

Daftar Isi

1	Pengenalan HTML	2
1.1	Mengenal Website	2
1.2	Cara Kerja Website	3
1.3	HTML (Hypertext Markup Language)	3
1.4	Struktur Dasar Halaman HTML	4
2	HTML tag	4
2.1	Memulai HTML	4
2.2	Format Teks	4
2.2.1	Heading	4
2.2.2	Paragraf	4
2.3	Menggunakan Font	4
2.4	Karakter Khusus	4
2.5	Garis Horizontal	4
2.6	Menggunakan List	4
2.7	Pewarnaan	4
2.8	Bekerja dengan Gambar	4
2.9	Link pada halaman Web	4
2.10	Tabel dan Pengaturannya	4
2.11	Form pada HTML	4
3	Pengenalan Cascading Style Sheet (CSS)	4
3.1	Pengertian CSS	4
3.2	Variasi atau Versi CSS	4
3.3	Struktur Kode atau Sintaks Dasar CSS	5
3.3.1	Selektor	5
3.3.2	Blok Deklarasi	9
3.3.3	Properti dan Nilai	10
4	Implementasi CSS	10
4.1	Penulisan Kode CSS	10
4.1.1	Internal CSS	10
4.1.2	Eksternal CSS	11
4.1.3	Inline CSS	15
5	Pengenalan Bootstrap	16
5.1	Pengertian Bootstrap	16
5.2	Sejarah dan Versi Bootstrap	16

5.2.1	Bootstrap 2 dan 3.....	17
5.2.2	Bootstrap 4.....	17
5.2.3	Bootstrap 5.....	18
6	Implementasi bootstrap pada halaman WEB.....	19
6.1	Memulai Menggunakan Bootstrap.....	19
6.2	Membuat Container.....	22
6.2.1	Menambahkan Background pada Container	24
6.3	Sistem Grid pada Bootstrap	25
7	Mengenal reactJS.....	26
7.1	Pengertian ReactJS.....	26
7.2	Konsep Dasar ReactJS.....	26
8	Sintaks, variable & konstanta.....	27
9	Operator, pengkondisian IF dan Switch pada PHP	27
10	Looping, array, function pada PHP	27
11	Form Handling pada PHP	27
12	Javascript	27
13	Implementasi Javascript.....	27

1 Pengenalan HTML

1.1 Mengenal Website

Website adalah sebuah kumpulan dari halaman web yang saling berhubungan dan dapat diakses melalui halaman utama (home page) menggunakan sebuah browser. Sebagai contoh, alamat www.polinema.ac.id, jika diakses maka akan tampil homepage, Sedangkan halaman web adalah bagian dari homepage yang berupa menu atau link di mana ketika di-klik maka akan membuka halaman baru dengan informasi yang berbeda dengan homepage.

Web Statis adalah website di mana penggunanya tidak bisa mengubah konten dari web tersebut secara langsung menggunakan browser. Interaksi yang terjadi antara pengguna dan server adalah seputar pemrosesan link saja. Halaman-halaman web tersebut tidak memiliki database. Data dan informasi yang ada pada web statis tidak berubah-ubah kecuali diubah sintaksnya. Dokumen web yang dikirim kepada client isinya akan sama dengan apa yang ada di web server.

Web Dinamis adalah website di mana interaksi yang terjadi antara pengguna dan server sangat kompleks. Seseorang bisa mengubah konten dari halaman tertentu dengan menggunakan

browser. Request dari pengguna dapat diproses oleh server yang kemudian ditampilkan dalam isi yang berbeda-beda menurut alur programnya. Halaman-halaman website tersebut terhubung dengan sebuah database, sehingga pada web dinamis akan memiliki informasi dan data yang berbeda-beda tergantung input yang disampaikan oleh client. Dokumen yang sampai di client akan berbeda dengan dokumen yang ada di server.

1.2 Cara Kerja Website

- 1) Pengguna menuliskan URL pada browser `http://www.google.com`
- 2) Browser akan menerjemahkan URL menjadi sebuah alamat IP dan menghubungkannya melalui TCP ke server pada port 80 (HTTP service)
 - menghubungkan ke sebuah komputer dengan IP address 216.239.39.99 port 80
- 3) Melalui koneksi yang sudah terhubung, browser mengirimkan sebuah GET request (HTTP request)
 - GET/HTTP/1.1
- 4) Server merespon dengan HTTP response (dengan header dan file yang diminta dalam body). MIME type juga diikutsertakan sehingga memberitahu browser mengenai tipe file (HTML, text, image, dll).
- 5) File di dalam halaman web seperti gambar tidak dikirimkan dengan file utama, sehingga browser harus melakukan request untuk mendapatkan semua file dalam halaman web.
 - GET /logo.gif HTTP/1.0
- 6) Browser akan melakukan render untuk ditampilkan kepada user dalam browser.
- 7) Koneksi dalam HTTP tidak berlangsung terus menerus, ketika sebuah file selesai diunduh, koneksi akan putus.

```
Content-type: text/html
<html>
<head><title> Google </title></head>
<body> Welcome to google <img src='logo.gif'></body>
</html>
```

1.3 HTML (Hypertext Markup Language)

HyperText Markup Language (HTML) adalah sebuah Bahasa yang digunakan untuk membuat sebuah halaman web, dimana pemformatan menggunakan hiperteks sederhana yang ditulis menggunakan format ASCII agar menghasilkan tampilan wujud yang terintegrasi. HTML ini merupakan standart internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan

Berners-Lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi Tinggi di jenewa). HTML dapat dibaca oleh berbagai macam platform. HTML juga merupakan Bahasa pemograman yang fleksible dan dapat digabungkan dengan Bahasa pemograman lain seperti PHP, ASP, JSP dan JavaScript. Beberapa tag dalam dokumen-dokumen HTML menentukan bagaimana teks diformat. Terdapat juga tag-tag yang lain yang memberitahukan bagaimana menanggapi aksi-aksi yang dilakukan oleh pengguna.

1.4 Struktur Dasar Halaman HTML

2 HTML tag

2.1 Memulai HTML

2.2 Format Teks

2.2.1 Heading

2.2.2 Paragraf

2.3 Menggunakan Font

2.4 Karakter Khusus

2.5 Garis Horizontal

2.6 Menggunakan List

2.7 Pewarnaan

2.8 Bekerja dengan Gambar

2.9 Link pada halaman Web

2.10 Tabel dan Pengaturannya

2.11 Form pada HTML

3 Pengenalan Cascading Style Sheet (CSS)

3.1 Pengertian CSS

CSS (Cascade Style Sheet) merupakan sebuah bahasa untuk mengatur tampilan web sehingga terlihat lebih menarik dan indah. Dengan CSS, kita dapat mengatur layout (tata letak), warna, font, garis, dan lain-lain. CSS pertama kali diperkenalkan oleh Håkon Wium Lie pada tahun 1994.

Seiring berkembangnya dunia pemrograman dan teknologi, CSS tidak hanya digunakan pada HTML dan XHTML saja, tapi sudah bisa digunakan untuk mendesain tampilan aplikasi android.

