

Modul Desain dan Pemrograman Web

Jurusan Teknologi Informasi
Program Studi Teknik Informatika

Penulis



Politeknik Negeri Malang

Daftar Isi

Daftar Isi	2
Daftar Gambar	5
1 Pengenalan HTML.....	8
1.1 Mengetahui Website	8
1.2 Cara Kerja Website	9
1.3 HTML (Hypertext Markup Language)	10
1.4 Struktur Dasar Halaman HTML	11
2 HTML tag.....	12
2.1 Memulai HTML	12
2.2 Format Teks	13
2.2.1 Heading	14
2.2.2 Paragraf.....	15
2.3 Menggunakan Font	17
2.4 Karakter Khusus	18
2.5 Garis Horizontal	21
2.6 Menggunakan List.....	22
2.7 Pewarnaan	23
2.8 Bekerja dengan Gambar	25
2.9 Link pada halaman Web	30
2.10 Tabel dan Pengaturannya	34
2.11 Form pada HTML.....	36
3 Pengenalan Cascading Style Sheet (CSS)	39
3.1 Pengertian CSS.....	39

3.2	Variasi atau Versi CSS	39
3.3	Struktur Kode atau Sintaks Dasar CSS.....	39
3.3.1	Selektor	40
3.3.2	Blok Deklarasi.....	46
3.3.3	Properti dan Nilai.....	46
4	Implementasi CSS	47
4.1	Penulisan Kode CSS	47
4.1.1	Internal CSS.....	47
4.1.2	Eksternal CSS	48
4.1.3	Inline CSS	54
5	Pengenalan Bootstrap.....	55
5.1	Pengertian Bootstrap.....	55
5.2	Sejarah dan Versi Bootstrap	56
5.2.1	Bootstrap 2 dan 3.....	57
5.2.2	Bootstrap 4.....	58
5.2.3	Bootstrap 5.....	59
6	Implementasi bootstrap pada halaman WEB	60
6.1	Memulai Menggunakan Bootstrap	60
6.2	Membuat Container	64
6.2.1	Menambahkan Background pada Container	66
6.3	Sistem Grid pada Bootstrap	69
6.3.1	Membuat Tata Letak Dua Kolom	71
6.3.2	Membuat Layout Tiga Kolom	72
6.3.3	Membuat Tata Letak Kolom secara Otomatis...	74
6.3.4	Praktikum System Grid.....	75

6.4	Membuat Fixed Layout Pada Bootstrap	81
7	Mengenal reactJS.....	86
7.1	Pengertian ReactJS.....	86
7.2	Konsep Dasar ReactJS	87
8	Menggunakan JSX	89
9	Struktur Direktori Project pada ReactJS	94
9.1	Mengenal Nodejs.....	95
10	Mengenal NPM dan NPX	96
11	Komponen pada ReactJS.....	98
12	Javascript	101
12.1	Dasar-Dasar JavaScript	101
12.2	Fungsi JavaScript	102
13	Implementasi Javascript.....	103
13.1	Vanilla JavaScript vs. Advanced JavaScript	104
13.2	JQuery	105
13.3	React JS.....	106
13.4	Tipe Data JavaScript.....	106
13.5	Cara Menulis Menggunakan Kode JavaScript	107
13.6	Praktikum	108

Daftar Gambar

Gambar 1.1 Tag HTML.....	11
Gambar 2.1 Hex Color Codes	24
Gambar 2.2 Gambar gagal ditampilkan.....	28
Gambar 2.3 Gambar berhasil ditampilkan	29
Gambar 2.4 Struktur tabel.....	34
Gambar 3.1 Struktur CSS	40
Gambar 3.2 Selector tag CSS.....	41
Gambar 3.3 Selector Class CSS.....	42
Gambar 3.4 Dua selector class.....	43
Gambar 3.5 Contoh selector ID	44
Gambar 3.6 Contoh selector attribute	45
Gambar 3.7 Contoh selector universal	45
Gambar 4.1 Internal CSS	48
Gambar 4.2 External CSS	50
Gambar 4.3 Membuat file external CSS	50
Gambar 4.4 Hasil praktikum Exsternal CSS.....	54
Gambar 4.5 Inline CSS.....	55
Gambar 5.1 Perbedaan button pada bootstrap 2 dan 3	58
Gambar 5.2 Bootstrap 4 dirilis pada oktober 2014	58
Gambar 5.3 Bootstrap versi 5.....	59
Gambar 6.1 menampilkan background warna putih	68
Gambar 6.2 Menampilkan Background dan garis tepi.....	69
Gambar 6.3 Implementasi tata letak dua kolom.....	72
Gambar 6.4 Tata letak tiga kolom.....	74
Gambar 6.5 Hasil dari system grid otomatis.....	75
Gambar 6.6 Desain layout praktikum.....	76
Gambar 6.7 Hasil praktikum system grid	79
Gambar 6.8 Layout praktikum.....	79
Gambar 6.9 Hasil akhir praktikum system grid	81
Gambar 6.10 Fixed Layout.....	84

Gambar 6.11 Desain navbar.....	84
Gambar 6.12 Desain navbar dengan menu	85
Gambar 6.13 Menamahkan register dan login pada navbar	86
Gambar 7.1 Struktur ReactJs.....	88
Gambar 7.2 Komponen ReactJs.....	89
Gambar 8.1 Hasil Latihan jsx.....	92
Gambar 8.2 Hasil praktikum JSX	94
Gambar 9.1 Kepopuleran Javascript.....	95
Gambar 10.1 perbedaan NPM dan NPX.....	97
Gambar 10.2 Elemen pada NPM.....	98
Gambar 11.1 Komponen pada ReactJs	99
Gambar 11.2 Method Render	100

Daftar Tabel

Tabel 1 Properties tabel34

Tabel 2 Attribut dan Deskripsi dari input type38

Tabel 3 Properties CSS warna dan backgroud.....51

Tabel 4 Properties CSS selector class.....51

Tabel 5 Properties attribute CSS52

Tabel 6 Tabel Container64

Tabel 7 Tabel system grid70

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 penggunaanya 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>
```

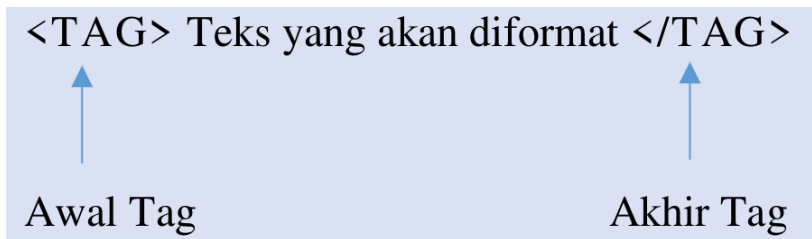
1.3 HTML (Hypertext Markup Language)

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

HTML merupakan format standar yang digunakan untuk membuat dokumen halaman web. Dokumen HTML memiliki sintaks tanda berpasangan yang disebut TAG untuk memberikan efek yang diinginkan. Secara umum format TAG adalah sebagai berikut :



Gambar 1.1 Tag HTML

Setiap dokumen HTML harus diawali dengan tag `<html>` dan diakhiri dengan komplementennya, yakni tag `</html>`. Dokumen HTML juga menyertakan tiga pasang tag.

- Tag `<head>` dan `</head>`: digunakan untuk menyatakan informasi mengenai dokumen HTML.
- Tag `<title>` dan `</title>`: digunakan untuk menambahkan title di title bar browser.
- Tag `<body>` dan `</body>`: digunakan untuk melingkupi semua teks yang terdapat di halaman HTML.

2 HTML tag

2.1 Memulai HTML

Salah satu persiapan yang harus dilakukan sebelum mempelajari pemrograman berbasis web adalah melakukan instalasi Xampp. Xampp adalah salah satu contoh dari web server. Web server ini nantinya akan berguna untuk menjadi server lokal yang dapat digunakan untuk menjalankan halaman web yang kita buat. Ikuti langkah-langkah berikut ini untuk petunjuk instalasi Xampp.

- 1) Download XAMPP yang sesuai dengan sistem operasi yang Anda gunakan pada link <https://www.apachefriends.org/download.html>
- 2) Install XAMPP yang sudah di download pada langkah 1
- 3) Buka browser Anda, ketikkan localhost kemudian jalankan. Catat dan jelaskan apa hasil yang ditampilkan, sertakan pula hasil tangkapan layar (screenshot) (soal No. 1)
- 4) Kemudian kita lanjutkan dengan memulai praktikum sederhana dengan menggunakan XAMPP.
- 5) Buka folder XAMPP di direktori tempat Anda menginstall XAMPP. Buka direktori htdocs kemudian buat satu folder baru bernama desWebExmp.
- 6) Kemudian buka text editor Anda, buat file baru bernama hello.html dan simpan di dalam direktori desWebExmp yang baru saja Anda buat.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>
      HTML Sederhana
    </title>
  </head>
  <body>
    Hello World
  </body>
</html>
```

- 7) Buka alamat pada browser anda lalu ketikkan :
localhost/desWebExmp/hello.html
- 8) Amati dan catat hasil praktikum yang sudah anda kerjakan.

2.2 Format Teks

HTML menyediakan beragam elemen yang dapat dimanfaatkan untuk pemformatan teks. Diantaranya adalah:

- 1) Format text untuk huruf tebal,
- 2) Format text untuk huruf miring,
- 3) Menambahkan garis bawah pada text,
- 4) Menambahkan coretan pada text,
- 5) Membuat pangkat,
- 6) Membuat marker,
- 7) Membuat paragraf dan,
- 8) Membuat heading.

2.2.1 Heading

Heading adalah tag penting dalam penulisan setiap halaman website. Dengan menggunakan heading maka struktur dokumen HTML akan terlihat rapi dan terorganisir. Berbagai mesin pencarian seperti google dan bing menggunakan Heading untuk membuat index pencarian.

Heading tidak disarankan untuk digunakan dengan tujuan mengubah format huruf suatu text, misal untuk mengubah suatu text menjadi tebal dan besar. Heading hanya digunakan untuk judul dan sub judul pada setiap halaman HTML. Heading didefinisikan menggunakan tag `<hn>` dan diakhiri dengan `</hn>`, di mana n menyatakan tipe dengan nilai 1 - 6.

Praktikum Heading

Langkah Praktikum:

- 1) Buat sebuah file bernama heading.html dan simpan file tersebut di dalam folder dewWebExmp yang telah dibuat pada Praktikum sebelumnya.
- 2) Ketikkan kode di bawah ini dalam **heading.html**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Membuat Heading</title>
  </head>
  <body>
    <h1> Heading 1 </h1>
    <h2> Heading 2 </h2>
```

```
<h3> Heading 3 </h3>
<h4> Heading 4</h4>
<h5> Heading 5 </h5>
<h6> Heading 6 </h6>
</body>
</html>
```

- 3) Buka browser anda lalu ketikkan alamat :
localhost/desWebExmp/heading.html
- 4) Amati tampilan yang muncul pada browser anda dan catat hasilnya.
- 5) Tambahkan atribut align di dalam tag heading seperti di bawah ini dan simpan file dengan nama **headingAlign.html**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Membuat Heading</title>
  </head>
  <body>
    <h1 align="left"> Heading 1 </h1>
    <h2 align="right"> Heading 2 </h2>
    <h3 align="center"> Heading 3 </h3>
    <h4> Heading 4</h4>
    <h5> Heading 5 </h5>
    <h6> Heading 6 </h6>
  </body>
</html>
```

- 6) Catat dan amati perubahanya.

2.2.2 Paragraf

Sebagaimana teks pada umumnya, dokumen HTML dapat terdiri dari kumpulan paragraf. Dalam konteks HTML, paragraf direpresentasikan dengan tag <p>. Tag <p> sebenarnya

merupakan tag pasangan, meski dalam implementasinya kerap kali diabaikan.

Praktikum membuat Paragraf

Langkah Praktikum:

- 1) Buat sebuah file bernama `paragraf.html` dan simpan file tersebut di dalam folder `dewWebExmp`.
- 2) Ketikkan kode di bawah ini dalam **paragraf.html**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Membuat Paragraf</title>
  </head>
  <body>
    <p>Ini adalah paragraf pertama</p>
    <p>Ini adalah paragraf kedua</p>
  </body>
</html>
```

- 3) Buka browser anda kemudian panggil dokumen htmlnya.
- 4) Amati dan catat hasilnya.
- 5) Untuk mengatur posisi paragraf kita dapat memanfaatkan atribut `align` yang diletakkan dalam tag paragraf.
- 6) Tambahkan atribut `align` di dalam tag paragraf seperti di bawah ini dan simpan file dengan nama **paragrafAlign.html**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Membuat Paragraf</title>
  </head>
  <body>
```



```
<p align="center">Ini adalah paragraf pertama</p>
<p align="left">Ini adalah paragraf kedua</p>
</body>
</html>
```

7) Amati dan catat perubahanya

2.3 Menggunakan Font

HTML menyediakan sejumlah elemen yang dapat dimanfaatkan untuk mengatur font seperti huruf tebal, huruf miring, garis bawah dan masih banyak lagi. Sebagai tambahan, di sini juga akan dijelaskan mengenai cara mencetak tag. Seperti diketahui, tag `<p>` di dokumen secara otomatis akan diterjemahkan sebagai paragraf. Adapun untuk mencetak karakter `<p>` di layar, kita perlu menggunakan nama entitas. Sebagai contoh, karakter `<` dinyatakan dengan nama entitas `<` dan karakter `>` dinyatakan dengan `>`.

