

Anatomi CSS

Code

```
P {  
Color: red;  
}
```

Penjelasan

- Tag `<p>` adalah selector yang ingin di modifikasi.
- property adalah `color` pada komponen textnya
- property values adalah `red` kita mau modifikasi seperti apa warnanya

Kesimpulan

warna teks (text color) dari elemen HTML yang memiliki tag P akan diubah menjadi merah (red).

Percobaan pertama

Code

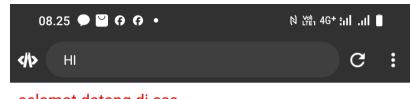
```
P {  
Color: red;  
}
```

Before



selamat datang di css

After



selamat datang di css

Penjelasan

mengatur warna teks (text color) dari elemen HTML yang memiliki tag P menjadi merah (red).

Kesimpulan

warna teks (text color) dari elemen HTML yang memiliki tag P akan diubah menjadi merah (red).

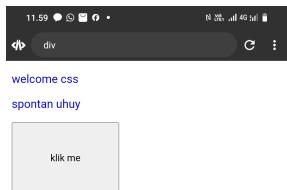
Percobaan kedua

code CSS

```
button {  
    width:150px;  
    height:100px;  
    background-color: #001A54;  
    color: green;  
    font-size: 50px;  
}
```

background-color

Before



After



Penjelasan

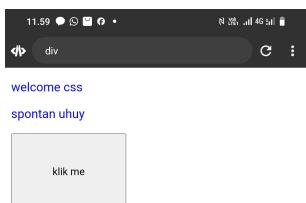
background-color: #001A54; : Menetapkan warna latar belakang tombol menjadi biru gelap dengan kode warna heksadesimal `#001A54` .

Kesimpulan

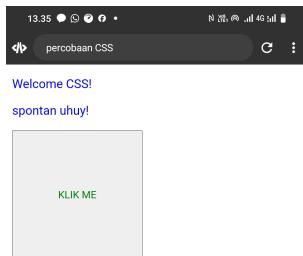
background-color dari elemen HTML akan diubah menjadi warna #001A54 .

color

Before



After



Penjelasan

mengatur warna teks (text color) menjadi hijau (green).

Kesimpulan

Warna text akan di ubah menjadi hijau

Font-size

Before



Welcome CSS!

spontan uhuy!

KLIK ME



After



Welcome CSS!

spontan uhuy!

KLIK
ME



Penjelasan

mengatur ukuran font menjadi 50 pixel.

kesimpulan

Ukuran font akan di ubah menjadi 50pixel

deklarasi Pemanggilan Css

External

External adalah pemanggilan dengan cara membuat file baru dengan format filenya nama_file.css Berbeda dengan internal dan inline yang cukup dengan 1 file saja.

Contoh

File Html

```
<!doctype html>
<html>
  <head>
    <title>web saya</title>
    <link rel="stylesheet" href="style.css"
  </head>
  <body>
    <p>Hallo RPL</p>
  </body>
</html>
```

File CSS

```
p {
  color: black;
}
```

Inline

Penjelasan

Tag `<p>` yang memiliki gaya inline yang diterapkan langsung ke dalam elemen tersebut. Gaya tersebut adalah `font-size: 48px;`, yang berarti teks di dalam paragraf akan ditampilkan dengan ukuran font 48 piksel. Gaya ini diterapkan menggunakan atribut style pada elemen `<p>`.

code

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

```
<body>
  <p style="font-size: 48px;">Welcome CSS!</p>

</body>

</html>
```

Internal

Internal adalah pemanggilan css secara satu file/ berada didalam file html yg ingin kita berikan style. Adapun tempat penulisan CSS secara internal, kita dapat tuliskan di dalam `<head>`

Contoh

```
<!doctype html>
<html>
  <head>
    <title>web saya</title>

    <style>
      p {
        color: orange;
      }
    </style>

  </head>
  <body>
    <p>cihuy</p>
  </body>
</html>
```

Selector

Elemen Selector

Elemen selector adalah jenis selector dengan cara pemanggilan langsung menyebut tag htmlnya. Contoh jika menggunakan tag `<p>` itu akan memilih semua elemen paragraf pada html.

Class Selector

Digunakan untuk memberikan gaya pada elemen HTML yang memiliki kelas tertentu. Kelas ini didefinisikan di dalam tag HTML dengan atribut class. Satu elemen dapat memiliki banyak

kelas, dan kelas dapat digunakan oleh beberapa elemen.

Id Selector

ID dalam CSS merujuk pada identifikasi unik dari sebuah elemen HTML. Setiap elemen dapat diberi ID yang berbeda, dan ID tersebut harus unik di dalam satu halaman web. ID digunakan untuk memberikan gaya atau memanipulasi elemen secara spesifik.

Text

Text align

Penjelasan

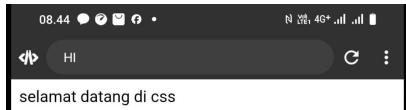
- `text-align:center;` mengatur alignment teks (text alignment) menjadi tengah (center).
- `text-align:left;` mengatur alignment teks (text alignment) menjadi kiri (left).
- `text-align:right` mengatur alignment teks (text alignment) menjadi kanan (right).
- `text-align:justify;` Kode di bawah ini mengatur alignment teks (text alignment) menjadi justifikasi (justify).

Kode Program

```
p{  
    text-align: center;  
}
```

Hasil

before



after



Kesimpulan

Jika ingin mengatur tata letak sebuah Text dalam website, kita bisa gunakan Property `text-align`

Text decoration

Penjelasan

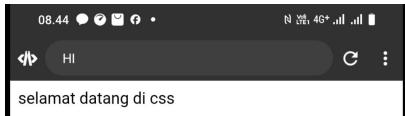
- `text-decoration: overline;` digunakan untuk menambahkan garis atas pada teks.
- `text-decoration: underline;` digunakan untuk menambahkan garis bawah pada teks.
- `text-decoration: line-through;` digunakan untuk menambahkan garis melalui pada teks.
- `text-decoration: none;` digunakan untuk menghapus dekorasi pada teks.

Kode Program

```
p{  
    text-decoration: overline;  
}
```

Hasil

Before



After



Kesimpulan

Text-decoration adalah fitur CSS yang digunakan untuk menambahkan dekorasi pada teks, seperti garis bawah, garis atas, dan garis melalui.

Text transform

Penjelasan

- `none` (default): Menjaga kapitalisasi teks seperti pada penulisan aslinya.
- `uppercase`: Mengubah semua huruf menjadi huruf besar (uppercase).
- `lowercase`: Mengubah semua huruf menjadi huruf kecil (lowercase).

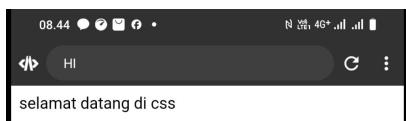
- `capitalize` : Mengubah huruf pertama setiap kata menjadi huruf besar (capitalize), mirip gaya penulisan judul.

Kode Program

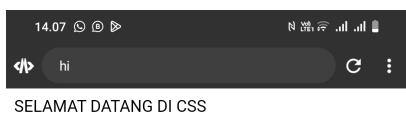
```
p {  
    text-transform: uppercase;  
}
```

Hasil

Before



After



Kesimpulan

Text-transform adalah fitur CSS yang digunakan untuk mengubah tampilan teks, seperti huruf besar, huruf kecil, dan huruf kapital.

Text indent

PENJELASAN

Text indent atau indentasi teks adalah proses membuat jarak antara teks dengan margin atas dari sebuah paragraf. Ini dapat membuat teks terlihat lebih rapi dan terstruktur. Teks dapat diatur dengan menggunakan tanda indentasi seperti tab atau spasi.

KODE PROGRAM

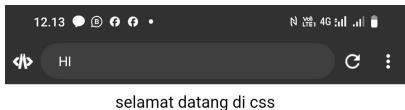
```
p {  
    text-indent: 100px;  
}
```

HASIL

BEFORE



AFTER



ANALISIS

- `text-indent: 100px;` : Properti ini mengatur indentasi baris pertama teks dalam elemen paragraf. Dalam hal ini, disetel ke 100 piksel, yang berarti baris pertama teks akan diindentasi 100 piksel dari margin kiri.

Letter spacing

PENJELASAN

Letter spacing merujuk pada jarak horizontal antara huruf-huruf dalam sebuah teks. kata lain, letter spacing mengontrol seberapa dekat atau berjauhan huruf-huruf tersebut ditempatkan satu sama lain.

KODE PROGRAM

```
p {  
letter-spacing: 50px;  
}
```

HASIL

BEFORE



AFTER



ANALISIS

- `p {` : Bagian ini mendefinisikan aturan gaya yang berlaku untuk semua `<p>` elemen dalam dokumen.
- `letter-spacing: 50px;` : Ini menetapkan `letter-spacing` properti untuk elemen yang ditargetkan.
- `letter-spacing` : Properti ini mengontrol jumlah spasi yang ditambahkan antar karakter dalam teks.
- `50px` : Nilai ini menentukan jumlah ruang yang akan ditambahkan, dalam hal ini, 50 piksel

Line height

Kesimpulan

Line height adalah properti dalam CSS yang menentukan **jarak vertikal** antara baris teks dalam sebuah elemen, biasanya elemen paragraf (`<p>`).

KODE PROGRAM

```
p {  
  line-height: 150px;  
}
```

HASIL

BEFORE

13.15 ⚡ 🗣️ 📸 🎵 •

N VoLTE 4G ⚡ ⚡ ⚡



font



Saya berada di kelas 11

AFTER



font



Saya berada di kelas 11

Penjelasan

- `p` : Pemilih ini menargetkan semua elemen tag `<p>` , yang biasa digunakan untuk paragraf dalam HTML.
- `line-height` : menentukan **tinggi kotak garis** dalam elemen paragraf. Kotak garis adalah kotak tak terlihat yang mengelilingi setiap baris teks, termasuk spasi di atas dan di bawah teks itu sendiri.
- `150px` : Nilai ini menetapkan tinggi garis menjadi **150 piksel** . Artinya jarak vertikal antara garis dasar garis berurutan dalam sebuah paragraf adalah 150 piksel.

Word spacing

PENJELASAN

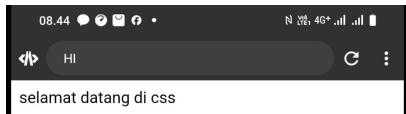
Word spacing merujuk pada pengaturan jarak antara kata dalam teks.

KODE PROGRAM

```
p {  
word-spacing: 150px;  
}
```

HASIL

BEFORE



AFTER



Kesimpulan

Program tersebut menggunakan properti CSS `word-spacing` pada elemen paragraf (`<p>`) dengan nilai `15px`. Properti ini mengatur jarak antar kata dalam teks pada paragraf tersebut. Dengan nilai `word-spacing: 15px;`, program menciptakan efek peningkatan jarak antar kata sebanyak 15 piksel. Sehingga, kata-kata dalam teks pada paragraf akan memiliki ruang lebih besar antar satu sama lain.

Background

Background-color

Penjelasan

Background-color adalah fitur CSS yang digunakan untuk mengatur warna latar belakang dari elemen HTML.

Kode Program

```
p {  
background-color: blue;
```

```
}
```

Hasil

Before



After



Kesimpulan

`Background-color` adalah kode buat kita menambahkan warna latar belakang dari elemen html

Background-image

Penjelasan

Background-image dalam CSS adalah properti yang digunakan untuk menentukan gambar yang akan digunakan sebagai latar belakang dari suatu elemen.

Kode Program

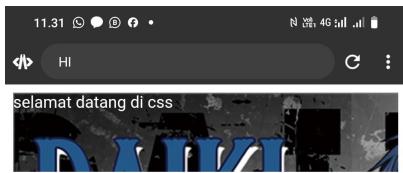
```
body {  
    background-image: url("fatirr.jpg");  
}
```

Hasil

Before



After



Kesimpulan

Background-image digunakan untuk mengatur gambar yang akan digunakan sebagai latar belakang dari suatu elemen.

Background-repeat

Penjelasan

Background-repeat adalah untuk mengulangi gambar latar

Kode Program

```
p {  
background-image: url("fatirr.jpg");
```

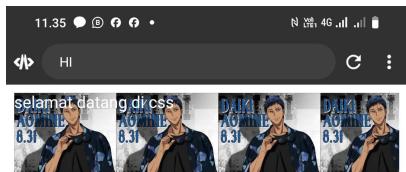
```
background-repeat: repeat;  
}
```

Hasil

Before



After



Kesimpulan

Mengulangi gambar latar belakang

Background-attachment

Penjelasan

background-attachment dalam CSS adalah properti yang digunakan untuk mengontrol apakah gambar latar (background image) akan tetap diam atau bergulir bersamaan dengan konten saat pengguna meng gulir halaman. Properti ini memiliki dua nilai utama:

- `scroll` : Gambar latar belakang akan bergulir bersamaan dengan konten saat halaman digulir.
- `fixed` : Gambar latar belakang akan tetap diam relatif terhadap tampilan, sehingga saat halaman digulir, gambar tersebut tetap pada posisi yang sama di layar.

Kode Program

```
<html>
  <head>
    <title>css</title>
    <style>
      p {
        background-image: url('2.jpg');
        height: 100vh;
        background-size:100px;
        background-attachment: fixed;
      }
    </style>
  </head>
  <body>
    <p>selamat datang</p>

    <p>selamat datang</p>

  </body>
</html>
```

Hasil

19.33



N Vol LTE 1 4G+ . . .



CSS





Kesimpulan

background-attachment dalam CSS adalah properti yang mengontrol apakah gambar latar belakang akan tetap diam atau bergulir bersamaan dengan konten saat pengguna meng gulir halaman.

Background-size

Penjelasan

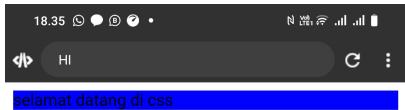
Kode `background-size: 50px;` dalam CSS digunakan untuk menentukan ukuran background image pada elemen HTML.

Kode Program

```
body {  
    background-image: url("fatirr.jpg");  
    background-size: 50px  
}
```

Hasil

Before



After



Kesimpulan

Background-size untuk mengatur ukuran background image baik panjang maupun lebarnya

Background-position

Penjelasan

Background-position di gunakan untuk mengatur posisi gambar

Kode Program

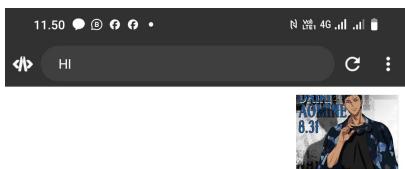
```
p {  
background-image: url("fatirr.jpg");  
background-position: right;  
}
```

Hasil

Before



After



Kesimpulan

Untuk mengatur posisi gambar

Font

Font-size

Penjelasan

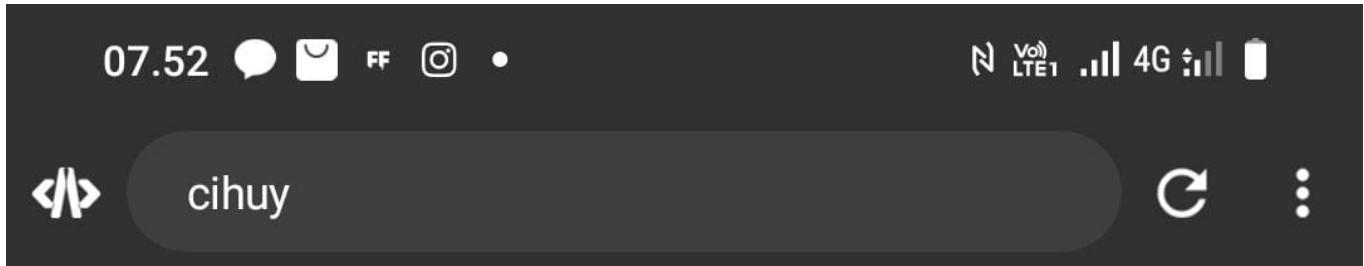
Kode `font-size` di gunakan untuk mengatur besar kecilnya Font yang di gunakan di css

Kode Program

```
p{  
font-size: 48px;  
}
```

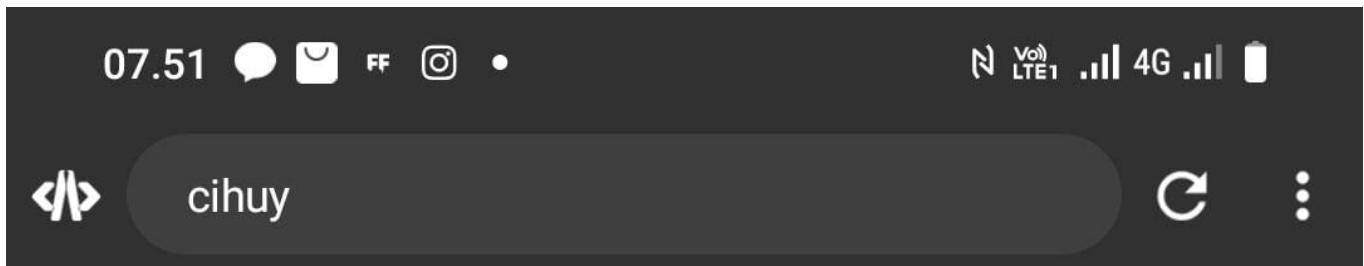
Hasil

Before



Anjay

After



Anjay

Kesimpulan

Font-size di gunakan untuk mengatur besar kecilnya text

Font-style

Penjelasan

Font style Property CSS yang digunakan untuk menentukan gaya penulisan Text. Beberapa contoh property value nya :

- `normal` menampilkan Text dengan gaya standar default .
- `italic` menampilkan Text dengan gaya miring.