3.2 Variasi atau Versi CSS

Sejak awal diperkenalkan CSS memiliki beberapa variasi:

- 1) CSS 1: adalah versi pertama (17 Desember 1996)
- 2) CSS 2: adalah versi ke-2 (Mei 1998)
- 3) CSS 2.1: (7 juni 2011)
- 4) CSS 3: (2012)
- 5) CSS 4: masih dalam pengembangan.

Apa bedanya dari setiap versi tersebut?

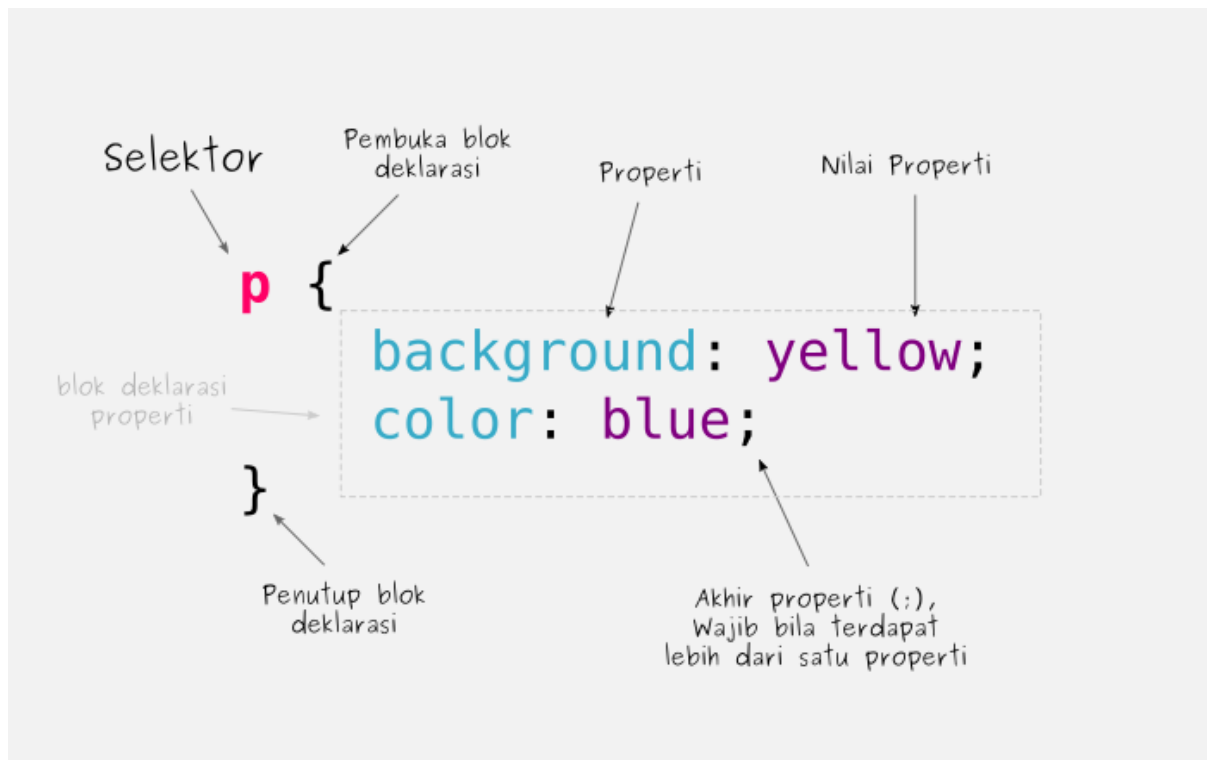
Tentunya ada penambahan dan pengurangan

3.3 Struktur Kode atau Sintaks Dasar CSS

Struktur kode CSS terdiri dari tiga bagian:

- 1) Selektor;
- 2) Blok Deklarasi;
- 3) Properti dan nilainya.

Contoh:



3.3.1 Selektor

Selektor adalah kata kunci untuk memilih elemen HTML yang akan kita atur.

Contohnya:

```
h1 {  
  color: red;  
}
```

Artinya: Kita memilih semua elemen `<h1>`, lalu diberikan warna teks red (merah).

Selektor dapat berupa nama tag, class, id, dan atribut.

3.3.1.1 Selector Tag

Selektor Tag disebut juga Type Selector. Selektor ini akan memilih elemen berdasarkan nama tag.

Contoh:

```
p {  
  color: blue;
```

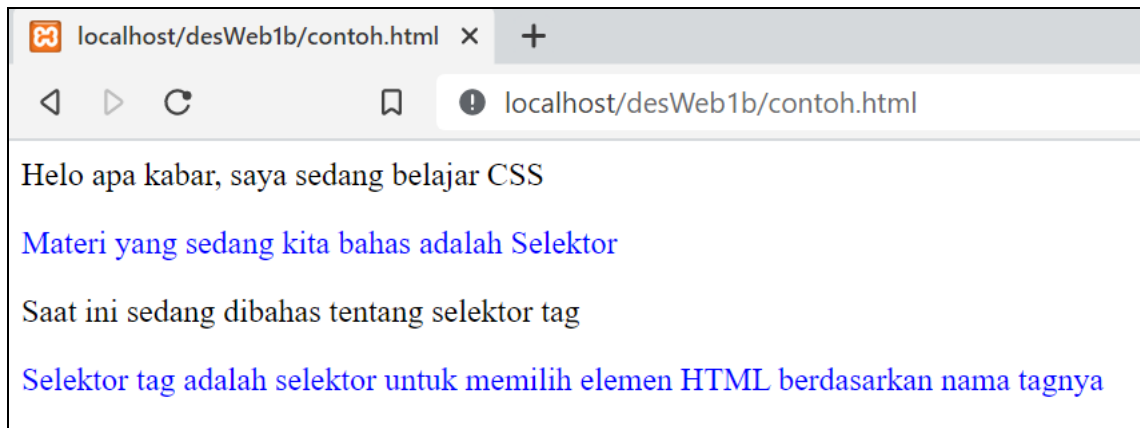
```
}
```

Artinya: Pilih semua elemen `<p>` lalu atur warna teksnya menjadi biru.

Kalau misalnya kita punya struktur HTML seperti ini:

```
<div>Helo apa kabar, saya sedang belajar CSS</div>
<p>Materi yang sedang kita bahas adalah Selektor</p>
<div>Saat ini sedang dibahas tentang selektor tag</div>
<p>Selektor tag adalah selektor untuk memilih elemen HTML berdasarkan
nama tagnya</p>
```

Maka akan menghasilkan output seperti ini:



Semua elemen `<p>` berwarna biru, sedangkan elemen `<div>` tidak.

3.3.1.2 Selector Class

Selektor class adalah selektor yang memilih elemen berdasarkan nama class yang diberikan. Selektor class dibuat dengan tanda titik di depannya.

