemrograman. Selain itu, JavaScript menyediakan berbagai Objek Bawaan (Built-in Objects) yang sangat kuat untuk memanipulasi data seperti teks (String), angka (Math), dan waktu (Date).

C. Pratikum

1. Manipulasi DOM (Document Object Model)

a. Penjelasan :

DOM adalah representasi dari dokumen HTML dalam bentuk objek. Dengan Javascrpit, kita bisa berinteraksi dengan objek ini untuk memanipulasi selemen.. Cara paling umum adalah memilih elemen berdasarkan atribut id nya

menggunakan `document.getElementById()`, lalu mengubah konten di dalamnya dengan properti `.innerHTML` atau mengubah gayanya dengan properti `.style`.

b. Contoh Kode

File: manipulasi_dom.html

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>Latihan Manipulasi DOM</title>
5. <meta charset="utf-8">

6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <style>
8. body { font-family: sans-serif; text-align: center; margin-top: 50px; }
9. #judul { color: steelblue; }
10. button { padding: 10px 15px; font-size: 16px; cursor: pointer; }
11. </style>
12. </head>
13. <body>
14. <h1 id="judul">Selamat Datang di Praktikum JavaScript</h1>
15. <button onclick="ubahTeks()">Ubah Teks Judul</button>
16. <script>
17. function ubahTeks() {
18.     var elemenJudul = document.getElementById("judul");
19.     elemenJudul.innerHTML = "Teks Berhasil Diubah!";
20.     elemenJudul.style.color = "crimson";
21. }
22. </script>
23. </body>
24. </html>
```

c. Latihan Pratikum

- Praktikkan contoh kode di atas.
- Berikan capture (tangkapan layar) sebelum dan sesudah tombol "Ubah Teks Judul" diklik

Teks Berhasil Diubah!

Ubah Teks Judul

- Modifikasi fungsi `ubahTeks()` untuk menambahkan perubahan lain, misalnya mengubah ukuran (`elemenJudul.style.fontSize = "36px";`). Font

Teks Berhasil Diubah!

Ubah Teks Judul

Dengan code:

```
function ubahTeks() {  
    var elemenJudul = document.getElementById("judul");  
  
    // Mengubah isi teks  
    elemenJudul.innerHTML = "Teks Berhasil Diubah!";  
  
    // Mengubah warna teks  
    elemenJudul.style.color = "crimson";  
  
    // Mengubah ukuran font  
    elemenJudul.style.fontSize = "36px";  
  
    // Menambahkan huruf tebal  
    elemenJudul.style.fontWeight = "bold";  
  
    // Mengubah jenis font  
    elemenJudul.style.fontFamily = "Georgia, serif";  
}
```

- Analisis dari hasil praktikum anda di laporan.

Pada praktikum kali ini, kami mempelajari cara memanipulasi elemen HTML secara dinamis menggunakan JavaScript melalui DOM (Document Object Model). Fokus utama adalah pada penggunaan getElementById untuk mengakses elemen tertentu, serta metode innerHTML dan style untuk mengubah isi dan tampilannya.

Setelah fungsi ubahTeks() dimodifikasi, terlihat bahwa perubahan tidak hanya terbatas pada isi teks, tetapi juga mencakup gaya visual seperti warna, ukuran, ketebalan huruf, dan jenis font. Hal ini membuktikan bahwa JavaScript sangat fleksibel dalam mengontrol elemen halaman web secara real-time tanpa harus me-reload halaman.

Dengan latihan ini, kami memahami bahwa:

- DOM memungkinkan interaksi dinamis dengan elemen HTML.
- JavaScript dapat mengubah konten, atribut, dan gaya CSS secara langsung.
- Perubahan kecil pada DOM dapat memberikan pengaruh besar terhadap tampilan dan pengalaman pengguna.

Kesimpulan: Praktikum ini berhasil menunjukkan penerapan dasar manipulasi DOM dan membuka wawasan mengenai pengembangan antarmuka web yang lebih interaktif.

2. Penanganan Kejadian (*Event Handling*)

a. Penjelasan :

Event adalah aksi yang terjadi di halaman web. JavaScript dapat “mendengarkan” event ini dan menjalankan fungsi tertentu ketika terjadi. Event yang umum digunakan adalah onclick, onmouseover, onchange, dan onfocus.

b. Contoh Kode

File: event_handling.html

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Latihan Event Handling</title>
5.   <style>
6.     #inputNama { border: 2px solid #ccc; padding: 8px; transition: border-color 0.3s; }
7.     #inputNama:focus { border-color: dodgerblue; }
8.   </style>
9. </head>
10. <body>
11.   <h3>Event onmouseover & onmouseout</h3>
```

```

12.      
14.      <hr>
15.      <h3>Event onfocus & onchange</h3>
16.      <label for="inputNama">Ketik Nama Anda:</label>
17.      <input type="text" id="inputNama" onchange="tampilkanOutput()>
18.      <p>Output: <span id="hasil"></span></p>
19.      <script>
20.          function gantiGambar() { document.getElementById("gambar").src =
21.              "https://placehold.co/200x150/3498db/fff?text=Gambar+Kedua"; }
22.          function kembalikanGambar() { document.getElementById("gambar").src =
23.              "https://placehold.co/200x150/e8e8e8/000?text=Gambar+Awal"; }
24.          function tampilkanOutput() {
25.              var nilaiInput = document.getElementById("inputNama").value;
26.              document.getElementById("hasil").innerHTML = "<b>" + nilaiInput + "</b>";
27.          }
28.      </script>
29.      </body>
30.  </html>

```

c. Latihan Pratikum

- Pratikkan contoh kode di atas.
- Berikan *capture* yang menunjukkan tampilan saat mouse di atas gambar dan setelah Anda mengetik di input

Event onmouseover & onmouseout



Event onfocus & onchange

Ketik Nama Anda:
berikut:

Hasil modifikasi:

Event onmouseover & onmouseout



Gambar Kedua

Event onfocus & onchange

Ketik Nama Anda: Martonius Epta Sidabutar

Output: **Martonius Epta SIDabutar**

Dengan code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Latihan Event Handling</title>
    <style>
        #inputNama {
            border: 2px solid #ccc;
            padding: 8px;
            transition: border-color 0.3s;
        }

        #inputNama:focus {
            border-color: dodgerblue;
        }

        /* Gaya dasar gambar */
        #gambar {
            transition: all 0.4s ease;
        }
    </style>
</head>
<body>
    <input type="text" id="inputNama" value="Martonius Epta Sidabutar" />
    <img alt="Blue rounded rectangle button labeled 'Gambar Kedua'" id="gambar" data-bbox="288 134 554 278"/>
</body>
</html>
```

```

border-radius: 8px;
}

/* Gaya saat mouse berada di atas gambar */
#gambar:hovered {
    transform: scale(1.05);
    box-shadow: 0 8px 16px rgba(0, 0, 0, 0.3);
}

</style>
</head>
<body>

<!-- Bagian 1: Event onmouseover dan onmouseout -->
<h3>Event onmouseover & onmouseout</h3>


<hr>

<!-- Bagian 2: Event onfocus dan onchange -->
<h3>Event onfocus & onchange</h3>
<label for="inputNama">Ketik Nama Anda:</label>
<input type="text" id="inputNama" onchange="tampilkanOutput()">
<p>Output: <span id="hasil"></span></p>

<script>
function gantiGambar() {
    const gambar = document.getElementById("gambar");

```

```

gambar.src =
"https://placehold.co/200x150/3498db/fff?text=Gambar+Kedua";
gambar.classList.add("hovered");
}

function kembalikanGambar() {
const gambar = document.getElementById("gambar");
gambar.src =
"https://placehold.co/200x150/e8e8e8/000?text=Gambar+Awal";
gambar.classList.remove("hovered");
}

function tampilkanOutput() {
const nilaiInput = document.getElementById("inputNama").value;
document.getElementById("hasil").innerHTML = "<b>" +
nilaiInput + "</b>";
}
</script>

</body>
</html>

```

- Analisis dari hasil praktikum Anda di laporan.
Latihan ini menunjukkan penerapan **event handling** JavaScript (onmouseover, onmouseout, onchange) untuk memanipulasi elemen HTML secara interaktif. Saat kursor diarahkan ke gambar, tampilannya berubah melalui **pergantian gambar dan efek CSS** seperti bayangan dan perbesaran. Input teks juga merespons perubahan dengan menampilkan hasil secara langsung. Dengan menggabungkan **JavaScript dan CSS**, pengalaman pengguna menjadi lebih menarik dan dinamis. Praktikum ini berhasil membuktikan pentingnya DOM dan event dalam pengembangan web yang interaktif dan responsif.

3. Logika Percabangan (Studi Kasus: Cek Diskon Belanja)

a. Penjelasan

Struktur kontrol if...else memungkinkan program untuk menjalankan blok kode yang berbeda berdasarkan kondisi tertentu. Ini adalah dasar dari pengambilan keputusan dalam pemrograman

b. Contoh Kode

File: logika_percabangan.html

```
1. <!DOCTYPE html>
2. <html>
3. <head><title>Latihan Logika Percabangan</title></head>
4. <body>
5. <h2>Kalkulator Diskon Belanja</h2>
6. <p>Masukkan total belanja Anda:</p>
7. <input type="number" id="totalBelanja" placeholder="Contoh: 150000">
8. <button onclick="cekDiskon()">Cek Diskon</button>
9. <h3 id="hasil"></h3>
10.
11. <script>
12.   function cekDiskon() {
13.     var total = parseFloat(document.getElementById("totalBelanja").value);
14.     var pesan = "";
15.     if (isNaN(total)) {
16.       pesan = "Mohon masukkan angka yang valid.";
17.     } else if (total >= 100000) {
18.       var diskon = total * 0.10;
19.       var totalBayar = total - diskon;
20.       pesan = "Selamat! Anda dapat diskon 10% sebesar Rp " + diskon + ". Total bayar: <b>Rp " +
totalBayar + "</b>";
21.     } else {
22.       pesan = "Maaf, tidak ada diskon. Total bayar: <b>Rp " + total + "</b>";
23.     }
24.     document.getElementById("hasil").innerHTML = pesan;
25.   }
26. </script>
27. </body>
28. </html>
```

c. Latihan Pratikum

- Praktikkan contoh kode di atas.
- Lakukan dua pengujian belanja di atas dan di bawah 100000. Berikan *capture*-nya

Kalkulator Diskon Belanja

Masukkan total belanja Anda:

Selamat! Anda dapat diskon 10% sebesar Rp 10.000. Total bayar: Rp 90.000

- Modifikasi kode untuk menambahkan diskon 15% jika belanja di atas 250000

Kalkulator Diskon Belanja

Masukkan total belanja Anda:

Selamat! Anda dapat diskon 15% sebesar Rp 375,000. Total bayar: Rp 2,125,000

- Analisis dari hasil praktikum Anda di laporan

Analisa : Logika percabangan dalam JavaScript sangat efektif untuk membuat keputusan dinamis berdasarkan kondisi input. Latihan ini berhasil menunjukkan bagaimana struktur kontrol dapat digunakan untuk menyederhanakan proses perhitungan dan meningkatkan pengalaman pengguna dalam aplikasi web sederhana.

4. Perulangan untuk membuat daftar

a. Penjelasan

Perulangan (loop) digunakan untuk menjalankan kode yang sama berulang kali. Perulangan for sangat efektif untuk mengelolah data yang ada di dalam array dan menampilkannya sebagai daftar (list) di HTML.

b. Contoh Kode

File: perulangan_daftar.html

```

1. <!DOCTYPE html>
2. <html>
3. <head><title>Latihan Perulangan</title></head>
4. <body>
5. <h2>Daftar Mata Kuliah</h2>
6. <ul id="daftar-matkul">
7.   <!-- Konten akan diisi oleh JavaScript -->
8. </ul>
9.
10. <script>
11. var matkul = ["Pemrograman Web", "Struktur Data", "Basis Data", "Jaringan Komputer"];
12. var daftarElement = document.getElementById("daftar-matkul");
13. for (let i = 0; i < matkul.length; i++) {
14.   daftarElement.innerHTML += "<li>" + matkul[i] + "</li>";
15. }
16. </script>
17.</body>
18.</html>

```

c. Latihan Praktikum

- Pratikkan contoh di atas. Jawab:

Daftar Mata Kuliah

- Pemrograman Web
 - Struktur Data
 - Basis Data
 - Jaringan Komputer
- Ganti isi dari array matkul dengan daftar nama teman sekelas Anda (minimal 5 nama). Amati perubahannya dan berikan *capture*.

Daftar Mata Kuliah

- Andi
 - Annysa
 - Gempur
 - Martonius
 - Iwan Hadi
- Analisis dari hasil praktikum anda di laporan.
- Berikut analisanya: Tujuan
- Memahami bagaimana JavaScript menggunakan perulangan (for) untuk mengakses dan menampilkan isi array ke dalam elemen HTML.



Data array berisi daftar teman sekelas.

Melalui for loop, setiap elemen array ditambahkan sebagai item dalam daftar .

Tampilan halaman berubah secara otomatis sesuai isi array.

Hasil

Halaman berhasil menampilkan seluruh nama yang ada di dalam array.

Perubahan isi array secara langsung memengaruhi konten yang ditampilkan tanpa perlu mengubah struktur HTML.

Kesimpulan

Latihan ini menunjukkan bahwa JavaScript dapat secara dinamis menampilkan data dalam jumlah berapa pun menggunakan perulangan. Praktikum ini juga memperkuat pemahaman bahwa DOM dapat dimodifikasi dengan script, menjadikannya dasar penting untuk membangun aplikasi web interaktif.

5. Validasi Form Sederhana

a. Penjelasan

Validasi memastikan pengguna memasukan data yang benar sebelum form diproses. Kita bisa menggunakan if untuk memeriksa apakah sebuah input kosong ("") dan memberikan peringatan kepada pengguna.

b. Contoh Kode

File: validasi_form.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Latihan Validasi Form</title>
<style>
```

```

.error {
    color: red;
    font-size: 0.9em;
}

</style>
</head>
<body>

<h2>Form Registrasi</h2>
<form onsubmit="return validasi()">
    <label for="nama">Nama Lengkap:</label><br>
    <input type="text" id="nama" name="nama"><br>
    <small id="nama-error" class="error"></small><br><br>
    <input type="submit" value="Daftar">
</form>

<script>
function validasi() {
    var inputNama = document.getElementById("nama").value;
    var errorElement = document.getElementById("nama-error");
    if (inputNama == "") {
        errorElement.innerHTML = "Nama tidak boleh kosong!";
        return false; // Mencegah form dikirim
    }
    errorElement.innerHTML = ""; // Menghapus pesan error jika valid
    alert("Registrasi berhasil!");
    return true;
}
</script>

</body>
</html>

```

C. Latihan Pratikum

- Pratikkan contoh di atas. Coba klik tombol “Daftar” saat input masih kosong dan berikan *capture*. Berikut hasilnya:

Form Registrasi

Nama Lengkap:

Nama tidak boleh kosong!

