

# **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 .....	iv
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
Daftar Pustaka .....	48

# 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 75313  
 ☎ 0548 - 28782 | 0811 530 2001 ✉ admin@stitek.ac.id 🌐 https://stitek.ac.id

## KARTU KENDALI PRAKTIKUM

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

Pertemuan 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.

## 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:

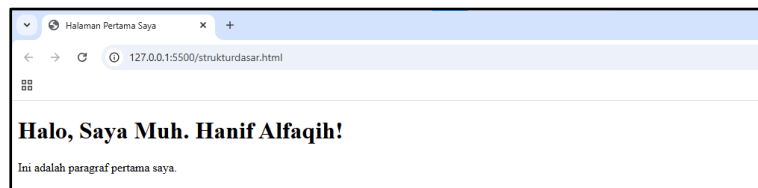
### 1. Pengenalan HTML dan Struktur Dasar Dokumen

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>`

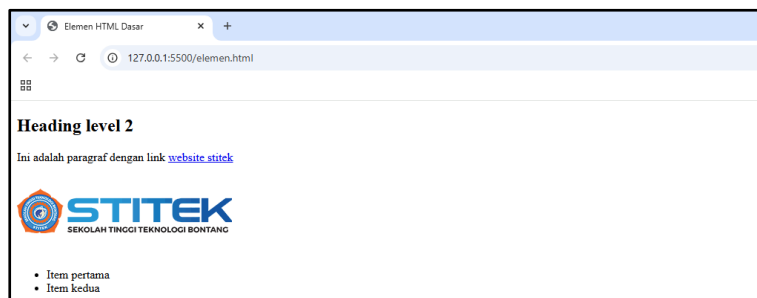
Menandai akhir seluruh dokumen HTML.

## 2. Elemen HTML Dasar

### Input

```
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>`

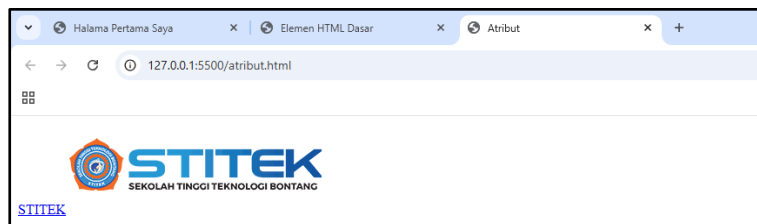
Menutup dokumen HTML sepenuhnya.

### 3. Atribut HTML

#### Input

```
strukturdasar.html  elemen.html  atribut.html X  format.html  linkgambar.html
atribut.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>

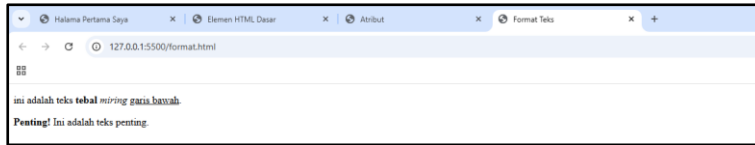
Menutup bagian isi dan dokumen HTML.

#### 4. Format Teks

##### Input

```
<?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>`

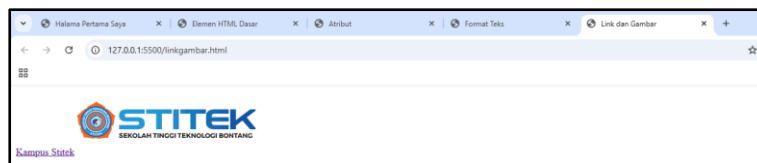
Menutup bagian isi (`</body>`) dan seluruh dokumen HTML (`</html>`).

## 5. Link dan Gambar

### Input

```
elemen.html atribut.html format.html linkgambar.html x daftar.html tabel.html
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>`).

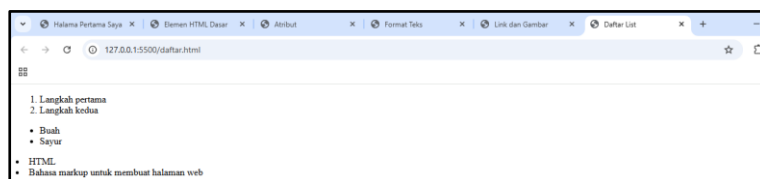
## 6. Daftar (List)