Kode Program

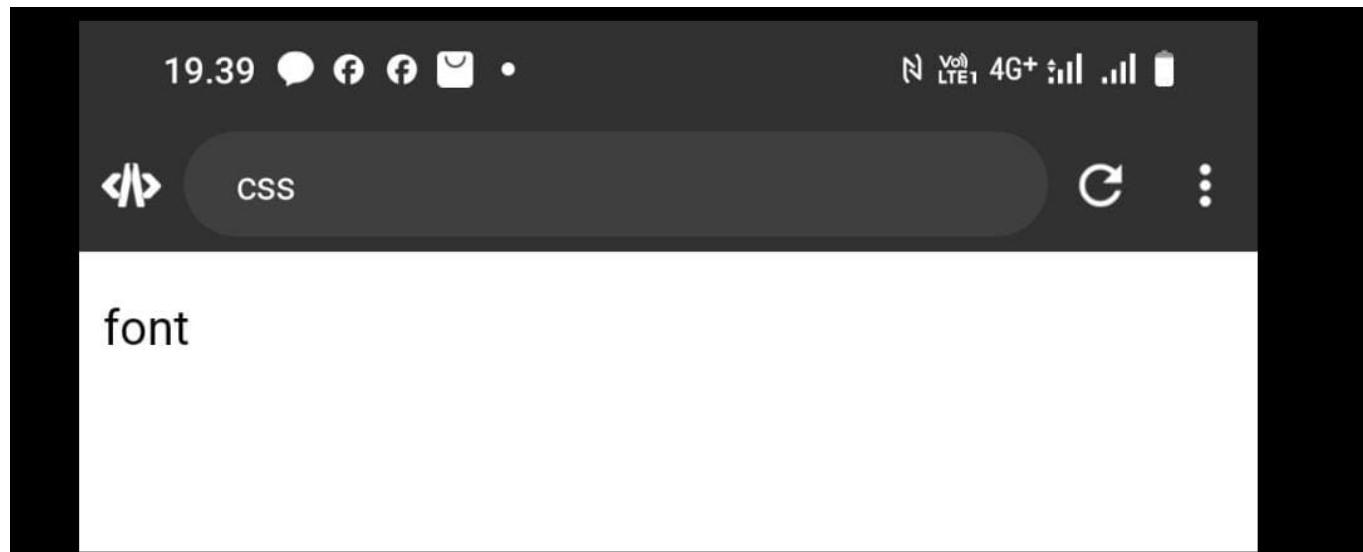
```
<!doctype html>
<html>
  <head>
    <title>css</title>
    <style>
      p {
        font-style: italic ;
      }
    </style>
  </head>
  <body>

    <p>font</p>

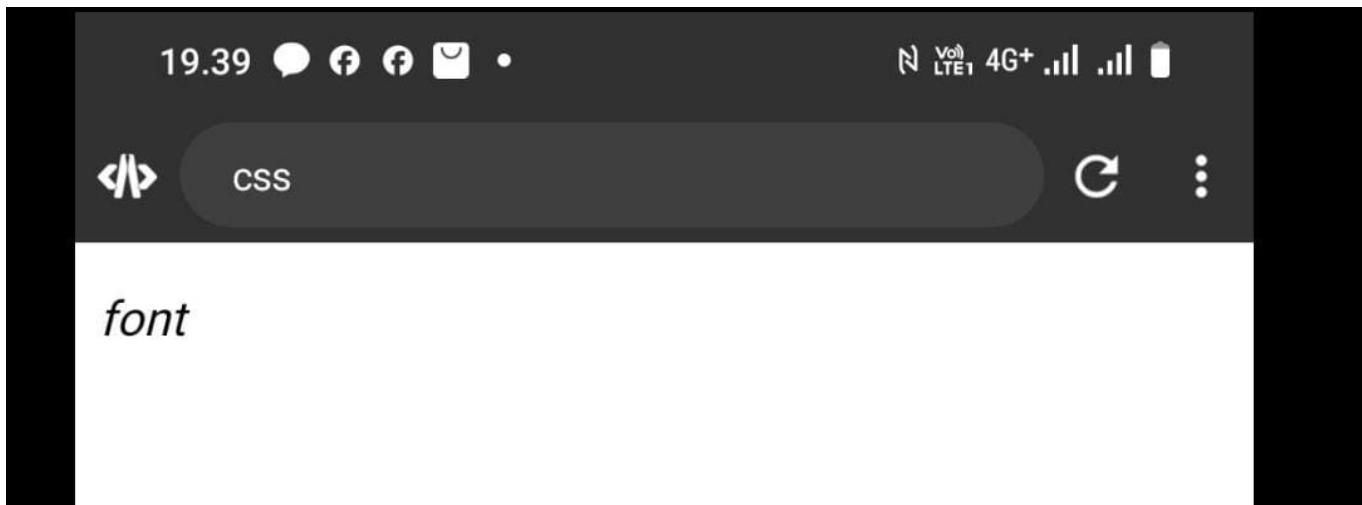
  </body>
</html>
```

Hasil

Before



After



Kesimpulan

Font style Property CSS yang digunakan untuk menentukan gaya penulisan Text.

Font-weight

Penjelasan

Font weight Property CSS yang digunakan untuk mengatur ketebalan Text dalam suatu elemen html.

Beberapa contoh property value nya :

- Normal menampilkan Text dengan gaya standar default
- Bold menampilkan Text dengan menambahkan ketebalan text
- Lighter menampilkan Text dengan gaya Text yang tipis
- Bolder menampilkan Text dengan gaya menambahkan ketebalan pada Text.

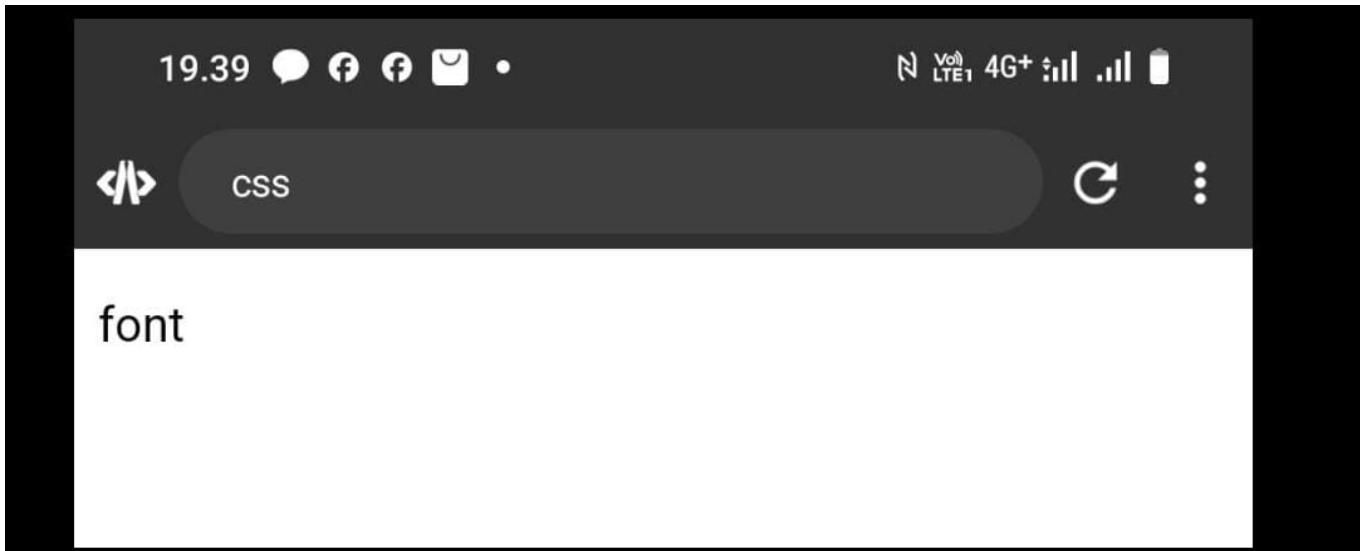
Kode Program

```
<html>
  <head>
    <title>css</title>
    <style>
      p {
        font-weight: bold;
      }
    </style>
  </head>
  <body>
```

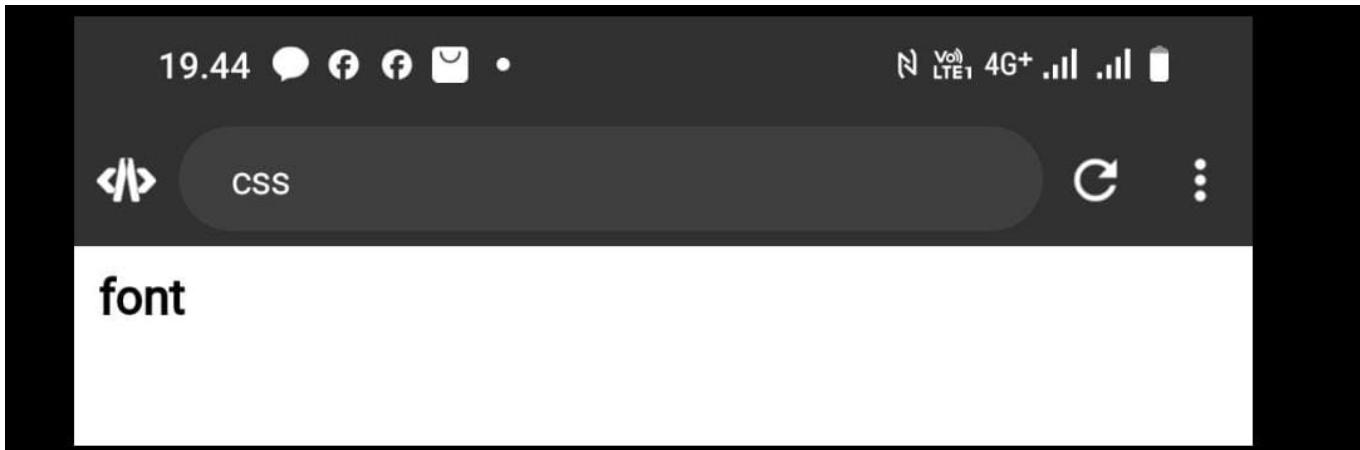
```
<p>font</p>  
  </body>  
</html>
```

Hasil

Before



After



Kesimpulan

Font-weight digunakan untuk mengatur pada suatu Text dalam elemen html.

Font-family

Penjelasan

Font family Property CSS yang digunakan untuk menentukan jenis font atau kumpulan font yang akan digunakan dalam elemen html. Properti ini menerima satu atau lebih nama font, dengan nama font utama diikuti oleh alternatif jika font utama tidak tersedia.

Kode Program

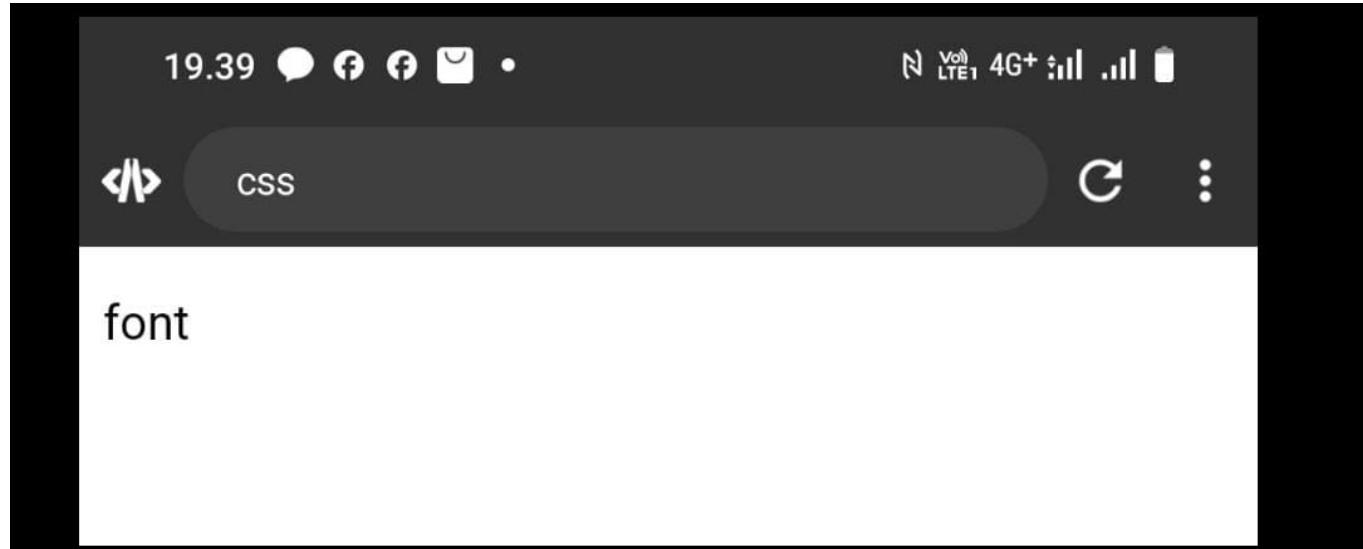
```
<html>
  <head>
    <title>css</title>
    <style>
      p {
        font-family:Times New Roman;
      }
    </style>
  </head>
  <body>

    <p>Materi font</p>

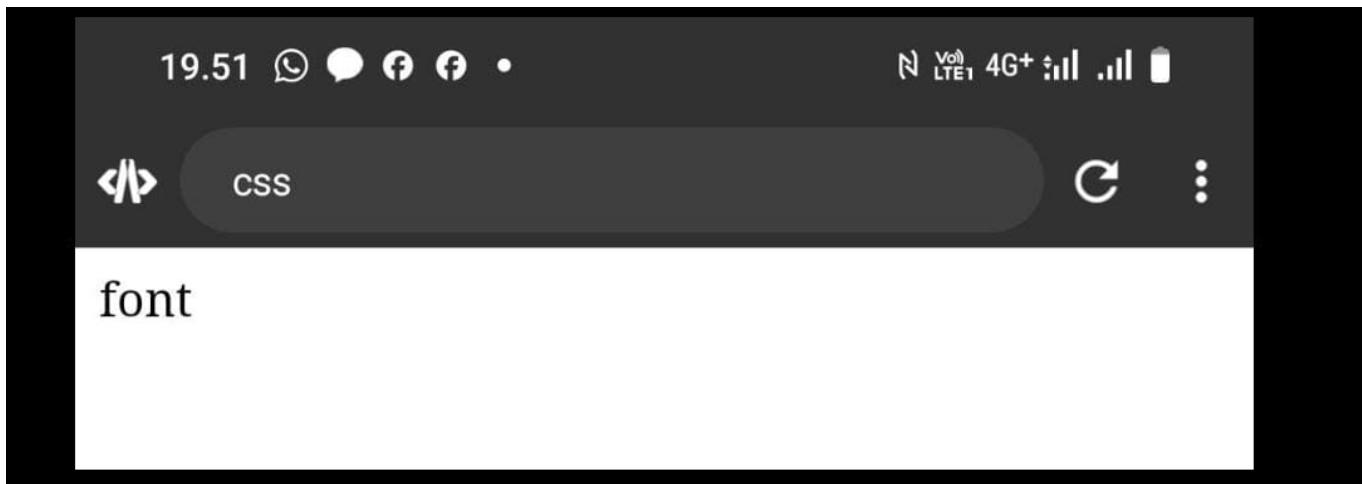
  </body>
</html>
```

Hasil

Before



After



Kesimpulan

Font-family digunakan jika ingin merubah font default pada elemen html.

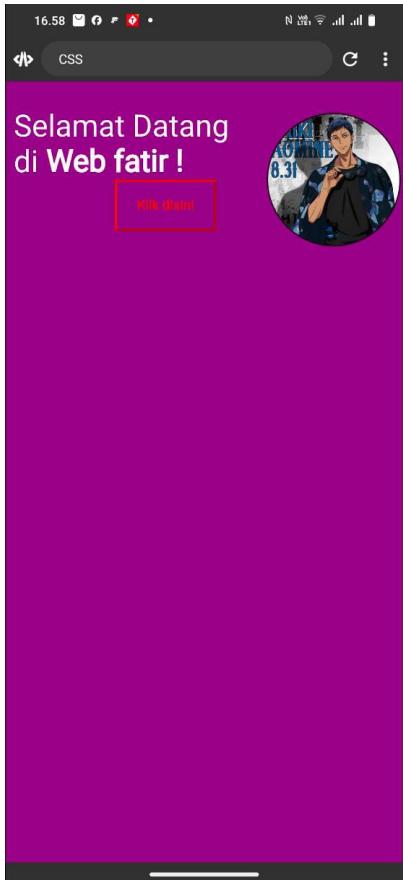
Tantangan Box model

Kode program

```
body{  
    background-color: purple;  
}  
  
span{  
    color: white;  
    font-size: 30px;  
}  
  
button{  
    color: orangered;  
    background-color: purple;  
    border-color: orangered;  
    width: 100px;  
    height: 50px;  
    margin-left: 100px;  
    margin-bottom: 1px;  
}  
  
img {  
    width: 130px;  
    height: 130px;  
    border-radius: 100px;  
    border-color: black;  
    border-style: solid;  
    margin-left: 250px;
```

```
margin-top: -150px;  
}
```

Hasil



Kesimpulan

1. body :

- Warna latar belakang diatur menjadi ungu (background-color: purple).

2. span :

- Teks dalam elemen span berwarna putih (color: white).
- Ukuran font diatur menjadi 30 piksel (font-size: 30px).

3. button :

- Teks pada tombol berwarna oranye-merah (color: orangered).
- Latar belakang tombol juga ungu (background-color: purple).
- Batas tombol berwarna oranye-merah (border-color: orangered).
- Ukuran tombol adalah 100 piksel untuk lebar dan 50 piksel untuk tinggi.

- Ada margin kiri 100 piksel dan margin bawah 1 piksel.

4. img :

- Lebar dan tinggi gambar diatur menjadi 130 piksel.
- Gambar memiliki bentuk bulat dengan radius 100 piksel (border-radius: 100px).
- Warna batas gambar hitam dengan gaya solid.
- Margin kiri 250 piksel dan margin atas negatif 150 piksel.

Box-Model

border

penjelasan

Border (batas) dalam CSS adalah garis yang mengelilingi suatu elemen HTML. Border dapat digunakan untuk memberikan tampilan visual yang jelas dan terpisah antara elemen-elemen di dalam halaman web.

Ada tiga properti utama yang dapat digunakan untuk mengatur border suatu elemen:

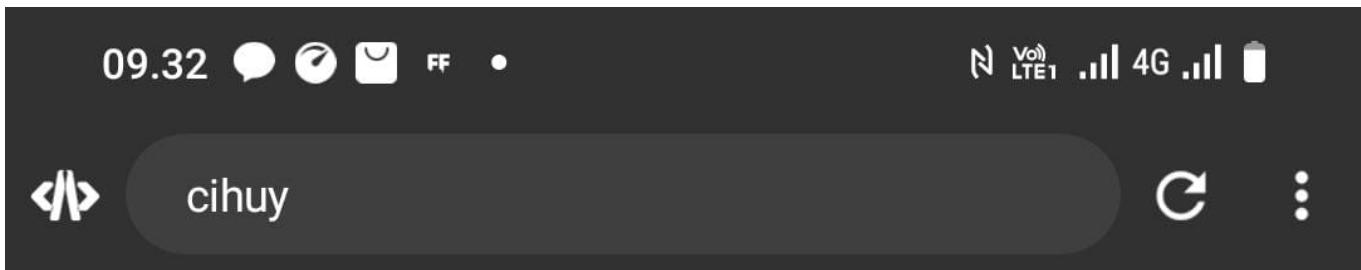
1. border-width : Mengatur lebar border.
2. border-style : Mengatur jenis atau gaya border, seperti solid, dashed, dotted, double, dsb.
3. border-color : Mengatur warna border.
4. border-radius : membuat sudut elemen HTML menjadi melengkung daripada tajam.

kode program

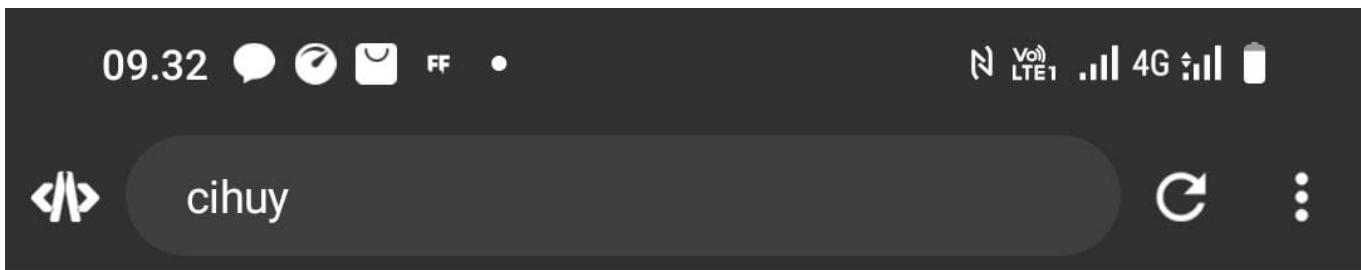
```
Button {  
border-color: red;  
}
```

hasil

before



after



kesimpulan

kesimpulan kode tersebut `div { border-color: red; }` adalah bahwa semua elemen `<div>` pada halaman web akan memiliki border dengan warna merah.

Dengan kode tersebut, setiap elemen `<div>` akan memiliki border dengan lebar default (biasanya 1 piksel), gaya default (biasanya solid), namun warna bordernya akan diatur menjadi merah. Ini berarti elemen `<div>` akan memiliki garis pinggiran berwarna merah mengelilingi seluruh elemennya.

Margin

penjelasan

Margin dalam CSS adalah ruang kosong di sekeliling elemen HTML yang digunakan untuk mengatur jarak antara elemen dengan elemen lainnya atau dengan batas-batas luar halaman web.