Praktikum menggunakan Font

Langkah Praktikum:

- 1) Buat sebuah file bernama `font.html` dan simpan file tersebut di dalam folder `dewWebExmp`.
- 2) Ketikkan kode di bawah ini dalam **font.html**

```
<html>
  <head>
    <title>Pengaturan Font</title>
  </head>
  <body>
```

```

<b>Menggunakan tag <b></b><br>
<strong>Menggunakan tag
</strong></strong></strong><br>
<i>Menggunakan tag <i></i><br>
<em>Menggunakan tag <em></em><br>
<u>Menggunakan tag <u></u><br>
<del>Menggunakan tag
</del></del></del>
</body>
</html>

```

- 3) Buka browser dan panggil dokumen html anda.
- 4) Amati dan catat hasil praktikum yang sudah anda kerjakan.

2.4 Karakter Khusus

Dalam HTML, kita juga bisa menampilkan karakter-karakter khusus dengan memanfaatkan nama entitas. Tabel berikut memperlihatkan beberapa jenis karakter khusus yang dapat digunakan beserta nama entitasnya.

1. & => &
2. < => kurang dari (<)
3. > => lebih dari (>)
4. => Spasi (non-breaking space)
5. © => ©
6. ® => ®
7. ™ => ™

Karakter khusus yang ada bukan hanya karakter yang merupakan bagian dari kode HTML. Beberapa huruf yang mungkin tidak ada di keyboard bisa kalian tulis dengan cara yang sama. Berikut ini adalah huruf-huruf Kapital Yunani :

1. $A \Rightarrow$ Α
2. $B \Rightarrow$ Β
3. $\Gamma \Rightarrow$ Γ
4. $\Delta \Rightarrow$ Δ
5. $E \Rightarrow$ Ε
6. $Z \Rightarrow$ Ζ
7. $H \Rightarrow$ Η
8. $\Theta \Rightarrow$ Θ
9. $I \Rightarrow$ Ι
10. $K \Rightarrow$ Κ
11. $\Lambda \Rightarrow$ Λ
12. $M \Rightarrow$ Μ
13. $N \Rightarrow$ Ν
14. $\Xi \Rightarrow$ Ξ
15. $O \Rightarrow$ Ο
16. $\Pi \Rightarrow$ Π
17. $P \Rightarrow$ Ρ
18. $\Sigma \Rightarrow$ Σ
19. $T \Rightarrow$ Τ
20. $Y \Rightarrow$ Υ
21. $\Phi \Rightarrow$ Φ
22. $X \Rightarrow$ Χ
23. $\Psi \Rightarrow$ Ψ
24. $\Omega \Rightarrow$ Ω

Selain karakter-karakter yg sudah saya sebutkan, ada beberapa karakter khusus lainnya yang digunakan dalam HTML. Semua karakter khusus tersebut diapit dengan "&" dan ";".

Beberapa karakter khusus ditulis dengan angka yang diapit dengan tanda "&#" dan titik koma(;). Angka yang ditulis diantara

kedua pasang tanda tersebut adalah angka yang mewakili unicode dari karakter yang ditampilkan. Contoh penggunaan unicode untuk menampilkan karakter, misalnya :

1. `α` => α
2. `β` => β
3. `γ` => γ
4. `δ` => δ
5. `ε` => ϵ
6. `ζ` => ζ
7. `η` => η
8. `θ` => θ
9. `ι` => ι
10. `κ` => κ
11. `λ` => λ
12. `μ` => μ
13. `ν` => ν
14. `ξ` => ξ
15. `ο` => \omicron
16. `π` => π
17. `ρ` => ρ
18. `ς` => ς
19. `σ` => σ
20. `τ` => τ
21. `υ` => υ
22. `φ` => ϕ
23. `χ` => χ
24. `ψ` => ψ
25. `ω` => ω
26. `∀` => \forall

Praktikum karakter khusus
Langkah praktikum: 1) Buat sebuah file bernama karakter.html dan simpan file

tersebut di dalam folder dewWebExmp

2) Ketikkan kode dibawah ini

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Implementasi Karakter
    Khusus</title>
  </head>
  <body>
    &pound; Pound <br>
    &euro; Euro <br>
    &copy; Copyright <br>
    &reg; Registered <br>
    &trade; Trademark <br>
  </body>
</html>
```

3) Buka browser anda dan panggil dokumen html karakter.html

4) Amati dan catat hasilnya

2.5 Garis Horizontal

Di HTML, garis horizontal direpresentasikan melalui tag `<hr>`. Meskipun kebanyakan browser merender elemen ini dengan visualisasi yang sedikit berbeda, namun pada hakekatnya mencerminkan sebuah bentuk garis horizontal.

Praktikum membuat garis horizontal

Langkah praktikum:

- 1) Buat sebuah file bernama garishorizontal.html dan simpan file tersebut di dalam folder desWebExmp.

2) Ketikkan kode di bawah ini dalam **garishorizontal.html**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Implementasi Garis
Horizontal</title>
  </head>
  <body>
    Roger S. Pressman <hr>
    Roger S. Pressman is an American software
engineer <hr>
    He received a BSE from the University of
Connecticut <hr>
    Experience working as a software engineer
<hr>
  </body>
</html>
```

3) Buka browser dan panggil dokumen html anda

4) Amati dan catat hasilnya

2.6 Menggunakan List

HTML mendukung list dalam bentuk terurut (ordered), tak terurut (unordered) dan definisi (definition). Untuk setiap bentuk list ini, terdapat list item yang dinyatakan dengan tag dan yang merepresentasikan item-item list.

Praktikum menggunakan list

Langkah praktikum :

- 1) Buat sebuah file bernama list.html dan simpan file tersebut di dalam folder dewWebExmp.
- 2) Ketikkan kode di bawah ini dalam **list.html**

```
<html>
```

```

<head>
<title>Bekerja dengan List</title>
</head>
<body>
  Ordered List
  <ol>
    <li>satu</li>
    <li>dua</li>
    <li>tiga</li>
  </ol>
  <br>
  <hr>

  Unordered List
  <ul>
    <li>satu</li>
    <li>dua</li>
    <li>tiga</li>
  </ul>
  <br>
  <hr>

  Definition List
  <dl>
    <dt>satu</dt>
    <dd>satu satu</dd>
    <dd>satu dua</dd>
    <dt>dua</dt>
    <dd>dua dua</dd>
  </dl>
</body>
</html>

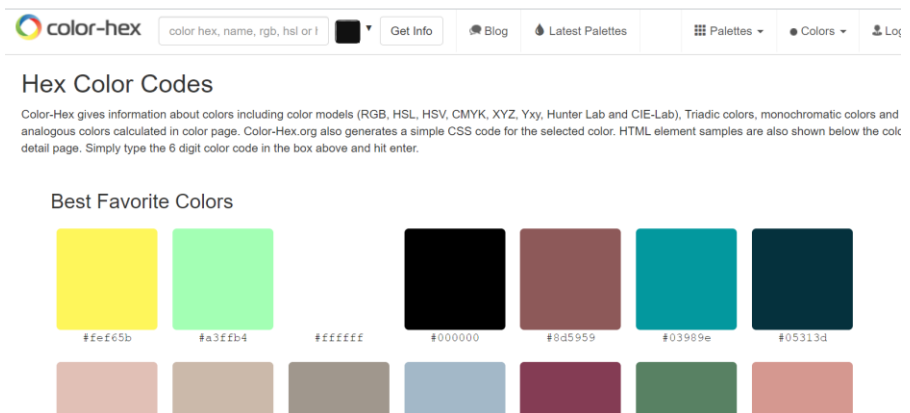
```

- 3) Buka browser dan panggil dokumen html anda
- 4) Amati dan catat hasilnya.

2.7 Pewarnaan

Untuk memberikan warna background, HTML menyediakan atribut bgcolor di tag <body>. Atribut ini dapat diisi dengan nama warna (misalnya red) atau kode heksadesimal (misalnya #FFFFFF). Khusus untuk elemen-elemen lain tertentu, tersedia

atribut color yang memungkinkan untuk melakukan pewarnaan. Sama seperti bgcolor, nilai atribut ini juga dapat berupa nama warna atau kode heksadesimal. Untuk menentukan warna hexa yang lebih detail anda bisa mengunjungi situs : <https://color-hex.org/> , pada situs tersebut terdapat lengkap dan detail dari warna hexa yang bisa anda manfaatkan.



Gambar 2.1 Hex Color Codes

Praktikum pewarnaan

Langkah praktikum

- 1) Buat sebuah file bernama warna.html dan simpan file tersebut di dalam folder dewWebExmp.
- 2) Ketikkan kode di bawah ini dalam **warna.html**

```
<html>
<!-- Memberi warna aqua di body -->
<head>
  <title>Pewarnaan</title>
</head>
```



```
<body bgcolor="#672d71">
  <h3 align="center">Heading 3</h3>
  <font color="red">Font berwarna
merah</font><br><br>
  <font color="#FF0000">Font berwarna merah
(menggunakan nilai heksa)</font>
</body>
</html>
```

- 3) Buka browser anda dan jalankan dokumen html yang sudah anda buat.
- 4) Amati dan catat hasil praktikumnya.

2.8 Bekerja dengan Gambar

Tak hanya teks, kita juga bisa menyisipkan gambar di dalam dokumen HTML. Untuk keperluan ini, HTML menyediakan tag `` yang didukung dengan sejumlah atribut.

Praktikum Implementasi Gambar

Langkah praktikum :

- 1) Langkah 1: Memastikan Format Gambar Sudah Sesuai
Langkah pertama yaitu memilih format gambar yang tepat untuk website Anda. Pastikan gambar yang ingin Anda masukkan ke dokumen HTML memiliki format yang sesuai. Jika format gambar tidak kompatibel, kemungkinan besar gambar akan gagal ditampilkan.
Berikut format gambar yang dapat disematkan pada dokumen HTML.

Format Gambar	Ekstensi Gambar
APNG	.apng
GIF	.gif
ICO	.ico atau .cur
JPEG	.jpg, .jpeg, .jfif, .pjpeg, .jpp
PNG	.png
SVG	.svg

2) Langkah 2: Memasukkan Atribut Sumber Gambar pada Tag HTML Images

Untuk menyematkan gambar pada website, Anda perlu menggunakan tag `` pada dokumen HTML. Sebab, cara memasukkan gambar di HTML berbeda dengan menempel gambar pada dokumen biasa. Apa sih perbedaannya?

HTML tidak bisa langsung menampilkan gambar begitu saja. Dengan kata lain, HTML harus mengambil gambar dari suatu tempat penyimpanan atau direktori (drive, website, dan lain-lain). Oleh karena itu, diperlukan atribut untuk memanggil gambar tersebut agar bisa tampil pada browser.

Atribut untuk menarik gambar dari server penyimpanan

tersebut dinamakan Src. Isi dari atribut ini bisa berupa link menuju direktori ataupun nama file gambar.

Untuk menggunakan atribut Src, Anda cukup memasukkan tag berikut.

```

```

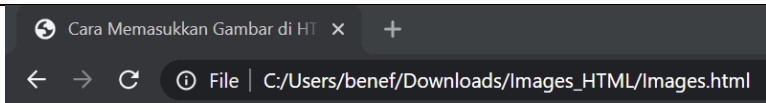
Sedangkan pada text editor, Anda bisa langsung menempelkan syntax ini. Mudah, bukan?

Buat dokumen html baru kemudian beri nama gambar.html dan masukkan tag berikut ini:


```
<!DOCTYPE html>
<html>
<head>
  <title>Cara Memasukkan Gambar di
  HTML</title>
</head>
<body>
  <p> Ini merupakan contoh gambar pada HTML
  
</body>
</html>
```

3) Langkah 3: Memberikan Atribut Alt Text pada Tag HTML Images

Atribut Alt berfungsi untuk memberikan teks alternatif pada gambar. Jadi, jika karena sebab tertentu gambar gagal ditampilkan, teks alternatif akan muncul sehingga pengunjung website tetap bisa mengidentifikasi gambar.



Ini merupakan contoh gambar pada HTML

 Gambar_Langit

Gambar 2.2 Gambar gagal ditampilkan

Selain itu, alt text pada gambar juga mampu meningkatkan ranking website. Sebab, alt text memudahkan mesin pencarian dalam memahami deskripsi dan konteks gambar. Dengan begitu, konten website pun akan lebih mudah diindeks dalam SERP.