### Input



```
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



1. Langkah pertama  
2. Langkah kedua

- Buah
- Sayur

• HTML

• Bahasa markup untuk membuat halaman web

### 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>

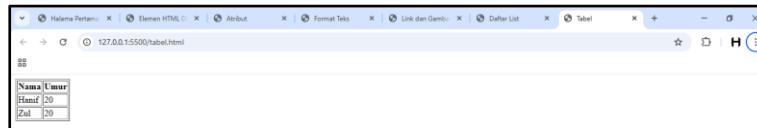
Menutup bagian isi halaman dan menutup seluruh dokumen HTML.

## 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.

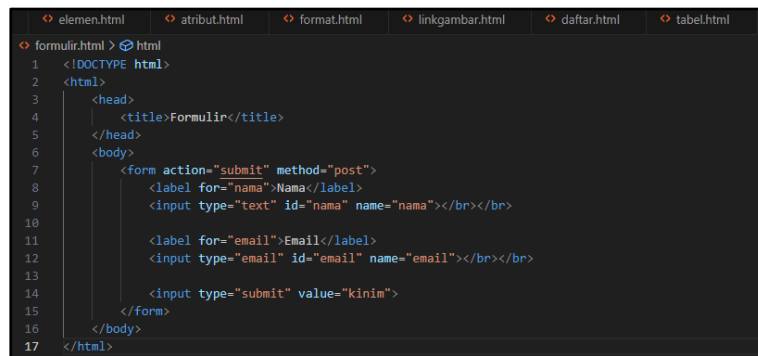
Data “Zul” dan “20” pada kolom nama dan umur.

i. `</table>`  
`</body>`  
`</html>`

Menutup tabel, body, dan dokumen HTML.

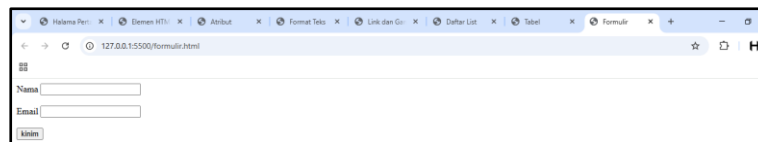
## 8. Formulir HTML

### 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="kirim">
15     </form>
16   </body>
17 </html>
```

### Output



127.0.0.1:5500/formulir.html

Nama

Email

### 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.

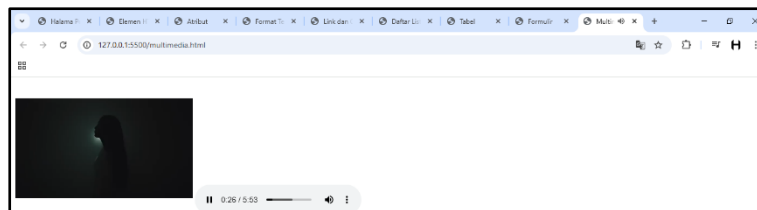
- f. `<label for="nama">Nama</label>`  
`<input type="text" id="nama" name="nama"></br></br>`  
`<label>` membuat teks “Nama” yang terhubung dengan input berikutnya (`for="nama"` menghubungkan label ke input dengan `id="nama"`).  
`<input type="text">` membuat kotak isian teks biasa.  
`id="nama"` memberi 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

```
multimedia.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Multimedia</title>
5 </head>
6 <body>
7   <video width="320" height="240" controls>
8     <source src="Pamungkas - To The Bone (Official Music Video).mp4" type="video/mp4">
9     browser anda tidak mendukung tag video.
10  </video>
11  <audio controls>
12    <source src="Pamungkas - To The Bone (Official Music Video) [oIYMenB637c].mp3" type="audio/mpeg">
13    browser anda tidak mendukung tag audio.
14  </audio>
15 </body>
16 </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

1. 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     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 </style>

```

```

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

```

```

111         <!-- Radio Button Jenis Kelamin -->
112         <label>Jenis Kelamin</label>
113         <div class="radio-group">
114             <input type="radio" id="pria" name="jenis_kelamin" value="Pria" required />
115             <label for="pria">Pria</label>
116             <input type="radio" id="wanita" name="jenis_kelamin" value="Wanita" />
117             <label for="wanita">Wanita</label>
118         </div>
119         <!-- Textarea Alamat -->
120         <label for="alamat">Alamat Lengkap</label>
121         <textarea id="alamat" name="alamat" placeholder="Masukkan alamat lengkap Anda" />
122     </div>
123 </body>
124 </html>

```

```

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>

```

```

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 </div>
166 </body>
167 </html>

```

## Output

The screenshot shows a web browser window with the URL '127.0.0.1:5500/formmahasiswa.html'. The page displays a registration form for STITEK. The form is titled 'Formulir Pendaftaran Mahasiswa Baru'. It contains several input fields: 'Nama Lengkap', 'Email', 'Jenis Kelamin' (with radio buttons for 'Pria' and 'Wanita'), 'Alamat Lengkap', and a 'Hobi' section with checkboxes for 'Membaca', 'Olahraga', 'Musik', and 'Traveling'. There is also a 'Pilihan Jurusan' dropdown menu. At the bottom of the form is a blue button labeled 'Daftar Sekarang'. Below the form, there is a link that says 'Untuk informasi lebih lanjut, kunjungi website kampus kami'.