Ada empat properti margin yang dapat digunakan:

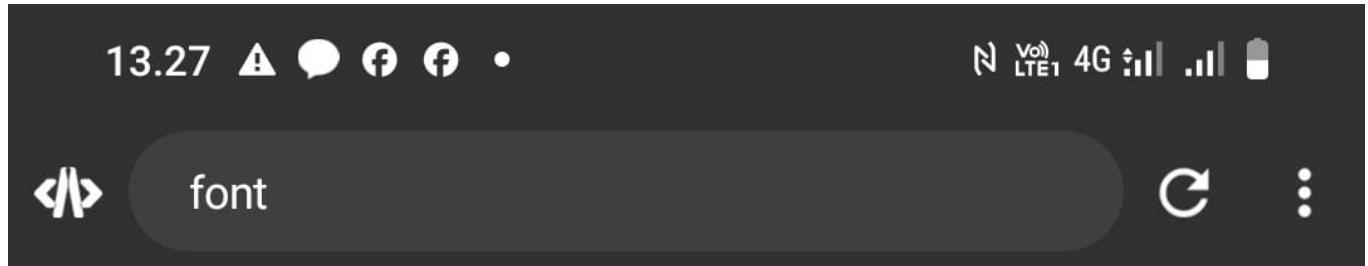
1. margin-top : Mengatur jarak atas elemen.
2. margin-right : Mengatur jarak kanan elemen.
3. margin-bottom : Mengatur jarak bawah elemen.
4. margin-left : Mengatur jarak kiri elemen.

kode program

```
img{  
    margin-left: 600px;  
}
```

hasil

before



Saya berada di kelas 11



after



font



Saya berada di kelas 11



kesimpulan

Kesimpulan dari kode CSS `div { margin-left: 600px; }` adalah bahwa semua elemen `` pada halaman web akan memiliki margin sebesar 600 piksel di sisi kanan.

Dengan kode tersebut, setiap elemen `` akan memiliki ruang kosong (margin) sebesar 600 piksel di sisi kiri, yang berarti elemen-elemen tersebut akan terpisah dengan elemen lain di sebelah kanannya sejauh 600 piksel. Ini akan memengaruhi tata letak (layout) dari elemen-elemen `` tersebut dalam halaman web.

Padding

penjelasan

`padding` digunakan dalam CSS untuk menentukan ruang kosong di sekeliling konten suatu elemen HTML. Properti `padding` dapat diatur secara terpisah untuk setiap sisi elemen, yaitu atas (top), kanan (right), bawah (bottom), dan kiri (left).

Properti `padding` digunakan dalam CSS untuk menentukan ruang kosong di sekeliling konten suatu elemen HTML. Properti `padding` dapat diatur secara terpisah untuk setiap sisi elemen,

yaitu atas (top), kanan (right), bawah (bottom), dan kiri (left).

Ini berarti elemen yang diatur akan memiliki padding sebagai berikut:

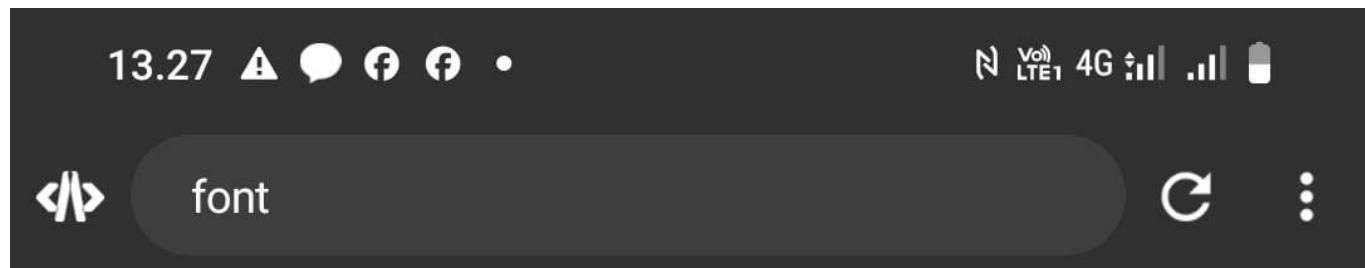
- `padding-left: 10px;` : Padding sebesar 10 piksel di sisi kiri elemen.
- `padding-bottom: 10px;` : Padding sebesar 10 piksel di sisi bawah elemen.
- `padding-right: 10px;` : Padding sebesar 10 piksel di sisi kanan elemen.
- `padding-top: 10px;` : Padding sebesar 10 piksel di sisi atas elemen.

kode program

```
Img {  
padding-left: 10px;  
}
```

hasil

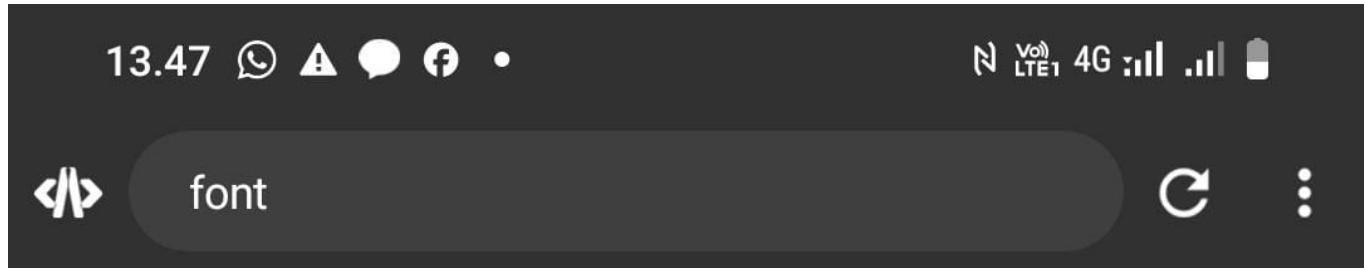
before



Saya berada di kelas 11



after



Saya berada di kelas 11



kesimpulan

Kesimpulan dari kode CSS `div { padding-left: 100px; }` adalah bahwa semua elemen `` pada halaman web akan memiliki padding sebesar 100 piksel di sisi kiri.

Dengan kode tersebut, setiap elemen `` akan memiliki ruang kosong 100 piksel di sisi kiri, yang berarti konten di dalam elemen tersebut akan tergeser ke kiri sejauh 100 piksel dari batas kiri elemen.

Jadi, setiap elemen `` akan memiliki padding sebesar 100 piksel di sisi kiri elemen.

Pseudo class

Hover

- `button:hover` : adalah pseudo-class selector yang menargetkan tombol (button) ketika pengguna mengarahkan kursor ke atasnya (hover) atau diklik.
- `background-color: red;` : Properti ini menetapkan warna latar belakang tombol menjadi merah ketika tombol sedang di-hover.
- `color: white;` : Properti ini menetapkan warna teks pada tombol menjadi putih ketika tombol di-hover.
- `height: 100px;` : Properti yang digunakan untuk membuat tinggi (height) tombol menjadi 100 piksel ketika tombol di-hover.
- `width: 100px;` : Properti yang digunakan untuk membuat lebar (width) tombol menjadi 100 piksel ketika tombol di-hover.

Kode Program

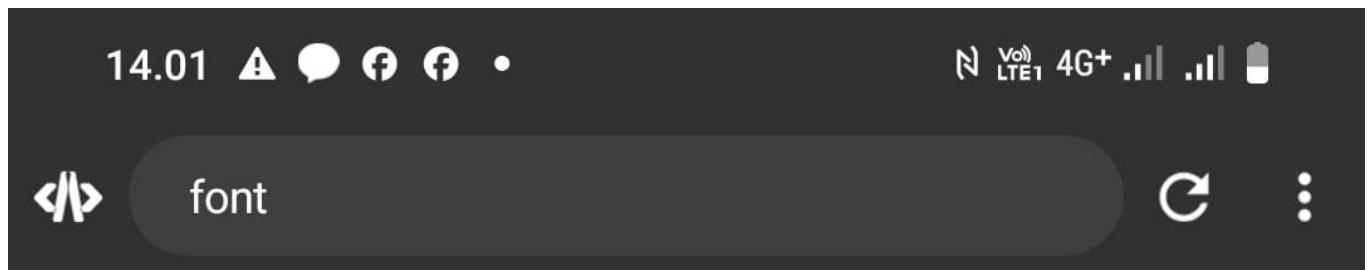
```
button:hover{
    background-color:red;
    color:white;
    height:100px;

    width:100px;

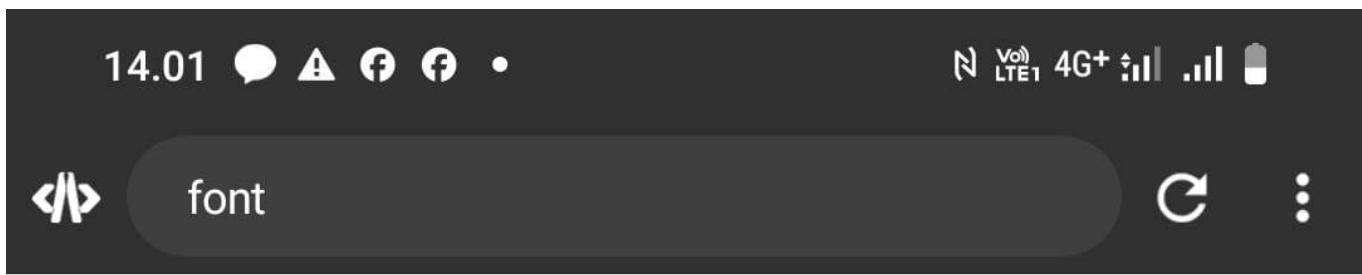
}
```

Hasil

Before



After



Kesimpulan

Setiap property memberikan perubahan yang berbeda dari hover akan memberikan perubahan saat di klik, width dan height untuk mengatur lebar dan tinggi, background-color red memberikan warna merah ketika diklik, dan color white digunakan untuk memberikan warna putih pada teks.

Active

`button:active` : adalah pseudo-class selector yang menargetkan tombol (`button`) ketika tombol tersebut sedang dalam keadaan "active", atau tombol sedang ditekan.

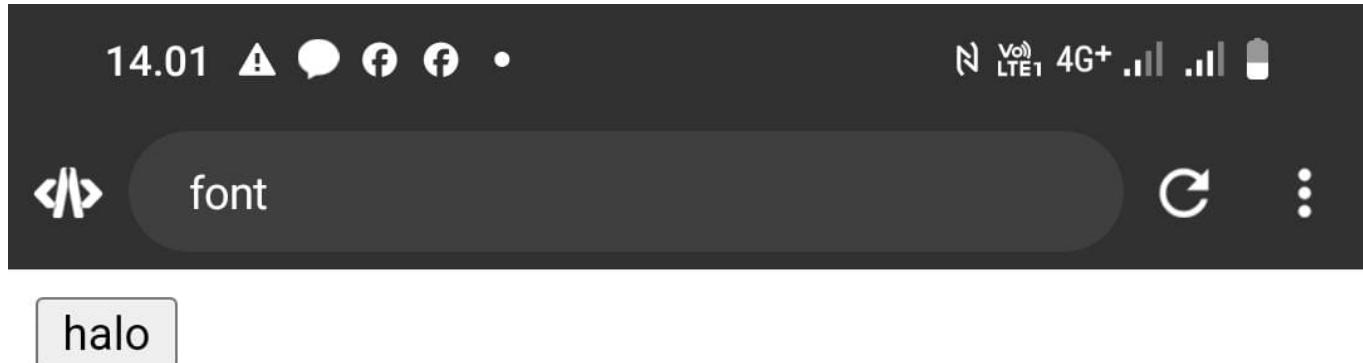
`color: yellow;` : Properti memberikan warna kuning pada teks ketika tombol sedang active atau tombol sedang ditekan.

Kode program

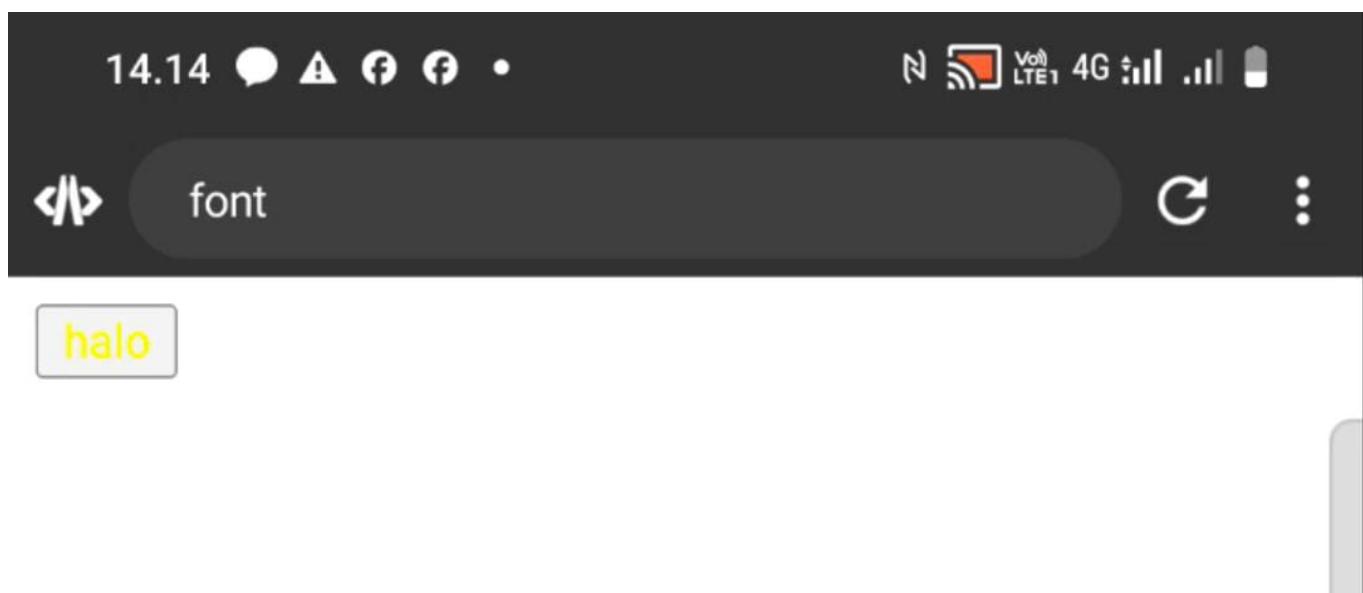
```
button:active{  
    color: yellow ;  
}
```

Hasil

Before



After



Kesimpulan

Teks klik akan berganti warna menjadi kuning jika ditekan dan berputar.

VISITED

Penjelasan

Visited adalah pseudo class yang digunakan untuk menentukan gaya pada elemen yang telah di kunjungi oleh user. umumnya digunakan untuk merubah tampilan tautan yang di klik agar

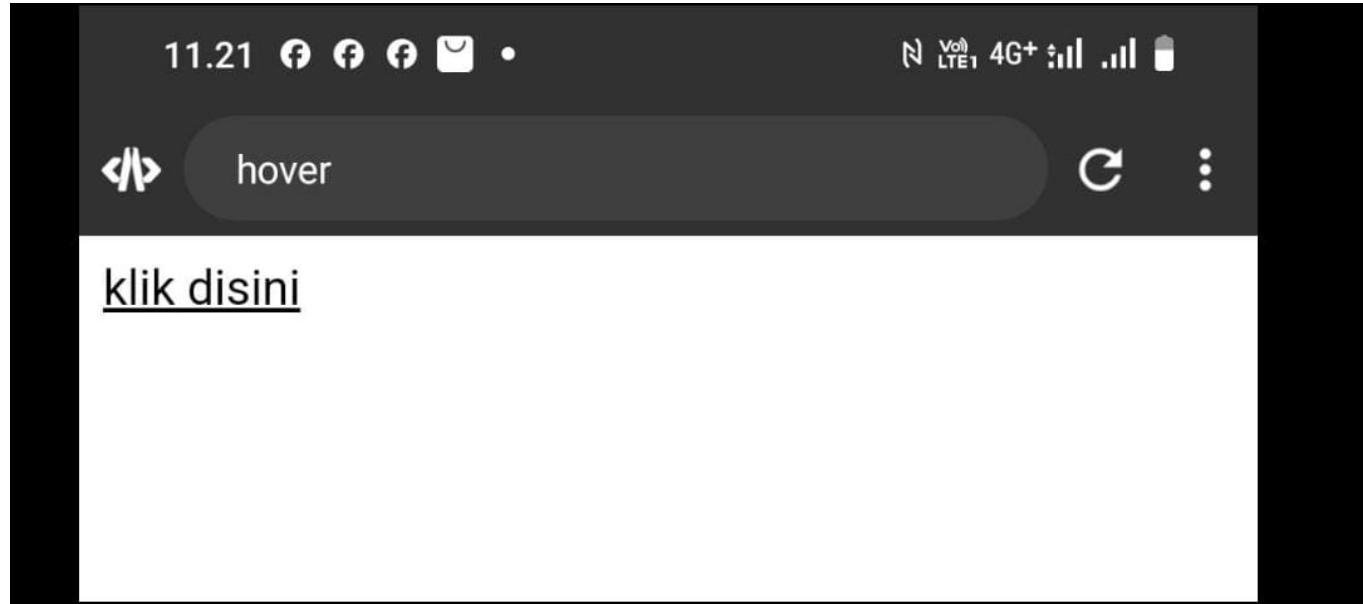
user bisa membedakan mana link yang sudah dikunjungi dan yang belum dikunjungi

Kode Program

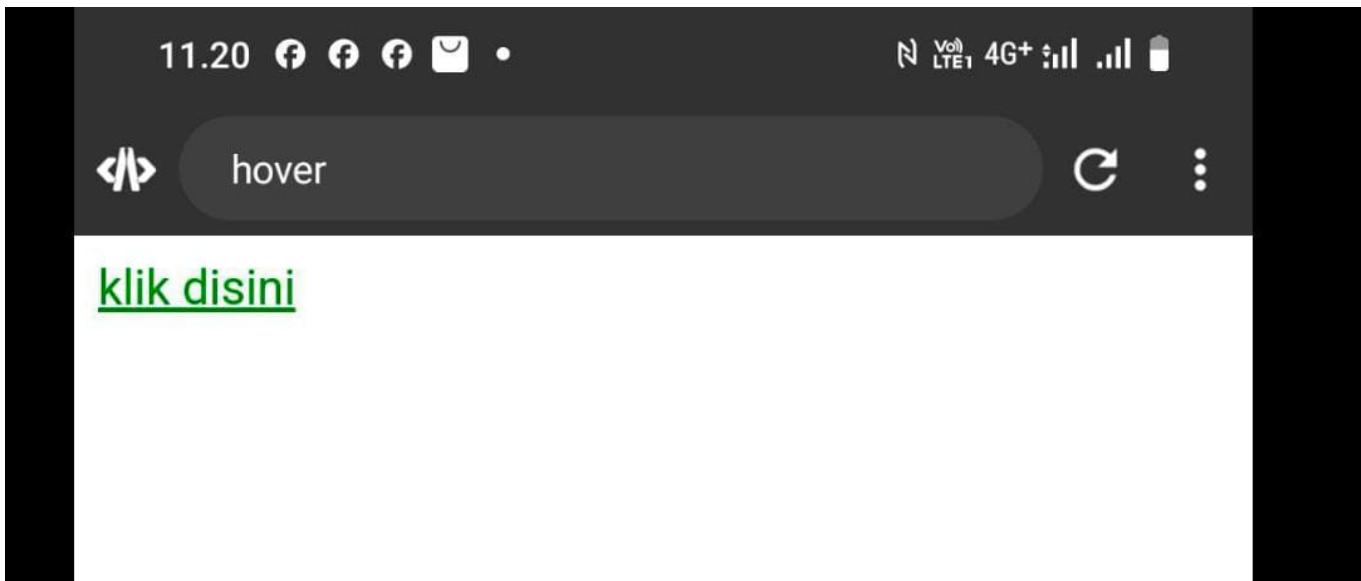
```
<html>
  <head>
    <title>hover</title>
    <style>
      a {
        color: black;
      }
      a:visited {
        color: lightpink;
      }
    </style>
  </head>
  <body>
    <a href="https://www.w3schools.com/html/html_colors.asp">klik disini</a>
  </body>
</html>
```

Hasil

Before



After



Kesimpulan

Visited pseudo Class ini digunakan untuk memberikan gaya pada tampilan link yang sudah dikunjungi.

