



# MODUL

AMSP  
2024

# PEMROGRAMAN WEBSITE DASAR

TETAP JAYA DALAM TANTANGAN

## **MODUL FRONT-END PEMROGRAMAN WEB**



### **Disusun Oleh :**

Wahyuni Edsa Safira	(220209502074)
Azzah Ulima Rahma	(220209500016)
Athiyyah Anandira	(220209500030)
Aan Alamsyah	(220209501108)
Muhammad Fadhil Hani	(220209502059)
Andi Asmul Akzam	(220209502033)
Muhammad Adrian	(220209502060)
Nur Aisyah Bahtiar	(220209502051)
Muhammad Syafaad Jufri	(220209502079)
Muhammad Zulfahrizi Zubair	(220209502080)

**PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER  
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI MAKASSAR**

**2024**

## **BAB I**

### **PENGENALAN PEMROGRAMAN WEB DAN FRONT-END**

#### **A. Tujuan**

1. Siswa mampu menjelaskan konsep dasar pemrograman web, termasuk definisi, tujuan, dan jenis-jenis situs web (statis dan dinamis).
2. Siswa mampu mengidentifikasi dan memahami fungsi dari komponen utama dalam pengembangan web, seperti Web Editor, Web Browser, Web Server, dan Database Server.
3. Siswa dapat membedakan antara client-side scripting dan server-side scripting, serta memahami contoh-contoh dari masing-masing jenis scripting.
4. Siswa mampu menjelaskan peran dan tanggung jawab seorang Front-End Developer, serta menguasai dasar-dasar HTML, CSS, dan JavaScript, termasuk penggunaan framework seperti jQuery atau Bootstrap.

#### **B. Teori**

##### **1. Pengenalan Pemrograman Web**

Pemrograman web diambil dari 2 suku kata yaitu pemrograman dan web. Pemrograman yang dalam bahasa English adalah programming dan diartikan proses, cara, perbuatan program (secara bahasa indonesia). Definisi Web: jaringan komputer yang terdiri dari kumpulan situs internet yang menawarkan teks dan grafik dan suara dan sumber daya animasi melalui protokol transfer hypertext.

Web programming adalah proses pembuatan program berupa aplikasi yang berisikan dokumen-dokumen multimedia (text, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol http (hypertext transfer protocol) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut browser.

World Wide Web adalah layanan internet yang paling populer saat ini internet dan dikenal juga sebagai halaman-halaman website yang dapat saling terkoneksi satu dengan lainnya (hyperlink) yang membentuk samudra belantara informasi. WWW berjalan dengan protokol HyperText Transfer Protokol (HTTP).

Situs/web dapat dikategorikan menjadi dua yaitu web statis dan web dinamis.

##### **a) Web Statis**

Web statis adalah web yang berisi/menampilkan informasi-informasi yang sifatnya statis (tetap). Pada sebuah web statis, pengguna hanya dapat melihat isi dokumen pada halaman web dan apabila diklik akan berpindah ke halaman web yang lain.

Interaksi pengguna hanya terbatas dapat melihat informasi yang ditampilkan, tetapi tidak dapat mengolah informasi yang dihasilkan. Web statis biasanya merupakan HTML yang ditulis pada editor teks dan disimpan dalam bentuk .html atau .htm.

b) Web Dinamis

Web dinamis adalah sebuah web yang menampilkan informasi serta dapat berinteraksi dengan pengguna. Web yang dinamis memungkinkan pengguna untuk berinteraksi menggunakan form sehingga dapat mengolah informasi yang ditampilkan web dinamis bersifat interaktif, tidak kaku, dan terlihat lebih indah.

Secara umum pemrograman web dapat dibagi 2, yaitu: clientside scripting dan server side scripting.

- a) Client side scripting (CSS), merupakan jenis script yang pengolahannya dilakukan di sisi client. pengolahan disini berarti “di terjemahkan/ interpreted”. yang memiliki tugas untuk menterjemahkan script jenis ini disisi client adalah web browser. agar semua script yang masuk kategori ini dapat diterjemahkan oleh web browser maka didalam web browser terdapat sebuah komponen/ modul/ “engine” yang memiliki daftar pustaka (library) yang mampu mengenali semua perintah-perintah yang terdapat pada kategori client side scripting. berikut contoh web browser yang populer digunakan: internet explorer, mozilla firefox, opera., safari. berikut adalah contoh client side scripting: HTML (hypertext markup language), java script, XML (extensible markup language), CSS (cascading style sheet). client side scripting merupakan script yang digunakan untuk membuat halaman web statis.
- b) Server side scripting (SSS), merupakan script yang pengolahannya (baca: diterjemahkan) di sisi server. server yang dimaksud disini adalah sebuah komponen yang biasa disebut web server yang didalamnya terintegrasi dengan sebuah mesin (engine)/ modul yang didalamnya terdapat daftar pustaka (library) yang mampu menterjemahkan script-script tersebut yang kemudian setelah di terjemahkan di server kemudian dikirim (ditampilkan) ke client (web browser) dalam format HTML (hypertext markup language).mesin (engine) tersebut harus di install terlebih dahulu karena mesin-mesin tersebut memiliki karakteristik yang berbeda untuk tiap script (bahasa pemrograman) yang digunakan. berikut contoh server side scripting: PHP (Hypertext Pre Processor), ASP (active server pages), JSP (java server pages). server side scripting digunakan untuk membuat halaman web dinamis. hal ini disebabkan karena server side scripting memiliki kelebihan, diantaranya: mampu berinteraksi dengan banyak perangkat lunak basis data

(database), mampu mengelola sumber daya yang terdapat di sistem operasi dan perangkat keras komputer, mampu dijalankan di semua sistem operasi (multi platform), aman karena scriptnya di proses di server yang tidak dapat dilihat dari sisi client.

Adapun komponen penyusun web, yaitu:

- a) Web Editor merupakan program aplikasi yang berfungsi untuk mengetikkan perintah-perintah dokumen web baik client side scripting maupun server side scripting.
- b) Web Browser Merupakan program yang berfungsi untuk menampilkan dokumen-dokumen web dalam format HTML.
- c) Web Server adalah program aplikasi yang memiliki fungsi sebagai tempat menyimpan dokumen-dokumen web.
- d) Database Server adalah program yang digunakan untuk menyimpan data yang akan diolah di halaman web. Database biasa disebut Database Management System (DBMS) adalah sebuah aplikasi yang mengembangkan user dan data-data di dalam database.

## **2. Pengenalan Front-End**

Front-End merupakan istilah yang mengacu pada developer yang bekerja dibagian desain layout suatu website atau aplikasi, dimana seorang developer akan mengerjakan user interface agar desain terlihat menarik dan tidak terlihat membosankan untuk dilihat. Seorang Front-End developer wajib menguasai program HTML, CSS, Javascript dan juga framework seperti jQuery, Bootstrap dan lain sebagainya.

## **BAB II**

### **HTML**

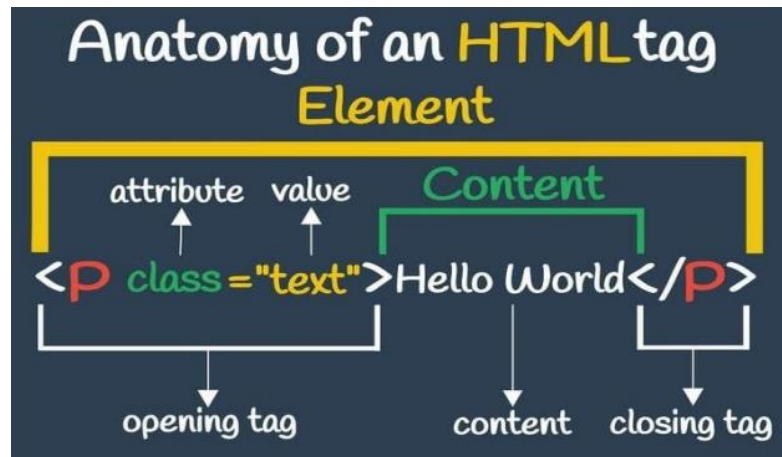
#### **A. Tujuan**

1. Siswa mampu menjelaskan pengertian HTML, fungsinya dalam pengembangan web, serta memahami struktur dasar dokumen HTML beserta komponen utamanya, seperti tag, elemen, dan atribut.
2. Siswa dapat mengidentifikasi dan menuliskan elemen-elemen dasar HTML, serta memahami fungsinya masing-masing dalam membangun halaman web.
3. Siswa mampu mengenali dan menggunakan berbagai tipe elemen HTML umum.
4. Siswa mampu menjelaskan dan menerapkan atribut-atribut pada elemen HTML.
5. Siswa mampu mengaplikasikan pengetahuan tentang elemen-elemen HTML dan atributnya dalam membuat halaman web yang terstruktur dan fungsional.

#### **B. Teori**

##### **1. Pengenalan HTML (HyperText Markup Language)**

HTML (HyperText Markup Language) adalah bahasa standar yang digunakan untuk membuat dan merancang halaman web. Dalam dunia web development, HTML berfungsi sebagai kerangka dasar yang mendefinisikan struktur dan isi dari sebuah halaman web. HTML berbasis pada teks mode, sehingga penulisannya menggunakan program-program yang berbasis pada teks, seperti Notepad, Textpad, Wordpad, dan lain-lain. Pada HTML, tag merupakan instruksi atau perintah dalam HTML, yaitu teks khusus berupa dua karakter “<” dan “>”. Adapun elemen yaitu komponen yang menyusun dokumen HTML, dibentuk dari tag pembuka, isi, dan tag penutup. Atribut merupakan kata khusus yang berada di dalam tag pembuka. Atribut juga disebut sebagai modifier yang akan menentukan perilaku dari elemen. Suatu atribut dapat ditambahkan ke elemen-elemen HTML. Suatu atribut memiliki nilai yang dapat diisi kesetiap elemen yang menggunakannya.



## 2. Elemen Dasar HTML

Sebelum membuat kode bahasa HTML sebaiknya dipersiapkan juga aplikasi web browser yang seperti internet explorer (browser) bisa juga Chrome, Mozilla, Netscape, Opera dan yang lainnya untuk menjalankan HTML. Adapun elemen dasar bahasa HTML, yaitu sebagai berikut.

Doctype	Doctype adalah perintah yang menentukan versi HTML yang digunakan. Untuk HTML 5, doctype ditulis sebagai berikut: <code>&lt;!DOCTYPE html&gt;</code>
HTML	Tag <code>&lt;html&gt;&lt;/html&gt;</code> menandai bahwa konten yang ada didalamnya adalah konten HTML.
Head	Tag <code>&lt;head&gt;&lt;/head&gt;</code> menyimpan informasi tentang halaman web, seperti judul, deskripsi, dan referensi file lain yang digunakan.
Title	Tag <code>&lt;title&gt;&lt;/title&gt;</code> menentukan judul halaman web. Judul ini akan ditampilkan di tab browser atau jendela browser.
Body	Tag <code>&lt;body&gt;&lt;/body&gt;</code> menyimpan konten utama dari halaman web.

Penulisan elemen dasar HTML adalah sebagai berikut :

```
<html>
<head><title> ..... </title> </head>
  <body>
    .....
  </body>
</html>
```

**Catatan :**

Setiap elemen dasar diawali dengan tanda pembuka < > dan diakhiri dengan tanda penutup sesuai jenis elemen.

### 3. Tipe-tipe Elemen HTML

Berikut adalah beberapa tipe elemen HTML yang umum digunakan:

a. Heading (<h1> sampai <h6>)

Elemen heading digunakan untuk membuat judul atau subjudul. Tingkatan heading lebih rendah memiliki ukuran teks yang lebih kecil.

Contoh:

```
<h1>Judul Utama</h1>
<h2>Judul Sub</h2>
<h3>Judul Sub Sub</h3>
```

Output:

**Judul Utama**

**Judul Sub**

**Judul Sub Sub**

b. Paragraf (<p>)

Elemen paragraf digunakan untuk membuat paragraf teks.

Contoh:

```
<p>Hello World!</p>
```

Output:

Hello World!



c. Garis Horizontal <hr>

Tag <hr> mendefinisikan pembuatan garis horizontal, dan merupakan elemen kosong tanpa tag penutup.