Supaya gambar memiliki alt text, Anda perlu menambahkan atribut alt text di dalam tag .

```

```

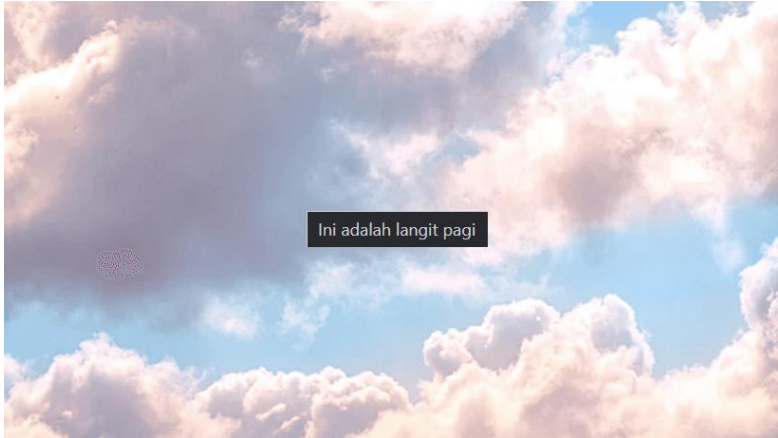
Untuk penggunaan atribut alt text dalam dokumen HTML bisa Anda temukan pada cara di bawah ini.

```
<!DOCTYPE html>
<html>
<head>
  <title>Cara Memasukkan Gambar di
  HTML</title>
</head>
<body>
  <p> Ini merupakan contoh gambar pada HTML
</p>
  
</body>
</html>
```

4) Langkah 4: Mencantumkan Keterangan Gambar

Dari poin sebelumnya, Anda sudah mengerti penggunaan

teks alternatif pada gambar. Namun, bagaimana jika Anda ingin mencantumkan keterangan yang bisa pengunjung lihat saat cursor mengarah ke gambar?



Gambar 2.3 Gambar berhasil ditampilkan

Jawabannya yaitu dengan menambahkan atribut title di dalam tag ``. Setelah menggunakan atribut ini, maka gambar bisa memiliki tooltips atau pesan yang muncul saat cursor diarahkan ke objek. Selain itu, halaman website juga tetap luas karena keterangan gambar tidak memakan tempat.

Tooltips pada dokumen HTML bisa Anda dapatkan melalui syntax ini.

```

```

Penempatan syntax atribut title silakan melihat contoh berikut.

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

```
<title>Cara Memasukkan Gambar di
HTML</title>
</head>
<body>
<p> Ini merupakan contoh gambar pada HTML
</p>

</body>
</html>
```

5) Amati setiap Langkah praktikumnya dan catat hasilnya.

2.9 Link pada halaman Web

Link pada HTML adalah Hyperlink. Hyperlink adalah sebuah elemen, teks atau sebuah image yang dapat diklik dan kemudian menuju ke dokumen lain. Browser akan menyorot (highlight) teks atau gambar yang diidentifikasi sebagai link dengan warna dan/atau garis bawah untuk menunjukkan bahwa itu adalah hypertext link (hyperlink atau link). Tag Anch <a> digunakan untuk membuat suatu link kepada dokumen lain dengan menambah atribut href sebagai definisi lokasi link. Berikut adalah sintaks link pada HTML:

teks yang mengandung link

- atribut href memberikan alamat tujuan link
- teks yang mengandung link adalah tulisan yang akan tampak pada halaman

web yang berisi link, dan jika diklik maka halaman akan menuju ke alamat yang ada di dalam atribut href. Berikut adalah contoh penulisan link pada html:

```
<a href="http://www.microsoft.com">click here to show it</a>
```

Macam-macam Link :

1. Link Relative

Link lokal yang memberikan link ke dalam website yang sama yang dispesifikasikan dengan sebuah URL (tanpa http://www.)

2. Link Absolute

Digunakan untuk membuat link ke halaman web lain yang berada pada website lain di internet.

3. Link dalam Dokumen yang Sama

Link ini dibuat untuk dokumen yang panjang sekali, sehingga apabila ditampilkan dalam browser web akan mengharuskan kita melakukan scroll layar berulang-ulang.

Navigasi untuk penelusuran dokumen dapat dimudahkan dengan membuat link antar bagian dengan menandai setiap bagian tersebut dengan memberinya nama, sehingga pada tempat di dalam dokumen akan ada bagian yang bernama dan di bagian lainnya dapat diletakkan link untuk menuju ke bagian-bagian tersebut.

- Langkah-langkah memberi nama suatu bagian dalam dokumen :
- Letakkan kursor pada baris atau teks yang menjadi awal dari bagian tersebut
- Sisipkan nama bagian tersebut dengan : ``
- Membuat link yang menuju pada bagian dokumen yang sama dapat dilakukan dengan cara yang sama seperti link absolut atau relatif, namun nama dokumen dalam link diganti dengan nama_bagian dokumen dengan ditambah tanda # yang ditulis sebelum nama bagian tersebut.

Contoh:

`Bagian tentang link `

Praktikum

Langkah praktikum:

- 1) Buat sebuah file bernama macamLink.html dan simpan file tersebut di dalam folder dewWebExmp.
- 2) Ketikkan kode di bawah ini dalam macamLink.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Membuat Link</title>
  </head>
  <body>
    <a name="TOP">
      <h4>Macam-macam Link</h4>
      <a href="#link Relatif">Link
Relatif</a></br>
```



```

<a href="#link Absolut">Link
Absolut</a></br>
<a href="#link with New Window">Link
with New Window</a></br>
<a href="#link ke email">Link ke
email</a></br>
</a></br>

<a name="link Relatif">
<h1>Ini adalah contoh link relatif</h1>
<p>
    Klik <a href="gambar.html">disini</a>
Jika anda ingin menuju
    ke halaman berikutnya</br>
    <a href="#TOP">kembali ke menu</a>
</p>
</a></br>

<a name="link Absolut">
<h1>Ini adalah contoh link absolut</h1>
<p>
    Buka <a href="http://www.google.com"
target="_blank">www.google.com</a>
    di halaman baru </br>
    <a href="#TOP">kembali ke menu</a>
</p>
</a></br>

<a name="link ke email">
<h1>Ini adalah contoh link relatif</h1>
<p>
    Anda bisa menghubungi saya di <a
href="mailto:yrevsanal0@gmail.com">
    email ini </a>jika ada pertanyaan
lebih lanjut</br>
    <a href="#TOP">kembali ke menu</a>
</p>
</a></br>
</body>
</html>

```

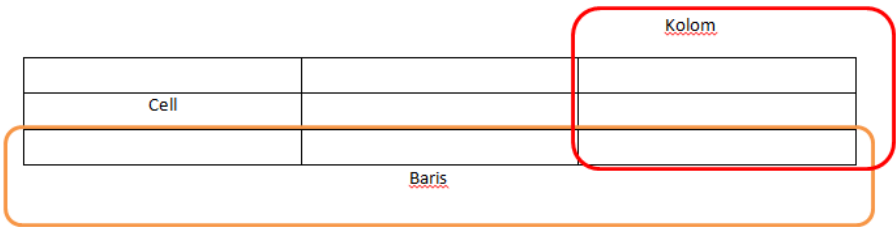
3) Amati dan catat hasil praktikumnya.

2.10Tabel dan Pengaturannya

Dalam sebuah halaman web, tabel dijadikan kerangka untuk meletakkan komponen-komponen isi web, sehingga isi dalam sebuah web dapat diatur dengan rapi. Pembuatan tabel dalam HTML dimulai dengan tak <table> dan diakhiri dengan tag </table>.

Tabel 2.1 Properties tabel

<table>	tag yang menerangkan kepada browser bahwa itu adalah sebuah tabel
<tr>	tag Tabel Row yang digunakan untuk membuat baris dalam tabel
<td>	tag Tabel Data yang digunakan untuk meletakkan data yang ingin kita isikan dalam tabel, TD dapat juga kita artikan sebagai kolom



Gambar 2.4 Struktur tabel

Sintaks pembuatan tabel adalah sebagai berikut :

```
<table atribut="value">
<tr>
<td atribut="value"> ....isi ....</td>
<td atribut="value"> ....isi1....</td>
</tr>
<tr>
.....
.....
</tr>
</table>
```

Praktikum membuat tabel

Langkah praktikum:

- 1) Pertama silakan buat file html baru. Beri nama file html tersebut sesuai selera atau dapat kita beri nama "tabel.html". Silakan perhatikan contoh kode berikut dan salin ke dalam file **tabel.html**.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>Menenal Tabel HTML</title>
</head>
<body>
  <table border="1">
```

```
<tr>
  <td>Baris ke 1 - Kolom ke 1</td>
  <td>Baris ke 1 - Kolom ke 2</td>
</tr>
<tr>
  <td>Baris ke 2 - Kolom ke 1</td>
  <td>Baris ke 2 - Kolom ke 2</td>
</tr>
</table>
</body>
</html>
```

- 2) Pada kode di atas terdapat atribut border dalam tag table, atribut border tersebut berfungsi untuk memberikan garis pada tabel. Sedangkan angka 1 di dalam atribut border merupakan nilai ketebalan garis yang akan ditampilkan, semakin besar nilai tersebut maka semakin tebal garis yang ditampilkan.
- 3) Amati dan catat hasil praktikum anda

2.11 Form pada HTML

Form merupakan tempat penginputan data sebelum diproses oleh sistem. Salah satunya adalah form login, form comment, form data user, dan lain sebagainya. untuk proses penginputan data dari form akan kita bahas di materi PHP, MYSQL, dan JavaScript. tag yang digunakan untuk membuat form ini adalah tag `<form>` yang didalamnya bisa berupa `<input>`, `<textarea>`, `<option>` dan `<select>`.

Form memiliki atribut berupa action dan method. Atribut action menentukan URL yang akan dijalankan dan menerima semua masukan pada formulir. Jika action tidak disebutkan akan dikirim ke URL yang sama dengan halaman web itu sendiri, sedangkan atribut method berisikan metode form melakukan proses pengiriman (GET/POST).

- GET→informasi dikirim dengan URLnya
- POST→informasi dikirim terpisah dengan URLnya.

Sintaks:

```
<form action= "action" method="GET | POST">  
....  
</form>
```

Elemen input berfungsi untuk mendefinisikan input yang akan dimasukkan pengguna. Elemen ini mempunyai atribut yaitu name, size, type, value, checked. Atribut name mendefinisikan nama dari input kontrol form, atribut size mendefinisikan ukuran teks pada input kontrol, type mendefinisikan bentuk-bentuk input kontrol, value mendefinisikan nilai awal/reset/submit, checked mendefinisikan pilihan terpilih pada type radio/checkbox. Element ini tidak mempunyai tag penutup dan harus berada di dalam elemen FORM. Sintaks:

```
<input name= "name" size= "number" type= "value"  
value= "isi yang ditampilkan di browser" checked>
```

Atribut type dapat berisi nilai (value) berikut ini:

Tabel I.1 Atribut dan Deskripsi dari input type

Value	Deskripsi
button	Mendefinisikan sebuah tombol yang bisa diklik (biasanya digunakan dengan sebuah script Javascript)
Checkbox	Mendefinisikan sebuah checkbox
color (HTML5)	Mendefinisikan color picker
date (HTML5)	Mendefinisikan pengaturan tanggal (year, month, day(no time))
datetime- local (HTML5)	Mendefinisikan sebuah pengaturan tanggal dan waktu (year, month, day, time (no timezone))
email (HTML5)	Mendefinisikan sebuah tempat untuk sebuah alamat email.
File	Mendefinisikan sebuah area untuk memilih sebuah file dan sebuah tombol "Browse" (untuk upload file)
hidden	Mendefinisikan sebuah area input yang disembunyikan
image	Mendefinisikan sebuah gambar sebagai tombol
month (HTML5)	Mendefinisikan sebuah pengaturan bulan dan tahun (no timezone)
number (HTML5)	Mendefinisikan sebuah area untuk memasukkan sebuah angka
password	Mendefinisikan sebuah area password
radio	Mendefinisikan sebuah radio button
range (HTML5)	Mendefinisikan sebuah pengaturan range (seperti slide control)
reset	Mendefinisikan tombol reset
search (HTML5)	Mendefinisikan sebuah text area untuk memasukkan string pencarian
Submit	Mendefinisikan tombol submit
tel (HTML5)	Mendefinisikan text area untuk memasukkan sebuah nomor telepon
Text	Default. Mendefinisikan sebuah text area satu baris.
time (HTML5)	Mendefinisikan sebuah pengaturan untuk memasukkan waktu (no timezone)

url (HTML5)	Mendefinisikan sebuah area untuk memasukkan alamat URL
Week	Mendefinisikan sebuah pengaturan minggu dan tahun (no timezone)

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

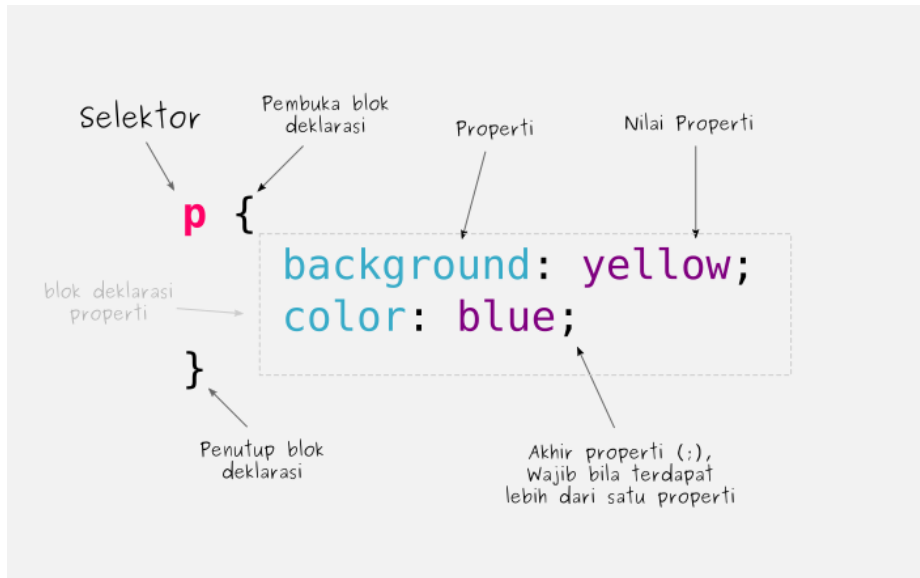
3.3 Struktur Kode atau Sintaks Dasar CSS