Contoh:

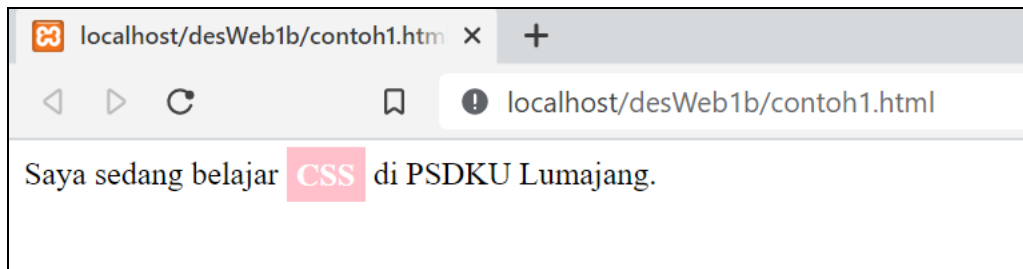
```
.pink {
  color: white;
  background: pink;
  padding: 5px;
}
```

Kita memiliki selektor class bernama `.pink`. Nah cara menggunakan selektor ini di HTML adalah dengan menambahkan atribut class di dalamnya.

Contoh:

```
<p>Saya sedang belajar <b class="pink">CSS</b> di PSDKU Lumajang.</p>
```

Maka hasilnya akan seperti berikut ini :



Selektor class dapat kita gunakan pada elemen yang kita inginkan. Sebuah elemen HTML dapat menggunakan satu atau lebih class.

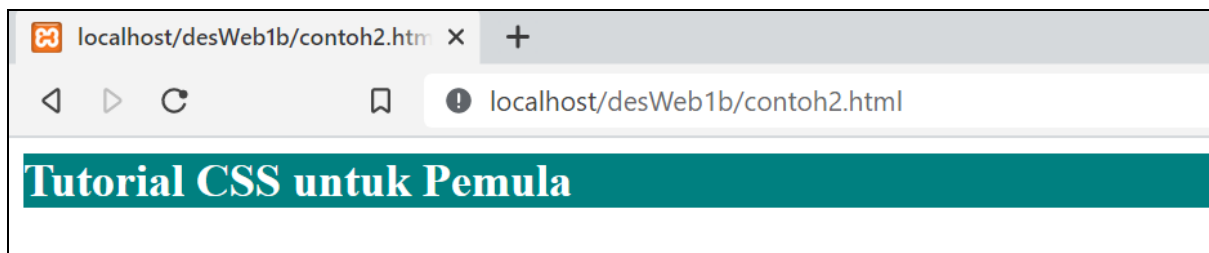
Contoh:

```
.text-white {  
    color: white;  
}  
  
.bg-teal {  
    background: teal;  
}
```

Lalu di HTML kita gunakan kedua class tersebut pada satu elemen:

```
<h2 class="text-white bg-teal">Tutorial CSS untuk Pemula</h2>
```

Hasilnya:



3.3.1.3 Selector ID

Selektor ID hampir sama dengan class. Bedanya, ID bersifat unik. Hanya boleh digunakan oleh satu elemen saja. Selektor ID ditandai dengan tanda pagar (#) di depannya.

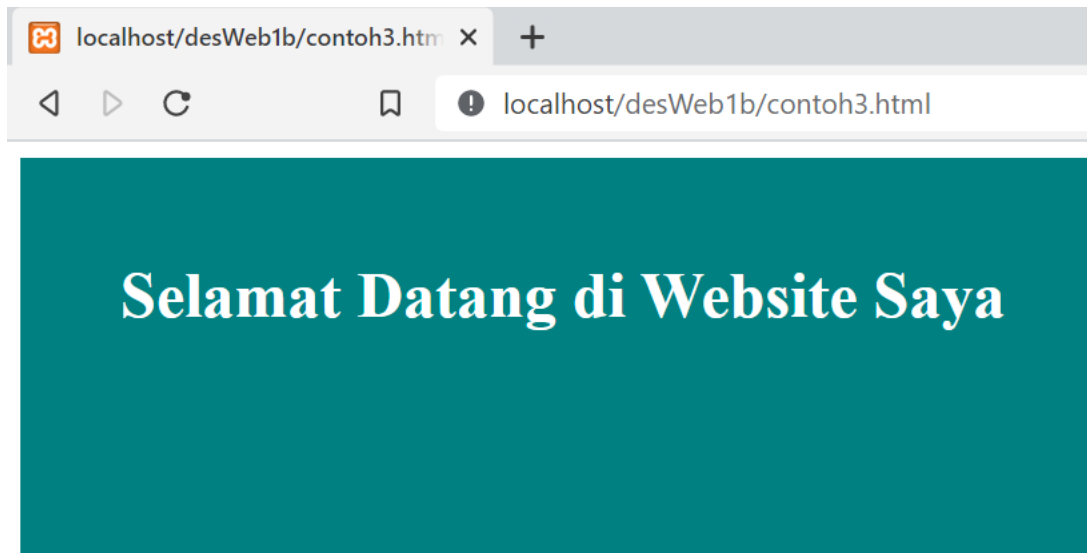
Contoh:

```
#header {  
    background: teal;  
    color: white;  
    height: 100px;  
    padding: 50px;  
}
```

Lalu pada kode HTML:

```
<h1 id="header">Selamat Datang di Website Saya</h1>  
<h1>Selamat Datang di Website Saya 2</h1>
```

Hasilnya :



Selamat Datang di Website Saya 2

3.3.1.4 Selector Attribute

Selektor atribut adalah selektor yang memilih elemen berdasarkan atribut. Selektor ini hampir sama seperti selektor Tag.

Contoh selektor Atribut:

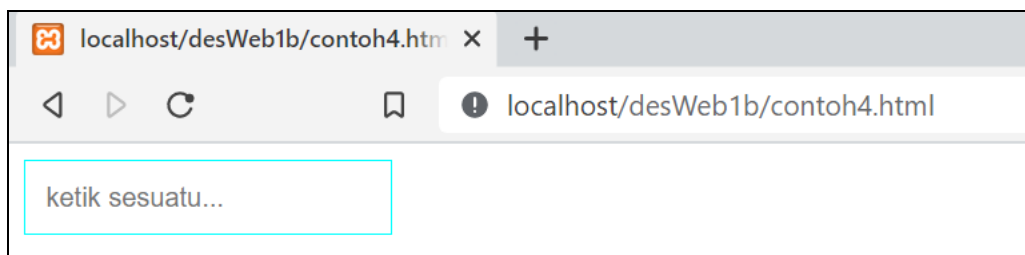
```
input[type=text] {  
  background: none;  
  color: cyan;  
  padding: 10px;  
  border: 1px solid cyan;  
}
```

Artinya kita akan memilih semua elemen `<input>` yang memiliki atribut `type='text'`.

Contoh kode HTML:

```
<input type="text" placeholder="ketik sesuatu..." />
```

Hasilnya :



3.3.1.5 Selector Universal

Selektor universal adalah selektor yang digunakan untuk menyeleksi semua elemen pada jangkauan (scope) tertentu.

