

# **LAPORAN PRAKTIKUM**

## **Pemrograman Web**



Disusun Oleh :

MUH. HANIF ALFAQIH

NIM : 202312003

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

## **KATA PENGANTAR**

Segala puji kepada Tuhan Yang Maha Esa, sehingga modul praktikum Pemrograman Web dapat diselesaikan. Modul ini dibuat sebagai petunjuk praktikum kepada mahasiswa yang megambil mata kuliah Pemrograman Web pada Program Studi Teknik Informatika STITEK Bontang.

Tim penyusun berterima kasih kepada semua pihak yang telah membantu dalam penyelesaian modul praktikum ini. Modul ini masih jauh dari kesempurnaan oleh karenanya dibutuhkan kritik dan saran yang membangun demi perbaikan modul praktikum ini. Semoga modul praktikum ini dapat berguna dan bermanfaat dalam proses pendidikan di Program Studi Teknik Informatika STITEK Bontang.

Bontang, 31 Mei 2025

Muh. Hanif Alfaqih

## DAFTAR ISI

KATA PENGANTAR .....	ii
DAFTAR ISI.....	iii
LEMBAR ASISTENSI .....	v
Modul 1. HTML Dasar .....	1
A.    Tujuan .....	1
B.    Alat dan Bahan.....	1
C.    Landasan Teori.....	1
D.    Kegiatan Praktikum .....	6
E.    Tugas.....	20
Modul 2. Pengenalan CSS ( Cascading Style Sheet ) .....	23
A.    Tujuan .....	23
B.    Alat dan Bahan.....	23
C.    Landasan Teori.....	23
D.    Kegiatan Praktikum .....	27
E.    Tugas.....	46
Modul 3. Bootstrap .....	48
A.    Tujuan .....	48
B.    Alat dan Bahan.....	48
C.    Landasan Teori.....	48
D.    Kegiatan Praktikum .....	53
E.    Tugas.....	71
Modul 4. Javascript .....	73
A.    Tujuan .....	73
B.    Alat dan Bahan.....	73

C.	Landasan Teori.....	73
D.	Kegiatan Praktikum .....	82
E.	Tugas.....	100
	Daftar Pustaka .....	104

# LEMBAR ASISTENSI



## SEKOLAH TINGGI TEKNOLOGI BONTANG LABORATORIUM TEKNIK INFORMATIKA

KAMPUS UTAMA | JL. Letjen S Parman No. 65 RT. 48 Kel. Belimbing, Kec. Bontang Barat, Kode Pos 75513  
t. 0548 - 28782 | 0811 530 2001 [admin@stitek.ac.id](mailto:admin@stitek.ac.id) <https://stitek.ac.id>

### KARTU KENDALI PRAKTIKUM

NAMA (nomer absensi) : \_\_\_\_\_  
NIM : \_\_\_\_\_  
MATA KULIAH PRAKTIKUM : \_\_\_\_\_  
RUANG LAB : \_\_\_\_\_  
DOSEN PRAKTIKUM : \_\_\_\_\_  
ASISTEN PRAKTIKUM : \_\_\_\_\_

Perte muhan Ke-	Hari/ Tanggal	Modul	Materi Praktikum	Percobaan		Paraf	
				Total	Tercapai	Praktikan	Asisten
1.							
2.							
3.							
4.							
5.							
6.							
7.							
8.							
9.							

Ketua Program Studi

Bontang,  
Dosen Pengampu

Abadi Nugroho, M.Kom.  
NIDN. 1104129002

\_\_\_\_\_

# **Modul 1. HTML Dasar**

## **A. Tujuan**

Setelah mengikuti praktikum, Mahasiswa diharapkan dapat untuk:

1. Mampu memahami struktur dokumen HTML
2. Mampu membuat web sederhana dengan menggunakan HTML

## **B. Alat dan Bahan**

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

1. Modul Pemrograman Web
2. Laptop
3. Aplikasi Visual Studio Code

## **C. Landasan Teori**

Apa itu HTML?

HTML adalah singkatan dari HyperText Markup Language, yakni sebuah bahasa markup yang dipakai untuk menyusun dan menampilkan konten di halaman web/aplikasi. Bahasa ini pertama kali dikembangkan oleh Tim Berners-Lee pada tahun 1991 dalam proyek “World Wide Web” atau “WWW”. Di versi pertamanya, HTML hanya mendukung 18 tag dasar untuk menautkan dokumen dan memformat teks. Namun HTML terus dikembangkan dengan berbagai pembaruan seperti HTML4 dan HTML5. Kini, HTML menjadi standar pengembangan web yang mendukung sekitar 140 tag untuk membuat elemen baru.

Perlu diingat, HTML hanya berperan dalam pembuatan struktur dasar seperti tabel, teks, paragraf, link, dan sebagainya. Ini berbeda dengan bahasa pemrograman seperti Python atau JavaScript yang mampu menjalankan fungsi dan logika pemrograman. Oleh karena itu, HTML tidak bisa disebut sebagai bahasa pemrograman.

Cara Kerja HTML

HTML pada dasarnya bekerja menggunakan serangkaian elemen atau tag untuk membangun struktur dan semantik pada konten. File-nya memiliki ekstensi

.html atau .htm yang bisa diakses melalui web browser seperti Chrome, Mozilla Firefox, Safari, dan banyak lagi. Lebih jelasnya, berikut cara kerja HTML:

1. Pembuatan file HTML: File HTML umumnya memiliki ekstensi .html atau .htm dan dibuat di software text editor seperti Visual Studio Code. Dalam file ini, kamu akan menulis kode HTML yang terdiri dari elemen-elemen terstruktur, contohnya:

```
<!DOCTYPE html>
<html>

<head>
    <title>Judul Halaman</title>
</head>

<body>
    <h1>Halo, Selamat Datang!</h1>
    <p>Ini adalah paragraf pertama dalam halaman web.</p>
</body>
</html>
```

Berikut penjelasan mengenai tag pada kode di atas:

- <html> adalah elemen menandakan awal dan akhir dokumen HTML.
- <head> berisi informasi meta.
- <title> merupakan judul halaman yang masuk ke dalam tag <head>.
- <body> memuat konten yang akan ditampilkan di web browser. Di sini terdapat beberapa elemen seperti <h1> untuk judul utama dan <p> untuk paragraf (text).

Perlu diingat, elemen HTML terdiri dari tag pembuka dan tag penutup untuk menandai bagian konten tertentu, misalnya <h1>Belajar HTML Dasar</h1>. Ini merupakan contoh tag “heading 1” yang menandai sebuah judul utama.

2. Hierarki elemen: Elemen-elemen HTML disusun secara hierarkis, membentuk bagian-bagian seperti judul, paragraf, daftar, tautan, gambar, dan blok konten lainnya. Susunan hierarki ini membantu browser memahami struktur dan urutan tampilan konten.

3. HTML dibaca oleh web browser: Ketika membuka suatu halaman, browser seperti Chrome, Firefox, atau Edge akan membaca kode HTML lalu menampilkannya kepada pengguna sesuai dengan elemen-elemen HTML yang ditulis. Prosesnya kurang lebih seperti berikut:
  - Request – Browser mengirimkan permintaan ke server tempat file HTML disimpan.
  - Response – Server mengirimkan file HTML ke browser pengguna.
  - Render – Browser akan membaca dan menerjemahkan kode HTML untuk menampilkan halaman sesuai dengan struktur yang ada dalam file HTML.

### Fungsi HTML

HTML berperan penting dalam membangun dan menampilkan halaman web di internet. Bahasa markup ini memungkinkan website untuk memiliki struktur yang jelas, mulai dari judul, teks, gambar, hingga elemen interaktif seperti tautan dan formulir. Berikut beberapa fungsi HTML dalam pengembangan web:

#### 1. Membuat website

Seperti yang telah dijelaskan bahwa HTML menjadi pondasi utama dalam pembuatan website. Sebab, hampir semua halaman web yang ada di internet terdiri dari elemen-elemen HTML yang terstruktur. Tanpa HTML, halaman website tidak akan memiliki bentuk serta tidak bisa diakses oleh pengguna. Selain itu, dalam pembuatan website, HTML juga biasanya dipadukan dengan CSS dan JavaScript untuk memberikan visual yang lebih interaktif.

#### 2. Menampilkan teks, gambar, dan multimedia

Dengan HTML, kamu dapat menyisipkan dan menampilkan berbagai jenis konten seperti teks, gambar, atau multimedia lainnya ke dalam halaman. Misalnya, elemen `<p>` untuk menyisipkan paragraf, `<img>` untuk gambar, dan `<video>` untuk menampilkan video.

#### 3. Hyperlink

Salah satu kemampuan penting yang dimiliki oleh HTML adalah hyperlink atau membuat tautan. Ini memungkinkan pemilik situs untuk mengarahkan pengguna ke halaman atau website lain melalui tautan. Tag yang digunakan untuk menjalankan fungsi ini adalah `<a>`.

#### 4. Membantu pembuatan website yang responsif

Sebenarnya, HTML tidak bertanggung jawab untuk membuat desain responsif. Tetapi, dengan elemen `<meta viewport>`, website masih dapat menyesuaikan tampilannya saat diakses pada berbagai perangkat. Selain itu, HTML juga bisa dikombinasikan dengan CSS dan JavaScript agar tampilannya tetap optimal di berbagai ukuran layar.

#### 5. Penanda teks dan bagian website

HTML memungkinkan website untuk menandai teks dan bagian-bagian tertentu menggunakan tag yang sesuai. Misalnya seperti `<b>` untuk teks cetak tebal, `<i>` untuk teks miring, atau `<header>`, `<footer>`, dan `<section>` untuk bagian konten. Selain membantu pengorganisasian konten, hal ini juga dapat meningkatkan SEO (Search Engine Optimization) jika dilakukan dengan benar.

### Struktur HTML

Untuk mengatur tampilan dan fungsionalitas halaman maka diperlukan HTML dengan struktur yang benar. Ini sangat penting karena berpengaruh terhadap kemampuan web browser dalam memahami dan menampilkan konten. Setidaknya, ada tiga bagian utama dalam struktur HTML, yakni tag, elemen, dan atribut. Berikut penjelasannya:

#### 1. Tag

Tag adalah penanda awal dan akhir dari sebuah elemen. Tag sendiri terdiri dari tanda kurung siku (`<>`) yang mengapit sebuah kode tertentu yang berisi perintah atau instruksi. Ada pun tag ini memiliki dua jenis yang berbeda, di antaranya:

- Tag dengan penutup, seperti `<p>...</p>` untuk paragraf, `<body>...</body>` untuk tubuh halaman, dan `<title>...</title>` untuk judul.
- Tag tanpa penutup, seperti `<br>` untuk baris baru, `<img>` untuk menyisipkan gambar, dan `<hr>` untuk membuat garis horizontal.

Contoh tag:

```
<h1>Judul Utama</h1>, <a href="https://example.com">Tautan</a>
```

## 2. Element

Element merupakan bagian yang lebih luas dari tag dan terdiri dari “tag pembuka, isi konten, serta tag penutup”. Contohnya seperti berikut:

```
<p>Ini adalah sebuah paragraf.</p>  
<a href="https://example.com">Tautan ke Website</a>
```

Selain itu, ada juga elemen yang bersarang (nested), seperti “`<div><p>Paragraf</p></div>`”. Contoh ini menunjukkan elemen `<p>` yang berada di dalam `<div>`.

## 3. Attribute

Atribut adalah informasi tambahan yang diberikan pada sebuah tag atau elemen. Tidak semua tag memiliki atribut, namun tag tertentu memerlukan atribut agar dapat berfungsi, seperti pada `<a>`, `<img>`, dan `<video>`.

Contoh atribut:

```
<a href="https://example.com">Tautan Website</a> (atribut href memberikan alamat tautan)  
 (atribut src memberikan sumber gambar, dan alt memberikan deskripsi teks alternatif)
```

## Sudah Tahu Apa itu HTML?

HTML adalah bahasa markup standar dalam pengembangan website yang digunakan untuk menyusun struktur halaman. Bahasa markup ini memungkinkan konten untuk dapat ditampilkan dengan benar di web browser. Namun perlu diingat, HTML berbeda dengan bahasa pemrograman yang memiliki logika pemrograman, HTML hanya bertugas untuk mengatur tata letak dan elemen dalam dokumen web. [1]

## D. Kegiatan Praktikum

Kegiatan praktikum Modul HTML Dasar pada mata kuliah Pemrograman Dasar dilaksanakan pada Kamis, 31 Mei 2025 dan jam 17.00-18.20 WITA di Laboratorium Multimedia STITEK BONTANG, menggunakan Komputer no 5. Adapun Langkah-langkah pengerjaannya sebagai berikut:

### 1. Pengenalan HTML dan Struktur Dasar Dokumen

#### Input



```
strukturdasar.html x elemen.html atribut.html format.html linkgambar.html
strukturdasar.html > html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Halaman Pertama Saya</title>
5   </head>
6   <body>
7     <h1>Halo, Saya Muh. Hanif Alfaqih!</h1>
8     <p>Ini adalah paragraf pertama saya.</p>
9   </body>
10 </html>
```

#### Output



#### Analisis

##### a. <!DOCTYPE html>

Baris ini memberitahu browser bahwa dokumen ini menggunakan standar HTML versi 5.

##### b. <html>

Ini adalah tag pembuka untuk seluruh dokumen HTML. Semua isi halaman web harus ada di dalam tag ini.

##### c. <head>

Tag ini berfungsi sebagai tempat untuk menyimpan informasi tentang halaman yang tidak langsung terlihat, seperti judul halaman, pengaturan karakter, dan link ke file CSS atau JavaScript.

##### d. <title>Halaman Pertama Saya</title>

Ini adalah judul halaman yang muncul di tab browser. Di sini tertulis "Halaman Pertama Saya" sehingga ketika membuka tab browser, judul ini akan terlihat.

e. </head>

Menandai akhir bagian kepala (head) dari dokumen HTML.

f. <body>

Tag pembuka untuk isi utama halaman yang akan dilihat pengunjung, seperti teks, gambar, dan elemen-elemen lainnya.

g. <h1>Halo, Saya Muh. Hanif Alfaqih!</h1>

Ini adalah teks yang sangat besar dan tebal (heading utama). Berfungsi sebagai judul utama yang terlihat jelas di halaman, menyapa pengunjung dengan kalimat “Halo, Saya Muh. Hanif Alfaqih!”.

h. <p>Ini adalah paragraf pertama saya.</p>

Sebuah paragraf teks biasa. Memberikan informasi tambahan di bawah judul.

i. </body>

Menutup bagian isi utama halaman.

j. </html>

Menandai akhir seluruh dokumen HTML.

## 2. Elemen HTML Dasar

### Input

```
struktdasar.html  elemen.html X  atribut.html  format.html  linkgambar.html  daftar.html  tabel.html
elemen.html > html > body > img
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Elemen HTML Dasar</title>
5  </head>
6  <body>
7  |   <h2>Heading level 2</h2>
8  |   <p>Ini adalah paragraf dengan link <a href="http://www.stitek.ac.id">website stitek</a></p>
9  |   
10 |   <ul>
11 |       <li>Item pertama</li>
12 |       <li>Item kedua</li>
13 |   </ul>
14 </body>
15 </html>
```

### Output



## Analisis

a. <!DOCTYPE html>

Baris ini menyatakan bahwa dokumen ini menggunakan HTML5

b. <html>

Ini adalah pembuka dokumen HTML. Semua elemen halaman harus ditulis di dalam tag ini.

c. <head>

<title>Elemen HTML Dasar</title>

</head>

Bagian <head> berisi informasi tentang halaman:

<title>: Judul halaman yang akan muncul di tab browser, yaitu "Elemen HTML Dasar".

d. <body>

Ini adalah bagian utama halaman yang terlihat oleh pengunjung saat mereka membuka halaman web.

e. <h2>Heading level 2</h2>

Ini adalah judul dengan tingkat heading ke-2. Artinya ukurannya sedikit lebih kecil dari <h1>, tapi masih cukup menonjol. Cocok untuk subjudul.

f. <p>Ini adalah paragraf dengan link <a href="http://www.stitek.ac.id">website stitek</a></p>

Ini adalah paragraf teks biasa. Di dalam paragraf, ada tautan/link (<a href=...>) ke situs STITEK. Saat diklik, pengguna akan dibawa ke "http://www.stitek.ac.id".

g. 

Ini adalah gambar yang ditampilkan di halaman:

src="stitek.png" menunjukkan lokasi file gambar (harus berada di folder yang sama dengan file HTML).

alt="Deskripsi Gambar" adalah teks alternatif yang muncul jika gambar tidak bisa ditampilkan. Juga berguna untuk pembaca layar (aksesibilitas).

h. <ul>

```
<li>Item pertama</li>
<li>Item kedua</li>
</ul>
```

Ini adalah daftar tak berurutan (unordered list) dengan dua item:

<ul> membuka daftar.

<li> adalah item-item dalam daftar: "Item pertama" dan "Item kedua".

Daftar ini akan tampil sebagai bullet points.

i. </body>

Menutup bagian isi halaman.

j. </html>

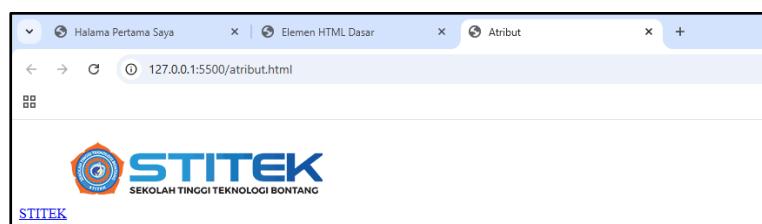
Menutup dokumen HTML sepenuhnya.

### 3. Atribut HTML

#### Input

```
strukturdasar.html | elemen.html | atribut.html | format.html | linkgambar.html
attrib.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Atribut</title>
5  </head>
6  <body>
7  |   <a href="http://www.stitek.ac.id" title="pergi ke stitek">STITEK</a>
8  |   
9  </body>
10 </html>
```

#### Output



#### Analisis

a. <!DOCTYPE html>

Menandakan bahwa ini adalah dokumen HTML5.

b. <html>

Tag pembuka untuk seluruh halaman HTML. Semua isi halaman harus berada di dalamnya.

c. <head>

<title>Atribut</title>

```
</head>
```

Bagian kepala (head) dari dokumen:

```
<title> menetapkan judul tab browser yang akan tampil sebagai "Atribut".
```

d. <body>

Ini adalah bagian isi utama halaman web yang akan terlihat oleh pengguna.

e. <a href="http://www.stitek.ac.id" title="persi ke stitek">STITEK</a>

Ini adalah tautan (link).

```
href="http://www.stitek.ac.id": Saat diklik, pengguna akan diarahkan ke situs STITEK.
```

```
title="persi ke stitek": Saat mouse diarahkan ke link, akan muncul tooltip bertuliskan "pergi ke stitek".
```

"STITEK" adalah teks yang terlihat oleh pengguna dan dapat diklik.

f. 

Ini adalah gambar.

```
src="stitek.png": Menunjukkan file gambar yang ingin ditampilkan (harus ada di folder yang sama).
```

```
title="Logo": Tooltip yang akan muncul saat mouse diarahkan ke gambar.
```

```
alt="Logo Kampus" akan muncul jika gambar gagal dimuat, dan juga membantu pembaca layar (aksesibilitas).
```

g. </body>

```
</html>
```

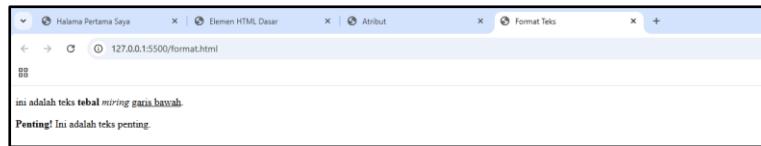
Menutup bagian isi dan dokumen HTML.

#### 4. Format Teks

Input

```
④ format.html > ⌘html
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Format Teks</title>
5      </head>
6      <body>
7          <p>ini adalah teks <b>tebal</b> <i>miring</i> <u>garis bawah</u>.</p>
8          <p><strong>Penting!</strong> Ini adalah teks penting.</p>
9      </body>
10 </html>
```

## Output



## Analisis

a. `<!DOCTYPE html>`

Menyatakan bahwa dokumen ini menggunakan HTML5.

b. `<html>`

Tag pembuka dari dokumen HTML. Semua isi halaman harus berada di dalamnya.

c. `<head>`

```
<title>Format Teks</title>
</head>
```

Bagian `<head>` menyimpan informasi tentang halaman:

`<title>` akan menjadi judul tab di browser, yaitu "Format Teks".

d. `<body>`

Ini bagian yang ditampilkan di halaman web. Segala sesuatu yang dilihat oleh pengguna akan ditulis di dalam `<body>`.

e. `<p>ini adalah teks <b>tebal</b> <i>miring</i> <u>garis bawah</u>.</p>`

Paragraf ini menampilkan teks dengan format visual:

`<b>` = tebal (bold), hanya untuk tampilan.

`<i>` = miring (italic), juga hanya tampilan.

`<u>` = garis bawah (underline).

f. `<p><strong>Penting!</strong> Ini adalah teks penting.</p>`

Paragraf kedua menggunakan:

`<strong>` = tebal dengan makna penting. Ini semantik, artinya mesin pencari dan pembaca layar tahu bahwa teks ini harus dianggap penting.

g. `</body>`

`</html>`

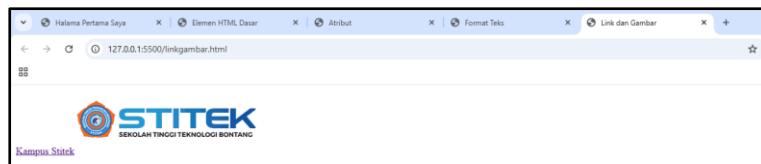
Menutup bagian isi (`</body>`) dan seluruh dokumen HTML (`</html>`).

## 5. Link dan Gambar

### Input

```
linkgambar.html > html
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Link dan Gambar</title>
5      </head>
6      <body>
7          <a href="https://www.stitek.ac.id">Kampus Stitek</a>
8          
9      </body>
10 </html>
```

### Output



### Analisis

a. `<!DOCTYPE html>`

Baris ini memberi tahu browser bahwa dokumen ini menggunakan HTML5.

b. `<html>`

Ini adalah tag pembuka untuk dokumen HTML. Semua isi halaman web harus berada di dalam tag ini.

c. `<head>`

```
<title>Link dan Gambar</title>
</head>
```

Bagian `<head>` berisi informasi tentang halaman, tidak terlihat langsung oleh pengguna.

`<title>` digunakan untuk menentukan judul halaman yang muncul di tab browser, yaitu "Link dan Gambar".

d. `<body>`

Ini adalah bagian utama halaman web yang akan ditampilkan kepada pengguna.

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

Ini adalah tautan/link:

`href="https://www.stitek.ac.id"`: Saat diklik, pengguna akan diarahkan ke situs STITEK.

"Kampus Stitek" adalah teks yang tampil dan bisa diklik.

f. 

Ini adalah gambar:

src="stitek.png" adalah lokasi file gambar. Pastikan file ini ada di folder yang sama dengan file HTML.

alt="logo" adalah teks alternatif yang muncul jika gambar gagal dimuat.

Ini juga membantu aksesibilitas (misalnya, pembaca layar untuk pengguna tunanetra).

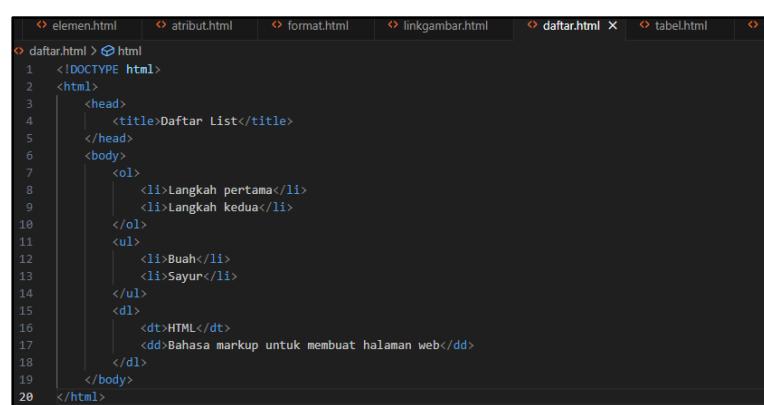
g. </body>

</html>

Menutup bagian isi halaman (</body>) dan seluruh dokumen HTML (</html>).

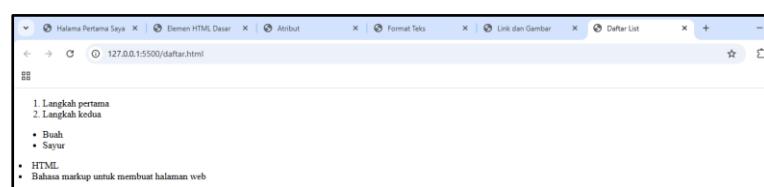
## 6. Daftar (List)

Input



```
daftar.html > html
1  <!DOCTYPE html>
2  <html>
3  |   <head>
4  |   |       <title>Daftar List</title>
5  |   </head>
6  |   <body>
7  |   |       <ol>
8  |   |       |           <li>Langkah pertama</li>
9  |   |       |           <li>Langkah kedua</li>
10 |   |       </ol>
11 |   |       <ul>
12 |   |       |           <li>Buah</li>
13 |   |       |           <li>Sayur</li>
14 |   |       </ul>
15 |   |       <dl>
16 |   |       |           <dt>HTML</dt>
17 |   |       |           <dd>Bahasa markup untuk membuat halaman web</dd>
18 |   |       </dl>
19 |   </body>
20 </html>
```

Output



Analisis

a. <!DOCTYPE html>

Menyatakan bahwa dokumen ini adalah dokumen HTML5.

b. <html>

Tag pembuka untuk dokumen HTML, menandai awal seluruh konten HTML.

- c. <head>
- ```
<title>Daftar List</title>
</head>
```
- Bagian <head> berisi informasi metadata halaman web.  
<title> menentukan judul halaman yang akan muncul pada tab browser, yaitu "Daftar List".
- d. <body>
- Tag pembuka bagian isi utama halaman yang akan ditampilkan ke pengguna.
- e. <ol>
- ```
<li>Langkah pertama</li>
<li>Langkah kedua</li>
</ol>
```
- Membuat daftar bernomor (ordered list).  
<li> adalah setiap item daftar, di sini berupa "Langkah pertama" dan "Langkah kedua".
- Akan tampil sebagai:
- Langkah pertama  
Langkah kedua
- f. <ul>
- ```
<li>Buah</li>
<li>Sayur</li>
</ul>
```
- Membuat daftar berpoin (unordered list).  
Item daftar berupa "Buah" dan "Sayur".
- Akan tampil dengan bullet points.
- g. <dl>
- ```
<dt>HTML</dt>
<dd>Bahasa markup untuk membuat halaman web</dd>
</dl>
```
- Membuat daftar definisi (definition list).
- <dt> adalah istilah yang didefinisikan, yaitu "HTML".

<dd> adalah deskripsi atau penjelasan dari istilah tersebut.

Akan tampil dengan istilah diikuti oleh penjelasan yang menjorok ke kanan.

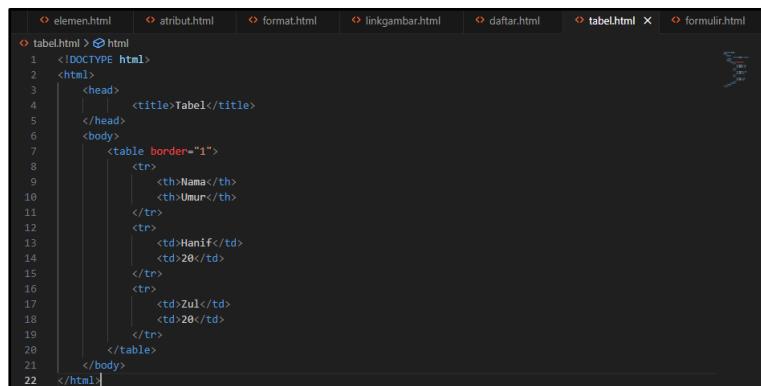
h. </body>

</html>

Menutup bagian isi halaman dan menutup seluruh dokumen HTML.