Struktur kode CSS terdiri dari tiga bagian:

- 1) Selektor;

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

Contoh:



Gambar 3.1 Struktur CSS

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:

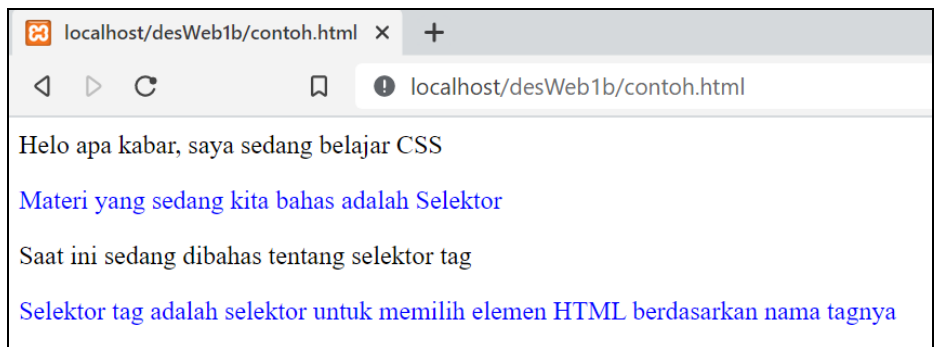
```
p {  
    color: blue;  
}
```

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:



Gambar 3.2 Selector tag CSS

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:

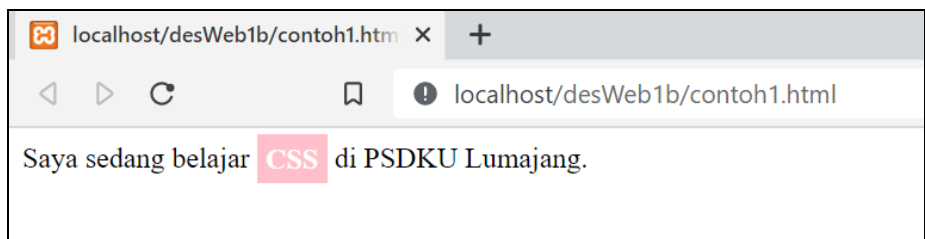
```
.pink {  
  color: white;  
  background: pink;  
  padding: 5px;  
}
```

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 :



Gambar 3.3 Selector Class CSS

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

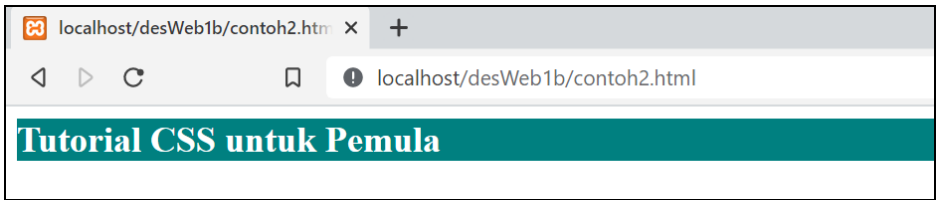
Contoh:

```
.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:



Gambar 3.4 Dua selector class

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 :



Gambar 3.5 Contoh selector ID

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:

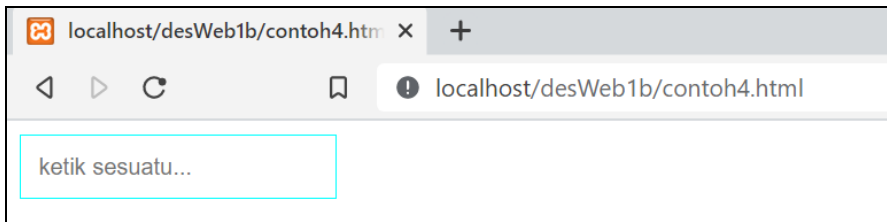
```
input[type=text] {  
    background: none;  
    color: cyan;  
    padding: 10px;  
    border: 1px solid cyan;  
}
```

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

Contoh kode HTML:

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

Hasilnya :



Gambar 3.6 Contoh selector attribute

3.3.1.5 Selector Universal

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

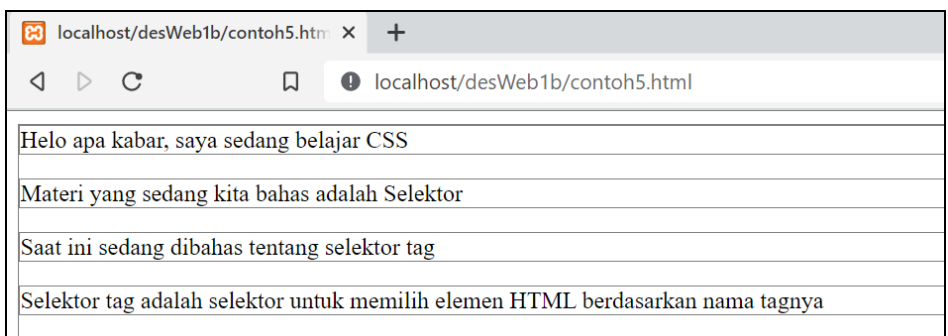
Contoh:

```
* {
  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 :



Gambar 3.7 Contoh selector universal

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 :



Gambar 4.1 Internal CSS

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 mendemonkan <i>internal css</i>. Seperti namanya,  
<i>inline CSS</i> adalah kode CSS yang ditulis  
langsung dalam file HTML.</p>  
</body>  
</html>
```

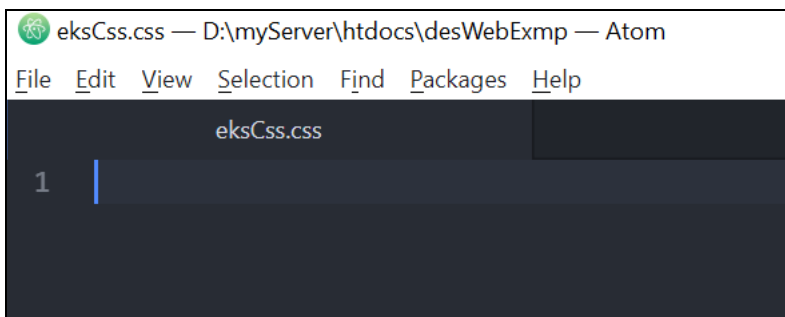
Hasilnya :



Gambar 4.2 External CSS

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.



Gambar 4.3 Membuat file external CSS

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

Tabel 4.1 Properties CSS warna dan backgroud

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 :

Tabel II.2 Properties CSS selector class

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 :

Tabel 4.3 Properties attribute CSS

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



Gambar 4.4 Hasil praktikum Exsternal CSS

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:

```

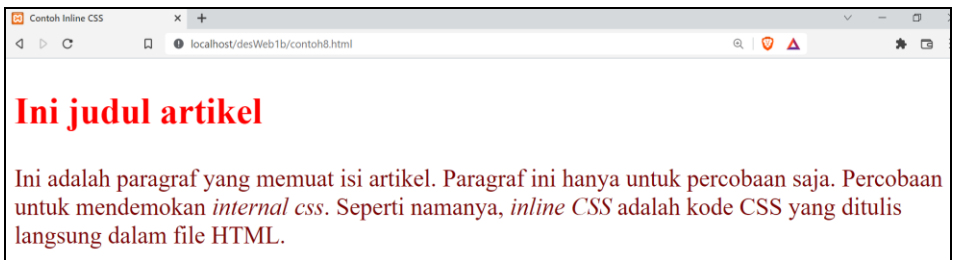
<!DOCTYPE html>
<html>
<head>
  <title>Contoh Inline CSS</title>
</head>

<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 :



Gambar 4.5 Inline CSS

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**



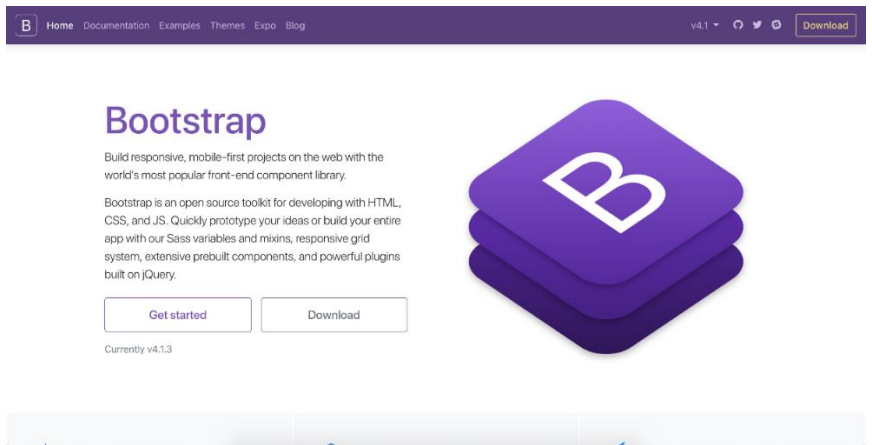
Buttons in **Bootstrap 3**



Gambar 5.1 Perbedaan button pada bootstrap 2 dan 3

5.2.2 Bootstrap 4

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



Gambar 5.2 Bootstrap 4 dirilis pada oktober 2014

Beberapa perubahan pada rilis ini:

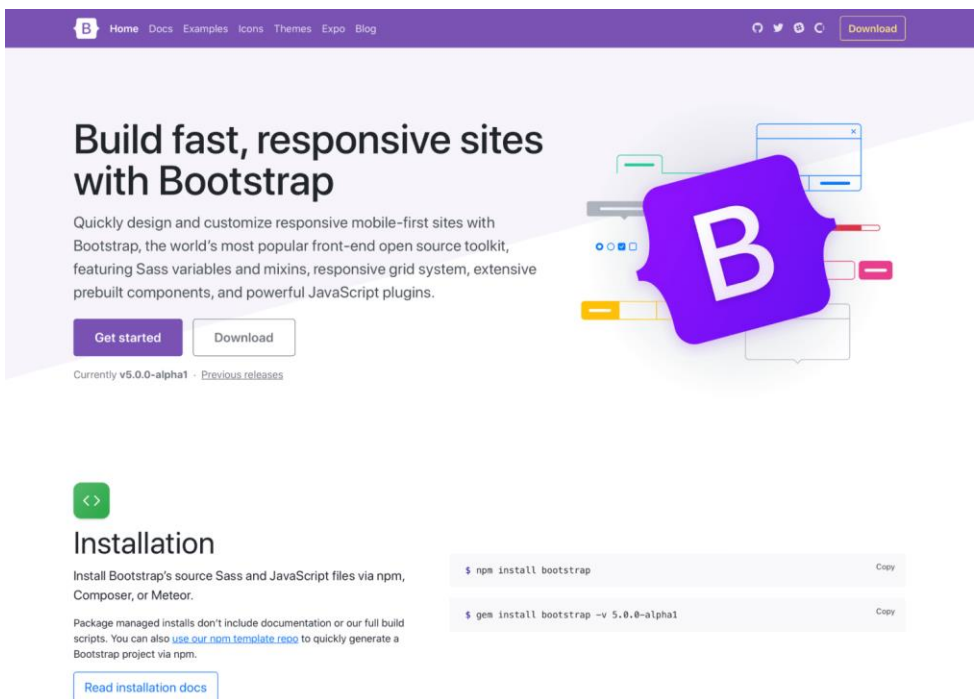
- Penulisan ulang kode mayor;
- Mengganti Less dengan Sass;

- 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.



Gambar 5.3 Bootstrap versi 5

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

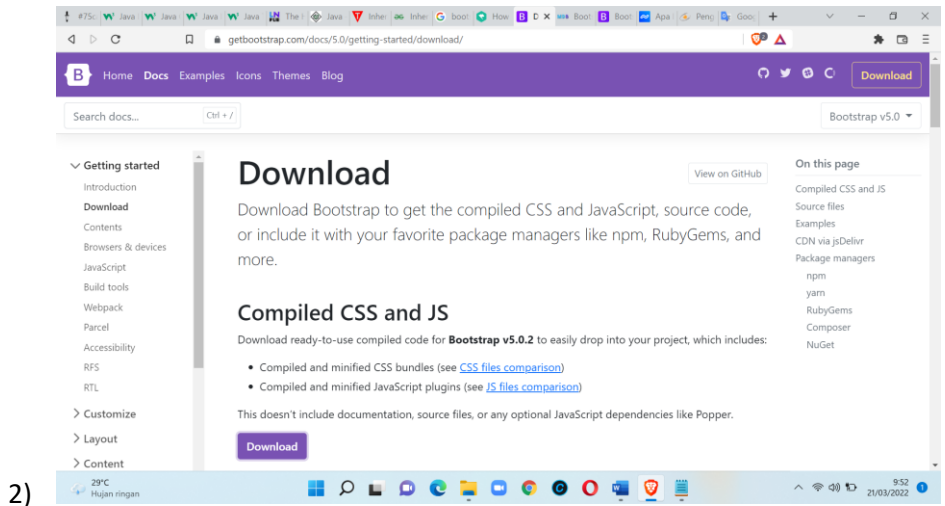
6 Implementasi bootstrap pada halaman WEB

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



Gambar 6.1 Halaman download Bootstrap versi 5

- 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

Gambar 6.2 File CSS dan Js pada Bootstrap versi 5

- 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.