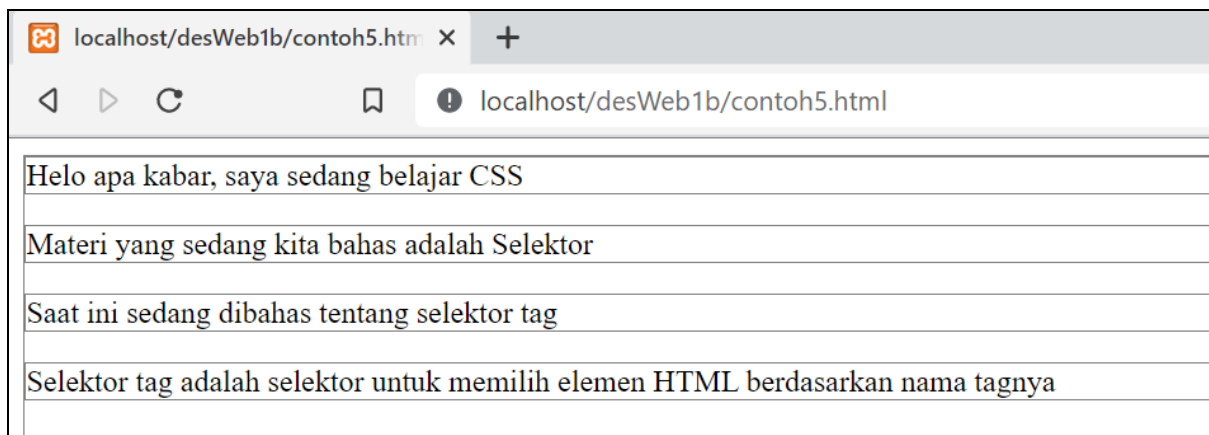
Contoh:

```
* {  
    border: 1px solid grey;  
}
```

Kemudian tambahkan kode HTML berikut ini :

```
<div>Helo apa kabar, saya sedang belajar CSS</div>  
<p>Materi yang sedang kita bahas adalah Selektor</p>  
<div>Saat ini sedang dibahas tentang selektor tag</div>  
<p>Selektor tag adalah selektor untuk memilih elemen HTML berdasarkan  
nama tagnya</p>
```

Hasilnya :



Selektor universal bisanya digunakan untuk me-reset CSS.

Kenapa harus di-reset?

Pada halaman HTML, ada beberapa CSS bawaan browser seperti padding dan margin pada elemen tertentu. Reset bertujuan untuk menghilangkan padding dan margin tersebut.

Contoh CSS reset:

```
* {  
    padding: 0;  
    margin: 0;  
}
```

Maka semua elemen tidak akan memiliki padding dan margin.

3.3.2 Blok Deklarasi

Blok deklarasi adalah tempat kita menuliskan atribut-atribut CSS yang akan diberikan ke pada selektor.

Contoh:

```
P {  
    font-size: 18px;  
}
```

Artinya, kita akan mengatur ukuran font dari tag <p> sebesar 18px. Blok deklarasi dimulai atau dibuka dengan tanda kurung { lalu ditutup dengan }.

3.3.3 Properti dan Nilai

Properti merupakan atribut atau sekumpulan aturan yang akan diberikan kepada elemen yang dipilih.

```
properti: "nilai";
```

Setiap properti harus diakhiri dengan titik koma (;). Apabila hanya terdapat satu properti, boleh tidak menggunakan titik koma. Properti harus ditulis di dalam blok deklarasi.

Contoh:

```
blockquote {  
    background: pink;  
}
```

Jangan tulis seperti ini:

```
background: pink;  
  
p {  
  
}
```

4 Implementasi CSS

4.1 Penulisan Kode CSS

Penulisan kode CSS dalam HTML dibagi menjadi tiga cara, internal, inline, dan eksternal. Pembagian ini berdasarkan letak kode CSS tersebut ditulis.

4.1.1 Internal CSS

Internal CSS adalah kode CSS yang ditulis di dalam tag <style>. Internal CSS juga dikenal dengan sebutan Embedded CSS.

Tag <style> biasanya ditulis di dalam tag <head>. Bisa juga ditulis di dalam <body>, namun lebih banyak ditulis di dalam <head>.

Contoh:

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>Contoh Internal CSS</title>  
    <!-- penulisan internal css dalam tag head -->  
    <style type="text/css">  
        p{  
            font-family: serif;  
            line-height: 1.75em;  
            font-size: 18px;  
        }  
        i {  
            font-family: sans;  
            color: orange;  
        }  
    </style>
```

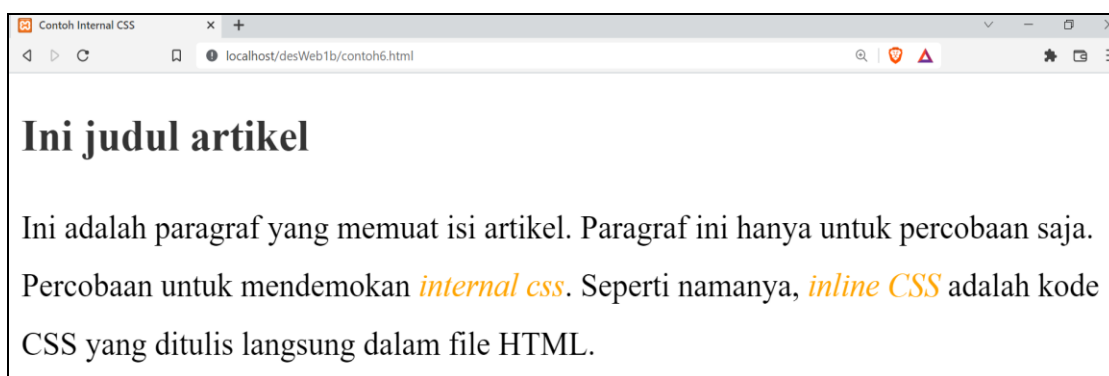
```

</head>

<body>
  <!-- penulisan internal css dalam tag body -->
  <style type="text/css">
    h2 {
      font-family: sans;
      color: #333;
    }
  </style>
  <h2>Ini judul artikel</h2>
  <p>Ini adalah paragraf yang memuat isi artikel. Paragraf ini hanya
  untuk percobaan saja. Percobaan untuk mendemokan <i>internal css</i>.
  Seperti namanya, <i>inline CSS</i> adalah kode CSS yang ditulis langsung
  dalam file HTML.</p>
</body>
</html>

```

Hasilnya :



4.1.2 Eksternal CSS

Eksternal CSS adalah kode CSS yang ditulis terpisah dengan kode HTML. Eksternal CSS ditulis disebuah file khusus yang berekstensi .css.

Sebagai contoh, saya akan membuat sebuah file bernama style-ku.css. Berikut ini cuplikan isi file style-ku.css

```

p {
  font-family: serif;
  line-height: 1.75em;
}

i {
  font-family: sans;
  color: blue;
}

h2 {
  font-family: sans;
  color: red;
}

```

Untuk menggunakan CSS tersebut dalam HTML, kita perlu mengimpornya. Ada beberapa cara memasukkan kode CSS dari berkas eksternal:

Pertama menggunakan tag <link>

```
<link rel="stylesheet" type="text/css" href="style-ku.css">
```

Atau bisa juga menggunakan @import