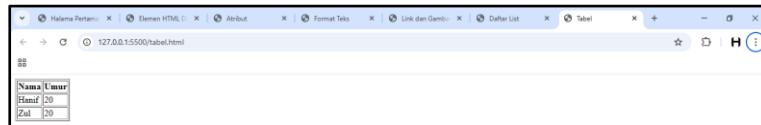
## 7. Tabel HTML

Input



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Tabel</title>
5   </head>
6   <body>
7     <table border="1">
8       <tr>
9         <th>Nama</th>
10        <th>Umur</th>
11      </tr>
12      <tr>
13        <td>Hanif</td>
14        <td>20</td>
15      </tr>
16      <tr>
17        <td>Zul</td>
18        <td>20</td>
19      </tr>
20    </table>
21  </body>
22 </html>
```

Output



Analisis

a. <!DOCTYPE html>

Menyatakan dokumen ini menggunakan HTML5.

b. <html>

Tag pembuka untuk keseluruhan dokumen HTML.

c. <head>

<title>Tabel</title>

</head>

Bagian kepala dokumen.

<title> menentukan judul halaman “Tabel” yang muncul di tab browser.

d. <body>

Awal bagian isi halaman yang terlihat pengguna.

e. <table border="1">

Membuat tabel dengan garis border tebal 1 piksel di sekeliling dan antar sel.

f. <tr>  
    <th>Nama</th>  
    <th>Umur</th>  
</tr>

Membuat baris pertama tabel (<tr>) sebagai header.

<th> adalah sel header yang biasanya teksnya dicetak tebal dan tengah.  
Kolom pertama berjudul “Nama”, kolom kedua “Umur”.

g. <tr>  
    <td>Hanif</td>  
    <td>20</td>  
</tr>

Baris kedua tabel.

<td> adalah sel data biasa.

Nama “Hanif” dan umur “20” diisikan dalam masing-masing kolom.

h. <tr>  
    <td>Zul</td>  
    <td>20</td>  
</tr>

Baris ketiga tabel.

Data “Zul” dan “20” pada kolom nama dan umur.

i. </table>  
   </body>  
   </html>

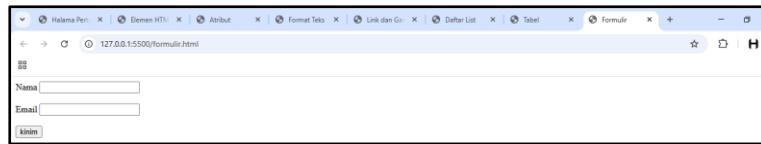
Menutup tabel, body, dan dokumen HTML.

## 8. Formulir HTML

### Input

```
 1 <!DOCTYPE html>
 2 <html>
 3   <head>
 4     <title>Formulir</title>
 5   </head>
 6   <body>
 7     <form action="submit" method="post">
 8       <label for="nama">Nama</label>
 9       <input type="text" id="nama" name="nama"><br><br>
10
11       <label for="email">Email</label>
12       <input type="email" id="email" name="email"><br><br>
13
14       <input type="submit" value="kinim">
15     </form>
16   </body>
17 </html>
```

### Output



### Analisis

a. `<!DOCTYPE html>`

Menyatakan bahwa dokumen ini menggunakan standar HTML5.

b. `<html>`

Tag pembuka untuk dokumen HTML secara keseluruhan.

c. `<head>`

```
<title>Formulir</title>
```

```
</head>
```

Bagian `<head>` berisi metadata dan pengaturan halaman.

`<title>` menentukan judul halaman yang muncul di tab browser, yaitu "Formulir".

d. `<body>`

Tag pembuka bagian isi utama halaman, tempat semua konten terlihat pengunjung.

e. `<form action="submit" method="post">`

Membuka formulir HTML.

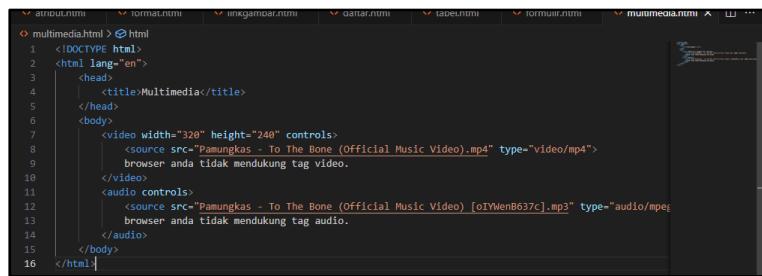
`action="submit"` menunjukkan URL atau alamat tujuan pengiriman data (di sini hanya contoh, biasanya diisi alamat server/skrip).

`method="post"` artinya data akan dikirim menggunakan metode POST, yang tidak terlihat di URL dan lebih aman untuk data sensitif.

- f. <label for="nama">Nama</label>  
<input type="text" id="nama" name="nama"></br></br>  
<label> membuat teks "Nama" yang terhubung dengan input berikutnya  
(for="nama" menghubungkan label ke input dengan id="nama").  
<input type="text"> membuat kotak isian teks biasa.  
id="nama" memberi identitas unik pada input agar bisa terhubung  
dengan label.  
name="nama" adalah nama data yang dikirimkan ke server saat submit.  
<br></br> (seharusnya cukup <br> saja) memberi baris baru/ spasi agar  
form tidak rapat.
- g. <label for="email">Email</label>  
<input type="email" id="email" name="email"></br></br>  
Label dan input untuk alamat email.  
<input type="email"> memudahkan validasi browser memastikan  
format email benar.  
Atribut id dan name sama fungsi seperti sebelumnya.  
<br></br> untuk memberi jarak antar elemen.
- h. <input type="submit" value="kinim">  
Tombol kirim formulir.  
type="submit" membuat tombol yang saat ditekan akan mengirim data  
formulir.  
value="kinim" adalah teks yang muncul di tombol (mungkin maksudnya  
"kirim"?).
- i. </form>  
</body>  
</html>
- Menutup tag formulir, body, dan dokumen HTML.

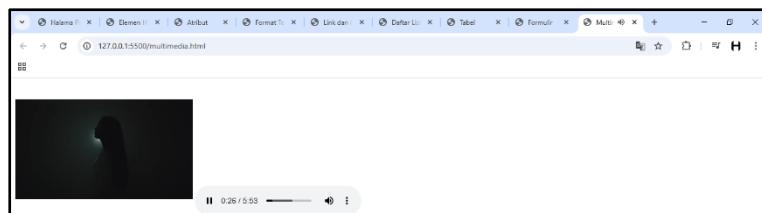
## 9. Multimedia

### Input



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Multimedia</title>
</head>
<body>
    <video width="320" height="240" controls>
        <source src="Pamungkas - To The Bone (Official Music Video).mp4" type="video/mp4">
        browser anda tidak mendukung tag video.
    </video>
    <audio controls>
        <source src="Pamungkas - To The Bone (Official Music Video) [ofYWeB637C].mp3" type="audio/mpeg">
        browser anda tidak mendukung tag audio.
    </audio>
</body>
</html>
```

### Output



### Analisis

a. <!DOCTYPE html>

Menandakan bahwa dokumen ini ditulis menggunakan HTML5.

b. <html lang="en">

Tag pembuka dokumen HTML.

lang="en" menunjukkan bahwa bahasa utama konten ini adalah Bahasa Inggris.

c. <head>

<title>Multimedia</title>

</head>

Bagian <head> berisi informasi metadata halaman.

<title> menentukan judul yang akan muncul di tab browser: "Multimedia".

d. <body>

Awal dari bagian isi halaman yang akan ditampilkan ke pengguna.

e. <video width="320" height="240" controls>

<source src="Pamungkas - To The Bone (Official Music Video).mp4" type="video/mp4"> browser anda tidak mendukung tag video.

</video>

Membuat pemutar video di halaman.

width dan height mengatur ukuran tampilan video: 320 piksel lebar, 240 piksel tinggi.

controls membuat kontrol video (play, pause, volume) muncul.

<source> menunjukkan file video yang akan diputar, dalam format mp4.

Jika browser tidak mendukung <video>, teks fallback “browser anda tidak mendukung tag video.” akan tampil.

f. <audio controls>

<source src="Pamungkas - To The Bone (Official Music Video) [oIYWenB637c].mp3" type="audio/mpeg">browser anda tidak mendukung tag audio.

</audio>

Membuat pemutar audio di halaman.

controls membuat tombol play, pause, volume, dll.

<source> menunjuk ke file MP3 yang akan diputar.

Jika browser tidak mendukung <audio>, akan muncul pesan fallback.

g. </body>

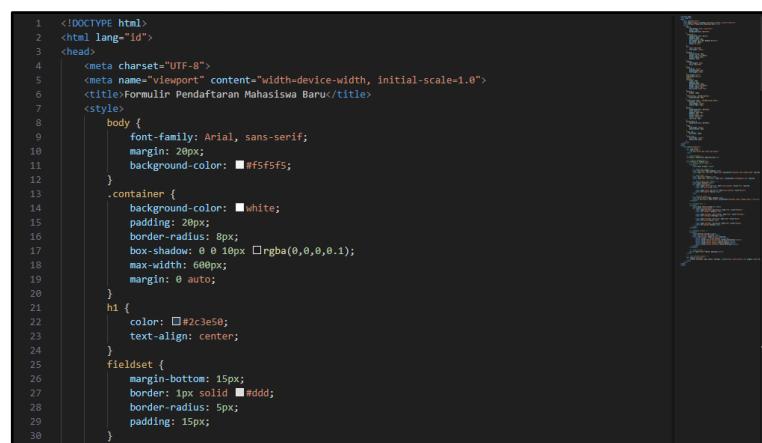
</html>

Menutup tag <body> dan <html>, menandakan akhir dokumen HTML.

## E. Tugas

- Buatlah sebuah halaman web HTML yang berisi **Formulir Pendaftaran Mahasiswa Baru**

Input



```
1  <!DOCTYPE html>
2  <html lang="id">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Formulir Pendaftaran Mahasiswa Baru</title>
7      <style>
8          body {
9              font-family: Arial, sans-serif;
10             margin: 20px;
11             background-color: #f5f5f5;
12         }
13         .container {
14             background-color: #fff;
15             padding: 20px;
16             border-radius: 8px;
17             box-shadow: 0 0 10px #rgba(0,0,0,0.1);
18             max-width: 600px;
19             margin: 0 auto;
20         }
21         h1 {
22             color: #2c3e50;
23             text-align: center;
24         }
25         fieldset {
26             margin-bottom: 15px;
27             border: 1px solid #ddd;
28             border-radius: 5px;
29             padding: 15px;
30         }
31     </style>
32 
```

```

31     legend {
32         font-weight: bold;
33         color: #2c3e50;
34     }
35     label {
36         display: block;
37         margin-bottom: 5px;
38         font-weight: bold;
39     }
40     input[type="text"],
41     input[type="email"],
42     select,
43     textarea {
44         width: 100%;
45         padding: 8px;
46         margin-bottom: 10px;
47         border: 1px solid #ddd;
48         border-radius: 4px;
49         box-sizing: border-box;
50     }
51     textarea {
52         height: 100px;
53     }
54     .radio-group, .checkbox-group {
55         margin-bottom: 10px;
56     }

```

```

57     .radio-group label, .checkbox-group label {
58         display: inline;
59         font-weight: normal;
60         margin-right: 15px;
61     }
62     button {
63         background-color: #3498db;
64         color: white;
65         padding: 10px 15px;
66         border: none;
67         border-radius: 4px;
68         cursor: pointer;
69         font-size: 16px;
70     }
71     button:hover {
72         background-color: #2980b9;
73     }
74     .logo {
75         text-align: center;
76         margin-bottom: 20px;
77     }
78     .logo img {
79         max-width: 150px;
80     }
81     .info-link {
82         text-align: center;
83         margin-top: 20px;
84     }
85 
```

```

86 </head>
87 <body>
88     <div class="container">
89         <!-- Logo Kampus -->
90         <div class="logo">
91             
92         </div>
93
94         <!-- Judul Halaman -->
95         <h1>Formulir Pendaftaran Mahasiswa Baru</h1>
96
97         <!-- Formulir Pendaftaran -->
98         <form action="#" method="post">
99             <!-- Kelompok Data Pribadi -->
100            <fieldset>
101                <legend>Data Pribadi</legend>
102
103                <!-- Input Nama Lengkap -->
104                <label for="nama">Nama Lengkap*</label>
105                <input type="text" id="nama" name="nama" placeholder="Masukkan nama lengkap Anda" required>
106
107                <!-- Input Email -->
108                <label for="email">Email*</label>
109                <input type="email" id="email" name="email" placeholder="contoh@email.com" required>
110

```

```

111
112
113
114
115
116
117
118
119
120
121
122
123
124

```

```

126    <!-- Kelompok Hobi -->
127    <fieldset>
128        <legend>Hobi (Pilih minimal 2)</legend>
129        <div class="checkbox-group">
130            <input type="checkbox" id="membaca" name="hobi" value="Membaca">
131            <label for="membaca">Membaca</label>
132
133            <input type="checkbox" id="olahraga" name="hobi" value="Olahraga">
134            <label for="olahraga">Olahraga</label>
135
136            <input type="checkbox" id="musik" name="hobi" value="Musik">
137            <label for="musik">Musik</label>
138
139            <input type="checkbox" id="gaming" name="hobi" value="Gaming">
140            <label for="gaming">Gaming</label>
141        </div>
142    </fieldset>
143
144    <!-- Kelompok Jurusan -->
145    <fieldset>
146        <legend>Pilihan Jurusan</legend>
147        <label for="jurusan">Jurusan</label>
148        <select id="jurusan" name="jurusan" required>
149            <option value="">-- Pilih Jurusan --</option>
150            <option value="Teknik Informatika">Teknik Informatika</option>
151            <option value="Teknik Elektro">Teknik Elektro</option>
152            <option value="Bisnis Digital">Bisnis Digital</option>
153            <option value="Sistem Informasi">Sistem Informasi</option>
154        </select>
155    </fieldset>
156
157    <!-- Tombol Submit -->
158    <button type="submit">Daftar Sekarang</button>
159    </form>
160
161    <!-- Link Informasi Kampus -->
162    <div class="info-link">
163        <p>Untuk informasi lebih lanjut, kunjungi <a href="https://www.stitek.ac.id" target="_blank">we
164    </div>
165    </body>
166    </html>
167

```

## Output

The screenshot shows a registration form titled "Formulir Pendaftaran Mahasiswa Baru". The form fields include:

- Data Pribadi** (Personal Data):
  - Nama Lengkap\***: Input field for full name.
  - Email\***: Input field for email address, containing "contoh@email.com".
  - Jenis Kelamin\***: Radio buttons for "Pria" and "Wanita".
  - Alamat Lengkap**: Input field for address.
- Hobi (Pilih minimal 2)**: Checkboxes for "Membaca", "Olahraga", "Musik", and "Traveling".
- Pilihan Jurusan** (Major Selection):
  - Jurusan\***: A dropdown menu with the placeholder "Pilih Jurusan".
- Bantuan** (Help):
  - Daftar Sekarang** (Register Now): A blue button.
  - Untuk informasi lebih lanjut, kunjungi website kampus kami.** (For more information, visit our university website.)

## 2. Link Github

<https://github.com/MuhHanifAlfaqih/Modul1>

## **Modul 2. Pengenalan CSS ( Cascading Style Sheet )**

### **A. Tujuan**

Setelah mengikuti praktikum, Mahasiswa diharapkan dapat untuk:

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

### **B. Alat dan Bahan**

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

1. Modul Pemrograman Web
2. Laptop
3. Aplikasi Visual Studio Code

### **C. Landasan Teori**

#### **1. Apa Itu CSS?**

CSS adalah singkatan dari Cascading Style Sheets, yaitu bahasa kode yang digunakan untuk menambahkan gaya pada elemen yang ditulis dalam bahasa markup seperti HTML. CSS berfungsi untuk membedakan desain visual website dari tampilan konten. HTML mengatur sususnan konten website, sedangkan CSS mengatur desain visual dan estetikanya.

#### **2. Kelebihan CSS**

Mudah bagi Anda untuk mengetahui mana website yang menggunakan CSS dan mana yang tidak.

Anda pasti pernah membuka website, tapi sesaat kemudian gagal loading dan malah menampilkan background putih yang didominasi teks hitam dan biru. Situasi ini menggambarkan bahwa CSS pada situs tersebut tidak dapat di-load dengan benar atau situs sama sekali tidak memiliki CSS. Umumnya, situs seperti ini hanya menerapkan HTML.

Sebelum menggunakan CSS, semua *styling* harus disertakan ke dalam markup HTML. Itu berati Anda harus mendeskripsikan semua background, warna font, *alignment*, dan lain-lain secara terpisah.

Dengan CSS, Anda dapat mengatur tampilan semua aspek pada file yang berbeda, lalu menentukan *style*, kemudian mengintegrasikan file CSS di atas markup HTML. Alhasil, markup HTML bisa lebih mudah *di-maintain*. Singkatnya, dengan CSS, Anda tidak perlu mendeskripsikan tampilan dari masing-masing elemen secara berulang-ulang. Anda tidak membuang-buang waktu, kode yang digunakan pun lebih singkat, dan error dapat diminimalisir.

Karena opsi kustomisasi yang ada hampir tak terbatas, CSS memungkinkan Anda untuk menerapkan berbagai macam *style* pada satu halaman HTML.

### 3. Cara Kerja CSS

CSS menggunakan bahasa Inggris sederhana berbasis syntax yang dilengkapi dengan sekumpulan *rule* yang mengaturnya. Seperti yang telah kami sebutkan sebelumnya, HTML tidak dibuat untuk menerapkan elemen *style*, hanya markup halaman saja. HTML dirancang semata-mata untuk mendeskripsikan konten. Sebagai contoh: <p>This is a paragraph.</p>.

Satu yang menjadi pertanyaannya sekarang, bagaimana caranya *style* paragraf? Struktur syntax CSS cukup sederhana. Struktur ini memiliki *selector* dan *declaration block*. Pilih elemen yang diinginkan, kemudian deklarasi (*declare*) yang harus Anda lakukan terhadap elemen tersebut. Sangat mudah, kan?

Ada berbagai *rule* yang harus diingat. Meskipun demikian, *rule* struktur cukup simpel dan sederhana.

*Selector* mengarah ke elemen HTML yang ingin Anda ubah tampilannya. *Declaration block* memuat satu atau lebih banyak deklarasi (*declaration*) yang dipisahkan dengan tanda titik koma.

Setiap deklarasi menyertakan nama dan value dari properti CSS, yang dipisahkan dengan tanda koma. Umumnya deklarasi CSS diakhiri dengan tanda titik koma, sedangkan declaration block dikelilingi oleh tanda kurung kurawal.

Mari kita lihat contoh di bawah ini:

Semua elemen <p> ditandai dengan warna biru dan hurufnya ditebalkan (*bold*).

```
p {  
    color: blue;  
    font-weight: bold;  
}
```

Contoh lain, semua elemen <p> diposisikan di tengah (center-aligned), dengan lebar 16x, dan berwarna pink.

```
p {  
    text-align: center;  
    font-size: 16px;  
    color: pink;  
}
```

Selanjutnya, mari kita bahas beberapa style CSS, yaitu Inline, External, dan Internal.

#### 4. CSS Style Internal, External, dan Inline

Kami akan membahas masing-masing CSS Style secara singkat. Untuk informasi atau pembahasan yang lebih mendalam, kami menyertakan link di bawah ikhtisar.

CSS Style **Internal** diload setiap kali website di-*refresh*, dan kekurangannya adalah waktu loading semakin lama. CSS style yang sama pun tidak dapat digunakan di halaman lain karena sudah aktif terlebih dulu di suatu halaman.

Namun di balik kekurangannya, CSS Style Internal memiliki beberapa kelebihan. Salah satunya adalah kemudahan dalam *sharing* template untuk pratinjau (*preview*) karena CSS hanya ada di satu halaman.

**External** merupakan CSS style yang paling mudah dan tidak menyulitkan. Semuanya dilakukan secara eksternal pada file **.css**. *Styling* dilakukan di file terpisah, lalu terapkan CSS ke halaman mana pun yang Anda inginkan. Sayangnya, CSS Style External juga memperlama waktu loading.

CSS Style **Inline** menggunakan elemen spesifik yang memuat tag <style>. Karena setiap komponen harus di-*stylize*, maka Inline bukan metode yang tepat jika Anda ingin menggunakan CSS dengan cepat.

Namun di sisi lain, hal tersebut mendatangkan keuntungan. Misalnya, jika Anda ingin mengubah satu elemen, atau menampilkan pratinjau dengan cepat, atau Anda tidak punya akses ke file CSS.

Untuk mengetahui lebih lanjut, silakan baca artikel kami tentang perbedaan CSS internal, external, dan inline.

## 5. Kesimpulan

Berikut beberapa poin yang dapat disimpulkan dari artikel ini:

- CSS dibuat untuk dapat bekerja bersama dengan bahasa markup seperti HTML. Biasanya CSS digunakan untuk mengatur tampilan halaman.
- Ada tiga style di CSS, dan untuk menerapkan CSS di banyak halaman pada waktu bersamaan, gunakan External style.
- Saat ini sudah banyak website yang menggunakan CSS. Hal ini dikarenakan CSS adalah salah satu bahasa markup yang diwajibkan ada. Semoga artikel ini dapat berguna bagi Anda. Jika punya pertanyaan tentang apa itu CSS, silakan tuliskan pada kolom komentar di bawah ini. [2]

## D. Kegiatan Praktikum

Kegiatan praktikum Modul Pengenalan CSS ( Cascading Style Sheet) pada mata kuliah Pemrograman Dasar dilaksanakan pada Kamis, 5 Juni 2025 dan jam 17.00-18.20 WITA di Laboratorium Multimedia STITEK BONTANG, menggunakan Komputer no 5.

Adapun Langkah-langkah pengerjaannya sebagai berikut:

### 1. Penulisan CSS (*Cascading Style Sheets*)

#### 1) Contoh

##### a. Input

```
contoh > index.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <style>
5          body {
6              background-color: #lightyellow;
7          }
8          h1 {
9              color: red;
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>
```

##### b. Output



#### 2) Latihan

##### a. Input

```
latihan > index.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Penulisan CSS</title>
5      <style>
6          body {
7              background-color: #lightyellow; /* Latar belakang kuning muda */
8          }
9          h1 {
10             color: red; /* Warna teks merah */
11             text-align: center; /* Rata tengah */
12         }
13     </style>
14 </head>
15 <body>
16     <h1>Latihan modul 2.1</h1>
17     <p>Teks ini berada di halaman dengan latar belakang kuning muda.</p>
18 </body>
19 </html>
```

##### b. Output



### c. Analisis

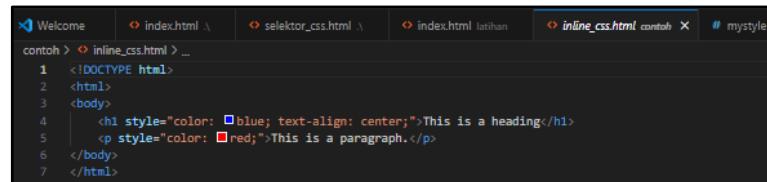
Perbedaan antara kedua kode HTML tersebut dapat dilihat dari beberapa aspek. Pertama, pada kode pertama, judul halaman (title) tidak didefinisikan, sedangkan pada kode kedua, judul halaman ditetapkan sebagai "Penulisan CSS". Kedua, konten yang ditampilkan di dalam body juga berbeda: kode pertama menampilkan pesan "Hai Abadi, selamat Datang di Praktikum CSS", sementara kode kedua menampilkan "Latihan modul 2.1" dan sebuah paragraf yang menjelaskan tentang latar belakang halaman. Selain itu, kode kedua dilengkapi dengan komentar dalam CSS yang menjelaskan fungsi dari setiap properti, seperti latar belakang kuning muda dan warna teks merah, yang tidak ada pada kode pertama. Simbol yang digunakan untuk menandai warna juga berbeda, meskipun tidak memengaruhi fungsionalitas kode. Secara keseluruhan, kode kedua lebih lengkap dan informatif dibandingkan dengan kode pertama.

## 2. Sintaks dan Cara Menyisipkan CSS

### 1) Contoh

#### a. Input

##### Inline



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

##### Internal



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

## External

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

## Mystyle

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

### b. Output

#### Inline



#### Internal



#### External



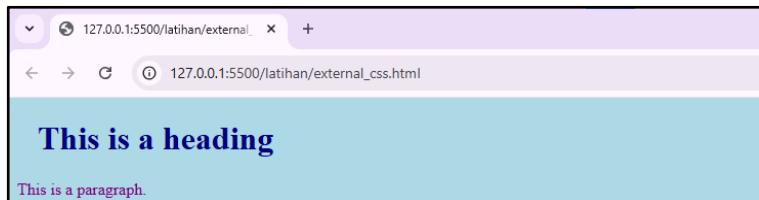
## 2) Latihan

### a. Input

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

```
latihan > # mystyle.css > h1
1  body {
2  |   background-color: lightblue;
3  }
4
5  h1 {
6  |   color: navy;
7  |   margin-left: 20px;
8  }
9
10 p {
11 |   color: purple;
12 }
```

b. Output



c. Analisis

- a) Inline CSS (pendekatan pertama) merupakan metode styling yang paling dasar namun paling tidak efisien. CSS ditulis langsung dalam atribut style elemen HTML, menyebabkan kode menjadi berantakan dan sulit dikelola. Kelebihannya hanya pada spesifisitas tinggi dan kemudahan implementasi awal, tetapi sangat tidak disarankan untuk proyek besar karena sulit di-maintain dan tidak reusable. Contohnya adalah <h1 style="color: blue;"> yang hanya mempengaruhi satu elemen tertentu.
- b) Internal CSS (pendekatan kedua) menunjukkan peningkatan organisasi kode dengan memisahkan styling dalam tag <style> di bagian <head>. Metode ini lebih terstruktur daripada inline CSS dan memungkinkan reuse style dalam satu halaman. Namun tetap memiliki keterbatasan karena style tidak bisa digunakan antar halaman. Contoh properti seperti body {background-color: linen;} berlaku untuk seluruh elemen body dalam halaman tersebut saja.
- c) External CSS (pendekatan ketiga dan keempat) merupakan best practice profesional dengan memisahkan sama sekali CSS dari HTML melalui file terpisah (.css). Pendekatan ini memungkinkan konsistensi design across multiple pages, lebih

mudah dikelola, dan meningkatkan kecepatan loading. File CSS seperti mystyle.css berisi semua aturan styling yang dapat digunakan ulang di banyak halaman. Contoh deklarasi `h1 {color: navy;}` dalam file eksternal akan berlaku untuk semua `h1` di seluruh halaman yang mengimpor file tersebut.

- d) Perkembangan Kode dari versi ketiga ke keempat menunjukkan perluasan fungsionalitas dimana file CSS eksternal diperkaya dengan lebih banyak selector dan properti (seperti penambahan style untuk tag `<p>`). Ini mencerminkan skalabilitas metode external CSS yang mudah dikembangkan tanpa perlu mengubah struktur HTML.

Perbedaan mendasar terletak pada tingkat modularitas dan skalabilitas - dimana inline CSS hanya cocok untuk quick fixes, internal CSS untuk single page prototypes, sementara external CSS untuk proyek profesional berskala besar. Semakin berkembang suatu proyek, semakin penting untuk menggunakan external CSS yang menawarkan maintainability terbaik, konsistensi visual, dan performa optimal.

### 3. Selector CSS

#### 1) Contoh

##### a. Input

```
contoh > selektor_css.html > html
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     <p>Paragraf biasa</p>
12     <p class="highlight">Paragraf dengan highlight</p>
13     <h1 id="judul">Judul Halaman</h1>
14 </body>
15 </html>
```

##### b. Output



## 2) Latihan

### a. Input

```
latihan > selektor.css.html > @html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Selektor</title>
5     <style>
6       /* Gaya dasar paragraf */
7       p {
8         color: #red;
9         padding: 8px;
10        margin: 5px 0;
11      }
12
13      /* Kelas untuk highlight */
14      .highlight {
15        background-color: #yellow;
16        font-weight: bold;
17      }
18
19      /* Gaya judul */
20      #judul {
21        font-size: 24px;
22        color: #darkblue;
23        text-align: center;
24      }
25    </style>
26  </head>
27  <body>
28    <h1 id="judul">Latihan Modul 2.3</h1>
29    <p>Paragraf biasa dengan teks merah</p>
30    <p class="highlight">Paragraf dengan latar kuning dan teks tebal</p>
31    <p>Paragraf biasa lainnya</p>
32  </body>
33 </html>
```

### b. Output



### c. Analisis

Selector class dan ID merupakan dua jenis selector dalam CSS yang memiliki perbedaan mendasar dalam penggunaannya. Selector class seperti `.highlight` bersifat reusable, artinya dapat diterapkan pada banyak elemen sekaligus. Dalam contoh kode ini, class `highlight` digunakan untuk memberikan latar belakang kuning dan teks tebal pada satu paragraf tertentu, namun class yang sama bisa ditambahkan ke elemen-elemen lain jika diperlukan. Sementara itu, selector ID seperti `#judul` bersifat unik dan hanya boleh digunakan untuk satu elemen saja dalam satu halaman. Pada kode ini, ID `judul` digunakan secara eksklusif untuk heading utama (`h1`) dengan gaya font besar dan warna biru tua.