Gambar 6.3 Hello World menggunakan Bootstrap

- 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 :

Tabel III.1 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

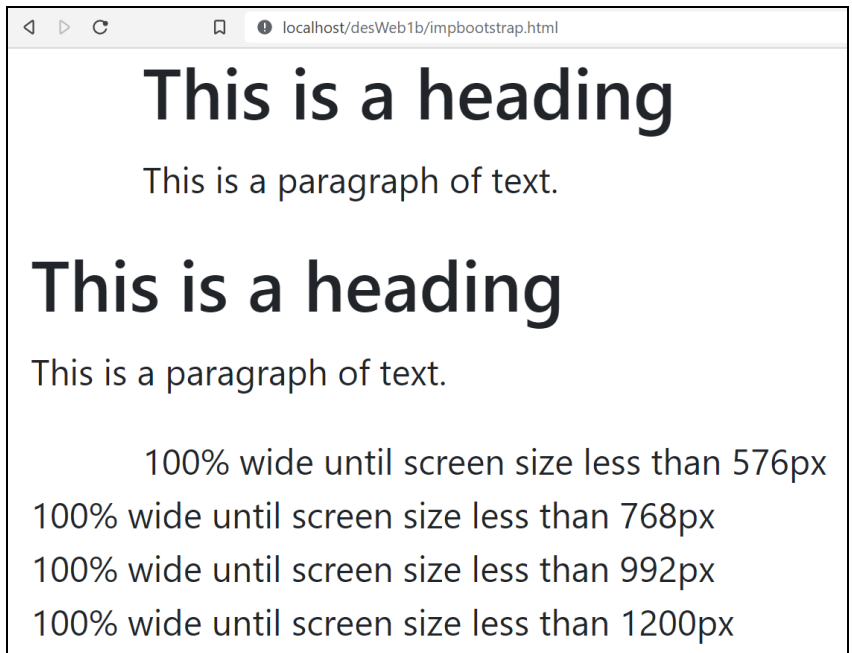
```
<!-- contoh .container -->
<div class="container">
  <h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</div>
```

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>
```



Gambar 6.4 Penerapan Container pada Bootstrap

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 :



Gambar 6.5 Menampilkan background menggunakan Bootstrap versi 5

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 :



Gambar 6.1 menampilkan background warna putih

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 :



Gambar 6.2 Menampilkan Background dan garis tepi

6.3 Sistem Grid pada Bootstrap

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

Kita dapat menggunakan Class grid Bootstrap yang telah ditentukan untuk membuat tata letak dengan cepat untuk berbagai jenis perangkat seperti ponsel, tablet, laptop, desktop, dan sebagainya. Misalnya, Kita dapat menggunakan `.col-*Class` untuk membuat kolom kisi untuk perangkat ekstra kecil seperti ponsel dalam mode potret, dan `.col-sm-*class` untuk ponsel dalam mode lanscape.

Demikian pula, Kita dapat menggunakan `.col-md-*class` untuk membuat kolom kisi untuk perangkat layar sedang seperti

tablet, `.col-lg-*class` untuk perangkat seperti laptop kecil, `.col-xl-*class` untuk laptop dan desktop, dan `.col-xxl-*class` untuk layar desktop besar.

Tabel berikut merangkum fitur-fitur utama dari sistem grid Bootstrap.

Tabel 6.2 Tabel system grid

Features Bootstrap Grid System	X-Small (xs) <576px	Small (sm) ≥576px	Medium (md) ≥768px	Large (lg) ≥992px	X-Large (xl) ≥1200px	XX- Large (xxl) ≥1400px
Container max-width	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>	<code>.col-xxl-</code>
Number of columns	12					
Gutter width	1.5rem (.75rem on left and right)					
Custom gutters	Yes					
Nestable	Yes					
Column ordering	Yes					

Tabel di atas menunjukkan satu hal penting, menerapkan `.col-sm-*class` apa pun ke elemen tidak hanya akan berpengaruh pada perangkat kecil, tetapi juga pada perangkat sedang, besar, dan ekstra besar (lebar viewport 768px), jika `.col-md-*`, `.col-lg-*`, `.col-xl-*`, atau `.col-xxl-*class` tidak ada .

Demikian pula, `.col-md-*class` tidak hanya akan berpengaruh pada perangkat menengah, tetapi juga pada perangkat besar dan ekstra besar jika `.col-lg-*`, `.col-xl-*`, atau `.col-xxl-*class` tidak ada.

6.3.1 Membuat Tata Letak Dua Kolom

Contoh berikut akan menunjukkan cara membuat tata letak dua kolom untuk perangkat berukuran sedang, besar, dan ekstra besar seperti meja, laptop, desktop, dll. Namun, pada ponsel (lebar layar kurang dari 768 piksel), kolom tersebut akan otomatis menjadi horizontal (2 baris , 1 kolom).

- A) Langkah pertama buat halaman HTML kemudian beri nama **systemGrid.html** dengan tag seperti berikut ini:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Membuat System Grid</title>
    <!-- memanggil bootstrap -->
    <link rel="stylesheet"
href=" ../desWebExmp/css/bootstrap.css">
  </head>
  <body>
  </body>
</html>
```

- B) Tambahkan tag berikut untuk menampilkan tata letak dua kolom pada system Grid :

```
<div class="container">
  <!--Membuat dua kolom yang sama besar
pada satu baris-->
  <div class="row">
    <div class="col-md-6">Column
left</div>
    <div class="col-md-6">Column
right</div>
  </div>

  <!--Membuat dua kolom dengan rasio 1:2--
>
  <div class="row">
```

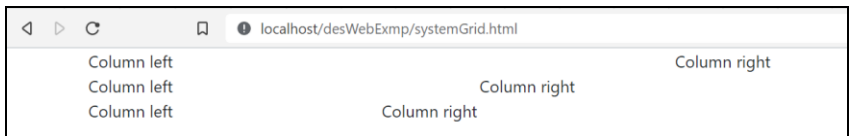
```

        <div class="col-md-4">Column
left</div>
        <div class="col-md-8">Column
right</div>
    </div>

    <!--Membuat dua kolom dengan rasio 1:3-->
>
    <div class="row">
        <div class="col-md-3">Column
left</div>
        <div class="col-md-9">Column
right</div>
    </div>
</div>

```

C) Berikut adalah hasil akhirnya



Gambar 6.3 Implementasi tata letak dua kolom

6.3.2 Membuat Layout Tiga Kolom

Demikian pula, Anda dapat membuat tata letak lain berdasarkan prinsip di atas. Misalnya, contoh berikut biasanya akan membuat tata letak tiga kolom untuk layar laptop dan desktop. Ini juga berfungsi di tablet dalam mode landscape jika resolusi layar lebih dari atau sama dengan 992 pixel (mis. Apple iPad). Namun, dalam mode potret, kolom kisi akan menjadi horizontal seperti biasa.

A) Langkah awal buatlah halaman html dengan nama **sgTigaKolom.html** dengan tag seperti berikut ini:


```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Membuat System Grid Tiga
Kolom</title>
    <link rel="stylesheet"
href="../desWebExmp/css/bootstrap.css">
  </head>
  <body>
  </body>
</html>

```

- B) Tambahkan tag berikut untuk menampilkan tata letak tiga kolom pada System Grid:

```

    <div class="container">
      <!--Membuat tiga kolom yang sama
besar-->
      <div class="row">
        <div class="col-lg-4">Column
left</div>
        <div class="col-lg-4">Column
middle</div>
        <div class="col-lg-4">Column
right</div>
      </div>

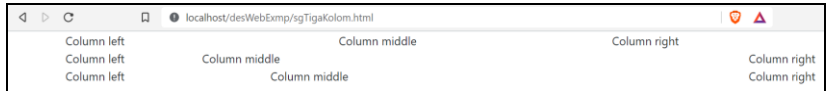
      <!--membuat tiga kolom dengan rasio
1:4:1-->
      <div class="row">
        <div class="col-lg-2">Column
left</div>
        <div class="col-lg-8">Column
middle</div>
        <div class="col-lg-2">Column
right</div>
      </div>

      <!--Amati kolom berikut-->
      <div class="row">
        <div class="col-lg-3">Column
left</div>
        <div class="col-lg-7">Column
middle</div>
        <div class="col-lg-2">Column

```

```
right</div>
</div>
</div>
```

C) Hasilnya



Gambar 6.4 Tata letak tiga kolom

6.3.3 Membuat Tata Letak Kolom secara Otomatis

Kita juga dapat membuat kolom dengan lebar yang sama untuk semua perangkat (x-small, small, medium, large, x-large, dan xx-large) hanya dengan menggunakan class `.col`, tanpa menentukan nomor kolom apa pun.

Mari kita coba contoh berikut untuk memahami cara kerjanya:

- A) Buatlah halaman html dengan nama **sgOtomatis.html** kemudian tambahkan tag berikut ini:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>System Grid Otomatis</title>
    <link rel="stylesheet"
href="../desWebExmp/css/bootstrap.css">
  </head>
  <body>
  </body>
</html>
```

- B) Tambahkan tag berikut ini ke dalam halaman html yang anda buat

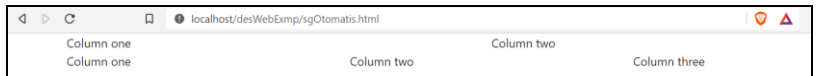
```

<div class="container">
  <!--Dua kolom yang sama-->
  <div class="row">
    <div class="col">Column one</div>
    <div class="col">Column two</div>
  </div>

  <!--Tiga kolom yang sama-->
  <div class="row">
    <div class="col">Column one</div>
    <div class="col">Column two</div>
    <div class="col">Column three</div>
  </div>
</div>

```

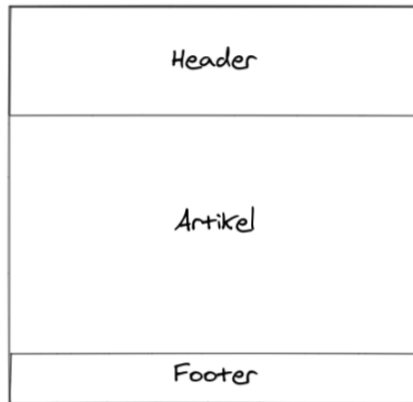
C) Hasilnya



Gambar 6.5 Hasil dari system grid otomatis

6.3.4 Praktikum System Grid

- A) Pada praktikum kali ini, kita akan membuat layout 1 kolom dan ada tiga komponen penting yang ada di dalam kolom tersebut, yakni Header, Artikel, dan Footer.
- B) Berikut adalah desain atau layout akan kita buat:



Gambar 6.6 Desain layout praktikum

- C) Langkah pertama kita buat halaman html dengan nama **praktikumSystemGrid.html**, kemudian tambahkan tag html seperti berikut:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Praktikum System Grid</title>
    <link rel="stylesheet"
href="../desWebExmp/css/bootstrap.css">
    <script
src="../desWebExmp/js/bootstrap.js"></script>
  </head>
  <body>
  </body>
</html>
```

- D) Tambahkan tag berikut dalam body dan amati perubahannya:

```
<body>
  <header class="container bg-primary text-
white">
  </header>

  <main class="container border">
```

```
</main>

<footer class="container bg-
light"></footer>
</body>
```

E) Pada tag header tambahkan tag berikut ini dan amati perubahannya:

```
<div class="row">
  <div class="col-12 py-4 text-center">
    <h1 class="display-1">Tutorial
Bootstrap</h1>
    <p class="lead">Belajar System
Grid</p>
  </div>
</div>
```

F) Pada tag main tambahkan tag berikut dan amati perubahannya:

```
<div class="row">
  <div class="col-12 py-5">
    <h1>Roger S Pressman</h1>
    <p>
      Roger S. Pressman adalah seorang
insinyur perangkat lunak Amerika , penulis dan
konsultan , dan Presiden RS Pressman &
Associates. Ia juga Pendiri dan Direktur Teknik
untuk EVANNEX, sebuah perusahaan yang menjual
suku cadang dan aksesoris untuk kendaraan
listrik.

      Ia menerima gelar BSE dari
University of Connecticut , MS dari University
of Bridgeport dan PhD dari University of
Connecticut. Dia memiliki lebih dari 40 tahun
pengalaman bekerja sebagai insinyur perangkat
lunak, manajer, profesor, penulis, dan
konsultan, dengan fokus pada masalah rekayasa
perangkat lunak. [1] Dia pernah menjadi Dewan
Editorial IEEE Software dan The Cutter IT
Journal. Dia adalah anggota IEEE dan Tau Beta
Pi . Pressman telah merancang dan mengembangkan
```

produk yang digunakan di seluruh dunia untuk pelatihan rekayasa perangkat lunak dan peningkatan proses. [2]