```
<style type="text/css">
@import "style-ku.css";
</style>
```

Penulisan pada HTML versi lengkapnya seperti ini:

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh Eksternal CSS</title>
  <link rel="stylesheet" type="text/css" href="style-ku.css">
</head>

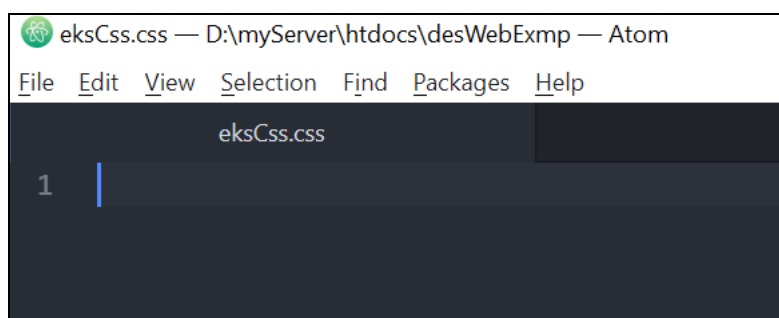
<body>
  <h2>Ini judul artikel</h2>
  <p>Ini adalah paragraf yang memuat isi artikel. Paragraf ini hanya
untuk percobaan saja. Percobaan untuk mendemokan <i>internal css</i>.
Seperti namanya, <i>inline CSS</i> adalah kode CSS yang ditulis langsung
dalam file HTML.</p>
</body>
</html>
```

Hasilnya :



4.1.2.1 Praktikum Eksternal CSS

- 1) Langkah pertama buatlah dokumen css baru dan simpan di dalam folder. Berikan nama eksCss.css pada file css anda.



- 2) Tambahkan selector tag css untuk menentukan warna dan background pada tag <h1> dengan properties seperti berikut :

Properties	Nilai
color	#7f2aec
background	#bff9f7

Berikut adalah implementasi dalam css nya :

```
/* membuat selector tag */  
h1 {  
    color: #7f2aec;  
    background: #bff9f7;  
}
```

- 3) Tambahkan selector class untuk menentukan warna dan background text dengan properties seperti berikut ini :

Selector	Properties	Nilai
.warnaText	color	#ae24b7
.bgText	background	#eed3f0

Berikut adalah implementasi dalam css nya :

```
/* membuat selector class */  
.warnaText {  
    color: #ae24b7;  
}  
  
.bgText {  
    background: #eed3f0;  
}
```

- 4) Tambahkan selector attribute untuk mengganti tampilan dari attribute text pada type input dengan properties sebagai berikut :

Selector	Properties	Nilai
input[type=text]	width	130px
	box-sizing	border-box
	border	2px solid #ccc
	border-radius	4px
	font-size	16px
	background-color	white
	background-image	url('img/search.png')

	background-size	25px 25px
	background-position	10px 10px
	background-repeat	no-repeat
	padding	12px 20px 12px 40px
	transition	width 0.4s ease-in-out
input[type=text]:focus	width	100%

Berikut adalah implementasi dalam css nya :

```
/* membuat selector attribute */
input[type=text] {
  width: 130px;
  box-sizing: border-box;
  border: 2px solid #ccc;
  border-radius: 4px;
  font-size: 16px;
  background-color: white;
  background-image: url('img/search.png');
  background-size: 25px 25px;
  background-position: 10px 10px;
  background-repeat: no-repeat;
  padding: 12px 20px 12px 40px;
  transition: width 0.4s ease-in-out;
}

input[type=text]:focus {
  width: 100%;
}
```

- 5) Buatlah halaman HTML baru dan beri nama eksCss.html
- 6) Lengkapi tag htmlnya seperti berikut ini :

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Membuat Eks CSS pada HTML</title>
  </head>
  <body>
  </body>
</html>
```

- 7) Tambahkan kode untuk memanggil css eksternal yang sudah kita buat :

```
<!-- load css -->
<link rel="stylesheet" href="eksCss.css" type="text/css">
```

- 8) Di dalam tag body tambahkan kode berikut ini :

```
<h1>Halaman Pencarian</h1>
<b class="warnaText bgText">*Note: </b><u class="warnaText">untuk
mencari data klik pada input pencarian</u></br></br>
```

```
<input type="text">
```

9) Jalankan halaman html



10) Amati pada input type yang anda buat, berikan Analisa terkait sebelum dan setelah di klik tombol pencariannya.

4.1.3 Inline CSS

Inline CSS adalah kode CSS yang ditulis langsung pada atribut elemen HTML. Setiap elemen HTML memiliki atribut style, di sana lah inline CSS ditulis. Contohnya seperti ini:

```
<h2 style="color:red; font-family: sans;">Ini judul artikel</h2>
```

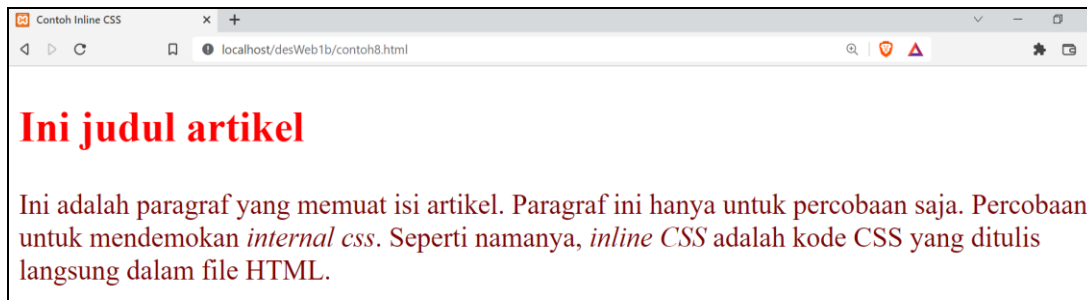
Contoh lengkap:

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh Inline CSS</title>
</head>

<body>

  <h2 style="color:red;font-family:sans">Ini judul artikel</h2>
  <p style="color:maroon">Ini adalah paragraf yang memuat isi artikel.
Paragraf ini hanya untuk percobaan saja. Percobaan untuk mendemokan
<i>internal css</i>. Seperti namanya, <i>inline CSS</i> adalah kode CSS
yang ditulis langsung dalam file HTML.</p>
</body>
</html>
```

Hasilnya :



5 Pengenalan Bootstrap

5.1 Pengertian Bootstrap

Bootstrap adalah framework HTML, CSS, dan JavaScript yang berfungsi untuk mendesain website responsive dengan cepat dan mudah. Framework open source ini diciptakan pada tahun 2011 oleh Mark Otto dan Jacob Thornton dari Twitter. Itulah kenapa dulunya Bootstrap dinamakan Twitter Blueprint. Bootstrap dengan cepat meraih popularitas digunakan oleh 27% website di seluruh dunia. Hal itu karena kesederhanaan dan konsistensi yang ditawarkan Bootstrap dibanding framework lainnya saat itu.

Kemudahan yang ditawarkan oleh Bootstrap adalah Anda tak perlu coding komponen website dari nol. Framework ini tersusun dari kumpulan file CSS dan JavaScript berbentuk class yang tinggal pakai. Class yang disediakan Bootstrap juga cukup lengkap. Mulai dari class untuk layout halaman, class menu navigasi, class animasi, dan masih banyak lainnya.