Perbedaan lainnya terletak pada tingkat spesifikasi. Selector ID memiliki bobot yang lebih berat dibandingkan class, sehingga jika terjadi konflik gaya, aturan dari ID akan selalu mengalahkan aturan dari class. Contohnya, meskipun semua paragraf diatur berwarna merah, heading dengan ID `judul` tetap berwarna biru tua karena

spesifisitas ID lebih tinggi. Selector class lebih fleksibel dan cocok untuk menerapkan gaya yang sama ke berbagai elemen, sedangkan ID lebih tepat digunakan untuk elemen unik yang membutuhkan penanganan khusus. Pemahaman perbedaan ini penting untuk mengorganisir kode CSS secara efektif dan menghindari konflik gaya.

#### 4. Properti Warna dan Background

##### 1) Contoh

###### a. Input

```
contoh > propertiwarna_css.html > 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('Mu.png');
12             background-repeat: no-repeat;
13             background-size: cover;
14         }
15     </style>
16 </head>
17 <body>
18     <p>Nama saya abadi nugroho:</p>
19 </body>
20 </html>
```

###### b. Output



##### 2) Latihan

###### a. Input

```
latihan > propertiwarna.css.html > html
1  <!DOCTYPE Html>
2  <html>
3  <head>
4      <title>Propertiwarna</title>
5      <style>
6          body {
7              background-image: url('Mu.png');
8              background-repeat: no-repeat;
9              background-size: cover;
10             background-attachment: fixed;
11             margin: 0;
12             padding: 20px;
13             font-family: Arial, sans-serif;
14         }
15
16         p {
17             color: white;
18             background-color: rgba(128, 128, 128, 0.7); /* Abu-abu dengan transparansi */
19             padding: 5px;
20             border-radius: 5px;
21             max-width: 600px;
22             margin: 10px auto;
23             line-height: 1.6;
24         }
25     </style>
26 </head>
27 <body>
28     <p>Nama saya Hanif:</p>
29     <p>Ini adalah contoh paragraf dengan latar belakang abu-abu dan teks berwarna putih, di atas gambar latar belakang.</p>
30 </body>
31 </html>
```

## b. Output



## c. Analisis

Kode pertama merupakan implementasi dasar yang hanya menampilkan teks sederhana dengan latar belakang hitam di atas gambar background. Kode tersebut hanya menggunakan properti CSS dasar seperti color dan background-color untuk paragraf, serta background-image untuk body. Sementara itu, kode kedua jauh lebih lengkap dan profesional dalam penerapan properti CSS-nya. Terdapat penambahan properti background-attachment: fixed yang membuat gambar latar tidak bergerak saat scrolling, serta penggunaan rgba untuk background-color paragraf yang memberikan efek transparansi.

Perbedaan utama terletak pada tingkat kompleksitas dan penyempurnaan tampilan. Kode kedua menambahkan banyak properti CSS tambahan seperti padding, border-radius, max-width, dan margin auto yang membuat tampilan menjadi lebih rapi dan terstruktur. Selain itu, kode kedua juga menerapkan line-height untuk memperbaiki jarak antar baris dan font-family untuk konsistensi tipografi. Pada kode pertama, teks hanya ditampilkan dengan warna putih di atas background hitam polos, sedangkan kode kedua menggunakan abu-abu transparan yang lebih estetis dan memberikan efek modern dengan border-radius.

Dari segi struktur, kode kedua juga lebih lengkap dengan adanya tag title dan penambahan paragraf kedua sebagai contoh. Penggunaan margin: 0 pada body di kode kedua menghilangkan default spacing browser yang tidak dilakukan di kode pertama. Secara keseluruhan, kode kedua menunjukkan penerapan best practice CSS yang lebih baik dengan mempertimbangkan aspek responsif, estetika, dan

keterbacaan kode yang lebih matang dibandingkan kode pertama yang masih sangat dasar.

## 5. Properti Teks dan Font

### 1) Contoh

#### a. Input

```
contoh > <!DOCTYPE html> <html>
1  <html>
2   <head>
3     <style>
4       h1 {
5         font-family: Arial, sans-serif;
6         font-size: 36px;
7         font-weight: bold;
8         text-align: center;
9         text-decoration: underline;
10      }
11    </style>
12  </head>
13  <body>
14   <h1 id="judul">Judul Halaman</h1>
15  </body>
16 </html>
```

#### b. Output



### 2) Latihan

#### a. Input

```
latihan > <!DOCTYPE html> <html>
1  <html>
2   <head>
3     <title>propertitext</title>
4     <style>
5       /* Gaya untuk judul */
6       h1 {
7         font-family: 'Georgia', serif; /* Font berbeda dari default */
8         font-size: 42px; /* Ukuran besar */
9         color: #2c3e50;
10        text-align: center;
11        margin-bottom: 30px;
12      }
13      /* Gaya untuk paragraf */
14      p {
15        font-family: Arial, sans-serif;
16        text-align: right; /* Kata kanan */
17        text-decoration: underline; /* Garis bawah */
18        line-height: 1.6;
19        padding: 10px;
20        margin: 20px;
21        border-right: 3px solid #3498db; /* Garis vertikal di kanan */
22        padding-right: 15px;
23      }
24    </style>
25  </head>
26  <body>
27   <h1>latihan 2.4.</h1>
28   <p>
29     Ini adalah contoh paragraf dengan teks rata kanan dan memiliki garis bawah.
30     Format ini sering digunakan untuk penekanan khusus pada teks.
31   </p>
32   <p>
33     Paragraf kedua dengan format yang sama. Perhatikan bahwa seluruh teks dalam
34     paragraf ini rata ke sisi kanan dan bergaris bawah.
35   </p>
36 </body>
37 </html>
```

#### b. Output



#### c. Analisis

Kode pertama menampilkan contoh yang sangat sederhana dengan hanya satu elemen `<h1>` yang diberi styling dasar seperti `font-family`

Arial, ukuran font 36px, dan text-decoration underline. Kode tersebut hanya menggunakan properti teks yang fundamental tanpa penyesuaian spacing atau border, dan tidak memiliki konten tambahan selain judul. Sementara itu, kode kedua jauh lebih kompleks dan menerapkan berbagai properti teks yang lebih canggih. Kode kedua menggunakan font Georgia untuk heading dengan ukuran lebih besar (42px) dan warna khusus (#2c3e50), serta menambahkan margin-bottom untuk memberikan jarak dengan elemen berikutnya.

Perbedaan utama terlihat pada penanganan paragraf yang tidak ada sama sekali di kode pertama. Kode kedua menampilkan dua paragraf dengan alignment kanan yang tidak biasa, dilengkapi underline, line-height yang diatur, serta border vertikal di sisi kanan yang memberikan efek visual unik. Kode kedua juga menunjukkan praktik yang lebih baik dengan menambahkan properti padding dan margin yang tepat pada paragraf, serta menggunakan kombinasi font serif dan sans-serif untuk kontras visual. Selain itu, kode kedua lebih lengkap dengan adanya tag <title> dan struktur HTML yang lebih sempurna dengan konten yang nyata, dibandingkan kode pertama yang hanya bersifat demonstrasi minimalis.

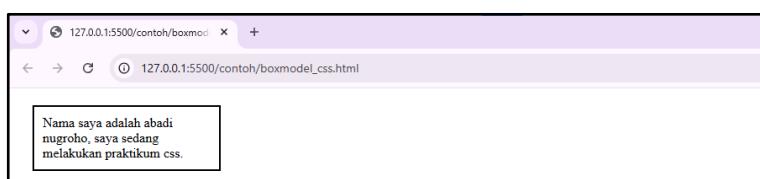
## 6. Box Model (Margin, Border, Padding, dan Content)

### 1) Contoh

#### a. Input

```
contoh > o boxmodel_css.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <style>
5  |   |   div {
6  |   |   |       margin: 20px;
7  |   |   |       padding: 10px;
8  |   |   |       border: 2px solid black;
9  |   |   |       width: 200px;
10 |   |   }
11 |   </style>
12 </head>
13 <body>
14 |   <div>Nama saya adalah abadi nugroho, saya sedang melakukan praktikum css.</div>
15 </body>
16 </html>|
```

#### b. Output



## 2) Latihan

### a. Input

```
latihan > boxmodel_css.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>boxmodel</title>
5      <style>
6          .kotak {
7              margin: 30px;           /* Margin 30px di semua sisi */
8              padding: 15px;         /* Padding 15px di semua sisi */
9              border: 5px solid red; /* Border merah tebal (5px) */
10             width: 300px;          /* Lebar kotak 300px */
11             background-color: #f9f9f9; /* Warna latar belakang */
12             font-family: Arial, sans-serif;
13             color: #333;
14             box-shadow: 0 0 10px rgba(0,0,0,0.1); /* Efek bayangan */
15         }
16     </style>
17 </head>
18 <body>
19     <div class="kotak">
20         Nama saya adalah Hanif, saya sedang melakukan praktikum CSS.
21         Kotak ini memiliki margin 30px, padding 15px, border merah tebal,
22         dan lebar 300px sesuai permintaan.
23     </div>
24 </body>
25 </html>
```

### b. Output



### c. Analisis

Kode pertama merupakan implementasi dasar box model dengan properti yang minimal dan sederhana. Div tersebut hanya memiliki margin 20px, padding 10px, border hitam selebar 2px, dan lebar fixed 200px tanpa warna latar atau properti dekoratif lainnya. Tidak ada class yang digunakan dan styling diterapkan langsung ke tag div. Teks yang ditampilkan juga sangat sederhana tanpa penjelasan tentang properti yang digunakan.

Sedangkan kode kedua menunjukkan implementasi box model yang jauh lebih lengkap dan profesional. Menggunakan class selector (.kotak) dengan margin yang lebih besar (30px), padding lebih lebar (15px), dan border merah tebal (5px) yang lebih mencolok. Kode ini menambahkan banyak properti tambahan seperti background color (#f9f9f9), box shadow untuk efek kedalaman, font family yang spesifik, dan color text (#333) yang lebih soft. Lebar kotak juga lebih besar (300px) dan yang penting, kode ini dilengkapi dengan title page serta penjelasan dalam kontennya tentang spesifikasi box model yang diterapkan.

Perbedaan utama terletak pada kelengkapan dan pendekatan styling. Kode pertama hanya menunjukkan fungsi dasar box model, sementara kode kedua memperhatikan aspek estetika, readability, dan praktik coding yang lebih baik dengan penggunaan class selector. Kode kedua juga lebih informatif baik dalam struktur HTML (dengan tag title) maupun dalam konten teks yang menjelaskan properti CSS yang digunakan.

## 7. Layout dengan Flexbox

### 1) Contoh

#### a. Input

```
contoh > flexbox_css.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <style>
5          .container {
6              display: flex;
7              justify-content: space-around;
8              align-items: center;
9          }
10         .item {
11             background-color: #lightcoral;
12             padding: 20px;
13             margin: 10px;
14         }
15     </style>
16 </head>
17 <body>
18     <div class="container">
19         <div class="item">Item 1</div>
20         <div class="item">Item 2</div>
21         <div class="item">Item 3</div>
22     </div>
23 </body>
24 </html>
```

#### b. Output



### 2) Latihan

#### a. Input

```
latihan > flexbox_css.html > html > body > div.container > div.item
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>flexbox</title>
5      <style>
6          .container {
7              display: flex;
8              justify-content: space-evenly; /* Peningkatan lebih merata */
9              align-items: center; /* Posisi vertikal di tengah */
10             min-height: 200px; /* Tinggi minimum container */
11             background-color: #f5f5f5; /* Warna latar belakang */
12             border: 1px solid #ddd; /* Garis tepi */
13             border-radius: 10px; /* Sudut melengkung */
14             padding: 20px;
15             margin: 20px auto; /* Posisi horizontal di tengah */
16             max-width: 800px; /* Lebar maksimum */
17         }
18
19         .item {
20             background-color: #ff6666; /* Warna lebih modern */
21             color: white; /* Warna teks */
22             padding: 10px 20px; /* Padding atas-bawah lebih besar */
23             margin: 10px; /* Jarak antara item */
24             border-radius: 6px; /* Sudut melengkung */
25             text-align: center; /* Tekst di tengah */
26             font-family: Arial, sans-serif;
27             font-weight: bold;
28             box-shadow: 0 4px 8px #rgba(0,0,0,1); /* Efek bayangan */
29             flex: 1; /* Item fleksibel yang sama lebar */
30             max-width: 200px; /* Lebar maksimum item */
31         }
32     </style>
33 </head>
34 <body>
35     <div class="container">
36         <div class="item">Mu&lt;br&gt;Hansf</div>
37         <div class="item">Alfaqiq</div>
38     </div>
39 </body>
40 </html>
```

## b. Output



## c. Analisis

Kode pertama merupakan implementasi dasar Flexbox yang hanya menampilkan fungsi inti dari flex layout tanpa dekorasi tambahan. Container hanya memiliki properti flex dasar seperti `display: flex`, `justify-content: space-around`, dan `align-items: center`, dengan item-item berwarna lightcoral polos. Tidak ada pengaturan dimensi container yang jelas, shadow, rounded corner, atau properti estetika lainnya. Item-item juga hanya memiliki padding dan margin dasar tanpa pengaturan fleksibilitas (flex property) atau pembatasan lebar maksimum.

Kode kedua menyajikan penyempurnaan yang signifikan dari segi visual dan fungsionalitas. Container sudah dilengkapi `background-color`, `border`, `rounded corner`, `shadow effect`, dan pengaturan dimensi yang lebih presisi (`min-height`, `max-width`). Properti `justify-content: space-evenly` digunakan untuk distribusi space yang lebih merata dibanding `space-around` pada kode pertama. Item-item memiliki desain yang lebih modern dengan warna yang lebih menarik, typography yang diatur, shadow effect, dan yang paling penting - properti `flex: 1` yang membuat item dapat tumbuh dan menyusut secara proporsional, dibatasi oleh `max-width`. Kode kedua juga lebih lengkap dengan struktur HTML yang sempurna termasuk `title page` dan penutup tag `div` yang benar.

Perbedaan utama terletak pada pendekatan desain - dimana kode pertama hanya menunjukkan fungsi dasar Flexbox, sementara kode kedua menerapkan best practice modern web design dengan memperhatikan aspek visual, responsivitas, dan detail teknis seperti

pengaturan fleksibilitas item dan batasan ukuran yang lebih terkontrol.

## 8. CSS Grid Layout

### 1) Contoh

#### a. Input

```
contoh > O grid.css.html > O html
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5      .grid-container {
6          display: grid;
7          grid-template-columns: 100px 200px 100px;
8          grid-template-rows: 50px 10px;
9          gap: 10px;
10         background-color: #f2f2f2;
11         padding: 10px;
12     }
13     .grid-item {
14         background-color: #e6f5e0;
15         color: black;
16         text-align: center;
17         padding: 20px;
18         font-size: 18px;
19     }
20  </style>
21 </head>
22 <body>
23     <div class="grid-container">
24         <div class="grid-item">1</div>
25         <div class="grid-item">2</div>
26         <div class="grid-item">3</div>
27         <div class="grid-item">4</div>
28         <div class="grid-item">5</div>
29         <div class="grid-item">6</div>
30     </div>
31 </body>
32 </html>
```

#### b. Output

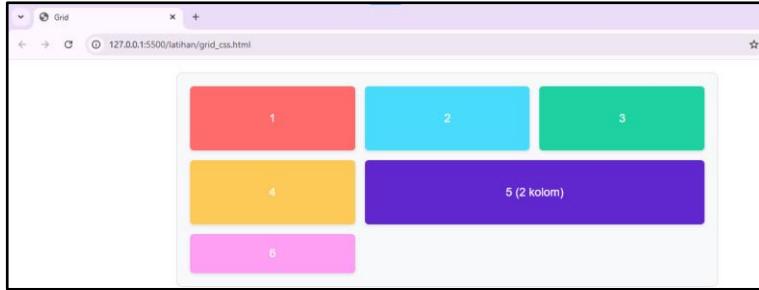


### 2) Latihan

#### a. Input

```
latihan > O grid.latihan > O html > O head > O style
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Grid Container</title>
5  <style>
6      .grid-container {
7          display: grid;
8          grid-template-columns: repeat(2, 1fr); /* 2 kolom dengan lebar sama */
9          grid-template-rows: repeat(2, 100px); /* 2 baris dengan tinggi sama */
10         gap: 10px; /* Jarak antar item type 7 */
11         background-color: #f0f0f0;
12         padding: 10px;
13         max-width: 800px;
14         margin: auto;
15         border: 1px solid #ddd;
16         border-radius: 10px;
17     }
18
19     .grid-item {
20         color: white;
21         text-align: center;
22         border: 1px solid black;
23         font-size: 18px;
24         font-family: Arial, sans-serif;
25         height: fit-content;
26         align-items: center;
27         justify-content: center;
28         border-radius: 10px;
29         box-shadow: 0 2px 5px rgba(0,0,0,0.1);
30     }
31
32     /* Item yang berada pada posisi index 0 */
33     .item0 { background-color: #ffbbbf; }
34     .item1 { background-color: #ffccbc; }
35     .item2 { background-color: #ffccbc; }
36     .item3 { background-color: #ffca28; }
37     .item4 { background-color: #ffca28; }
38     .item5 { background-color: #ffca28; }
39
40     /* Item yang memenuhi dua kondisi */
41     .item6 { background-color: #ffbbbf; }
42     .item7 { background-color: #ffccbc; }
43     .item8 { background-color: #ffccbc; }
44     .item9 { background-color: #ffca28; }
45     .item10 { background-color: #ffca28; }
46
47     /* Item yang memenuhi tiga kondisi */
48     .item11 { background-color: #ffccbc; }
49     .item12 { background-color: #ffccbc; }
50     .item13 { background-color: #ffca28; }
51     .item14 { background-color: #ffca28; }
52     .item15 { background-color: #ffca28; }
53
54     /* Item yang memenuhi empat kondisi */
55     .item16 { background-color: #ffccbc; }
56     .item17 { background-color: #ffccbc; }
57     .item18 { background-color: #ffccbc; }
58     .item19 { background-color: #ffca28; }
59     .item20 { background-color: #ffca28; }
60
61     /* Item yang memenuhi lima kondisi */
62     .item21 { background-color: #ffccbc; }
63     .item22 { background-color: #ffccbc; }
64     .item23 { background-color: #ffccbc; }
65     .item24 { background-color: #ffccbc; }
66     .item25 { background-color: #ffccbc; }
67
68     /* Item yang memenuhi semua kondisi */
69     .item26 { background-color: #ffccbc; }
70     .item27 { background-color: #ffccbc; }
71     .item28 { background-color: #ffccbc; }
72     .item29 { background-color: #ffccbc; }
73     .item30 { background-color: #ffccbc; }
74
75     /* Item yang memenuhi semua kondisi */
76     .item31 { background-color: #ffccbc; }
77     .item32 { background-color: #ffccbc; }
78     .item33 { background-color: #ffccbc; }
79     .item34 { background-color: #ffccbc; }
80     .item35 { background-color: #ffccbc; }
81
82     /* Item yang memenuhi semua kondisi */
83     .item36 { background-color: #ffccbc; }
84     .item37 { background-color: #ffccbc; }
85     .item38 { background-color: #ffccbc; }
86     .item39 { background-color: #ffccbc; }
87     .item40 { background-color: #ffccbc; }
88
89     /* Item yang memenuhi semua kondisi */
90     .item41 { background-color: #ffccbc; }
91     .item42 { background-color: #ffccbc; }
92     .item43 { background-color: #ffccbc; }
93     .item44 { background-color: #ffccbc; }
94     .item45 { background-color: #ffccbc; }
95
96     /* Item yang memenuhi semua kondisi */
97     .item46 { background-color: #ffccbc; }
98     .item47 { background-color: #ffccbc; }
99     .item48 { background-color: #ffccbc; }
100    .item49 { background-color: #ffccbc; }
101    .item50 { background-color: #ffccbc; }
102
103    /* Item yang memenuhi semua kondisi */
104    .item51 { background-color: #ffccbc; }
105    .item52 { background-color: #ffccbc; }
106    .item53 { background-color: #ffccbc; }
107    .item54 { background-color: #ffccbc; }
108    .item55 { background-color: #ffccbc; }
109
110    /* Item yang memenuhi semua kondisi */
111    .item56 { background-color: #ffccbc; }
112    .item57 { background-color: #ffccbc; }
113    .item58 { background-color: #ffccbc; }
114    .item59 { background-color: #ffccbc; }
115    .item60 { background-color: #ffccbc; }
116
117    /* Item yang memenuhi semua kondisi */
118    .item61 { background-color: #ffccbc; }
119    .item62 { background-color: #ffccbc; }
120    .item63 { background-color: #ffccbc; }
121    .item64 { background-color: #ffccbc; }
122    .item65 { background-color: #ffccbc; }
123
124    /* Item yang memenuhi semua kondisi */
125    .item66 { background-color: #ffccbc; }
126    .item67 { background-color: #ffccbc; }
127    .item68 { background-color: #ffccbc; }
128    .item69 { background-color: #ffccbc; }
129    .item70 { background-color: #ffccbc; }
130
131    /* Item yang memenuhi semua kondisi */
132    .item71 { background-color: #ffccbc; }
133    .item72 { background-color: #ffccbc; }
134    .item73 { background-color: #ffccbc; }
135    .item74 { background-color: #ffccbc; }
136    .item75 { background-color: #ffccbc; }
137
138    /* Item yang memenuhi semua kondisi */
139    .item76 { background-color: #ffccbc; }
140    .item77 { background-color: #ffccbc; }
141    .item78 { background-color: #ffccbc; }
142    .item79 { background-color: #ffccbc; }
143    .item80 { background-color: #ffccbc; }
144
145    /* Item yang memenuhi semua kondisi */
146    .item81 { background-color: #ffccbc; }
147    .item82 { background-color: #ffccbc; }
148    .item83 { background-color: #ffccbc; }
149    .item84 { background-color: #ffccbc; }
150    .item85 { background-color: #ffccbc; }
151
152    /* Item yang memenuhi semua kondisi */
153    .item86 { background-color: #ffccbc; }
154    .item87 { background-color: #ffccbc; }
155    .item88 { background-color: #ffccbc; }
156    .item89 { background-color: #ffccbc; }
157    .item90 { background-color: #ffccbc; }
158
159    /* Item yang memenuhi semua kondisi */
160    .item91 { background-color: #ffccbc; }
161    .item92 { background-color: #ffccbc; }
162    .item93 { background-color: #ffccbc; }
163    .item94 { background-color: #ffccbc; }
164    .item95 { background-color: #ffccbc; }
165
166    /* Item yang memenuhi semua kondisi */
167    .item96 { background-color: #ffccbc; }
168    .item97 { background-color: #ffccbc; }
169    .item98 { background-color: #ffccbc; }
170    .item99 { background-color: #ffccbc; }
171    .item100 { background-color: #ffccbc; }
172
173    /* Item yang memenuhi semua kondisi */
174    .item101 { background-color: #ffccbc; }
175    .item102 { background-color: #ffccbc; }
176    .item103 { background-color: #ffccbc; }
177    .item104 { background-color: #ffccbc; }
178    .item105 { background-color: #ffccbc; }
179
180    /* Item yang memenuhi semua kondisi */
181    .item106 { background-color: #ffccbc; }
182    .item107 { background-color: #ffccbc; }
183    .item108 { background-color: #ffccbc; }
184    .item109 { background-color: #ffccbc; }
185    .item110 { background-color: #ffccbc; }
186
187    /* Item yang memenuhi semua kondisi */
188    .item111 { background-color: #ffccbc; }
189    .item112 { background-color: #ffccbc; }
190    .item113 { background-color: #ffccbc; }
191    .item114 { background-color: #ffccbc; }
192    .item115 { background-color: #ffccbc; }
193
194    /* Item yang memenuhi semua kondisi */
195    .item116 { background-color: #ffccbc; }
196    .item117 { background-color: #ffccbc; }
197    .item118 { background-color: #ffccbc; }
198    .item119 { background-color: #ffccbc; }
199    .item120 { background-color: #ffccbc; }
200
201    /* Item yang memenuhi semua kondisi */
202    .item121 { background-color: #ffccbc; }
203    .item122 { background-color: #ffccbc; }
204    .item123 { background-color: #ffccbc; }
205    .item124 { background-color: #ffccbc; }
206    .item125 { background-color: #ffccbc; }
207
208    /* Item yang memenuhi semua kondisi */
209    .item126 { background-color: #ffccbc; }
210    .item127 { background-color: #ffccbc; }
211    .item128 { background-color: #ffccbc; }
212    .item129 { background-color: #ffccbc; }
213    .item130 { background-color: #ffccbc; }
214
215    /* Item yang memenuhi semua kondisi */
216    .item131 { background-color: #ffccbc; }
217    .item132 { background-color: #ffccbc; }
218    .item133 { background-color: #ffccbc; }
219    .item134 { background-color: #ffccbc; }
220    .item135 { background-color: #ffccbc; }
221
222    /* Item yang memenuhi semua kondisi */
223    .item136 { background-color: #ffccbc; }
224    .item137 { background-color: #ffccbc; }
225    .item138 { background-color: #ffccbc; }
226    .item139 { background-color: #ffccbc; }
227    .item140 { background-color: #ffccbc; }
228
229    /* Item yang memenuhi semua kondisi */
230    .item141 { background-color: #ffccbc; }
231    .item142 { background-color: #ffccbc; }
232    .item143 { background-color: #ffccbc; }
233    .item144 { background-color: #ffccbc; }
234    .item145 { background-color: #ffccbc; }
235
236    /* Item yang memenuhi semua kondisi */
237    .item146 { background-color: #ffccbc; }
238    .item147 { background-color: #ffccbc; }
239    .item148 { background-color: #ffccbc; }
240    .item149 { background-color: #ffccbc; }
241    .item150 { background-color: #ffccbc; }
242
243    /* Item yang memenuhi semua kondisi */
244    .item151 { background-color: #ffccbc; }
245    .item152 { background-color: #ffccbc; }
246    .item153 { background-color: #ffccbc; }
247    .item154 { background-color: #ffccbc; }
248    .item155 { background-color: #ffccbc; }
249
250    /* Item yang memenuhi semua kondisi */
251    .item156 { background-color: #ffccbc; }
252    .item157 { background-color: #ffccbc; }
253    .item158 { background-color: #ffccbc; }
254    .item159 { background-color: #ffccbc; }
255    .item160 { background-color: #ffccbc; }
256
257    /* Item yang memenuhi semua kondisi */
258    .item161 { background-color: #ffccbc; }
259    .item162 { background-color: #ffccbc; }
260    .item163 { background-color: #ffccbc; }
261    .item164 { background-color: #ffccbc; }
262    .item165 { background-color: #ffccbc; }
263
264    /* Item yang memenuhi semua kondisi */
265    .item166 { background-color: #ffccbc; }
266    .item167 { background-color: #ffccbc; }
267    .item168 { background-color: #ffccbc; }
268    .item169 { background-color: #ffccbc; }
269    .item170 { background-color: #ffccbc; }
270
271    /* Item yang memenuhi semua kondisi */
272    .item171 { background-color: #ffccbc; }
273    .item172 { background-color: #ffccbc; }
274    .item173 { background-color: #ffccbc; }
275    .item174 { background-color: #ffccbc; }
276    .item175 { background-color: #ffccbc; }
277
278    /* Item yang memenuhi semua kondisi */
279    .item176 { background-color: #ffccbc; }
280    .item177 { background-color: #ffccbc; }
281    .item178 { background-color: #ffccbc; }
282    .item179 { background-color: #ffccbc; }
283    .item180 { background-color: #ffccbc; }
284
285    /* Item yang memenuhi semua kondisi */
286    .item181 { background-color: #ffccbc; }
287    .item182 { background-color: #ffccbc; }
288    .item183 { background-color: #ffccbc; }
289    .item184 { background-color: #ffccbc; }
290    .item185 { background-color: #ffccbc; }
291
292    /* Item yang memenuhi semua kondisi */
293    .item186 { background-color: #ffccbc; }
294    .item187 { background-color: #ffccbc; }
295    .item188 { background-color: #ffccbc; }
296    .item189 { background-color: #ffccbc; }
297    .item190 { background-color: #ffccbc; }
298
299    /* Item yang memenuhi semua kondisi */
300    .item191 { background-color: #ffccbc; }
301    .item192 { background-color: #ffccbc; }
302    .item193 { background-color: #ffccbc; }
303    .item194 { background-color: #ffccbc; }
304    .item195 { background-color: #ffccbc; }
305
306    /* Item yang memenuhi semua kondisi */
307    .item196 { background-color: #ffccbc; }
308    .item197 { background-color: #ffccbc; }
309    .item198 { background-color: #ffccbc; }
310    .item199 { background-color: #ffccbc; }
311    .item200 { background-color: #ffccbc; }
312
313    /* Item yang memenuhi semua kondisi */
314    .item201 { background-color: #ffccbc; }
315    .item202 { background-color: #ffccbc; }
316    .item203 { background-color: #ffccbc; }
317    .item204 { background-color: #ffccbc; }
318    .item205 { background-color: #ffccbc; }
319
320    /* Item yang memenuhi semua kondisi */
321    .item206 { background-color: #ffccbc; }
322    .item207 { background-color: #ffccbc; }
323    .item208 { background-color: #ffccbc; }
324    .item209 { background-color: #ffccbc; }
325    .item210 { background-color: #ffccbc; }
326
327    /* Item yang memenuhi semua kondisi */
328    .item211 { background-color: #ffccbc; }
329    .item212 { background-color: #ffccbc; }
330    .item213 { background-color: #ffccbc; }
331    .item214 { background-color: #ffccbc; }
332    .item215 { background-color: #ffccbc; }
333
334    /* Item yang memenuhi semua kondisi */
335    .item216 { background-color: #ffccbc; }
336    .item217 { background-color: #ffccbc; }
337    .item218 { background-color: #ffccbc; }
338    .item219 { background-color: #ffccbc; }
339    .item220 { background-color: #ffccbc; }
340
341    /* Item yang memenuhi semua kondisi */
342    .item221 { background-color: #ffccbc; }
343    .item222 { background-color: #ffccbc; }
344    .item223 { background-color: #ffccbc; }
345    .item224 { background-color: #ffccbc; }
346    .item225 { background-color: #ffccbc; }
347
348    /* Item yang memenuhi semua kondisi */
349    .item226 { background-color: #ffccbc; }
350    .item227 { background-color: #ffccbc; }
351    .item228 { background-color: #ffccbc; }
352    .item229 { background-color: #ffccbc; }
353    .item230 { background-color: #ffccbc; }
354
355    /* Item yang memenuhi semua kondisi */
356    .item231 { background-color: #ffccbc; }
357    .item232 { background-color: #ffccbc; }
358    .item233 { background-color: #ffccbc; }
359    .item234 { background-color: #ffccbc; }
360    .item235 { background-color: #ffccbc; }
361
362    /* Item yang memenuhi semua kondisi */
363    .item236 { background-color: #ffccbc; }
364    .item237 { background-color: #ffccbc; }
365    .item238 { background-color: #ffccbc; }
366    .item239 { background-color: #ffccbc; }
367    .item240 { background-color: #ffccbc; }
368
369    /* Item yang memenuhi semua kondisi */
370    .item241 { background-color: #ffccbc; }
371    .item242 { background-color: #ffccbc; }
372    .item243 { background-color: #ffccbc; }
373    .item244 { background-color: #ffccbc; }
374    .item245 { background-color: #ffccbc; }
375
376    /* Item yang memenuhi semua kondisi */
377    .item246 { background-color: #ffccbc; }
378    .item247 { background-color: #ffccbc; }
379    .item248 { background-color: #ffccbc; }
380    .item249 { background-color: #ffccbc; }
381    .item250 { background-color: #ffccbc; }
382
383    /* Item yang memenuhi semua kondisi */
384    .item251 { background-color: #ffccbc; }
385    .item252 { background-color: #ffccbc; }
386    .item253 { background-color: #ffccbc; }
387    .item254 { background-color: #ffccbc; }
388    .item255 { background-color: #ffccbc; }
389
390    /* Item yang memenuhi semua kondisi */
391    .item256 { background-color: #ffccbc; }
392    .item257 { background-color: #ffccbc; }
393    .item258 { background-color: #ffccbc; }
394    .item259 { background-color: #ffccbc; }
395    .item260 { background-color: #ffccbc; }
396
397    /* Item yang memenuhi semua kondisi */
398    .item261 { background-color: #ffccbc; }
399    .item262 { background-color: #ffccbc; }
400    .item263 { background-color: #ffccbc; }
401    .item264 { background-color: #ffccbc; }
402    .item265 { background-color: #ffccbc; }
```