## 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 susunan 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 *stylizing* harus disertakan ke dalam markup HTML. Itu berarti 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 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** di-load 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.

## 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:

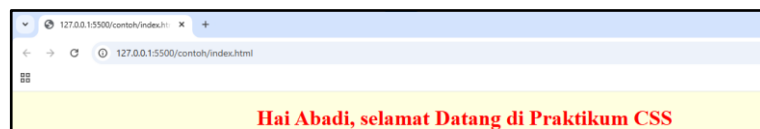
### 1. Penulisan CSS (*Cascading Style Sheets*)

#### 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

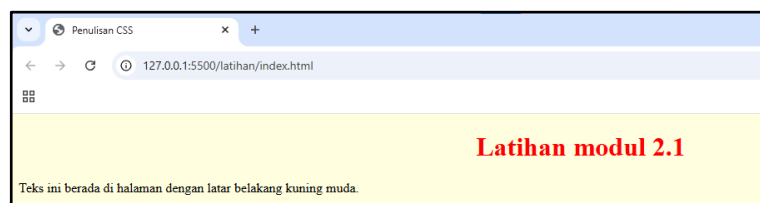


#### 2) Latihan

##### 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

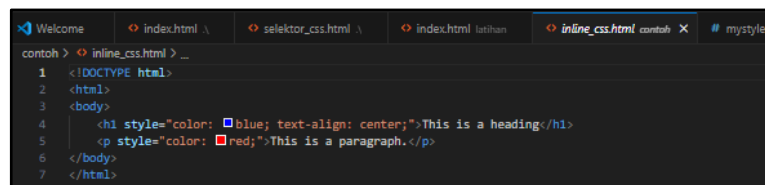
Perbedaan antara kedua kode HTML tersebut dapat dilihat dari beberapa aspek. Pertama, pada kode pertama, judul halaman (title) tidak didefinisikan, sedangkan pada kode kedua, judul halaman ditetapkan sebagai "Penulisan CSS". Kedua, konten yang ditampilkan di dalam body juga berbeda: kode pertama menampilkan 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



```
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



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <style>
5     body {
6       background-color: #ffff00;
7     }
8     h1 {
9       color: #ff0000;
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

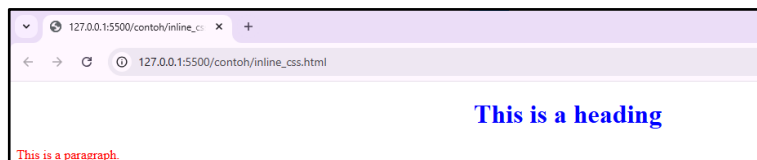
```
contoh > external_css.html > _
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <link rel="stylesheet" href="mystyle.css">
5 </head>
6 <body>
7   <h1>This is a heading</h1>
8   <p>This is a paragraph.</p>
9 </body>
10 </html>
```

## Mystyle

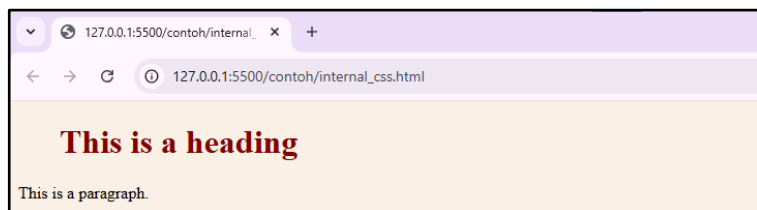
```
contoh > # mystyle.css > h1
1 body {
2   background-color: lightblue;
3 }
4
5 h1 {
6   color: navy;
7   margin-left: 20px;
8 }
```

### b. Output

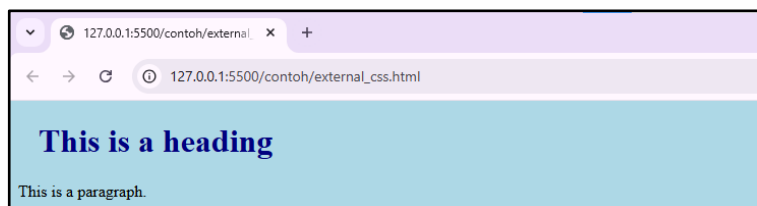
#### Inline



#### Internal



#### External



## 2) Latihan

### a. Input

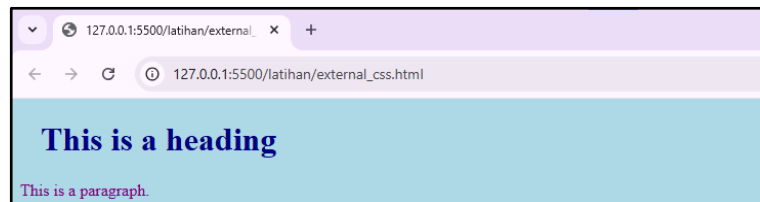
```
latihan > external_css.html > html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <link rel="stylesheet" href="mystyle.css">
5 </head>
6 <body>
7   <h1>This is a heading</h1>
8   <p>This is a paragraph.</p>
9 </body>
10 </html>
```