Daftar

- Tambahkan satu input lagi untuk “Email” dan buat validasi untuk memastikan email tidak kosong. Berikut kodenya:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Latihan Validasi Form</title>
<style>
.error {
    color: red;
    font-size: 0.9em;
}
</style>
</head>
<body>

<h2>Form Registrasi</h2>
<form onsubmit="return validasi()">
    <!-- Input Nama -->
    <label for="nama">Nama Lengkap:</label><br>
    <input type="text" id="nama" name="nama"><br>
    <small id="nama-error" class="error"></small><br><br>
```

```

<!-- Input Email -->
<label for="email">Email:</label><br>
<input type="text" id="email" name="email"><br>
<small id="email-error" class="error"></small><br><br>

<!-- Tombol Submit -->
<input type="submit" value="Daftar">
</form>

<script>
    function validasi() {
        var inputNama = document.getElementById("nama").value;
        var inputEmail = document.getElementById("email").value;
        var errorNama = document.getElementById("nama-error");
        var errorEmail = document.getElementById("email-error");

        var valid = true;

        // Validasi Nama
        if (inputNama.trim() === "") {
            errorNama.innerHTML = "Nama tidak boleh kosong!";
            valid = false;
        } else {
            errorNama.innerHTML = "";
        }

        // Validasi Email
        if (inputEmail.trim() === "") {
            errorEmail.innerHTML = "Email tidak boleh kosong!";
            valid = false;
        } else {
            errorEmail.innerHTML = "";
        }
    }

```

```

// Jika valid, munculkan alert
if (valid) {
    alert("Registrasi berhasil!");
}

return valid; // Return true jika semua input valid
}
</script>

</body>
</html>

```

Form Registrasi

Nama Lengkap:

Nama tidak boleh kosong!

Email:

Email tidak boleh kosong!

Daftar

- Analisis dari hasil praktikum anda di laporan. Analisa:

⌚ Tujuan

Praktikum ini bertujuan untuk mengimplementasikan validasi form menggunakan JavaScript agar pengguna tidak dapat mengirimkan form tanpa mengisi **nama** dan **email**.

⚙️ Hasil Implementasi

Setelah input email ditambahkan dan divalidasi, sistem form kini:

- Menolak pengiriman jika **nama atau email kosong**.

- Memberi **pesan kesalahan langsung di bawah input** (dengan teks merah).
 - Menampilkan **alert “Registrasi berhasil!”** hanya jika semua input telah terisi.
-

Hasil Pengujian

Input Nama	Input Email	Output	
Kosong	Kosong	Dua pesan error tampil, form tidak terkirim	
Diisi	Kosong	Pesan error hanya untuk email	
Kosong	Diisi	Pesan error hanya untuk nama	
Diisi	Diisi	Alert muncul, form terkirim	

6. Bekerja dengan Array (Aplikasi To-Do List)

a. Penjelasan

Array adalah variabel yang bisa menampung banyak nilai. Ini sangat berguna untuk mengelola daftar data. Dalam latihan ini kita akan membuat aplikasi “To-Do List” di mana pengguna bisa menambahkan item baru ke dalam daftar.

b. Contoh Kode

File: todo_list.html

```

1. <!DOCTYPE html>
2. <html>
3. <head><title>Aplikasi To-Do List</title></head>
4. <body>
5. <h2>To-Do List Sederhana</h2>
6. <input type="text" id="input-todo" placeholder="Ketik tugas baru...">
7. <button onclick="tambahTugas()">Tambah</button>
8. <ul id="daftar-tugas"></ul>
9.
10. <script>
11. var tugas = [];
12. function tambahTugas() {
13.     var inputElement = document.getElementById("input-todo");
14.     tugas.push(inputElement.value);
15.     inputElement.value = "";
16.     tampilkanDaftar();
17. }
18. function tampilkanDaftar() {
19.     var daftarElement = document.getElementById("daftar-tugas");
20.     daftarElement.innerHTML = "";
21.     for (let i = 0; i < tugas.length; i++) {
22.         daftarElement.innerHTML += "<li>" + tugas[i] + "</li>";
23.     }
24. }
25. </script>
26. </body>
27. </html>

```

C. Latihan Pratikum

- Pratikkan kode di atas. Coba tambahkan 3-4 tugas dan berikan *capture* hasilnya.

To-Do List Sederhana

- Mandi
- Makan
- Berangkat Kerja
- Kerja
- Makan

Berikut:

- Tantangan: Tambahkan tombol “Hapus” di sebelah setiap item daftar yang bisa menghapus item tersebut dari arry dan tampilan. (Petunjuk:gunakan array.splice(index,1))

To-Do List Sederhana

Ketik tugas baru... Tambah

- Bekerja Hapus
- Berolahraga Hapus

Berikut screenshot:

Dengan code:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Aplikasi To-Do List</title>
<style>
li {
    margin-bottom: 5px;
}
button.hapus-btn {
    margin-left: 10px;
    color: white;
    background-color: red;
    border: none;
    padding: 3px 8px;
    cursor: pointer;
}
</style>
</head>
<body>

<h2>To-Do List Sederhana</h2>
<input type="text" id="input-todo" placeholder="Ketik tugas baru...">
<button onclick="tambahTugas()">Tambah</button>

<ul id="daftar-tugas"></ul>
```

```

<script>

var tugas = [];

function tambahTugas() {
    var inputElement = document.getElementById("input-todo");
    var isiTugas = inputElement.value.trim();

    if (isiTugas !== "") {
        tugas.push(isiTugas);
        inputElement.value = "";
        tampilkanDaftar();
    } else {
        alert("Tugas tidak boleh kosong!");
    }
}

function hapusTugas(index) {
    tugas.splice(index, 1); // Menghapus item dari array
    tampilkanDaftar(); // Menampilkan ulang daftar yang sudah
diperbarui
}

function tampilkanDaftar() {
    var daftarElement = document.getElementById("daftar-tugas");
    daftarElement.innerHTML = "";

    for (let i = 0; i < tugas.length; i++) {
        daftarElement.innerHTML +=
        "<li>" + tugas[i] +
        " <button class='hapus-btn' onclick='hapusTugas(" + i +
        ")'>Hapus</button></li>";
    }
}

```

```
        }
    </script>

</body>
</html>
```

Analisa:

Analisis Hasil Praktikum – Versi dengan Tombol Hapus

Tujuan

Menambahkan fitur interaktif untuk menghapus item tertentu dari daftar tugas yang ditampilkan, serta dari array tugas.

Cara Kerja

- Saat tombol **Hapus** diklik, fungsi hapusTugas(index) dijalankan.
- Fungsi tersebut akan menghapus elemen array pada posisi index menggunakan array.splice(index, 1).
- Daftar diperbarui secara dinamis menggunakan tampilkanDaftar().

Hasil Pengujian

Aksi	Hasil
Menambah tugas	Tugas ditampilkan di daftar 
Klik tombol Hapus	Tugas dihapus dari array & tampilan 
Klik "Tambah" dengan kosong	Alert: "Tugas tidak boleh kosong!" 

Kesimpulan

- Praktikum ini menunjukkan bahwa JavaScript dapat memanipulasi **data dan tampilan secara dinamis** menggunakan array, perulangan, dan event handling.
- Dengan menggunakan splice() dan perulangan, aplikasi menjadi lebih **interaktif dan modular**.
- Ini menjadi fondasi penting dalam pembuatan aplikasi **To-Do List modern**, seperti yang ada pada framework seperti Vue atau React.

Kesimpulan

- Validasi menggunakan JavaScript efektif untuk **mencegah input kosong** dan memastikan kelengkapan data sebelum dikirim ke server.
- Memberi umpan balik langsung kepada pengguna meningkatkan **user experience**.
- Praktikum ini memperkuat pemahaman tentang **event handling, DOM manipulation**, dan **struktur logika validasi** di sisi klien (*client-side validation*).

7. Kalkulator Sederhana

a. Penjelasan

Ini adalah contoh yang menggabungkan sebuah konsep: Manipulasi DOM, event handling pada banyak tombol, dan logika untuk melakukan perhitungan

b. Contoh Kode

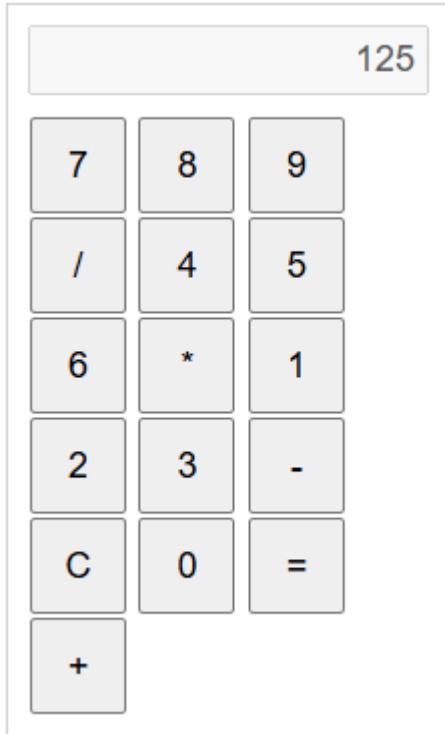
File: Kalkulaotr.html

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <title>Kalkulator</title>
5.   <style>
6.     .calculator { width: 200px; border: 1px solid #ccc; padding: 10px; }
7.     #display { width: 100%; margin-bottom: 10px; padding: 5px; box-sizing: border-box; }
8.     .buttons button { width: 48px; height: 48px; margin: 1px; }
9.   </style>
10. </head>
11. <body>
12.   <div class="calculator">
13.     <input type="text" id="display" disabled>
14.     <div class="buttons">
15.       <button onclick="appendValue('7')">7</button><button onclick="appendValue('9')">9</button><button onclick="appendValue('/')"/></button>
16.       <button onclick="appendValue('4')">4</button><button onclick="appendValue('6')">6</button><button onclick="appendValue('*')"/>*</button>
17.       <button onclick="appendValue('1')">1</button><button onclick="appendValue('3')">3</button><button onclick="appendValue('-')"/></button>
18.       <button onclick="appendValue('0')">0</button><button onclick="appendValue('+')"/>+</button>
19.     </div>
20.   </div>
21.
22.   <script>
23.     var display = document.getElementById('display');
24.     function appendValue(value) { display.value += value; }
25.     function clearDisplay() { display.value = ""; }
26.     function calculate() {
27.       try {
28.         display.value = eval(display.value);
29.       } catch (error) {
30.         display.value = 'Error';
31.       }
32.     }
33.   </script>
34. </body>
35. </html>

```

- Pratikan kode di atas: $5*25 = 125$



-  **Tujuan Praktikum**
- Membangun aplikasi kalkulator sederhana berbasis web untuk memahami bagaimana input angka dan operator matematika dapat diproses menggunakan **JavaScript** dan fungsi **eval()**.
- _____
-  **Cara Kerja Kalkulator**
- Pengguna mengetik angka dan operator menggunakan tombol.
- Setiap tombol menambahkan karakter ke dalam elemen input dengan `appendValue()`.
- Saat tombol = ditekan, fungsi `calculate()` dijalankan.
- Di dalam `calculate()`, ekspresi matematika dalam bentuk string (misalnya "2+3*4") dievaluasi menggunakan fungsi `eval()` dan hasilnya ditampilkan ke layar.
- Jika terjadi kesalahan (input tidak valid), akan ditampilkan pesan Error menggunakan try-catch.
- _____
-  **Penjelasan Fungsi eval()**
- Fungsi `eval()` dalam JavaScript digunakan untuk **mengeksekusi string sebagai kode JavaScript**. Dalam konteks kalkulator ini:

- javascript
 - CopyEdit
 - `eval("2 + 3 * 4") // → 14`
 - Fungsi eval():
 - Membaca string "2 + 3 * 4".
 - Menjalankan ekspresi sebagai **perintah JavaScript**.
 - Mengembalikan hasil evaluasinya ke variabel `display.value`.
 - Keunggulan:
 - Mudah digunakan untuk kalkulasi ekspresi matematika secara dinamis.
 - Kekurangan:
 - Kurang aman jika input berasal dari pengguna secara langsung, karena eval() dapat mengeksekusi kode JavaScript berbahaya (misalnya: `eval("alert('hacked')")`).
 - Tidak dianjurkan untuk digunakan dalam aplikasi produksi tanpa kontrol validasi ketat.
-

-  **Hasil Praktikum**

- Kalkulator berfungsi dengan baik dalam menangani input valid dan memberikan hasil yang akurat.
 - Saat pengguna melakukan kesalahan input (misalnya `5++3`), fungsi try-catch menangkap error dan menampilkan "Error" di layar.
 - Fungsi eval() memudahkan evaluasi ekspresi matematika sederhana secara langsung.
-

-  **Kesimpulan**

- Praktikum ini menunjukkan bagaimana eval() dapat dimanfaatkan untuk menyelesaikan kalkulasi secara instan dari string ekspresi matematika. Namun, **keamanan menjadi pertimbangan penting** saat menggunakan eval(), terutama jika aplikasi dikembangkan lebih lanjut untuk pengguna umum. Disarankan menggunakan parser matematika khusus seperti **math.js** untuk implementasi yang lebih aman dan fleksibel.

8. Bekerja dengan Objek String

a. Penjelasan

Onjek string menyediakan banyak metode (method) untuk memanipulasi teks. Contohnya .toUpperCase() untuk mengubah menjadi huruf kapital, .length untuk menghitung jumlah karakter, dan .substring() untuk mengambil sebagian teks.

b. Contoh Kode

File: objek_string.html

```
1. <!DOCTYPE html>
2. <html>
3. <head><title>Latihan Objek String</title></head>
4. <body>
5.   <h2>Analisis Nama</h2>
6.   <input type="text" id="inputNama" placeholder="Masukkan nama Anda">
7.   <button onclick="analisisNama()">Analisis</button>
8.   <p id="hasilAnalisis"></p>
9.
10.  <script>
11.    function analisisNama() {
12.      var nama = document.getElementById("inputNama").value;
13.      var hasil = "";
14.
15.      hasil += "Nama dalam huruf kapital: <b>" + nama.toUpperCase() + "</b><br>";
16.      hasil += "Jumlah karakter: <b>" + nama.length + "</b> karakter<br>";
17.      hasil += "3 karakter pertama: <b>" + nama.substring(0, 3) + "</b>"; 
18.
19.      document.getElementById("hasilAnalisis").innerHTML = hasil;
20.    }
21.  </script>
22. </body>
23. </html>
```

C. Latihan Pratikum

- Pratikkan kode di atas, masukan nama lengkap Anda dan klik tombol “Analisis”
- Berikan *capture* hasilnya.

Analisis Nama

Martonius Analisis

Nama dalam huruf kapital: MARTONIUS
Jumlah karakter: 9 karakter
3 karakter pertama: Mar

- Analisis dari hasil pratikkum Anda di laporan.

Praktikum berhasil menunjukkan bahwa metode toUpperCase(), length, dan substring() pada objek String dapat digunakan untuk menganalisis data teks secara efisien. Input "Martonius" dapat diubah ke huruf kapital, dihitung panjangnya, dan diekstrak 3 huruf awalnya. Ini membuktikan bahwa manipulasi string di JavaScript sangat berguna untuk validasi dan analisis data pengguna.

9. Menggunakan Objek Math

a. Penjelasan

Objek Math berisi properti dan metode untuk konstanta dan fungsi matematika. Math.random() menghasilkan angka acak antara 0 dan 1, sementara Math.floor() membulatkan angka ke bawah ke bilangan bulat terdekat. Kombinasi keduanya sangat berguna untuk menghasilkan angka acak dalam rentang tertentu.

b. Contoh Kode

File: objek_math.html