LINK

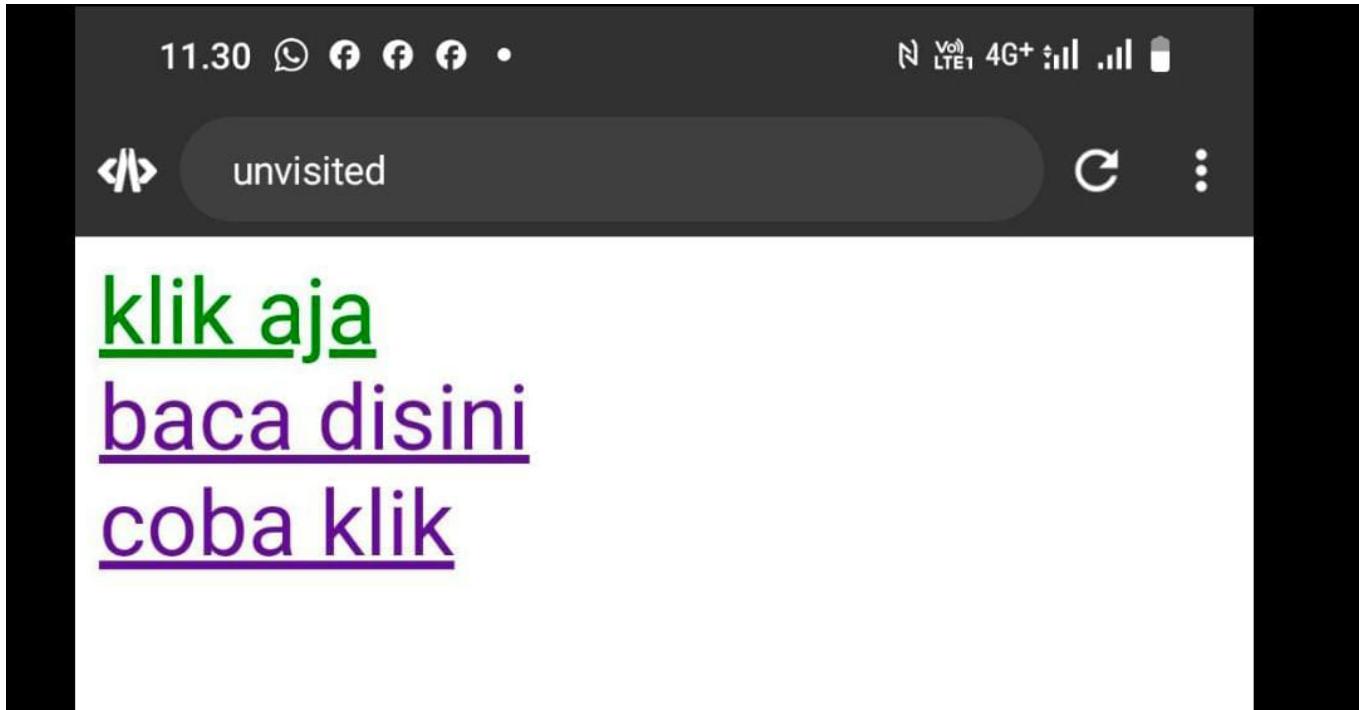
Penjelasan

Link Pseudo Class yang digunakan untuk memberikan gaya pada elemen/link yang belum dikunjungi, kebalikannya dari Visited.

Kode Program

```
<html>
  <head>
    <title>unvisited</title>
    <style>
      a:link {
        font-size: 30px;
        color:lightpink;
      }
    </style>
  </head>
  <body>
    <a href="https://developer.mozilla.org/">klik aja</a><br>
    <a href="https://www.w3schools.com/">baca disini</a><br>
    <a href="https://css-tricks.com/">coba klik</a>
  </body>
</html>
```

Hasil



Kesimpulan

Jadi fungsi dari Pseudo Class link ini memberikan gaya pada tautan yang belum dikunjungi.

Transition

Penjelasan

Transition dalam css adalah sebuah mekanisme untuk mengendalikan perubahan pada sebuah elemen secara halus.

Transition-property menentukan property apa yang akan diberi efek Transition, seperti width,color, atau all(semua property).

transition-duration menentukan waktu berapa lama efek Transitionnya di ukur dalam detik seperti 0,5s.

trasition-timing-function menentukan bagaimana kecepatan perubahan nilai properti selama transisi. Seperti **ease** : lambat diawal, cepat di tengah, dan lambat di akhir **ease-in** : dimulai dengan lambat dan semakin cepat seiring waktu **ease-out** : cepat diawal dan lambat diakhiri **ease-in-out** : Kombinasi dari ease-in dan ease-out, menciptakan efek transisi mulai lambat, cepat di tengah, dan melambat kembali **linear** : memberikan perubahan langsung secara konstan

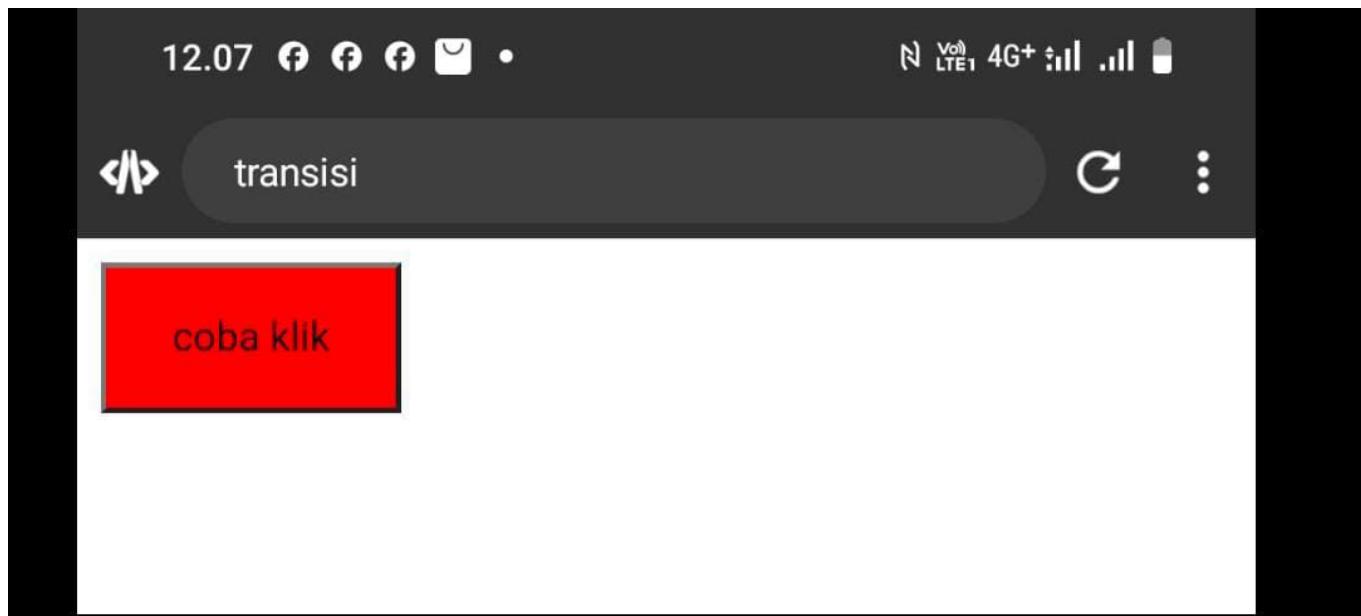
Kode Program

```
<html>
  <head>
    <title>transisi</title>
  </head>
  <style>
    button {
      height: 50px;
      width: 100px;
      background-color: red;
    }

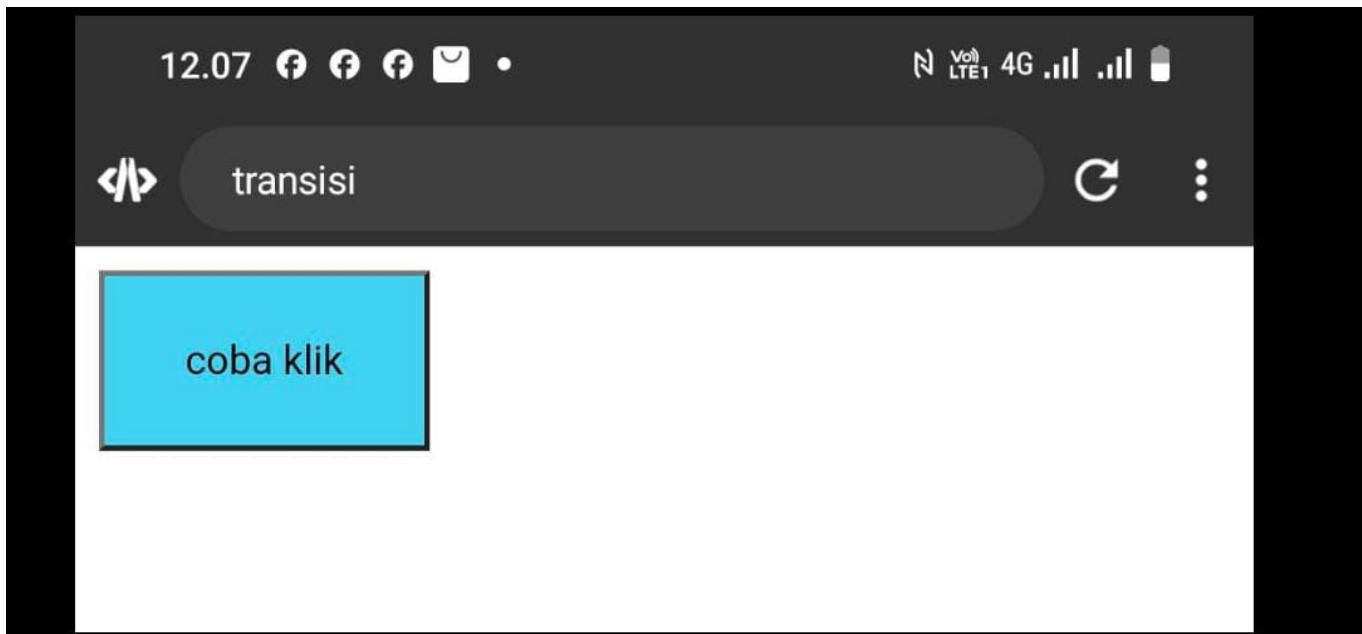
    button:hover {
      height: 60px;
      width: 110px;
      background-color: skyblue;
      transition: all 0.5s ease-in-out;
    }
  </style>
  <body>
    <button>coba klik</button>
  </body>
</html>
```

Hasil

Before



After



Kesimpulan

Transition Property CSS yang digunakan untuk memberikan efek perubahan secara halus dalam sebuah elemen html.

Tantangan Transition

Code

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS</title>
    <link rel="stylesheet" href="Style.css">
  </head>
  <body>

    <div class="containersub">
      <br>
      <span>
        Selamat Datang
      </span>
      <br>
      <span>
        di <b>Web fatir !</b><br>
        <button>Klik disini</button>
      </span>
    </div>
    
```

```
</body>
```

```
</html>
```

```
body{  
    background-color: purple;  
}  
  
span{  
    color: white;  
    font-size: 30px;  
}  
  
button{  
    color: orangered;  
    background-color: purple;  
    border-color: orangered;  
    width: 100px;  
    height: 50px;  
    margin-left: 100px;  
    margin-bottom: 1px;  
}  
  
button:hover {  
    height: 60px;  
    width: 110px;  
    background-color: skyblue;  
    transition: all 0.5s ease-in-out;  
}  
  
img {  
    width: 130px;  
    height: 130px;  
    border-radius: 100px;  
    border-color: black;  
    border-style: solid;  
    margin-left: 250px;  
    margin-top: -150px;  
}
```

Hasil

Before



Selamat Datang di Web fatir !

Klik disini





After

19.23 ☺ ● ○ ⏱ •

N VoLTE 4G ⏱



Belajar CSS / Filee CSS



⋮

17.08 ☺ ● ○ ⏱ •

N VoLTE ⏱



CSS

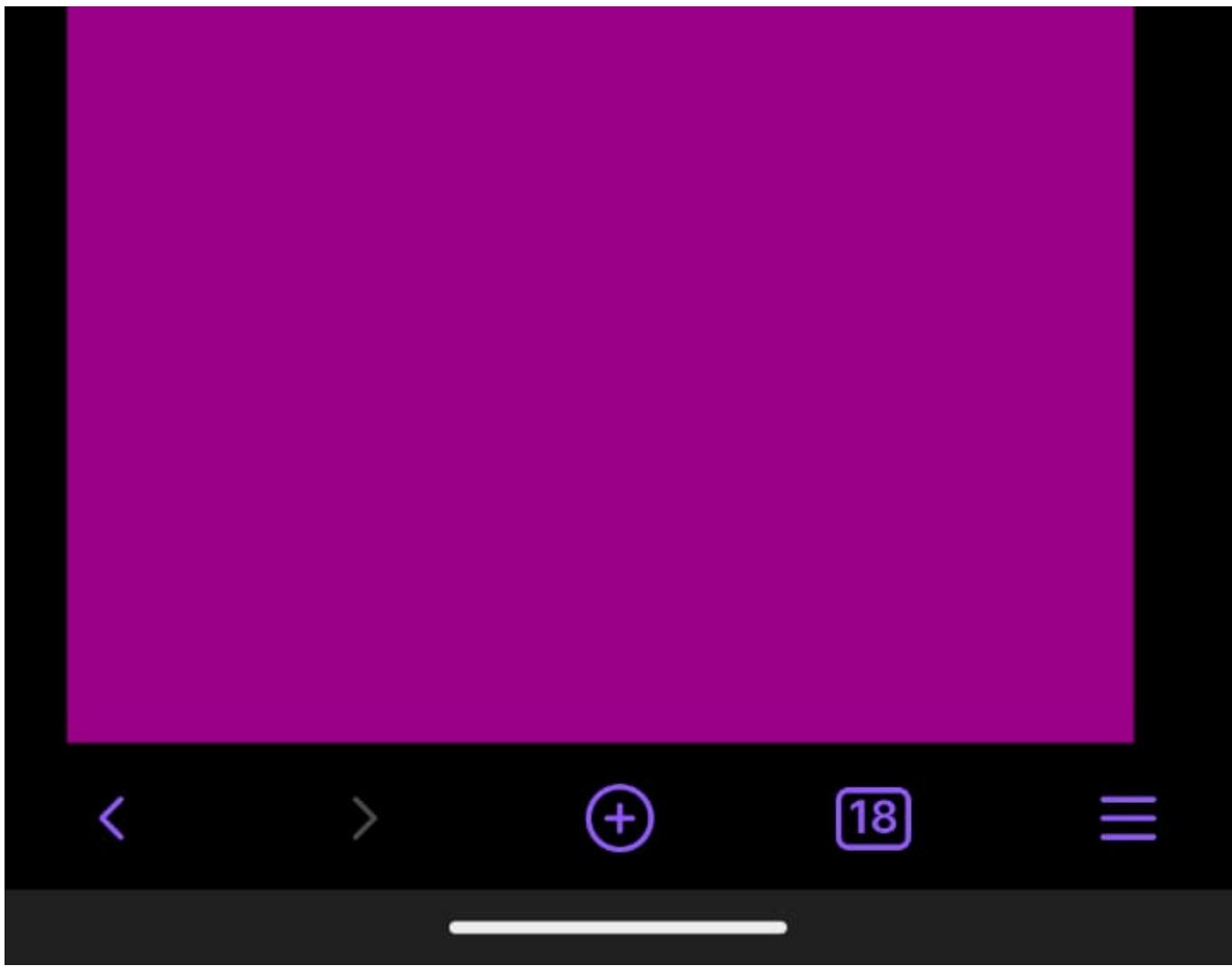


⋮

Selamat Datang di Web fatir !

Klik disini





Kesimpulan

1. body :

- Warna latar belakang diatur menjadi ungu (background-color: purple).

2. span :

- Teks dalam elemen span berwarna putih (color: white).
- Ukuran font diatur menjadi 30 piksel (font-size: 30px).

3. button :

- Teks pada tombol berwarna oranye-merah (color: orangered).
- Latar belakang tombol juga ungu (background-color: purple).
- Batas tombol berwarna oranye-merah (border-color: orangered).
- Ukuran tombol adalah 100 piksel untuk lebar dan 50 piksel untuk tinggi.
- Ada margin kiri 100 piksel dan margin bawah 1 piksel.

- Pada saat tombol di-hover (diseleksi dengan mouse), tinggi bertambah menjadi 60 piksel, lebar menjadi 110 piksel, latar belakang berubah menjadi biru langit (`background-color: skyblue`), dan transisi terjadi dalam 0.5 detik.

1. `img` :

- Lebar dan tinggi gambar diatur menjadi 130 piksel.
- Gambar memiliki bentuk bulat dengan radius 100 piksel (`border-radius: 100px`).
- Warna batas gambar hitam dengan gaya solid.
- Margin kiri 250 piksel dan margin atas negatif 150 piksel.

Transform

Scale

Penjelasan

digunakan untuk mengubah tampilan suatu elemen, seperti memindahkannya, memutarinya, atau mengubah ukurannya. nilai Scale didefinisikan dengan menyebut faktor perubahannya seperti Scale(2) akan memperbesar elemen menjadi 2 kali lipat.

Code

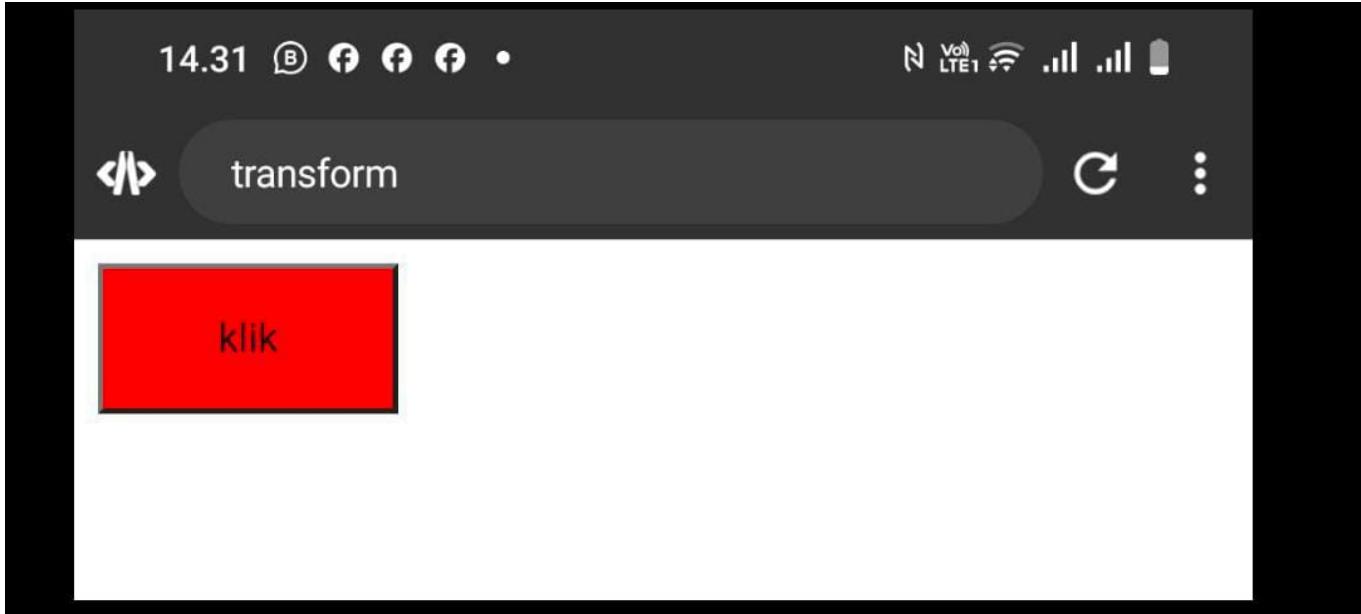
```
<html>
  <head>
    <title>transform</title>
  </head>
  <style>
    button {
      height: 50px;
      width: 100px;
      background-color: red;
    }