```

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.

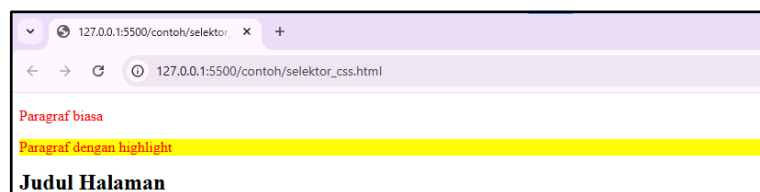
### 3. Selector CSS

#### 1) Contoh

##### a. Input

```
contoh > <? selector_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

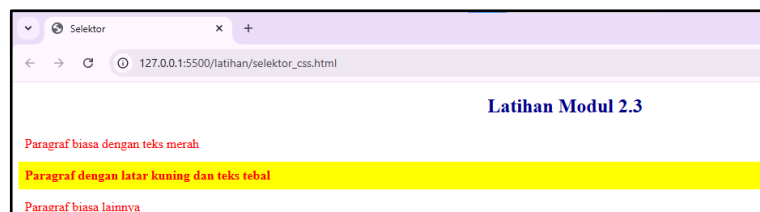


## 2) Latihan

### a. Input

```
latihan > selector_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: blue;
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 spesifisitas. 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: 10px;
20        border-radius: 5px;
21        max-width: 600px;
22        margin: 10px auto;
23        line-height: 1.5;
24    }
25 </style>
26 </head>
27 <body>
28     <p>Nama saya Hani</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 estetik 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.

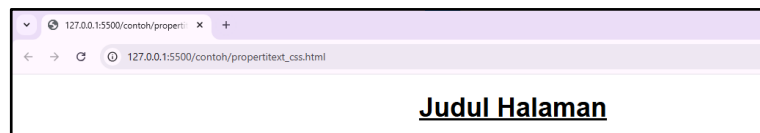
## 5. Properti Teks dan Font

### 1) Contoh

#### a. Input

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

#### b. Output



### 2) Latihan

#### a. Input

```
latihan > propertitext_css.html > html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>propertitext</title>
5 <style>
6   /* gaya untuk judul */
7   h1 {
8     font-family: Georgia, serif; /* font berbeda dari default */
9     font-size: 48px; /* ukuran besar */
10    color: #4682B4;
11    text-align: center;
12    margin-bottom: 30px;
13  }
14
15  /* gaya untuk paragraf */
16  p {
17    font-family: Arial, sans-serif; /* Rata kanan */
18    text-align: right;
19    text-decoration: underline; /* Garis bawah */
20    line-height: 1.6;
21    padding: 10px;
22    margin: 20px;
23    border-right: 3px solid #4682B4; /* Garis vertikal di kanan */
24    padding-right: 15px;
25  }
26 </style>
27 </head>
28 <body>
29 <h1>Latihan 2.4</h1>
30
31 <p>
32   Ini adalah contoh paragraf dengan teks rata kanan dan memiliki garis bawah. Format ini sering digunakan untuk penekanan khusus pada teks.
33 </p>
34
35 <p>
36   Paragraf kedua dengan format yang sama. Perhatikan bahwa seluruh teks dalam
37   paragraf ini rata ke sisi kanan dan bergaris bawah.
38 </p>
39 </body>
40 </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.

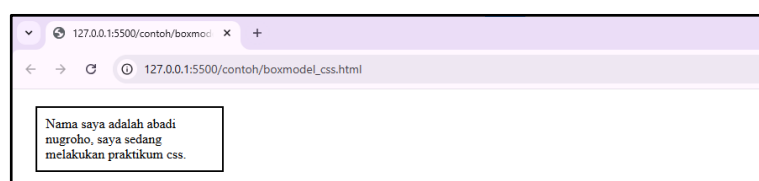
## 6. Box Model (Margin, Border, Padding, dan Content)

### 1) Contoh

#### a. Input