```
1. <!DOCTYPE html>
2. <html>
3. <head><title>Latihan Objek Math</title></head>
4. <body>
5.   <h2>Generator Angka Acak</h2>
6.   <p>Klik tombol untuk mendapatkan angka acak antara 1 dan 100.</p>
7.   <button onclick="generateRandom()">Hasilkan Angka</button>
8.   <h3>Angka Anda: <span id="angkaAcak">?</span></h3>
9.
10.  <script>
11.    function generateRandom() {
12.      // Menghasilkan angka acak dari 0 hingga 99, lalu tambah 1
13.      var angka = Math.floor(Math.random() * 100) + 1;
14.      document.getElementById("angkaAcak").innerText = angka;
15.    }
16.  </script>
17. </body>
18. </html>
```

c. Latihan Pratikum

- Pratikkan kode di atas. Klik tombol beberapa kali untuk melihat angka yang berbeda
- Berikan *capture* yang menunjukkan salah satu angka acak yang dihasilkan.
- Analisis dari hasil pratikkum Anda di laporan

Generator Angka Acak

Klik tombol untuk mendapatkan angka acak antara 1 dan 100.

Hasilkan Angka

Angka Anda: 44

Kode ini menggunakan **JavaScript** dan **Math API** untuk menghasilkan angka acak antara 1 hingga 100.

- **Math.random()** menghasilkan angka acak antara 0 (inklusif) dan 1 (eksklusif).
- **Math.floor()** digunakan untuk mengubah angka desimal menjadi bilangan bulat.
- Dengan perkalian `Math.random() * 100`, angka acak dihasilkan antara 0 hingga 99, kemudian ditambah 1 untuk memastikan angka acak berada di rentang 1 hingga 100.

Fungsi generateRandom() dijalankan saat tombol diklik, dan angka acak yang dihasilkan ditampilkan dalam elemen dengan ID `angkaAcak`.

10. Menampilkan Tanggal dan Jam Dinamis

a. Penjelasan

Objek Date digunakan untuk bekerja dengan tanggal dan waktu. Dengan membuat instance baru (`new Date()`), kita bisa mendapatkan waktu saat ini dan menampilkannya. Untuk membuat jam yang berjalan, kita perlu memanggil fungsi pembaruan waktu secara berulang menggunakan `setInterval()`

b. Contoh Kode

File: objek_date.html

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <title>Jam Digital</title>
5.   <style>
6.     #jam { font-size: 2em; font-family: 'Courier New', Courier, monospace; }
7.   </style>
8. </head>
9. <body onload="jalankanJam()">
10.  <h2>Jam Digital Real-Time</h2>
11.  <div id="jam"></div>
12.
13.  <script>
14.    function updateJam() {
15.      var sekarang = new Date();
16.      var jam = sekarang.getHours();
17.      var menit = sekarang.getMinutes();
18.      var detik = sekarang.getSeconds();
19.
20.      // Tambahkan nol di depan jika angka < 10
21.      jam = (jam < 10 ? "0" : "") + jam;
22.      menit = (menit < 10 ? "0" : "") + menit;
23.      detik = (detik < 10 ? "0" : "") + detik;
24.
25.      var waktuString = jam + ":" + menit + ":" + detik;
26.      document.getElementById("jam").innerText = waktuString;
27.    }
28.
29.    function jalankanJam() {
30.      // Panggil updateJam setiap 1000 milidetik (1 detik)
31.      setInterval(updateJam, 1000);
32.    }
33.  </script>
34. </body>
35. </html>

```

C. latihan pratikum

- Pratikkan kode di atas dan amati jam yang berjalan.
- Berikan *capture* halaman tersebut

Jam Digital Real-Time

07 : 50 : 25

Berikut:

- Analisis dari hasil praktikum Anda di laporan, jelaskan fungsi dari setInterval()
- Analisis Kode – Jam Digital Real-Time**
- Kode ini membuat aplikasi jam digital yang memperbarui waktu setiap detik.

- Fungsi **updateJam()** mengambil waktu saat ini menggunakan objek **Date()**, kemudian memformat jam, menit, dan detik untuk selalu menampilkan dua digit (misalnya, 09:03:05).
- Fungsi **setInterval()** digunakan untuk memanggil **updateJam()** setiap 1000 milidetik (1 detik), sehingga tampilan waktu selalu diperbarui secara real-time. Praktikum ini menunjukkan cara sederhana untuk menggunakan **JavaScript** dalam membuat aplikasi berbasis waktu.

11. Membuat Image Slideshow Sederhana

a. Penjelasan

Manipulasi gambar adalah salah satu aplikasi DOM yang paling visual. Dengan mengubah properti .src dari sebuah elemen **Kesalahan! Nama file tidak ditentukan.**, kita dapat membuat slideshow. Kita akan menyimpan URL gambar dalam sebuah array dan menggunakan sebuah counter untuk beralih antar gambar.

b. Contoh Kode

File: Slideshow.html

```

1. <!DOCTYPE html>
2. <html>
3. <head><title>Image Slideshow</title></head>
4. <body>
5.   <h2>Galeri Gambar</h2>
6.   <img id="slideshow-image" src="" alt="Slideshow" width="500">
7.   <br>
8.   <button onclick="gantiGambar(-1)">Previous</button>
9.   <button onclick="gantiGambar(1)">Next</button>
10.
11.  <script>
12.    var gambar = [
13.      "https://placehold.co/500x300/3498db/fffff?text=Gambar+1",
14.      "https://placehold.co/500x300/e74c3c/fffff?text=Gambar+2",
15.      "https://placehold.co/500x300/2ecc71/fffff?text=Gambar+3"

```

```

16.    };
17.    var indexGambar = 0;
18.    var imageElement = document.getElementById("slideshow-image");
19.
20.    function tampilkanGambar() {
21.        imageElement.src = gambar[indexGambar];
22.    }
23.
24.    function gantiGambar(arah) {
25.        indexGambar += arah;
26.        // Jika sudah gambar terakhir, kembali ke awal
27.        if (indexGambar >= gambar.length) {
28.            indexGambar = 0;
29.        }
30.        // Jika sudah gambar pertama dan menekan previous, pergi ke akhir
31.        if (indexGambar < 0) {
32.            indexGambar = gambar.length - 1;
33.        }
34.        tampilkanGambar();
35.    }
36.
37.    // Tampilkan gambar pertama saat halaman dimuat
38.    tampilkanGambar();
39. </script>
40. </body>
41. </html>

```

C. Latihan Pratikum

- Pratikkan kode di atas. Coba Klik tombol “Next” dan “Previous”.

Galeri Gambar



[Previous](#) [Next](#)

- Tambahkan dua URL gambar baru ke dalam *array* gambar.

```

var gambar = [
    "https://placehold.co/500x300/3498db/fffff?text=Gambar+1",
    "https://placehold.co/500x300/e74c3c/fffff?text=Gambar+2",
    "https://placehold.co/500x300/2ecc71/fffff?text=Gambar+3",
    "1.jpg",
    "2.jpg",
    "3.png"
];

```

- Berikan *capture* salah satu gambar yang tampil dalam slideshow

Galeri Gambar



[Previous](#)

[Next](#)

- Analisis dari hasil praktikum Anda di laporan

Kode ini berfungsi untuk menampilkan **image slideshow** yang memungkinkan pengguna untuk mengganti gambar menggunakan tombol "Next" dan "Previous".

Fungsi utama:

1. **Gambar lokal** digunakan dengan mengubah URL gambar dalam array gambar ke jalur file lokal (misalnya, images/gambar1.jpg).
2. **gantiGambar(arah)** mengubah gambar berdasarkan arah yang diberikan (tombol "Next" atau "Previous").
3. **tampilkanGambar()** memperbarui elemen dengan gambar yang sesuai berdasarkan **index** dalam array.

Dengan **jalur file lokal**, gambar ditampilkan langsung dari folder lokal, memungkinkan pengelolaan lebih mudah pada aplikasi berbasis file lokal tanpa bergantung pada URL eksternal.

D. Tugas Akhir Praktikum: Mini E-Commerce Product Page

Buatlah sebuah halaman web interaktif untuk menampilkan dan memesan produk yang mengimplementasikan seluruh materi praktikum.

Struktur Halaman & Fitur

1. Header & Waktu Transaksi:

- Judul Halaman: "Toko Elektronik Cepat".
- Tampilkan tanggal dan jam saat ini secara dinamis menggunakan objek Date dan setInterval.

2. Display Produk:

- Gunakan **array** untuk menyimpan data produk (minimal 3 produk). Setiap produk harus memiliki: nama, harga, dan array berisi beberapa url_gambar.
- Tampilkan produk pertama dari array saat halaman dimuat.
- Gunakan **Image Slideshow** (Praktikum 11) untuk menampilkan gambar-gambar produk yang sedang aktif. Pengguna bisa klik "Next" atau "Previous" untuk melihat gambar lain dari produk tersebut.
- Di bawah slideshow, tampilkan nama dan harga produk yang sedang aktif.

3. Daftar Produk:

- Di samping display utama, gunakan **perulangan** (Praktikum 4) untuk menampilkan seluruh produk yang ada di **array** sebagai daftar yang bisa diklik.
- Ketika salah satu item di daftar ini diklik, display utama (slideshow, nama, harga) harus berganti sesuai produk yang dipilih.

4. Form Pemesanan:

- Input untuk **Nama Pemesan**.
- Input untuk **Jumlah Pesan** (tipe number).
- Input untuk **Kode Promo**.
- Tombol "Pesanan Sekarang".

5. Logika Pemesanan (Event onclick pada tombol):

- Validasi **Form** (Praktikum 5): Nama pemesan dan jumlah tidak boleh kosong. Jumlah harus lebih dari 0.
- Perhitungan **Total**:
 - Subtotal = harga_produk * jumlah_pesanan.
 - **Logika Percabangan** (Praktikum 3): Jika kode promo yang dimasukkan (setelah diubah ke huruf besar menggunakan metode objek String)

adalah "DISKON10", berikan potongan 10%.

- Hitung Total Akhir.
- Generate **Order ID** (Praktikum 9): Buat ID Pesanan unik, contoh: INV- diikuti 5 digit angka acak (gunakan Math.random()).
- **Tampilkan Struk/Ringkasan Pesanan** di area output, berisi: Order ID, Nama Pemesan, Nama Produk, Jumlah, Subtotal, Potongan, dan Total Akhir.

Desain dan Styling

- Gunakan CSS atau kelas-kelas dari **Bootstrap 5** (yang telah dipelajari di Modul 3) untuk membuat layout yang rapi dan responsif.
- Gunakan layout grid untuk memisahkan display produk utama dengan daftar produk.
- Buat tampilan form dan area output yang mudah dibaca.

Rubrik Penilaian

Aspek Penilaian	Keterangan	Bobot (%)
Implementasi Lengkap	Semua fitur dari 1-5 berhasil diimplementasikan.	30%
Manipulasi DOM & Event	Interaksi klik pada daftar produk, tombol slideshow, dan form berjalan lancar.	20%
Logika & Array	Logika percabangan diskon, validasi form, dan pengelolaan data produk dalam array berfungsi dengan benar.	25%
Penggunaan Objek Bawaan	Objek Date, Math, dan String digunakan sesuai instruksi (jam dinamis, Order ID, kode promo).	15%
Kerapihan & Tampilan	Kode terstruktur, mudah dibaca, dan tampilan halaman rapi serta responsif.	10%

Toko Elektronik Cepat

Waktu Transaksi: 7/10/2025, 8:38:17 AM

Headphone Sony 1

Headphone Sony

Harga: Rp 1,500,000

[Previous](#) [Next](#)

Daftar Produk

Laptop ASUS

Smartphone Samsung

Headphone Sony

Form Pemesanan

Nama Pemesan

Marton

Jumlah Pesan

21

Kode Promo

DISKON10

[Pesanan Sekarang](#)

Ringkasan Pesanan

Order ID: INV-321

Nama Pemesan: Marton

Produk: Headphone Sony

Jumlah: 21

Subtotal: Rp 31,500,000

Potongan: Rp 3,150,000

Total Akhir: Rp 28,350,000

© 2025 Toko Elektronik Cepat | Semua Hak Cipta Dilindungi

Vidio penjelasan web mini e-commerce : [link](#)

Dengan Code:

```
<!DOCTYPE html>
<html>
<head>
<title>Latihan Logika Percabangan</title>
</head>
<body>
<h2>Kalkulator Diskon Belanja</h2>
<p>Masukkan total belanja Anda:</p>
<input type="number" id="totalBelanja" placeholder="Contoh: 150000">
<button onclick="cekDiskon()">Cek Diskon</button>
<h3 id="hasil"></h3>
<script>
function cekDiskon() {
    var total = parseFloat(document.getElementById("totalBelanja").value);
    var pesan = "";
    if (isNaN(total)) {
        pesan = "Mohon masukkan angka yang valid.";
    } else if (total >= 100000) {
        var diskon = total * 0.15; // Ubah dari 0.10 menjadi 0.15 (diskon 15%)
        var totalBayar = total - diskon;
        pesan = "Selamat! Anda dapat diskon 15% sebesar Rp " +
        diskon.toLocaleString() +
        ". Total bayar: <b>Rp " + totalBayar.toLocaleString() + "</b>";
    } else {
        pesan = "Maaf, tidak ada diskon. Total bayar: <b>Rp " +
        total.toLocaleString() + "</b>";
    }
    document.getElementById("hasil").innerHTML = pesan;
}
</script>
</body>
</html>
```

Modul 5. Dasar-dasar PHP

A. Tujuan

- Setelah mengikuti praktikum, Mahasiswa diharapkan mampu:
1. Mampu memahami konsep dasar dan sintaks bahasa pemrograman PHP.
 2. Dapat membuat program dengan PHP untuk menghasilkan halaman web yang dinamis
 3. Mampu mengolah data dari input form dan variabel superglobals.

B. Teori Dasar PHP

File PHP adalah dokumen teks dengan ekstensi .php yang dapat berisi campuran kode HTML, CSS, JavaScript, dan skrip PHP. Skrip PHP dieksekusi di server, dan hasilnya dikirim ke browser sebagai HTML biasa.

PHP dapat digunakan untuk membuat halaman web yang dinamis, mengelola file di server, mengumpulkan data dari form, mengatur cookies dan sesi, serta berinteraksi dengan berbagai jenis database.

PHP populer karena bersifat open-source (gratis), berjalan di berbagai platform (Windows, Linux, macOS), kompatibel dengan hampir semua server web, dan didukung oleh komunitas yang besar serta dokumentasi yang melimpah.

Untuk pengembangan PHP di komputer lokal, Anda memerlukan lingkungan server yang terdiri dari Web Server (seperti Apache), PHP, dan Database (seperti MySQL/MariaDB).

Cara termudah adalah dengan menginstal paket perangkat lunak terintegrasi seperti XAMPP (disarankan), Laragon atau WAMP. Paket ini sudah mencakup semua komponen yang dibutuhkan dalam satu kali instalasi.