Menariknya lagi, Bootstrap bersifat responsive berkat grid system yang digunakan. Sistem grid pada bootstrap menggunakan rangkaian containers, baris, dan kolom untuk menyesuaikan bentuk layout dan konten website Anda. Dengan kata lain, Bootstrap menjamin tampilan website Anda akan tetap rapi dan konsisten di berbagai perangkat pengunjung. Baik melalui smartphone, tablet, atau laptop.

5.2 Sejarah dan Versi Bootstrap

Bootstrap awalnya bernama Twitter Blueprint. Karena waktu itu pertama kali dikembangkan oleh Mark Otto dan Jacob Thornton di Twitter.

Tujuan awalnya untuk menjaga konsistensi pada tool internal di Twitter. Karena sebelum adanya Bootstrap, tim Twitter sudah menggunakan beberapa library untuk membuat tampilan (GUI) Twitter. Akan tetapi tidak konsisten sehingga kode jadi sulit dirawat (maintenance).

Berangkat dari masalah tersebut: Mark Otto dan tim kecil yang berisi programmer mulai mendesain dan membuat sebuah tool internal yang dinamakan Twitter Blueprint. Setelah beberapa bulan pengembangan, banyak programmer lainnya ikut berkontribusi pada proyek ini sebagai partisipasi dalam Hack Week.

O ya, Hack Week adalah sebuah Hackathon yang ada di tim Development Twitter. Kemudian Twitter Blueprint diubah namanya menjadi Bootstrap dan dirilis sebagai proyek Open Source pada 19 Agustus 2011.

5.2.1 Bootstrap 2 dan 3

- 1) Pada tanggal 31 Januari 2012, Bootstrap 2 dirilis dengan penambahan dukungan pada Glyphicons, penambahan beberapa komponen, dan juga perubahan pada komponen yang sudah ada.
- 2) Pada Versi ini sudah mendukung responsive web design, yakni sebuah layout yang dapat menyesuaikan diri pada ukuran perangkat yang digunakan.
- 3) Selanjutnya Bootstrap 3, dirilis pada tanggal 19 Agustus 2013. Pada versi ini komponen Bootstrap didesain ulang dengan konsep Flat design dan menggunakan pendekatan Mobile first.

Buttons in **Bootstrap 2**

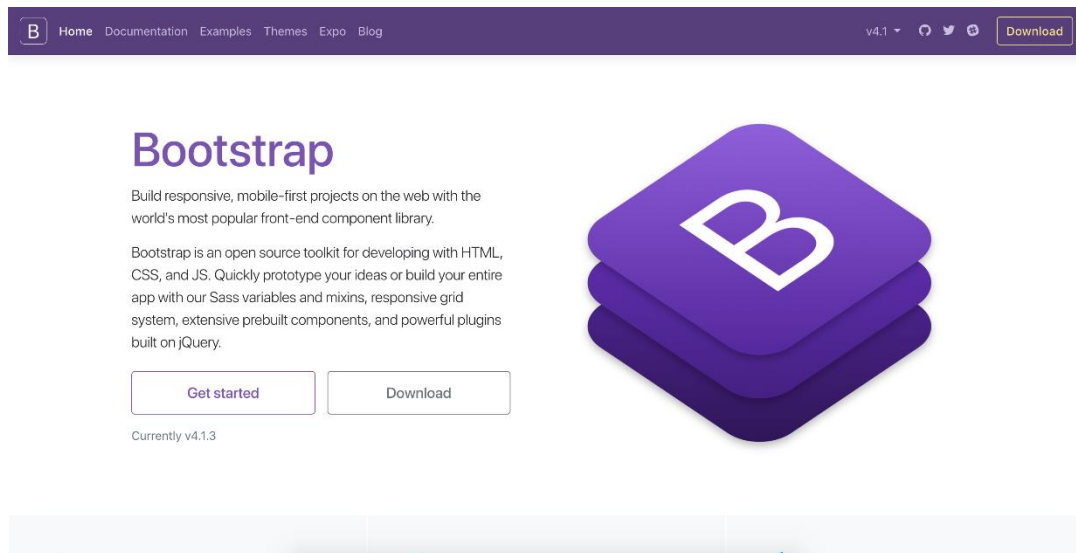


Buttons in **Bootstrap 3**



5.2.2 Bootstrap 4

Pada tanggal 29 Oktober 2014, Bootstrap 4 rilis dan versi finalnya rilis pada tanggal 18 Januari 2018. Versi final artinya versi yang tidak dilanjutkan pengembangannya.




Beberapa perubahan pada rilis ini:

- Penulisan ulang kode mayor;
- Mengganti Less dengan Sass;
- Menambahkan Normalize CSS;
- Menghapus dukungan IE8, IE9, dan iOS6;
- Mendukung CSS Flex Box

5.2.3 Bootstrap 5

Bootstrap 5 resmi dirilis versi Alpha pada tanggal 16 Juni 2020. Berikutnya versi Beta dirilis pada 7 Desember 2020.

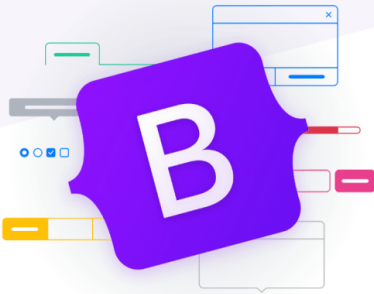
 [Home](#) [Docs](#) [Examples](#) [Icons](#) [Themes](#) [Expo](#) [Blog](#) [Download](#)


Build fast, responsive sites with Bootstrap

Quickly design and customize responsive mobile-first sites with Bootstrap, the world's most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.

[Get started](#) [Download](#)

Currently v5.0.0-alpha1 · [Previous releases](#)





Installation

Install Bootstrap's source Sass and JavaScript files via npm, Composer, or Meteor.

Package managed installs don't include documentation or our full build scripts. You can also [use our npm template repo](#) to quickly generate a Bootstrap project via npm.

[Read installation docs](#)

```
$ npm install bootstrap
```

[Copy](#)

```
$ gem install bootstrap -v 5.0.0-alpha1
```

[Copy](#)

Beberapa perubahan pada Bootstrap 5:

- Menghapus jQuery;
- Penulisan ulang pada sistem grid;
- Migrasi dari Jekyll ke Hugo untuk web dokumentasi Bootstrap;
- Menghapus dukungan untuk IE10 dan IE11;
- Menambahkan sekumpulan SVG icons;
- dan masih banyak lagi