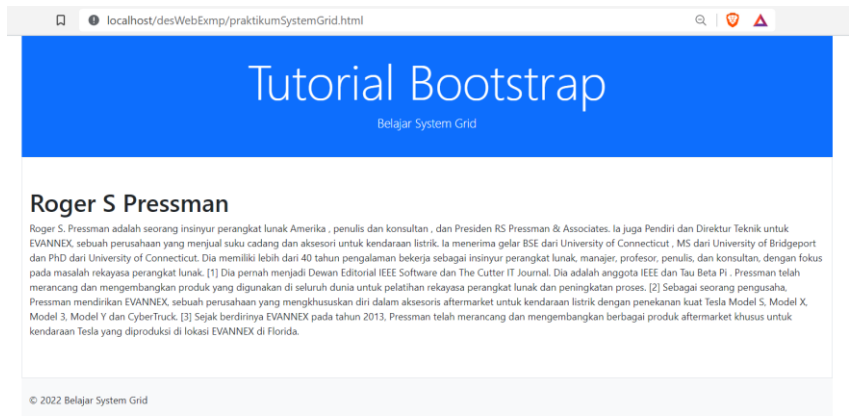
Sebagai seorang pengusaha, Pressman mendirikan EVANNEX, sebuah perusahaan yang mengkhususkan diri dalam aksesoris aftermarket untuk kendaraan listrik dengan penekanan kuat Tesla Model S, Model X, Model 3, Model Y dan CyberTruck. [3] Sejak berdirinya EVANNEX pada tahun 2013, Pressman telah merancang dan mengembangkan berbagai produk aftermarket khusus untuk kendaraan Tesla yang diproduksi di lokasi EVANNEX di Florida.

```
</p>
</div>
</div>
```

G) Pada tag footer tambahkan tag berikut dan amati perubahannya:

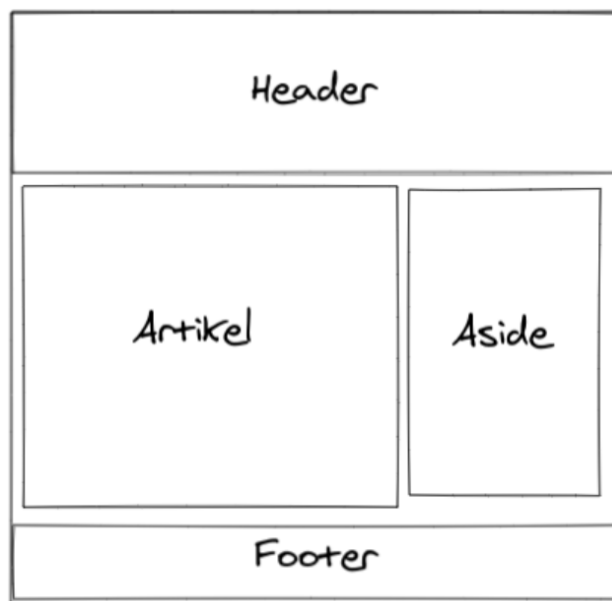
```
<div class="row">
  <div class="col-12 py-4">
    &copy; 2022 Belajar System Grid
  </div>
</div>
```

H) Jika hasil akhir dari halaman html anda tidak sama dengan tampilan di bawah ini, ulangi langkah - langkah diatas dengan lebih teliti.



Gambar 6.7 Hasil praktikum system grid

- I) Langkah berikutnya adalah kita tambahkan satu kolom pada main konten sehingga layout akan menjadi seperti ini:



Gambar 6.8 Layout praktikum

- J) Ganti isi pada tag main agar tampilan bisa terbagi menjadi dua kolom seperti layout diatas:

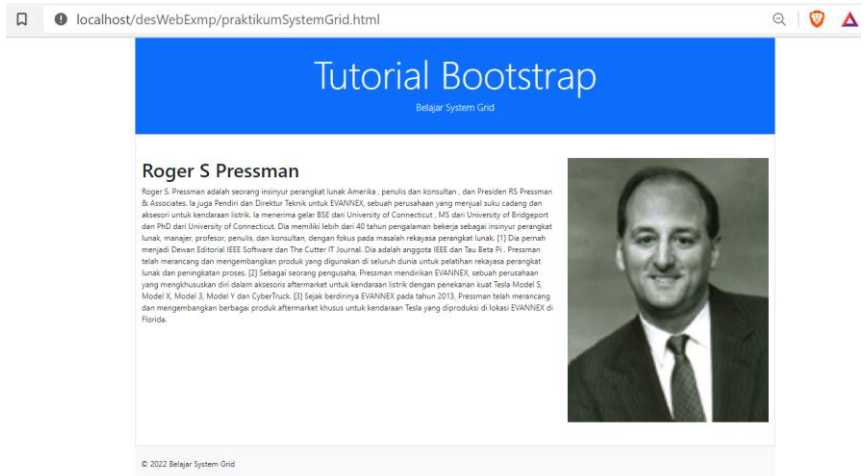
```
<div class="row">
  <div class="col-md-8 py-5">
    <h1>Roger S Pressman</h1>
    <p>
      Roger S. Pressman adalah seorang
      insinyur perangkat lunak Amerika , penulis dan
      konsultan , dan Presiden RS Pressman &
      Associates. Ia juga Pendiri dan Direktur Teknik
      untuk EVANNEX, sebuah perusahaan yang menjual
      suku cadang dan aksesoris untuk kendaraan
      listrik.

      Ia menerima gelar BSE dari
      University of Connecticut , MS dari University
      of Bridgeport dan PhD dari University of
      Connecticut. Dia memiliki lebih dari 40 tahun
      pengalaman bekerja sebagai insinyur perangkat
      lunak, manajer, profesor, penulis, dan
      konsultan, dengan fokus pada masalah rekayasa
      perangkat lunak. [1] Dia pernah menjadi Dewan
      Editorial IEEE Software dan The Cutter IT
      Journal. Dia adalah anggota IEEE dan Tau Beta
      Pi . Pressman telah merancang dan mengembangkan
      produk yang digunakan di seluruh dunia untuk
      pelatihan rekayasa perangkat lunak dan
      peningkatan proses. [2]

      Sebagai seorang pengusaha, Pressman
      mendirikan EVANNEX, sebuah perusahaan yang
      mengkhususkan diri dalam aksesoris aftermarket
      untuk kendaraan listrik dengan penekanan kuat
      Tesla Model S, Model X, Model 3, Model Y dan
      CyberTruck. [3] Sejak berdirinya EVANNEX pada
      tahun 2013, Pressman telah merancang dan
      mengembangkan berbagai produk aftermarket
      khusus untuk kendaraan Tesla yang diproduksi di
      lokasi EVANNEX di Florida.
    </p>
  </div>
  <div class="col-md-4 py-5">
    
  </div>
```


</div>

K) Berikut adalah hasil akhirnya



Gambar 6.9 Hasil akhir praktikum system grid

6.4 Membuat Fixed Layout Pada Bootstrap

Dengan Bootstrap kita masih dapat membuat tata letak halaman web berdasarkan jumlah pixel yang tetap, namun lebar wadah bervariasi tergantung pada lebar area dan tata letak dari halaman web tersebut.

Proses pembuatan tata letak tetap namun responsif pada dasarnya dimulai dengan `.container` class. Setelah itu kita dapat membuat baris dengan `.row` class untuk membungkus grup kolom horizontal. Baris harus ditempatkan di dalam sebuah class `.container` untuk perataan dan bantalan yang tepat.

Contoh berikut akan membuat tata letak responsif dengan lebar tetap yaitu lebar 720 pixel pada perangkat sedang seperti tablet (area pandang 768 pixel), sedangkan lebar 960 pixel pada perangkat besar seperti laptop kecil (area pandang 992 pixel), lebar 1140 pixel pada perangkat ekstra besar seperti desktop (viewport 1200px), dan lebar 1320px pada perangkat ekstra-ekstra besar seperti desktop besar (viewport 1400px).

Namun, pada perangkat kecil seperti ponsel (576px viewport < 768px) tata letak akan menjadi lebar 540px. Namun, pada perangkat ekstra kecil (area pandang < 576px) tata letak akan mencakup lebar 100%. Juga, kolom akan ditumpuk secara vertikal dan navbar akan diciutkan dalam kedua kasus tersebut.

A) Langkah pertama kita buat halaman HTML dengan nama **fixedLayout.html**, kemudian simpan halaman tersebut di dalam **htdocs** local server anda. Tambahkan tag sederhana seperti berikut ini:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport"
content="width=device-width, initial-scale=1">
    <title>Membuat Fixed Layout pada
Bootstrap</title>
    <link rel="stylesheet"
href="css/bootstrap.min.css">
    <script
src="js/bootstrap.bundle.min.js"></script>
  </head>
  <body>
    <!-- konten disini -->
```

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

- B) Membuat kerangka web yang ingin kita buat, mulai dari menyiapkan tempat untuk header, container dan juga footer. Langkah pertama adalah membuat header yang di dalamnya akan kita buat kumpulan tombol navigasi atau sering disebut dengan **navbar** (nama yang disematkan untuk mendefenisikan sistem navigasi yang dikumpulkan pada satu bagian pada tampilan website).

```
<nav>
    <!-- isi dari sistem navigasi disini -
->
</nav>
```

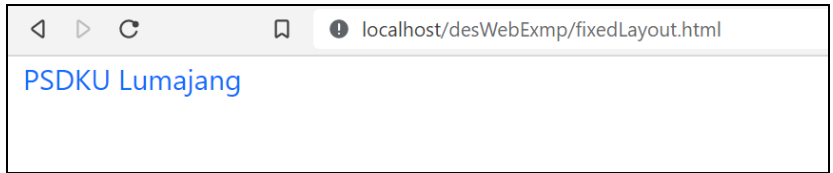
Selanjutnya kita akan tambahkan jenis layout pada navbar yaitu dengan menggunakan class `.container-fluid` yang akan kita panggil setelah `<nav>` dibuat.

```
<nav>
    <div class="container-fluid">
        <!-- sistem navigasi berada disini --
    >
    </div>
</nav>
```

Kemudian kita tambahkan satu navigasi dan kita lihat hasilnya :

```
<nav>
    <div class="container-fluid">
        <a href="#" class="navbar-
brand">PSDKU Lumajang</a>
    </div>
</nav>
```

Hasilnya :

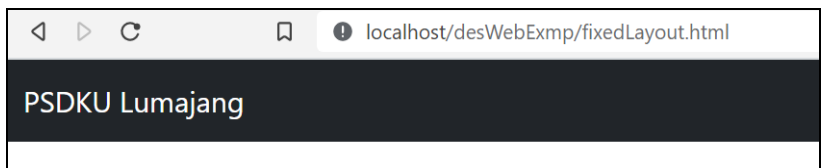


Gambar 6.10 Fixed Layout

- C) Langkah selanjutnya memanggil class untuk membuat tampilan navbar menjadi lebih bagus dan enak untuk dilihat:

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
```

Setelah itu kita lihat hasilnya:



Gambar 6.11 Desain navbar

- D) Setelah itu kita tambahkan navigasi / menu dengan model collapse yang akan kita buat setelah menu PSDKU Lumajang :

```
<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse"><span class="navbar-toggler-icon"></span></button>
```

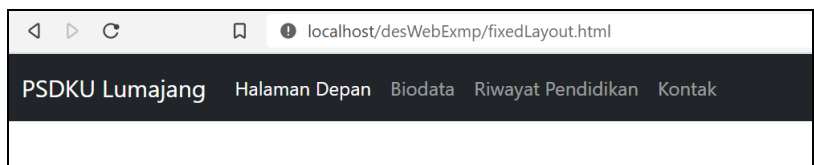
E) Kemudian kita tambahkan sistem menu seperti berikut ini (taruh tag berikut setelah button) :

```
<div class="collapse navbar-collapse"
id="navbarCollapse">
  <div class="navbar-nav">
    <!-- menu utama -->
  </div>
  <div class="navbar-nav ms-auto">
    <!-- menu pojok kanan -->
  </div>
</div>
```

F) Isi menu utama seperti berikut ini:

```
<!-- menu utama -->
<a href="#" class="nav-item nav-link
active">Halaman Depan</a>
<a href="#" class="nav-item nav-link">Biodata</a>
<a href="#" class="nav-item nav-link">Riwayat
Pendidikan</a>
<a href="#" class="nav-item nav-link">Kontak</a>
```

Hasilnya :

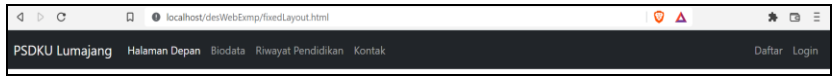


Gambar 6.12 Desain navbar dengan menu

G) Isi menu pada bagian pojok sebelah kanan seperti berikut:

```
<!-- menu pojok kanan -->
<a href="#" class="nav-item nav-link">Daftar</a>
<a href="#" class="nav-item nav-link">Login</a>
```

Hasilnya :



Gambar 6.13 Menamahkan register dan login pada navbar

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 breaksi saat ada perubahan data (reaktif).

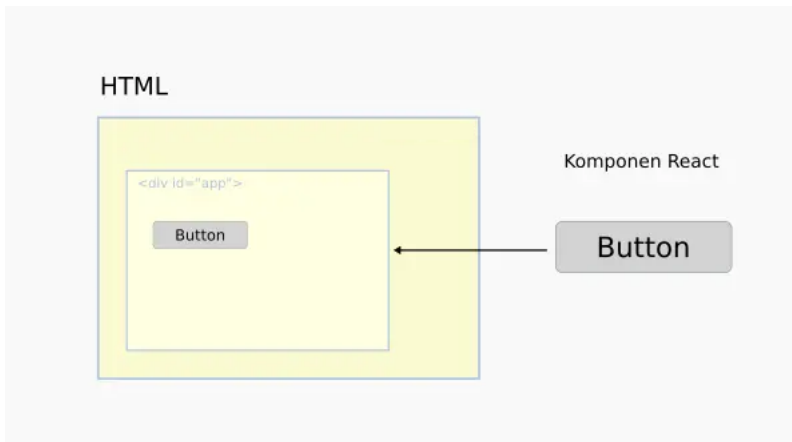
Lalu komponen itu apa?