C. Praktikum

1. Basic syntax

a. Penjelasan

Skrip PHP ditulis di antara <?php dan ?>. Perintah echo digunakan untuk menampilkan output. Setiap pernyataan diakhiri dengan titik koma (;).

b. Contoh kode File: syntax.php

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.   <h1>Halaman PHP Pertamaku</h1>
5.   <?php
6.     echo "Halo, Dunia!";
7.   ?>
8. </body>
9. </html>
```

c. latihan praktikum

- Pratikkan contoh kode di atas.

Berikut hasilnya

Halaman PHP Pertamaku

Halo, Dunia!

- Buat file baru, lalu gunakan echo untuk menampilkan Nama dan NIM Anda pada baris yang berbeda
-

Nama: Martonius Epta Sidabutar
NIM: 202312051

Dengan code .php

```
<!DOCTYPE html>
<html>
<body>
<?php
// Menampilkan Nama
echo "Nama: Martonius Epta Sidabutar<br>";

// Menampilkan NIM
echo "NIM: 202312051";
?>
</body>
</html>
```

Analisis:

Perbandingan dan Kesimpulan:

• **Integrasi PHP dengan HTML:**

- Kedua kode ini menggunakan integrasi PHP di dalam HTML. Perbedaannya terletak pada tujuan dan isi yang ditampilkan.
- Kode pertama menampilkan pesan statis sederhana "Halo, Dunia!" di dalam HTML, sementara kode kedua lebih berfokus pada menampilkan dua informasi terpisah (Nama dan NIM) dengan penataan sedikit lebih kompleks (menggunakan
 untuk pemisahan baris).

• **Fungsionalitas echo:**

- Pada kedua kode, perintah echo digunakan untuk menampilkan teks. echo adalah metode yang paling umum digunakan di PHP untuk menampilkan informasi atau output.
- Di kode pertama, hanya satu output yang dihasilkan, sedangkan di kode kedua, ada dua output terpisah, dan satu di antaranya diberi pemisahan baris dengan tag
.

• **Output:**

- Output dari kedua kode sangat berbeda dalam konteks tampilan. Kode pertama hanya menampilkan satu kalimat di halaman web, sedangkan kode kedua memisahkan dua informasi berbeda (Nama dan NIM) dengan cara yang lebih terstruktur menggunakan baris baru.

• **Praktik dan Penggunaan:**

- Kode pertama cocok untuk aplikasi sederhana atau halaman yang hanya perlu menampilkan teks langsung dari PHP.
- Kode kedua lebih cocok untuk aplikasi yang memerlukan tampilan terstruktur, seperti menampilkan data pengguna atau informasi lainnya yang dibutuhkan dalam baris yang berbeda.

Secara keseluruhan, kedua kode ini menunjukkan penggunaan dasar PHP untuk menampilkan teks, namun dengan pendekatan yang berbeda untuk tujuan masing-masing.

2. PHP Coments

a. Penjelasan

Komentar adalah catatan dalam kode yang tidak dieksekusi oleh PHP

b. Contoh kode File: comments.php

```
1. <?php
2. // Ini adalah komentar satu baris
3. # Ini juga komentar satu baris
4. /*
5. Ini adalah blok komentar
6. untuk beberapa baris.
7. */
8. echo "Kode ini akan dieksekusi.";
9. ?>
```

c. Latihan Pratikum

- Pratikkan contoh kode:

Kode ini akan dieksekusi.

- Tambahkan ketiga jenis komentar pada skrip dari latihan sebelumnya untuk menjelaskan fungsinya.

Halo, Dunia!
Nama: Martonius
NIM: 202312051

Dengan code

```
<?php
    // Ini adalah komentar satu baris: Komentar ini hanya mencakup
    // satu baris

    // Digunakan untuk memberikan penjelasan singkat atau catatan
    // pada kode

    echo "Halo, Dunia!"; // Menampilkan teks "Halo, Dunia!" pada
    // halaman

    # Ini juga komentar satu baris: Komentar ini juga digunakan untuk
    // penjelasan singkat

    # Biasanya digunakan di baris yang lebih sederhana

    echo "Nama: Martonius<br>"; // Menampilkan "Nama: James" di
    // halaman, kemudian pindah ke baris baru

    /*
```

```

Ini adalah blok komentar
untuk beberapa baris.

Komentar ini dapat digunakan untuk menjelaskan kode yang lebih
panjang atau memberikan
penjelasan detail tentang fungsi atau blok kode tertentu.

*/
echo "NIM: 202312051"; // Menampilkan "NIM: 1234567890" di
halaman
?>

```

- Analisis:

Analisis Kode PHP dengan Komentar:

1. Komentar dalam PHP:

- Komentar sangat penting dalam kode untuk memberikan penjelasan atau deskripsi mengenai fungsi atau tujuan bagian kode tertentu.
- PHP mendukung tiga jenis komentar: komentar satu baris (// atau #), dan komentar blok yang mencakup beberapa baris /* ... */.

2. Jenis Komentar:

- **Komentar Satu Baris (// dan #):**

- Digunakan untuk memberikan penjelasan singkat tentang kode di satu baris.
- Misalnya, // Menampilkan teks "Halo, Dunia!" memberikan penjelasan tentang baris yang berisi perintah echo.
- # adalah bentuk lain dari komentar satu baris yang memiliki fungsi yang sama dengan //.

- **Komentar Blok /* ... */:**

- Digunakan untuk memberikan penjelasan lebih panjang yang mencakup beberapa baris kode.
- Dalam kode ini, komentar blok digunakan untuk menjelaskan fungsi dari seluruh bagian kode atau untuk memberikan dokumentasi tambahan.
- Menjadi lebih berguna ketika kode yang perlu dijelaskan lebih kompleks atau ketika ada beberapa baris yang perlu didokumentasikan bersama-sama.

3. Manfaat Komentar:

- **Pemeliharaan Kode:** Membantu developer lain atau diri kita sendiri untuk memahami maksud dan fungsi kode, terutama ketika proyek menjadi lebih besar dan kompleks.
- **Debugging:** Komentar juga digunakan untuk menonaktifkan sementara bagian dari kode tanpa menghapusnya.
- **Dokumentasi:** Memberikan konteks lebih jelas tentang apa yang dilakukan kode dan mengapa.

4. Penggunaan Komentar pada Kode:

- Kode yang diberikan menggunakan semua jenis komentar untuk menjelaskan bagian-bagian kode dengan cara yang berbeda. Ini memberikan dokumentasi yang memadai, memudahkan pemahaman, dan membantu dalam kolaborasi tim.

Secara keseluruhan, penggunaan komentar yang baik sangat penting dalam pengembangan perangkat lunak untuk menjaga kode tetap terorganisir, mudah dipahami, dan mudah dipelihara.

3. PHP Variables

a. Penjelasan

Variabel digunakan untuk menyimpan data dan diawali dengan tanda dolar (\$)

b. Contoh Kode File:variables.php

```

1. <?php
2. $nama_mahasiswa = "Abadi Nugroho";
3. $semester = 3;
4. echo "Selamat datang, " . $nama_mahasiswa . "! Anda sekarang berada di
semester " . $semester . ".";
5. ?>
```

- Jika kode di atas di praktikan

Selamat datang, Abadi Nugroho! Anda sekarang berada di semester 3.

- Membuat dua variabel untuk menyimpan panjang dan lebar, lalu hitung dan tampilkan luasnya

Dengan skrip

```

<?php
// Mendeklarasikan variabel panjang dan lebar
$panjang = 5; // panjang dalam satuan meter
$lebar = 3; // lebar dalam satuan meter
```

```

// Menghitung luas
$luas = $panjang * $lebar;

// Menampilkan hasil luas
echo "Luas persegi panjang dengan panjang $panjang meter dan lebar
$lebar meter adalah $luas meter persegi.";

?>

```

Dan jika di buka di browser:

Luas persegi panjang dengan panjang 5 meter dan lebar 3 meter adalah 15 meter persegi.

- Analisis:

Analisis Kode PHP untuk Menghitung Luas Persegi Panjang

1. Deklarasi Variabel:

- **\$panjang = 5;** dan **\$lebar = 3;**: Kode ini mendeklarasikan dua variabel yang menyimpan nilai panjang dan lebar dari persegi panjang. Nilai-nilai ini bisa disesuaikan sesuai kebutuhan.
- Variabel ini menggunakan tipe data numerik (integer), yang cocok untuk perhitungan matematis.

2. Perhitungan Luas:

- **\$luas = \$panjang * \$lebar;**: Dengan rumus dasar geometri untuk menghitung luas persegi panjang ($\text{Luas} = \text{Panjang} \times \text{Lebar}$), kode ini melakukan perhitungan dan menyimpan hasilnya dalam variabel **\$luas**.

3. Menampilkan Hasil:

- **echo** digunakan untuk menampilkan hasil perhitungan kepada pengguna. Pesan yang ditampilkan menggabungkan nilai dari variabel menggunakan string concatenation (.) .
- Output dihasilkan dalam format yang mudah dimengerti, dengan menyebutkan nilai panjang, lebar, dan hasil luas.

4. Keuntungan Penggunaan Variabel:

- **Pemeliharaan:** Dengan menyimpan panjang dan lebar dalam variabel, kode menjadi lebih fleksibel dan mudah dipelihara. Anda hanya perlu mengubah nilai variabel untuk menghitung luas persegi panjang yang berbeda.

- **Pembacaan Kode yang Lebih Jelas:** Menggunakan variabel dengan nama yang deskriptif (\$panjang, \$lebar, \$luas) meningkatkan keterbacaan kode dan memudahkan orang lain memahami apa yang dihitung.

Kesimpulan:

Kode ini memberikan contoh dasar bagaimana PHP dapat digunakan untuk melakukan perhitungan matematika sederhana dan menampilkan hasilnya dalam format yang ramah pengguna. Ini juga menunjukkan penggunaan variabel dalam PHP untuk menyimpan dan memanipulasi data.

4. PHP Data Types

a. Penjelasan

PHP Mendukung tipe data seperti String, Integer, Float, Boolean, dan Array

b. Contoh Kode File: datatypes.php

```

1. <?php
2. $judul_buku = "Dasar-Dasar Web"; // String
3. $harga = 95000; // Integer
4. $rating = 4.5; // Float
5. $tersedia = true; // Boolean
6. var_dump($rating); // Menampilkan informasi detail tentang variabel
7. ?>

```

c. Latihan Pratikum

- Contoh kode di atas jika diperlakukan akan menghasilkan **float(4.5)**
- Buat variabel untuk setiap tipe data dan gunakan var_dump() untuk memeriksa tipe dan nilainya.

string(8) "John Doe" int(25) float(175.5) bool(true) array(3) { [0]=> string(5) "Merah" [1]=> string(4) "Biru" [2]=> string(5) "Hijau" } object(Mobil)#1(2) { ["merk"]=> string(6) "Toyota" ["model"]=> string(7) "Corolla" } NULL

Dengan code:

```

<?php
// String
$nama = "John Doe";
var_dump($nama); // Menampilkan tipe dan nilai variabel $nama

// Integer
$umur = 25;
var_dump($umur); // Menampilkan tipe dan nilai variabel $umur

```

```

// Float
$tinggi = 175.5;
var_dump($tinggi); // Menampilkan tipe dan nilai variabel $tinggi

// Boolean
$aktif = true;
var_dump($aktif); // Menampilkan tipe dan nilai variabel $aktif

// Array
$warna_favorit = array("Merah", "Biru", "Hijau");
var_dump($warna_favorit); // Menampilkan tipe dan nilai variabel
$warna_favorit

// Object (contoh menggunakan kelas sederhana)
class Mobil {
    public $merek = "Toyota";
    public $model = "Corolla";
}
$mobil = new Mobil();
var_dump($mobil); // Menampilkan tipe dan nilai objek $mobil

// Null
$nilai = null;
var_dump($nilai); // Menampilkan tipe dan nilai variabel $nilai
?>
```

- Analisis:

Analisis Pratikum: Pemeriksaan Tipe Data dengan var_dump() di PHP

Pratikum ini bertujuan untuk memperkenalkan berbagai tipe data dasar dalam PHP dan cara memeriksa tipe data serta nilai dari variabel menggunakan fungsi var_dump(). Fungsi ini sangat berguna untuk debugging dan memahami struktur data dalam sebuah program. Berikut adalah analisis dari pratikum di atas:

1. Pengenalan Tipe Data dalam PHP

PHP mendukung beberapa tipe data dasar yang digunakan untuk menyimpan berbagai jenis nilai. Dalam praktikum ini, kita melihat berbagai tipe data yang dapat digunakan di PHP:

- **String:** Digunakan untuk menyimpan data teks. Dalam contoh ini, variabel \$nama berisi nama seseorang.
- **Integer:** Digunakan untuk menyimpan angka bulat. Variabel \$umur menyimpan usia dalam bentuk integer.
- **Float:** Digunakan untuk menyimpan angka desimal atau pecahan. Variabel \$tinggi menyimpan tinggi badan dengan angka desimal.
- **Boolean:** Menyimpan dua nilai, yaitu true atau false. Variabel \$aktif menunjukkan apakah seseorang aktif atau tidak.
- **Array:** Digunakan untuk menyimpan beberapa nilai dalam satu variabel. Variabel \$warna_favorit adalah array yang menyimpan beberapa warna favorit.
- **Object:** Digunakan untuk menyimpan data dalam bentuk objek, yang merupakan instance dari sebuah kelas. Variabel \$mobil adalah objek dari kelas Mobil.
- **Null:** Tipe data ini digunakan untuk menyatakan bahwa variabel tidak memiliki nilai atau tidak terdefinisi.

2. Penggunaan var_dump()

- Fungsi **var_dump()** digunakan untuk menampilkan tipe data dan nilai variabel. Ini memberikan informasi yang lebih detail daripada hanya menggunakan echo. var_dump() juga menunjukkan panjang string untuk tipe data string dan jumlah elemen dalam array.

3. Keuntungan Menggunakan var_dump()

- **Mempermudah Debugging:** Fungsi var_dump() sangat berguna ketika Anda bekerja dengan data yang lebih kompleks atau saat mencoba menemukan masalah dengan tipe data. Ini memungkinkan Anda untuk melihat tidak hanya nilai variabel tetapi juga tipe data yang sedang digunakan.
- **Memahami Struktur Data:** Dalam contoh array dan objek, var_dump() memberikan gambaran jelas tentang struktur data yang lebih kompleks, misalnya jumlah elemen dalam array atau properti dalam objek.
- **Verifikasi Data:** Dengan menggunakan var_dump(), Anda bisa memastikan bahwa tipe data yang Anda harapkan benar-benar sesuai dengan yang ada di dalam variabel, misalnya memverifikasi bahwa sebuah variabel yang seharusnya menyimpan angka tidak mengandung string.

4. Praktik dan Aplikasi dalam Pengembangan

- **Pengembangan Aplikasi Dinamis:** Dalam pengembangan aplikasi web menggunakan PHP, seringkali kita bekerja dengan data yang masuk dari formulir atau API. Dengan var_dump(), Anda bisa melihat struktur data yang diterima dan memverifikasi bahwa data tersebut sesuai dengan yang diharapkan.
- **Pengelolaan Tipe Data:** Mengelola tipe data yang tepat dalam PHP sangat penting untuk menghindari kesalahan dalam perhitungan atau manipulasi data, terutama ketika data tersebut dikirim ke database atau diproses lebih lanjut.
- **Objek dan Array:** Dengan objek dan array yang lebih kompleks, var_dump() memberi Anda gambaran yang sangat berguna untuk memeriksa nilai-nilai dalam koleksi data tersebut, memungkinkan Anda untuk memastikan bahwa data sudah terstruktur dengan benar.

5. Kesimpulan

Pratikum ini menunjukkan cara menggunakan var_dump() untuk memeriksa tipe data dan nilai variabel dalam PHP. Ini adalah keterampilan penting untuk debugging, pemahaman data, dan pemrograman yang efisien. Dengan memahami berbagai tipe data yang tersedia di PHP dan menggunakan alat seperti var_dump(), Anda dapat menulis kode yang lebih robust dan bebas dari kesalahan terkait tipe data.

5. PHP Strings

a. Penjelasan

PHP Memiliki banyak fungsi bawaan untuk memanipulasi teks (string).

b. Contoh Kode File: strings.php