    button:hover {
      height: 60px;
      width: 110px;
      background-color: skyblue;
      transform:scale(0.5);
    }
  </style>
  <body>
    <button>klik</button>
```

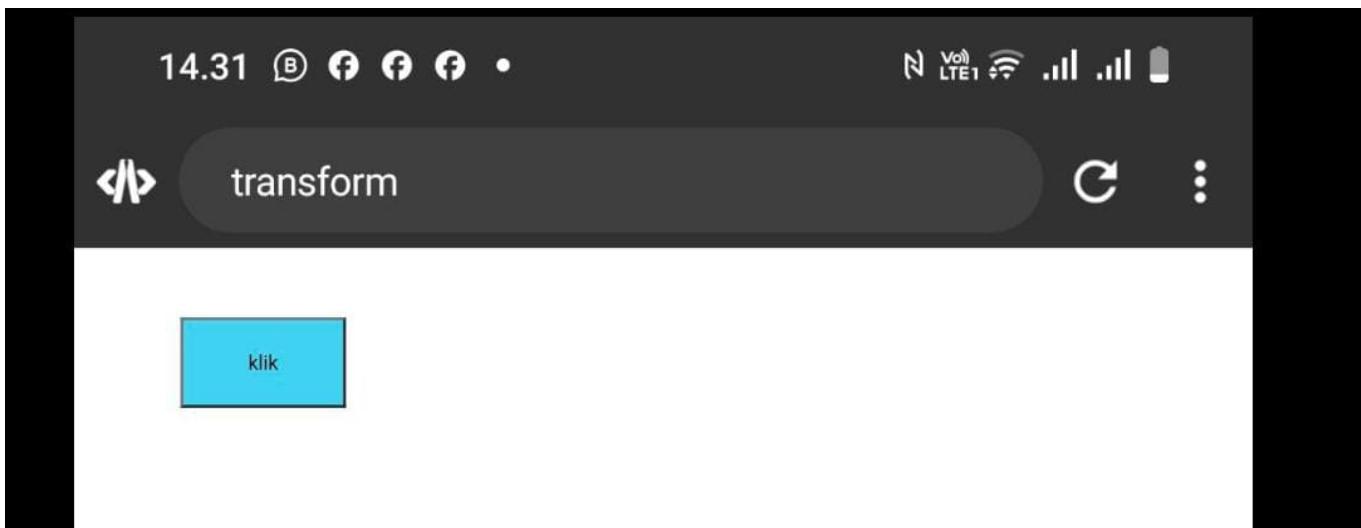
```
</body>  
</html>
```

Hasil

BEFORE



AFTER



Kesimpulan

Efek hover ini mengubah ukuran tombol dan warna saat kursor mouse ditempatkan di atasnya, memberikan interaksi visual yang menarik bagi pengguna.

ScaleX

Penjelasan

ScaleX Property CSS yang digunakan untuk mengubah skala elemen hanya pada sumbu horizontal saja. Misalnya, transform: scaleX(2) akan membuat elemen menjadi dua kali lebih lebar dari ukurannya.

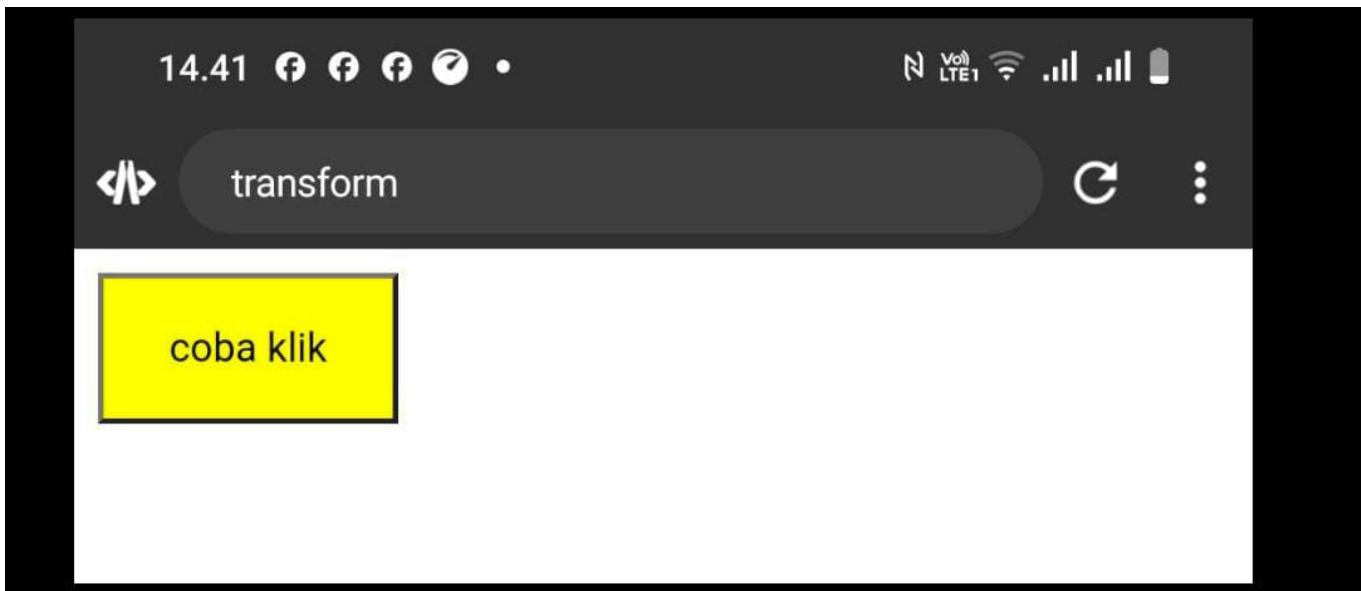
Kode Program

```
<html>
  <head>
    <title>transform</title>
  </head>
  <style>
    button {
      height: 50px;
      width: 100px;
      background-color: yellow;
    }

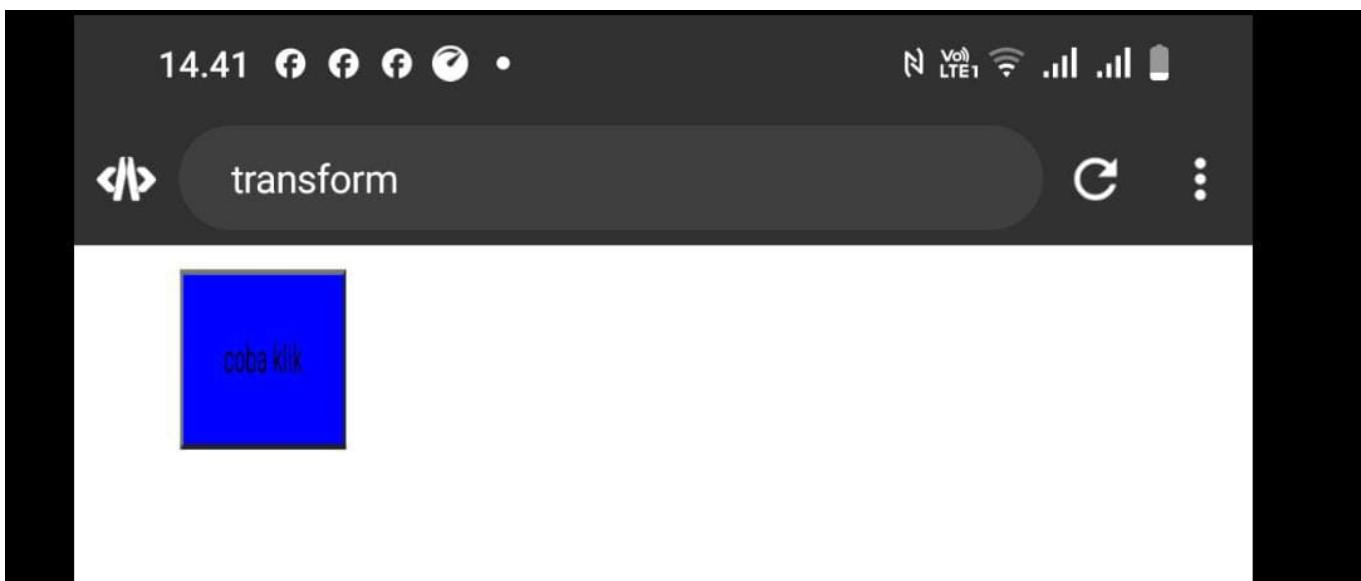
    button:hover {
      height: 60px;
      width: 110px;
      background-color: blue;
      transform:scaleX(0.5);
    }
  </style>
  <body>
    <button>coba klik</button>
  </body>
</html>
```

Hasil

Before



After



Kesimpulan

Efek ini memberikan umpan balik visual yang menarik dan responsif kepada pengguna, dengan perubahan warna, ukuran, dan bentuk ketika tombol di-hover. Dalam hal ini, tombol akan terlihat lebih ramping secara horizontal ketika di-hover karena transformasi skala X.

Rotate

Penjelasan

`rotate` Property CSS yang digunakan untuk memutar elemen disekitar titik pusatnya. Nilai `rotate` dapat dinyatakan dalam derajat(deg) seperti `rotate(45deg)`.

Code

```
<html>
  <head>
    <title>transform</title>
  </head>
  <style>
    button {
      height: 50px;
      width: 100px;
      background-color: yellow;
    }

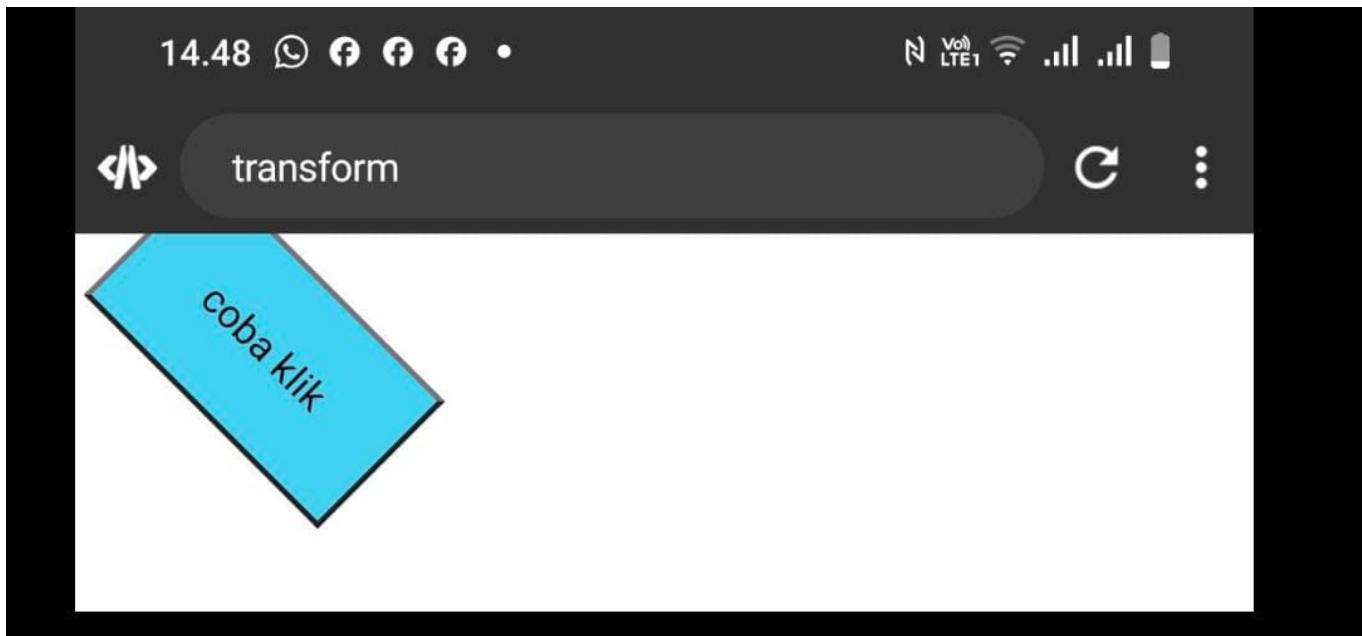
    button:hover {
      height: 60px;
      width: 110px;
      background-color: skyblue;
      transform:rotate(45deg);
    }
  </style>
  <body>
    <button>coba klik</button>
  </body>
</html>
```

Hasil

Before



After



Kesimpulan

Dengan efek `hover` ini, tombol memberikan umpan balik visual yang menarik dengan perubahan warna, ukuran, dan rotasi ketika kursor mouse berada di atasnya. Perubahan rotasi 45 derajat menambah dimensi unik pada interaksi, sehingga tombol akan terlihat miring saat di-hover.

SkewX

Penjelasan

`SkewX` digunakan untuk memiringkan elemen sepanjang horizontal. Nilai `skewX` dinyatakan dengan derajat(deg) seperti `skewX(30deg)`.

Kode Program

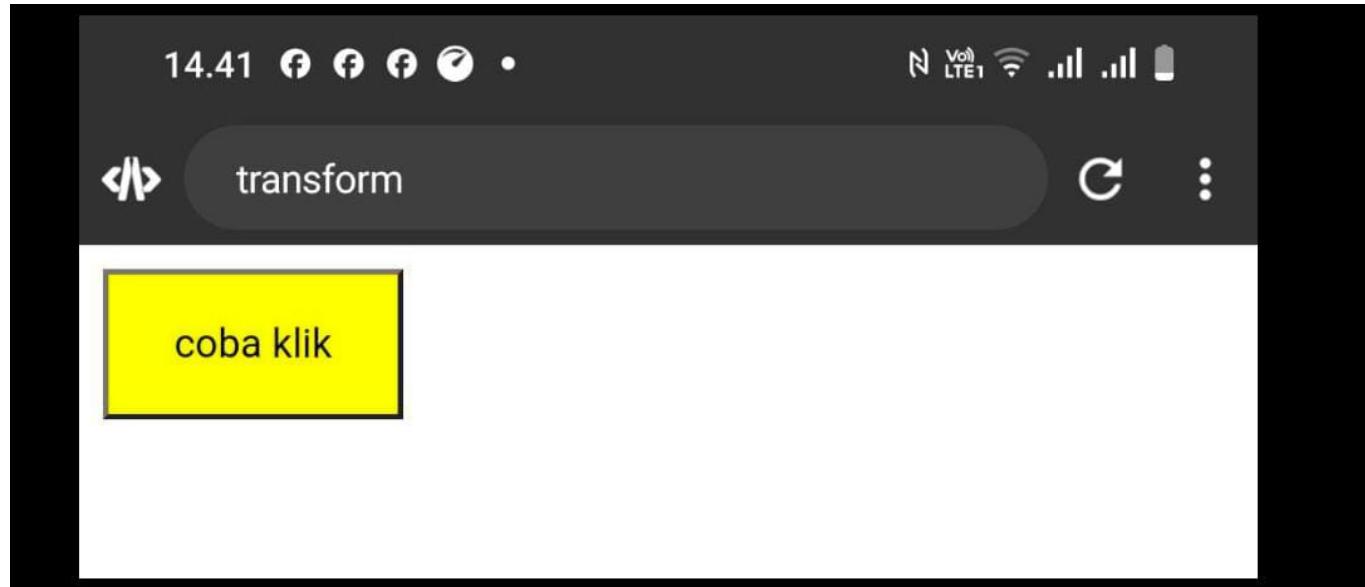
```
<html>
  <head>
    <title>transform</title>
  </head>
  <style>
    button {
      height: 50px;
      width: 100px;
      background-color: yellow;
    }

    button:hover {
```

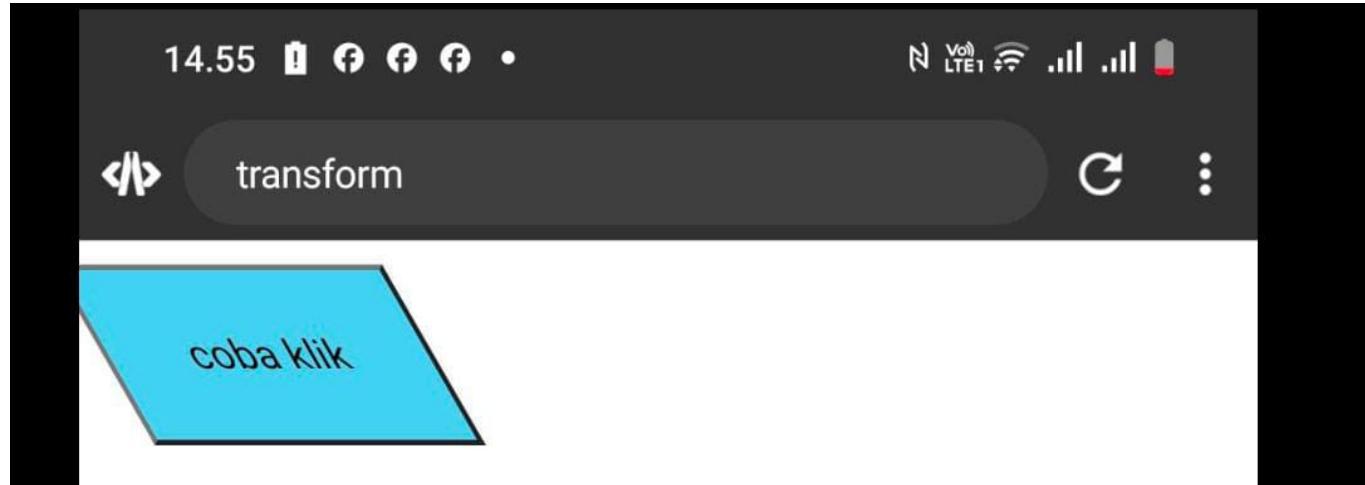
```
height: 60px;  
width: 110px;  
background-color: skyblue;  
transform:skewX(30deg);  
}  
</style>  
<body>  
  <button>coba klik</button>  
</body>  
</html>
```

Hasil

Before



After



Kesimpulan

Efek hover ini menghasilkan umpan balik visual yang menarik karena perubahan warna, ukuran, dan bentuk tombol saat kursor mouse berada di atasnya. Perubahan `skewX(30deg)` memberikan tampilan miring pada tombol, sehingga tombol terlihat lebih miring atau terdistorsi ke samping.

Skew

Penjelasan

`Skew` digunakan untuk memiringkan elemen baik secara horizontal maupun vertikal dalam satu nilai. Nilai skew dapat dinyatakan dalam derajat(deg) seperti `skew(45deg, 20deg)` nilai yang pertama untuk horizontal sedangkan nilai yang kedua untuk vertikal.

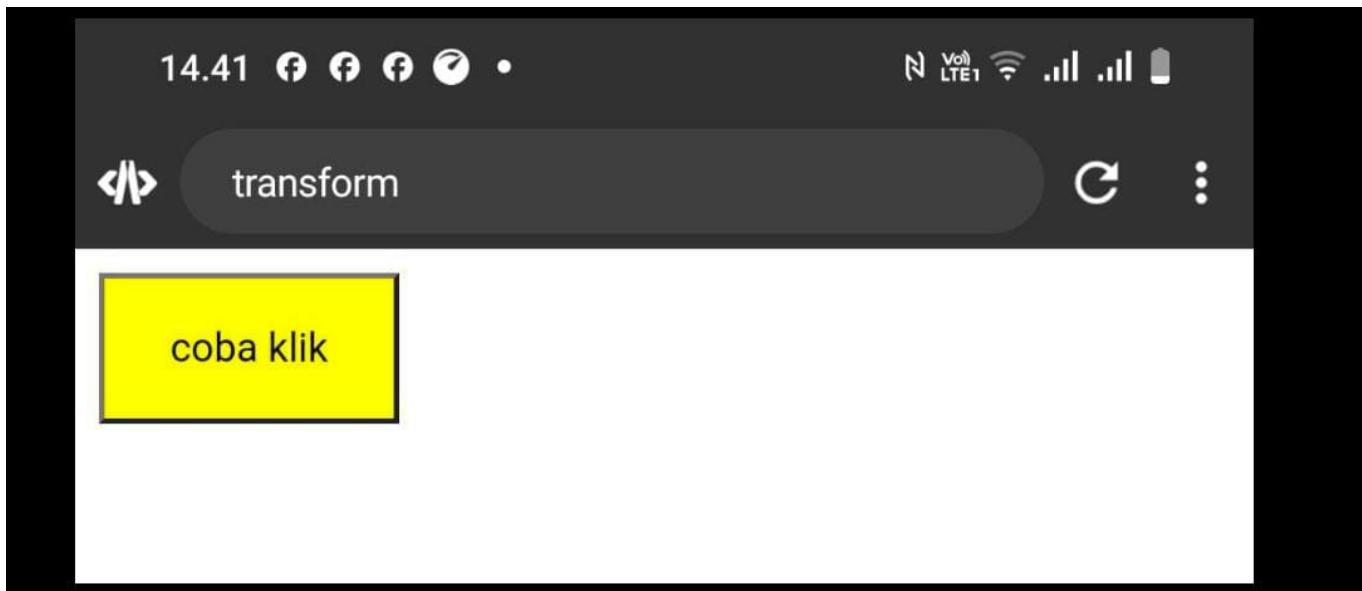
Code