Contoh:

```
<h1>Judul Utama</h1>
<hr>
<h2>Judul Sub</h2>
```

Output:

**Judul Utama**

---

**Judul Sub**

d. Baris Baru <br>

Tag <br> mendefinisikan jeda baris, dan merupakan elemen kosong tanpa tag penutup.

Contoh:

```
<p>Hello World! <br> Today is the day!</p>
```

Output:

Hello World!  
Today is the day!

e. Tautan (<a>)

Elemen tautan digunakan untuk membuat hyperlink. Atribut yang biasa dipakai ialah href yaitu atribut yang menentukan alamat atau lokasi dari elemen HTML.

Contoh:

```
<a href="https://www.example.com/">Kunjungi Situs Web Kami</a>
```

Output:

[Kunjungi Situs Web Kami](https://www.example.com/)

f. Gambar (<img>)

Elemen gambar digunakan untuk menampilkan gambar. Atribut yang sering dipakai di <img> adalah src yaitu input gambar dari lokasi penyimpanan gambar, alt (alternative text) yaitu deskripsi dari gambar untuk membantu pengguna yang tidak

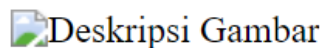
dapat melihat gambar, width merupakan lebar dari gambar dan height merupakan panjang dari gambar. Jika ukuran width dan height tidak dituliskan maka ukuran gambar yang ditampilkan adalah ukuran sebenarnya dari file gambar tersebut.

Contoh:

```

```

Output:



g. Form (<form>)

Elemen form digunakan untuk membuat formulir input, seperti formulir kontak atau login. Formulir memungkinkan pengguna untuk mengirimkan data ke server web, seperti mengisi formulir pendaftaran atau pencarian. Pada form terdapat elemen <label> yang memberikan teks, "Nama:" sebagai label untuk input, dan terdapat atribut for yang menghubungkan label dengan input yang memiliki id="nama". Elemen selanjutnya ialah <input type="text">, yaitu kotak teks untuk memasukkan nama, id berupa identitas unik untuk input, dan name berupa nama yang digunakan saat data dikirim. Adapun elemen terakhir yaitu <input type="submit"> sebagai tombol untuk mengirimkan form, dan value="Kirim" untuk menampilkan teks "Kirim" pada tombol.

Contoh:

```
<form>
  <label for="nama">Nama:</label>
  <input type="text" id="nama" name="nama">
  <input type="submit" value="Kirim">
</form>
```

Output:

Nama:

h. Tabel (<table>)

Elemen tabel digunakan untuk membuat tabel data di halaman web. Tabel terdiri dari baris dan kolom, dengan elemen-elemen tambahan seperti <tr>(baris tabel), <th>(sel kepala tabel), dan <td>(sel data) yang digunakan untuk mengatur konten tabel. Tabel berguna untuk menampilkan data dalam format terstruktur dan rapi.

Contoh:

```
<table border="1">
  <tr>
    <th>Nama</th>
    <th>Umur</th>
  </tr>
  <tr>
    <td>Kim Namjoon</td>
    <td>30</td>
  </tr>
  <tr>
    <td>Jeon Jungkook</td>
    <td>27</td>
  </tr>
</table>
```

Output:

Nama	Umur
Kim Namjoon	30
Jeon Jungkook	27

Adapun beberapa atribut dari tabel yaitu sebagai berikut.

Atribut	Fungsi
<i>Border</i>	Menentukan ukuran <i>border</i> /garis tabel
<i>Width</i>	Menentukan lebar tabel
<i>Height</i>	Menentukan tinggi tabel
<i>Bgcolor</i>	Menentukan <i>background</i> tabel
<i>Background</i>	Menentukan gambar yang digunakan untuk <i>background</i> tabel
<i>Color</i>	Untuk mengatur warna suatu sel dalam tabel
<i>Align</i>	Mengatur bentuk perataan horisontal
<i>Valign</i>	Mengatur bentuk perataan vertikal
<i>Rowspan</i>	Menggabungkan beberapa baris
<i>Colspan</i>	Menggabungkan beberapa kolom
<i>Cellspacing</i> dan <i>cellpadding</i>	Mengatur spasi antar sel dan spasi dalam sel

i. Iframe (<iframe>)

Elemen digunakan untuk menanamkan dokumen HTML atau sumber eksternal ke dalam halaman web saat ini. Iframe dapat menampilkan konten dari situs web lain atau dokumen HTML terkait di dalam halaman web. Iframe sering digunakan untuk menanamkan peta, video, atau konten lainnya dari sumber eksternal.

Contoh:

```
<iframe  
  src=https://www.smkmuhda.sch.id/  
  name="smkmuhda"  
  title="smkmuhda">  
</iframe>
```

Output:



j. List

List atau daftar dalam HTML digunakan untuk mengorganisasi dan menyajikan informasi dalam format yang terstruktur:

- Unordered List (Daftar Tidak Berurutan): Digunakan untuk menyajikan elemen dalam urutan yang tidak teratur, seperti dengan simbol bullet points atau tanda centang.

Contoh:

```
<ul>  
  <li>Mobil</li>  
  <li>Sepeda</li>  
  <li>Truk</li>  
</ul>
```

Output:

- Mobil
  - Sepeda
  - Truk
- Ordered List (Daftar Berurutan): Menyajikan elemen dalam urutan tertentu, seperti menggunakan nomor atau huruf.

Contoh:

```
<ol>
  <li>Mobil</li>
  <li>Sepeda</li>
  <li>Truk</li>
</ol>
```

Output:

1. Mobil
2. Sepeda
3. Truk

- Definition List (Daftar Definisi): Digunakan untuk menyajikan pasangan kata kunci (term) dan definisi yang sesuai. Tag <dl> digunakan untuk mendefinisikan daftar deskripsi, tag <dt> digunakan untuk mendefinisikan istilah (nama), dan tag <dd> digunakan untuk mendeskripsikan setiap istilah.

Contoh:

```
<dl>
  <dt>Mobil</dt>
  <dd>- Kendaraan roda empat</dd>
  <dt>Sepeda</dt>
  <dd>- Kendaraan roda dua</dd>
</dl>
```

Output:

- Mobil
- Kendaraan roda empat
- Sepeda
- Kendaraan roda dua

#### k. Komentar

Dalam sebuah pengembangan halaman web, komentar dapat diberikan atau digunakan untuk memberikan penjelasan atau keterangan mengenai script yang dibuat programmer. Untuk membuat komentar, menggunakan tag, diawali dengan <!-- dan

diakhiri dengan -->. Komentar tidak ditampilkan oleh browser, tetapi dapat membantu mendokumentasikan kode sumber HTML.

Contoh:

```
<!-- tuliskan komentar disini -->
```

## BAB III

### CSS

#### A. Tujuan

1. Siswa memahami peran dan jenis-jenis CSS.
2. Siswa menulis kode CSS menggunakan selector, property, dan value.
3. Siswa mengatur elemen dengan Box Model (content, padding, border, margin).
4. Siswa mengenal dan menggunakan framework CSS dasar.
5. Siswa mengelola margin, padding, dan posisi elemen pada web.
6. Siswa mengatur font, ukuran, dan efek teks pada web.
7. Siswa mengatur warna dan gambar latar pada web.
8. Siswa menggunakan DevTools untuk memperbaiki kode CSS.

#### B. Teori

1. Pengenalan CSS (Cascading Style Sheet)

Cascading Style Sheet (CSS) adalah suatu bahasa yang bekerja sama dengan dokumen HTML untuk mendefinisikan cara bagaimana suatu isi halaman web ditampilkan atau dipresentasikan. Presentasi ini meliputi style atau gaya teks, link, maupun tata letak (layout) halaman. Dengan adanya teknologi seperti ini, kita dapat memilah atau memisahkan antara kode untuk isi halaman web dan kode yang diperlukan khusus untuk menangani tampilan.

Saat ini masih sering dijumpai para pengembang web yang menggunakan tag `<table>` untuk mengatur layout atau tata letak dari halaman web. Ada juga yang menggunakan CSS, tetapi kodenya hanya diperoleh dari internet (hasil download) tanpa mengetahui lebih dalam akan arti dan makna dari kode-kode CSS tersebut. Hal-hal semacam ini disebabkan karena minimnya pengetahuan mereka tentang CSS sehingga tidak tahu harus memulai dari mana cara membuat halaman web yang memanfaatkan teknologi CSS. Berikut adalah sintaks dasar dari CSS, yaitu:

```
selector {  
Property: value;  
}
```

Sintaks adalah pengaturan kata dan simbol dalam urutan tertentu untuk membuat ekspresi atau pernyataan yang memiliki struktur yang sesuai. Sintaks CSS terdiri dari:

- a. Pemilih (selector) digunakan untuk memberi tahu CSS elemen mana yang akan ditata
- b. Properti (property) digunakan untuk memberi tahu CSS apa yang akan diubah
- c. Nilai (value) digunakan untuk penunjuk bagaimana kita akan mengubah elemen

Adapun CSS frameworks yaitu sekumpulan kode CSS atau library yang sudah ada dan siap digunakan. CSS frameworks menyediakan berbagai macam elemen desain seperti: sistem grid, pola user interface (UI) yang interaktif, tipografi, tombol, ikon dan lain-lain. Adapun beberapa contoh CSS Frameworks, yaitu sebagai berikut.

- a. Bootstrap: pertama kali diciptakan oleh salah satu desainer dan developer Twitter pada tahun 2011
- b. Foundation: hadir sejak 2011 dibawah lisensi MIT (Massachusetts Institute of Technology)
- c. Tailwind: kode-kodenya muda dikostumisasi dan sudah terintegrasi dengan vue JS, react JS, dan HTML. Dapat diandalkan untuk menciptakan UI yang cepat dan praktis.
- d. Materialize: framework yang dirancang oleh perusahaan Google. Digunakan oleh banyak pengembang untuk mendesain website atau aplikasi android.

## 2. Penggunaan CSS

Terdapat tiga cara penggunaan CSS, yaitu sebagai berikut.

### a. Inline CSS

Inline CSS digunakan untuk memberikan style pada tag atau elemen HTML tertentu. Atribut yang digunakan ialah style yang memberikan style pada tag HTML, dan style hanya berlaku pada tag HTML tersebut. Contoh:

```
<html>
  <body>
    <h1 style="color:red;text-align:center;">Judul</h1>
    <p style="color:purple;">Paragraf</p>
  </body>
</html>
```

Output:



# Judul

Paragraf

## b. Internal CSS

Internal CSS digunakan untuk memberi style pada satu laman web. Style yang diberikan diapit oleh tag `<style></style>` dan diletakkan di bagian elemen head atau tag `<head>` pada sebuah laman web. Contoh:

```
<html>
<head>
<style>
body {
  background-color: gray;
}

p {
  color: purple;
}
</style>
</head>
<body>
  <p>Aku suka warna ungu</p>
</body>
</html>
```

Output:

Aku suka warna ungu

## c. Eksternal CSS

Eksternal CSS adalah file berekstensi `.css` yang khusus memuat kode-kode style. Tujuannya adalah untuk memisahkan kode style dengan struktur dan tipe konten pada laman web. Kelebihannya, eksternal CSS dapat diaplikasikan pada seluruh laman web. Sehingga semua laman web memiliki tampilan yang seragam.