```
1. <?php
2. $kalimat = "STITEK Bontang adalah kampus IT terbaik";
3. echo "Panjang kalimat: " . strlen($kalimat) . "<br>";
4. echo "Jumlah kata: " . str_word_count($kalimat) . "<br>";
5. echo "Mengganti kata: " . str_replace("terbaik", "favorit", $kalimat);
6. ?>
7.
```

c. Latihan Pratikum

- Kode di atas dipraktikkan:

Panjang kalimat: 39

Jumlah kata: 6

Mengganti kata: STITEK Bontang adalah kampus IT favorit

- Gunakan fungsi strtoupper() untuk membuat semua teks menjadi huruf kapital

Berikut codenya:

```
<?php
$kalimat = "STITEK Bontang adalah kampus IT terbaik";

echo "Panjang kalimat: " . strlen($kalimat) . "<br>";
echo "Jumlah kata: " . str_word_count($kalimat) . "<br>";

// Mengganti kata 'terbaik' menjadi 'favorit' lalu mengubah seluruh
// kalimat jadi huruf kapital
$kalimatBaru = str_replace("terbaik", "favorit", $kalimat);
echo "Mengganti kata dan kapitalisasi: " .
strtoupper($kalimatBaru);

?>
```

Dan jika dipraktikkan :

Panjang kalimat: 39

Jumlah kata: 6

Mengganti kata dan kapitalisasi: STITEK BONTANG ADALAH KAMPUS IT FAVORIT

- Analisis:

Kode PHP di atas melakukan tiga operasi dasar pada sebuah string:

1. **strlen(\$kalimat)**

Fungsi ini menghitung total jumlah karakter dalam string, termasuk spasi.

► Berguna untuk validasi panjang teks atau data input.

2. **str_word_count(\$kalimat)**

Menghitung jumlah kata dalam string berdasarkan pemisah spasi.

► Bermanfaat untuk analisis teks dasar seperti memeriksa kelengkapan input kalimat.

3. **str_replace("terbaik", "favorit", \$kalimat) + strtoupper()**

Pertama-tama mengganti kata "terbaik" dengan "favorit", lalu mengubah seluruh string menjadi huruf kapital.

► Berguna dalam normalisasi data teks, seperti membuat heading atau data konsisten untuk keperluan pencarian/penyimpanan.

Kelebihan:

- Kode ringkas dan mudah dibaca.
- Menggabungkan beberapa fungsi string dasar untuk manipulasi teks yang umum dilakukan di aplikasi web.

Kekurangan:

- Tidak memperhitungkan sensitivitas huruf besar-kecil dalam str_replace(), jadi hanya cocok jika input selalu dalam format yang sama.
- Tidak menangani karakter khusus atau multibahasa (non-UTF-8) jika dibutuhkan. Secara keseluruhan, kode ini sangat cocok untuk pembelajaran dasar manipulasi string di PHP.

6. PHP Math

a. Penjelasan

PHP dapat melakukan operasi matematika dasar dan kompleks.

b. Contoh Kode File: math.php

```
1. <?php  
2. echo "Hasil dari 15 % 4 adalah " . (15 % 4) . " (sisa bagi)<br>";  
3. echo "Angka acak antara 10 dan 20: " . rand(10, 20);  
4. ?>
```

c. Latihan Praktikum:

- Jika kode di atas di praktikan akan menghasilkan:

Hasil dari 15 % 4 adalah 3 (sisa bagi)
Angka acak antara 10 dan 20: 11

- Hitung nilai dari $(10+2)/3*5$ dan tampilkan hasilnya

Berikut hasilnya:

Hasil dari 15 % 4 adalah 3 (sisa bagi)
Angka acak antara 10 dan 20: 19
Hasil dari $(10 + 2) / 3 * 5$ adalah 20

Dengan kode:

```
<?php  
echo "Hasil dari 15 % 4 adalah " . (15 % 4) . " (sisa bagi)<br>";  
echo "Angka acak antara 10 dan 20: " . rand(10, 20) . "<br>";  
  
// Menghitung nilai dari  $(10 + 2) / 3 * 5$   
$hasil = (10 + 2) / 3 * 5;
```

```
echo "Hasil dari (10 + 2) / 3 * 5 adalah " . $hasil;
```

```
?>
```

- Analisis:

Analisis Singkat Kode PHP

Kode ini mendemonstrasikan penggunaan operator aritmatika dan fungsi built-in untuk manipulasi angka dalam PHP. Berikut rincian analisisnya:

1. `15 % 4`

- Operator modulus (%) digunakan untuk menghitung sisa bagi dari pembagian 15 dengan 4.
 - **Hasilnya adalah 3**, karena 15 dibagi 4 hasilnya 3 dengan sisa 3.
 - Cocok untuk operasi logika seperti menentukan bilangan ganjil/genap, pengulangan, atau siklus rotasi.
-

2. `rand(10, 20)`

- Fungsi `rand()` menghasilkan angka acak antara dua nilai yang ditentukan (inklusif).
 - **Contoh output: 13, 17, 20, dll.**
 - Digunakan dalam simulasi, permainan, atau keamanan dasar (misal kode OTP).
-

3. `(10 + 2) / 3 * 5`

- Evaluasi ekspresi aritmatika mengikuti urutan operator (aturan matematika):
 - **10 + 2 = 12**
 - **12 / 3 = 4**
 - **4 * 5 = 20**
 - PHP mengikuti **urutan operasi matematika (PEMDAS)**.
 - Berguna untuk menghitung rumus matematis yang lebih kompleks.
-

Kelebihan Kode:

- Menunjukkan pemahaman dasar tentang operator aritmatika dan fungsi random dalam PHP.

- Dapat dijadikan dasar untuk membuat kalkulator, simulasi, atau sistem berbasis angka acak.

Kekurangan:

- Tidak ada validasi input karena semua nilai sudah ditentukan (hardcoded).
 - `rand()` kurang aman untuk aplikasi kriptografi, lebih baik gunakan `random_int()` untuk keamanan.
-

Kesimpulan:

Kode ini cocok untuk pemula yang ingin belajar tentang operasi matematika dan penggunaan fungsi random di PHP. Ini merupakan latihan yang baik untuk memahami dasar-dasar logika pemrograman.

7. PHP Constants

a. Penjelasan

Konstanta adalah variabel yang nilainya tidak dapat diubah. Didefinisikan menggunakan `define()`.

b. Contoh Kode File: constants.php

```
1. <?php
2. define("NAMA_KAMPUS", "STITEK Bontang");
3. echo "Saya kuliah di " . NAMA_KAMPUS;
4. ?>
```

- jika diperlakukan:
`Saya kuliah di STITEK Bontang`
- pembuatan konstanta untuk menyimpan nama mata kuliah dan tampilkan.
`Saya kuliah di STITEK Bontang`

`Mata kuliah yang sedang saya pelajari adalah Pemrograman Web`

Dengan kode:

```
<?php
define("NAMA_KAMPUS", "STITEK Bontang");
define("MATA_KULIAH", "Pemrograman Web");

echo "Saya kuliah di " . NAMA_KAMPUS . "<br>";
echo "Mata kuliah yang sedang saya pelajari adalah " .
MATA_KULIAH;
?>
```

- Analisis:
Kode ini menggunakan fungsi `define()` untuk mendeklarasikan **konstanta**, yaitu nilai tetap yang tidak dapat diubah selama eksekusi program.

◆ Elemen Kode:

1. **define("NAMA_KAMPUS", "STITEK Bontang")**
→ Menyimpan nama kampus sebagai konstanta.
2. **define("MATA_KULIAH", "Pemrograman Web")**
→ Menyimpan nama mata kuliah sebagai konstanta tambahan.
3. **Output**
→ Menampilkan kedua nilai konstanta dengan echo.

◆ Kelebihan:

- **✓ Aman dari perubahan nilai:** Konstanta tidak bisa ditimpas secara tidak sengaja.
- **✓ Rapi dan terstruktur:** Memudahkan pemeliharaan program, terutama untuk nilai tetap seperti nama instansi, mata kuliah, dsb.
- **✓ Meningkatkan keterbacaan kode:** Nama konstanta mewakili isi nilainya secara jelas.

◆ Kekurangan:

- **✗ Tidak bisa menyimpan nilai yang akan berubah atau dinamis (misalnya nama pengguna).**
- **✗ Tidak bisa menggunakan konstanta di dalam class secara fleksibel tanpa const (perlu pendekatan berbeda).**

◆ Kesimpulan:

Kode ini sangat cocok sebagai dasar untuk memahami penggunaan **konstanta global** dalam PHP, ideal untuk nilai tetap yang sering digunakan dalam aplikasi.

8. PHP Operators

a. Penjelasan

Operator digunakan untuk melakukan operasi, seperti perbandingan (==,>) dan logika (&&,||).

b. Contoh Kode File: operators.php

```
<?php  
  
$umur = 18;  
  
$sudah_punya_sim = false;  
  
if ($umur >= 17 && $sudah_punya_sim == true) {  
  
    echo "Anda boleh mengemudi.";  
  
} else {
```

```
echo "Anda tidak boleh mengemudi.";  
}  
?>
```

- Kode di atas di praktikkan:

Anda tidak boleh mengemudi.

- Ubah nilai variabel agar kondisi terpenuhi dan pesan “Anda boleh mengemudi” ditampilkan.

Dengan kode:

```
<?php  
$umur = 18;  
$sudah_punya_sim = true; // diubah dari false ke true  
  
if ($umur >= 17 && $sudah_punya_sim == true) {  
    echo "Anda boleh mengemudi.>";  
} else {  
    echo "Anda tidak boleh mengemudi.>";  
}  
?>
```

Dan jika diperlakukan:

Anda boleh mengemudi.

- Analisis:

Tujuan Kode:

Untuk menentukan apakah seseorang **boleh mengemudi** berdasarkan **usia** dan **kepemilikan SIM**.

◆ **Struktur Logika:**

php

CopyEdit

if (\$umur >= 17 && \$sudah_punya_sim == true)

- **\$umur >= 17** → syarat minimal usia legal untuk mengemudi.
- **\$sudah_punya_sim == true** → memastikan sudah memiliki Surat Izin Mengemudi (SIM).
- Kedua kondisi **harus bernilai benar (true)** agar pernyataan dalam blok if dijalankan.

◆ **Perubahan:**

- Nilai \$sudah_punya_sim diubah dari false ke true agar kondisi terpenuhi.
- ◆ **Hasil:**
- Menampilkan: "**Anda boleh mengemudi.**"

Kesimpulan:

Kode ini merupakan contoh sederhana penggunaan **percabangan if-else** dan operator logika **AND (&&)** dalam PHP.

Cocok untuk skenario keputusan berbasis banyak syarat. Sangat bermanfaat dalam validasi data pada aplikasi berbasis form atau sistem izin akses.

9. PHP if Statements

a. Penjelasan

Struktur if...elseif...else digunakan untuk membuat keputusan berdasarkan berbagai kondisi.

b. Contoh Kode File: if_statements.php

```
1. <?php
2. $nilai_ujian = 82;
3. if ($nilai_ujian >= 85) {
4.     echo "Sangat Baik (A)";
5. } elseif ($nilai_ujian >= 75) {
6.     echo "Baik (B)";
7. } else {
8.     echo "Cukup (C)";
9. }
10. ?>
```

c. Latihan Praktikum

- Kode di atas jika dipraktikkan:
Baik (B)
- Buat if statement untuk memeriksa apakah sebuah angka positif, negatif, atau nol.
Kode:

```
<?php
$angka = -7;

if ($angka > 0) {
    echo "Angka positif";
} elseif ($angka < 0) {
    echo "Angka negatif";
} else {
    echo "Angka nol";
}
?>
```

Dan jika dipraktikkan:

Angka negatif

- Analisis:
🎯 Tujuan:
Menentukan **kategori nilai sebuah angka**: apakah **positif**, **negatif**, atau **nol**.
🔍 Logika Kondisional:
 - if (\$angka > 0) → Mengecek apakah angka lebih besar dari nol.
 - elseif (\$angka < 0) → Mengecek apakah angka lebih kecil dari nol.
 - else → Menangani kemungkinan terakhir, yaitu angka **nol**.
- **Kelebihan:**

- Struktur kode jelas dan mudah dibaca.
- Menangani semua kemungkinan input angka (positif, negatif, nol).
- Berguna untuk validasi numerik dasar, seperti dalam kalkulator, input user, atau sistem monitoring.

 **Kesimpulan:**

Kode ini merupakan contoh praktis penggunaan **struktur percabangan** dalam PHP. Cocok untuk pemula yang ingin memahami dasar-dasar pengambilan keputusan berdasarkan kondisi numerik.

10. PHP switch Statement

a. Penjelasan

Switch adalah alternatif if-elseif-else yang lebih efisien untuk satu variabel dengan banyak kemungkinan nilai.

b. Contoh Kode File: switch.php

```

1. <?php
2. $level_user = "admin";
3. switch ($level_user) {
4.     case "admin":
5.         echo "Selamat datang, Administrator!";
6.         break;
7.     case "editor":
8.         echo "Selamat datang, Editor!";
9.         break;
10.    default:
11.        echo "Selamat datang, Pengguna!";
12. }
13. ?>
```

c. Latihan praktikum

- Kode di atas jika dipraktikkan:
Selamat datang, Administrator!
- Buat switch statement untuk menampilkan pesan berbeda berdasarkan \$ukuran_baju (“S”, “M”, “L”, “XL”).
Berikut kodenya:

```

<?php
$ukuran_baju = "M";

switch ($ukuran_baju) {
    case "S":
        echo "Ukuran Small - cocok untuk anak-anak atau remaja.";
        break;
    case "M":
        echo "Ukuran Medium - pas untuk orang dewasa berpostur sedang.";
        break;
    case "L":
        echo "Ukuran Large - cocok untuk postur tubuh besar.";
```