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



Gambar 7.1 Struktur ReactJs

Apakah harus dibuat dengan id="app"?

Tidak harus, karena nanti kita akan tentukan elemennya dengan method `getElementById()`.

Berikutnya kita membuat kode aplikasinya dengan tipe text/babel:

```
<script type="text/babel">
  class Hello extends React.Component {
    render() {
      return <h1>Hello World Reactjs!</h1>
    }
  }

  ReactDOM.render(<Hello/>,
    document.getElementById("app"));
</script>
```

Pada kode tersebut, kita membuat komponen bernama Hello dari class `React.Component`.

Lalu me-rendernya ke dalam div.



Gambar 7.2 Komponen ReactJs

Mungkin kamu akan melihat sesuatu yang aneh di sini.

Kok bisa menulis HTML di dalam Javascript tanpa tanda petik?

Biasanya kan kita menulis seperti ini:

```
var elemenHTML = "<h1>Hello World!</h1>";
```

Tapi kok di React bisa begitu?

Ini adalah JSX.

8 Menggunakan JSX

JSX (Javascript XML) adalah extension dari Javascript. JSX membuat kita bisa menggunakan HTML di dalam Javascript.

JSX sama seperti XML dan HTML, ia juga memiliki nama tag, atribut, dan elemen anak.

Sebenarnya kita bisa saja menggunakan React tanpa JSX. Tapi tidak direkomendasikan, karena lebih susah dibaca dan ditulis.

Contoh:

```
// membuat komponen React tanpa JSX
class Btn extends React.Component {
  render() {
    let element = React.createElement('h1', {}, 'I
do not use JSX!');
    return element;
  }
}

// membuat komponen dengan JSX
class Btn2 extends React.Component {
  render() {
    let element = <h1>I Love JSX!</h1>;
    return element;
  }
}

// render komponen ke HTML
ReactDOM.render(<Btn/>,
document.getElementById("app"));
```

Tanpa JSX, kita harus membuat elemen dengan method createElement(). Bandingkan dengan yang menggunakan JSX, kita hanya perlu menulis sintaks XML atau HTML-nya saja.

Mana yang lebih gampang?

Tentu saja JSX, karena kita gak perlu repot-repot menghafal method-method React untuk membuat komponen.

Praktikum JSX

Langkah Praktikum

- 1) Buatlah file baru bernama latihan-jsx.html, kemudian isi dengan kode berikut.

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
  <title>Belajar Reactjs</title>
  <script
src="https://unpkg.com/react@16/umd/react.prod
uction.min.js"></script>
  <script src="https://unpkg.com/react-
dom@16/umd/react-
dom.production.min.js"></script>
  <script src="https://unpkg.com/babel-
standalone@6.15.0/babel.min.js"></script>
</head>

<body>

  <div id="app"></div>

  <script type="text/babel">
    // membuat elemen JSX
    let message = <h2>Belajar menggunakan
JSX pada React</h2>;

    // render elemen ke DOM
    ReactDOM.render(message,
document.getElementById("app"));
  </script>

</body>

</html>
```

- 2) Berikut adalah hasilnya:



Pada contoh ini, kita membuat elemen HTML dengan JSX, kemudian langsung me-rendernya tanpa harus membuat class komponen.

Latihan berikutnya, cobalah untuk membuat JSX yang memiliki anak dan atribut.

Praktikum JSX kedua

Langkah praktikum

- 1) Buatlah file baru bernama latihan-jsx2.html, kemudian isi dengan kode berikut:

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
  <title>Belajar Reactjs</title>
  <script
src="https://unpkg.com/react@16/umd/react.prod
uction.min.js"></script>
  <script src="https://unpkg.com/react-
```

```

dom@16/umd/react-
dom.production.min.js"></script>
<script src="https://unpkg.com/babel-
standalone@6.15.0/babel.min.js"></script>
</head>

<style>
  .header {
    background-color: blueviolet;
    color: white;
    padding: 10px;
    font-family: Lato, sans-serif;
  }
</style>

<body>

  <div id="app"></div>

  <script type="text/babel">
    let header = (
      <header className="header">
        <h1>Belajar menggunakan JSX pada
React</h1>
        <p>Tutorial Reactjs untuk
Pemula</p>
      </header>
    );

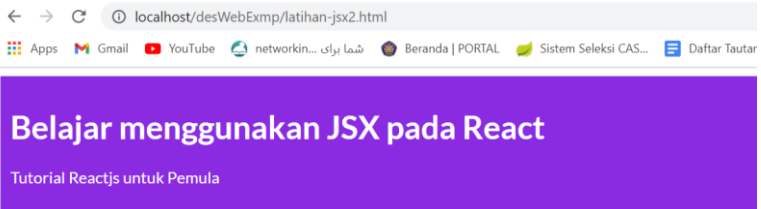
    ReactDOM.render(header,
document.getElementById("app"));
  </script>

</body>

</html>

```

2) Hasilnya :



Gambar 8.2 Hasil praktikum JSX

- 3) Coba perhatikan kodenya!
- 4) Saat kita membuat elemen JSX yang memiliki anak, kita harus mengapitnya dengan tanda kurung.

```
let header = (  
  <header className="header">  
    <h1>Belajar menggunakan JSX pada  
React</h1>  
    <p>Tutorial Reactjs untuk  
Pemula</p>  
  </header>  
);
```

9 Struktur Direktori Project pada ReactJS

Pada pembahasan sebelumnya, kita sudah belajar dasar Reactjs. Namun, di sana kita hanya sebatas menggunakan Reactjs sebagai library dengan menyisipkan langsung ke HTML melalui CDN. Ini tidak salah. Hanya saja saat nanti kita membuat aplikasi besar, kita akan menemukan masalah. Yaitu:

Kesulitan mengelola source code, karena banyaknya kode. Mungkin anda bisa bayangkan, betapa berantakannya kode tersebut. Oleh karena itu kita harus membuat project Reactjs.

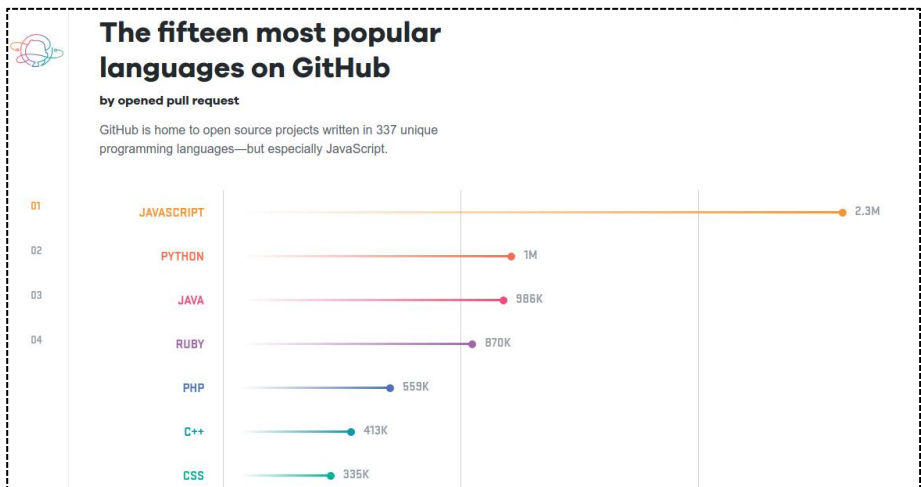
Langkah awal dari membuat project pada react JS adalah melakukan langkah-langkah persiapan seperti berikut ini:

1. Melakukan instalasi Nodejs

2. Mengetahui NPM dan NPX (akan dibahas pada materi berikutnya)

9.1 Mengetahui Nodejs

Sejak kehadirannya, Nodejs banyak membawa perubahan terhadap dunia pemrograman, khususnya web. Dengan Nodejs, kita bisa membuat aplikasi Web, Desktop, Mobile, bahkan IoT (Internet of Things). Dan hebatnya lagi, Javascript menjadi bahasa yang paling populer di github saat ini.



Gambar 9.1 Kepopuleran Javascript

Awalnya Javascript hanya digunakan untuk membuat web saja. Namun sekarang, berkat Nodejs... kita bisa membuat aplikasi desktop, web, mobile, CLI, IoT, dll.

Apakah Nodejs itu bahasa pemrograman baru?

Tidak, Nodejs bukan bahasa pemrograman.

Untuk lebih jelasnya, ada baiknya kita menyimak sejarahnya.

Pada tahun 2009, seorang programmer bernama Ryan Dahl menciptakan Nodejs. Bahasa Pemrograman Javascript awalnya hanya bisa berjalan di atas browser, karena ada runtime engine di dalamnya. Kemudian, munculah ide:

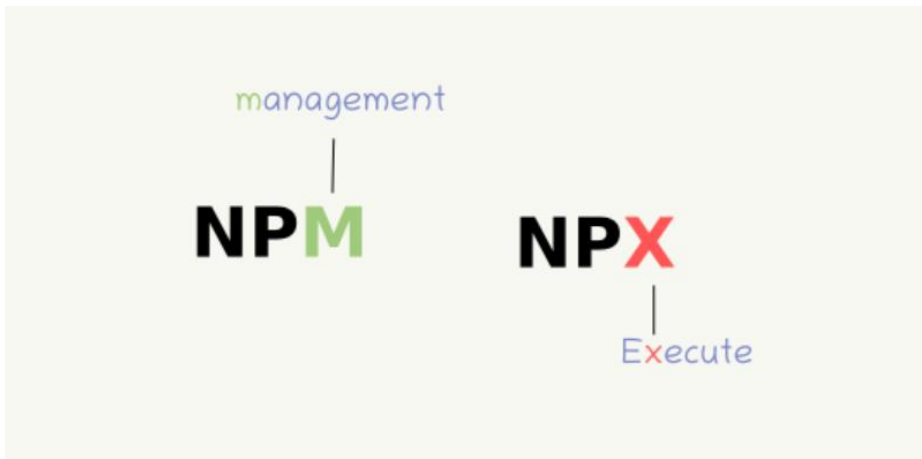
“Bagaimana kalau engine yang ada di dalam browser, kita keluarkan agar kita bisa mengeksekusi Javasript di luar browser”

Maka lahirlah Nodejs.

Nodejs adalah sebuah platform untuk mengeksekusi program Javascript di luar browser. Nodejs menggunakan runtime engine bernama V8, yang merupakan javascript runtime engine dari Google Chrome.

10 Mengenal NPM dan NPX

NPM adalah singkatan dari Node Package Manager. Sebuah program berbasis teks untuk manajemen paket Nodejs. Sedangkan NPX adalah Node Package Runner. Fungsinya untuk mengeksekusi package Nodejs.



Gambar 10.1 perbedaan NPM dan NPX

NPX akan mengeksekusi file binary dari package Nodejs, baik yang sudah terinstal maupun yang belum.

Instalasi NPX

NPX mulai ditambahkan pada NPM versi 5.2.0. Jadi, apabila kamu menggunakan NPM versi ini, maka tidak perlu menginstal npx. Tapi kalau di komputermu belum terinstal, kamu bisa menginstalnya dengan perintah:

```
[sudo] npm install -g npx
```

Gunakan sudo apabila kamu menginstal Nodejs pada root direktori.

Cara Menggunakan NPX

Cara menggunakan NPX hampir sama seperti NPM.

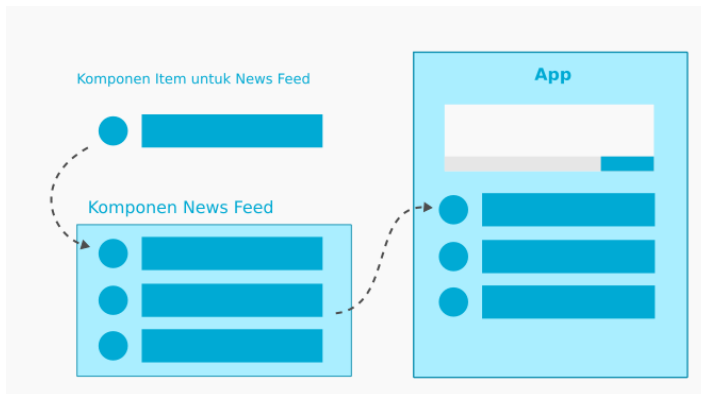


Gambar 10.2 Elemen pada NPM

11 Komponen pada ReactJS

Komponen adalah bagian terpenting dari aplikasi React, karena aplikasi react sendiri terusun dari komponen-komponen. Komponen di Reactjs ada beberapa macam, Ada yang disebut dengan stateful component dan stateless component. Ada juga function components dan class components. Ada Dumb Components dan Smart Components.