## b. Output



## c. Analisis

Kode pertama merupakan contoh dasar yang hanya menampilkan fungsi minimal grid layout dengan pendekatan kaku (rigid). Grid tersebut menggunakan ukuran kolom absolut (100px, 200px, 100px) dan baris dengan dimensi tetap (50px, 100px), yang kurang fleksibel terhadap berbagai ukuran layar. Desainnya sangat sederhana dengan warna background hijau seragam untuk semua item (#4CAF50), tanpa variasi warna, efek visual, atau penyesuaian khusus. Jarak antar item (gap) hanya 10px dengan padding container minimalis 10px, dan tidak ada pengaturan radius sudut (border-radius) atau shadow effect. Kode ini juga tidak memiliki title page dan struktur HTML yang lengkap.

Kode kedua menyajikan penyempurnaan signifikan dengan pendekatan responsive dan desain modern. Menggunakan unit fr (fractional unit) dalam grid-template-columns: repeat(3, 1fr) yang membuat layout lebih fleksibel dan adaptif terhadap berbagai ukuran layar. Terdapat fitur advanced seperti grid-column: span 2 pada item5 yang memungkinkan item menempati multiple kolom. Setiap item memiliki warna background berbeda yang didefinisikan melalui class khusus (item1-item6), menciptakan visual yang lebih menarik. Kode ini menerapkan berbagai properti desain modern seperti border-radius (8px untuk container, 6px untuk item), box-shadow, dan flexbox di dalam grid item untuk penempatan konten yang presisi (menggunakan display: flex dengan justify-content dan align-items). Container memiliki max-width 800px dan margin auto

untuk posisi tengah halaman, serta padding yang lebih generous (20px) dibanding kode pertama.

Perbedaan utama terletak pada pendekatan desain - dimana kode pertama hanya menunjukkan konsep dasar grid dengan ukuran fixed, sementara kode kedua menerapkan teknik layout modern dengan kombinasi grid dan flexbox, sistem warna yang terencana, efek visual yang halus, dan struktur yang lebih responsif. Kode kedua juga lebih profesional dengan penambahan title page dan memperhatikan aspek accessibility melalui kontras warna teks yang baik.

## 9. Responsive Design (Desain Responsif)

### 1) Contoh

#### a. Input

```
contoh > responsive_css.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <style>
5          .grid-container {
6              display: grid;
7              grid-template-columns: repeat(3, 1fr);
8              gap: 10px;
9              padding: 10px;
10         }
11         .grid-item {
12             background-color: #2196F3;
13             color: white;
14             padding: 20px;
15             font-size: 20px;
16             text-align: center;
17         }
18
19         /* Responsive untuk layar kecil */
20         @media only screen and (max-width: 600px) {
21             .grid-container {
22                 grid-template-columns: 1fr; /* 1 kolom */
23             }
24         }
25     </style>
26 </head>
27 <body>
28     <div class="grid-container">
29         <div class="grid-item">Item 1</div>
30         <div class="grid-item">Item 2</div>
31         <div class="grid-item">Item 3</div>
32     </div>
33 </body>
34 </html>
```

#### b. Output



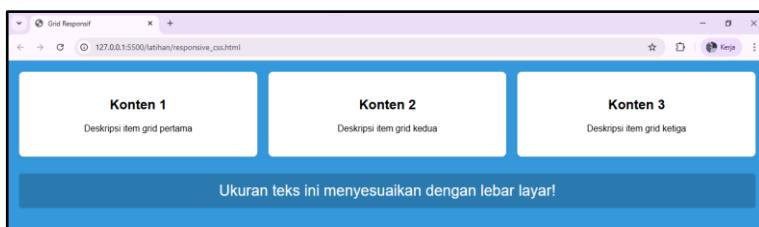
## 2) Latihan

### a. Input

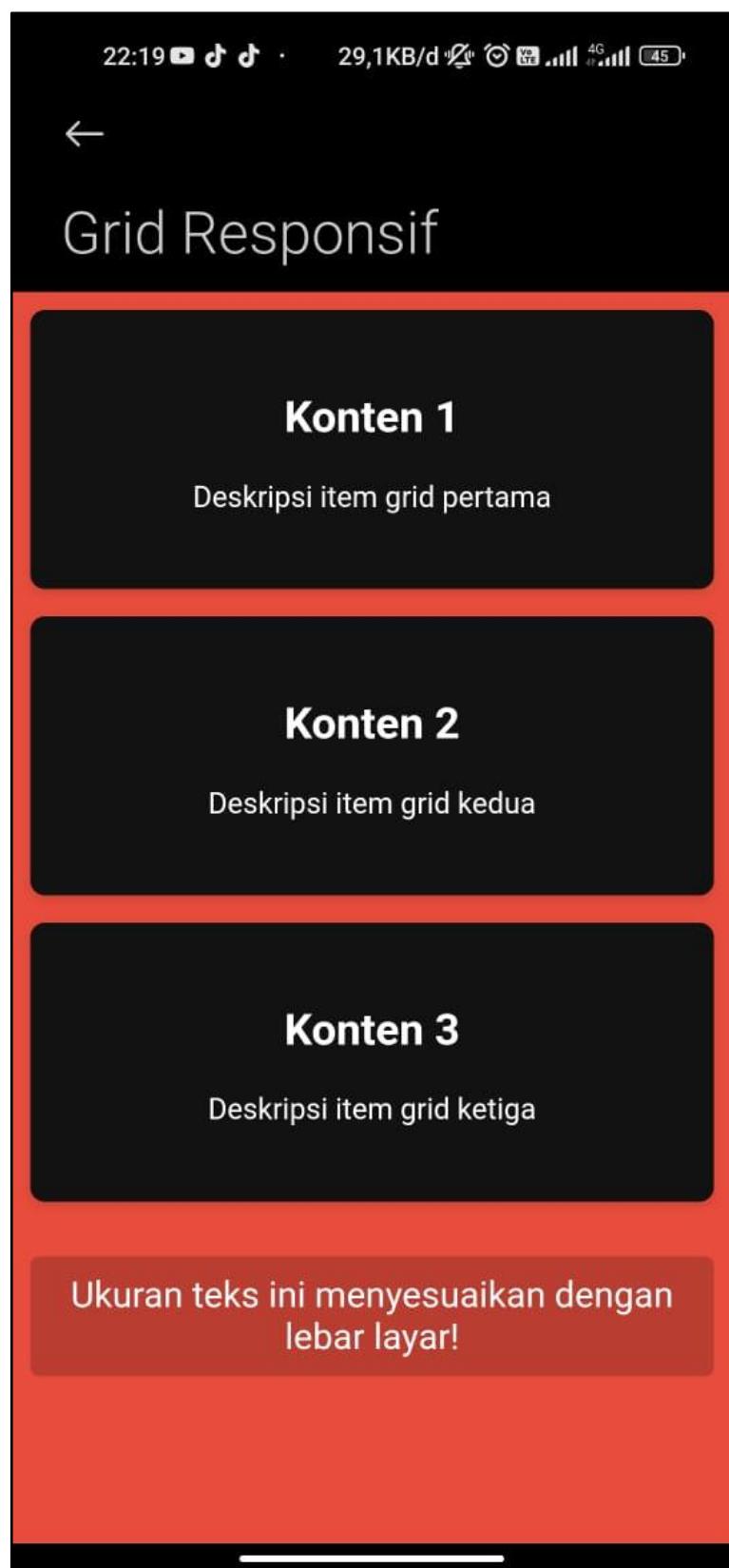
```
lathan > responsive.css.html > html > head > style > {} @media screen and (max-width: 600px)
1  <!DOCTYPE html>
2  <html lang="id">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Grid Responsif</title>
7      <style>
8          /* Gaya dasar untuk desktop */
9          body {
10              background-color: #3498db; /* Biru untuk desktop */
11              margin: 0;
12              padding: 20px;
13              font-family: Arial, sans-serif;
14              transition: all 0.3s ease;
15          }
16
17          .grid-container {
18              display: grid;
19              grid-template-columns: repeat(3, 1fr);
20              gap: 20px;
21              margin-bottom: 30px;
22          }
23
24          .grid-item {
25              background-color: white;
26              border-radius: 8px;
27              padding: 25px;
28              box-shadow: 0 2px 5px rgba(0,0,0,0.1);
29              text-align: center;
30          }
31
32          .responsive-text {
33              font-size: 1em; /* Ukuran teks responsif */
34              color: black;
35              text-align: center;
36              padding: 15px;
37              background-color: rgba(0,0,0,0.2);
38              border-radius: 5px;
39              margin-top: 20px;
40          }
41
42
43          /* Media query untuk mobile (600px) */
44          @media screen and (max-width: 600px) {
45              body {
46                  background-color: #e74c3c; /* Merah untuk mobile */
47                  padding: 10px;
48              }
49
50              .grid-container {
51                  grid-template-columns: 1fr; /* 1 kolom untuk mobile */
52                  gap: 15px;
53              }
54
55              .responsive-text {
56                  font-size: 5vw; /* Ukuran lebih besar di mobile */
57                  padding: 10px;
58              }
59          }
60      </style>
61  </head>
62  <body>
63      <div class="grid-container">
64          <div class="grid-item">
65              <div>Konten 1</div>
66              <p>Deskripsi item grid pertama</p>
67          </div>
68          <div class="grid-item">
69              <div>Konten 2</div>
70              <p>Deskripsi item grid kedua</p>
71          </div>
72          <div class="grid-item">
73              <div>Konten 3</div>
74              <p>Deskripsi item grid ketiga</p>
75          </div>
76
77          <div class="responsive-text">
78              (Ukuran teks ini menyesuaikan dengan lebar layar!
79          </div>
80      </div>
81  </body>
82 </html>
```

### b. Output

#### Desktop



Smartphone



### c. Analisis

Kode pertama merupakan contoh dasar yang hanya menampilkan fungsi minimal grid layout dengan pendekatan yang sangat sederhana. Grid tersebut menggunakan 3 kolom dengan lebar sama (1fr) dan hanya memiliki satu media query dasar untuk mengubah layout menjadi 1 kolom saat layar  $\leq 600\text{px}$ . Desainnya sangat minimalis dengan warna background biru tetap (#2196F3) untuk semua item, tanpa variasi warna, efek visual, atau penyesuaian khusus. Kode ini tidak memiliki meta tag viewport yang penting untuk responsivitas sebenarnya, serta tidak mengatur transisi atau perubahan warna antara mode desktop dan mobile.

Kode kedua menyajikan implementasi grid yang jauh lebih lengkap dan profesional. Terdapat beberapa peningkatan signifikan: (1) Penambahan meta tag viewport dan charset UTF-8 yang penting untuk responsivitas dan kompatibilitas karakter; (2) Sistem warna yang berubah secara dinamis (dari biru ke merah) saat beralih ke mode mobile; (3) Desain grid item yang lebih elaboratif dengan white background, shadow effect, dan border radius; (4) Penggunaan unit vw untuk ukuran teks yang benar-benar responsif; (5) Efek transisi halus (transition: all 0.3s ease) saat perubahan layout; serta (6) Penambahan komponen teks responsif terpisah yang menunjukkan fleksibilitas ukuran font. Kode ini juga lebih semantik dengan penggunaan tag h2 dan p yang tepat dalam grid item.

Perbedaan utama terletak pada pendekatan holistik - dimana kode pertama hanya menunjukkan fungsi grid dasar, sementara kode kedua menerapkan prinsip mobile-first design yang sesungguhnya dengan mempertimbangkan aspek UX, aksesibilitas, dan transisi visual yang halus. Kode kedua juga lebih siap untuk produksi dengan struktur HTML yang lengkap dan properti CSS yang lebih matang.

## E. Tugas

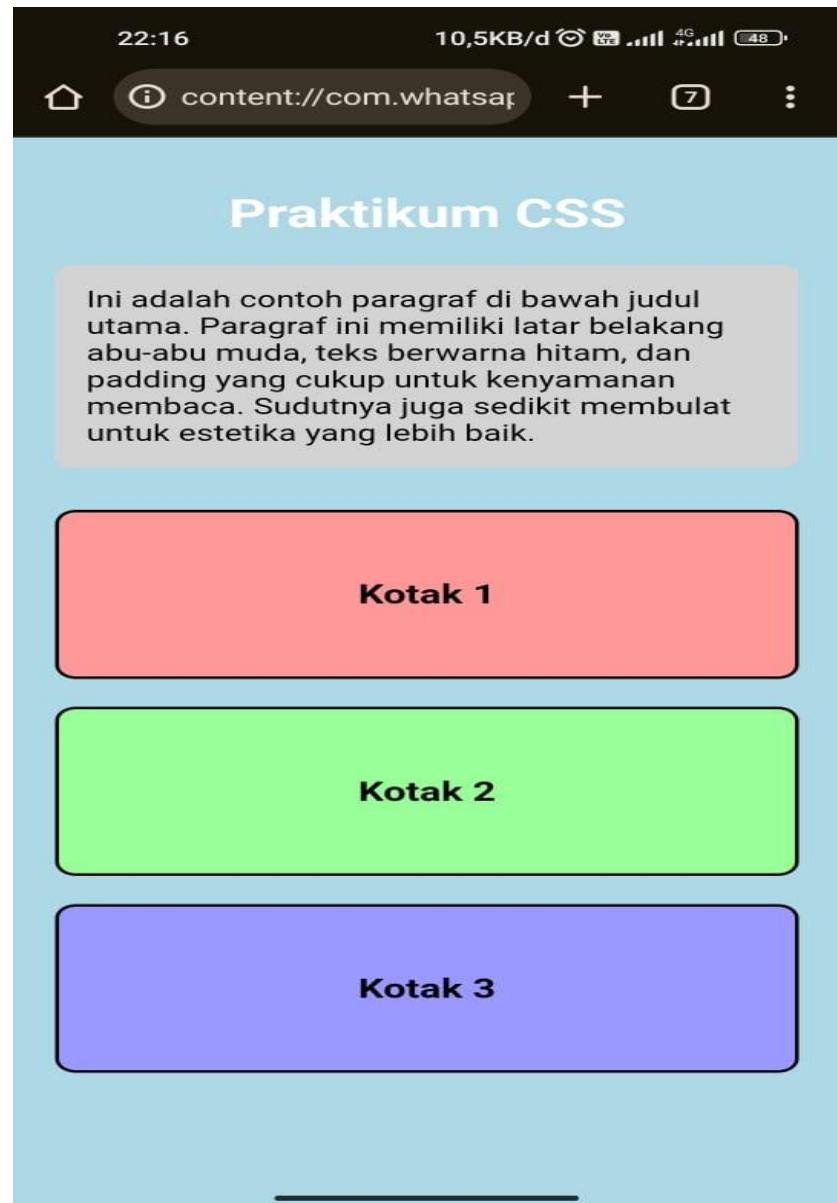
1. Buatlah sebuah halaman web sederhana dengan ketentuan sebagai berikut:

a. Input

```
1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Tugas 2 CSS</title>
7   <style>
8     body {
9       background-color: #faddbd;
10      font-family: Arial, sans-serif;
11      margin: 0;
12      padding: 20px;
13      box-sizing: border-box;
14    }
15
16    h1 {
17      color: #fff;
18      font-size: 40px;
19      text-align: center;
20      margin-bottom: 20px;
21    }
22
23    p {
24      background-color: #d3d3d3;
25      color: #000;
26      padding: 10px;
27      margin: 0 auto 20px auto;
28      max-width: 80px;
29      border-radius: 8px;
30      border: 1px solid black;
31      box-sizing: border-box;
32    }
33
34    .container {
35      display: grid;
36      grid-template-columns: repeat(3, 1fr);
37      gap: 20px;
38      max-width: 800px;
39      margin: 0 auto;
40      box-sizing: border-box;
41    }
42
43    .box {
44      border: 2px solid black;
45      padding: 10px;
46      height: 120px;
47      border-radius: 10px;
48      justify-content: center;
49      align-items: center;
50      font-weight: bold;
51      font-size: 1.2em;
52    }
53
54
55    .box1 {
56      background-color: #ff9999;
57    }
58
59    .box2 {
60      background-color: #99ff99;
61    }
62
63    .box3 {
64      background-color: #9999ff;
65    }
66
67    @media (max-width: 768px) {
68      .container {
69        grid-template-columns: repeat(2, 1fr);
70      }
71    }
72
73    @media (max-width: 480px) {
74      h1 {
75        font-size: 28px;
76      }
77
78      p {
79        padding: 15px;
80      }
81
82      .container {
83        grid-template-columns: 1fr;
84      }
85
86      .box {
87        height: 120px;
88      }
89    }
90  </style>
91 </head>
92 <body>
93   <h1>Praktikum CSS</h1>
94   <p>Ini adalah contoh paragraf di bawah judul utama. Paragraf ini memiliki latar belakang abu-abu muda, teks berwarna hitam, dan padding yang cukup untuk kenyamanan membaca. Sudutnya juga sedikit membulat untuk estetika yang lebih baik.</p>
95
96   <div class="container">
97     <div class="box box1">Kotak 1</div>
98     <div class="box box2">Kotak 2</div>
99     <div class="box box3">Kotak 3</div>
100   </div>
101 </body>
102 </html>
```

b. Output





2. Link Github

<https://github.com/MuhHanifAlfaqih/Modul2>

# **Modul 3. Bootstrap**

## **A. Tujuan**

Setelah mengikuti praktikum, Mahasiswa diharapkan dapat untuk:

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

## **B. Alat dan Bahan**

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

1. Modul Pemrograman Web
2. Laptop
3. Aplikasi Visual Studio Code

## **C. Landasan Teori**

### **1. Apa itu Bootstrap Framework?**

Merupakan kerangka kerja CSS yang bersifat open source dan digunakan untuk kebutuhan pembuatan tampilan desain visual dari aplikasi web atau situs website. Kerangka kerja yang digunakan berbentuk template desain berbasis HTML dan CSS untuk kebutuhan pengembangan navigasi, tombol, tipografi, formulir, dan komponen antarmuka yang lainnya.

Selain itu, Bootstrap juga memiliki fitur yang mencakup library dari JavaScript. Untuk penggunaan dari framework ini digunakan untuk membantu dalam menyusun program aplikasi pada sisi front end (client – side). Untuk sekarang, Bootstrap sangat diminati oleh berbagai pengembang web melalui platform Github untuk membantu proses pembuatan desain aplikasi atau website yang lebih komprehensif dan modern.

### **2. Sejarah Perkembangan Bootstrap**

Bootstrap terus mengalami perbaikan dan pembaharuan sistem untuk memberikan layanan atau fitur yang lebih kompleks. Sehingga, dapat membantu mengatasi permasalahan para *developer* serta mempermudah proses pengembangan produk dari sisi *coding*. Berikut ini merupakan

beberapa perubahan yang dialami oleh kerangka kerja ini dari waktu ke waktu.

1) Versi Pertama

Awal mula terbentuknya Bootstrap dinamakan Twitter Blueprint yang dikembangkan oleh Mark Otto dan Jacob Thornton sebagai kerangka kerja untuk mendorong sistem yang lebih konsistensi pada perangkat internal yang sesuai. Pada tanggal 19 Agustus 2011, resmi diperkenalkan sebagai proyek yang bersifat *open source* (sumber terbuka).

2) Versi Kedua & Ketiga

Pada tanggal 31 Januari 2012, Bootstrap 2 resmi dirilis dengan penambahan fitur terkait sistem *layout* dengan menggunakan 12 kolom grid yang responsif. Kemudian, mendapat dukungan juga dari Glyphicons, serta beberapa komponen baru.

Berikutnya, pada tanggal 19 Agustus 2013, Bootstrap merilis *update version* yang ke – 3 dengan penambahan fitur dari sisi *flat design* dan tampilan *mobile – first* yang lebih responsif.

3) Versi Keempat

Versi stabil yang dapat diakses menggunakan Bootstrap 4 pada tanggal 18 Januari 2018 dengan terbanyak banyak fitur yang mengalami perubahan dari versi sebelumnya. Komponen yang mengalami perubahan diantaranya adalah peralihan Less menjadi Sass, perubahan kustomisasi pada navigasi, perubahan dari piksel menjadi em, dan masih banyak lagi komponen yang lainnya.

4) Versi kelima

Untuk versi Bootstrap 5 merupakan perubahan terbaru saat ini, dimana terdapat beberapa tambahan fitur secara cukup signifikan. Diantaranya adalah menghapus dukungan jQuery dan diganti dengan Vanilla JavaScript, migrasi perangkat lunak dokumentasi dari Jekyll ke Hugo, dan fitur yang lainnya.

### 3. Kelebihan dari Bootstrap

Setelah mengetahui sejarah singkat dari perkembangannya, selanjutnya kita akan masuk pada pembahasan mengenai keunggulan dari *framework* tersebut.

#### 1) Fleksibel

Kelebihan yang pertama dari penggunaan Bootstrap adalah bersifat fleksibel, dimana dalam setiap pengembangannya dapat dilakukan dengan mudah oleh *developer* menggunakan sebuah *frame*. Sehingga, dalam membangun sebuah website menjadi lebih mudah, cepat, dan efisien.

#### 2) Mudah Digunakan

Kedua, kemudahan struktur dari penggunaan kerangka kerja membuat proses pengembangan tampilan website atau *web app* menjadi lebih mudah, dan *maintenance* dapat dilakukan dengan lebih terstruktur.

#### 3) Desain yang Responsif

Desain yang responsif merupakan salah satu keunggulan dari penggunaan *framework* ini. Dimana, saat ini kebutuhan akan aplikasi berbasis web yang dapat diakses di berbagai perangkat menjadi prioritas utama untuk kemudahan dari sisi *user experience* (pengalaman pengguna).

#### 4) Struktur Javascript

Kelebihan yang terakhir adalah mendapat dukungan (*support*) dari *library JavaScript* untuk mengembangkan produk aplikasi atau website dengan tampilan interaktif dan menarik.

### 4. Cara Menginstall Bootstrap

Bagaimana cara untuk menginstall Bootstrap framework? Terdapat dua cara untuk dapat mengintegrasikannya ke dalam format kode program kerangka HTML dan CSS, yaitu dengan cara manual atau online. Untuk kali ini, kami akan memberikan tips untuk instalasi secara online saja.

Untuk proses instalasi secara online menggunakan format berbasis CDN (Content Delivery Network), sehingga proses integrasi menjadi lebih praktis dan mudah. Namun, kekurangan dari pemanfaatan CDN ini adalah

dapat mengurangi tingkat kecepatan dari website. Berikut ini merupakan script untuk memanggil fungsi Bootstrap menggunakan CDN.

```
<script  
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"  
></script>
```

Sehingga, perintah atau instruksi di atas akan memanggil file Bootstrap dan anda dapat menyalin script tersebut ke dalam kode program anda agar dapat digunakan untuk kebutuhan pemanggilan class atau identifier yang lainnya.

## 5. Cara Menggunakan Bootstrap

Penggunaan dari kerangka kerja framework ini terbilang cukup mudah dan tidak memerlukan penulisan rangkaian kode yang panjang. Sehingga, sangat memudahkan aktivitas kerja dari setiap web developer untuk mengembangkan website yang lebih optimal dari sisi front end.

Berbagai fitur dan metode telah ditambahkan pada versi terbaru, sehingga menambah fleksibilitas dari Bootstrap itu sendiri. Berikut merupakan beberapa fitur atau elemen yang dapat dikustomisasi pada Bootstrap.

### 1) Pembuatan Tabel

Bootstrap memberikan berbagai kemudahan dalam mengembangkan struktur tabel dengan penulisan kode program yang lebih singkat, jelas, dan mudah dipahami. Anda hanya perlu menambahkan kode pada atribut class saja. Berikut ini terdapat beberapa contoh penulisannya:

```
<table class="table table-striped"> atau <table class="table table-hover"> atau <table class="table table-responsive">
```

### 2) Penambahan Button

Selanjutnya, kita juga dapat menambahkan button dengan menggunakan berbagai perintah sebagai berikut.

Anda dapat menuliskan perintah pada atribut class, “.btn”, “.btn-primary (untuk tombol berwarna biru)”, “.btn-warning (tombol berwarna kuning)”, “.btn-danger (tombol berwarna merah)”, “.btn-secondary (tombol berwarna abu – abu)”, “.btn-success (tombol berwarna hijau)”, dan masih terdapat jenis tombol yang lainnya.

### 3) Membuat Pesan Notifikasi (Alert)

Untuk pembuatan fitur pesan notifikasi sendiri, anda dapat menuliskan kode yang hampir sama dengan button di atas. Anda hanya perlu untuk merubah perintah “.btn”, menjadi “.alert” saja. Fungsi dari *alert* sendiri adalah untuk menginformasikan dan mengindikasikan bahwa terdapat pesan penting atau proses yang tidak berjalan dengan semestinya.