```

        break;
    case "XL":
        echo "Ukuran Extra Large - untuk tubuh sangat besar.";
        break;
    default:
        echo "Ukuran tidak tersedia.";
    }
?>
```

Jika dipraktikkan:

Ukuran Medium - pas untuk orang dewasa berpostur sedang.

- Analisis:

 **Tujuan:**

Kode ini menggunakan struktur switch untuk menentukan pesan yang sesuai berdasarkan nilai dari variabel \$ukuran_baju, seperti "S", "M", "L", atau "XL".

 **Logika Program:**

- switch (\$ukuran_baju) akan mencocokkan nilai dengan setiap case.
- Jika cocok, maka pesan yang relevan akan ditampilkan menggunakan echo.
- break mencegah eksekusi lanjut ke case lain.
- default menangani nilai yang tidak sesuai dengan case yang tersedia.

 **Kelebihan:**

-  Lebih efisien dan rapi dibandingkan if-elseif saat menangani banyak nilai tetap.
-  Mudah dibaca dan dimodifikasi jika perlu menambahkan ukuran baru (misalnya XXL).
-  Cocok digunakan dalam **form input pengguna, stok produk, atau pengaturan preferensi ukuran.**

 **Kesimpulan:**

Struktur switch dalam contoh ini sangat tepat digunakan untuk seleksi nilai diskrit seperti ukuran baju. Ini membantu membuat program lebih terorganisir, khususnya ketika menangani banyak pilihan yang saling eksklusif.

11. PHP Loops

a. Penjelasan

Perulangan (for, while, foreach) digunakan untuk mengeksekusi blok kode berulang kali.

b. Contoh kode file: Loops.php

```

<?php
// Perulangan for
for ($i = 0; $i < 5; $i++) {
    echo "Perulangan ke- " . ($i + 1) . "<br>";
}

// Perulangan while
```

```

$angka = 5;
while ($angka > 0) {
    echo "Hitung mundur: $angka <br>";
    $angka--;
}
?>

```

c. Latihan Praktikum:

- Silahkan praktikkan contoh kode di atas.

[Perulangan ke-1](#)

[Perulangan ke-2](#)

[Perulangan ke-3](#)

[Perulangan ke-4](#)

[Perulangan ke-5](#)

Hitung mundur: 5

Hitung mundur: 4

Hitung mundur: 3

Hitung mundur: 2

Hitung mundur: 1

- Gunakan foreach untuk menampilkan semua elemen dari sebuah array.

Berikut kodennya:

```

<?php
$buah = ["Apel", "Jeruk", "Mangga", "Pisang"];

foreach ($buah as $item) {
    echo "Nama buah: $item<br>";
}
?>

```

Dengan hasil praktiknya:

Nama buah: Apel
Nama buah: Jeruk
Nama buah: Mangga
Nama buah: Pisang

- Analisis:

Tujuan:

Menampilkan seluruh elemen dari array \$buah menggunakan perulangan foreach.

Logika Program:

- \$buah adalah array berisi elemen string: "Apel", "Jeruk", "Mangga", dan "Pisang".
- foreach (\$buah as \$item) menelusuri setiap elemen dalam array dan menyimpannya sementara di variabel \$item.

- Setiap nilai \$item ditampilkan menggunakan echo.

 **Kelebihan:**

-  foreach sangat cocok digunakan untuk **array** karena tidak perlu menggunakan indeks atau panjang array secara manual.
-  Membuat kode lebih **ringkas**, **mudah dibaca**, dan **minim kesalahan** saat memproses elemen satu per satu.
-  Cocok untuk data dinamis seperti daftar produk, nama siswa, hasil query database, dsb.

 **Kesimpulan:**

foreach adalah struktur perulangan yang efisien dan praktis untuk menelusuri semua isi array. Dalam konteks pemrograman web, ini adalah salah satu cara terbaik untuk menampilkan data list secara dinamis.

12. PHP Functions

a. Penjelasan

Fungsi adalah blok kode yang dapat digunakan kembali. Dapat menerima parameter dan mengembalikan nilai.

b. Contoh Kode File: functions.php

```
<?php
function hitung_diskon($harga, $persen_diskon) {
    $potongan = $harga * ($persen_diskon / 100);
    return $harga - $potongan;
}

$harga_baju = 150000;
$harga_setelah_diskon = hitung_diskon($harga_baju, 10); // Diskon 10%
echo "Harga akhir: Rp " . $harga_setelah_diskon;
?>
```

c. Latihan Praktikum

- Jika kode di atas di praktikkan:

Harga akhir: Rp 135000

- Buat fungsi sapa(\$nama, \$waktu) yang menampilkan pesan seperti “Selamat Pagi, Budi!”.

Berikut codenya:

```
<?php
function sapa($nama, $waktu) {
    echo "Selamat $waktu, $nama!";
}

// Contoh pemanggilan fungsi
sapa("Budi", "Pagi");

?>
```

Jika dipraktikkan:

Selamat Pagi, Budi!

- Analisis:

Tujuan:

Membuat fungsi yang menghasilkan **pesan sapaan personal** berdasarkan nama dan waktu (misal: pagi, siang, sore, malam).

Struktur Fungsi:

- **function** **sapa(\$nama, \$waktu)**

Menerima dua parameter: \$nama (nama orang) dan \$waktu (periode waktu).

- **echo** **"Selamat \$waktu, \$nama!"**;

Menampilkan pesan dengan format dinamis berdasarkan input.

Kelebihan:

-  Sederhana dan mudah digunakan.
-  Fleksibel untuk berbagai situasi, misalnya sapa("Ani", "Malam").
-  Cocok digunakan di halaman web dinamis seperti dashboard, aplikasi absen, atau e-learning.

Kesimpulan:

Fungsi ini adalah contoh penerapan **parameter dinamis** dalam fungsi PHP.

Berguna untuk membuat aplikasi yang lebih interaktif dan personal.

13. PHP Arrays

a. Penjelasan

Array adalah variabel yang bisa menampung banyak nilai, baik dengan indeks numerik (indexed array) maupun nama (associative array).

b. Contoh Kode File: arrays.php

```
<?php  
// Associative Array  
$data_dosen = [  
    "nama" => "Ir. ABADI NUGROHO, S.Kom., M.Kom.",  
    "nidn" => "1104129002",  
    "matkul" => "Pemrograman Web"  
];  
  
echo "Dosen Pengampu: " . $data_dosen["nama"];  
?>
```

c. Latihan Praktikum:

- Mempraktikkan kode di atas:

Dosen Pengampu: Ir. ABADI NUGROHO, S.Kom., M.Kom.

- Buat sebuah *indexed array* yang berisi daftar nama teman sekelas Anda dan tampilkan menggunakan foreach.

Berikut kodennya:

```
<?php  
$teman_kelas = ["Andi", "Budi", "Citra", "Dewi", "Eko"];  
  
foreach ($teman_kelas as $nama) {  
    echo "Teman sekelas: $nama<br>";  
}  
?>
```

Jika dipraktikkan:

Teman sekelas: Andi
Teman sekelas: Agung
Teman sekelas: Marton
Teman sekelas: Gempur
Teman sekelas: Iwan

- Analisis:

Tujuan:

Menampilkan daftar nama teman sekelas yang disimpan dalam bentuk **indexed array** menggunakan perulangan foreach.

Struktur Program:

- **Array** **\$teman_kelas**
Merupakan array dengan indeks numerik otomatis (0, 1, 2, dst), berisi elemen string seperti "Andi", "Budi", dll.
 - **foreach** **(\$teman_kelas)** **as** **\$nama**
Mengeksekusi blok kode sebanyak jumlah elemen array. Setiap elemen disimpan sementara dalam variabel \$nama dan ditampilkan.
-

Kelebihan:

-  Indexed array mudah digunakan saat urutan data penting atau tidak memerlukan key khusus.
 -  foreach menyederhanakan iterasi tanpa perlu menghitung panjang array (count()).
 -  Cocok digunakan dalam data daftar, absensi, maupun sistem presensi online.
-

Kekurangan:

-  Tidak menyimpan informasi tambahan selain nama; jika butuh data tambahan (seperti NIM, jurusan), lebih cocok pakai **associative array** atau **array multidimensi**.
-

Kesimpulan:

Contoh ini menunjukkan implementasi dasar foreach dengan **indexed array**, cocok untuk pemula yang ingin memahami struktur data linear dalam PHP. Efisien untuk menampilkan list sederhana seperti nama siswa, produk, atau entri data lainnya.

14. PHP Global Variables – Superglobals

a. Penjelasan

Superglobals adalah variabel bawaan PHP yang selalu dapat diakses. `$_POST` dan `$_GET` digunakan untuk mengambil data dari form.

b. Contoh Kode File: Superglobals.php (form yang mengirim data ke dirinya sendiri)

```
<!DOCTYPE html>
<html>
<body>
<form method="post" action="14_superglobals.php">
Nama: <input type="text" name="nama">
<input type="submit">
</form>
<br>
<?php
if (isset($_POST['nama'])) {
$nama = htmlspecialchars($_POST['nama']);
echo "Halo, " . $nama . "!";
}
?>
</body>
</html>
```

c. Latihan Praktikum:

- Jika kode di atas di praktikkan:

Nama:

Tombol tidak bisa di submit karena 14_superglobals.php belum ada

- Coba ubah method form menjadi get dan ubah `$_POST` menjadi `$_GET`, lalu amati URL Browser

URL Browser akan menjadi:

localhost/14_superglobals.php?nama=Martonius

Ketika diketikkan Martonius dan di submit

- Analisis:

 **Tujuan:**

Menguji perbedaan penggunaan **POST** dan **GET** pada form HTML dalam mengirim data ke server dan mengamati dampaknya di **URL browser**.

Perubahan yang Dilakukan:

- method="post" diubah menjadi method="get".
 - Akses variabel dari \$_POST['nama'] diubah menjadi \$_GET['nama'].
-

Hasil Pengamatan:

Setelah pengguna mengisi form dan klik submit:

- URL berubah menjadi:
14_superglobals.php?nama=Martonius
 - Nilai input dikirim dan **terlihat secara langsung** di address bar browser.
-

Perbandingan GET vs POST:

Aspek	GET	POST
Visibilitas data	Terlihat di URL (?nama=...)	Tidak terlihat di URL
Keamanan	Kurang aman (mudah dimanipulasi)	Lebih aman untuk data sensitif
Ukuran data	Terbatas (sekitar 2KB)	Lebih besar (tergantung server)
Cocok untuk	Filter, pencarian, navigasi berbasis URL	Login, registrasi, pengisian form
Bookmarkable	 Ya, karena URL menyimpan data	 Tidak, karena data tidak ada di URL

Kelebihan:

- GET berguna saat pengguna ingin menyimpan atau membagikan hasil form via URL.
- Data lebih transparan dan mudah di-debug saat testing.

Kekurangan:

- Tidak cocok untuk mengirim **data pribadi atau sensitif**.
 - Mudah dimodifikasi di address bar, sehingga rawan manipulasi.
-

Kesimpulan:

Perubahan metode form dari POST ke GET mengungkapkan perbedaan penting terkait **keamanan, transparansi, dan kegunaan URL**. Pilihan antara keduanya harus disesuaikan dengan konteks: gunakan GET untuk keperluan ringan yang tidak sensitif, dan POST untuk data penting atau rahasia.

D. Tugas Praktikum

Buatlah sebuah halaman web "**Buku Tamu Sederhana**" yang dinamis menggunakan PHP dengan ketentuan sebagai berikut:

Struktur Halaman

1. **Header:** Berisi judul website, misalnya "Buku Tamu Digital STITEK Bontang".
2. **Form Input:**
 - Form harus memiliki kolom input untuk **Nama Lengkap, Alamat Email, dan Pesan/Komentar.**
 - Form harus memiliki tombol **Submit** dengan tulisan "Kirim Pesan".
3. **Area Tampilan Data:**
 - Di bawah form, tampilkan pesan konfirmasi dan data yang baru saja dikirim oleh pengguna.

Fungsionalitas

1. Halaman harus berupa satu file PHP yang memproses formnya sendiri (*self-processing form*).
2. Gunakan metode POST untuk mengirimkan data form.
3. Terapkan validasi sederhana di sisi server:
 - Pastikan tidak ada kolom input yang kosong. Jika ada yang kosong, tampilkan pesan kesalahan yang jelas.
 - Gunakan htmlspecialchars() untuk membersihkan input sebelum menampilkannya untuk mencegah serangan XSS sederhana.
4. Jika semua data valid, tampilkan data yang dikirim dengan format yang rapi.

Desain dan Styling

- Gunakan CSS internal atau eksternal untuk mempercantik tampilan form dan area hasil.
- Pastikan halaman memiliki tata letak yang bersih, rapi, dan mudah dibaca.

Tambahan

- Simpan kode di repositori GitHub Anda masing-masing.
- Buatlah video untuk menjelaskan skrip dari tugas yang Anda buat, upload kedalam YouTube lalu sertakan link GitHub dan video di laporan Anda.

Rubrik Penilaian

Aspek Penilaian	Keterangan	Bobot (%)
Struktur HTML	Form dibuat dengan benar menggunakan input, label, dan button yang sesuai.	15%
Fungsionalitas PHP	Logika pengolahan <code>\$_POST</code> , validasi (pengecekan kolom kosong), dan penggunaan <code>htmlspecialchars()</code> berjalan dengan baik.	35%
Desain & Tampilan	Halaman memiliki tata letak yang rapi, bersih, dan mudah digunakan (user-friendly).	15%
Kerapihan Kode	Kode terstruktur, diberi komentar seperlunya, dan mudah dibaca.	10%
Dokumentasi	Menyertakan link GitHub yang valid dan video penjelasan yang jelas dan informatif.	25%

Jawab: berikut full kode tugasphp.php

```
<!DOCTYPE html>
<html lang="id">
<head>
<meta charset="UTF-8">
<title>Buku Tamu Digital STITEK Bontang</title>
<style>
body {
    font-family: Arial, sans-serif;
    background: #f4f4f4;
    margin: 0;
    padding: 20px;
}
.container {
    max-width: 600px;
    background: #fff;
    margin: 0 auto;
    padding: 30px;
    border-radius: 10px;
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}
h1 {
    text-align: center;
    color: #2c3e50;
}
label {
    display: block;
    margin-top: 15px;
    font-weight: bold;
}
input[type="text"], input[type="email"], textarea {
    width: 100%;
    padding: 10px;
    margin-top: 5px;
    border-radius: 5px;
    border: 1px solid #ccc;
```

```

        }
    
```

```

    textarea {
        resize: vertical;
    }

```

```

    .btn {
        margin-top: 20px;
        background-color: #2c3e50;
        color: #fff;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
    }

```

```

    .btn:hover {
        background-color: #34495e;
    }

```

```

    .output {
        margin-top: 30px;
        background: #ecff0f1;
        padding: 20px;
        border-radius: 5px;
    }

```

```

    .error {
        color: red;
    }

```

```

</style>

```

```

</head>

```

```

<body>

```

```

<div class="container">

```

```

    <h1>Buku Tamu Digital STITEK Bontang</h1>

```

```

    <form method="post" action="">

```

```

        <label for="nama">Nama Lengkap</label>
        <input type="text" id="nama" name="nama">

```

```

        <label for="email">Alamat Email</label>
        <input type="email" id="email" name="email">

```

```

<label for="pesan">Pesanan / Komentar</label>
<textarea id="pesan" name="pesan" rows="5"></textarea>