Eksternal CSS dapat terhubung ke tag HTML dengan menggunakan tag link href yang dapat memanggil file eksternal CSS. File eksternal dapat ditulis dalam editor teks apa pun, dan harus disimpan dengan ekstensi .css. File .css eksternal tidak boleh berisi tag HTML apa pun. Contoh:

index.html

```
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>Judul</h1>
<p>Paragraf</p>

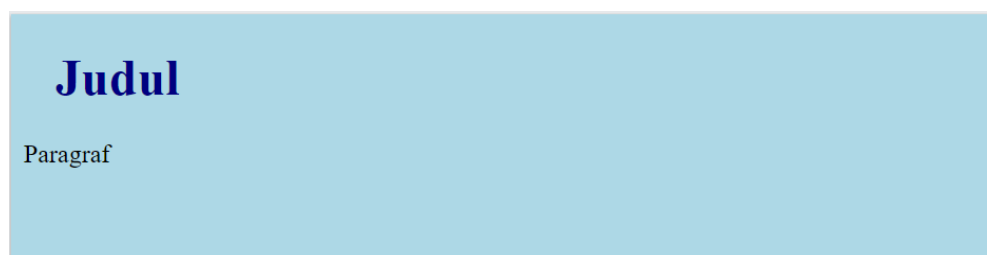
</body>
</html>
```

mystyle.css

```
body {
  background-color: lightblue;
}

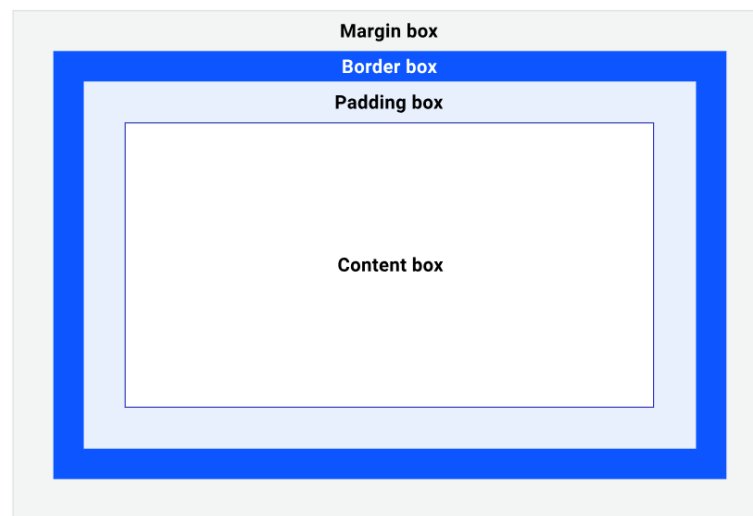
h1 {
  color: navy;
  margin-left: 20px;
}
```

Output:



### 3. Box Model

CSS Box Model adalah konsep dasar dalam CSS yang menjelaskan bagaimana elemen di dalam halaman web direpresentasikan sebagai kotak. Setiap elemen memiliki bentuk kotak yang terdiri dari beberapa area: content, padding, border, dan margin. Jika dianalogikan sebagai tiga bingkai foto yang tergantung di dinding, dimana content box adalah karya seni, padding box adalah area putih di antara karya seni dan bingkai, border box adalah bingkai itu sendiri, serta margin box adalah ruang antara bingkai satu dengan yang lain.



#### Area Box Model

- Content Box:** Tempat konten elemen (teks, gambar, dll.). Ukuran konten mempengaruhi ukuran keseluruhan box.
- Padding Box:** Ruang antara content box dan border box. Padding menambah ruang di dalam box dan dapat diberi warna latar.
- Border Box:** Batas yang mengelilingi content dan padding. Border bisa berwarna dan memiliki gaya tertentu.
- Margin Box:** Ruang di luar box model yang memisahkan elemen satu dengan yang lain. Margin tidak memiliki warna dan mengatur jarak antar elemen.

Box model yang dibuat harus di debugging, yaitu proses mengidentifikasi dan menghapus bug atau error di dalam kode. Debugging memiliki beberapa peran yaitu:

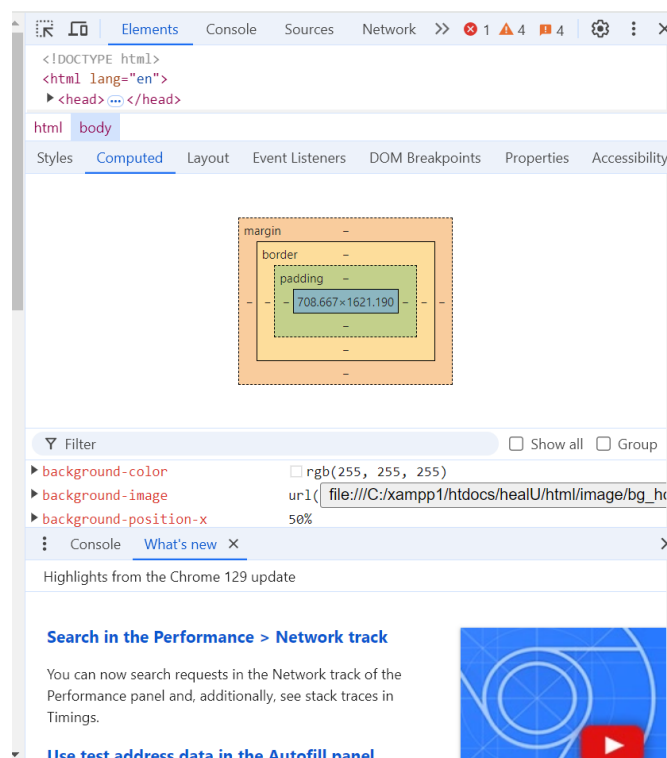
- Eror dapat dideteksi lebih awal
- Proses perbaikan dapat dilakukan sedini mungkin
- Menghindari kesalahan desain program lebih lanjut
- Dapat memberikan informasi struktur data

- e. Membantu developer mengurangi informasi yang tidak berguna
- f. Mengurangi resiko hacker yang menyusup celah bug atau eror
- g. Developer dapat menghindari proses testing yang rumit sehingga menghemat waktu dan energi ketika coding

Disetiap browser terdapat Browser DevTools yang menyediakan visualisasi perhitungan box model yang dipilih. Sehingga dapat membantu dalam memahami cara kerja box model.

DevTools dapat dibuka dengan cara klik icon titik tiga atau menu Chrome yang berada di pojok kanan atas. Setelah itu pilih More tools, lalu pilih Developer tools. Cara lain untuk mengakses DevTools juga bisa dengan klik kanan pada mouse kemudian pilih Inspect. Selain itu, menekan shortcut Ctrl+Shift+I juga bisa digunakan untuk mengakses DevTools di Windows. Setelah muncul tampilan DevTools, dapat dilihat pada sebelah kiri merupakan HTML dari file yang sedang dikerjakan. Kemudian di sebelah kanan terdapat browser DevTools.

Dengan DevTools dapat memilih berbagai tampilan seperti tampilan di desktop maupun di mobile. Kemudian dengan klik icon tanda panah, dapat memilih elemen yang akan diinspeksi. Sehingga bisa melihat detail-detail pada padding, margin, dan lain-lain. Pada tab Style, kita dapat mengatur, mengubah, mengedit bentuk box model. Sehingga dapat mengira-ngira dan memiliki gambaran mengenai tampilan dari website yang sedang didevelop.



Dalam mengontrol CSS Box Model di browser, ada beberapa hal yang perlu diperhatikan:

- a. Stylesheet Default Browser: Setiap browser memiliki gaya bawaan yang diterapkan melalui stylesheet agen pengguna. Misalnya, Mozilla sering kali memiliki margin yang lebih besar dibandingkan Chrome.
- b. Display Default:
  - Elemen `<div>` secara default adalah block, yang berarti mengisi seluruh lebar halaman.
  - Elemen `<span>` secara default adalah inline, yang berarti hanya mengisi lebar sebesar kontennya.
- c. Inline Block: Elemen dengan display inline-block menggabungkan sifat inline dan block. Elemen ini memiliki margin seperti elemen block, tetapi ukurannya disesuaikan dengan konten di dalamnya.
- d. Pengaturan Padding dan Border: Secara default, elemen menggunakan box-sizing: content-box, yang berarti padding dan border ditambahkan ke ukuran konten, sehingga memperbesar ukuran total elemen.

#### 4. Spacing

HTML menyediakan berbagai cara untuk mengatur spasi pada elemen-elemen. Dua elemen yang sering digunakan adalah `<br>` dan `<hr>`. Margin digunakan untuk menambahkan ruang di luar elemen. Properti margin mengatur spasi di sisi atas, bawah, kiri, dan kanan elemen. Urutan penulisan margin mengikuti akronim TRouBLe (Top, Right, Bottom, Left), yaitu searah jarum jam:

- a. Satu value: Spasi seragam untuk semua sisi (contoh: margin: 20px).
- b. Dua value: Spasi untuk atas & bawah, serta kiri & kanan (contoh: margin: 20px 40px).
- c. Tiga value: Spasi untuk atas, kiri & kanan, dan bawah (contoh: margin: 20px 40px 30px).
- d. Empat value: Spasi individual untuk atas, kanan, bawah, dan kiri (contoh: margin: 20px 40px 30px 10px).

Padding menciptakan ruang di bagian dalam elemen, antara konten dan batas (border). Sama seperti margin, padding juga dapat menggunakan satu hingga empat value:

- a. Satu value: Diterapkan untuk semua sisi (contoh: padding: 20px).

- b. Dua value: Untuk atas & bawah, serta kiri & kanan (contoh: padding: 20px 40px).
- c. Tiga value: Untuk atas, kiri & kanan, dan bawah (contoh: padding: 20px 40px 30px).
- d. Empat value: Untuk atas, kanan, bawah, dan kiri (contoh: padding: 20px 40px 30px 10px).

HTML dan CSS juga memungkinkan pengaturan posisi elemen menggunakan properti top, right, bottom, dan left. Elemen dapat memiliki berbagai jenis posisi:

- a. Relative: Elemen tetap berada dalam flow dokumen, tetapi diposisikan relatif terhadap tempatnya saat ini.
- b. Absolute: Elemen diposisikan berdasarkan parent yang memiliki posisi relative.
- c. Fixed: Elemen diposisikan relatif terhadap view port (area tampilan layar), dan akan tetap berada di posisi yang sama saat pengguna menggulir layar.
- d. Sticky: Elemen ini akan tetap berada di posisinya saat pengguna menggulir sampai mencapai titik tertentu, lalu "menempel" di tempat itu.

## 5. Font-Syling

Memilih font yang tepat memiliki dampak yang sangat besar terhadap pengalaman pembaca di sebuah situs web. Font yang tepat dapat menciptakan identitas yang kuat untuk web yang dibuat. Menggunakan font yang mudah dibaca adalah hal yang penting. Font menambah nilai pada teks. Penting juga untuk memilih warna dan ukuran teks yang tepat untuk font.

### a. Font family

Dalam CSS, kita menggunakan font-family properti untuk menentukan fonta suatu teks. Properti font-family menerima daftar string yang dipisahkan koma, baik mengacu pada font-family spesifik atau generik. Font-family spesifik adalah string yang dikutip contohnya seperti “Helvetica”, atau “Times New Roman”. Font-family generik adalah kata kunci seperti serif, sans-serif, monospace, cursive dan fantasy.

Pada website, tidak ada font yang 100% aman untuk web. Selalu ada kemungkinan font tidak ditemukan atau tidak terpasang dengan benar. Oleh karena itu, sangat penting untuk selalu menggunakan font fallback, yaitu dengan cara harus menambahkan daftar "font cadangan" yang serupa di font-family properti. Jika font pertama tidak berfungsi, browser akan mencoba font berikutnya, dan berikutnya lagi, dan seterusnya. Selalu akhiri daftar dengan nama

keluarga font generik. Berikut ini adalah daftar font web aman terbaik untuk HTML dan CSS:

- Arial (sans-serif)
- Verdana (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)
- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)
- Courier New (monospace)
- Brush Script MT (cursive)

Contoh:

```
<html>
  <head>
    <style>
      .p1 {
        font-family: "Times New Roman", Times, serif;
      }
      .p2 {
        font-family: Arial, Helvetica, sans-serif;
      }
      .p3 {
        font-family: "Lucida Console", "Courier New", monospace;
      }
    </style>
  </head>
  <body>
    <p class="p1">Times New Roman font.</p>
    <p class="p2">Arial font.</p>
    <p class="p3">Lucida Console font.</p>
  </body>
</html>
```

Output:

Times New Roman font.

Arial font.

Lucida Console font.

b. Font size

Properti font-size digunakan untuk mengontrol ukuran elemen teks. Properti ini menerima nilai panjang, persentase, dan juga kata kunci absolut (xx-small, x-small, small, medium, large, x-large, xx-large) serta beberapa nilai kata kunci relatif (smaller, larger) yang relatif terhadap ukuran font elemen induk. Jika tidak menggunakan font-size, ukuran default untuk teks normal, seperti paragraf, adalah 16px (16px=1em). Nilai font-size dapat berupa ukuran absolut atau relatif.