```
<html>
  <head>
    <title>transform</title>
  </head>
  <style>
    button {
      height: 50px;
      width: 100px;
      background-color: yellow;
    }

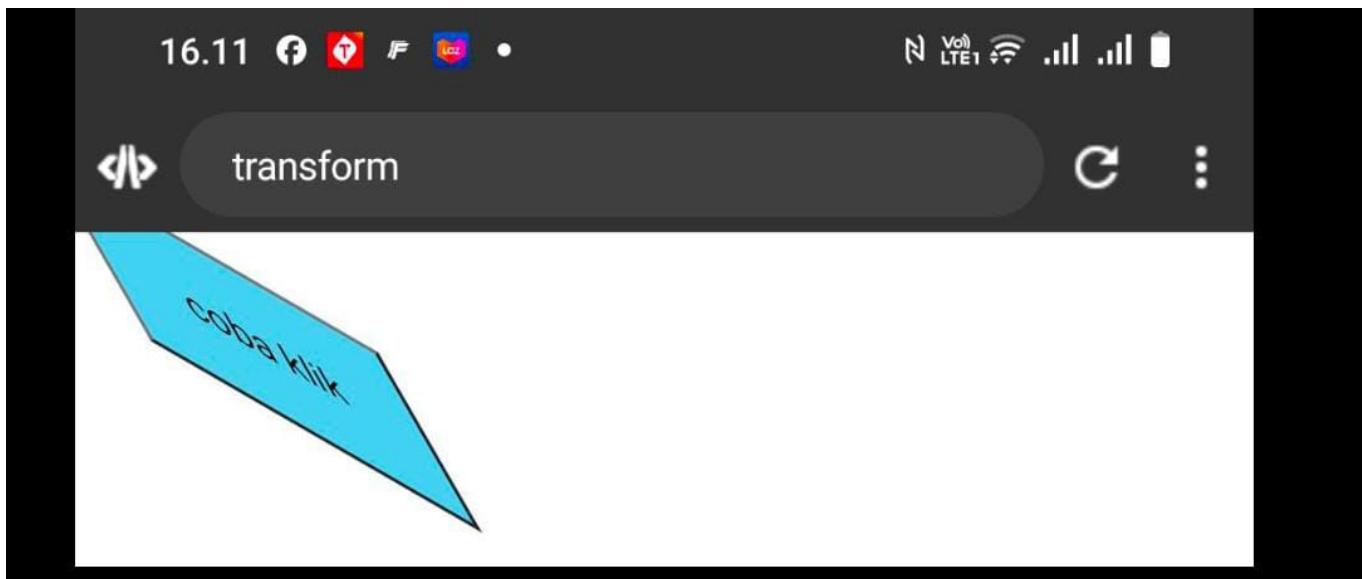
    button:hover {
      height: 60px;
      width: 110px;
      background-color: skyblue;
      transform: skew(30deg, 30deg);
    }
  </style>
  <body>
    <button>coba klik</button>
  </body>
</html>
```

Hasil

Before



After



Kesimpulan

`hover` dapat menciptakan efek interaktif ketika pengguna berinteraksi dengan elemen pada halaman web. Dengan mengubah ukuran, warna, dan transformasi pada elemen tombol, kode ini memberikan umpan balik visual yang menarik bagi pengguna.

Translate

Penjelasan

`Translate` digunakan untuk memindahkan elemen dari posisi awalnya. Nilai translate dapat berupa piksel(px) atau persen(%), seperti `translate(100px, 50px)`, nilai awalnya akan memindahkan elemen 100px ke kanan dan nilai yang kedua digunakan untuk memindahkan

kebawah sebanyak 50px, jika ingin geser kekiri kita beri nilai awalnya dengan minus, begitupun jika kita ingin geser keatas hanya perlu tambahkan minus pada nilai yang kedua.

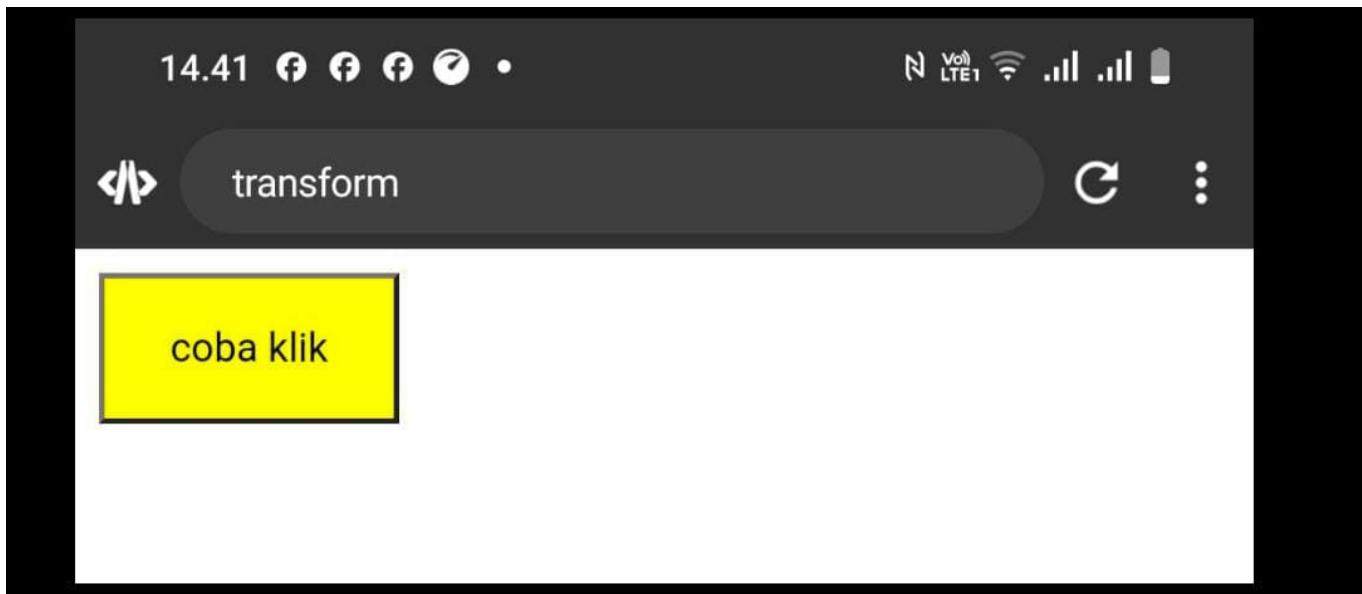
Code

```
<html>
  <head>
    <title>transform</title>
  </head>
  <style>
    button {
      height: 50px;
      width: 100px;
      background-color: yellow;
    }

    button:hover {
      height: 60px;
      width: 110px;
      background-color: skyblue;
      transform:translate(100px, 50px);
    }
  </style>
  <body>
    <button>coba klik</button>
  </body>
</html>
```

Hasil

Before



After

"AsetCSS/A22.png" could not be found.

Kesimpulan

`hover` dapat digunakan untuk mengubah ukuran, warna, dan posisi elemen ketika pengguna mengarahkan kursor ke elemen tersebut. Efek `hover` dalam kode ini menghasilkan umpan balik visual yang jelas dan menarik, serta menunjukkan efek terjemahan (perpindahan) yang memindahkan tombol ke lokasi baru saat kursor diarahkan ke atasnya.

Matrix

Penjelasan

`Matrix` adalah salah satu fungsi transformasi yang digunakan untuk mengubah tata letak elemen HTML. Contoh property value nya seperti `matrix(1, -0.3, 0.6, 1, 50, 20)`.

Kode Program

```
<html>
  <head>
    <title>transform</title>
  </head>
  <style>
    button {
      height: 50px;
      width: 100px;
      background-color: yellow;
    }
  </style>
</html>
```

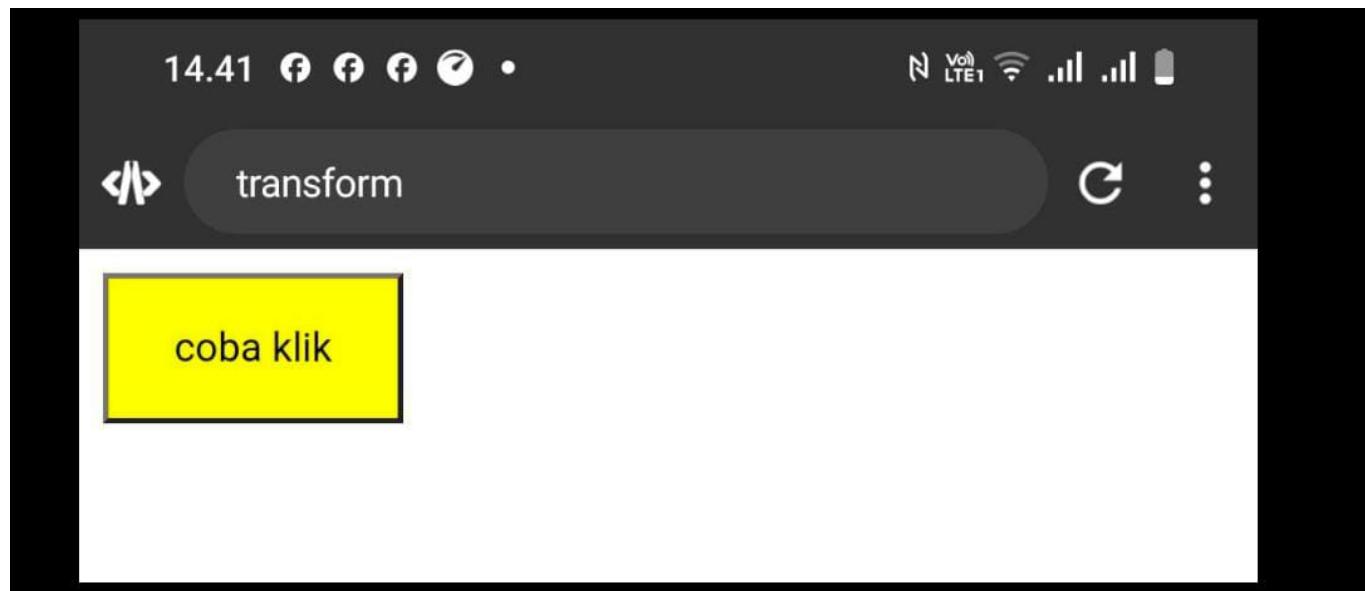
```
}

button:hover {
    height: 60px;
    width: 110px;
    background-color: skyblue;
    transform: matrix(1, -0.3, 0.6, 1, 50, 20);
}

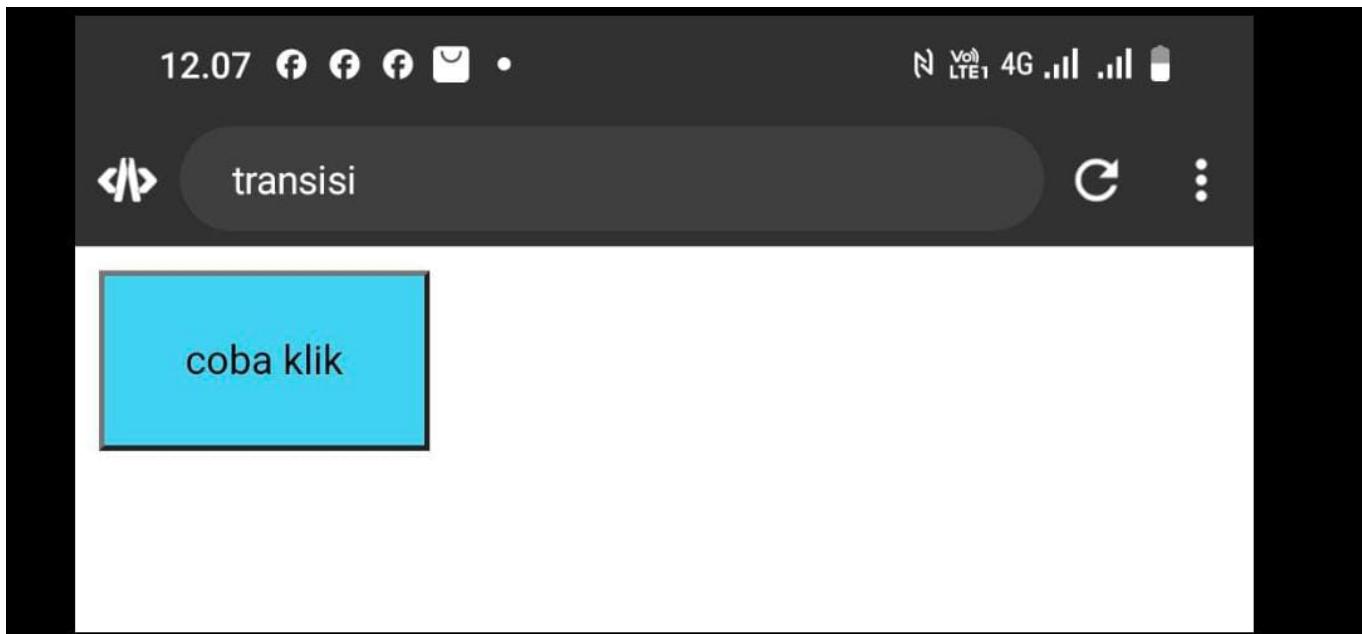
</style>
<body>
    <button>coba klik</button>
</body>
</html>
```

Hasil

Before



After



kesimpulan

`hover` bisa digunakan untuk mengubah atribut elemen seperti ukuran, warna, dan transformasi menggunakan transformasi matriks. Dengan efek ini, tombol memberikan umpan balik visual dan efek animasi yang kompleks saat pengguna mengarahkannya.

Flex Box

Flex Container

FLEX DIRECTION

Penjelasan

Flex direction digunakan untuk mengatur arah tata letak elemen dalam sebuah flexbox container.

Beberapa contoh property value nya :

- `row` : Flex item disusun dalam satu baris dari kiri ke kanan.
- `row-reverse` : Sama seperti `row`, tetapi urutan item dibalik, dari kanan ke kiri.
- `column` : Flex item disusun dalam satu kolom dari atas ke bawah.
- `column-reverse` : Sama seperti column, tetapi urutan item dibalik, dari bawah ke atas

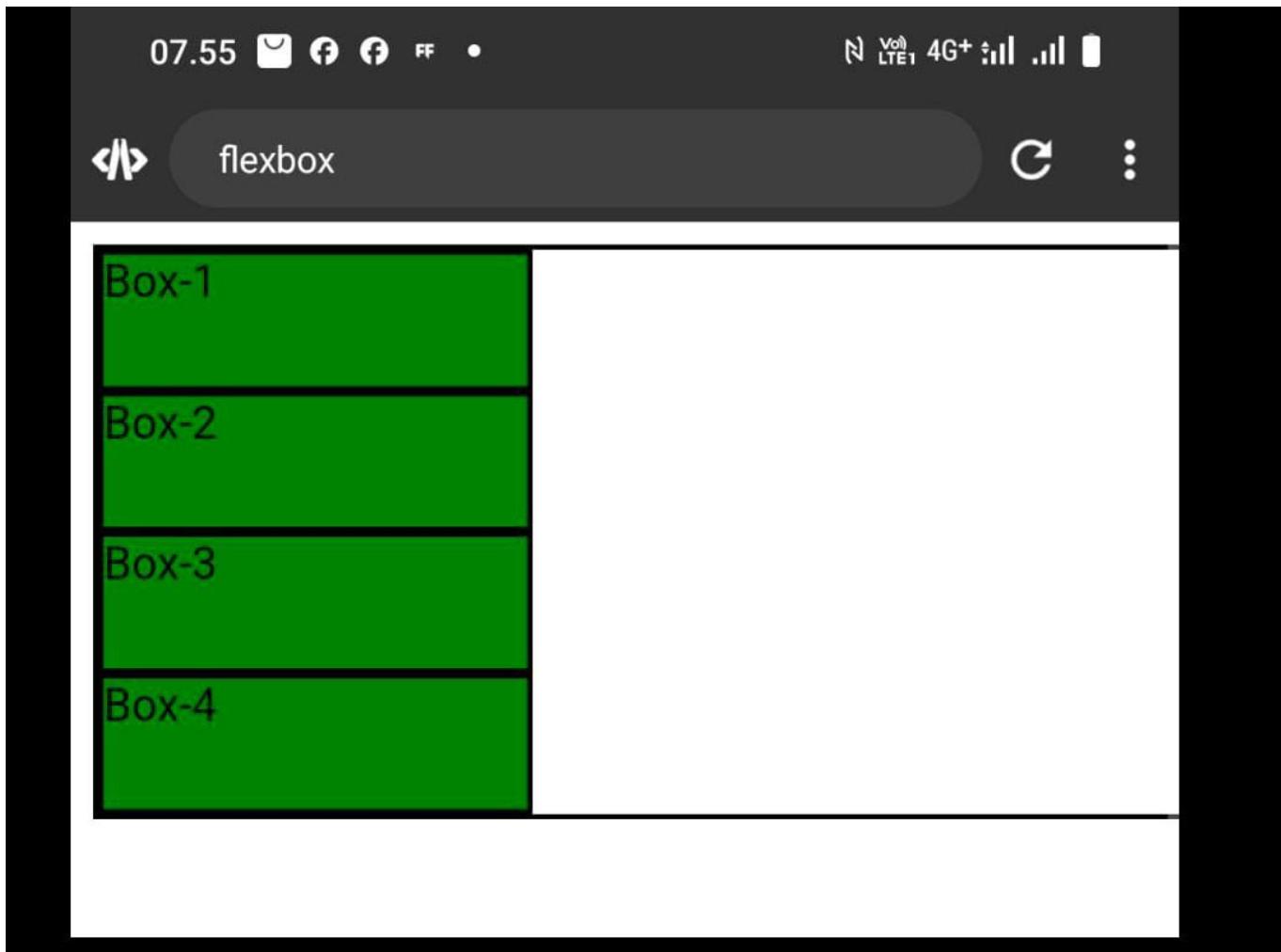
Kode Program

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

```
<title>flexbox</title>
</head>
<style>
    .container {
        display: flex;
        flex-direction: column;
        border: 2px solid black;
        height: 200px;
        width: 400px;
    }