```
contoh > 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



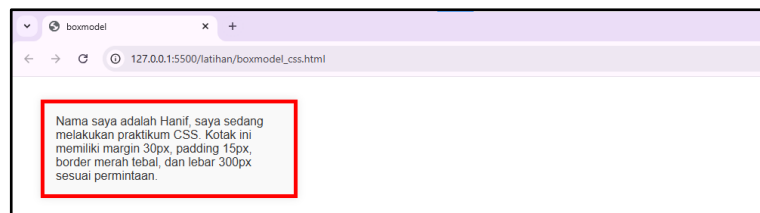


## 2) Latihan

### 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

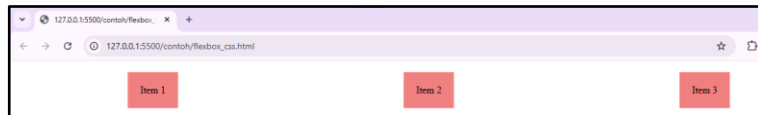
#### a. Input

```

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
11     .item {
12       background-color: #lightcoral;
13       padding: 20px;
14       margin: 10px;
15     }
16   </style>
17 </head>
18 <body>
19   <div class="container">
20     <div class="item">Item 1</div>
21     <div class="item">Item 2</div>
22     <div class="item">Item 3</div>
23   </div>
24 </body>
25 </html>

```

#### b. Output



### 2) Latihan

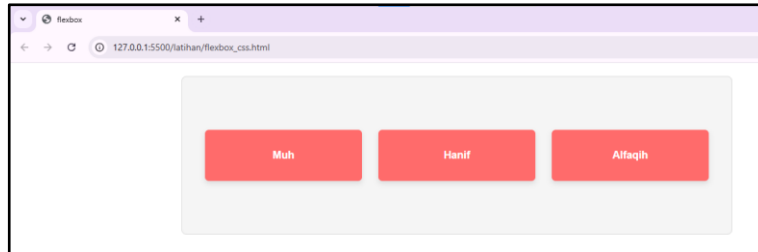
#### a. Input

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Flexbox</title>
5   <style>
6     .container {
7       display: flex;
8       justify-content: space-around; /* Penyebaran 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: 8px; /* Sudut melengkung */
14      padding: 20px; /* Sudut melengkung */
15      margin: 20px auto; /* Posisi horizontal di tengah */
16      max-width: 800px; /* Lebar maksimum */
17    }
18
19    .item {
20      background-color: #fff9c4; /* Warna lebih modern */
21      color: #333; /* Warna teks */
22      padding: 30px 20px; /* Padding atas-bawah lebih besar */
23      margin: 10px; /* Sudut melengkung */
24      border-radius: 6px; /* Sudut melengkung */
25      text-align: center; /* Teks di tengah */
26      font-family: Arial, sans-serif;
27      font-weight: bold;
28      box-shadow: 0 4px 8px rgba(0,0,0,0.1); /* 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">Muhammad</div>
37     <div class="item">Hanif</div>
38     <div class="item">Alfaqih</div>

```

## 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.

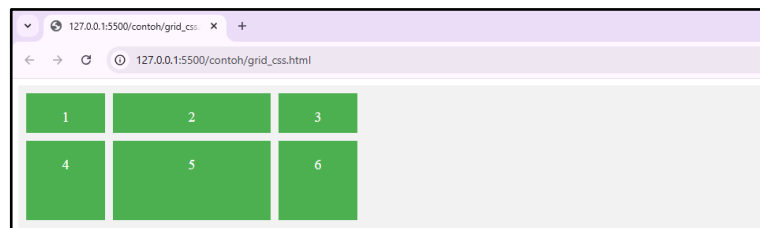
## 8. CSS Grid Layout

### 1) Contoh

#### a. Input