Ukuran absolut:

- Mengatur teks ke ukuran tertentu
- Tidak mengizinkan pengguna untuk mengubah ukuran teks di semua browser (buruk untuk alasan aksesibilitas)
- Ukuran absolut berguna ketika ukuran fisik output diketahui

Ukuran relatif:

- Mengatur ukuran relatif terhadap elemen di sekitarnya
- Memungkinkan pengguna untuk mengubah ukuran teks di browser

Untun mengatur font-size bisa dengan:

- Pixel yang dapat memberi kontrol penuh atas ukuran teks
- Em yang memungkinkan pengguna mengubah ukuran teks sebagai pengganti pixel. Ukurannya dapat dihitung dari pixel ke em menggunakan rumus:  $\text{pixel}/16 = \text{em}$
- Persen(%) yang memberikan solusi di semua browser dengan menetapkan font-size default untuk elemen <body>
- Viewport (vw) yang dapat mengikuti ukuran jendela browser. 1 vw = 1% dari lebar viewport. Jika lebar viewport 50 cm, 1 vw adalah 0,5 cm.

Contoh:

```
<html>
```



```
<head>
  <style>
    body {
      font-size: 100%;
    }
    h2 {
      font-size: 40px;
    }
    p {
      font-size: 1.875em;
    }
  </style>
</head>
<body>
  <h2>This is heading 1</h2>
  <p>This is heading 2</p>
</body>
</html>
```

Output:

# This is heading 1

## This is heading 2

### c. Font weight

Properti font-weight digunakan untuk mengatur ketebalan teks. Properti ini menerima value kata kunci (normal, bold), value kata kunci relatif (lighter, bolder), dan value kata kunci numerik (100 sampai 900). Kata kunci normal dan bold masing-masing setara dengan nilai numerik 400 dan 700.

Contoh:

```
<p style="font-weight: normal">Normal</p>
<p style="font-weight: lighter">Lighter</p>
<p style="font-weight: bold">Bold</p>
```

```
<p style="font-weight: 900">900</p>
```

Output:

Normal

Lighter

**Bold**

**900**

d. Font variant

Dalam font huruf kecil, semua huruf kecil dikonversi menjadi huruf besar. Namun, huruf besar yang dikonversi muncul dalam ukuran huruf yang lebih kecil daripada huruf besar asli dalam teks. Properti font-variant menentukan apakah teks harus ditampilkan dalam huruf kecil atau tidak.

Contoh:

```
<p style="font-variant: normal">variant normal.</p>
```

```
<p style="font-variant: small-caps">variant small caps</p>
```

Output:

variant normal.

VARIANT SMALL CAPS

e. Font style

Properti font-style digunakan untuk mengatur apakah teks harus miring atau tidak. Font-style menerima salah satu kata kunci berikut: normal, italic, dan oblique.

Contoh:

```
<p style="font-style: normal">Normal</p>
```

```
<p style="font-style: italic">Italic</p>
```

```
<p style="font-style: oblique">Oblique</p>
```

Output:

Normal

*Italic*

*Oblique*

f. Line height

Properti line-height menentukan tinggi spasi antar baris. Di properti ini nilai negatif tidak diperbolehkan.

Contoh:

```
<h1>line-height: 1.6 (recommended):</h1>  
<div style="line-height: 1.6">This is a paragraph with the recommended line-height.<br>  
The line height is here set to 1.6. This is a unitless value;<br>  
meaning that the line height will be relative to the font size.</div>
```

Output:

## line-height: 1.6 (recommended):

This is a paragraph with the recommended line-height.  
The line height is here set to 1.6. This is a unitless value;  
meaning that the line height will be relative to the font size.

g. Font google

Google Fonts gratis untuk digunakan dan memiliki lebih dari 1000 font untuk dipilih. Untuk menggunakannya tambahkan tautan lembar gaya khusus di bagian <head> lalu rujuk font di CSS. Ketika menentukan font dalam CSS, selalu cantumkan minimal satu font cadangan (untuk menghindari perilaku yang tidak diharapkan). Jadi, di sini Anda juga harus menambahkan keluarga font generik (seperti serif atau sans-serif) di akhir daftar. Untuk menggunakan beberapa font Google, cukup pisahkan nama font dengan karakter pipa (|). Untuk mengaktifkan effect pada font, pertama-tama tambahkan effect=effectname ke Google API, lalu tambahkan nama kelas khusus ke elemen yang akan menggunakan efek khusus. Nama kelas selalu dimulai dengan font-effect- dan diakhiri dengan nama efek.

Contoh:

```
<html>  
  <head>
```

```

<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Sofia">
<style>
  body {
    font-family: "Sofia", sans-serif;
  }
</style>
</head>
<body>
  <h1>Sofia Font</h1>
  <p>Lorem ipsum dolor sit amet.</p>
  <p>123456790</p>
</body>
</html>

```

Output:

# Sofia Font

Lorem ipsum dolor sit amet.

123456790

## 6. Text-Styling

CSS memiliki banyak properti untuk memformat teks.

### a. Color

Properti warna digunakan untuk mengatur warna teks. Biasa warna text default untuk halaman ditentukan di body selector. Warna tersebut ditentukan oleh:

- nama warna seperti “red”
- nilai HEX seperti #ff0000”
- nilai RGB seperti “rgb(255,0,0)”

Contoh:

```

<html>
  <head>

```

```
<style>
body {
  color: blue;
}
h1 {
  color: green;
}
</style>
</head>
<body>
  <h1>Judul</h1>
  <p>Paragraf pertama</p>
  <p>Paragraf kedua</p>
</body>
</html>
```

Output:

**Judul**

Paragraf pertama

Paragraf kedua

#### b. Text align

Properti text-align digunakan untuk menentukan perataan horizontal teks dalam blok atau elemen sel tabel. Properti ini menerima nilai kata kunci left, right start, end, center, dan justify.

Contoh:

```
<p style="text-align: center">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
<p style="text-align: left">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
```

ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>

<p style="text-align: start">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>

<p style="text-align: right">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>

<p style="text-align: end">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>

<p style="text-align: justify">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>

## Output:

### Center

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

### Left

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

### Start

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

### Right

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

### End

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

### Justify

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

c. Text indent

Properti ini digunakan untuk menambahkan indentasi ke blok teks. Properti ini membutuhkan panjang (misalnya 10px, 2em) atau persentase dari lebar blok yang berisi teks.

Contoh:

```
<p style="text-align: justify; text-indent: 3em";>  
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod  
    tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim  
    veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea  
    commodo consequat.  
</p>
```

Output:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

d. Text decoration

Properti text-decoration digunakan untuk menambahkan garis ke teks. Garis bawah paling sering digunakan, tetapi mungkin juga menambahkan garis di atas teks atau bahkan menembus teksnya. Kita juga bisa menggunakan ketiganya sekaligus.

Contoh:

```
<h1 style="text-decoration: overline">Overline text decoration</h1>  
<h2 style="text-decoration: line-through">Line-through text decoration</h2>  
<h3 style="text-decoration: underline">Underline text decoration</h3>  
<p style="text-decoration: overline-underline">Overline and underline text  
decoration.</p>
```

Output:

**Overline text decoration**

**~~Line-through text decoration~~**

**Underline text decoration**

**Overline and underline text decoration.**

e. Text transform

Properti text-transform digunakan untuk mengubah kapitalisasi teks tanpa perlu mengubah HTML yang mendasarinya. Properti ini menerima nilai kata kunci berikut: uppercase, lowercase, dan capitalize.

Contoh:

```
<p style="text-transform: uppercase">Contoh paragraf uppercase</p>
<p style="text-transform: lowercase">Contoh paragraf lowercase.</p>
<p style="text-transform: capitalize">Contoh paragraf capitalize</p>
```

Output:

CONTOH PARAGRAF UPPERCASE

contoh paragraf lowercase.

Contoh Paragraf Capitalize

f. Text overflow

Properti text-overflow digunakan untuk menentukan bagaimana konten tersembunyi direpresentasikan. Terdapat dua opsi, pertama clip (default) yang memotong teks pada batas, yang kedua elipsis.

Contoh:

```
<html>
  <head>
    <style>
      div.a {
        white-space: nowrap;
        width: 50px;
        overflow: hidden;
        text-overflow: clip;
      }
      div.b {
        white-space: nowrap;
        width: 50px;
        overflow: hidden;
```



```
        text-overflow: ellipsis;
    }
</style>
</head>
<body>
    <h2>text-overflow: clip (default):</h2>
    <div class="a">Hello world!</div>
    <h2>text-overflow: ellipsis:</h2>
    <div class="b">Hello world!</div>
</body>
</html>
```

Output:

## **text-overflow: clip (default):**

Hello w

## **text-overflow: ellipsis:**

Hell...

### g. Text shadow

Properti text-shadow digunakan untuk menambahkan bayangan pada teks. Properti ini menggunakan tiga dimensi panjang (x-offset, y-offset, dan blur-radius) dan warna.

Contoh:

```
<h1 style="color: white; text-shadow: 2px 2px 8px #000000">Text-shadow on  
white text</h1>
```

Output:

**Text-shadow on white text**

#### h. Letter spacing

Properti ini digunakan untuk mengontrol jumlah spasi horizontal antar karakter dalam teks. Properti ini menerima nilai panjang seperti em, px, dan rem.

Contoh:

```
<p style="letter-spacing: 3px">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. </p>
<p style="letter-spacing: 2px">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. </p>
<p style="letter-spacing: -1px">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. </p>
```

Output:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

#### i. Word spacing

Properti ini digunakan untuk menambah atau mengurangi panjang spasi di antara setiap kata dalam teks. Properti ini menerima nilai panjang seperti em, px, dan rem. Untuk nilai word-spacing: 0 sama dengan word-spacing: normal.

Contoh:

```
<p style="word-spacing: 1cm">This is some text. This is some text.</p>
```

Output:

This is some text. This is some text.

### 7. Background

#### a. Background color

Nilai awal background-color atau background warna adalah transparan. Hal itu memungkinkan konten pada parent terlihat. Set warna yang valid pada lapisan latar belakang akan berada di belakang hal-hal lain yang terdapat pada elemen itu. Untuk memberikan background warna, dapat menggunakan beberapa cara

mendeskripsikan value. Dapat menggunakan nama warna, function hsl, hex code, function rgb, dsb.

Contoh:

```
<html>
  <head>
    <style>
      body {
        background-color: coral;
      }
    </style>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

Output:



Hello World!

b. Background image

Di atas lapisan background-color, kita dapat menambahkan background-image atau background gambar. Background gambar menerima beberapa hal berikut:

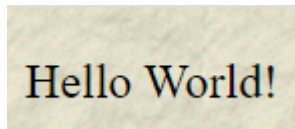
- URL gambar atau URI data menggunakan fungsi CSS url.
- Gambar yang dibuat secara dinamis oleh fungsi CSS gradient.

Contoh:

```
<html>
  <head>
    <style>
      body {
        background-image: url("paper.gif");
      }
    </style>
```

```
</head>
<body>
  <p>Hello World!</p>
</body>
</html>
```

Output:



c. Background position

Properti background-position memungkinkan kita untuk mengubah posisi background gambar. Properti background-position memungkinkan untuk memposisikan gambar di sepanjang sumbu x dan y. Apabila tidak melakukan pengulangan, maka secara default posisi background berada di kiri atas.

Contoh:

```
body {
  background-image: url('w3css.gif');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: center;
}
```

Output:



d. Background size

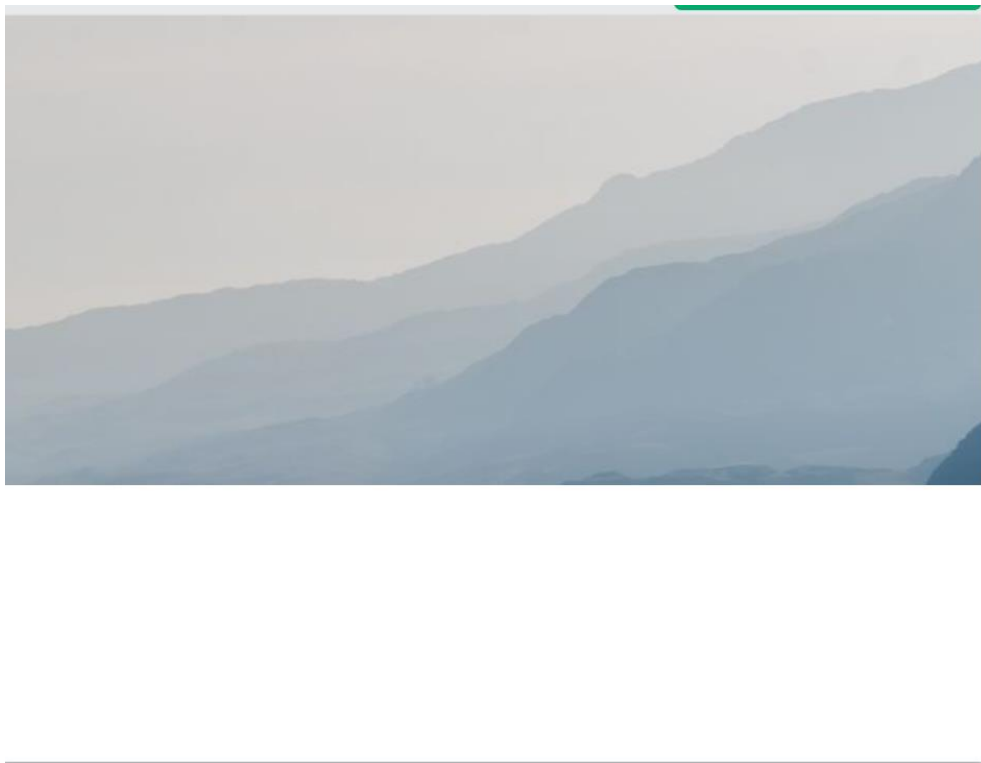
Properti background-size berfungsi untuk mengatur ukuran gambar background. Default-nya, gambar background diukur berdasarkan lebar, tinggi, dan rasio aspek intrinsiknya (aktual). Properti background-size menerima keyword berikut:

- auto: saat digunakan, ukuran gambar background berdasarkan lebar dan tinggi intrinsiknya.
- cover: meliputi seluruh area layer background. Hal ini berarti gambar diregangkan atau dipotong.
- contain: ukuran gambar untuk mengisi ruang tanpa meregangkan atau memotong. Akibatnya ruang kosong dapat tetap ada. Hal itu menyebabkan gambar berulang.

Contoh:

```
background: url(mountain.jpg);  
background-repeat: no-repeat;  
background-size: auto;
```

Output:



e. Background repeat

Secara default, gambar background diulang (repeating background images) secara horizontal dan vertikal untuk mengisi seluruh ruang lapisan latar belakang. Gunakan value di bawah ini untuk menggunakan properti background-repeat:

- repeat: gambar berulang dalam ruang yang tersedia, dipotong seperlunya.
- round: gambar berulang secara horizontal dan vertikal agar sesuai dengan banyak instance ke dalam ruang yang tersedia, tanpa memotong, mengompres, atau diregangkan.
- space: Gambar berulang secara horizontal dan vertikal agar sesuai di dalam ruang yang tersedia tanpa memotong. Gambar berulang menyentuh tepi dengan spasi merata di antara gambar tersebut.

Contoh:

```
<html>
  <head>
    <style>
      body {
        background-image: url("paper.gif");
        background-repeat: space;
      }
    </style>
  </head>
  <body>
    <h1>The background-repeat Property</h1>
  </body>
</html>
```

Output:



8. Selector

Selector dibutuhkan untuk menerapkan CSS ke elemen. CSS memberi beberapa cara berbeda untuk melakukan ini. Aturan CSS merupakan suatu blok kode yang berisi satu selector atau lebih yang didalamnya terdapat satu atau lebih declaration. Declaration

atau deklarasi adalah pasangan property dan value yang menerapkan style ke elemen yang cocok dengan selector. Adapun yang namanya simple selector, yaitu adalah kelompok dari selector yang paling mudah menargetkan elemen HTML. Termasuk selector class, selector ID dan atribut lain yang dapat ditambahkan pada tag HTML.

a. Simple Selector Universal

Selector ini juga dikenal sebagai wildcard. Universal selector (\*) cocok dengan elemen apapun. Dengan selector ini dapat dipilih semua tag HTML yang terdapat pada suatu halaman dan akan menerapkan style yang akan dibuat.

Contoh:

```
<html>
  <head>
    <style>
      * {
        text-align: center;
        color: blue;
      }
    </style>
  </head>
  <body>
    <h1>Hello world!</h1>
    <p>Every element on the page will be affected by the style.</p>
    <p id="para1">Me too!</p>
    <p>And me!</p>
  </body>
</html>
```

Output:

**Hello world!**

Every element on the page will be affected by the style.

Me too!

And me!

b. Simple Selector Type

Type selector adalah sebuah selector yang akan mencocokkan dengan tipe elemen HTML secara langsung. Selector section memberikan style pada semua tag <section>. Sehingga outputnya, tag <section> memiliki style yang sama.

Contoh:

```
<html>
  <head>
    <style>
      section {
        border: 1px solid red;
        margin-bottom: 10px;
      }
    </style>
  </head>
  <body>
    <section >
      <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. </p>
    </section>
    <section >
      <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. </p>
    </section>
  </body>
</html>
```

Output:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

### c. Simple Selector Class

Dengan menggunakan class selector, elemen HTML dapat memiliki satu atau lebih class yang ditentukan dalam atribut class-nya. Selector class cocok dengan elemen



apapun yang menerapkan kelas tersebut di HTML-nya. Setiap class yang dideklarasikan akan ditambahkan style-nya pada tag HTML tersebut. Karakter (.) menginstruksikan bahasa CSS untuk mencocokkan anggota atribut class. Hal tersebut umum ditemukan di CSS. Dimana karakter khusus atau kumpulan karakter digunakan untuk menentukan selector.

Contoh:

```
<html>
  <head>
    <style>
      .class {
        background-color: yellow;
      }
    </style>
  </head>
  <body>
    <div class="class">
      <p>Nama saya adalah Donal.</p>
      <p>Saya tinggal di Kota Bebek.</p>
    </div>
    <p>Sahabat terbaik saya adalah Miki.</p>
    <p class="class">Sahabat terbaik saya adalah Miki.</p>
  </body>
</html>
```

Output:

Nama saya adalah Donal.

Saya tinggal di Kota Bebek.

Sahabat terbaik saya adalah Miki.

Sahabat terbaik saya adalah Miki.

#### d. Simple Selector Id

Dengan menggunakan selector id, elemen HTML dengan atribut id harus menjadi satu-satunya (unique) elemen pada halaman dengan nilai ID tersebut. Berbeda dengan selector class, selector id tidak dapat digunakan secara multiple. Apabila browser menemukan lebih dari satu instance id, itu akan tetap menerapkan aturan

CSS apapun yang cocok dengan selector-nya. Namun elemen apapun yang memiliki atribut id seharusnya hanya dimiliki oleh 1 elemen. Karakter (#) menginstruksikan bahasa CSS untuk mencocokkan anggota atribut id.

Contoh:

```
<html>
  <head>
    <style>
      #nama {
        background-color: yellow;
      }
    </style>
  </head>
  <body>
    <div class="pengenalan">
      <p id="nama">Nama saya adalah Donal.</p>
      <p id="kota">Saya tinggal di Kota Bebek.</p>
    </div>
    <p>Sahabat terbaik saya adalah Miki.</p>
  </body>
</html>
```

Output:

Nama saya adalah Donal.

Saya tinggal di Kota Bebek.

Sahabat terbaik saya adalah Miki.

Dalam CSS juga terdapat grouped selector, yang berarti kita menggabungkan beberapa elemen yang berbeda dalam satu aturan dan juga descendant selector, yang artinya hanya elemen tag html yang berada di dalam elemen dengan suatu class akan diubah menjadi satu aturan.

Contoh:

```
<html>
<head>
  <style>
```

```
p, h2 {
  color: red;
}
.container h2 {
  color: blue;
}
</style>
</head>
<body>
<h2>Elemen h2 di luar container</h2>
<p>Paragraf</p>
<div class="container">
  <h2>Elemen h2 di dalam container</h2>
</div>
<p>Paragraf 2</p>
</body>
</html>
```

Output:

**Elemen h2 di luar container**

Paragraf

**Elemen h2 di dalam container**

Paragraf 2

## 9. Pseudo-Class

Pseudo-class memungkinkan untuk menerapkan CSS berdasarkan perubahan status. Pseudo-class juga adalah Kelas semu yang dimiliki oleh sebuah elemen HTML, yang membuat kita dapat mendefinisikan style pada “keadaan tertentu” dari elemen tersebut. Hal tersebut berarti bahwa desain kita dapat bereaksi terhadap masukan/input pengguna contohnya seperti alamat email yang tidak valid.

a. :link

Pseudo-class :link dapat diterapkan ke elemen <a> apapun yang memiliki nilai href. Pada :link hanya dapat menata color, background-color, border-color, outline-color, dan fill and stroke dari SVG.

Contoh:

```
<html>
<head>
<style>
a:link {
  color: green;
}
</style>
</head>
<body>
<a href="http://www.wikipedia.org">Wikipedia</a>
</body>
</html>
```

Output:

[Wikipedia](http://www.wikipedia.org)

b. :visited

Pseudo-class :visited adalah style ketika sebuah link sudah pernah dikunjungi sebelumnya (menggunakan browser yang sama).

Contoh:

```
<html>
<head>
<style>
a:visited {
  color: pink;
}
</style>
</head>
<body>
<a href="http://www.wikipedia.org">Wikipedia</a>
```

```
</body>
```

```
</html>
```

Output:

Sebelum  
[Wikipedia](#)

Sesudah  
[Wikipedia](#)

c. :hover

Apabila kita memiliki perangkat penunjuk seperti mouse atau trackpad, kemudian kita menempatkannya di atas elemen, kita dapat menghubungkan status tersebut dengan :hover untuk menerapkan style. Hal tersebut merupakan cara yang berguna untuk mengisyaratkan bahwa suatu elemen dapat berinteraksi dengannya.

Contoh:

```
a:hover {  
  background-color: yellow;  
  font-size: 18px;  
}
```

Output:

[Wikipedia](#)

d. :active

Status ini dipicu saat elemen sedang berinteraksi secara aktif seperti klik dan sebelum klik dilepaskan kembali. Jika menggunakan kursor, keadaan ini adalah ketika diklik dan belum dilepaskan.

Contoh:

```
a:active {  
  background-color: yellow;  
}
```

e. :first-child dan :last-child

Apabila kita ingin mencari item pertama atau terakhir, kita dapat menggunakan `:first-child` dan `:last-child`. Pseudo-class ini membantu mengembalikan elemen pertama atau terakhir dalam sekelompok sibling element.

Contoh:

```
<html>
<head>
<style>
li:first-child {
color: red;
}
li:last-child {
color: blue;
}
</style>
</head>
<body>
<ul>
<li>mobil</li>
<li>motor</li>
<li>sepeda</li>
<li>truk</li>
</ul>
</body>
</html>
```

Output:

- mobil
- motor
- sepeda
- truk

f. `:nth-child(n)`

Memilih elemen ke-n dari sebuah parent (elemen pembungkusnya). n bisa berarti urutan (1), (2), .... atau pola (2n), (3n+2), (4n-1) atau ganjil dan genap (even) & (odd).

Contoh:

```
<html>
<head>
<style>
li:nth-child(even) {
color: blue;
}
</style>
</head>
<body>
<ul>
<li>mobil</li>
<li>motor</li>
<li>sepeda</li>
<li>truk</li>
</ul>
</body>
</html>
```

Output:

- mobil
- motor
- sepeda
- truk

g. :first-of-type dan :last-of-type

Pseudo-class yang berhubungan dengan posisi elemen. :first-of-type memilih elemen pertama dari sebuah jenis / tipe tag. :last-of-type memilih elemen terakhir dari sebuah jenis / tipe tag.

## 10. Inheritance

Inheritance dalam CSS adalah mekanisme di mana elemen HTML mewarisi properti dari parent-nya. Properti seperti warna teks (color), jenis huruf (font-family), dan jarak antar teks (line-height) secara otomatis diwariskan, sementara properti lainnya, seperti margin, padding, dan border, tidak diwariskan kecuali kita secara eksplisit mengaturnya. Jika menetapkan gaya pada elemen body, maka semua elemen anak akan mewarisi gaya tersebut, kecuali ada aturan lain yang lebih spesifik.

a. Inherit Keyword

Kata kunci inherit memungkinkan untuk memaksa pewarisan nilai properti dari elemen induk.

Contoh:

```
.class{  
  font-weight: 900;  
}  
  
.class p {  
  font-weight: inherit;  
}
```

b. Initial Keyword

Initial adalah keyword yang digunakan untuk mengembalikan properti ke nilai default bawaannya.

Contoh:

```
<html>  
<head>  
  <style>  
    body {  
      font-family: Arial, sans-serif;  
      color: darkblue;  
    }  
    h1, p {  
      color: green;  
    }  
    p {
```



```
font-family: initial;
color: initial;
}
</style>
</head>
<body>
  <h1>Judul yang di-reset</h1>
  <p>Paragraf pertama.</p>
  <p>Paragraf kedua.</p>
</body>
</html>
```

c. Unset Keyword

Kata kunci unset adalah kombinasi antara inherit dan initial. Jika properti dapat diwariskan, maka unset bertindak seperti inherit. Jika tidak dapat diwariskan, unset bertindak seperti initial. Hal ini berguna untuk mereset properti tanpa harus mengingat apakah properti tersebut dapat diwariskan atau tidak.

Contoh:

```
.class {
  color: red;
  margin: 10px;
}
.class p {
  color: unset;
  margin: unset;
}
```

## **BAB IV**

### **JAVASCRIPT**

#### **A. Tujuan**

1. Siswa memahami cara menulis dan menyisipkan kode JavaScript dalam HTML
2. Siswa dapat mendeklarasikan dan menggunakan variabel dalam JavaScript
3. Siswa mengerti tipe data dan operator dasar dalam JavaScript untuk melakukan perhitungan dan evaluasi logika
4. Siswa mampu membuat fungsi dan menerapkan pengulangan dalam kode JavaScript.
5. Siswa memahami cara memanipulasi elemen HTML dengan DOM serta menggunakan pengkondisian

#### **B. Teori**

##### **1. Dasar Javascript**

JavaScript adalah bahasa pemrograman yang bisa dimasukkan ke dalam halaman web dan digunakan untuk membuat situs lebih interaktif. Contohnya, JavaScript memungkinkan kita membuat form atau formulir di situs web yang bisa langsung diproses tanpa harus memuat ulang halaman.

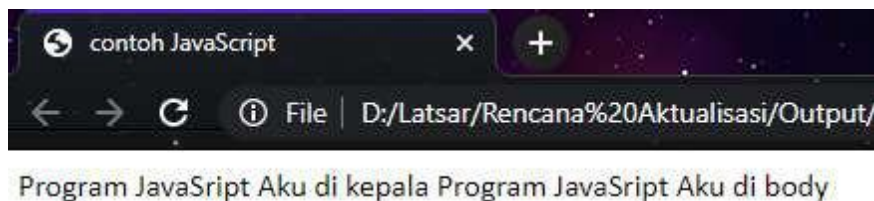
Bahasa ini tidak membutuhkan program khusus untuk menjalankannya, cukup dengan menggunakan browser.

JavaScript juga disebut "case sensitive", yang berarti huruf besar dan huruf kecil pada kode memiliki arti berbeda. Setiap perintah dalam JavaScript diakhiri dengan tanda titik koma (;). Untuk menambahkan catatan atau komentar dalam kode, kita bisa menggunakan dua garis miring (//) untuk komentar singkat atau tanda /\* dan \*/ untuk komentar yang lebih panjang. Kode JavaScript biasanya ditulis di dalam tag `<script>`. Tag ini dapat ditempatkan di bagian kepala (antara `<head>` dan `</head>`) atau di badan dokumen (antara `<body>` dan `</body>`) dari halaman web. Cara yang umum yang dipakai adalah menuliskannya dalam tag `<script>`. Tag `<script>` bisa dibuat di dalam tag `<head>`, maupun di dalam tag `<body>`. Contoh:

```
<HTML>
  <HEAD>
    <TITLE> contoh JavaScript</TITLE>
    <script language="JavaScript"> document.write("Program JavaScript Aku di
kepala");
```

```
</script>
</HEAD>
<BODY>
  <script language="JavaScript"> document.write("Program JavaSript Aku di
body");
  </script>
</BODY>
</HTML>
```

Output:



Menuliskan kode program JavaScript dalam suatu file teks dan kemudian file teks yang berisi kode JavaScript di panggil dari dalam dokumen HTML

```
<SCRIPT LANGUAGE= Javascript SRC= url/file.js > </SCRIPT>
```

Dimana url/file.js adalah lokasi dan nama file yang berisi kode JavaScript, jika perintah tambahan SRC tidak disertakan maka tag Script akan mencari kode yang terletak di dalam tag Script.

JavaScript memiliki peran yang sangat penting dalam pengembangan web modern. Beberapa fungsi utamanya meliputi:

- JavaScript memungkinkan pembuatan elemen-elemen halaman web yang dapat berinteraksi dengan pengguna. Ini termasuk tombol yang merespons klik, formulir yang dapat divalidasi secara real-time, dan animasi yang membuat situs web lebih menarik.
- JavaScript memungkinkan pengembang untuk mengakses dan memanipulasi elemen-elemen DOM. Dengan demikian, Kita dapat mengubah teks, atribut, atau bahkan menambahkan atau menghapus elemen dengan JavaScript.
- JavaScript memungkinkan pengembang mengirim permintaan ke server dan menerima data secara asinkron melalui AJAX (Asynchronous JavaScript and XML). Ini memungkinkan pengembangan aplikasi web yang responsif tanpa perlu me-refresh seluruh halaman.

- d. Saat pengguna mengisi formulir, JavaScript dapat digunakan untuk memeriksa apakah data yang dimasukkan sesuai dengan format yang diharapkan sebelum data dikirim ke server.

## 2. Variabel

Dalam JavaScript, variabel adalah sebuah wadah atau tempat yang digunakan untuk menyimpan data. Variabel digunakan untuk menyimpan informasi yang dapat diubah atau diakses dalam skrip JavaScript. Data yang disimpan dalam variabel bisa berupa angka, teks, objek, atau tipe data lainnya. Dengan menggunakan variabel, Kita dapat menyimpan, mengolah, dan memanipulasi data dalam program JavaScript. Cara mendefinisikan variabel dalam JavaScript adalah dengan menggunakan kata kunci `var`, `let`, atau `const`, diikuti dengan nama variabel.

- a. `var` digunakan untuk mendeklarasikan variabel dengan cakupan (scope) fungsi atau global. Namun, variabel `var` memiliki cakupan yang lebih luas, yang dapat menyebabkan masalah tertentu dalam kode.
- b. `let` digunakan untuk variabel dengan cakupan blok (block scope). Ini adalah pilihan yang lebih modern dan aman dibandingkan dengan `var`.
- c. `const` digunakan untuk variabel yang nilainya tidak akan berubah setelah diberikan nilai awal. Variabel ini bersifat konstan.

Contoh:

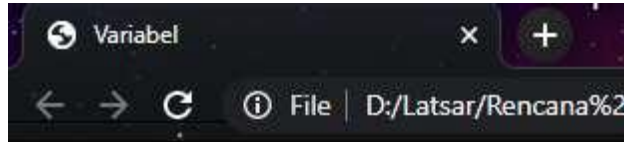
```
var namaVariabel; // Variabel dengan kata kunci var
let usia; // Variabel dengan kata kunci let
const pi = 3.14; // Variabel konstan dengan kata kunci const
```

Jika dideklarasikan dibagian awal dari skrip program, yang artinya sebelum pendeklarasian semua fungsi, maka semua fungsi di dalam program bisa mengakses variabel ini, dan variabel ini menjadi variabel global. Contoh:

```
<SCRIPT language= Javascript >
var a = 12;
var b = 4;
function Perkalian_Dengan2(b) {
a = b * 2;
return a;
}
```

```
document.write( Dua kali dari ,b, adalah ,Perkalian_Dengan2(b));  
document.write( Nilai dari a adalah ,a);  
</SCRIPT>
```

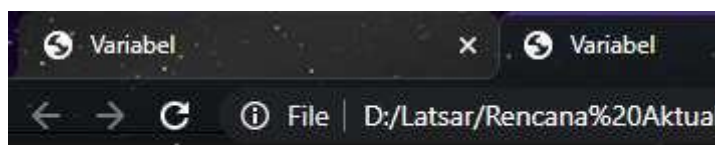
Output:



Jika dia deklarasikan dengan menggunakan kata kunci var di dalam suatu fungsi tertentu, maka variabel itu hanya bisa di akses dari dalam fungsi tersebut, dan artinya variabel ini tidak berguna bagi fungsi fungsi yang lain, dan kita sebut variabel ini menjadi variabel lokal. Contoh:

```
<SCRIPT language="Javascript">  
var a = 12;  
var b = 4;  
function PerkalianDengan2(b) {  
var a = b * 2;  
return a;  
}  
document.write)("Dua kali dari “,b,” adalah “, Perkalian dengan2(b));  
document.write("Nilai dari a adalah",a);  
</SCRIPT>
```

Output:



### 3. Tipe Data

Tipe data adalah cara komputer mengorganisasi, menyimpan, dan memanipulasi informasi. Hampir setiap bahasa pemrograman memiliki tipe data yang berbeda-beda. Ini adalah beberapa tipe data umum :

#### a. Tipe Data Dasar

- Number (Angka) digunakan untuk merepresentasikan angka, baik bilangan bulat maupun pecahan. Contoh:

```
let bilBulat = 42;
let bilPecahan = 3.14;
```

- String (Teks) digunakan untuk merepresentasikan teks. Teks harus diapit oleh tanda kutip tunggal ( ' ') atau ganda ( " "). Contoh:

```
let teks1 = 'Halo, dunia!';
let teks2 = "Ini adalah contoh string.";
```

- Boolean (Boolean) yang hanya memiliki dua nilai: true atau false. Tipe data ini digunakan untuk ekspresi logika. Contoh:

```
let benar = true;
let salah = false;
```

#### b. Tipe Data Khusus

- Null digunakan untuk menyatakan bahwa sebuah variabel tidak memiliki nilai atau referensi. Contoh :

```
let tidakAdaNilai = null;
```

- Undefined digunakan untuk variabel yang telah dideklarasikan tetapi belum diberi nilai. Contoh :

```
let variabelBelumDiisi;
console.log(variabelBelumDiisi);
```

- Symbol (Simbol) digunakan untuk membuat nilai yang unik dan tidak dapat diubah. Ini berguna dalam konteks pengoptimalan dan penggunaan objek.

Contoh :

```
const simbolUnik = Symbol('deskripsi-simbol');
```

#### c. Tipe Data Referensi

- Object (Objek) digunakan untuk menyimpan banyak nilai dalam satu variabel dalam bentuk pasangan kunci-nilai. Contoh:

```
let mahasiswa = {
  nama: 'John';
  usia: 25,
```

```
jurusan: 'Teknik Informatika'  
};
```

- Array (Larik) digunakan untuk menyimpan sejumlah nilai dalam urutan. Contoh:

```
let angka = [1, 2, 3, 4, 5];
```

- Function (Fungsi): Fungsi adalah tipe data khusus yang digunakan untuk menjalankan blok kode tertentu. Contoh:

```
function tambah(a, b) {  
    return a + b;  
}
```

#### 4. Operator

Operator adalah simbol atau karakter khusus yang digunakan untuk melakukan operasi pada tipe data. Mereka memungkinkan kita untuk melakukan perhitungan matematika, perbandingan, manipulasi teks, dan banyak lagi.

##### a. Operator Aritmatika

Operator aritmatika digunakan untuk melakukan operasi aritmatika antara variabel dan atau nilai. Misal diberikan  $y = 5$ , tabel berikut menjelaskan tentang operator aritmatika.

Operator	Deskripsi	Contoh	Nilai x	Nilai y
+	Penjumlahan	$x = y + 2$	7	5
-	Pengurangan	$x = y - 2$	3	5
*	Perkalian	$x = y * 2$	10	5
/	Pembagian	$x = y / 2$	2.5	5
%	Modulus (Sisa bagi)	$x = y \% 2$	1	5
++	Increment	$x = ++y$	6	6
		$x = y++$	5	6
--	Decrement	$x = --y$	4	4
		$x = y--$	5	4

Contoh:

```
<SCRIPT LANGUAGE = "JavaScript">  
  
<!--  
    document.write("2 + 3 = " + (5 + 2) ); document.write("<BR>");  
    document.write("20 - 3 = " + (15 - 5) ); document.write("<BR>");  
    document.write("20 * 3 = " + (5 * 3) ); document.write("<BR>");
```