    .item {
        background-color: green;
        border: 2px solid black;
        height: 50px;
        width: 150px;
    }
</style>
<body>
    <div class="container">
        <div class="item">Box-1</div>
        <div class="item">Box-2</div>
        <div class="item">Box-3</div>
        <div class="item">Box-4</div>
    </div>
</body>
</html>
```

Hasil



Kesimpulan

Flex direction berfungsi ketika kita ingin mengatur tataletak item yang berada didalam container.

FLEX WRAP

Penjelasan

Flex wrap dalam flexbox untuk mengontrol apakah item-item flexbox dalam sebuah container akan melintasi baris atau tidak ketika ruang yang tersedia tidak cukup untuk menampung semua item tersebut dalam satu baris atau kolom.

Beberapa Property Value nya :

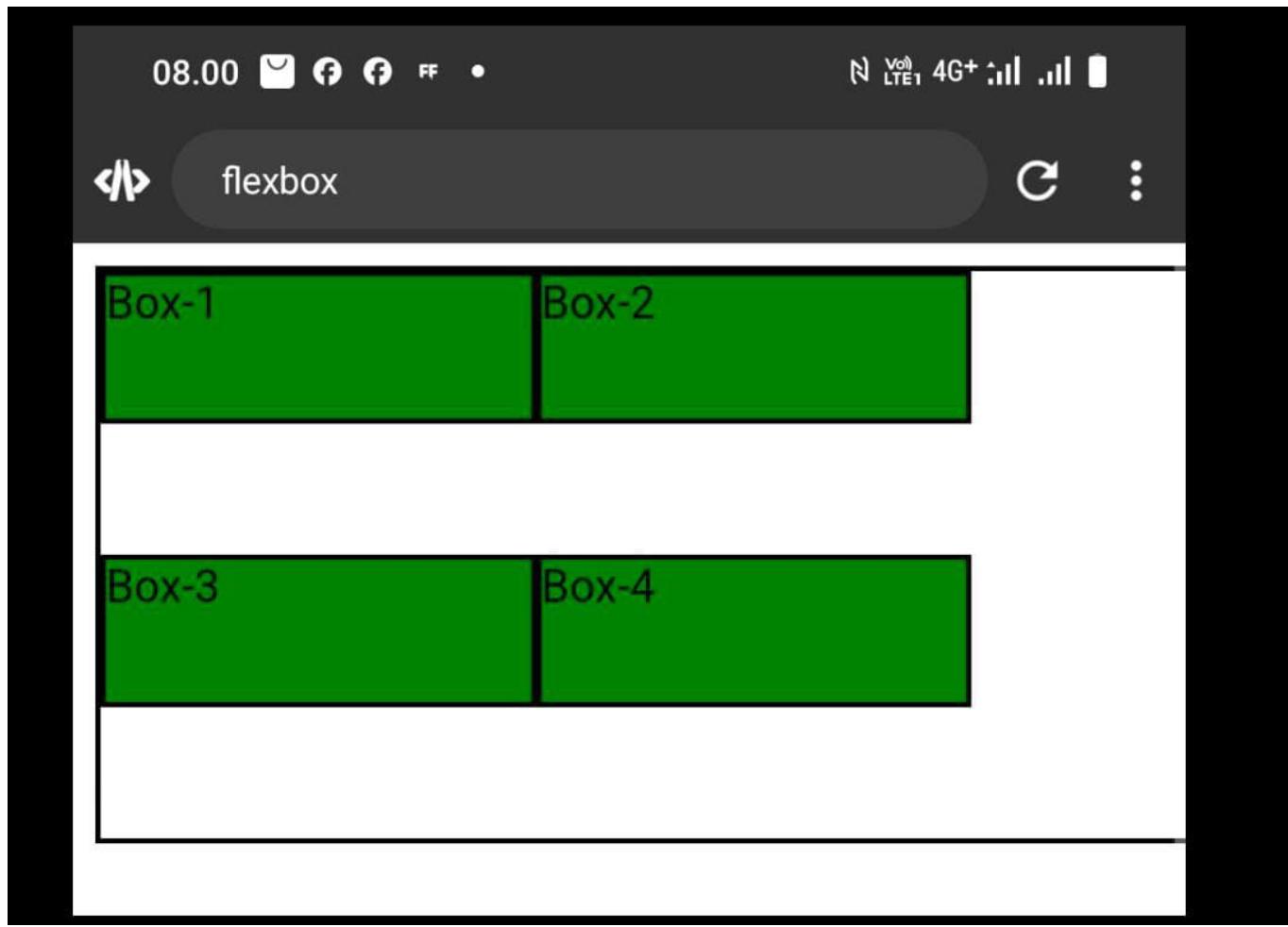
- `wrap` berfungsi ketika dalam 1 baris tidak dapat lagi di tampung maka akan membuat baris baru di bawahnya
- `nowrap` item-item akan berada dalam satu baris atau kolom, meskipun ruang tidak cukup.
- `wrap-reverse` sama dengan `wrap` cuma kebalikannya saja.

Kode Program

```
<html>
  <head>
    <title>flexbox</title>
  </head>
  <style>
    .container {
      display: flex;
      flex-wrap: wrap;
      border: 2px solid black;
      height: 200px;
      width: 400px;
    }

    .item {
      background-color: green;
      border: 2px solid black;
      height: 50px;
      width: 150px;
    }
  </style>
  <body>
    <div class="container">
      <div class="item">Box-1</div>
      <div class="item">Box-2</div>
      <div class="item">Box-3</div>
      <div class="item">Box-4</div>
    </div>
  </body>
</html>
```

Hasil



Kesimpulan

Flex wrap dominan digunakan ketika item yang berada dalam 1 baris sudah tidak cukup di tampung oleh container, maka item selanjutnya akan membuat baris baru.

ALIGN ITEMS

Penjelasan

Align items Property CSS yang digunakan untuk mengatur tataletak elemen secara vertikal dalam flex container.

Beberapa contoh Property Value nya :

- `Flex-start` item akan diletakkan di awal container
- `Flex-end` item akan diletakkan di akhir container
- `center` item akan diletakkan di tengah container
- `stretch` item akan meregang mengisi seluruh tinggi container
- `base-line` item akan diletakkan digaris dasar dari Text mereka.

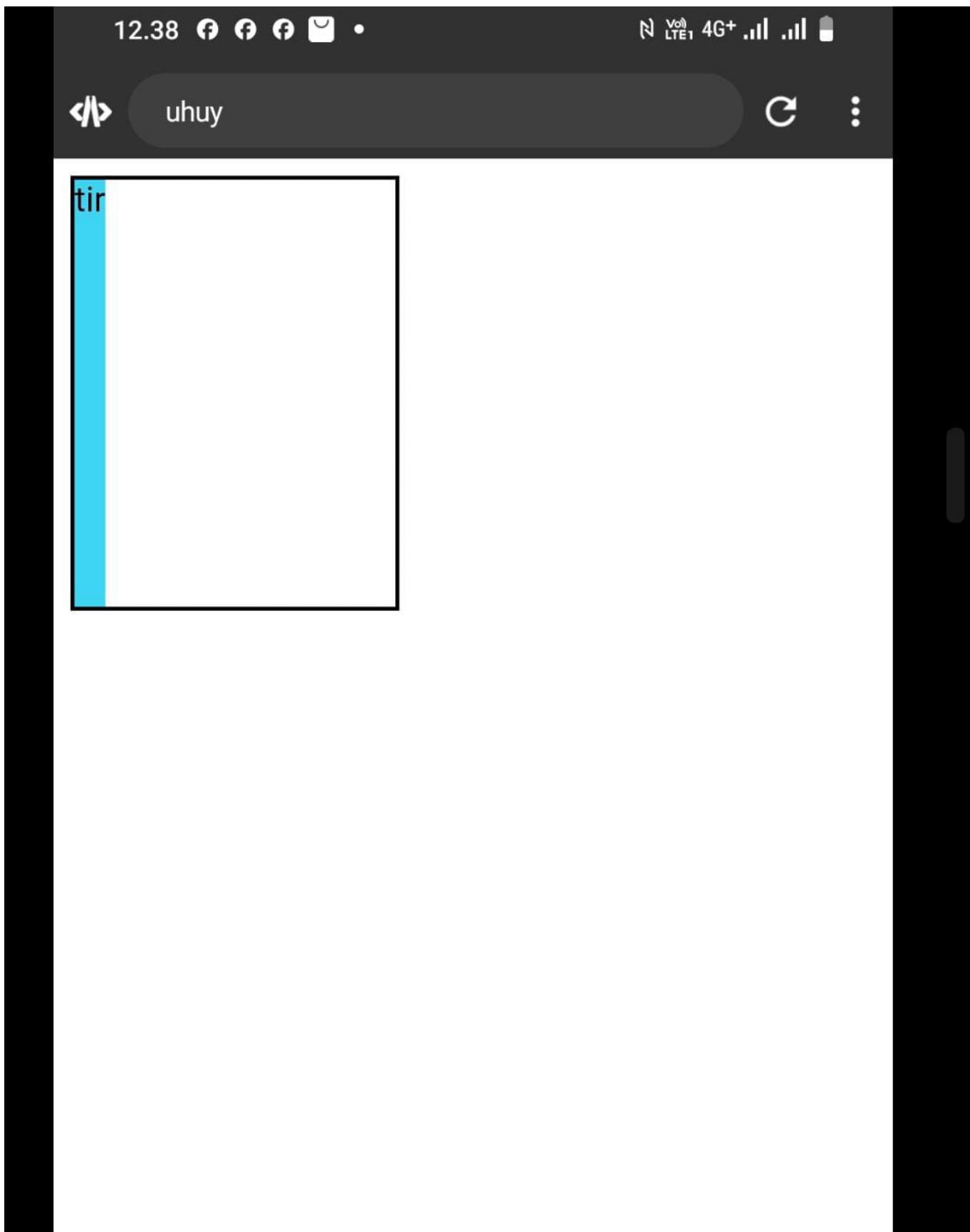
Kode Program

```
<html>
  <head>
    <title>uhuy</title>
  </head>
  <style>
    .container {
      display: flex;
      height: 200px;
      width: 150px;
      border: 2px solid black;
      align-items: stretch;
    }

    .item {
      background-color: skyblue;
    }
  </style>
  <body>
    <div class="container">
      <div class="item">tir</div>
    </div>

  </body>
</html>
```

Hasil



Kesimpulan

Align items digunakan untuk mengatur tataletak suatu elemen secara vertikal

JUSTIFY CONTENT

Penjelasan

justify-content adalah properti CSS yang digunakan untuk mengatur cara penempatan dan penyebaran ruang ekstra dari sebuah flex container secara horizontal.

Beberapa contoh property value nya :

- `flex-start`: Item-item akan diletakkan di awal container.
- `flex-end`: Item-item akan diletakkan di akhir container.
- `center`: Item-item akan diletakkan di tengah container.
- `space-between`: Item-item akan ditempatkan dengan jarak yang sama di antara mereka, tetapi tidak di sisi kanan dan kiri container.
- `space-around`: Item-item akan ditempatkan dengan jarak yang sama di antara mereka, termasuk di sisi kanan dan kiri container.
- `space-evenly`: Item-item akan ditempatkan dengan jarak yang sama di antara mereka, termasuk di sisi kanan dan kiri container, serta jarak yang sama di sekeliling mereka.

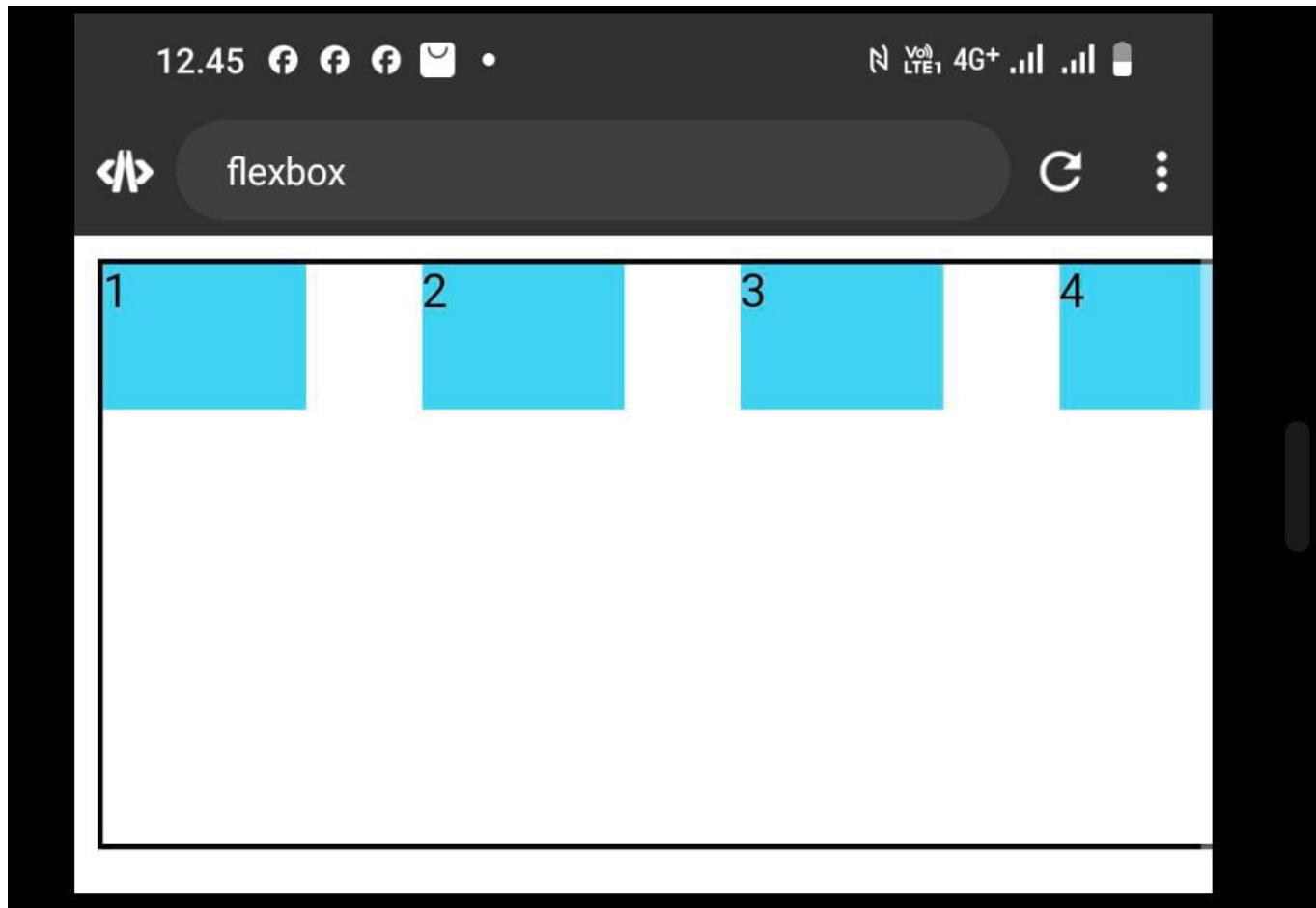
Kode Program

```
<html>
  <head>
    <title>flexbox</title>
  </head>
  <style>
    .container {
      display: flex;
      height: 200px;
      width: 400px;
      border: 2px solid black;
      justify-content: space-between;
    }

    .item {
      background-color: skyblue;
      height: 50px;
      width: 70px;
    }
  </style>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
  </div>
</body>
```

```
<div class="item">3</div>
<div class="item">4</div>
</div>
</body>
</html>
```

Hasil



Kesimpulan

Justify content digunakan untuk mengatur jarak antar elemen secara horizontal atau baris.

ALIGN CONTENT

Penjelasan

Align-content adalah properti CSS yang digunakan untuk mengatur cara konten dalam sebuah flex container disusun secara vertikal ketika ada ruang ekstra di dalam kontainer flex tersebut.

Beberapa Property Value nya :

- **flex-start:** item akan berada di awal container

- `flex-end`: item akan berada di akhir container
- `center`: item akan diletakkan ditengah container
- `space-between`: item diperintahkan diletakkan dengan jarak yang sama, pertama di awal dan terakhir di akhir.
- `space-around`: item diperintahkan diletakkan dengan jarak yang sama di sekitar setiap item.
- `space-evenly`: item diperintahkan untuk diletakkan dengan jarak yang sama di antara dan sekitar setiap item.
- `stretch`: item diperintahkan untuk meregangkan untuk mengisi seluruh kontainer fleksibel.

Kode Program

```

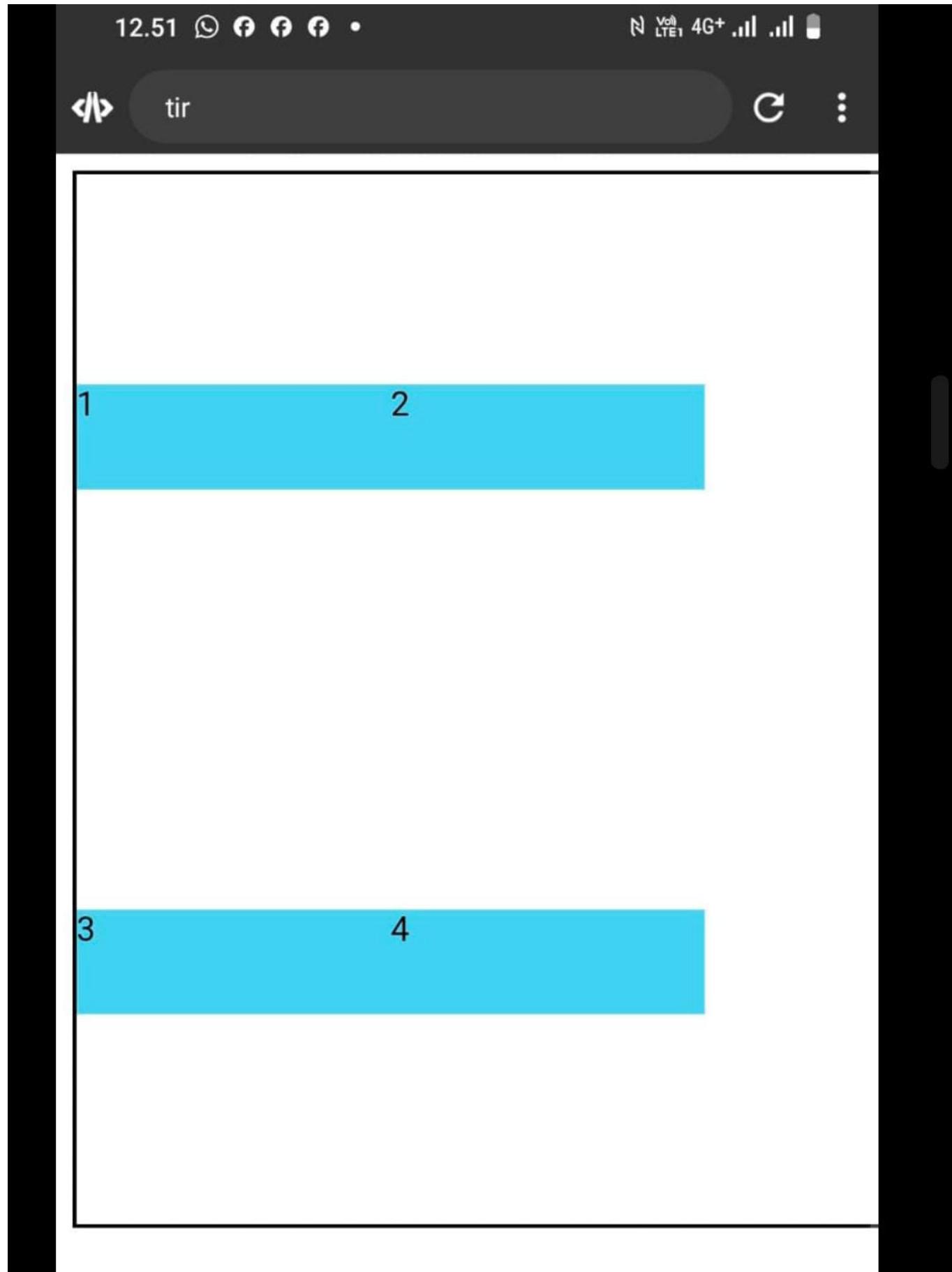
<html>
  <head>
    <title>tir</title>
  </head>
  <style>
    .container {
      display: flex;
      height: 500px;
      width: 400px;
      border: 2px solid black;
      align-content: space-around;
      flex-wrap: wrap;
    }