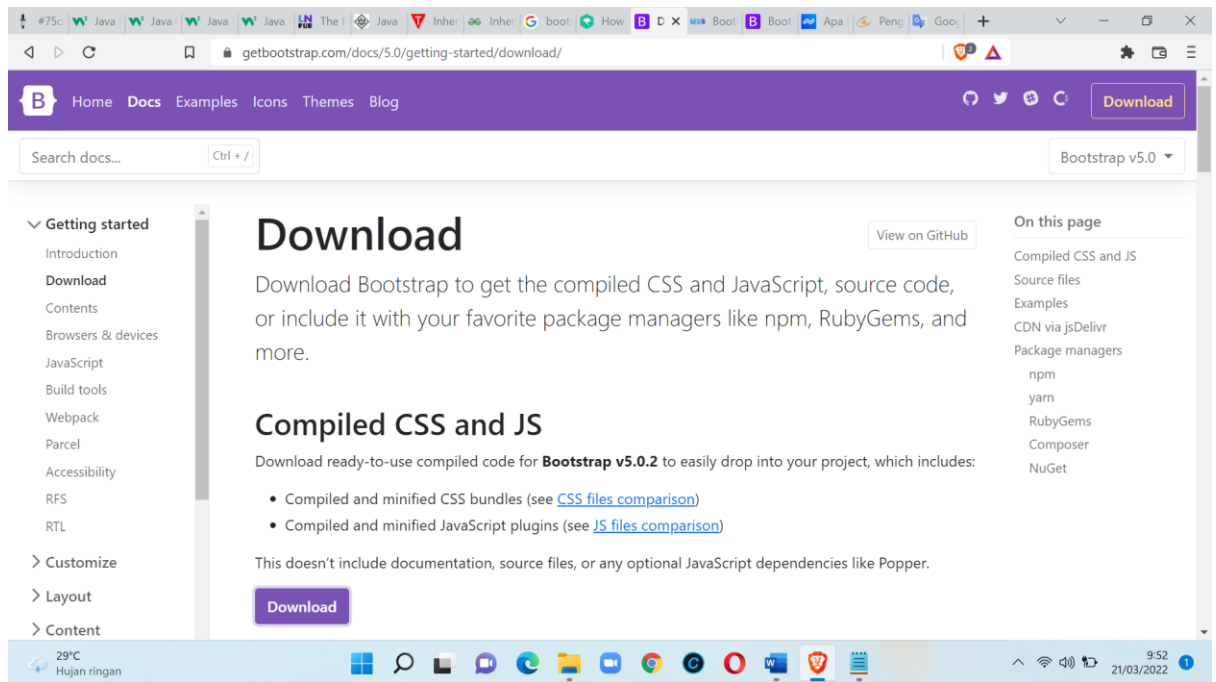
6 Implementasi bootstrap pada halaman WEB

6.1 Memulai Menggunakan Bootstrap

Untuk mengimplementasikan bootstrap dalam halaman HTML, maka kita harus mendownload file bootstrap dan kita panggil dalam halaman HTML yang ingin kita buat.

Berikut adalah Langkah – Langkah untuk memanggil dan mengimplementasikan bootstrap pada halaman HTML :

- 1) Buka halaman <https://getbootstrap.com/docs/5.0/getting-started/download/> setelah itu akan tampil halaman seperti dibawah ini.



- 2)
- 3) Setelah itu download Compiled Css dan JS pada halaman web tersebut dan simpan filenya.
- 4) Extract Bootstrap5 dan letakkan file Css dan JS kedalam folder yang sama pada halaman web kalian masing – masing.

Name	Date modified	Type
css	23/06/2021 1:25	File folder
js	23/06/2021 1:25	File folder

- 5) Selanjutnya buatlah halaman web sederhana dengan tag sebagai berikut dan simpan dengan nama **impBootstrap.html** :

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<title>Basic HTML File</title>

</head>

<body>

  <h1>Hello, world!</h1>

</body>

</html>
```

- 6) Kemudian simpan impBootstrap.html dan jalankan pada browser.
- 7) Berikut adalah hasil dari tampilan dari halaman web tersebut.



- 8) Langkah berikutnya adalah panggil bootstrap dengan cara menambahkan tag <link >di dalam halaman htmlnya.

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <title>Basic HTML File</title>

  <!-- Kode untuk memanggil Bootstrap CSS -->
```

```

<link rel="stylesheet" href="../desWeb1b/css/bootstrap.css">

</head>

<body>

  <h1>Hello, world!</h1>

</body>

</html>

```

9) Simpan dan jalankan pada tampilan web, kemudian amati perubahannya.

6.2 Membuat Container

Container adalah element dari tampilan dalam Bootstrap yang paling basic dan untuk implementasinya Container membutuhkan sistem Grid. Ada tiga tipe Container di dalam Bootstrap, berikut adalah ketiganya :

- 1) .container
- 2) .container-fluid
- 3) .container-{breakpoint}, dibagi menjadi beberapa bagian.

Berikut adalah tabel container :

Classes Bootstrap Grid System	X-Small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
.container	100%	540px	720px	960px	1140px	1320px
.container-sm	100%	540px	720px	960px	1140px	1320px
.container-md	100%	100%	720px	960px	1140px	1320px
.container-lg	100%	100%	100%	960px	1140px	1320px
.container-xl	100%	100%	100%	100%	1140px	1320px
.container-xxl	100%	100%	100%	100%	100%	1320px
.container-fluid	100%	100%	100%	100%	100%	100%

Contoh implementasi container :

- a) .container

```

<!-- contoh .container -->
<div class="container">
  <h1>This is a heading</h1>

```

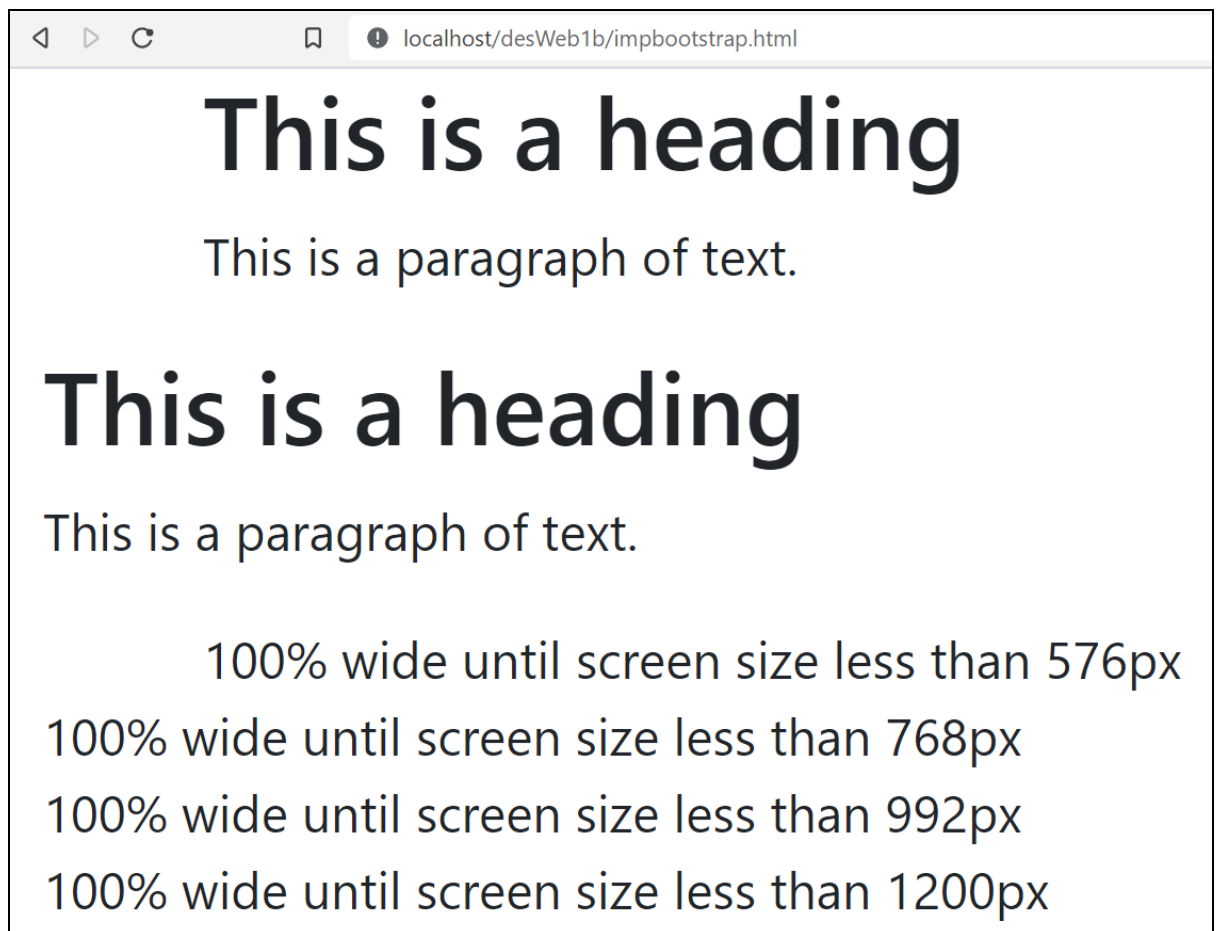
```
<p>This is a paragraph of text.</p>
</div>
```