<button type="submit" class="btn">Kirim Pesan</button>
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = trim($_POST['nama']);
    $email = trim($_POST['email']);
    $pesan = trim($_POST['pesan']);
    $errors = [];

    if (empty($nama)) $errors[] = "Nama tidak boleh kosong.";
    if (empty($email)) $errors[] = "Email tidak boleh kosong.";
    if (empty($pesan)) $errors[] = "Pesanan tidak boleh kosong.";

    if (!empty($errors)) {
        echo '<div class="output error">';
        foreach ($errors as $e) {
            echo "<p>• $e</p>";
        }
        echo '</div>';
    } else {
        // Sanitasi input
        $nama = htmlspecialchars($nama);
        $email = htmlspecialchars($email);
        $pesan = htmlspecialchars($pesan);

        echo '<div class="output">';
        echo "<h3>Terima kasih atas pesanan Anda!</h3>";
        echo "<p><strong>Nama:</strong> $nama</p>";
        echo "<p><strong>Email:</strong> $email</p>";
        echo "<p><strong>Pesanan:</strong><br>$pesan</p>";
        echo '</div>';
    }
}

```

```
    }
}
?>
</div>
</body>
</html>
```

Berikut tampilannya jika dijalankan dan diisi form dan disubmit:

Buku Tamu Digital STITEK Bontang

Nama Lengkap

Alamat Email

Pesan / Komentar

Kirim Pesan

Terima kasih atas pesan Anda!

Nama: Martonius Epta Sidabutar
Email: d3athnotez@gmail.com
Pesan:
Saya pinjam buku C++

Berikut penjelasan video pada link github: [Link/Tautan](#)

Modul 6. Pemrograman Web dengan PHP dan MySQL

A. Tujuan

- Setelah mengikuti praktikum, Mahasiswa diharapkan mampu:
1. Mampu mengenal dan memahami bentuk dasar skrip dan variabel-variabel PHP.
 2. Mampu menggunakan *statement* kondisional (if-else) dan pengulangan (for, while) dengan PHP.
 3. Mampu memahami dan mengelola database menggunakan MySQL dengan bantuan Laragon dan phpMyAdmin.
 4. Dapat merancang dan membuat aplikasi web dinamis sederhana yang menghubungkan PHP dengan database MySQL untuk melakukan operasi CRUD (Create, Read, Update, Delete).

B. Teori Dasar PHP dan MySQL

PHP (Hypertext Preprocessor) adalah bahasa skrip open-source yang dirancang khusus untuk pengembangan web. Kode PHP dieksekusi di server, dan hasilnya dikirim ke browser dalam bentuk HTML biasa. Ini memungkinkan pembuatan halaman web yang dinamis dan interaktif.

MySQL adalah sistem manajemen basis data relasional (RDBMS) open-source yang menggunakan Structured Query Language (SQL). MySQL sangat populer untuk aplikasi web karena keandalan, kecepatan, dan kemudahan penggunaannya.

Laragon adalah lingkungan pengembangan universal untuk PHP, Node.js, Python, Java, Go, dan Ruby. Ini menyediakan cara yang cepat dan mudah untuk membuat lingkungan pengembangan lokal yang terisolasi, termasuk server web Apache, MySQL, dan alat lainnya seperti phpMyAdmin.

C. Pratikum

1. Instalasi dan Pengaturan Lingkungan (Laragon)

Penjelasan sebelum mulai, kita perlu menyiapkan server lokal. Laragon akan menginstal semua yang kita butuhkan: server web (Apache), database (MySQL), dan PHP. **Langkah-langkah Instalasi:**

- Unduh Laragon dari situs resminya (<https://laragon.org/download>)
- Jalankan Installer dan ikuti petunjuknya. Anda dapat membiarkan pengaturan default.

- Setelah instalasi selesai, jalankan Laragon
- Klik tombol “**Start All**” untuk menjalankan server Apache dan MySQL
- Folder www (biasanya di C:\laragon\www) adalah tempat Anda akan menyimpan semua file proyek PHP Anda.

2. Menjalankan File PHP di Browser

Penjelasan

Tidak seperti file HTML biasa yang bisa dibuka langsung, file PHP harus diakses melalui server web agar kodennya dapat dieksekusi. Laragon bertindak sebagai server web lokal Anda. **Langkah-langkah Menjalankan File PHP:**

- **Buat Folder Proyek:** Buka direktori www Laragon Anda (misalnya C:\laragon\www). Buat sebuah folder baru untuk proyek praktikum ini, misalnya dengan nama praktikum_php.
- **Buat File PHP:** Di dalam folder praktikum_php, buat sebuah file baru dengan nama halo.php menggunakan teks editor (seperti Visual Studio Code, Notepad++, dll.).
- **Isi Kode:** Salin dan tempel kode berikut ke dalam file halo.php lalu simpan

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <title>Tes PHP</title>
5. </head>

6. <body>
7.   <h1><?php echo "Halo dari Server PHP!"; ?></h1>
8.   <p>Jika Anda melihat teks ini, artinya server PHP Anda berjalan dengan baik.</p>
9. </body>
10. </html>
```

- **Buka di Browser:** Buka browser web Anda (seperti Chrome atau Firefox) dan ketikkan alamat berikut http://localhost/praktikum_php/halo.php
- **Verifikasi Hasil:** Anda akan melihat halaman web dengan tulisan "Halo dari Server PHP!". Ini menandakan bahwa Laragon berhasil memproses file PHP Anda.

Berikut hasilnya:

Halo dari Server PHP!

Jika Anda melihat teks ini, artinya server PHP Anda berjalan dengan baik.

3. Manajemen Database dengan phpMyAdmin

Penjelasan

Laragon menyertakan phpMyAdmin, alat berbasis web untuk mengelola database MySQL Anda. Membuat Database dan Tabel:

- Buka Laragon, lalu klik tombol "Database". Ini akan membuka phpMyAdmin di browser Anda.
- Buat database baru dengan mengklik "New", beri nama (misalnya, db_perusahaan), dan klik "Create".
- Di dalam database baru Anda, buat tabel baru (misalnya, karyawan) dengan kolom yang diperlukan (misalnya: id, nama, jabatan, email). • Atur id sebagai PRIMARY KEY dan aktifkan A.I (AUTO_INCREMENT).

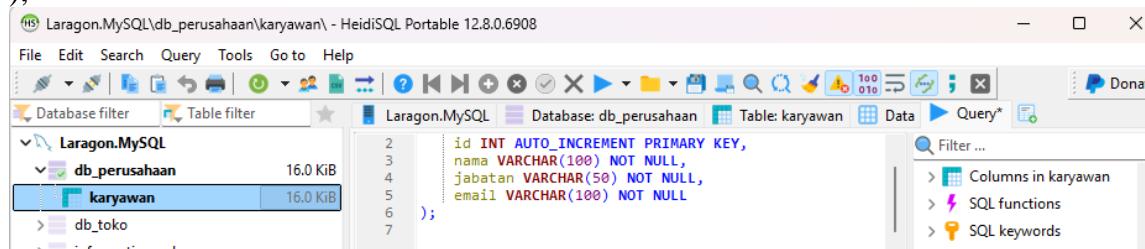
Contoh Kode

```
Sql  
CREATE DATABASE db_perusahaan;
```

```
USE db_perusahaan;
```

```
CREATE TABLE karyawan (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nama VARCHAR(100) NOT NULL,  
    jabatan VARCHAR(50) NOT NULL,  
    email VARCHAR(100) NOT NULL
```

```
);
```



4. Dasar-dasar Pemrograman PHP

Penjelasan

Pada bagian ini, kita akan mempelajari elemen-elemen fundamental dari bahasa pemrograman PHP yang akan menjadi dasar untuk membangun aplikasi yang lebih kompleks.

4.1 Mengenal Skrip dan Variabel PHP Kode PHP ditulis di antara tag . Variabel digunakan untuk menyimpan informasi dan selalu diawali dengan tanda dolar (\$). Perintah echo digunakan untuk menampilkan output ke browser.

File dasar_variabel.php

```
1. <!DOCTYPE html>
2. <html>
3.
4. <head>
5.   <title>Latihan Variabel PHP</title>
6. </head>
7.
8. <body>
9.   <h1>Mengenal Variabel</h1>
10.  <?php
11.    // Deklarasi variabel string (teks)
12.    $salam = "Halo, selamat pagi!";
13.
14.    // Deklarasi variabel integer (angka)
15.    $tahun = 2024;
16.
17.    echo $salam;
18.    echo "<br>"; // membuat baris baru
19.    echo "Sekarang adalah tahun " . $tahun;
20.  ?>
21. </body>
22.
23. </html>
```

Jika dibuka di browser akan menampilkan:

Mengenal Variabel

Halo, selamat pagi!
Sekarang adalah tahun 2024

4.2 Menggunakan Pernyataan Kondisional Pernyataan if-else memungkinkan program untuk membuat keputusan dan menjalankan kode yang berbeda berdasarkan kondisi yang benar atau salah.

File dasar_kondisi.php

```
1. <!DOCTYPE html>
2. <html>
3.
4. <head>
5.   <title>Latihan Kondisi PHP</title>
6. </head>
7.
8. <body>
9.   <h1>Cek Nilai</h1>
10.  <?php
11.  $nilai = 85;
12.  echo "<p>Nilai Anda: $nilai</p>";
13.
14.  if ($nilai >= 75) {
15.    echo "<p style='color:green;'>Selamat, Anda Lulus!</p>";
16.  } else {
17.    echo "<p style='color:red;'>Maaf, Anda perlu belajar lagi.</p>";
18.  }
19. ?
20. </body>
21.
22. </html>
```

jika dijalankan di browser akan menampilkan:

Cek Nilai

Nilai Anda: 85

Selamat, Anda Lulus!

4.3 Menggunakan Pernyataan Perulangan Perulangan digunakan untuk menjalankan blok kode yang sama berulang kali. for loop cocok digunakan ketika kita tahu jumlah perulangan yang pasti.

File dasar_perulangan.php

```
1. <!DOCTYPE html>
2. <html>
3.
4. <head>
5.   <title>Latihan Perulangan PHP</title>
6. </head>
7.
8. <body>
9.   <h1>Daftar Angka 1 sampai 5</h1>
10.  <ul>
11.    <?php
12.      for ($i = 1; $i <= 5; $i++) {
13.        echo "<li>Ini adalah item nomor " . $i . "</li>";
14.      }
15.    ?
16.  </ul>
17. </body>
18.
19. </html>
```

Jika dijalankan di browser akan menampilkan:

Daftar Angka 1 sampai 5

- Ini adalah item nomor 1
- Ini adalah item nomor 2
- Ini adalah item nomor 3
- Ini adalah item nomor 4
- Ini adalah item nomor 5

Latihan Pratikum (dasar-dasar PHP)

- Pratikkan ketiga contoh kode di atas (sudah)
- Modifikasi file dasar_kondisi.php dengan menambahkan kondisi elseif untuk kategori nilai (Contoh: > 90 = Sangat Baik, > 80 = Baik, > 70 = Cukup).

Berikut kode nya:

```
<!DOCTYPE html>
<html>
<head>
<title>Latihan Kondisi PHP</title>
</head>

<body>
<h1>Cek Nilai</h1>
<?php
$nilai = 85;
echo "<p>Nilai Anda: $nilai</p>";

if ($nilai > 90) {
    echo "<p style='color:blue;'>Kategori: Sangat Baik</p>";
} elseif ($nilai > 80) {
    echo "<p style='color:green;'>Kategori: Baik</p>";
} elseif ($nilai > 70) {
    echo "<p style='color:orange;'>Kategori: Cukup</p>";
} else {
    echo "<p style='color:red;'>Kategori: Tidak Lulus</p>";
}
?>
</body>
</html>
```

Jika diperlakukan:

Cek Nilai

Nilai Anda: 85

Kategori: Baik

- Berikan capture (sudah) dan berikan analisis:

Analisis:

1. **dasar_kondisi.php**: File ini memeriksa nilai suatu variabel dan menampilkan pesan berdasarkan kondisi. Jika nilai lebih dari 75, pesan yang ditampilkan adalah "Selamat, Anda Lulus!", jika tidak, akan muncul pesan "Maaf, Anda Tidak Lulus".
2. **dasar_kondisi1.php**: Mirip dengan file pertama, script ini mengkategorikan nilai siswa menggunakan beberapa kondisi. Jika nilai lebih dari 90, ditampilkan "Sangat Baik", antara 80-90 ditampilkan "Baik", antara 70-80 "Cukup", dan di bawah 70 akan tampil "Tidak Lulus".
3. **dasar_perulangan.php**: File ini menunjukkan penggunaan loop for dasar di PHP. Loop ini berjalan dari angka 1 hingga 5 dan menampilkan setiap angka sebagai elemen daftar. Ini adalah contoh dasar cara menggunakan perulangan untuk mengiterasi suatu rentang.
4. **dasar_variabel.php**: Memperkenalkan variabel PHP, termasuk tipe string dan integer. Script ini menampilkan pesan sapaan dan tahun saat ini menggunakan variabel-variabel tersebut. Ini menunjukkan penugasan variabel dan cara menampilkan output dasar.
5. **halo.php**: File PHP sederhana yang menampilkan pesan sapaan "Halo dari Server PHP!". File ini digunakan sebagai tes dasar untuk memastikan lingkungan PHP sudah terpasang dengan benar.
6. **koneksi.php**: File ini menghubungkan ke database MySQL menggunakan MySQLi. File ini memberikan umpan balik tentang status koneksi, baik itu sukses atau menampilkan pesan kesalahan jika koneksi gagal.
7. **tugasphp.php**: File ini mengimplementasikan form "Digital Guestbook" di mana pengguna dapat mengirimkan nama, email, dan pesan. File ini melakukan validasi dasar (memeriksa kolom yang kosong) dan menyaring input untuk mencegah serangan XSS sebelum menampilkan data yang dikirimkan.

Masing-masing file ini menunjukkan konsep-konsep dasar PHP seperti pernyataan kondisi, perulangan, variabel, koneksi database, penanganan form, dan validasi input.

5. Menghubungkan PHP ke Database MySQL

Penjelasan

Untuk berinteraksi dengan database, PHP perlu membuat koneksi terlebih dahulu. Kita akan menggunakan ekstensi mysqli yang merupakan cara modern dan lebih aman.

Contoh Kode Koneksi File koneksi.php

```
<?php
// Define database connection parameters
$servername = "localhost"; // Database server (usually localhost for local servers)
$username = "root";      // Database username (default for local servers is usually 'root')
$password = "";           // Database password (default for local servers is usually empty)
$dbname = "db_perusahaan"; // Database name (the name of your database)

// Create a connection to the MySQL server
$conn = new mysqli($servername, $username, $password, $dbname);

// Check if the connection was successful
if ($conn->connect_error) {
    // If connection fails, display the error message and stop execution
    die("Connection failed: " . $conn->connect_error);
} else {
    // If connection is successful, display a message
    echo "Connected successfully to the database!";
}
?>
```

6. Menampilkan Data dari Database (Read)

Penjelasan Setelah terhubung, kita dapat mengambil data dari tabel menggunakan query SELECT dan menampilkannya di halaman web menggunakan perulangan while.