```
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 100px;
9       gap: 10px;
10      background-color: #f2f2f2;
11      padding: 10px;
12    }
13    .grid-item {
14      background-color: #808080;
15      color: white;
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

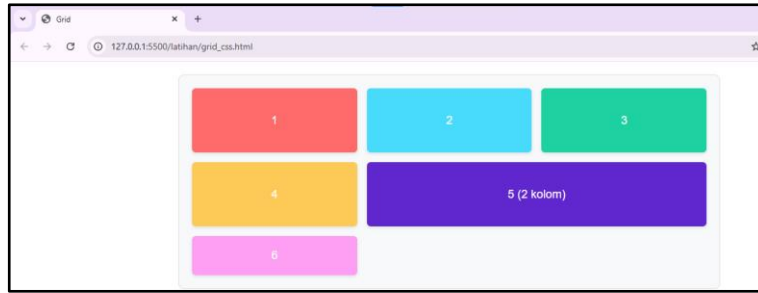


### 2) Latihan

#### a. Input

```
1 <doctype html>
2 <html>
3 <head>
4   <title>Grid</title>
5   <style>
6     .grid-container {
7       display: grid;
8       grid-template-columns: repeat(3, 1fr); /* 3 kolom dengan lebar sama */
9       grid-template-rows: repeat(2, 100px); /* 2 baris dengan tinggi 100px */
10      gap: 10px;
11      background-color: #f2f2f2;
12      padding: 10px;
13      min-width: 300px;
14      margin: 10px auto;
15      border: 1px solid #ccc;
16      border-radius: 5px;
17    }
18    .grid-item {
19      color: white;
20      text-align: center;
21      padding: 20px;
22      font-size: 18px;
23      font-family: Arial, sans-serif;
24      display: flex;
25      align-items: center;
26      justify-content: center;
27      border-radius: 5px;
28      min-height: 80px;
29    }
30    /* Warna latar belakang untuk setiap item */
31    .item1 { background-color: #ffcccc; }
32    .item2 { background-color: #ccccff; }
33    .item3 { background-color: #ffcccc; }
34    .item4 { background-color: #ccccff; }
35    .item5 { background-color: #ffcccc; }
36    .item6 { background-color: #ccccff; }
37    /* Item yang menempati dua kolom */
38    .item7 {
39      grid-column: span 2; /* Menempati 2 kolom */
40    }
41  </style>
42 </head>
43 <body>
44   <div class="grid-container">
45     <div class="grid-item">1</div>
46     <div class="grid-item">2</div>
47     <div class="grid-item">3</div>
48     <div class="grid-item">4</div>
49     <div class="grid-item">5</div>
50     <div class="grid-item">6</div>
51     <div class="grid-item">7</div>
52   </div>
53 </body>
54 </html>
```