#### 4) Membuat Tampilan Gambar

Pada Bootstrap, anda juga dapat menambahkan elemen gambar dengan berbagai jenis bentuk. Anda perlu untuk menambahkan perintah dengan contoh penulisan sebagai berikut.

```
<img class="img-responsive"> atau <img class="rounded"> atau  
<img class="img-thumbnail">
```

#### 5) Pembuatan Panel

Untuk pembuatan panel sendiri, proses penulisannya hampir sama dengan fitur di atas, anda hanya perlu untuk merubah struktur penulisan mengikuti perintah, misalnya dari “.btn” menjadi “.panel”. Terdapat beberapa contoh penulisannya sebagai berikut.

```
<div class="panel panel-default"> atau <div class="panel panel-primary">
```

### 6. Kesimpulan

Bootstrap merupakan salah satu *framework* dari CSS untuk mempermudah proses pembuatan aplikasi berbasis web atau website. Anda dapat mulai mencoba dengan menggunakan versi terbaru saat ini, baik versi 4 atau 5 memiliki kelebihan masing – masing. Anda dapat menyesuaikan dengan kebutuhan bisnis atau pengembangan situs anda. [3]

## D. Kegiatan Praktikum

Kegiatan praktikum Modul Bootstrap pada mata kuliah Pemrograman Dasar dilaksanakan pada Kamis, 5 Juni 2025 dan jam 17.00-18.20 WITA di Laboratorium Multimedia STITEK BONTANG, menggunakan Komputer no 5. Adapun Langkah-langkah pengerjaannya sebagai berikut:

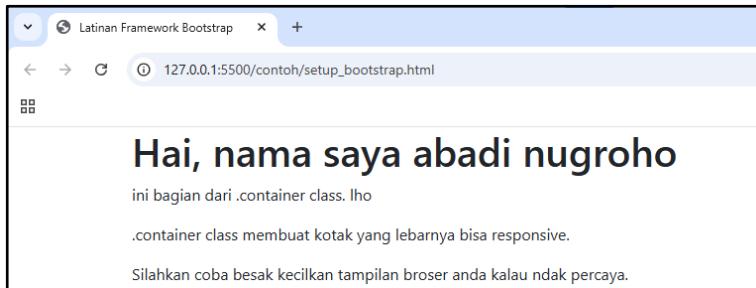
### 1. Instalasi dan Setup Bootstrap

#### 1) Contoh

##### a. Input

```
contoh > setup_bootstrap.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Latinan 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"></script>
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>
```

##### b. Output



##### c. Analisis

Program di atas merupakan sebuah halaman web sederhana yang dibangun menggunakan framework Bootstrap versi 5.3.3. Halaman ini menggunakan struktur dasar HTML5 dengan deklarasi `<!DOCTYPE html>` dan elemen `<html>` yang memiliki atribut `lang="en"` untuk menunjukkan bahwa konten utama dalam bahasa Inggris.

Pada bagian `<head>`, terdapat beberapa elemen meta yang penting, seperti pengaturan karakter set UTF-8 dan viewport untuk memastikan tampilan yang responsif di berbagai perangkat. Framework Bootstrap di-load dari CDN (Content Delivery

Network) jsDelivr, mencakup baik CSS (bootstrap.min.css) maupun JavaScript (bootstrap.bundle.min.js) yang diperlukan untuk fungsi-fungsi Bootstrap.

Konten utama halaman berada dalam <div class="container"> yang merupakan salah fitur fundamental Bootstrap untuk membuat layout yang responsif. Div container ini akan secara otomatis menyesuaikan lebar berdasarkan ukuran layar perangkat pengguna. Di dalam container tersebut, terdapat beberapa elemen teks berupa heading <h1> yang menampilkan "Hai, nama saya abadi nugroho" dan beberapa paragraf <p> yang menjelaskan tentang fungsi container class dalam Bootstrap. Paragraf terakhir bahkan mengajak pengguna untuk mencoba mengubah ukuran browser sebagai demonstrasi sifat responsif dari container tersebut.

## 2. Sistem grid di Bootstrap 5

### 1) Contoh

#### a. Input

```
contoh > o grid_bootstrap.html > 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"></script>
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>
```

#### b. Output

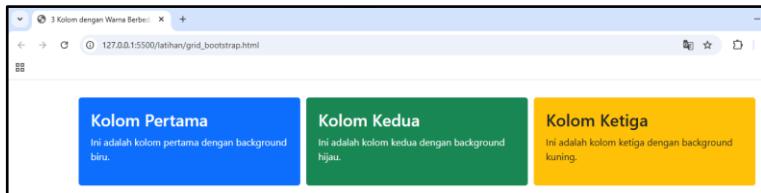


## 2) Latihan

### a. Input

```
latihan > o grid_bootstrap.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>3 Kolom dengan Warna Berbeda</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      <style>
10         .col {
11             padding: 20px;
12             border-radius: 5px;
13             margin: 5px;
14         }
15     </style>
16 </head>
17 <body>
18     <div class="container mt-4">
19         <div class="row">
20             <div class="col bg-primary text-white">
21                 <h3>Kolom Pertama</h3>
22                 <p>Ini adalah kolom pertama dengan background biru.</p>
23             </div>
24             <div class="col bg-success text-white">
25                 <h3>Kolom Kedua</h3>
26                 <p>Ini adalah kolom kedua dengan background hijau.</p>
27             </div>
28             <div class="col bg-warning">
29                 <h3>Kolom Ketiga</h3>
30                 <p>Ini adalah kolom ketiga dengan background kuning.</p>
31             </div>
32         </div>
33     </div>
34 </body>
35 </html>
```

### b. Output



### c. Analisis

Program pertama merupakan implementasi dasar sistem grid Bootstrap dengan tiga kolom sederhana. Program ini hanya menampilkan teks "Kolom 1", "Kolom 2", dan "Kolom 3" tanpa styling tambahan, menggunakan kelas .col yang secara otomatis membagi lebar menjadi tiga bagian sama besar. Tujuannya adalah untuk memperkenalkan struktur grid Bootstrap dalam bentuk yang paling minimalis, tanpa elemen desain yang rumit. Layout-nya sepenuhnya mengandalkan fitur default Bootstrap, sehingga kolom-kolom tampil rapat tanpa padding atau margin khusus.

Program kedua merupakan pengembangan dari konsep grid dasar dengan menambahkan berbagai elemen desain. Selain menggunakan sistem grid yang sama, program ini memperkaya tampilan dengan CSS internal yang menambahkan padding, border-radius, dan margin pada setiap kolom. Setiap kolom diberi latar berbeda menggunakan kelas utilitas Bootstrap (bg-primary, bg-success, bg-warning) dan dilengkapi dengan konten yang lebih

detail berupa judul (`<h3>`) dan paragraf (`<p>`). Program ini tidak hanya mendemonstrasikan fungsi grid, tetapi juga menunjukkan bagaimana memanfaatkan fitur Bootstrap untuk membuat tampilan yang lebih menarik dan siap pakai dalam pengembangan website nyata.

### 3. Typography dan Utilities

#### 1) Contoh

##### a) Input

###### a) Typo

```
kontoh > o type_bootstrap.html > HTML
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta name="viewport" content="width=device-width, initial-scale=1">
5     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3/dist/css/bootstrap.min.css" rel="stylesheet">
6     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3/dist/js/bootstrap.bundle.min.js" rel="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       <ul class="list-unstyled">
13         <li>h1 Bootstrap heading (h1)</li>
14         <li>h2 Bootstrap heading (h2)</li>
15         <li>h3 Bootstrap heading (h3)</li>
16         <li>h4 Bootstrap heading (h4)</li>
17         <li>h5 Bootstrap heading (h5)</li>
18         <li>h6 Bootstrap heading (h6)</li>
19       </ul>
20       <p class="text-lowercase">LOWERCASE TEXT.</p>
21       <p class="text-uppercase">UPPERCASE TEXT.</p>
22       <p class="text-capitalize">CAPITALIZED TEXT.</p>
23       <p>Untuk menulis dengan style standart</p>
24       <p class="text-start">Left-aligned text.</p>
25       <p class="text-end">Right-aligned text.</p>
26       <p class="text-center">Center-aligned text.</p>
27       <p>dengan keyboard atau copy paste</p>
28       <p>Untuk menulis dengan style font "GITTEK" (font program studi Teknik Informatika).</p>
29       <p>Control performatan dengan style code: <code><span></span></code>, <code><section></code>, and <code><div></code></p>
30       <ul class="list-unstyled">
31         <li>Coffee</li>
32         <li>Tea</li>
33           <ul class="list-unstyled">
34             <li>Black tea</li>
35             <li>Green tea</li>
36           </ul>
37         <li>Milk</li>
38       </ul>
39       <blockquote class="blockquote">
40         <p>Ini merupakan contoh performatan dengan style quote.</p>
41         <code><span></span></code> ini adalah tag<code><span></span></code></p>
42       </blockquote>
43     </div>
44   </body>
45 </html>
```

###### b) Text Colors

```
kontoh > o text_colors_bootstrap.html > 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/dist/css/bootstrap.min.css" rel="stylesheet">
8     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3/dist/js/bootstrap.bundle.min.js" rel="script">
9   </head>
10  <body>
11    <div class="container mt-3">
12      <h2>Warna pada font</h2>
13      <p>Use the contextual classes to provide "meaning through colors":</p>
14      <p class="text-muted">This text is muted.</p>
15      <p class="text-primary">This text is important.</p>
16      <p class="text-success">This text indicates success.</p>
17      <p class="text-info">This text represents information.</p>
18      <p class="text-warning">This text represents a warning.</p>
19      <p class="text-danger">This text represents danger.</p>
20      <p class="text-secondary">Secondary text.</p>
21      <p class="text-dark">This text is dark grey.</p>
22      <p class="text-body">Default body color (often black).</p>
23      <p class="text-light bg-dark">This text is light grey (on dark background).</p>
24      <p class="text-white bg-dark">This text is white (on dark background).</p>
25    </div>
26  </body>
27 </html>
```

###### c) Background Colors

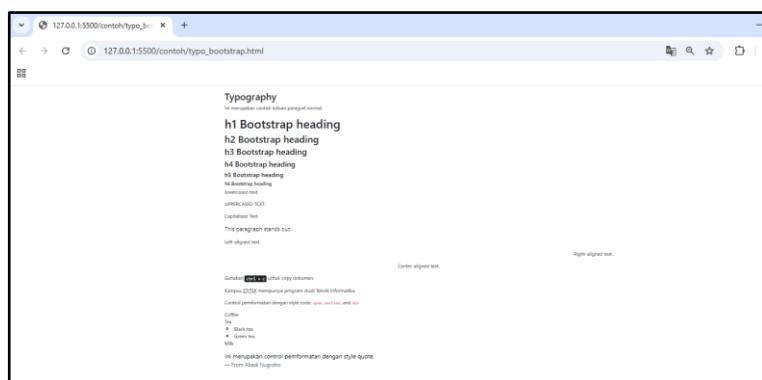
```

contoh > o pg.colors.bootstrap.html > 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         <p>Berbagai macam warna background yang tersedia di Bootstrap:</p>
14
15         <div class="bg-primary p-3 text-white">Primary background</div>
16         <div class="bg-secondary p-3 text-white">Secondary background</div>
17         <div class="bg-info p-3 text-dark">Info background</div>
18         <div class="bg-warning p-3 text-dark">Warning background</div>
19         <div class="bg-danger p-3 text-white">Danger background</div>
20         <div class="bg-secondary p-3 text-white">Secondary background</div>
21         <div class="bg-dark p-3 text-white">Dark background</div>
22         <div class="bg-light p-3 text-dark">Light background</div>
23     </div>
24 </body>
25 </html>

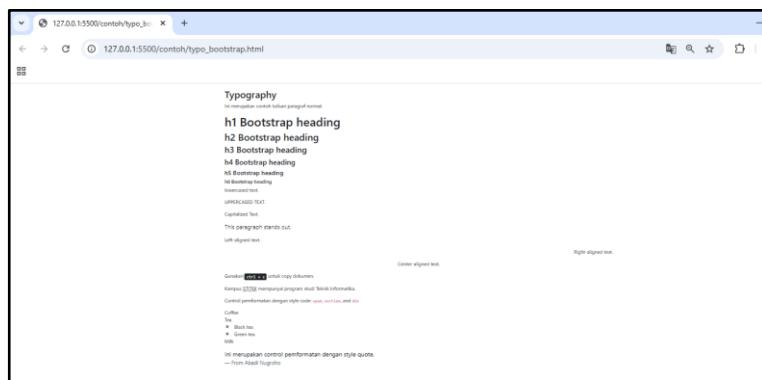
```

## b. Output

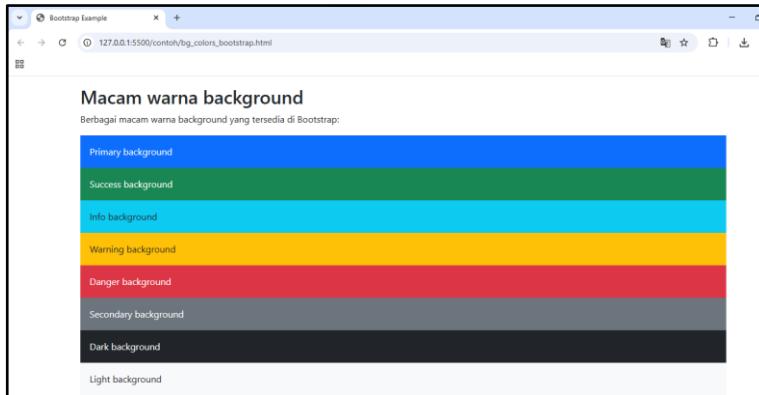
### a) Typo



### b) Text Colors



### c) Background Colors



c. Analisis

a) Typo

Program HTML ini merupakan demonstrasi komprehensif tentang penggunaan typography dalam framework Bootstrap 5.3.3. Dokumen ini diawali dengan struktur dasar HTML yang menyertakan viewport meta tag untuk memastikan tampilan responsif di berbagai perangkat, serta mengimpor CSS dan JavaScript Bootstrap dari CDN jsDelivr. Konten utama ditempatkan dalam container dengan margin top 3 unit (mt-3) untuk memberikan jarak dari bagian atas halaman.

Program ini menampilkan berbagai fitur typography Bootstrap secara sistematis. Dimulai dengan contoh paragraf normal, diikuti oleh enam level heading (h1 hingga h6) yang menunjukkan ukuran font bertingkat sesuai standar Bootstrap. Kemudian diperlihatkan berbagai utility class untuk transformasi teks seperti text-lowercase, text-uppercase, dan text-capitalize yang masing-masing mengubah tampilan huruf menjadi kecil semua, besar semua, atau kapital di awal kata.

Bagian berikutnya menampilkan class khusus untuk penekanan dan perataan teks, termasuk lead untuk paragraf yang menonjol, serta text-start, text-end, dan text-center untuk mengatur perataan teks. Program ini juga memperagakan elemen-elemen khusus seperti kbd untuk menampilkan kombinasi keyboard, abbr untuk singkatan dengan tooltip, dan code untuk menandai kode program.

Fitur lanjutan yang ditunjukkan mencakup list-unstyled untuk menghilangkan bullet point default pada daftar, serta blockquote dengan class khusus untuk menampilkan kutipan yang dilengkapi dengan footer. Setiap contoh dilengkapi dengan teks deskriptif yang menjelaskan fungsinya, membuat program ini tidak hanya berfungsi sebagai demonstrasi tetapi juga sebagai bahan pembelajaran yang informatif. Program ini secara efektif memamerkan berbagai kemampuan typography Bootstrap dalam satu halaman yang terorganisir dengan baik.

b) Text Colors

Program HTML ini merupakan demonstrasi penggunaan kelas kontekstual Bootstrap untuk mengatur warna teks yang memberikan makna visual tertentu. Dokumen ini menggunakan Bootstrap 5.3.3 yang di-load melalui CDN jsDelivr, dengan struktur dasar yang mencakup viewport meta tag untuk memastikan tampilan responsif di berbagai perangkat. Konten utama ditempatkan dalam container dengan margin top 3 unit (mt-3) untuk memberikan jarak yang tepat dari bagian atas halaman.

Program ini secara khusus menampilkan berbagai variasi warna teks yang tersedia dalam Bootstrap melalui kelas-kelas kontekstual. Dimulai dengan teks muted (warna abu-abu redup) yang biasanya digunakan untuk teks sekunder, diikuti oleh warna-warna bermakna seperti primary (biru - untuk informasi penting), success (hijau - menandakan keberhasilan), info (biru muda - informasi netral), warning (kuning - peringatan), dan danger (merah - menandakan bahaya). Program juga menampilkan variasi warna netral seperti secondary (abu-abu), dark (abu-abu gelap), dan body (warna default, biasanya hitam). Yang menarik, program ini juga menunjukkan kombinasi warna teks dan latar belakang, seperti teks light (abu-abu muda) dan white (putih) pada background dark (hitam), yang sangat

berguna untuk situasi dengan kontras tinggi atau tema gelap. Setiap contoh teks disertai label deskriptif yang menjelaskan makna dari warna tersebut, membuat program ini tidak hanya berfungsi sebagai demonstrasi visual tetapi juga sebagai panduan pembelajaran tentang penggunaan warna yang bermakna dalam desain antarmuka.

c) Background Colors

Program HTML ini merupakan demonstrasi penggunaan kelas warna background yang tersedia dalam framework Bootstrap 5.3.3. Dokumen ini diawali dengan struktur dasar HTML yang mencakup pengaturan viewport untuk responsivitas dan mengimpor library Bootstrap dari CDN jsDelivr. Konten utama ditempatkan dalam container dengan margin top 3 unit (mt-3) untuk memberikan jarak yang proporsional dari bagian atas halaman.

Program ini menampilkan delapan variasi warna background yang disediakan Bootstrap, masing-masing diimplementasikan dalam div terpisah. Setiap div menggunakan kombinasi kelas yang mencakup: kelas warna background (bg-primary, bg-success, dll), padding 3 unit (p-3) untuk memberikan ruang internal yang nyaman, dan kelas warna teks yang kontras (text-white atau text-dark) untuk memastikan keterbacaan. Warna-warna yang ditampilkan mencakup primary (biru), success (hijau), info (biru muda), warning (kuning), danger (merah), secondary (abu-abu), dark (hitam), dan light (abu-abu muda), yang masing-masing memiliki makna kontekstual dalam desain antarmuka.

#### 4. Tables

1) Contoh

a. Input

```

contoh@Ubuntu-OptiPlex-5090:~/Desktop$ nano basic_table_bootstrap.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@4.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
8   <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.bundle.min.js"></script>
9 </head>
10  <body>
11    <div class="container mt-4">
12      <h3>Basic Table</h3>
13      <p>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@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>

```

## b. Output

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

## 2) Latihan

### a. Input

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Bootstrap Table Styles</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@4.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
8   <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.bundle.min.js"></script>
9 </head>
10 <body>
11   <div class="container mt-4">
12     <h2>Berbagai Tipe Tabel di Bootstrap</h2>
13     <p>Berikut contoh penggunaan berbagai class tabel Bootstrap:</p>
14
15     <!-- Basic Table -->
16     <h3>1. Tabel Dasar (<table>)</h3>
17     <table class="table">
18       <thead><tr><th>Depan</th><th>Belakang</th><th>Email</th></tr></thead>
19       <tbody>
20         <tr><td>Muh.Hanif</td><td>Alfaqih</td><td>muhhanifihanif@gmail.com</td></tr>
21         <tr><td>Zulkarnaen</td><td>Hasan</td><td>Hasan88@gmail.com</td></tr>
22         <tr><td>Agus</td><td>Septiawan</td><td>agus@gmail.com</td></tr>
23       </tbody>
24     </table>
25
26     <!-- Striped Table -->
27     <h3>2. .table-striped</h3>
28     <table class="table table-striped">
29       <thead><tr><th>Depan</th><th>Belakang</th><th>Email</th></tr></thead>
30       <tbody>
31         <tr><td>Muh.Hanif</td><td>Alfaqih</td><td>muhhanifihanif@gmail.com</td></tr>
32         <tr><td>Zulkarnaen</td><td>Hasan</td><td>Hasan88@gmail.com</td></tr>
33         <tr><td>Agus</td><td>Septiawan</td><td>agus@gmail.com</td></tr>
34       </tbody>
35     </table>
36
37     <!-- Bordered Table -->
38     <h3>3. .table-bordered</h3>
39     <table class="table table-bordered">
40       <thead><tr><th>Depan</th><th>Belakang</th><th>Email</th></tr></thead>
41       <tbody>
42         <tr><td>Muh.Hanif</td><td>Alfaqih</td><td>muhhanifihanif@gmail.com</td></tr>
43         <tr><td>Zulkarnaen</td><td>Hasan</td><td>Hasan88@gmail.com</td></tr>
44         <tr><td>Agus</td><td>Septiawan</td><td>agus@gmail.com</td></tr>
45       </tbody>
46     </table>

```

```

48    <!-- Hover Table -->
49    <h5>4. .table-hover</h5>
50    <table class="table table-hover">
51        <thead><tr><th>Depan</th><th>Belakang</th><th>Email</th></tr></thead>
52        <tbody>
53            <tr><td>Muh.Hanif</td><td>AlFaqih</td><td>muhhanif10hanif@gmail.com</td></tr>
54            <tr><td>Zulkarnaen</td><td>Hasan</td><td>Hasan88@gmail.com</td></tr>
55            <tr><td>Agus</td><td>Septiawan</td><td>agus@gmail.com</td></tr>
56        </tbody>
57    </table>
58
59    <!-- Dark Table -->
60    <h5>5. .table-dark</h5>
61    <table class="table table-dark">
62        <thead><tr><th>Depan</th><th>Belakang</th><th>Email</th></tr></thead>
63        <tbody>
64            <tr><td>Muh.Hanif</td><td>AlFaqih</td><td>muhhanif10hanif@gmail.com</td></tr>
65            <tr><td>Zulkarnaen</td><td>Hasan</td><td>Hasan88@gmail.com</td></tr>
66            <tr><td>Agus</td><td>Septiawan</td><td>agus@gmail.com</td></tr>
67        </tbody>
68    </table>
69
70    <!-- Borderless Table -->
71    <h5>6. .table-borderless</h5>
72    <table class="table table-borderless">
73        <thead><tr><th>Depan</th><th>Belakang</th><th>Email</th></tr></thead>
74        <tbody>
75            <tr><td>Muh.Hanif</td><td>AlFaqih</td><td>muhhanif10hanif@gmail.com</td></tr>
76            <tr><td>Zulkarnaen</td><td>Hasan</td><td>Hasan88@gmail.com</td></tr>
77            <tr><td>Agus</td><td>Septiawan</td><td>agus@gmail.com</td></tr>
78        </tbody>
79    </table>
80
81    <!-- Small Table -->
82    <h5>7. .table-sm</h5>
83    <table class="table table-sm table-bordered">
84        <thead><tr><th>Depan</th><th>Belakang</th><th>Email</th></tr></thead>
85        <tbody>
86            <tr><td>Muh.Hanif</td><td>AlFaqih</td><td>muhhanif10hanif@gmail.com</td></tr>
87            <tr><td>Zulkarnaen</td><td>Hasan</td><td>Hasan88@gmail.com</td></tr>
88            <tr><td>Agus</td><td>Septiawan</td><td>agus@gmail.com</td></tr>
89        </tbody>
90    </table>
91
92    <!-- Responsive Table -->
93    <h5>8. .table-responsive</h5>
94    <div class="table-responsive">
95        <table class="table table-bordered">
96            <thead><tr><th>Depan</th><th>Belakang</th><th>Email</th></tr></thead>
97            <tbody>
98                <tr><td>Muh.Hanif</td><td>AlFaqih</td><td>muhhanif10hanif@gmail.com</td></tr>
99                <tr><td>Zulkarnaen</td><td>Hasan</td><td>Hasan88@gmail.com</td></tr>
100               <tr><td>Agus</td><td>Septiawan</td><td>agus@gmail.com</td></tr>
101            </tbody>
102        </table>
103    </div>
104
105 </body>
106 </html>

```

## b. Output

The screenshot shows a web page titled "Berbagai Tipe Tabel di Bootstrap". It displays four examples of Bootstrap tables:

- 1. Tabel Dasar (.table)**: A simple table with three columns: Depan, Belakang, and Email. The data rows are Muh.Hanif (AlFaqih, muhhanif10hanif@gmail.com), Zulkarnaen (Hasan, Hasan88@gmail.com), and Agus (Septiawan, agus@gmail.com).
- 2. table-striped**: A table with alternating row colors (light gray and white). The data rows are the same as the first example.
- 3. .table-bordered**: A table with a border around each cell. The data rows are the same as the first example.
- 4. .table-hover**: A table where hovering over a row changes its background color to light gray. The data rows are the same as the first example.

5. .table-dark		
Depan	Belakang	Email
Muh.Hanif	Alfaqih	muhhanif10hanif@gmail.com
Zulkarnaen	Hasan	Hasan88@gmail.com
Agus	Septiawan	agus@gmail.com

6. .table-borderless		
Depan	Belakang	Email
Muh.Hanif	Alfaqih	muhhanif10hanif@gmail.com
Zulkarnaen	Hasan	Hasan88@gmail.com
Agus	Septiawan	agus@gmail.com

7. .table-sm		
Depan	Belakang	Email
Muh.Hanif	Alfaqih	muhhanif10hanif@gmail.com
Zulkarnaen	Hasan	Hasan88@gmail.com
Agus	Septiawan	agus@gmail.com

8. .table-responsive		
Depan	Belakang	Email
Muh.Hanif	Alfaqih	muhhanif10hanif@gmail.com
Zulkarnaen	Hasan	Hasan88@gmail.com
Agus	Septiawan	agus@gmail.com

### c. Analisis

Kedua program HTML tersebut sama-sama mendemonstrasikan penggunaan tabel dalam Bootstrap, namun memiliki perbedaan signifikan dalam hal cakupan dan kompleksitas fitur yang ditampilkan. Program pertama hanya menampilkan contoh tabel dasar dengan struktur minimalis, menggunakan kelas .table saja tanpa variasi styling tambahan. Tabel ini menampilkan data sederhana dengan tiga kolom (Nama Depan, Nama Belakang, Email) dan tiga baris data, cocok untuk pemula yang ingin memahami struktur dasar tabel Bootstrap.

Program kedua jauh lebih komprehensif, memperlihatkan delapan variasi tabel Bootstrap yang berbeda dengan contoh penggunaan berbagai kelas utilitas. Mulai dari .table-striped untuk efek zebra-stripe, .table-bordered dengan garis tepi, .table-hover dengan efek hover pada baris, hingga .table-dark dengan tema gelap. Program ini juga menunjukkan fitur-fitur lanjutan seperti .table-sm untuk tabel kompak dan .table-responsive untuk tampilan yang adaptif di perangkat mobile. Setiap variasi tabel diberi judul penjelasan dan menggunakan data konsisten, memudahkan pemahaman perbedaan masing-masing kelas.

Perbedaan utama terletak pada tujuan penyajiannya. Program pertama bersifat pengenalan dasar, sementara program kedua

berfungsi sebagai referensi lengkap berbagai opsi styling tabel Bootstrap. Keduanya menggunakan struktur HTML serupa dengan container dan impor Bootstrap dari CDN yang sama, tetapi program kedua memiliki nilai edukasi lebih tinggi dengan menunjukkan berbagai kemungkinan customisasi tabel dalam satu halaman. Program pertama cocok untuk pembelajaran awal, sedangkan program kedua lebih bermanfaat sebagai panduan praktis untuk pengembangan antarmuka yang lebih kompleks.

## 5. Komponen Bootstrap 5

### 1) Contoh

#### a. Input

##### a) Button

```
contoh > buttons_bootstrap.html > 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>
```

##### b) Navbars

```
contoh > navbar_looking.html > 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)">Profile</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 href="#">A first link</a></li>
32                             <li><a href="#">Another link</a></li>
33                             <li><a 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 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>
```

### c) Modal

```
contoh > ⌂ modal_bootstrap.html ⌂ 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
19     <!-- The Modal -->
20     <div class="modal fade" id="myModal">
21         <div class="modal-dialog">
22             <div class="modal-content">
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             </div>
39         </div>
40     </div>
41 </body>
42 </html>]
```

### d) Pagination

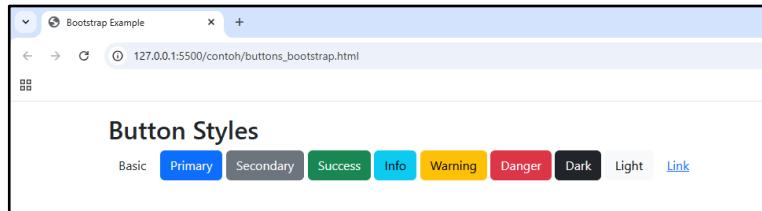
```
contoh > ⌂ pagination_bootstrap.html ⌂ html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Bootstrap pagination</title>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" 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         <h3>Pagination</h3>
13         <p>For 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 </body>
23 </html>]
```

### e) Card