```
document.write("40 / 3 = " + (15 / 3) ); document.write("<BR>");  
//-->  
</SCRIPT>
```

Output:

```
2 + 3 = 7  
20 - 3 = 10  
20 * 3 = 15  
40 / 3 = 5
```

#### b. Operator Logika dan Pembandingan

Operator perbandingan biasanya berfungsi untuk menentukan kode mana dalam program yang harus dijalankan, operasi mana yang perlu dijalankan jika diberikan kondisi yang spesifik. Operator ini biasanya digunakan bersamaan dengan statement if..else, for loop, while loop, switch, dll.

Operator	Keterangan	Kategori
==	Kesamaan	Pembandingan
!=	Ketidaksamaan	Pembandingan
<	Kurang dari	Pembandingan
<=	Kurang dari atau sama dengan	Pembandingan
>	Lebih dari	Pembandingan
>=	Lebih dari atau sama dengan	Pembandingan
!	Bukan	Logika
&&	Dan	Logika
	Atau	Logika
?	<i>Kondisi ? Nilai Benar : Nilai Salah</i>	Pembandingan

Contoh:

```
<HTML>  
  <HEAD>  
    <TITLE>Operator ?</TITLE>  
  </HEAD>  
  <BODY>  
    <SCRIPT LANGUAGE = "JavaScript">  
      <!--  
      var nilai = prompt("Nilai (0-100): ", 0);
```

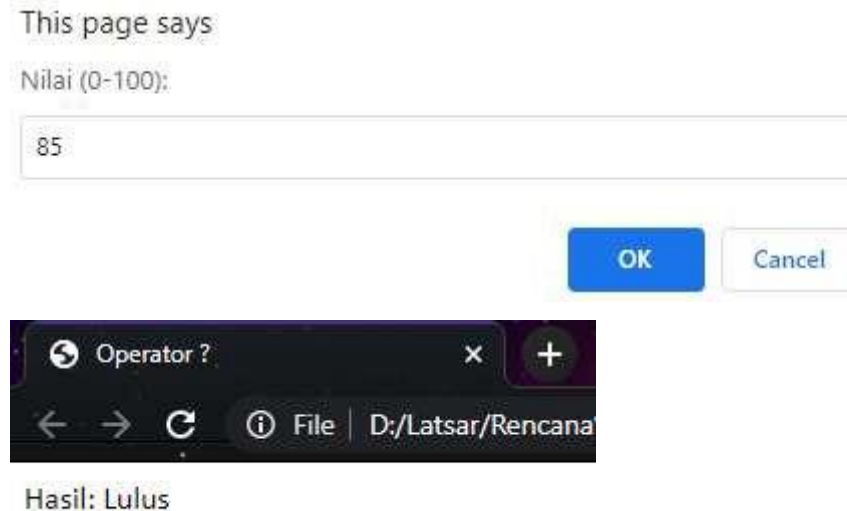


```

        var hasil = (nilai >= 60) ? "Lulus" : "Tidak Lulus"; document.write("Hasil:
        " + hasil);
        //-->
    </SCRIPT>
</BODY>
</HTML>

```

Output:



## 5. Fungsi Javascript (Function)

JavaScript adalah bahasa pemrograman yang bisa dimasukkan ke dalam halaman web dan digunakan untuk membuat situs lebih interaktif. Contohnya, JavaScript memungkinkan kita membuat form atau formulir di situs web yang bisa langsung diproses tanpa harus memuat ulang halaman. Bahasa ini tidak membutuhkan program khusus untuk menjalankannya, cukup dengan menggunakan browser.

Sebuah fungsi JavaScript didefinisikan dengan keyword function, diikuti dengan sebuah nama, diikuti oleh tanda kurung (). Nama fungsi dapat berisi huruf, angka, garis bawah, dan tanda dolar (aturan yang sama dengan variabel). Di dalam tanda kurung, dapat menyertakan nama parameter yang dipisahkan oleh koma: (parameter1, parameter2, ...) Kode yang akan dieksekusi oleh fungsi tersebut ditempatkan di dalam kurung kurawal: {}. Contoh:

```

function name(parameter1, parameter2, parameter3) {
    // code to be executed
}

```

Untuk membuat fungsi di JavaScript, terdapat tiga cara utama yang dapat digunakan cara biasa (function declaration), ekspresi fungsi (function expression) dan fungsi panah (arrow function).

a. Function Declaration

Function declaration adalah cara mendefinisikan function di JavaScript yang menggunakan kata kunci function diikuti oleh nama function dan parameter (jika ada) yang diletakkan dalam tanda kurung. Function declaration mendeklarasikan function di scope global, sehingga dapat diakses dari mana saja dalam program.

Contoh:

```
function sapa(nama) {  
    console.log('Halo, $(nama)!');  
}
```

b. Function Expression

Dalam metode ini, fungsi didefinisikan sebagai nilai yang disimpan dalam variabel. Fungsi ini sering kali merupakan fungsi anonim (tanpa nama). Untuk memanggilnya, gunakan nama variabel diikuti tanda kurung. Contoh:

```
var namaFungsi = function() {  
    console.log("Hello World!");  
};
```

c. Arrow Function

Fungsi panah diperkenalkan dalam ES6 dan menawarkan sintaks yang lebih ringkas. Fungsi ini juga sering kali lebih mudah dibaca, terutama untuk fungsi satu baris. Seperti sebelumnya, untuk memanggil fungsi ini, cukup gunakan namaFungsi();. Contoh:

```
var namaFungsi = () => {  
    console.log("Hello World!");  
};  
  
// Atau jika hanya memiliki satu pernyataan:  
var namaFungsi = () => console.log("Hello World!");
```

#### d. Return Statement

Function dalam JavaScript juga dapat mengembalikan nilai dengan menggunakan pernyataan return. Ini berguna ketika Kita ingin mengembalikan hasil perhitungan atau nilai tertentu dari function. Contoh:

```
function tambah(angka1, angka2) {  
    return angka1 - angka2;  
}  
let hasil = tambah(5, 3);  
console.log(hasil); //output 2
```

### 6. Looping

Looping pada bidang pemrograman adalah sebuah teknik untuk mengeksekusi kumpulan kode secara berulang-ulang berdasarkan suatu kondisi. Looping digunakan untuk mengulang suatu blok kode secara otomatis hingga kondisi yang diinginkan tercapai. Dalam pemrograman, terdapat beberapa jenis loop seperti perulangan for loop, for of loop, foreach loop, while loop, dan do-while loop.

- a. For Loop: loop ini digunakan untuk menjalankan blok kode beberapa kali. Biasanya digunakan saat Anda tahu berapa kali loop harus dijalankan.

```
for (let i = 0; i < 5; i++) {  
    console.log(i);  
}
```

Pada contoh di atas, loop akan mencetak angka 0 hingga 4.

- b. For of Loop: loop ini digunakan untuk mengiterasi nilai dari objek yang dapat diiterasi (iterable), seperti array, string, map, atau set.

```
const array = ['a', 'b', 'c'];  
for (let value of array) {  
    console.log(value);  
}
```

Loop ini akan mencetak setiap elemen dalam array.

- c. ForEach Loop: loop ini adalah metode yang tersedia pada array untuk menjalankan fungsi tertentu pada setiap elemen array.

```
const array = ['a', 'b', 'c'];
array.forEach((value) => {
  console.log(value);
});
```

Loop ini juga akan mencetak setiap elemen dalam array.

- d. While Loop: loop ini akan terus menjalankan blok kode selama kondisi yang diberikan bernilai true.

```
let i = 0;
while (i < 5) {
  console.log(i);
  i ++;
}
```

Pada contoh di atas, loop akan mencetak angka 0 hingga 4.

- e. Do...While Loop: loop ini mirip dengan while loop, tetapi akan menjalankan blok kode setidaknya sekali sebelum memeriksa kondisi.

```
let i = 0;
do {
  console.log(i);
  i++;
} while (i < 5);
```

Pada contoh di atas, loop akan mencetak angka 0 hingga 4.

## 7. Switch/Case statement

Switch statement merupakan salah satu jenis conditional statement pada bahasa pemrograman JavaScript yang digunakan untuk melakukan percabangan kode program. Switch statement digunakan ketika ingin mengevaluasi sebuah variabel atau ekspresi untuk beberapa nilai yang berbeda dan mengeksekusi kode program yang spesifik berdasarkan nilai tersebut. Nilai switch dibandingkan dengan nilai setiap case. Jika ada kecocokan, blok kode terkait akan dieksekusi. Jika tidak ada kecocokan, blok kode default akan dieksekusi.

```
switch (ekspresi) {
case nilai1:
```

```
// Blok kode untuk case nilai1
break;
case nilai2:
// Blok kode untuk case nilai2
break;
default:
// Blok kode untuk default (opsional)
}
```

Contoh:

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
let day;
switch (new Date().getDay()) {
  case 0:
    day = "Minggu";
    break;
  case 1:
    day = "Senin";
    break;
  case 2:
    day = "Selasa";
    break;
  case 3:
    day = "Rabu";
    break;
  case 4:
    day = "Kamis";
    break;
  case 5:
    day = "Jumat";
```

```
        break;
    case 6:
        day = "Sabtu";
    }
    document.getElementById("demo").innerHTML = "Hari ini hari " + day;
</script>
</body>
</html>
```

Output:

Hari ini hari Minggu

## 8. If/Else Statement

If else statement adalah salah satu bentuk pernyataan bersyarat yang digunakan untuk menentukan suatu keputusan pada kondisi tertentu. Pada saat kode sudah kompleks, maka penggunaan if else tentunya tidak terhitung lagi, penggunaan if else untuk menentukan keputusan sangatlah penting agar dapat mencapai suatu tujuan pada sebuah program. Contoh:

```
<!DOCTYPE html>
<html>
<body>

<p>Sapaan berdasarkan waktu:</p>

<p id="demo"></p>

<script>
const time = new Date().getHours();
let greeting;
if (time < 10) {
    greeting = "Selamat pagi";
} else if (time < 14) {
    greeting = "Selamat siang";
} else if (time < 18) {
```

```
        greeting = "Selamat sore";
    }
    else {
        greeting = "Selamat malam";
    }
    document.getElementById("demo").innerHTML = greeting;
</script>

</body>
</html>
```

Output:

Sapaan berdasarkan waktu:

Selamat siang

## 9. DOM Manipulation

Document Object Model (DOM) adalah antarmuka pemrograman yang digunakan JavaScript untuk memanipulasi konten, struktur, dan gaya halaman web. DOM merepresentasikan elemen HTML atau XML sebagai objek yang bisa diakses dan dimodifikasi. Dengan DOM, pengembang web bisa mengubah halaman web menjadi lebih interaktif dan dinamis.

Adapun fungsi DOM pada JavaScript, yaitu sebagai berikut.

- Mengubah teks, atribut, dan gaya dari elemen HTML.
- Menambahkan elemen baru atau menghapus elemen yang sudah ada.
- Mengelola event (misalnya klik, hover) pada elemen HTML, atau menciptakan event baru.

Beberapa cara untuk memilih elemen HTML menggunakan JavaScript:

- `getElementById()`: Mengambil elemen berdasarkan ID yang unik. Contoh: ``document.getElementById("id-elemen")``.
- `getElementsByName()`: Mengambil elemen-elemen yang memiliki class yang sama. Contoh: ``document.getElementsByName("class-name")``.
- `getElementsByTagName()`: Mengambil elemen-elemen berdasarkan tag HTML (misalnya ``p``, ``div``). Contoh: ``document.getElementsByTagName("tag-name")``.

- d. `querySelector()`: Mengambil elemen pertama yang sesuai dengan selector CSS yang diberikan. Contoh: ``document.querySelector(".class")``.
- e. `querySelectorAll()`: Mengambil semua elemen yang cocok dengan selector CSS yang diberikan. Contoh: ``document.querySelectorAll(".class")``.

Beberapa cara metode manipulasi elemen menggunakan JavaScript:

- a. `createElement()`: Membuat elemen HTML baru. Contoh: ``document.createElement("p")`` untuk membuat elemen paragraf.
- b. `appendChild()`: Menambahkan elemen baru yang telah dibuat ke dalam dokumen. Misalnya, menambahkan elemen baru ke dalam body atau elemen lain.
- c. `remove()`: Menghapus elemen dari halaman web. Misalnya, ``document.getElementById("id-elemen").remove()``.
- d. `removeChild()`: Menghapus elemen anak dari elemen induk. Contoh: ``parentElement.removeChild(childElement)``.