## 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



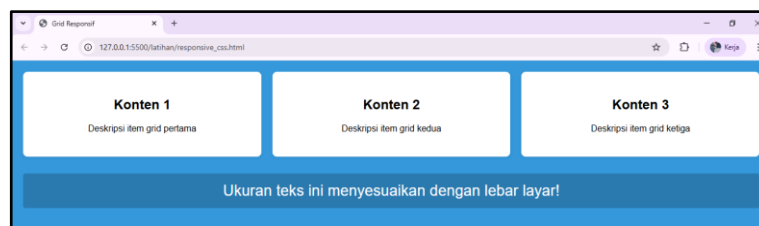
## 2) Latihan

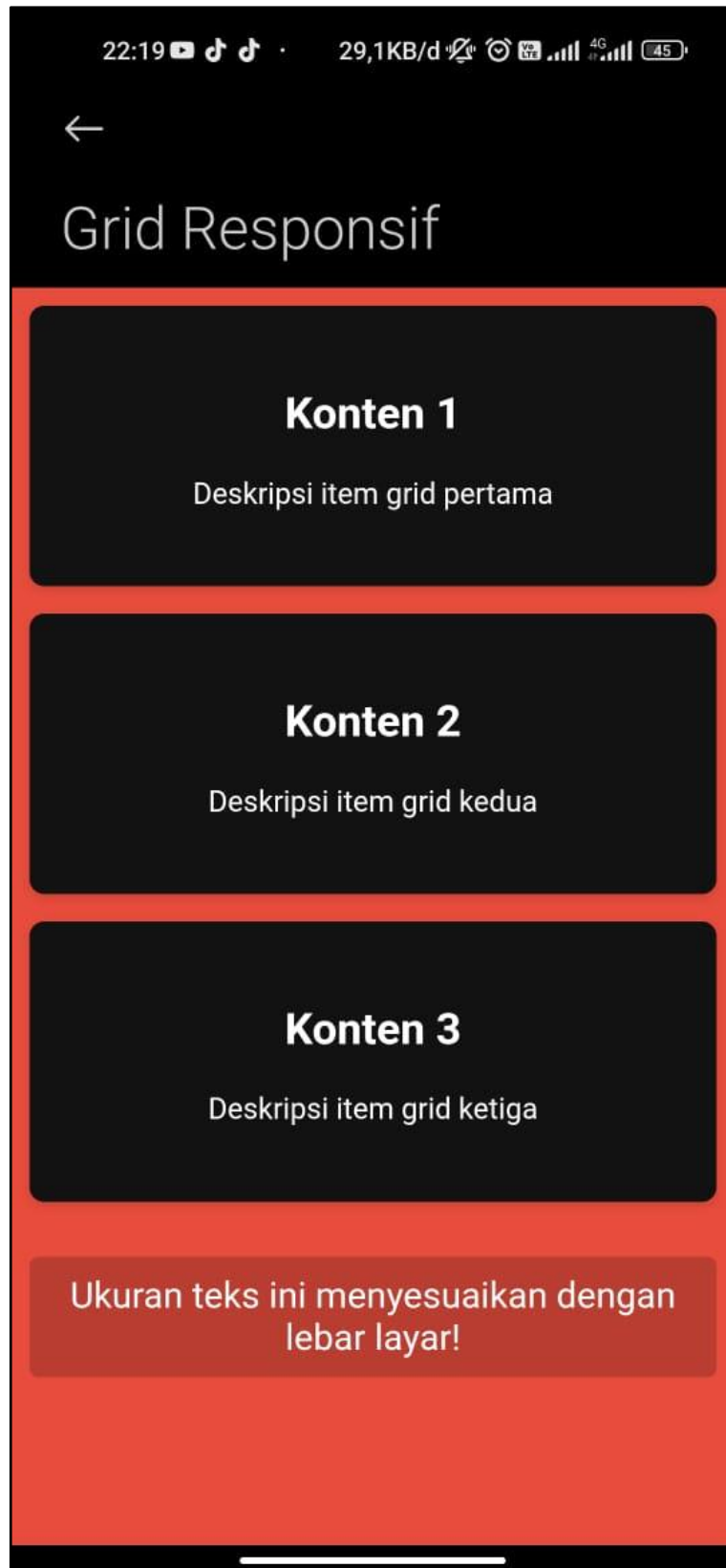
### a. Input

```
latihan > 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: 5px;
27       padding: 25px;
28       box-shadow: 0 2px 5px rgba(0,0,0,0.3);
29       text-align: center;
30     }
31
32     .responsive-text {
33       font-size: 20px; /* Ukuran teks responsif */
34       color: white;
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     /* Media query untuk mobile (screenshot) */
43     @media screen and (max-width: 600px) {
44       body {
45         background-color: #e74c3c; /* Merah untuk mobile */
46         padding: 10px;
47       }
48
49       .grid-container {
50         grid-template-columns: 1fr; /* 1 kolom untuk mobile */
51         gap: 10px;
52       }
53
54       .responsive-text {
55         font-size: 30px; /* Ukuran lebih besar di mobile */
56         padding: 10px;
57       }
58     }
59   </style>
60 </head>
61 <body>
62   <div class="grid-container">
63     <div class="grid-item">
64       <h2>Konten 1</h2>
65       <p>Deskripsi item grid pertama</p>
66     </div>
67     <div class="grid-item">
68       <h2>Konten 2</h2>
69       <p>Deskripsi item grid kedua</p>
70     </div>
71     <div class="grid-item">
72       <h2>Konten 3</h2>
73       <p>Deskripsi item grid ketiga</p>
74     </div>
75   </div>
76
77   <div class="responsive-text">
78     Ukuran teks ini menyesuaikan dengan lebar layar!
79   </div>
80 </body>
81 </html>
```