```
contoh > ⌂ card_bootstrap.html ⌂ html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Bootstrap Examples</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>]
```

## b. Output

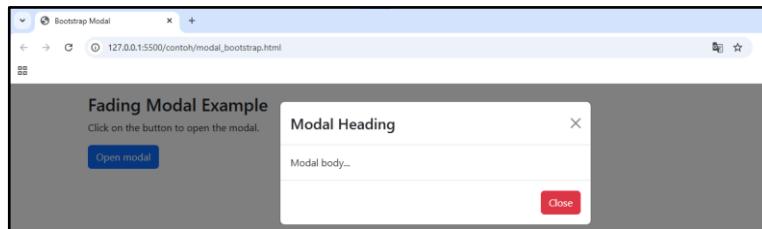
### a) Button



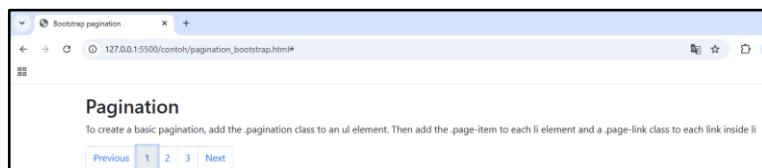
### b) Navbars



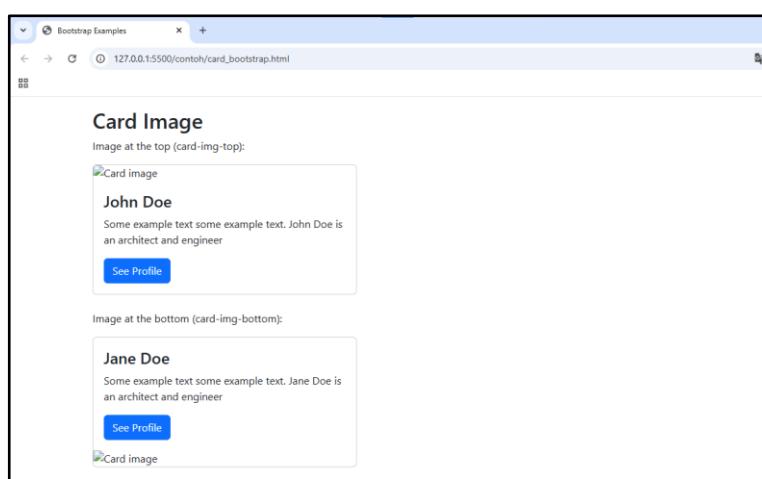
### c) Modal



### d) Pagination



### e) Card

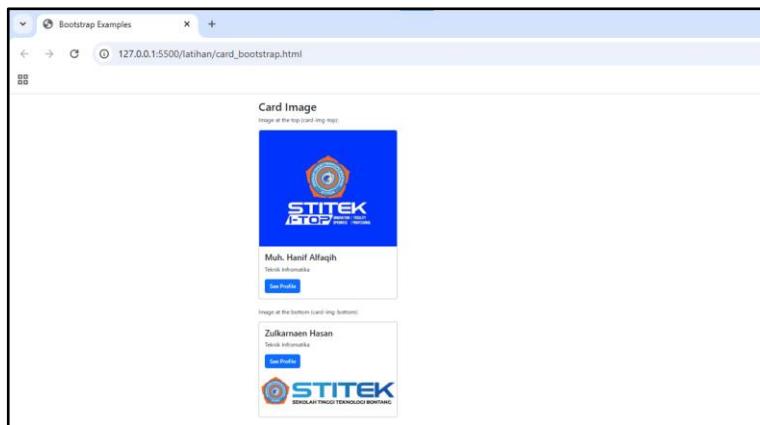


## 2) Latihan

### a. Input

```
latihan > card_bootstrap.html > HTML > body > div.container.mt-3 > div.card
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Bootstrap Examples</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                 <h3>Card title</h3>
18                 <h4>Muh. Hanif Alfaqih</h4>
19                 <p class="card-text">Teknik Informatika</p>
20                 <a href="#" class="btn btn-primary" href="#">See Profile</a>
21             </div>
22         </div>
23         <br>
24         <p>Image at the bottom (card-img-bottom):</p>
25         <div class="card" style="width:400px">
26             <div class="card-body">
27                 <h3>Card title</h3>
28                 <h4>Zulkarnaen Hasan</h4>
29                 <p class="card-text">Teknik Informatika</p>
30                 <a href="#" class="btn btn-primary" href="#">See Profile</a>
31             </div>
32         </div>
33     </div>
34 </body>
35 </html>
```

### b. Output



### c. Analisis

#### a) Button

Program HTML ini merupakan demonstrasi penggunaan berbagai gaya tombol (button styles) yang tersedia dalam framework Bootstrap 5.3.3. Dokumen ini menggunakan struktur dasar HTML5 yang dilengkapi dengan viewport meta tag untuk memastikan tampilan responsif di berbagai perangkat, serta mengimpor library Bootstrap dari CDN jsDelivr. Konten utama ditempatkan dalam container dengan margin top 3 unit (mt-3) untuk memberikan jarak yang proporsional dari bagian atas halaman.

Program ini menampilkan sebelas variasi gaya tombol Bootstrap yang berbeda, dimulai dari tombol dasar (basic) tanpa kelas

styling khusus, diikuti oleh tombol-tombol dengan kelas kontekstual yang memberikan makna visual tertentu. Setiap tombol menggunakan kelas btn sebagai dasar, ditambah dengan kelas modifier seperti btn-primary (biru), btn-secondary (abu-abu), btn-success (hijau), btn-info (biru muda), btn-warning (kuning), btn-danger (merah), btn-dark (hitam), dan btn-light (abu-abu muda). Terdapat juga tombol dengan gaya link (btn-link) yang menyerupai tautan teks.

b) Navbars

Program HTML ini mendemonstrasikan cara membuat navigation bar (navbar) responsif menggunakan Bootstrap 5.3.3 dengan berbagai fitur modern. Navbar ini dirancang dengan tema gelap menggunakan kelas navbar-dark bg-dark dan akan menyesuaikan tampilannya di berbagai ukuran layar berkat kelas navbar-expand-sm. Pada tampilan mobile, navbar akan mencuat dan menampilkan tombol toggle (navbar-toggler) dengan ikon khusus Bootstrap yang ketika diklik akan memperlihatkan menu navigasi.

Struktur navbar terdiri dari beberapa komponen penting: logo/merk di sebelah kiri menggunakan navbar-brand, menu navigasi utama di tengah yang berisi item biasa dan dropdown menu, serta form pencarian di sebelah kanan. Menu dropdown diimplementasikan dengan sempurna menggunakan komponen dropdown Bootstrap yang muncul saat hover/klik. Form pencarian menggunakan utility class Bootstrap seperti form-control untuk input field dan btn btn-primary untuk tombol search, menunjukkan integrasi yang baik antara komponen navbar dengan elemen form.

c) Modal

Program HTML ini mendemonstrasikan implementasi modal dialog menggunakan Bootstrap 5.3.3 dengan efek fade animasi. Modal merupakan komponen UI yang muncul di atas halaman

utama untuk menampilkan konten tambahan atau interaksi khusus. Kode ini terdiri dari dua bagian utama: tombol trigger untuk membuka modal dan struktur modal itu sendiri.

Struktur modal dibangun dengan elemen <div> utama yang memiliki kelas modal fade, dimana kelas fade memberikan efek animasi halus saat modal muncul dan menghilang. Modal ini menggunakan layout standar Bootstrap yang terdiri dari tiga bagian: header (dengan judul dan tombol close), body (untuk konten utama), dan footer (berisi tombol aksi). Setiap elemen modal menggunakan kelas khusus Bootstrap seperti modal-dialog untuk wadah utama dan modal-content untuk area konten. Mekanisme kerja modal diaktifkan melalui atribut HTML data-bs-toggle="modal" dan data-bs-target="#myModal" pada tombol trigger. Ketika tombol diklik, modal akan muncul dengan efek fade-in, dan bisa ditutup baik melalui tombol close (ikon X di header) maupun tombol "Close" di footer.

d) Pagination

Program HTML ini merupakan contoh implementasi komponen pagination (penomoran halaman) menggunakan framework Bootstrap 5.3.3. Pagination ini dibangun dengan struktur dasar berupa unordered list (<ul>) yang diberi class pagination, dimana setiap item list (<li>) menggunakan class page-item dan link (<a>) di dalamnya menggunakan class page-link.

Program ini menampilkan pagination sederhana yang terdiri dari lima elemen: tombol 'Previous', tiga nomor halaman (1, 2, 3), dan tombol 'Next'. Desainnya mengikuti standar Bootstrap dengan tampilan yang clean dan responsif, dimana setiap item pagination memiliki styling konsisten termasuk efek hover dan active state. Penjelasan teks dalam paragraf di atas komponen pagination memberikan petunjuk jelas tentang struktur class yang diperlukan untuk membuat pagination di Bootstrap.

Implementasi ini sangat berguna untuk navigasi di antara berbagai halaman konten, seperti artikel blog atau hasil pencarian. Meskipun link pada contoh ini masih menggunakan href="#" (tidak mengarah ke halaman tertentu), dalam penerapan nyata dapat dihubungkan dengan sistem pagination backend.

e) Card

Kedua program HTML tersebut sama-sama mendemonstrasikan penggunaan komponen card dengan gambar dalam Bootstrap 5.3.3, namun memiliki perbedaan signifikan dalam konten dan konteks penggunaannya. Program pertama menggunakan contoh generik dengan nama "John Doe" dan "Jane Doe" sebagai placeholder, sementara program kedua menampilkan data yang lebih spesifik terkait mahasiswa Teknik Informatika dengan nama "Muh. Hanif Alfaqih" dan "Zulkarnaen Hasan".

Perbedaan utama terletak pada sumber gambar dan informasi yang ditampilkan. Program pertama menggunakan gambar placeholder ("gantl\_foto\_kalian.png" dan "gmt1\_foto\_kalian.png") dengan deskripsi profesi arsitek dan engineer, sedangkan program kedua menggunakan gambar yang lebih kontekstual ("stitek.jpeg" dan "stitek2.png") dengan informasi program studi Teknik Informatika. Meskipun struktur HTML dan kelas Bootstrap yang digunakan identik (seperti card-img-top, card-body, dan btn-primary), program kedua menyesuaikan kontennya untuk keperluan yang lebih spesifik dalam lingkungan akademik.

Dari segi fungsi, kedua program sama-sama menunjukkan cara menampilkan gambar di bagian atas (card-img-top) dan bawah (card-img-bottom) card Bootstrap, namun program kedua memberikan contoh yang lebih realistik dan siap pakai untuk keperluan pembuatan profil mahasiswa atau staff di lingkungan kampus. Keduanya tetap mempertahankan fitur responsif dan

styling konsisten khas Bootstrap, dengan lebar card yang ditetapkan 400px melalui inline style.

## E. Tugas

1. Buatlah sebuah halaman web company profile

a. Input

```
tugas > o tugas3.html > html > body > div.container.mt-5 > div.row > div.col-md-4.mb-4 > div.card.h-100 > div.card-body > ul.text-start.ps-3 > li
1   <!DOCTYPE html>
2   <html lang="id">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Company Profile - PT Maju Jaya</title>
7       <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8       <style>
9         .card:hover {
10           transform: scale(1.03);
11           transition: 0.3s;
12         }
13         footer {
14           background-color: #343a40;
15           color: white;
16           padding: 20px 0;
17         }
18       </style>
19     </head>
20   <body>
21
22   <!-- Navbar -->
23   <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
24     <div class="container-fluid">
25       <a class="navbar-brand" href="#">PT Maju Jaya</a>
26       <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
27         <span class="navbar-toggler-icon"></span>
28       </button>
29       <div class="collapse navbar-collapse" id="navbarNav">
30         <ul class="navbar-nav ms-auto">
31           <li class="nav-item"><a class="nav-link active" href="#">Home</a></li>
32           <li class="nav-item"><a class="nav-link" href="#">About</a></li>
33           <li class="nav-item"><a class="nav-link" href="#">Services</a></li>
34           <li class="nav-item"><a class="nav-link" href="#">Contact</a></li>
35         </ul>
36       </div>
37     </div>
38   </nav>
```

```
<!-- Konten Utama -->
<div class="container mt-5">
  <!-- Tabel Informasi Perusahaan -->
  <h2 class="text-center mb-4">Informasi Perusahaan</h2>
  <div class="table-responsive">
    <table class="table table-bordered">
      <tr><th>Nama Perusahaan</th><td>PT Surya Mandiri (Agen Resmi PELNI)</td></tr>
      <tr><th>Tahun Berdiri</th><td>2010</td></tr>
      <tr><th>Status</th><td>Agen Resmi Terdaftar PELNI</td></tr>
      <tr><th>Kantor Pusat</th><td>Jl. Pelabuhan No. 15, Tanjung Priok, Jakarta Utara</td></tr>
    </table>
  </div>
```

```
53   <!-- Card Produk / Layanan -->
54   <h2 class="text-center mt-5 mb-4">Produk & Layanan Kami</h2>
55   <div class="row">
56     <div class="col-md-4 mb-4">
57       <div class="card h-100">
58         
59         <div class="card-body">
60           <h5 class="card-title">Pemesanan Tiket</h5>
61           <p class="card-text">Layanan pemesanan tiket kapal PELNI semua kelas dengan proses cepat dan mudah.</p>
62           <ul class="text-start ps-3">
63             <li>Tiket Ekonomi</li>
64             <li>Tiket Bisnis</li>
65             <li>Tiket VIP</li>
66             <li>Tiket Diskon Khusus</li>
67           </ul>
68           <a href="#" class="btn btn-primary">Selengkapnya</a>
69         </div>
70       </div>
71     </div>
```

```

72   <div class="col-md-4 mb-4">
73     <div class="card h-100">
74       
75       <div class="card-body">
76         <h5 class="card-title">Layanan Konsumen</h5>
77         <p class="card-text">Dukungan penuh untuk kebutuhan perjalanan laut Anda.</p>
78         <ul class="text-start ps-3">
79           <li>Hotline 24 Jam</li>
80           <li>Bantuan Tiket Darurat</li>
81           <li>Informasi Jadwal</li>
82           <li>Penanganan Keluhan</li>
83         </ul>
84         <a href="#" class="btn btn-primary">Selengkapnya</a>
85     </div>
86   </div>
87   <div class="col-md-4 mb-4">
88     <div class="card h-100">
89       
90       <div class="card-body">
91         <h5 class="card-title">Pembayaran</h5>
92         <p class="card-text">Berbagai metode pembayaran yang aman dan terpercaya.</p>
93         <ul class="text-start ps-3">
94           <li>Transfer Bank</li>
95           <li>Virtual Account</li>
96           <li>E-Wallet</li>
97           <li>Bayar di Tempat</li>
98         </ul>
99         <a href="#" class="btn btn-primary">Selengkapnya</a>
100    </div>
101  </div>
102 </div>
103 </div>
104 </div>
105 </div>

107 <!-- Footer -->
108 <footer class="text-center mt-5">
109   <div class="container">
110     <p>© 2025 PT Maju Jaya. All rights reserved.</p>
111   </div>
112   <div class="text-center">
113     Kontak: info@majujaya.co.id | 
114     <a href="#" class="text-light">Facebook</a> | 
115     <a href="#" class="text-light">Instagram</a> | 
116     <a href="#" class="text-light">LinkedIn</a>
117   </div>
118 </footer>
119
120 <!-- Script Bootstrap -->
121 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
122 </body>
123 </html>

```

## b. Output

The screenshot displays the PT Maju Jaya website. At the top, there's a navigation bar with links for Home, About, Services, and Contact. Below the header, there are two main sections: 'Informasi Perusahaan' (Company Information) and 'Produk & Layanan Kami' (Products & Services). The 'Informasi Perusahaan' section contains a table with details about the company: Nama Perusahaan (PT Surya Mandiri (Agen Resmi PELNI)), Tahun Berdiri (2010), Status (Agen Resmi Terdaftar PELNI), and Kantor Pusat (Jl. Pelabuhan No. 15, Tanjung Priok, Jakarta Utara). The 'Produk & Layanan Kami' section features three cards: 'TICKET' (with a ticket icon), 'Layanan Konsumen' (with a phone icon), and 'Pembayaran' (with a credit card icon). Each card has a brief description and a 'Selengkapnya' button. At the bottom of the page is a dark footer bar containing copyright information and social media links.

## 2. Link Github

<https://github.com/MuhHanifAlfaqih/Modul3>

## **Modul 4. Javascript**

### **A. Tujuan**

Setelah mengikuti praktikum, Mahasiswa diharapkan dapat untuk:

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. Alat dan Bahan**

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

1. Modul Pemrograman Web
2. Laptop
3. Aplikasi Visual Studio Code

### **C. Landasan Teori**

#### 1) Apa Itu JavaScript dan Sejarahnya

JavaScript adalah bahasa skrip yang digunakan untuk membuat konten halaman web dinamis, berfungsi untuk membuat elemen yang mampu meningkatkan interaksi pengunjung seperti menu drop-down, animasi, dan warna background dinamis.

Sejarah JavaScript dimulai pada tahun 1995, diciptakan oleh Brendan Eich di Netscape Communications. Awalnya, JavaScript hanya digunakan untuk keperluan internal dengan web browser perusahaan tersebut, Netscape Navigator.

Dulunya, JavaScript bernama LiveScript, yang kemudian diubah menjadi JavaScript agar bisa menjadi “teman” bagi bahasa pemrograman Java milik

mitra mereka, Sun Microsystems. Kami akan menjelaskan perbedaan Java dan JavaScript nanti.

JavaScript terus berkembang bersama browser web baru seperti Mozilla Firefox dan Google Chrome sejak saat itu. Bahkan, saat ini sedang dikembangkan mesin JavaScript modern pertama, yaitu V8, yang bertugas untuk mengompilasi bytecode menjadi kode mesin asli.

## 2) Apa Fungsi JavaScript?

Fungsi JavaScript yang paling umum adalah untuk pengembangan aplikasi web dan mobile, membangun web server dan aplikasi server, membuat website yang interaktif, serta game development.

Bahasa pemrograman JavaScript sekarang memiliki banyak framework dan library yang bisa membantu mempercepat proses, seperti AngularJS, jQuery, and ReactJS.

Meskipun umumnya melayani program berbasis web, fitur pemrograman JavaScript memiliki implementasi lain di area yang berbeda. Untuk lebih jelasnya, berikut kegunaan JavaScript:

### 1) Efisiensi Pengembangan Aplikasi Web dan Seluler

Pengembangan framework JavaScript, yang terdiri dari library kode JavaScript, memungkinkan developer menggunakan kode JS siap pakai dalam proyek mereka. Proses yang harus dilalui pun menjadi lebih cepat dan efisien karena mereka tidak perlu menulis kode dari nol.

Setiap framework JavaScript memiliki fitur yang berfungsi untuk menyederhanakan proses development dan debugging.

Misalnya, framework front-end seperti jQuery dan ReactJS bisa membantu meningkatkan efisiensi desain. Developer bisa menggunakan ulang dan memperbarui komponen kode tanpa memengaruhi satu sama lain, baik dari segi fungsi maupun nilai.

Sementara itu, framework pengembangan aplikasi mobile seperti Cordova dan Titanium bisa digunakan untuk membuat aplikasi native atau hybrid.

Implementasi kode JS di Node.js juga berperan penting dalam web development. Node.js bisa mengurangi waktu respons server karena sifatnya yang single-threaded dan memiliki arsitektur non-blocking.

## 2) Membangun Web Server dan Aplikasi Server

Melalui Node.js, JavaScript memungkinkan developer membuat web server dan infrastruktur back-end sehingga akan menghemat waktu dan tenaga dalam hal pembuatan web server.

Modul HTTP built-in pada bahasa pemrograman ini memungkinkan Anda mengembangkan server HTTP basic yang menampilkan teks biasa saat pengunjung mengakses halaman web.

Anda bisa menggunakan web server Node.js sendiri, Node-OS, atau milik pihak luar seperti Microsoft Internet Information Services (IIS) dan Apache untuk menangani permintaan HTTP.

Node-OS bekerja paling baik di sistem operasi Linux karena dikembangkan berdasarkan kernel Linux.

## 3) Membuat Website yang Interaktif

Fungsi JavaScript berikutnya adalah untuk menciptakan halaman web yang dinamis, seperti untuk menampilkan animasi, mengubah visibilitas teks, dan membuat menu drop-down.

Meskipun Anda bisa menggunakan HTML dan CSS saja untuk membuat website, halaman yang dihasilkan hanya akan memiliki tampilan statis tanpa adanya JavaScript. Bahasa pemrograman ini memungkinkan pengunjung berinteraksi dengan halaman web, dan Anda pun bisa menyajikan user experience yang lebih baik.

Selain itu, JavaScript memungkinkan Anda mengubah konten dan nilai atribut HTML tanpa harus merefresh halaman web dulu. Sebab, JavaScript mendukung tipe data berikut:

- a) String – terdiri dari data teks yang ditulis di dalam tanda kutip. Misalnya, “Hello World”, ‘Hello World’, dan “Display ‘Hello world’ text“.
- b) Number – mencakup bilangan bulat dan floating-point antara  $(2^{53} - 1)$  dan  $-(2^{53} - 1)$ .

- c) Boolean – tipe data logis dengan nilai true dan false.
- d) BigInt – merepresentasikan data bilangan bulat (integer) dengan panjang arbitrer.
- e) Null – berisi nilai nol.
- f) Undefined – termasuk variabel yang dideklarasikan, tapi tidak ditetapkan.
- g) Symbol – menyediakan identifier unik untuk objek.
- h) Object – untuk struktur data kompleks yang ditulis dengan kurung kurawal. Misalnya, {item:"Buku", information:"biografi"}.

Tipe data primitif, yang terdiri dari semua tipe data kecuali object, hanya bisa menyimpan satu data. Sementara itu, tipe data object bisa berisi sekumpulan nilai.

Dengan JavaScript, Anda juga bisa meningkatkan pengalaman browsing pengunjung menggunakan cookie. Untuk membuat, membaca, dan menghapus cookie dalam JavaScript, diperlukan properti `document.cookie` yang berfungsi sebagai “getter and setter” nilai cookie.

#### 4) Memudahkan Pengembangan Game

JavaScript juga bisa membantu Anda membuat game kalau digunakan dengan HTML5 dan Application Programming Interface (API) seperti WebGL. Ada banyak game engine berbasis JavaScript seperti Phaser, GDevelop, dan Kiwi.js yang tersedia untuk rendering grafis, daur ulang kode, dan aplikasi lintas platform.

Beberapa contoh game yang dibuat dengan JavaScript antara lain adalah Angry Birds, The Wizard, dan 2048.

#### 5) Perbedaan Java dan JavaScript

Setelah memahami apa itu JavaScript beserta fungsinya, di bagian ini kami akan membahas perbedaannya dengan Java. Sebab, walaupun namanya mirip dan sama-sama digunakan dalam konteks pemrograman, keduanya tidaklah sama.

Menurut informasi dari website resmi Java, bahasa pemrograman JavaScript yang dikembangkan oleh Netscape, Inc. bukanlah bagian dari

platform Java yang dikembangkan oleh Sun Microsystems (sebelum diakuisisi Oracle).

Perbedaan Java dan JavaScript adalah, Java merupakan bahasa pemrograman, sedangkan JavaScript merupakan skrip pemrograman. Kode JavaScript ditulis dalam teks dan bisa langsung diinterpretasikan browser, sedangkan Java harus di-compile menjadi bytecode yang bisa dipahami dan dijalankan komputer.

Apa bedanya scripting language dan bahasa pemrograman?

Nah, bahasa scripting termasuk ke dalam bahasa pemrograman, tapi digunakan untuk memanipulasi, menyesuaikan, dan mengotomatiskan apa yang sudah ada di sistem. Sedangkan bahasa pemrograman biasanya digunakan untuk membuat program dari nol.

Kemudian, Java digunakan untuk membuat aplikasi di perangkat atau browser, sedangkan JavaScript umumnya digunakan pada dokumen HTML. Plugin yang digunakan untuk menggunakan Java dan JavaScript pun berbeda.

Untuk lebih jelasnya, ini perbedaan Java dan JavaScript:

Java	Javascript
Java mengharuskan variabel dideklarasikan lebih dulu. Jenis variabel diperiksa saat compile.	JavaScript memiliki sintaksis dan aturan yang lebih santai.
Java adalah bahasa pemrograman “berorientasi” objek.	JavaScript adalah bahasa scripting “berbasis” objek.
Aplikasi bisa berjalan di virtual machine (JVM) atau semua browser.	Sebelumnya, kode hanya bisa dijalankan di browser, sekarang bisa di server melalui Node.js.
Objek berbasis class, tidak bisa untuk membuat program tanpa membuat class.	Objek berbasis prototipe.
Ekstensi filenya adalah “.Java”, menerjemahkan source code	Ekstensinya adalah “.js”, diinterpretasikan tapi tidak di-

menjadi bytecode yang dijalankan oleh JVM.	compile. Setiap browser punya interpreter untuknya.
Merupakan bahasa pemrograman standalone.	Ada di halaman web, terintegrasi dengan konten HTML.
Menggunakan metode berbasis thread untuk konkurensi.	Menggunakan metode berbasis event untuk konkurensi.
Mendukung multithreading.	Tidak mendukung multithreading.

6) Kelebihan dan Kekurangan JavaScript

JavaScript memiliki sejumlah keunggulan yang membuatnya menjadi pilihan yang lebih baik daripada bahasa pemrograman lainnya. Antara lain, berikut kelebihan JavaScript:

- a) Struktur yang sederhana. Strukturnya yang sederhana membuat JavaScript lebih mudah dipelajari dan diterapkan, serta lebih cepat daripada sejumlah bahasa lain. Error juga mudah diidentifikasi dan diperbaiki.
- b) Eksekusi lebih cepat. JavaScript mengeksekusi skrip langsung di browser web tanpa harus koneksi ke server atau menggunakan compiler. Selain itu, sebagian besar browser memungkinkan JavaScript meng-compile kode pada saat eksekusi program.
- c) Fungsi yang serbaguna. JavaScript kompatibel dengan bahasa pemrograman lain seperti PHP, Perl, dan Java. Bahasa ini juga menjadikan data science dan machine learning bisa diakses oleh developer.
- d) Cukup populer dan banyak digunakan. Ada berbagai resource dan forum yang bisa membantu pemula mempelajari bahasa scripting ini.
- e) Mengurangi beban server. karena berjalan di sisi klien, JavaScript bisa mengurangi permintaan yang dikirim ke server. Validasi data bisa dilakukan melalui browser, dan update hanya berlaku pada bagian halaman web tertentu.

- f) Selalu diperbarui dan dikembangkan. Tim developer JavaScript dan ECMA International terus memperbarui serta merancang framework dan library baru sehingga bisa terus relevan.

Namun, JavaScript juga memiliki beberapa batasan yang perlu Anda pertimbangkan. Berikut adalah beberapa kekurangan JavaScript:

- a) Terkadang memiliki masalah kompatibilitas. Beberapa web browser menginterpretasikan kode JS dalam cara yang berbeda, sehingga terkadang tidak konsisten. Jadi, Anda harus menguji skrip JavaScript di semua web browser utama, termasuk versi lamanya agar tidak mengganggu user experience.
- b) Keamanan kurang terjaga. Kode JavaScript yang berjalan secara client-side rawan terkena eksloitasi oleh pengguna yang tidak bertanggung jawab.
- c) Debugging kurang efisien. Meskipun beberapa editor HTML mendukung debugging, fungsinya masih kurang efisien dibanding editor lain. Karena browser tidak menampilkan peringatan error, Anda mungkin akan kesulitan mengidentifikasi masalah.

## 7) Cara Kerja JavaScript pada Website

JavaScript di-embed langsung ke halaman web atau direferensikan melalui file .js terpisah. Saat pengunjung membuka halaman web, browser akan menjalankan skrip beserta kode HTML dan CSS untuk membuat halaman fungsional yang disajikan melalui tab browser.

Skrip kemudian didownload dan diproses di perangkat pengunjung. Yap, seperti yang kami jelaskan tadi, JavaScript adalah bahasa sisi klien. Jadi tidak seperti bahasa server side di mana server memproses skrip sebelum dikirim ke browser.

Saat menemukan blok kode JavaScript, browser web akan memprosesnya dari atas ke bawah. Karena urutan akan diperhitungkan, pastikan Anda mereferensikan objek atau variabel di dalam blok lebih dulu sebelum memodifikasinya. Apabila memiliki variabel tanpa nilai, error undefined akan muncul.

## 8) Cara Menggunakan JavaScript pada HTML

Ada dua cara menggunakan JavaScript pada HTML, yaitu secara internal dan eksternal. JavaScript internal menggunakan tag <script> pada body script HTML untuk mencakup semua kode JavaScript.

Berikut contoh JavaScript internal dengan tag <script>:

```
<html>
<head>
    <title>Inline JavaScript</title>
    <script type="text/javascript">
        alert("Hello World");
    </script>
</head>
<body>
</body>
</html>
```

Anda juga bisa meng-embed skrip pada atribut event HTML, meminta browser untuk mengeksekusinya saat sebuah event dipicu. Jenis skrip ini disebut inline JavaScript (hapus tanda "\*" pada "onclick" untuk menggunakan kode):

```
<button      *onclick*="alert('Click      for      more
details')">Click</button>
```

Di sisi lain, JavaScript eksternal dilakukan dengan menyimpan kode dalam file .js terpisah, lalu memanggilnya kembali di halaman HTML. Developer sering menggunakan metode ini saat mengerjakan proyek besar, karena akan membuat skripnya lebih teratur.

Anda juga bisa menggunakan lagi skrip JavaScript eksternal di beberapa halaman HTML, yang tentunya akan sangat membantu.

Misalnya, menambahkan skrip berikut ke halaman HTML memungkinkan Anda memanggil kembali file JavaScript eksternal bernama script.js:

```
<script src="script.js" defer></script>
```

Ini adalah contoh cara mereferensikan file JavaScript eksternal:

```
<html>
<head>
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Time right now:</title>
</head>
<body>
    <button type="button"
        onclick="myFunction()">Select</button>
</body>
<script src="js/script.js"></script>
</html>
```

Anda bisa membuat dan memodifikasi kode JavaScript menggunakan text editor. Aplikasi text editor yang bisa digunakan juga bermacam-macam, silakan lihat rekomendasi HTML text editor terbaik kami untuk memilih mana yang paling sesuai kebutuhan.

Perlu diingat bahwa ada dua jenis text editor, yaitu WYSIWYG (what-you-see-is-what-you-get) dan editor tekstual. Editor WYSIWYG lebih cocok untuk pemula karena menampilkan tampilan blok kode pada halaman.

Contohnya, TinyMCE adalah text editor WYSIWYG populer yang digunakan di WordPress.

Di sisi lain, editor berbasis teks memberi Anda kebebasan untuk menyesuaikan kode, tapi membutuhkan skill HTML. Beberapa text editor yang paling populer mencakup Sublime Text, Atom, dan Notepad++.

#### 9) Jadi, Apa Itu JavaScript?

Selesai! Di artikel ini kami sudah menjelaskan apa itu JavaScript, fungsi dan manfaatnya, serta perbedaan Java dan JavaScript yang umumnya terletak pada segi kegunaan.

Jadi, JavaScript adalah bahasa pemrograman (lebih tepatnya bahasa penulisan skrip) berbasis objek, umumnya digunakan untuk membuat halaman web interaktif. JavaScript juga sering digunakan bersama HTML dan CSS.

Kemudian, JavaScript memiliki beberapa keunggulan, di antaranya:

- a) Mudah dipelajari dan diimplementasikan pada elemen atau event halaman web
- b) Punya komunitas developer yang luas

- c) Mengurangi request ke server
- d) Kompatibel dengan banyak bahasa lain
- e) Lebih cepat dan ringan dibanding bahasa pemrograman lainnya [4]

## D. Kegiatan Praktikum

Kegiatan praktikum Modul Pengenalan Javascript pada mata kuliah Pemrograman Dasar dilaksanakan pada Kamis, 5 Juni 2025 dan jam 17.00-18.20 WITA di Laboratorium Multimedia STITEK BONTANG, menggunakan Komputer no 5.

Adapun Langkah-langkah pengerjaannya sebagai berikut:

### 1. Manipulasi DOM (Document Object Model)

#### 1) Contoh

##### a. Input

```

contoh > manipulasi_dom.html >
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 {
9              font-family: sans-serif;
10             text-align: center;
11             margin-top: 50px;
12         }
13         #judul { color: steelblue; }
14         button {
15             padding: 10px 15px;
16             font-size: 16px;
17             cursor: pointer;
18         }
19     </style>
20 </head>
21 <body>
22     <h1 id="judul">Selamat Datang di Praktikum JavaScript</h1>
23     <button onclick="ubahTeks()">Ubah Teks Judul</button>
24
25     <script>
26         function ubahTeks() {
27             var elementJudul = document.getElementById("judul");
28             elementJudul.innerHTML = "Teks Berhasil Diubah";
29             elementJudul.style.color = "crimson";
30         }
31     </script>
32 </body>
33 </html>

```

##### b. Output

The figure consists of two screenshots of a web browser window titled "Latihan Manipulasi DOM". Both screenshots show the URL `127.0.0.1:5500/contoh/manipulasi_dom.html`.  
 In the first screenshot (top), the page displays the text "Selamat Datang di Praktikum JavaScript" in black font inside an 

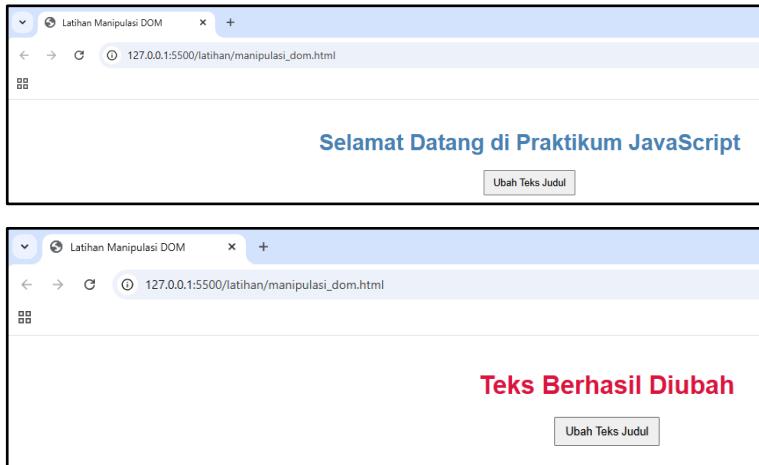
# element, and a button below it labeled "Ubah Teks Judul". In the second screenshot (bottom), the text inside the element has been changed to "Teks Berhasil Diubah" in red font, while the button label remains "Ubah Teks Judul".

## 2) Latihan

### a. Input

```
latihan > manipulasi_dom.html > head > style > button
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 {
9              font-family: sans-serif;
10             text-align: center;
11             margin-top: 50px;
12         }
13         #judul {
14             color: steelblue;
15             font-size: 36px;
16         }
17         button {
18             padding: 10px 15px;
19             font-size: 16px;
20             cursor: pointer;
21         }
22     </style>
23 </head>
24 <body>
25     <h1 id="judul">Selamat Datang di Praktikum JavaScript</h1>
26     <button onclick="ubahTeks()">Ubah Teks Judul</button>
27
28 <script>
29     function ubahTeks() {
30         var elementJudul = document.getElementById("judul");
31         elementJudul.innerHTML = "Teks Berhasil Diubah";
32         elementJudul.style.color = "crimson";
33         elementJudul.style.fontSize = "36px";
34     }
35 </script>
36 </body>
37 </html>
```

### b. Output



### c. Analisis

Perbedaan utama antara kedua program HTML di atas terletak pada pengaturan ukuran font (font size) untuk elemen judul (#judul). Pada program pertama, ukuran font tidak didefinisikan secara eksplisit dalam CSS, sehingga browser akan menggunakan ukuran default (biasanya 32px untuk <h1>). Ketika tombol diklik, fungsi ubahTeks() hanya mengubah teks dan warnanya menjadi "Teks Berhasil Diubah" dan "crimson", tanpa memodifikasi ukuran font. Sementara itu, pada program kedua, ukuran font untuk judul telah ditetapkan secara eksplisit dalam CSS sebagai 36px. Selain itu, fungsi ubahTeks() juga menambahkan perubahan ukuran font

melalui JavaScript (`elementJudul.style.fontSize = "36px"`), meskipun sebenarnya hal ini redundan karena ukuran font sudah diatur di CSS. Perubahan ini membuat kedua program tampak berbeda saat tombol diklik: program pertama akan mempertahankan ukuran font default, sedangkan program kedua memastikan ukuran font tetap 36px meskipun teks diubah.

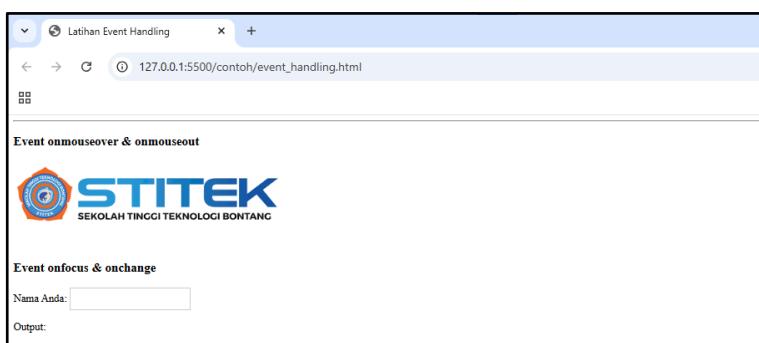
## 2. Penanganan Kejadian (*Event Handling*)

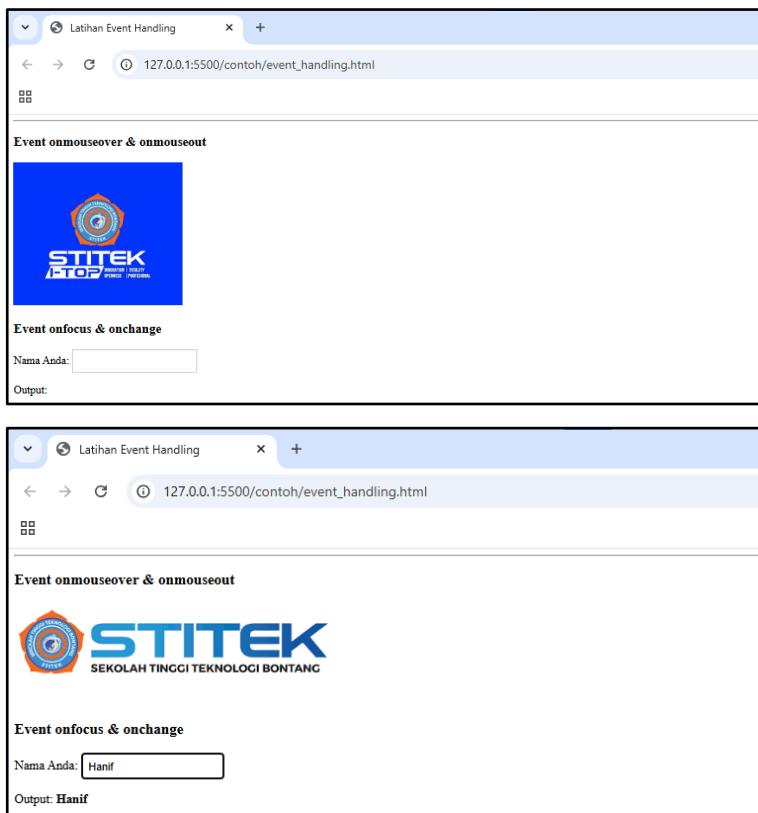
### 1) Contoh

#### a. Input

```
contoh > event_handling.html > event_handling.html > head > style > #inputName
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Latihan Event Handling</title>
5   <style>
6     #inputName {
7       border: 2px solid #ccc;
8       padding: 8px;
9       transition: border-color 0.3s;
10    }
11    #inputName:focus {
12      border-color: dodgerblue;
13    }
14  </style>
15 </head>
16 <body>
17   <h3>Event onmouseover & onmouseout</h3>
18   
20
21   <h3>Event onfocus & onchange</h3>
22
23   <label for="inputName">Nama Anda:</label>
24   <input type="text" id="inputName" onchange="tampilkanOutput()">
25   <p>Output: <span id="hasil"></span></p>
26
27 <script>
28   function gantiGambar() {
29     document.getElementById("gambar").src = "stitek.jpeg";
30   }
31
32   function kembalikanGambar() {
33     document.getElementById("gambar").src = "stitek2.png";
34   }
35
36   function tampilkanOutput() {
37     var nilaiInput = document.getElementById("inputName").value;
38     document.getElementById("hasil").innerHTML = "<b>" + nilaiInput + "</b>";
39   }
40 </script>
41 </body>
42 </html>
```

#### b. Output





### c. Analisis

Program di atas merupakan contoh implementasi event handling dalam JavaScript yang memanfaatkan beberapa jenis event seperti onmouseover, onmouseout, onfocus, dan onchange untuk memodifikasi tampilan secara dinamis.

Pertama, program menggunakan event onmouseover dan onmouseout pada elemen gambar (<img>). Saat kursor mouse berada di atas gambar (onmouseover), fungsi gantiGambar() dipanggil untuk mengubah src gambar menjadi "stitek.jpeg". Sebaliknya, ketika kursor mouse meninggalkan gambar (onmouseout), fungsi kembalikanGambar() akan mengembalikan gambar ke "stitek2.png". Ini menciptakan efek interaktif yang merespons pergerakan mouse pengguna.

Kedua, program menggunakan event onchange pada elemen input teks (<input>). Ketika pengguna mengubah nilai input dan menekan Enter atau mengklik di luar input, fungsi tampilkanOutput() akan mengambil nilai dari input dan menampilkannya dalam elemen

<span id="hasil"> dengan format teks tebal (<b>). Selain itu, input teks memiliki styling CSS yang memberikan efek visual saat difokuskan (:focus), yaitu perubahan warna border menjadi dodgerblue, meningkatkan pengalaman pengguna dengan memberikan umpan balik visual.

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

#### 1) Contoh

##### a. Input

```
contoh@0:~/logika_percabangan.html$ cat logika_percabangan.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Latihan Logika Percabangan</title>
5 </head>
6 <body>
7   <h2>Kalkulator Diskon Belanja</h2>
8   <p>Masukkan total belanja</p>
9   <input type="number" id="totalBelanja" placeholder="Contoh: 150000" value="1000000" />
10  <button onclick="cekDiskon()">Cek Diskon</button>
11  <div id="hasil"></div>
12
13  <script>
14    function cekDiskon() {
15      var total = parseFloat(document.getElementById("totalBelanja").value);
16      var pesan = "";
17
18      if (isNaN(total)) {
19        pesan = "Masukkan angka yang valid.";
20      } else if (total >= 100000) {
21        var diskon = total * 0.10;
22        var totalBayar = total - diskon;
23        pesan = "Selamat! Anda dapat diskon 10% sebesar Rp " + diskon.toLocaleString() + ". Total bayar: <b>Rp " + totalBayar.toLocaleString() + "</b>";
24      } else {
25        pesan = "Maaf, tidak ada diskon. Total bayar: <b>Rp " + total.toLocaleString() + "</b>";
26      }
27
28      document.getElementById("hasil").innerHTML = pesan;
29    }
30  </script>
31 </body>
32 </html>
```

##### b. Output



#### 2) Latihan

##### a. Input

```
latihan@0:~/logika_percabangan.html$ cat logika_percabangan.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Latihan Logika Percabangan</title>
5 </head>
6 <body>
7   <h2>Kalkulator Diskon Belanja</h2>
8   <p>Masukkan total belanja Anda</p>
9   <input type="number" id="totalBelanja" placeholder="Contoh: 150000" value="1000000" />
10  <button onclick="cekDiskon()">Cek Diskon</button>
11  <div id="hasil"></div>
12
13  <script>
14    function cekDiskon() {
15      var total = parseFloat(document.getElementById("totalBelanja").value);
16      var pesan = "";
17
18      if (isNaN(total)) {
19        pesan = "Masukkan angka yang valid.";
20      } else if (total >= 200000) {
21        var diskon = total * 0.15;
22        var totalBayar = total - diskon;
23        pesan = "Selamat! Anda dapat diskon 15% sebesar Rp " + diskon.toLocaleString() + ". Total bayar: <b>Rp " + totalBayar.toLocaleString() + "</b>";
24      } else {
25        pesan = "Maaf, tidak ada diskon. Total bayar: <b>Rp " + total.toLocaleString() + "</b>";
26      }
27
28      document.getElementById("hasil").innerHTML = pesan;
29    }
30  </script>
31 </body>
32 </html>
```

## b. Output



## c. Analisis

Perbedaan utama antara kedua program HTML di atas terletak pada kriteria pemberian diskon dan besaran diskon yang diberikan.

Pada program pertama, diskon sebesar 10% diberikan jika total belanja mencapai Rp 100.000 atau lebih. Jika pengguna memasukkan nilai di bawah itu, program akan menampilkan pesan bahwa tidak ada diskon. Selain itu, program ini juga menangani input yang tidak valid (bukan angka) dengan menampilkan pesan error. Format output menampilkan jumlah diskon dan total bayar dalam format mata uang yang mudah dibaca menggunakan `toLocaleString()`.

Sementara itu, pada program kedua, syarat untuk mendapatkan diskon lebih ketat. Diskon 15% baru diberikan jika total belanja mencapai Rp 250.000 atau lebih. Jika tidak, program akan memberi tahu bahwa tidak ada diskon. Seperti program pertama, program ini juga memeriksa apakah input valid dan menampilkan hasil dalam format yang sama.

## 4. Perulangan untuk Membuat Daftar

### 1) Contoh

#### a. Input

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

## b. Output

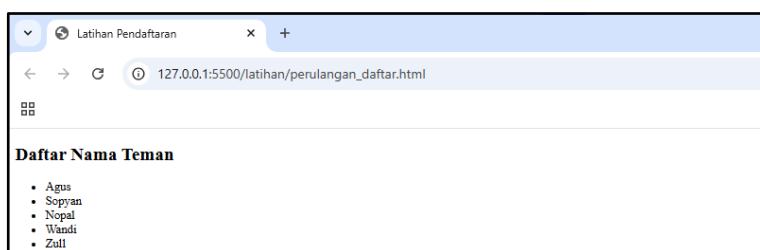


## 2) Latihan

### a. Input

```
latihan > perlengangan_daftar.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Latihan Pendaftaran</title>
5  |</head>
6  <body>
7  |   <h2>Daftar Nama Teman</h2>
8  |   <ul id="daftar-teman">
9  |       <!-- Konten akan diisi oleh JavaScript -->
10 |   </ul>
11
12 <script>
13     var teman = ["Agus", "Sopyan", "Nopal", "Wandi", "Zull"];
14     var daftarElement = document.getElementById("daftar-teman");
15
16     for (let i = 0; i < teman.length; i++) {
17         daftarElement.innerHTML += '<li>' + teman[i] + '</li>';
18     }
19 </script>
20 </body>
21 </html>
```

### b. Output



### c. Analisis

Perbedaan utama antara kedua program HTML di atas terletak pada jenis data yang ditampilkan dan tujuan penggunaannya, meskipun keduanya menggunakan struktur dan logika JavaScript yang serupa. Program pertama bertujuan untuk menampilkan daftar mata kuliah dengan menggunakan array matkul yang berisi elemen-elemen seperti "Pemrograman Web", "Struktur Data", "Basis Data", dan "Jaringan Komputer". Program ini kemudian menggunakan perulangan for untuk menambahkan setiap mata kuliah ke dalam elemen <ul> dengan ID daftar-matkul sebagai daftar item (<li>).

Program kedua memiliki struktur yang sama tetapi digunakan untuk menampilkan daftar nama teman dengan array teman yang berisi nama-nama seperti "Agus", "Sopyan", "Nopal", "Wandi", dan "Zull". Program ini juga menggunakan perulangan for untuk memproses setiap nama dan menambahkannya ke dalam elemen <ul> dengan ID daftar-teman.

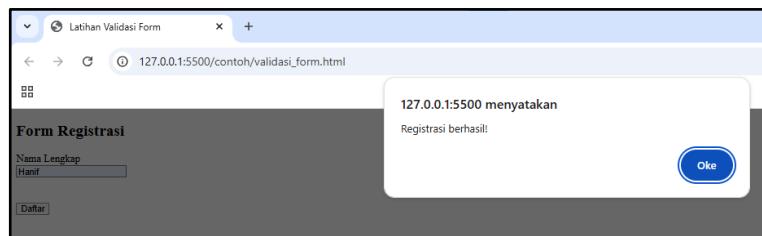
## 5. Validasi Form Sederhana

### 1) Contoh

#### a. Input

```
contoh > validasiform.html > html > body > script
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Latihan Validasi Form</title>
5      <style>
6          .error {
7              color: red;
8              font-size: 0.9em;
9          }
10     </style>
11 </head>
12 <body>
13     <h2>Form Registrasi</h2>
14     <form onsubmit="return validasi()">
15         <label for="name">Nama Lengkap</label><br>
16         <input type="text" id="name" name="name"><br>
17         <small id="name-error" class="error"></small><br>
18         <input type="submit" value="Daftar">
19     </form>
20
21     <script>
22         function validasi() {
23             var inputName = document.getElementById("name").value;
24             var errorElement = document.getElementById("name-error");
25
26             if (inputName == "") {
27                 errorElement.innerHTML = "Nama tidak boleh kosong!";
28                 return false; // Mencegah form untuk dikirim
29             }
30
31             alert("Registrasi berhasil!");
32             return true;
33         }
34     </script>
35 </body>
36 </html>
```

#### b. Output

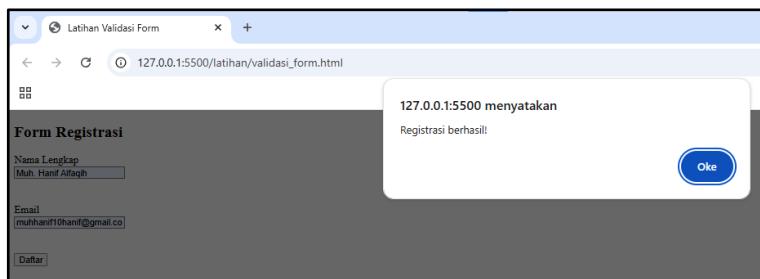


## 2) Latihan

### a. Input

```
latihan > ⑥ validasi_form.html > ⌂ html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Latihan Validasi Form</title>
5      <style>
6          .error {
7              color: red;
8              font-size: 0.9em;
9          }
10     </style>
11  </head>
12 <body>
13     <h2>Form Registrasi</h2>
14     <form onsubmit="return validasi()">
15         <label for="name">Nama Lengkap</label><br>
16         <input type="text" id="name" name="name"><br>
17         <small id="name-error" class="error"></small><br><br>
18
19         <label for="email">Email</label><br>
20         <input type="text" id="email" name="email"><br>
21         <small id="email-error" class="error"></small><br><br>
22
23         <input type="submit" value="Daftar">
24     </form>
25
26
27     <script>
28         function validasi() {
29             var inputName = document.getElementById("name").value;
30             var inputEmail = document.getElementById("email").value;
31             var errorName = document.getElementById("name-error");
32             var errorEmail = document.getElementById("email-error");
33
34             if (inputName == "") {
35                 errorName.innerHTML = "Nama tidak boleh kosong!";
36                 return false; // Mencegah form untuk dikirim
37             }
38
39             alert("Registrasi berhasil!");
40             return true;
41         }
42     </script>
43 </html>
```

### b. Output



### c. Analisis

Kedua program HTML tersebut memiliki tujuan yang sama, yaitu melakukan validasi form registrasi, namun terdapat beberapa perbedaan signifikan dalam implementasinya. Program pertama merupakan contoh validasi dasar yang hanya memeriksa field nama lengkap, dimana jika field tersebut kosong akan menampilkan pesan error dan mencegah pengiriman form. Sementara itu, program kedua seharusnya merupakan pengembangan dari program pertama dengan menambahkan validasi untuk field email, namun terdapat beberapa kesalahan implementasi yang membuat fungsionalitasnya tidak bekerja optimal. Perbedaan utama terletak pada jumlah field yang divalidasi dan kelengkapan kode validasinya. Program pertama sukses melakukan validasi sederhana dengan rapi, sedangkan program kedua meskipun memiliki struktur HTML untuk dua field

input, validasi JavaScript-nya belum diimplementasikan dengan benar untuk field email dan mengandung beberapa bug seperti penggunaan ID yang tidak konsisten dan redeklarasi variabel. Kedua program sama-sama menggunakan pendekatan onsubmit dan styling error message yang identik, namun program kedua membutuhkan penyempurnaan kode agar dapat memvalidasi kedua field secara efektif. Program pertama cocok untuk kebutuhan validasi sederhana, sementara program kedua berpotensi untuk validasi lebih kompleks jika kesalahan implementasinya diperbaiki.

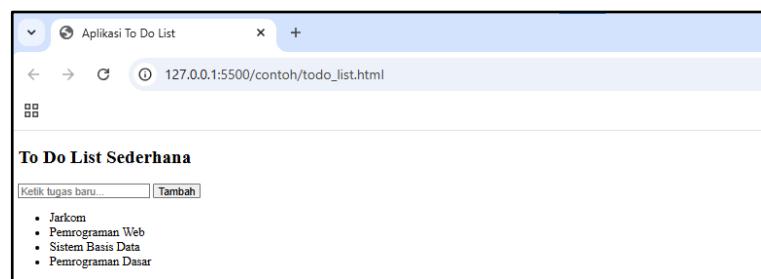
## 6. Bekerja dengan Array (Aplikasi To-Do List)

### 1) Contoh

#### a. Input

```
contoh > O todo_list.html > HTML
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Aplikasi To Do List</title>
5  </head>
6  <body>
7  |   <h2>To Do List Sederhana</h2>
8  |   <input type="text" id="input-todo" placeholder="Ketik tugas baru...">
9  |   <button onclick="tambahTugas()">Tambah</button>
10 |   <ul id="daftar-tugas"></ul>
11 |
12 |
13 <script>
14     var tugas = [];
15
16     function tambahTugas() {
17         var inputElement = document.getElementById("input-todo");
18         if (inputElement.value.trim() !== "") {
19             tugas.push(inputElement.value);
20             inputElement.value = "";
21             tampilkanDaftar();
22         }
23     }
24
25     function tampilkanDaftar() {
26         var daftarElement = document.getElementById("daftar-tugas");
27         daftarElement.innerHTML = "";
28
29         for (let i = 0; i < tugas.length; i++) {
30             daftarElement.innerHTML += "<li>" + tugas[i] + "</li>";
31         }
32     }
33 </script>
34 </body>
35 </html>
```

#### b. Output

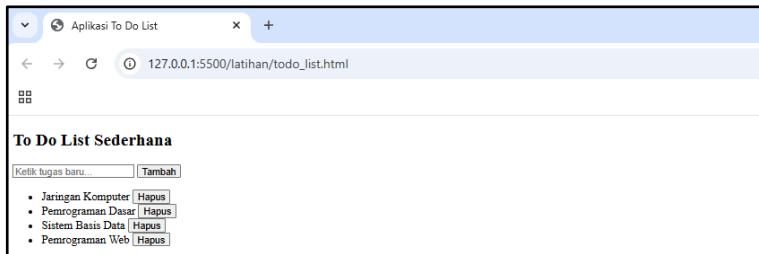


## 2) Latihan

### a. Input