b) .container-fluid

```
<!-- contoh .container-fluid -->
<div class="container-fluid">
  <h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</div>
```

c) .container{breakpoint}

```
<!-- contoh .container breakpoint -->
<div class="container-sm">100% wide until screen size less than
576px</div>
<div class="container-md">100% wide until screen size less than
768px</div>
<div class="container-lg">100% wide until screen size less than
992px</div>
<div class="container-xl">100% wide until screen size less than
1200px</div>
```



6.2.1 Menambahkan Background pada Container

Pada dasarnya container tidak terdapat background warna ataupun garis tepi (border), akan tetapi jika anda membutuhkan background ataupun garis tepi, kita bisa menambahkannya sesuai dengan style yang kita mau atau kita bisa memanfaatkan background warna ataupun garis tepi yang sudah disediakan oleh Bootstrap itu sendiri. Berikut adalah contoh membuat background warna dan garis tepi pada halaman HTML dengan memanfaatkan Bootstrap :

A) Menambahkan background dark dan warna text putih

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Menambahkan Background warna dan Garis Tepi</title>
    <!-- memanggil bootstrap -->
    <link rel="stylesheet"
href="../../desWebExmp/css/bootstrap.css">
  </head>
  <body>
    <!-- Container dengan backgroud dark dan warna text putih -->
    <div class="container bg-dark text-white">
      <h1>This is a heading</h1>
      <p>This is a paragraph of text.</p>
    </div>
  </body>
</html>
```

Hasilnya :



B) Menambahkan background warna light

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Menambahkan Background warna dan Garis Tepi</title>
    <!-- memanggil bootstrap -->
    <link rel="stylesheet"
href="../../desWebExmp/css/bootstrap.css">
  </head>
  <body>
    <!-- Container dengan background light -->
    <div class="container bg-light">
      <h1>This is a heading</h1>
      <p>This is a paragraph of text.</p>
    </div>
  </body>
```



```
</html>
```

Hasilnya :



C) Menambahkan border pada container

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Menambahkan Background warna dan Garis Tepi</title>
    <!-- memanggil bootstrap -->
    <link rel="stylesheet"
href="../../desWebExmp/css/bootstrap.css">
  </head>
  <body>
    <!-- Container dengan border -->
    <div class="container border">
      <h1>This is a heading</h1>
      <p>This is a paragraph of text.</p>
    </div>
  </body>
</html>
```

Hasilnya :



6.3 Sistem Grid pada Bootstrap

Sistem Grid pada Bootstrap digunakan untuk membuat website yang responsive dengan cara yang cepat.

Sistem grid bootstrap menyediakan cara yang mudah dan kuat untuk membuat tata letak responsif dari semua bentuk dan ukuran. Sistem Grid tersebut dibangun dengan flexbox

dengan pendekatan mobile-first. Dan juga, sistem Grid ini sepenuhnya responsif dan menggunakan sistem dua belas kolom (12 kolom tersedia per baris) dan enam tingkat responsif yang sudah disediakan secara default.

7 Mengenal reactJS

7.1 Pengertian ReactJS

Reactjs adalah sebuah library untuk membuat UI (user interface) Web (dan juga mobile). Banyak orang menyebutnya sebagai framework, namun secara teknis Reactjs bukanlah sebuah framework, Melainkan sebuah library untuk membuat UI.

Reactjs awalnya dibuat oleh Jordan Walke yang merupakan karyawan Facebook. Ia merilis prototipe pertama Reactjs dengan nama “FaxJS” yang terinspirasi dari XHP. Facebook menyadari, aplikasi (web) mereka semakin kompleks. Bayangkan saja, ada banyak bagian yang harus terus di-update seperti news feed, chat list, chat box, dll. secara realtime. Jika menggunakan cara tradisional seperti JQuery, ini akan memakan banyak biaya. Biaya yang saya maksud di sini adalah biaya resource komputasinya. Bukan biaya dalam bentuk uang atau materi. Oleh karena itulah React hadir memberikan solusi, yakni dengan menggunakan VirtualDOM yang menurut mereka lebih cepat.

Reactjs kemudian mulai digunakan di Facebook untuk membuat halaman News Feed di tahun 2011 dan kemudian di instagram pada tahun 2012. Setahun berikutnya, di tahun 2013 Reactjs dibuka untuk umum alias menjadi proyek open-source.

7.2 Konsep Dasar ReactJS

Pada dasarnya Reactjs hanya melakukan render komponen saat ada data yang berubah. Seperti namanya “React” ia akan bereaksi saat ada perubahan data (reaktif).

Lalu komponen itu apa?

Komponen adalah bagian-bagian dari UI, contohnya seperti tombol, label, input, dll. Komponen juga bisa dibentuk dari komponen yang lain.

Langkah-langkah yang harus dilakukan untuk membuat aplikasi react adalah sebagai berikut:

- 1) Menambahkan library react ke HTML;
- 2) Membuat elemen HTML untuk wadah aplikasi;
- 3) Membuat komponen;

4) Me-render komponen ke HTML;

Setiap aplikasi react membutuhkan sebuah wadah. Kita membuat sebuah elemen div dengan id="app" sebagai wadah untuk reactJS

8 Sintaks, variable & konstanta

9 Operator, pengkondisian IF dan Switch pada PHP

10 Looping, array, function pada PHP

11 Form Handling pada PHP

12 Javascript

13 Implementasi Javascript