### b. Output

#### Desktop







### 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:

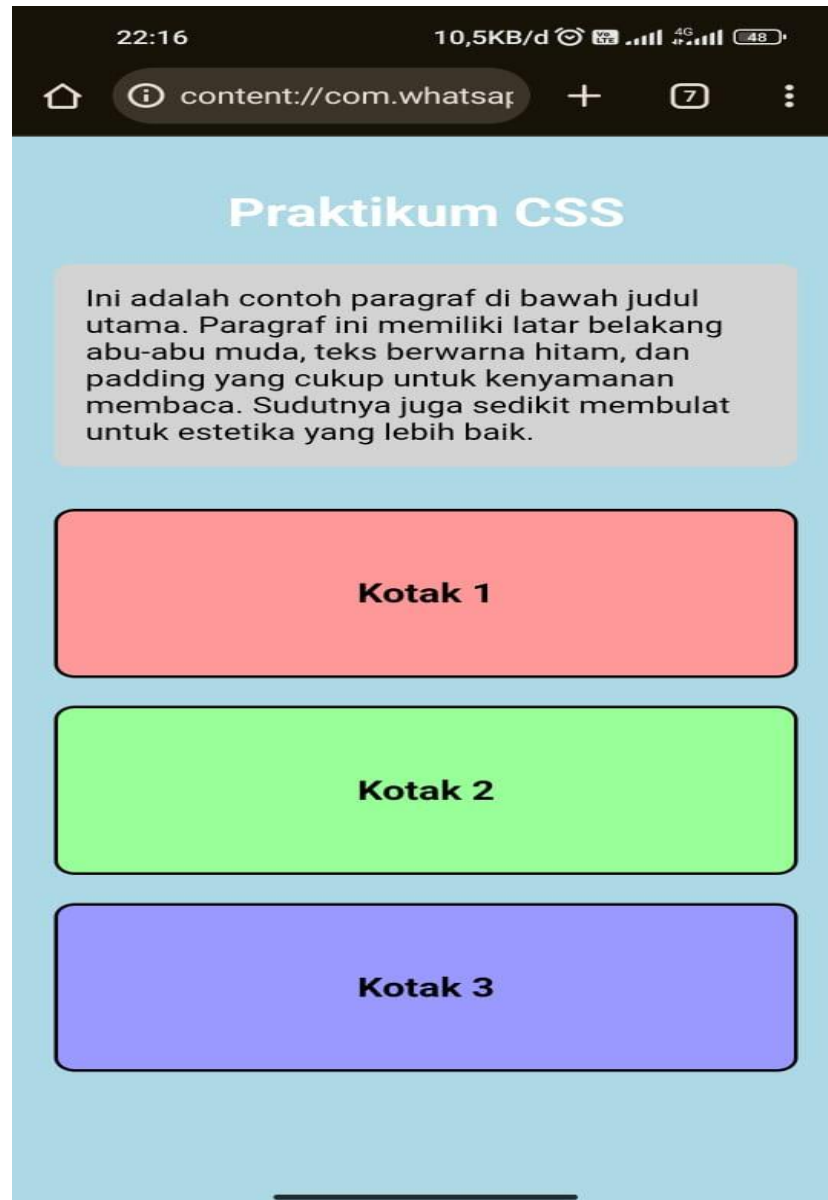
a. Input

```
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: #f0f0f0;
10      font-family: Arial, sans-serif;
11      margin: 0;
12      padding: 20px;
13      box-sizing: border-box;
14    }
15
16    h1 {
17      color: #333;
18      font-size: 48px;
19      text-align: center;
20      margin-bottom: 20px;
21    }
22
23    p {
24      background-color: #f0f0f0;
25      color: #333;
26      padding: 10px;
27      margin: 0 auto;
28      width: 80%;
29      border-radius: 8px;
30      border: 1px solid #ccc;
31      box-sizing: border-box;
32    }
33
34    .container {
35      display: grid;
36      grid-template-columns: repeat(2, 1fr);
37      gap: 20px;
38      max-width: 800px;
39      margin: 0 auto;
40      box-sizing: border-box;
41    }
42
43    .box {
44      border: 1px solid #ccc;
45      padding: 10px;
46      height: 100px;
47      box-sizing: border-box;
48      border-radius: 8px;
49      display: flex;
50      justify-content: center;
51      align-items: center;
52      font-weight: bold;
53      font-size: 1.2em;
54    }
55  </style>
56</head>
57<body>
58  <h1>Praktikum CSS</h1>
59  <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>
60  <div class="container">
61    <div class="box box1">Kotak 1</div>
62    <div class="box box2">Kotak 2</div>
63    <div class="box box3">Kotak 3</div>
64  </div>
65</body>
66</html>
```

```
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
```

b. Output





2. Link Github

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

## **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].