Contoh Kode Menampilkan Data File index.php

```
<!DOCTYPE html>
<html>

<head>
    <title>Data Karyawan</title>
</head>

<body>
    <h2>Daftar Karyawan Perusahaan</h2>
    <a href="form_tambah.html">Tambah Karyawan Baru</a>
    <br><br>
    <table border="1" cellpadding="5" cellspacing="0">
        <thead>
            <tr>
                <th>ID</th>
                <th>Nama Lengkap</th>
                <th>Jabatan</th>
                <th>Email</th>
                <th>Aksi</th>
            </tr>
        <tbody>
```

```

</thead>
<tbody>
<?php
include 'koneksi.php'; // Menyertakan file koneksi
$sql = "SELECT id, nama, jabatan, email FROM karyawan";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // Loop untuk menampilkan setiap baris data
    while ($row = $result->fetch_assoc()) {
        echo "<tr>";
        echo "<td>" . $row["id"] . "</td>";
        echo "<td>" . $row["nama"] . "</td>";
        echo "<td>" . $row["jabatan"] . "</td>";
        echo "<td>" . $row["email"] . "</td>";
        echo "<td><a href='form_edit.php?id=" . $row["id"] .
">Edit</a> | <a href='hapus.php?id=" . $row["id"] . "' onclick='return
confirm(\"Apakah Anda yakin ingin menghapus data ini?\")'>Hapus</a></td>";
        echo "</tr>";
    }
} else {
    echo "<tr><td colspan='5'>Tidak ada data</td></tr>";
}
$conn->close();
?>
</tbody>
</table>
</body>

</html>

```

7. Menyisipkan, Memperbarui, dan Menghapus Data (Create, Update, Delete)

Penjelasan Bagian ini mencakup inti dari aplikasi web dinamis: mengizinkan pengguna untuk memanipulasi data di database melalui antarmuka web

7.1 Membuat Form Tambah Data (Create) File *form_tambah.html*

```

<!DOCTYPE html>
<html>
<head>
    <title>Tambah Karyawan</title>
</head>
<body>
    <h2>Form Tambah Karyawan Baru</h2>
    <form action="proses_tambah.php" method="post">
        Nama: <input type="text" name="nama" required><br><br>
        Jabatan: <input type="text" name="jabatan" required><br><br>
        Email: <input type="email" name="email" required><br><br>
        <input type="submit" value="Simpan">
    </form>
</body>
</html>

```

7.2 Memproses dan Menyimpan Data Baru file *proses_tambah.php*

```

<?php
include 'koneksi.php';

// Mengambil data dari form
$nama = $_POST['nama'];
$jabatan = $_POST['jabatan'];
$email = $_POST['email'];
// Query untuk menyisipkan data
$sql = "INSERT INTO karyawan (nama, jabatan, email) VALUES ('$nama', '$jabatan', '$email')";
if ($conn->query($sql) === TRUE) {
    header("Location: index.php"); // Arahkan kembali ke halaman utama jika berhasil
    exit();
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();

```

7.3 Mengedit Data (Update) File *form_edit.php*

```

<?php
include 'koneksi.php';

// Memeriksa apakah 'id' ada di parameter URL
if (isset($_GET['id'])) {
    $id = $_GET['id'];

    // Menjalankan query untuk mengambil data berdasarkan 'id'
    $sql = "SELECT * FROM karyawan WHERE id=$id";
    $result = $conn->query($sql);

    // Memastikan bahwa data ditemukan
    if ($result->num_rows > 0) {
        $row = $result->fetch_assoc();
    } else {
        echo "Data tidak ditemukan!";
        exit();
    }
} else {
    echo "ID tidak ditemukan di URL!";
    exit();
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Edit Karyawan</title>
</head>
<body>
    <h2>Form Edit Data Karyawan</h2>
    <form action="proses_edit.php" method="post">
        <input type="hidden" name="id" value="<?php echo $row['id']; ?>">
        Nama: <input type="text" name="nama" value="<?php echo $row['nama']; ?>" required><br><br>
        Jabatan: <input type="text" name="jabatan" value="<?php echo $row['jabatan']; ?>" required><br><br>
    </form>

```

```

Email: <input type="email" name="email" value=<?php echo $row['email']; ?>" required><br><br>
<input type="submit" value="Update">
</form>
</body>
</html>

```

7.4 Memproses dan Menyimpan Data yang Diubah *file proses_edit.php*

```

<?php
include 'koneksi.php';

// Memeriksa apakah semua data ada dalam $_POST
if (isset($_POST['id']) && isset($_POST['nama']) && isset($_POST['jabatan']) && isset($_POST['email'])) {
    // Mengambil data dari form
    $id = $_POST['id'];
    $nama = $_POST['nama'];
    $jabatan = $_POST['jabatan'];
    $email = $_POST['email'];

    // Query untuk memperbarui data
    $sql = "UPDATE karyawan SET nama='$nama', jabatan='$jabatan', email='$email' WHERE id=$id";

    if ($conn->query($sql) === TRUE) {
        header("Location: index.php"); // Arahkan kembali ke halaman utama jika berhasil
        exit();
    } else {
        echo "Error updating record: " . $conn->error; // Menampilkan error jika query gagal
    }
} else {
    echo "Data tidak lengkap!"; // Menampilkan pesan jika data tidak lengkap
}

$conn->close();

```

7.5 Menghapus Data (Delete) *file hapus.php*

```

<?php
include 'koneksi.php';

// Memeriksa apakah parameter 'id' ada di URL
if (isset($_GET['id'])) {
    // Mengambil id dari URL
    $id = $_GET['id'];

    // Query untuk menghapus data
    $sql = "DELETE FROM karyawan WHERE id=$id";

    if ($conn->query($sql) === TRUE) {
        header("Location: index.php"); // Arahkan kembali ke halaman utama jika berhasil
    }
}

```

```
    exit();
} else {
    echo "Error deleting record: " . $conn->error; // Menampilkan pesan jika query gagal
}
} else {
    echo "ID tidak ditemukan di URL!"; // Menampilkan pesan jika parameter 'id' tidak ada
}

$conn->close();
```

D. Tugas Praktikum

Buatlah sebuah aplikasi CRUD (*Create, Read, Update, Delete*) sederhana untuk mengelola data produk pada sebuah toko online.

Ketentuan:

1. **Database:** Buat database baru bernama db_toko dengan sebuah tabel bernama produk yang memiliki kolom: id_produk (INT, PRIMARY, AUTO_INCREMENT), nama_produk (VARCHAR), harga (INT), dan stok (INT).
2. **Halaman Utama (index.php):**
 - Menampilkan semua data produk dalam sebuah tabel.
 - Setiap baris data memiliki tombol "Edit" dan "Hapus".
 - Terdapat tombol/link "Tambah Produk Baru" yang mengarah ke form tambah data.
3. **Halaman Tambah Produk (tambah.php):**
 - Berisi form untuk memasukkan nama_produk, harga, dan stok.
 - Data yang di-submit akan disimpan ke database, kemudian pengguna diarahkan kembali ke halaman utama.
4. **Halaman Edit Produk (edit.php):**
 - Form akan terisi otomatis dengan data produk yang dipilih untuk diedit.
 - Setelah data diperbarui, pengguna diarahkan kembali ke halaman utama.

5. Fungsionalitas Hapus (hapus.php):

- Ketika tombol "Hapus" diklik, data produk yang sesuai akan dihapus dari database.
- Pengguna kemudian diarahkan kembali ke halaman utama.

Tambahan:

- Simpan semua kode proyek Anda di repositori GitHub.
- Buat video singkat (maksimal 5 menit) yang menjelaskan struktur kode dan mendemonstrasikan cara kerja aplikasi CRUD Anda.
- Sertakan link GitHub dan link video di dalam laporan praktikum Anda.

Rubrik Penilaian

Aspek Penilaian	Keterangan	Bobot (%)
Struktur Database	Database dan tabel dibuat dengan benar sesuai instruksi (nama, kolom, tipe data, primary key).	10%
Fungsionalitas Crud	Semua fungsi (Create, Read, Update, Delete) berjalan tanpa eror dan sesuai alur.	30%
Logika & Kode Php	Kode PHP untuk koneksi, pemrosesan data, dan logika aplikasi ditulis dengan benar dan efisien.	15%
Integrasi Bootstrap	Tampilan menarik, rapi, dan responsif di berbagai perangkat berkat penggunaan kelas Bootstrap.	10%
Kerapian Kode	Kode terstruktur dengan baik, mudah dibaca, dan diberi komentar jika diperlukan.	5%
Dokumentasi	Tugas didokumentasikan dengan baik (disimpan di GitHub dan disertai video penjelasan).	30%
Total		100%

Jawab: Berikut kode yang diperlukan:

Struktur Database (db_toko)

```
Sql
CREATE DATABASE db_toko;
USE db_toko;

CREATE TABLE produk (
    id_produk INT AUTO_INCREMENT PRIMARY KEY,
    nama_produk VARCHAR(100),
    harga INT,
    stok INT
);
```

Koneksi Database (koneksi.php)

```
<?php
$host = "localhost";
$user = "root";
$pass = "";
$db = "db_toko";

$conn = new mysqli($host, $user, $pass, $db);

if ($conn->connect_error) {
    die("Koneksi gagal: " . $conn->connect_error);
}
?>
```

Halaman Utama (indeks.php)

```
<?php include 'koneksi.php'; ?>
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <title>Data Produk</title>
```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body class="container mt-5">
    <h2>Daftar Produk</h2>
    <a href="tambah.php" class="btn btn-primary mb-3">Tambah Produk Baru</a>
    <table class="table table-bordered table-striped">
        <tr>
            <th>ID</th><th>Nama Produk</th><th>Harga</th><th>Stok</th><th>Aksi</th>
        </tr>
        <?php
            $result = $conn->query("SELECT * FROM produk");
            while ($row = $result->fetch_assoc()):
                ?>
                <tr>
                    <td><?= $row['id_produk'] ?></td>
                    <td><?= $row['nama_produk'] ?></td>
                    <td><?= number_format($row['harga']) ?></td>
                    <td><?= $row['stok'] ?></td>
                    <td>
                        <a href="edit.php?id=<?= $row['id_produk'] ?>" class="btn btn-sm btn-warning">Edit</a>
                        <a href="hapus.php?id=<?= $row['id_produk'] ?>" onclick="return
                        confirm('Yakin hapus?')" class="btn btn-sm btn-danger">Hapus</a>
                    </td>
                </tr>
            <?php endwhile; ?>
        </table>
    </body>
</html>

```

Tambah produk (tambah.php)

```

<?php
include 'koneksi.php';
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = $_POST['nama_produk'];
    $harga = $_POST['harga'];
}

```

```

$stok = $_POST['stok'];
$conn->query("INSERT INTO produk (nama_produk, harga, stok) VALUES ('$nama',
'$harga', '$stok')");
header("Location: index.php");
}
?>
<!DOCTYPE html>
<html lang="id">
<head>
<meta charset="UTF-8">
<title>Tambah Produk</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body class="container mt-5">
<h2>Tambah Produk</h2>
<form method="post">
<div class="mb-3">
<label>Nama Produk</label>
<input type="text" name="nama_produk" class="form-control" required>
</div>
<div class="mb-3">
<label>Harga</label>
<input type="number" name="harga" class="form-control" required>
</div>
<div class="mb-3">
<label>Stok</label>
<input type="number" name="stok" class="form-control" required>
</div>
<button class="btn btn-success">Simpan</button>
<a href="index.php" class="btn btn-secondary">Batal</a>
</form>
</body>
</html>

```

Edit Produk (edit.php)

```
<?php
include 'koneksi.php';
$id = $_GET['id'];
$data = $conn->query("SELECT * FROM produk WHERE id_produk = $id")-
>fetch_assoc();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nama = $_POST['nama_produk'];
    $harga = $_POST['harga'];
    $stok = $_POST['stok'];
    $conn->query("UPDATE produk SET nama_produk='$nama', harga='$harga',
stok='$stok' WHERE id_produk=$id");
    header("Location: index.php");
}
?>
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <title>Edit Produk</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body class="container mt-5">
    <h2>Edit Produk</h2>
    <form method="post">
        <div class="mb-3">
            <label>Nama Produk</label>
            <input type="text" name="nama_produk" class="form-control" value="<?=
$data['nama_produk'] ?>" required>
        </div>
        <div class="mb-3">
            <label>Harga</label>
            <input type="number" name="harga" class="form-control" value="<?=
$data['harga'] ?>" required>
        </div>
    </form>
</body>
</html>
```

```

    </div>
    <div class="mb-3">
        <label>Stok</label>
        <input type="number" name="stok" class="form-control" value="<?= $data['stok'] ?>" required>
    </div>
    <button class="btn btn-warning">Update</button>
    <a href="index.php" class="btn btn-secondary">Batal</a>
</form>
</body>
</html>

```

Hapus Produk (hapus.php)

```

<?php
include 'koneksi.php';
$id = $_GET['id'];
$conn->query("DELETE FROM produk WHERE id_produk=$id");
header("Location: index.php");
?>

```

Style.css

```

/* style.css */
body {
    background-color: #f4f6f8;
    font-family: 'Arial', sans-serif;
}

.container {
    max-width: 900px;
    margin-top: 50px;
}

h2 {
    font-size: 24px;
    font-weight: bold;
    color: #333;
    text-align: center;
}

```

```
margin-bottom: 30px;
}

.table {
    background-color: #fff;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

.table th, .table td {
    text-align: center;
    vertical-align: middle;
    padding: 15px;
}

.table th {
    background-color: #007bff;
    color: white;
}

.table td {
    background-color: #f9f9f9;
}

.btn {
    border-radius: 5px;
    padding: 5px 15px;
    font-size: 14px;
}

.btn-primary {
    background-color: #28a745;
    border: none;
}

.btn-primary:hover {
```

```
background-color: #218838;  
}  
  
.btn-warning {  
    background-color: #ffc107;  
    border: none;  
}  
  
.btn-warning:hover {  
    background-color: #e0a800;  
}  
  
.btn-danger {  
    background-color: #dc3545;  
    border: none;  
}  
  
.btn-danger:hover {  
    background-color: #c82333;  
}  
  
form {  
    background-color: #fff;  
    padding: 30px;  
    border-radius: 8px;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
}  
  
form .form-control {  
    border-radius: 5px;  
}  
  
form button {  
    width: 100%;  
    padding: 10px;  
    font-size: 16px;
```

```
}

form a {

margin-top: 10px;
display: inline-block;
text-align: center;
font-size: 16px;
}
```

Integrasi dengan HTML (Tambah link CSS pada halaman)

Pada setiap halaman, Anda cukup menambahkan referensi ke file style.css yang telah dibuat. Pastikan untuk menaruh file CSS di direktori yang sama dengan file PHP atau menyesuaikan path-nya. (html)

```
<head>

<meta charset="UTF-8">
<title>Data Produk</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
<link href="style.css" rel="stylesheet"> <!-- Link ke file CSS -->
</head>
```

Berikut tangkapan layar index.php :

Daftar Produk				
Tambah Produk Baru				
ID	Nama Produk	Harga	Stok	Aksi

Berikut link/tautan penjelasan vidio github: [Klik](#)