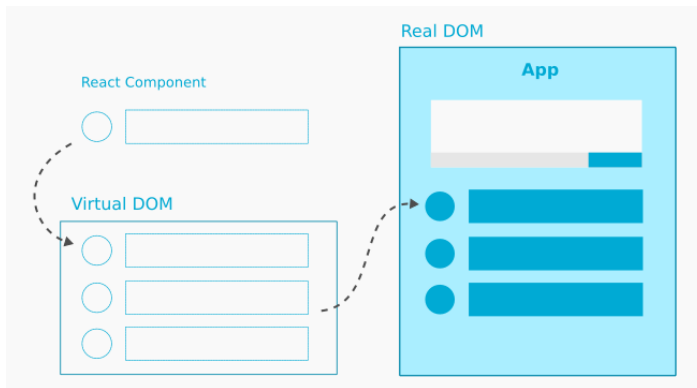
Komponen adalah bagian-bagian yang menyusun aplikasi React, berikut adalah bagan yang bisa anda buat belajar untuk lebih memahami reactJS



Gambar 11.1 Komponen pada ReactJs

Mengapa React membutuhkan komponen?

Reactjs bekerja dengan Virtual DOM. Nah, di dalam Virtual DOM ini kita harus membuat komponen untuk memberitahu React tentang apa saja yang harus ditampilkan (render) ke Real DOM (HTML). Karena itu, setiap komponen di React pasti akan menghasilkan HTML. Coba saja perhatikan di method `render()`, di sana pasti akan ada kode JSX yang akan di-render menjadi HTML.



Gambar 11.2 Method Render

Ini mungkin terdengar 'ribet' bagi anda yang sudah terbiasa bekerja dengan framework lama seperti JQuery. Karena ketika kita mengimplementasikan reactJS kita harus selalu membuat komponen ketika akan membuat sesuatu yang baru dan tidak seperti JQuery yang bisa langsung memanipulasi DOMnya.

Komponen di React sendiri bersifat reusable, artinya bisa digunakan kembali. Di luar sana ada banyak komponen React yang siap pakai dan akan memudahkan hidup kita dalam membuat aplikasi.

Lalu apa perbedaan antara komponen dengan elemen?

Elemen adalah bagian-bagian (elemen HTML) yang menyusun komponen. Biasanya kita buat dengan JSX. Secara umum, ada tiga bagian yang biasanya ada di dalam komponen:

- 1) Bagian Data (State, Props, Variabel)
- 2) Bagian method atau fungsi

- 3) Bagaimana method `render()` (wajib ada jika menggunakan class).

12 Javascript

Pada dasarnya, pengertian JavaScript adalah suatu bahasa kode atau pemrograman yang digunakan untuk menciptakan sekaligus mengendalikan konten website agar menjadi dinamis. Contoh konten situs yang dinamis adalah apa pun yang dapat bergerak atau mengubah apa pun yang tampak di layar tanpa mengharuskan Anda memuat ulang laman situs tersebut secara manual.

Beberapa fitur yang dapat membuat situs menjadi dinamis (dan tentunya membutuhkan bahasa pemrograman) di antaranya adalah gambar animasi, slideshow foto, saran pengisian teks otomatis, form otomatis, dan banyak lagi. Jadi, di balik semua animasi interaktif dan form otomatis pada di suatu situs, ada kumpulan rumus bahasa pemrograman seperti JavaScript.

12.1 Dasar-Dasar JavaScript

Setelah memahami pengertian JavaScript, maka selanjutnya mari membahas lebih jauh dan terperinci tentang salah satu bahasa pemrograman yang penting dalam website development ini. Untuk lebih mendalami JavaScript, maka pertama Anda harus berkenalan terlebih dahulu dengan tiga poin

penting dalam pemrograman, yaitu HTML, CSS, dan yang terakhir baru JavaScript.

HTML adalah struktur halaman yang terdiri dari header, teks badan, serta gambar yang Anda sertakan pada laman situs. CSS mengendalikan tampilan laman dan inilah yang digunakan untuk merancang jenis huruf, warna latar belakang, dan lain-lain. Setelah struktur (HTML) dan estetika (CSS) sudah selesai dibuat, maka JavaScript adalah yang akan membuat laman situs menjadi dinamis.

Dalam versi pengertian JavaScript yang paling sederhana, bahasa pemrograman ini memerintahkan gambar agar bergerak, foto agar bergeser, atau membuat form terisi secara otomatis dalam sebuah situs. Inilah mengapa JavaScript merupakan bagian penting dari sebuah website agar dapat berfungsi dengan baik dan interaktif.

12.2 Fungsi JavaScript

Dari pembahasan tentang pengertian JavaScript di atas, Anda mungkin sudah memiliki gambaran umum tentang fungsi bahasa pemrograman ini. Berikut ini daftar singkat beberapa fungsi JavaScript.

- Membuat situs tampak lebih interaktif dan menarik karena akan lebih dari sekadar laman statis yang penuh tulisan.

- Berfungsi untuk mobile app development atau menciptakan aplikasi yang beroperasi di smartphone dan tablet.
- Menciptakan game berbasis web browser.
- Meskipun JavaScript umumnya digunakan untuk menciptakan bagian frontend pada sebuah situs, bahasa pemrograman ini sangat serbaguna sehingga bisa digunakan untuk bagian backend situs.

13 Implementasi Javascript

JavaScript bisa terletak di dalam laman suatu situs atau berada pada file .js yang terpisah. Mengingat JavaScript merupakan bahasa sisi klien, maka script atau naskah tersebut akan diunduh terlebih dahulu di perangkat pengunjung situs untuk diproses di sana.

Pemrosesan JavaScript dilakukan oleh pengunjung situs dengan mengirim permintaan yang selanjutnya diproses di server. Selanjutnya, peramban web mengirim hasil pemrosesan tersebut ke pengunjung situs.

Bagi mereka yang masih awam, proses ini memang terlihat rumit. Meski demikian, menambahkan kode JavaScript ke suatu laman situs sebenarnya merupakan proses yang sederhana. Bahkan, proses ini juga akan terasa lebih familier dan simpel jika Anda sudah pernah bekerja menggunakan HTML dan CSS.

JavaScript bisa langsung ditambahkan pada kode laman dengan menggunakan tag `<script>` ditambah dengan teks atribut JavaScript. Berikut ini contoh kode teks JavaScript:

```
<script type = "text/javascript"> masukkan kode JavaScript di sini  
</script>
```

Jika Anda ingin perintah JavaScript tersebut berlaku pada beberapa halaman sekaligus, maka kode tersebut bisa ditambahkan pada suatu laman sebagai file header terpisah dengan ekstensi `.js`. Namun, perlu diingat bahwa JavaScript diproses oleh peramban masing-masing pengunjung. Jadi, jika peramban milik pengunjung tidak mendukung JavaScript, maka tampilan laman pada layar pengunjung tersebut akan berbeda dan fungsi interaktif yang menggunakan JavaScript tidak akan bekerja.

13.1 Vanilla JavaScript vs. Advanced JavaScript

Makin Anda mempelajari tentang JavaScript lebih jauh, Anda akan menemukan lebih banyak macam bahasa pemrograman JavaScript. Salah satunya adalah vanilla JavaScript. Ini adalah bahasa JavaScript paling sederhana dan tanpa harus menggunakan tool atau alat apa pun untuk membuat proses pengkodean lebih mudah dan efisien. Anda juga bisa menambahkan kode tersebut langsung pada laman HTML.

Vanilla JavaScript seperti ini bisa digunakan untuk menciptakan berbagai proyek JavaScript. Meski demikian, tentu akan memakan waktu lebih lama dan tidak efisien untuk membuatnya.

Namun, Anda tidak perlu khawatir karena ada berbagai macam tool untuk membuat JavaScript bisa digunakan dengan lebih mudah dan efisien. Berikut ini beberapa kerangka kerja dan library JavaScript yang akan memudahkan Anda dalam menggunakan JavaScript untuk web dan software development.

13.2JQuery

Semakin sering bekerja menggunakan JavaScript, Anda akan menyadari ada beberapa fungsi dan fitur JavaScript yang ternyata sering muncul dan digunakan di berbagai situs atau aplikasi. Misalnya, menu animasi, fade out, galeri gambar, dan lain-lain. Anda memang bisa menulis secara manual kode ini setiap kali membutuhkannya. Namun, dengan adanya library kode seperti JQuery, Anda tidak perlu lagi menulis kode tersebut secara manual dari awal.

JQuery berisi berbagai fungsi kode yang dapat dibuat hanya dengan menuliskan satu baris perintah JQuery saja. Pendekatan pemrograman ini jelas jauh lebih ringkas dan dapat digunakan kembali kapan pun dibutuhkan. Selain bisa digunakan sebagai cuplikan kode yang bisa disisipkan langsung dari JQuery

library, kode JQuery juga bisa digabungkan untuk membuat plugin yang lebih rumit.

Plugin JQuery bisa ditemukan langsung di repositori JQuery UI (User Interface) dan Anda bisa langsung menyalin dan menggunakannya.

13.3 React JS

Pustaka JavaScript selanjutnya adalah ReactJS yang cocok untuk membangun frontend sebuah situs. ReactJS merupakan pustaka JavaScript frontend yang dikembangkan oleh Facebook pada 2011, serta dirancang khusus untuk membangun UI atau user interface. Yang dimaksud dengan UI di sini adalah kumpulan menu pada layar, kolom pencarian, tombol-tombol, dan berbagai fungsi lain yang digunakan pengguna untuk berinteraksi dengan situs tersebut.

Tentu saja UI bisa dibuat secara manual menggunakan vanilla JavaScript, tetapi bisa Anda bayangkan betapa merepotkan dan tidak efisien cara tersebut. Dengan pustaka ReactJS ini, para web developer bisa langsung menggunakan kode yang sudah tersedia dan tertulis sebelumnya untuk membuat objek menu dan efek. (Detail materi reactJs lebih detail ada di materi sebelumnya).

13.4 Tipe Data JavaScript

JavaScript memiliki berbagai tipe data yang digunakan saat menuliskan variabel. Berikut ini beberapa di antaranya:

- String, yaitu variabel dengan rangkaian karakter yang membentuk teks.
- Number, yaitu untuk variabel angka.
- Boolean, yaitu untuk variabel yang memberikan nilai benar atau salah (true/false).
- Array, yaitu untuk variabel yang memiliki lebih dari satu nilai dalam satu variabel.
- Object, yaitu tipe data yang dapat digunakan untuk menyimpan semua bentuk dalam JavaScript.
- Function, yaitu potongan kode yang dapat dikumpulkan dan digunakan kembali.

13.5 Cara Menulis Menggunakan Kode JavaScript

JavaScript memang terdiri dari rangkaian rumus dan kata-kata yang jika Anda sudah memahami artinya, akan cukup sederhana dan mudah digunakan. Meski demikian, JavaScript memiliki peraturan penulisan tertentu yang harus diikuti agar kode yang Anda tulis bisa berfungsi.

Berikut ini beberapa contoh peraturan dasar penulisan kode JavaScript yang harus diketahui.

- JavaScript sangatlah sensitif, jadi pastikan kode ditulis dengan benar dan tidak ada kesalahan satu karakter pun.
- Setiap statement atau pernyataan harus diakhiri dengan titik koma (;).

- Variabel harus terdefinisi terlebih dahulu sebelum digunakan dan harus selalu diawali dengan sebuah huruf atau garis bawah ("_").
- Tentukan jenis data yang dimasukkan ke dalam variabel, tetapi tidak harus secara eksplisit disebutkan.
- String harus ditulis di antara tanda petik ganda ataupun tunggal.
- Jika ingin menampilkan karakter khusus, maka harus diawali dengan tanda backslash (\).
- Jika ingin memasukkan komentar dalam naskah, gunakan `"//"` untuk memulai komentar satu baris, atau gunakan `"/*"` atau `"*/"` jika ingin menulis komentar yang terdiri dari lebih dari satu baris.
- Pisahkan laman situs Anda menjadi beberapa komponen terkecil yang independen dan dapat beroperasi sendiri.
- Dengan masing-masing laman situs yang independen dan dapat bekerja sendiri, Anda tidak perlu menggunakan variabel global. Hal ini membuat aplikasi atau situs Anda bekerja dengan lebih cepat. Debugging pun lebih mudah.
- Pastikan kode Anda sederhana dan tidak ada duplikasi kode. Hal ini akan memudahkan Anda jika suatu saat harus melakukan debugging karena akan lebih mudah menelusuri kembali kode yang sudah lama Anda tulis.

13.6Praktikum

Praktikum Statement

Langkah Praktikum:

- 1) Buatlah dokumen html dengan nama statementjs.html dan simpan di dalam folder dewWebExmp.
- 2) Tambahkan kode seperti dibawah ini:

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

<h2>JavaScript Statements</h2>

<p>JavaScript akan diseeksekusi oleh
browser.</p>

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

<script>
    document.getElementById("demo").innerHTML
= "Hello JavaScript.";
</script>

</body>
</html>
```

- 3) Amati dan catat hasilnya

Praktikum Statemen 2

Langkah praktikum:

- 1) Buatlah dokumen html dengan nama statementjs2.html dan simpan dalam folder desWebExmp
- 2) Tambahkan koding seperti dibawah ini:

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

<h2>JavaScript Statements</h2>

<p>JavaScript statements dipisah oleh tanda
semicolons.</p>

<p id="demo1"></p>

<script>
let a, b, c;
a = 5;
b = 6;
c = a + b;
document.getElementById("demo1").innerHTML =
c;
</script>

</body>
</html>
```

3) Amati dan catat hasilnya