`CreateTextNode` adalah metode untuk membuat teks baru yang dapat ditambahkan ke elemen HTML, memungkinkan penambahan teks ke halaman web. Adapun `style` property yang digunakan untuk mengubah tampilan elemen HTML. Misalnya, Anda dapat mengubah warna latar, ukuran font, atau gaya lainnya dengan mengakses properti ``style`` dari elemen. Contoh: ``element.style.color = "red"``.



## **BAB V**

### **BOOTSTRAP**

#### **A. Tujuan**

1. Siswa dapat menjelaskan apa itu Bootstrap dan mengapa digunakan untuk membuat situs web yang responsif
2. Siswa dapat menginstal Bootstrap melalui beberapa cara
3. Siswa dapat memahami dan menggunakan komponen atau class Bootstrap
4. Siswa mampu membuat desain web yang responsif dan dapat menyesuaikan tampilan di berbagai perangkat

#### **B. Teori**

##### **1. Pengenalan Bootstrap**

Bootstrap adalah framework frontend open-source yang digunakan untuk merancang, mengembangkan, dan menyesuaikan situs web yang responsif serta mobile-first dengan cepat. Framework ini menawarkan sistem grid yang fleksibel, berbagai komponen interaktif dan aksesibel yang sudah siap digunakan, serta beragam utilitas dan alat bantu yang sangat berguna.

Bootstrap menyediakan gaya CSS bawaan untuk semua komponennya, mencakup elemen-elemen seperti tipografi, warna, ukuran, spasi, dan breakpoint, yang mendukung sistem grid. Bootstrap 5 bisa digunakan dengan pengaturan gaya default, namun pengguna juga bisa menyesuaikannya sesuai kebutuhan.

Sebagai kerangka kerja front-end gratis, Bootstrap mempermudah dan mempercepat pengembangan web dengan menyediakan template desain berbasis HTML dan CSS untuk berbagai elemen seperti tipografi, tombol, formulir, tabel, navigasi, modals, carousel, serta tambahan plugin JavaScript opsional. Dengan Bootstrap, pembuatan desain web responsif menjadi lebih sederhana, memungkinkan situs web menyesuaikan tampilannya dengan baik di berbagai perangkat, mulai dari ponsel hingga desktop.

Bootstrap sendiri memiliki 5 versi dan bootstrap 5 adalah versi terbaru dari Bootstrap yang dirilis pada tahun 2021. Versi ini menawarkan komponen baru, stylesheet yang lebih cepat, dan peningkatan responsivitas. Bootstrap 5 mendukung versi stabil terbaru dari semua browser dan platform utama. Namun, Internet Explorer 11 dan versi sebelumnya tidak didukung. Perbedaan utama antara Bootstrap 5 dan versi

sebelumnya (Bootstrap 3 & 4) adalah bahwa Bootstrap 5 telah beralih menggunakan JavaScript murni, bukan jQuery.

## 2. Instalasi Bootstrap

Ada beberapa cara untuk mulai menggunakan Bootstrap 5 di situs web yaitu sebagai berikut.

### a. Menggunakan CDN

CDN (Content Delivery Network) adalah jaringan server yang terdistribusi secara geografis di berbagai lokasi di seluruh dunia. Untuk instalasi bootstrap cukup copy link CDN Bootstrap yang bisa di dapat pada website resmi bootstrap, yaitu <https://getbootstrap.com/docs/5.2/getting-started/introduction/> lalu di taruh pada kode HTML. Kelebihan dari cara ini adalah kita bisa langsung menggunakan Bootstrap tanpa harus download. Tapi tetap butuh koneksi internet. Berikut langkah-langkah secara rinci.

- Buat file baru *index.html* di root proyek Anda. Sertakan `<meta name="viewport">` juga tag untuk perilaku responsif yang tepat di perangkat seluler.
- Sertakan CSS dan JS Bootstrap. Tempatkan `<link>` tag di dalam `<head>` untuk CSS kita, dan `<script>` tag untuk bundel JavaScript kita (termasuk Popper untuk memposisikan dropdown, popper, dan tooltip) sebelum penutup `</body>` contoh:

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css
"
rel="stylesheet"
integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2
Kadq2F9CUG65" crossorigin="anonymous">
  </head>
```

```

<body>
  <h1>Hello, world!</h1>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
                                integrity="sha384-
kenU1KFdBIe4zVF0s0G1M5b4hcxpD9F7jL+jjXkk+Q2h455rYXK/7HAuo
Jl+0I4" crossorigin="anonymous"></script>
  </body>
</html>

```

Adapun link CDN bootstrap 5 versi terbaru, yaitu sebagai berikut:

CSS	<a href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css</a>
JavaScript	<a href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js">https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js</a>

#### b. Download secara manual

Bootstrap dapat diunduh dari website resminya, tetapi tidak termasuk dokumentasi, file sumber, atau dependensi JavaScript opsional seperti Popper.

### Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v5.2.3** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see [CSS files comparison](#))
- Compiled and minified JavaScript plugins (see [JS files comparison](#))

This doesn't include documentation, source files, or any optional JavaScript dependencies like Popper.

[Download](#)

Setelah diunduh, maka akan mendapatkan file zip dengan isi folder CSS dan JavaScript. Setlah itu ekstrak folder CSS dan JavaScript ke dalam folder HTML masing-masing. Siapkan tag link disamping pada tag head halaman HTML, dan tag script di dalam tag body.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">

```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Bootstrap dari Lokal</title>

<link rel="stylesheet" href="css/bootstrap.min.css" />
</head>
<body>

<header class="py-5 bg-success">
  <div class="container">
    <h1 class="display-4 text-white">Hello Bootstrap Lokal!</h1>
  </div>
</header>

<script src="js/bootstrap.min.js"></script>

</body>
</html>
```

c. Download dengan package manager (npm)

Masukkan berkas sumber Bootstrap ke hampir semua proyek dengan beberapa pengelola paket yang paling populer. Apa pun pengelola pakatnya, Bootstrap akan memerlukan kompiler Sass dan Autoprefixer untuk pengaturan yang sesuai dengan versi resmi yang kami kompilasi. Berikut ini adalah langkah-langkah dalam mendownload bootstrap dengan package manager

- Install Node.js dari website resminya, yaitu: <https://nodejs.org/en/download/package-manager>, karena npm (node package manager) adalah bagian dari instalasi Node.js
- Buat direktori proyek baru atau navigasikan ke direktori proyek yang sudah ada di terminal
- Jika belum menginisialisasi proyek npm, jalankan perintah berikut untuk membuat file package.json: *npm init -y*

- Jalankan perintah berikut untuk mengunduh Bootstrap menggunakan npm:  
*npm install bootstrap@5.2.3*
- Setelah Bootstrap diunduh, maka dapat digunakan file CSS Bootstrap yang ada di dalam proyek.

```
<link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.min.css">
```

- Pastikan telah menyesuaikan path (alamat file) sesuai dengan struktur direktori proyek
- Jika perlu menggunakan komponen interaktif Bootstrap yang memerlukan JavaScript, maka dapat disertakan JavaScript Bootstrap.

```
<script src="node_modules/bootstrap/dist/js/bootstrap.min.js"></script>
```

- Class-class Bootstrap dalam elemen HTML sudah dapat digunakan

### 3. Class Bootstrap

Class dalam konteks Bootstrap mengacu pada atribut CSS class yang dapat ditambahkan ke elemen HTML untuk mengaplikasikan gaya dan perilaku tertentu sesuai dengan aturan yang telah ditentukan oleh Bootstrap framework. Bootstrap menggunakan sistem class untuk mengontrol tampilan dan perilaku elemen-elemen web secara konsisten. Untuk referensi terdapat di website: [https://www.w3schools.com/bootstrap/bootstrap\\_ref\\_all\\_classes.asp](https://www.w3schools.com/bootstrap/bootstrap_ref_all_classes.asp)

#### a. Container

Class ini digunakan untuk membungkus konten dengan lebar tetap yang responsif. Ini membuat konten Anda tetap dalam lebar tertentu pada berbagai perangkat dan ukuran layar. Contoh:

```
<body>
  <div class="container">
    <h1>Judul</h1>
    <p>Paragraf</p>
  </div>
</body>
```

Output:

## Judul

Paragraf

### b. Grid

Sistem grid Bootstrap dibangun dengan flexbox dan memungkinkan hingga 12 kolom di seluruh halaman. Jika tidak ingin menggunakan semua 12 kolom secara individual, maka dapat mengelompokkan kolom-kolom tersebut bersama-sama untuk membuat kolom yang lebih lebar. Sistem grid responsif, dan kolom akan tersusun ulang secara otomatis tergantung pada ukuran layar. Pastikan jumlahnya mencapai 12 atau kurang. Pada grid terdapat class row dan col-{breakpoint}-{number}. Class row digunakan untuk membuat baris dalam grid Bootstrap. Ini membantu dalam mengatur elemen-elemen dalam baris-baris horizontal untuk membuat layout responsif. Sedangkan col-{breakpoint}-{number} digunakan untuk membuat kolom dalam grid. {breakpoint} dapat diganti dengan xs, sm, md, atau lg untuk mengatur perilaku responsif pada berbagai ukuran layar. {number} menunjukkan jumlah kolom yang ingin digunakan dalam baris tersebut. Contoh:

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

Output:

Kolom 1

Kolom 1

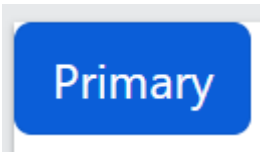
c. Button (btn)

Class btn digunakan untuk mengubah tampilan tombol. Class ini memberikan gaya dasar pada tombol dan membuatnya terlihat lebih menarik.

Contoh:

```
<body>
  <div class="container">
    <button class="btn btn-primary">Primary</button>
  </div>
</body>
```

Output:



d. Form-control

Class form-control digunakan untuk mengubah tampilan input dan textarea dalam formulir. Ini memberikan tampilan yang konsisten dan bersih untuk elemen-elemen form. Contoh:

```
<body>
  <div class="container">
    <form>
      <div class="form-group">
        <label for="nama">Nama: </label>
        <input type="text" class="form-control" id="nama">
      </div>
      <div class="form-group">
        <label for="email">Email:</label>
        <input type="email" class="form-control" id="email">
      </div>
    </form>
  </div>
</body>
```

```
        </div>
        <button type="submit" class="btn btn-primary">Submit</button>
    </form>
</div>
</body>
```

Output:

Nama:

Email:

Submit

#### e. Navbar

Navbar adalah komponen antarmuka yang digunakan untuk membuat navigasi di bagian atas sebuah halaman web. Navbar menyediakan struktur dan gaya yang responsif, sehingga bisa beradaptasi dengan berbagai ukuran layar, dari desktop hingga perangkat mobile. Navbar sangat fleksibel dan dapat disesuaikan dengan elemen-elemen lain seperti logo, link, tombol, dropdown, form pencarian, dan lain-lain.

Navbars dilengkapi dengan dukungan bawaan untuk beberapa sub-komponen. Pilih dari yang berikut ini sesuai kebutuhan:

- `.navbar-brand`` untuk nama perusahaan, produk, atau proyek Anda.
- `.navbar-nav`` untuk navigasi dengan tinggi penuh dan ringan (termasuk dukungan untuk dropdown).
- `.navbar-toggler`` untuk digunakan dengan plugin collapse kami dan perilaku toggling navigasi lainnya.
- Utilitas flex dan spasi untuk kontrol formulir dan aksi apa pun.
- `.navbar-text`` untuk menambahkan string teks yang diposisikan secara vertikal di tengah.
- `.collapse.navbar-collapse`` untuk mengelompokkan dan menyembunyikan isi navbar berdasarkan breakpoint induk.
- Tambahkan `.navbar-scroll`` opsional untuk menetapkan tinggi maksimal dan menggulir konten navbar yang diperluas.



Contoh:

```
<nav class="navbar navbar-expand-sm bg-light">

<div class="container-fluid">
  <!-- Links -->
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link" href="#">Link 1</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link 2</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link 3</a>
    </li>
  </ul>
</div>

</nav>
```

Output:

Link 1 Link 2 Link 3