```
latihan > o todo_list.html > HTML > body > script >hapusTugas
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Aplikasi To Do List</title>
5  </head>
6  <body>
7  <h2>To Do List Sederhana</h2>
8  <input type="text" id="input-todo" placeholder="Ketik tugas baru...">
9  <button onclick="tambahTugas()">Tambah</button>
10 <ul id="daftar-tugas"></ul>
11 </body>
12 <script>
13   var tugas = [];
14
15   function tambahTugas() {
16     var inputElement = document.getElementById("input-todo");
17     if (inputElement.value.trim() != "") {
18       tugas.push(inputElement.value);
19       inputElement.value = "";
20       tampilkanDaftar();
21     }
22   }
23
24   function hapusTugas(index) {
25     tugas.splice(index, 1);
26     tampilkanDaftar();
27   }
28
29   function tampilkanDaftar() {
30     var daftarElement = document.getElementById("daftar-tugas");
31     daftarElement.innerHTML = "";
32
33     for (let i = 0; i < tugas.length; i++) {
34       daftarElement.innerHTML += `
35         <li>
36           ${tugas[i]}
37           <button onclick="hapusTugas(${i})">Hapus</button>
38         </li>
39       `;
40     }
41   }
42 </script>
43 </html>
```

### b. Output



### c. Analisis

Perbedaan utama antara ketiga program To Do List tersebut terletak pada fitur dan fungsionalitas yang ditawarkan, meskipun memiliki struktur dasar yang serupa. Program pertama merupakan versi paling sederhana yang hanya menyediakan fitur penambahan tugas, dimana pengguna dapat memasukkan tugas baru yang kemudian ditampilkan dalam bentuk daftar tanpa opsi interaksi lebih lanjut. Program ini menggunakan array untuk menyimpan tugas dan perulangan for sederhana untuk menampilkannya, namun tidak menyediakan kemampuan untuk menghapus item yang sudah dimasukkan.

Program kedua dan ketiga (yang identik) merupakan pengembangan dari program pertama dengan menambahkan fitur penghapusan tugas yang signifikan. Kedua program ini memperkenalkan fungsi hapusTugas(index) yang memungkinkan pengguna untuk

menghapus item tertentu dari daftar dengan mengklik tombol "Hapus" yang disertakan pada setiap item. Implementasi ini menggunakan method splice() untuk menghapus item dari array berdasarkan indeksnya, kemudian memanggil tampilkanDaftar() untuk memperbarui tampilan. Perbedaan visual utama terlihat pada template string yang digunakan untuk menambahkan tombol hapus bersamaan dengan setiap item tugas.

## 7. Kalkulator Sederhana

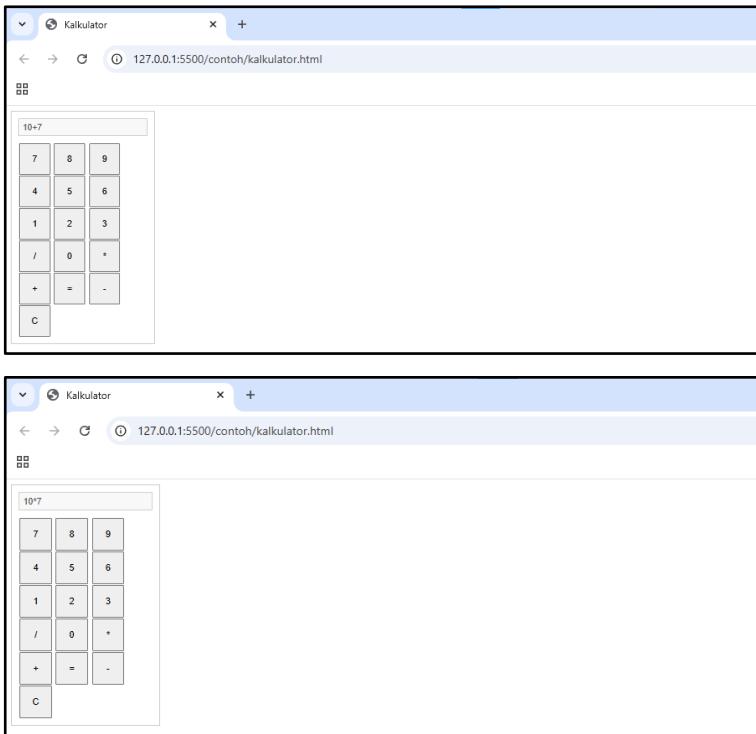
### 1) Contoh

#### a. Input

```
contoh > kalkulator.html > body > div.buttons > button
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Kalkulator</title>
5      <style>
6          .calculator {
7              width: 200px;
8              border: 1px solid #ccc;
9              padding: 10px;
10         }
11         #display {
12             width: 100%;
13             margin-bottom: 10px;
14             padding: 5px;
15             box-sizing: border-box;
16         }
17         .buttons button {
18             width: 48px;
19             height: 48px;
20             margin: 1px;
21         }
22     </style>
23 </head>
```

```
24 <body>
25     <div class="calculator">
26         <input type="text" id="display" disabled>
27         <div class="buttons">
28             <button onclick="appendValue('7')">7</button>
29             <button onclick="appendValue('8')">8</button>
30             <button onclick="appendValue('9')">9</button>
31             <button onclick="appendValue('4')">4</button>
32             <button onclick="appendValue('5')">5</button>
33             <button onclick="appendValue('6')">6</button>
34             <button onclick="appendValue('1')">1</button>
35             <button onclick="appendValue('2')">2</button>
36             <button onclick="appendValue('3')">3</button>
37             <button onclick="appendValue('/')">*</button>
38             <button onclick="appendValue('0')">0</button>
39             <button onclick="calculate()">=</button>
40             <button onclick="appendValue('.')">.0</button>
41             <button onclick="clearDisplay()"/>C</button>
42         </div>
43     </div>
44 </body>
45 <script>
46     var display = document.getElementById('display');
47
48     function appendValue(value) {
49         display.value += value;
50     }
51
52     function clearDisplay() {
53         display.value = '';
54     }
55
56     function calculate() {
57         try {
58             display.value = eval(display.value);
59         } catch (error) {
60             display.value = 'Error';
61         }
62     }
63 </script>
64 </html>
65
```

## b. Output



## c. Analisis

Program kalkulator HTML di atas merupakan implementasi sederhana dari kalkulator digital yang memungkinkan pengguna untuk melakukan operasi aritmatika dasar. Program ini terdiri dari tiga komponen utama: antarmuka pengguna (UI), logika operasi, dan fungsi tampilan. Antarmuka dibangun menggunakan elemen HTML dengan styling CSS yang memberikan tampilan kalkulator klasik dengan tombol-tombol berukuran seragam dan display untuk menampilkan input dan hasil perhitungan.

Fungsi utama kalkulator ini diimplementasikan melalui tiga fungsi JavaScript. Fungsi `appendValue()` bertanggung jawab untuk menambahkan nilai yang diklik pengguna ke dalam display, memungkinkan pembentukan ekspresi matematika. Fungsi `clearDisplay()` menyediakan kemampuan untuk mereset kalkulator dengan mengosongkan display. Fungsi `calculate()` merupakan inti dari kalkulator ini, yang menggunakan fungsi `eval()` bawaan JavaScript untuk mengevaluasi ekspresi matematika yang dimasukkan pengguna, dengan penanganan error sederhana yang

akan menampilkan pesan 'Error' jika terjadi kesalahan dalam evaluasi.

Program ini memiliki beberapa karakteristik penting. Pertama, penggunaan CSS inline yang sederhana namun efektif untuk mengatur tata letak kalkulator. Kedua, pendekatan event handling langsung pada elemen HTML melalui atribut onclick. Ketiga, implementasi logika kalkulasi yang minimalis namun fungsional dengan memanfaatkan kemampuan evaluasi string JavaScript. Keempat, antarmuka pengguna yang intuitif dengan tombol-tombol angka dan operator yang tersusun rapi.

## 8. Bekerja dengan Objek String

### 1) Contoh

#### a. Input

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

#### b. Output



#### c. Analisis

Program HTML di atas merupakan aplikasi sederhana untuk menganalisis string nama yang dimasukkan pengguna. Program ini memanfaatkan berbagai metode manipulasi string dalam JavaScript untuk mengekstrak informasi dari input teks. Ketika pengguna memasukkan nama dan mengklik tombol "Analisis", program akan

menampilkan tiga jenis informasi: versi kapital dari nama tersebut, jumlah karakter yang membentuk nama, serta tiga karakter pertama dari nama tersebut.

Struktur program terdiri dari tiga komponen utama: antarmuka pengguna, fungsi analisis, dan output hasil. Antarmuka pengguna yang minimalis hanya membutuhkan satu field input dan sebuah tombol. Fungsi analisisNama() menjadi inti program yang melakukan tiga operasi string utama: konversi ke huruf kapital menggunakan toUpperCase(), penghitungan panjang string dengan length, dan ekstraksi substring melalui substring(0, 3). Hasil analisis kemudian ditampilkan dalam format HTML yang diformat dengan tag <b> untuk penekanan visual.

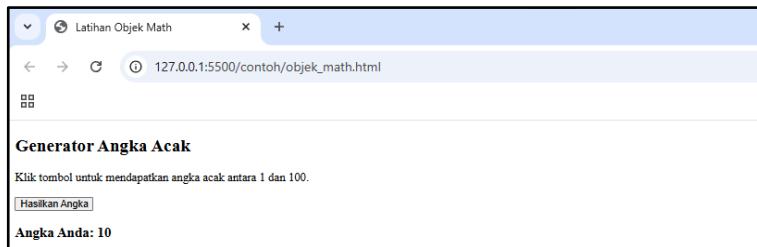
## 9. Menggunakan Objek Math

### 1) Contoh

#### a. Input

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

#### b. Output



#### c. Analisis

Program HTML di atas merupakan generator angka acak sederhana yang memanfaatkan objek Math dalam JavaScript. Aplikasi ini dirancang untuk menghasilkan dan menampilkan bilangan bulat acak antara 1 sampai 100 setiap kali pengguna mengklik tombol.

Program ini terdiri dari antarmuka pengguna minimalis dengan judul, petunjuk singkat, tombol interaksi, dan area tampilan hasil.

Fungsi utama program terdapat pada generateRandom() yang menggunakan tiga metode penting dari objek Math: random() untuk menghasilkan angka desimal acak antara 0 (inklusif) hingga 1 (eksklusif), perkalian dengan 100 untuk memperluas rentang menjadi 0-99.999..., dan floor() untuk membulatkan ke bawah ke bilangan bulat terdekat. Penambahan angka 1 di akhir rumus menggeser rentang menjadi 1-100. Hasil akhir kemudian ditampilkan dalam elemen span dengan innerText untuk menghindari interpretasi HTML.

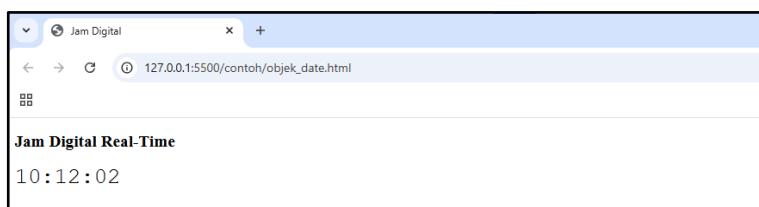
## 10. Menampilkan Tanggal dan Jam Dinamis

### 1) Contoh

#### a. Input

```
contoh > objek_date.html > HTML > body > divjam
<!DOCTYPE html>
<html>
<head>
<title>Jam Digital</title>
<style>
#jam {
    font-size: 2em;
    font-family: 'Courier New', Courier, monospace;
}
</style>
</head>
<body><h2>Jam Digital</h2>
<div id="jam"></div>
<script>
function updateJam() {
    var sekarang = new Date();
    var jam = sekarang.getHours();
    var menit = sekarang.getMinutes();
    var detik = sekarang.getSeconds();
    // Tambahan 0 di depan jika angka < 10
    jam = (jam < 10 ? "0" + jam);
    menit = (menit < 10 ? "0" + menit);
    detik = (detik < 10 ? "0" + detik);
    var waktustring = jam + ":" + menit + ":" + detik;
    document.getElementById("jam").innerHTML = waktustring;
}
function jalankanJam() {
    setInterval(updateJam, 1000);
    setInterval(updateJam, 1000);
    // Panggil sekali di awal untuk menghindari delay 1 detik
    updateJam();
}
</script>
</body>
</html>
```

#### b. Output



#### c. Analisis

Program jam digital real-time ini merupakan contoh implementasi sederhana namun efektif yang menampilkan waktu terkini secara dinamis menggunakan teknologi web dasar. Program ini memanfaatkan objek Date JavaScript untuk mendapatkan waktu

sistem, kemudian menampilkannya dalam format HH:MM:SS yang diperbarui setiap detik. Fungsi utama updateJam() bertanggung jawab untuk mengambil komponen waktu (jam, menit, detik), memformatnya dengan menambahkan nol di depan angka satuan, dan menampilkannya ke dalam elemen DOM. Program ini dijalankan pertama kali saat halaman dimuat (melalui event onload) dan terus berjalan secara periodik berkat fungsi setInterval yang memperbarui tampilan setiap 1000 milidetik.

Dari segi antarmuka, program ini didesain secara minimalis namun fungsional dengan menggunakan font monospace Courier New yang khas tampilan digital dan ukuran font besar (2em) untuk memastikan keterbacaan yang baik. Aspek teknis yang menarik termasuk penanganan formatting waktu dengan operator ternary untuk leading zero, serta strategi pemanggilan ganda (setInterval dan pemanggilan awal) untuk menghindari delay tampilan pertama kali. Program ini juga menunjukkan praktik coding yang baik dengan pemisahan yang jelas antara logika bisnis (perhitungan waktu) dan presentasi (tampilan ke user).

## 11. Membuat Image Slideshow Sederhana

### 1) Contoh

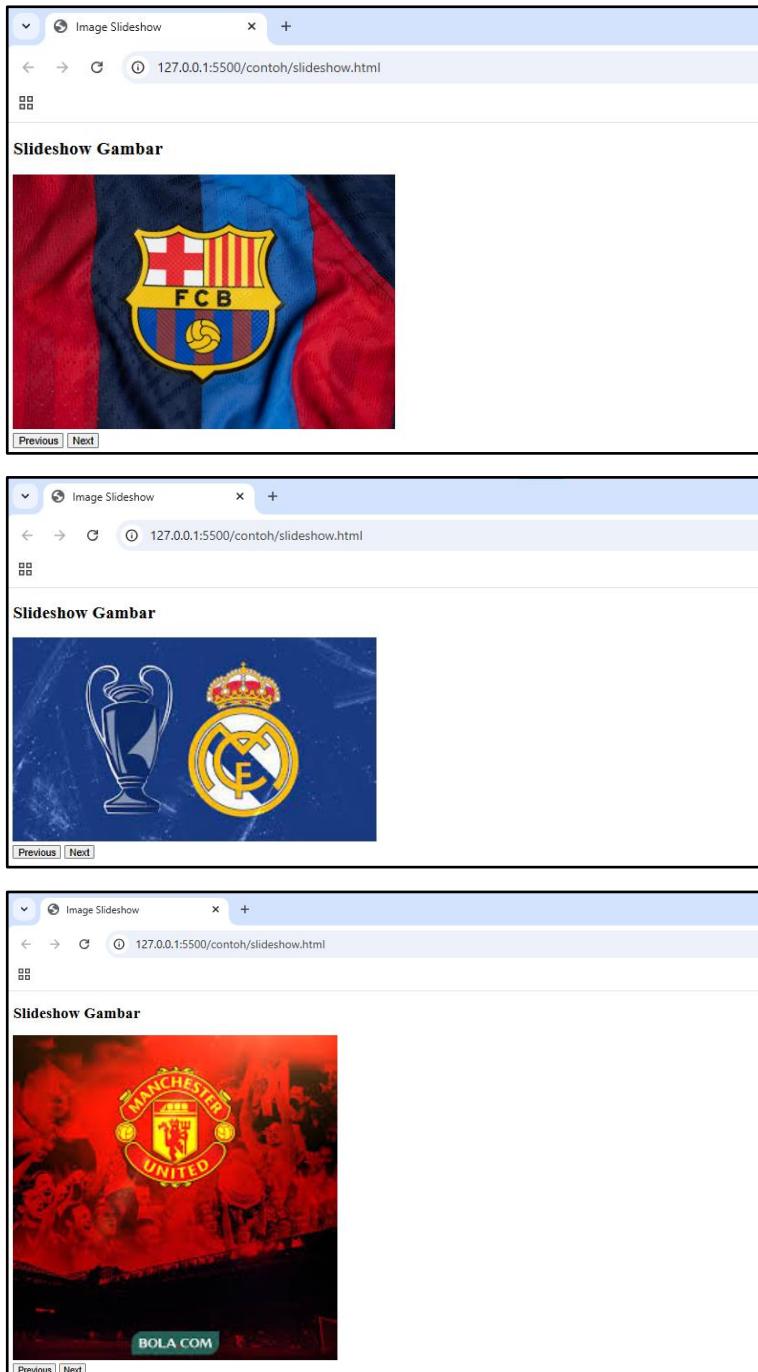
#### a. Input

```

const > <!DOCTYPE html> </html>
1 <html>
2   <head>
3     <title>Image Slideshow</title>
4   </head>
5   <body>
6     <h2>Slideshow Gambar</h2>
7     <img id="slideshow-image" alt="slideshow" width="500" />
8     <button onclick="gantiGambar(-1)">Previous</button>
9     <button onclick="gantiGambar(1)">Next</button>
10    </body>
11  </html>
12
13
14  <script>
15    var gambar = [
16      "https://speziali-images.concerting.com/images/av/4d/08/7e/1f/slideshow/0/0-0_Rome-would-like-to-be-dissociated-from-FC-Barcelona-/960x800.jpg?fit=scale",
17      "https://speziali-ytnd-thed.getstatic.com/images/7-0h/AbnOicCs-tpU050900224cExX1mzZ57wLwfrBRQI",
18    ];
19    var indexgambar = 0;
20    var imagelement = document.getElementById("slideshow-image");
21
22    function tampilkanGambar() {
23      imagelement.src = gambar[indexgambar];
24    }
25
26    function gantiGambar(brah) {
27      indexgambar += brah;
28
29      // Jika sang gambar terakhir, kembali ke awal
30      if (indexgambar == gambar.length) {
31        indexgambar = 0;
32      }
33      // Jika sudah gambar pertama dan menukar previous, pergi ke akhir
34      if (indexgambar < 0) {
35        indexgambar = gambar.length - 1;
36      }
37      tampilkanGambar();
38    }
39
40    // Tampilan gambar pertama saat halaman dimuat
41    tampilkanGambar();
42  </script>
43 </body>
44 </html>

```

## b. Output



## c. Analisis

Program Image Slideshow ini merupakan implementasi sederhana dari galeri gambar yang dapat dinavigasi menggunakan tombol previous dan next. Program ini menggunakan pendekatan dasar JavaScript untuk menciptakan fungsi slideshow dengan tiga gambar yang telah didefinisikan dalam array.

Secara struktural, program ini terdiri dari tiga komponen utama: elemen <img> sebagai tempat menampilkan gambar, dua tombol navigasi, dan logika JavaScript yang mengatur perpindahan gambar. Array gambar menyimpan URL tiga gambar berbeda, sementara variabel indexGambar berfungsi sebagai penanda posisi gambar yang sedang aktif. Fungsi tampilkanGambar() bertanggung jawab menampilkan gambar sesuai indeks saat ini, sedangkan gantiGambar() menangani logika navigasi dengan parameter arah (1 untuk next, -1 untuk previous).

Mekanisme navigasinya menerapkan sistem sirkular - ketika mencapai gambar terakhir dan menekan next, akan kembali ke gambar pertama, begitu pula sebaliknya jika berada di gambar pertama dan menekan previous akan menuju gambar terakhir. Program ini langsung menampilkan gambar pertama saat halaman dimuat melalui pemanggilan tampilkanGambar() di akhir script.

## E. Tugas

- Buatlah sebuah halaman web interaktif untuk menampilkan dan memesan produk
  - Input

### Kelas head

```
tugas > ⌘ tugas.html ⌘ HTML ⌘ body ⌘ script > ↵ products > ↵ price
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Toko Elektronik Cepat</title>
7    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
8    <style>
9      .product-display {
10        border: 1px solid #ddd;
11        padding: 20px;
12        border-radius: 5px;
13        margin-bottom: 20px;
14      }
15      .product-list {
16        cursor: pointer;
17      }
18      .product-list:hover {
19        background-color: #f0f0f0;
20      }
21      .receipt {
22        background-color: #fff9f9;
23        padding: 20px;
24        border-radius: 5px;
25        margin-top: 20px;
26      }
27    </style>
28  </head>
```

## Kelas div

```
29 <body>
30   <div class="container mt-4">
31     <!-- Header & Waktu Transaksi -->
32     <div class="row mb-4">
33       <div class="col-12 text-center">
34         <h1>Toko Elektronik Cepat</h1>
35         <h5 id="current-time"></h5>
36       </div>
37     </div>
38
39     <div class="row">
40       <!-- Display Produk Utama -->
41       <div class="col-md-8">
42         <div class="product-display">
43           <div id="slideshow-image" style="background-color: #ccc; height: 300px; position: relative; width: 100%;">
44             <img alt="Product Image" style="max-height: 300px; position: absolute; top: 0; left: 0; width: 100%; height: 100%; object-fit: cover;" id="image"/>
45             <div class="mt-2">
46               <button onclick="changeImage(-1)" class="btn btn-secondary"><Previous/>
47               <button onclick="changeImage(1)" class="btn btn-secondary"><Next/>
48             </div>
49           </div>
50           <h3 id="product-name"></h3>
51           <h4 id="product-price"></h4>
52         </div>
53
54       <!-- Form Pemesanan -->
55       <div class="mt-4">
56         <h4>Form Pemesanan</h4>
57         <form id="order-form" style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;">
58           <div class="mb-3">
59             <label for="customer-name" class="form-label">Nama Pemesan</label>
60             <input type="text" class="form-control" id="customer-name" required>
61           </div>
62           <div class="mb-3">
63             <label for="quantity" class="form-label">Jumlah Pesan</label>
64             <input type="number" class="form-control" id="quantity" min="1" required>
65           </div>
66           <div class="mb-3">
67             <label for="promo-code" class="form-label">Kode Promo (Optional)</label>
68             <input type="text" class="form-control" id="promo-code" style="width: 150px;">
69           </div>
70           <button type="button" onclick="processOrder()" class="btn btn-primary">Pesan Sekarang</button>
71         </form>
72       </div>
73
74       <!-- Struk Pesanant -->
75       <div id="receipt" class="receipt mt-4" style="display: none;">
76         <h4>Struk Pesanant</h4>
77         <p><strong>Order ID:</strong> <span id="order-id"></span></p>
78         <p><strong>Nama Pemesan:</strong> <span id="receipt-name"></span></p>
79         <p><strong>Produk:</strong> <span id="receipt-product"></span></p>
80         <p><strong>Jumlah:</strong> <span id="receipt-quantity"></span></p>
81         <p><strong>Subtotal:</strong> Rp <span id="subtotal"></span></p>
82         <p><strong>Potongan:</strong> Rp <span id="discount"></span></p>
83         <p><strong>Total Akhir:</strong> Rp <span id="total"></span></p>
84       </div>
85
86     <!-- Daftar Produk -->
87     <div class="col-md-4">
88       <h4>Daftar Produk</h4>
89       <div id="product-list" class="list-group"></div>
90     </div>
91   </div>
92 </div>
93
```

## Kelas script (data produk)

```
95 <script>
96   // Data Produk
97   const products = [
98     {
99       name: "iPhone 11",
100      price: 3556100,
101      images: [
102        "https://cdn.alloallo.media/catalog/product/apple/iphone-11/iphone-11-white.webp",
103        "https://cdn.alloallo.media/catalog/product/apple/iphone-11/iphone-11-black.jpg",
104        "https://cdn.alloallo.media/catalog/product/apple/iphone-11/iphone-11-green.jpg"
105      }
106    },
107    {
108      name: "Laptop Asus TUF A15 Ryzen 5-7535H",
109      price: 9599000,
110      images: [
111        "https://static.retailerworldvn.com/Products/Images/12217/33217/asus-tuf-a15-ryzen-5-7535hs-fd506nf-r525hd-om-blakv-1.jpg",
112        "https://static.retailerworldvn.com/Products/Images/12217/33217/asus-tuf-a15-ryzen-5-7535hs-fd506nf-r525hd-om-blakv-2.jpg"
113      ]
114    },
115    {
116      name: "Redmi Note 14 50",
117      price: 3699000,
118      images: [
119        "https://i02.apimgfile.com/mi-com-product/fly-birds/redmi-note-14-5g-pc/387b259a8b8e9f9536eddd110fa76b18.jpg",
120        "https://i02.apimgfile.com/mi-com-product/fly-birds/redmi-note-14-5g-pc/5c39f725c733118a9751bc7.jpg",
121        "https://i02.apimgfile.com/mi-com-product/fly-birds/redmi-note-14-5g-pc/4069f725c733118a9751bc7.jpg",
122        "https://i02.apimgfile.com/mi-com-product/fly-birds/redmi-note-14-5g-pc/ab5c9947fc724471648aa3ab3ec7f.jpg"
123      ]
124    }
125  ];
126 </script>
```

## Kelas script (variabel global)

```
127   // Variabel Global
128   let currentProductIndex = 0;
129   let currentImageIndex = 0;
130   const slideshowImage = document.getElementById('slideshow-image');
131   const productNameElement = document.getElementById('product-name');
132   const productPriceElement = document.getElementById('product-price');
133   const productListElement = document.getElementById('product-list');
134   const currentTimeElement = document.getElementById('current-time');
```

## Kelas script (inisialisasi dan menampilkan produk)

```
136 // Inisialisasi
137 function init() {
138   displayProductList();
139   showProduct(currentProductIndex);
140   updateTime();
141   setInterval(updateTime, 1000);
142 }
143
144 // Menampilkan Daftar Produk
145 function displayProductList() {
146   productListElement.innerHTML = '';
147   products.forEach(product, index) => {
148     const productItem = document.createElement('div');
149     productItem.innerHTML = `<div>${product.name} - Rp ${product.price.toLocaleString()}</div>`;
150     productItem.onclick = () => {
151       currentProductIndex = index;
152       currentImageIndex = 0;
153       showProduct(currentProductIndex);
154     };
155     productListElement.appendChild(productItem);
156   });
157 }
158
159 // Menampilkan Produk yang Dipilih
160 function showProduct(index) {
161   const product = products[index];
162   productNameElement.textContent = product.name;
163   productPriceElement.textContent = `Rp ${product.price.toLocaleString()}`;
164   slideshowImage.src = product.images[0];
165   currentImageIndex = 0;
166 }
167 }
```

## Kelas script (mengganti gambar di slideshow)

```
169 // Mengganti Gambar di Slideshow
170 function changeImage(direction) {
171   const product = products[currentProductIndex];
172   currentImageIndex += direction;
173
174   if (currentImageIndex >= product.images.length) {
175     currentImageIndex = 0;
176   } else if (currentImageIndex < 0) {
177     currentImageIndex = product.images.length - 1;
178   }
179
180   slideshowImage.src = product.images[currentImageIndex];
181 }
```

## Kelas script (memperbarui waktu)

```
183 // Memperbarui Waktu
184 function updateTime() {
185   const now = new Date();
186   currentTimeElement.textContent = now.toLocaleString();
187 }
```

## Kelas script (proses pesanan)

```
189 // Proses Pemesanan
190 function processOrder() {
191   const customerName = document.getElementById('customer-name').value;
192   const quantity = parseInt(document.getElementById('quantity').value);
193   const promoCode = document.getElementById('promo-code').value;
194 }
```

## Kelas script (validasi form)

```
195 // Validasi Form
196 if (!customerName || !quantity || quantity <= 0) {
197   alert('Mohon isi nama pemesan dan jumlah pesanan (minimal 1)');
198   return;
199 }
200
201 const product = products[currentProductIndex];
202 const subtotal = product.price * quantity;
```

## Kelas script (hitung diskon)

```
204 // Hitung Diskon
205 let discount = 0;
206 if (promoCode.toUpperCase() === 'DISKON10') {
207   discount = subtotal * 0.1;
208 }
209
210 const total = subtotal - discount;
```

## Kelas script (generate order ID)

```
211 // Generate Order ID
212 const orderId = 'INV-' + Math.floor(10000 + Math.random() * 90000);
213
```

## Kelas script (tampilkan struk)

```
215     // Tampilkan Struk
216     document.getElementById('order-id').textContent = orderId;
217     document.getElementById('receipt-name').textContent = customerName;
218     document.getElementById('receipt-product').textContent = product.name;
219     document.getElementById('receipt-quantity').textContent = quantity;
220     document.getElementById('subtotal').textContent = subtotal.toLocaleString();
221     document.getElementById('discount').textContent = discount.toLocaleString();
222     document.getElementById('total').textContent = total.toLocaleString();
223
224     document.getElementById('receipt').style.display = 'block';
225   }
226
227   // Jalankan saat halaman dimuat
228   window.onload = init;
229 </script>
```

230 | <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
231 </body>
232 </html>

### b. Output



The screenshot shows the "Form Pemesanan" section. It includes fields for "Nama Pemesan" (Hanif), "Jumlah Pesan" (1), and "Kode Promo (Opcional)". A "Pesan Sekarang" button is at the bottom. Below this, a "Struk Pesanan" section displays the order details: Order ID: INV-60401, Nama Pemesan: Hanif, Produk: Iphone 11, Jumlah: 1, Subtotal: Rp 3.556.180, Potongan: Rp 0, and Total Akhir: Rp 3.556.180.



## 2. Link Github

<https://github.com/MuhHanifAlfaqih/Modul4>

## Daftar Pustaka

- [1] N. Huda, "Mengenal Apa itu HTML, Cara Kerja, Fungsi dan Strukturnya!," 7 Februari 2025. [Online]. Available: <https://www.dewaweb.com/blog/panduan-lengkap-untuk-belajar-html-bagi-pemula/>. [Accessed 4 Juni 2025].
- [2] A. C., "Apa Itu CSS? Pengertian, Fungsi, dan Cara Kerjanya," 27 MEI 2025. [Online]. Available: <https://www.hostinger.com/id/tutorial/apa-itu-css>. [Accessed 11 JUNI 2025].

- [3] M. R. Adani, "Cara Menggunakan Framework Bootstrap dan Keunggulan yang Dimiliki," 21 April 2021. [Online]. Available: <https://www.sekawanmedia.co.id/blog/apa-itu-bootstrap/>. [Accessed 19 Juni 2025].
- [4] F. Ayunindya, "Apa Itu JavaScript: Definisi, Fungsi, dan Perbedaannya dengan Java," 14 Maret 2025. [Online]. Available: <https://www.hostinger.com/id/tutorial/apa-itu-javascript>. [Accessed 27 Juni 2025].