    .item {
      background-color: skyblue;
      height: 50px;
      width: 150px;
    }
  </style>
  <body>
    <div class="container">
      <div class="item">1</div>
      <div class="item">2</div>
      <div class="item">3</div>
      <div class="item">4</div>
    </div>

  </body>
</html>

```

Hasil



Kesimpulan

Align content digunakan untuk mengatur jarak antar elemen secara vertikal

Flex Item

ORDER

Penjelasan

Order adalah Property CSS dalam flexbox yang digunakan untuk mengatur tata letak flex item dalam sebuah flex container, value yang di berikan kepada order adalah bilangan bulat.

Kode Program

```
<html>
  <head>
    <title>tir</title>
  </head>
  <style>
    .container {
      display: flex;
      flex-direction: row;
      border: 2px solid red;
      height: auto;
      width: 400px;
    }

    .item {
      background-color: skyblue;
      border: 2px solid black;
      height: 50px;
      width: 150px;
      margin: 5px;
    }

    .satu {
      order:4;
    }

  </style>
  <body>
    <div class="container">
      <div class="item satu">1</div>
      <div class="item dua">2</div>
```

```
<div class="item tiga">3</div>
<div class="item empat">4</div>
</div>
</body>
</html>
```

Hasil

13.22

N Vol LTE1



tir



:

2

3

4

1



Kesimpulan

Order digunakan ketika kita ingin mengatur tata letak flex item dalam flex container.

FLEX-GROW

Penjelasan

Flex-Grow adalah Property CSS yang mengatur seberapa besar flex item akan memperluas dirinya dalam flex container relatif terhadap item-item lain di dalam container tersebut. Nilai dari flex-grow menentukan seberapa banyak ruang tambahan yang akan diambil oleh item flex dalam container jika ada ruang kosong yang tersedia setelah item-item lain telah menempati ruang mereka sendiri.

Tidak perlu memberikan width pada itemnya, jika kita berikan maka Flex-Grow tidak tereksekusi.

Code

```
<html>
  <head>
    <title>flexbox</title>
  </head>
  <style>
    .container {
      display: flex;
      flex-direction: row;
      border: 2px solid blue;
      height: auto;
      width: 400px;
    }

    .item {
      display: flex;
      background-color: skyblue;
      border: 2px solid black;
      height: 50px;
      margin: 5px;
    }
  </style>
</html>
```

```
.satu {
  flex-grow: 2;
}

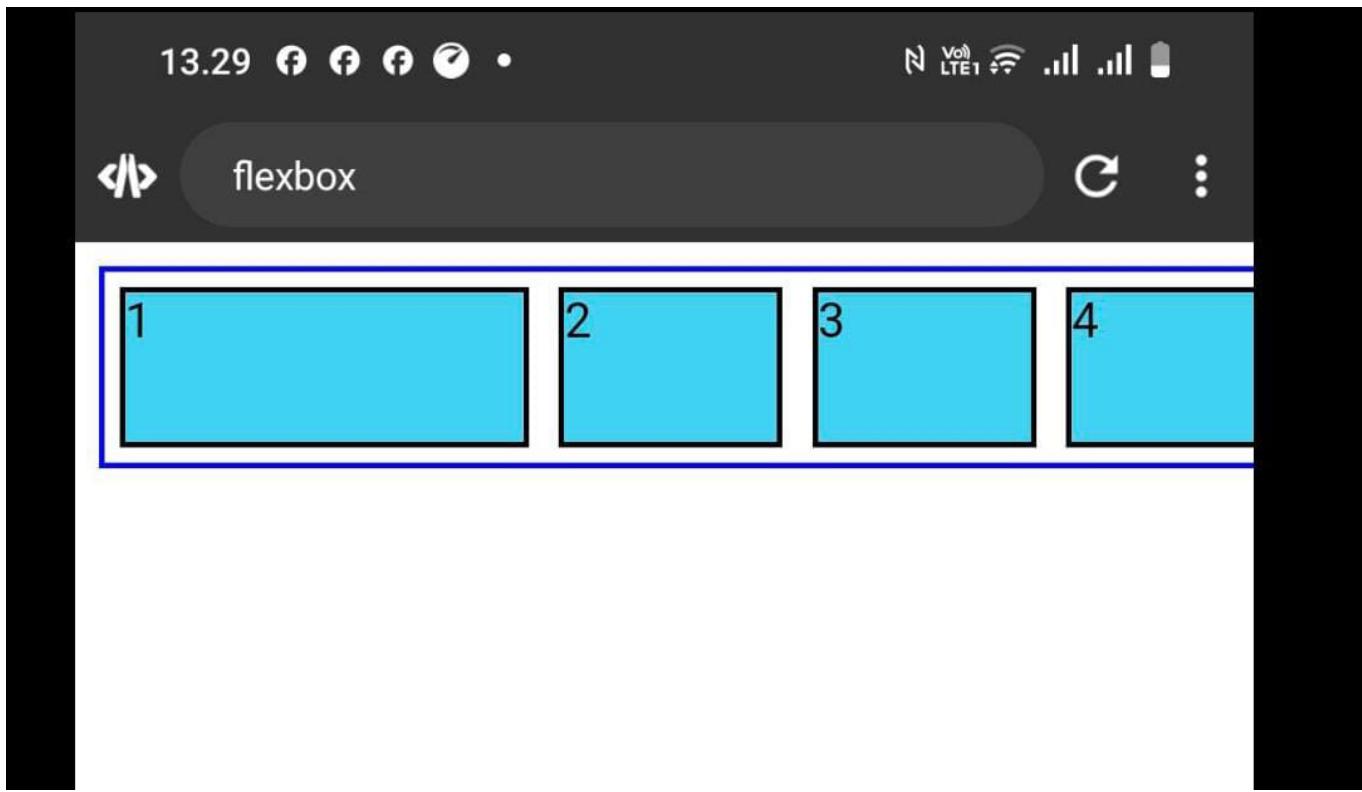
.dua {
  flex-grow: 1;
}

.tiga {
  flex-grow: 1;
}

.empat {
  flex-grow: 1;
}

</style>
<body>
  <div class="container">
    <div class="item satu">1</div>
    <div class="item dua">2</div>
    <div class="item tiga">3</div>
    <div class="item empat">4</div>
  </div>
</body>
</html>
```

Hasil



Kesimpulan

`Flex-Grow` digunakan ketika kita ingin memperluas salah satu flex item dalam sebuah flex container.

FLEX-SHRINK

Penjelasan

`flex-shrink` adalah properti dalam CSS yang digunakan dalam konteks Flexbox untuk mengontrol seberapa besar sebuah elemen fleksibel dapat menyusut jika tidak ada cukup ruang di dalam kontainernya. Properti ini mengatur tingkat penyusutan relatif dari elemen-elemen fleksibel dalam sebuah flex container.

Kode Program

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

<title>tir</title>
<style>
    .flex-container {
        display: flex;
    }
```

```
.flex-item {
    flex: 1;
    padding: 10px;
    border: 1px solid blue;
}

.item1 {
    flex-shrink: 1;
}

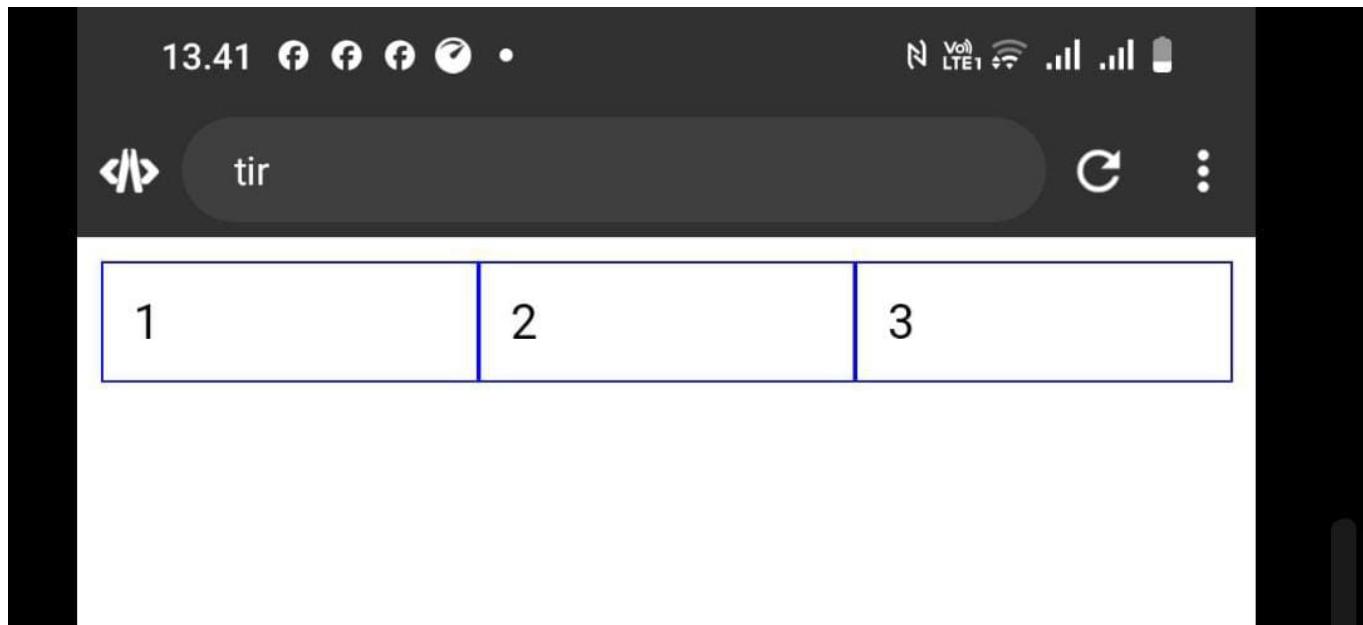
.item2, .item3 {
    flex-shrink: 2;
}

</style>
</head>
<body>

<div class="flex-container">
    <div class="flex-item item1"> 1</div>
    <div class="flex-item item2"> 2</div>
    <div class="flex-item item3"> 3</div>
</div>

</body>
</html>
```

Hasil



Kesimpulan

Flex-shrink mengatur ukuran item ketika ruang tidak cukup.

FLEX-BASIS

Penjelasan

`flex-basis` adalah properti CSS dalam desain tata letak flexbox yang menentukan ukuran awal (basis) dari item fleksibel sebelum penyesuaian ukuran fleksibel terjadi.

Kode Program

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

<title>tir</title>
<style>
    .flex-container {
        display: flex;
    }

    .flex-item {
        flex-grow: 1;
        border: 1px solid skyblue;
        margin: 5px;
    }

    .item1 {
        flex-basis: 50px;
    }

    .item2 {
        flex-basis: 70px;
    }

    .item3 {
        flex-basis: 90px;
    }
</style>
</head>
<body>

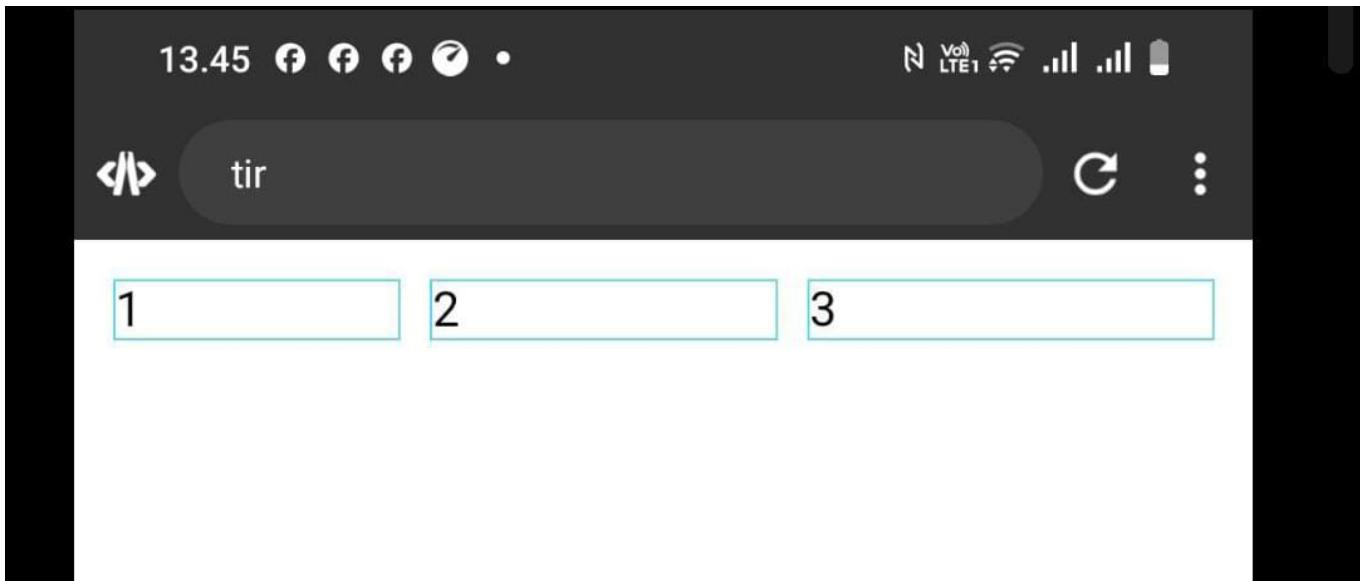
<div class="flex-container">
    <div class="flex-item item1"> 1</div>
    <div class="flex-item item2"> 2</div>

```

```
<div class="flex-item item3"> 3</div>
</div>

</body>
</html>
```

Hasil



Kesimpulan

`flex-basis` menentukan ukuran awal elemen sebelum fleksbox membagikan ruang yang tersisa.

ALIGN-SELF

Penjelasan

align-self adalah properti CSS yang digunakan dalam konteks Flexbox untuk mengontrol penempatan vertikal individu dari elemen flex dalam container. Properti ini mengatur penempatan vertikal elemen tunggal dalam halaman, menggesampingkan nilai align-items yang diterapkan pada container

Nilai yang umum digunakan untuk align-self adalah:

- `flex-start`: Elemen diletakkan di bagian atas container.
- `flex-end`: Elemen diletakkan di bagian bawah container.
- `center`: Elemen diletakkan di tengah container.
- `baseline`: Elemen diletakkan pada garis dasar container.
- `stretch`: Elemen diregangkan untuk mencakup tinggi container.

Kode Program

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

<title>tir</title>
<style>
    .flex-container {
        display: flex;
        height: 200px;
        border: 1px solid black;
    }

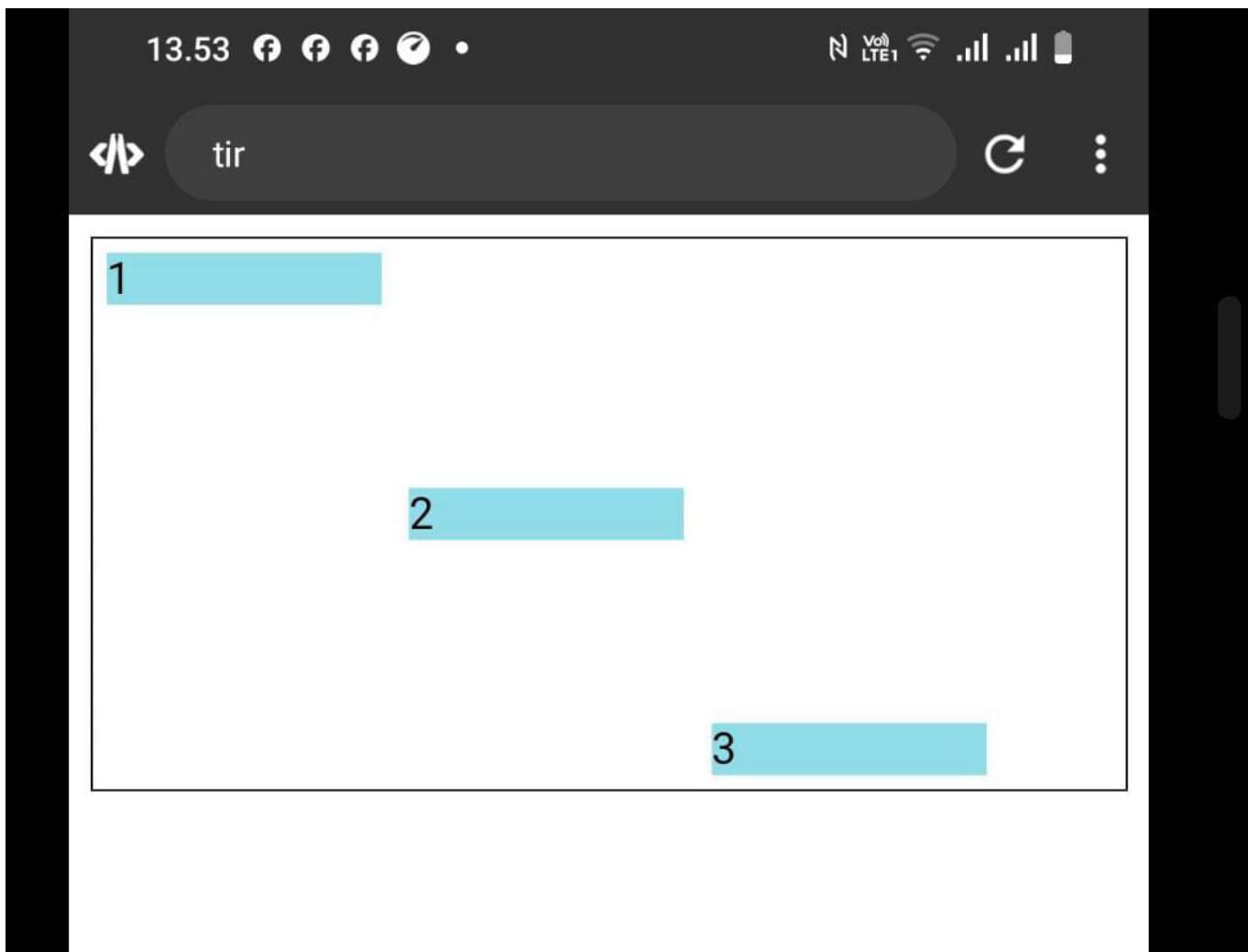
    .flex-item {
        width: 100px;
        margin: 5px;
        background-color: lightblue;
    }

    .item1 {
        align-self: flex-start;
    }

    .item2 {
        align-self: center;
    }

    .item3 {
        align-self: flex-end;
    }
</style>
</head>
<body>
<div class="flex-container">
    <div class="flex-item item1"> 1</div>
    <div class="flex-item item2"> 2</div>
    <div class="flex-item item3"> 3</div>
</div>
</body>
</html>
```

Hasil



Kesimpulan

Align self mengatur penempatan item secara vertikal

FLEX

Penjelasan

Properti flex memungkinkan Anda untuk secara singkat menentukan bagaimana elemen flex akan mengisi ruang dalam flex container. Dengan menggunakan flex, Anda dapat mengatur elemen flex untuk: flex-grow, flex-shrink, dan flex-basis.

Kode Program

```
<!DOCTYPE html>
<head>
<title>Contoh Penggunaan flex</title>
<style>
.flex-container {
```

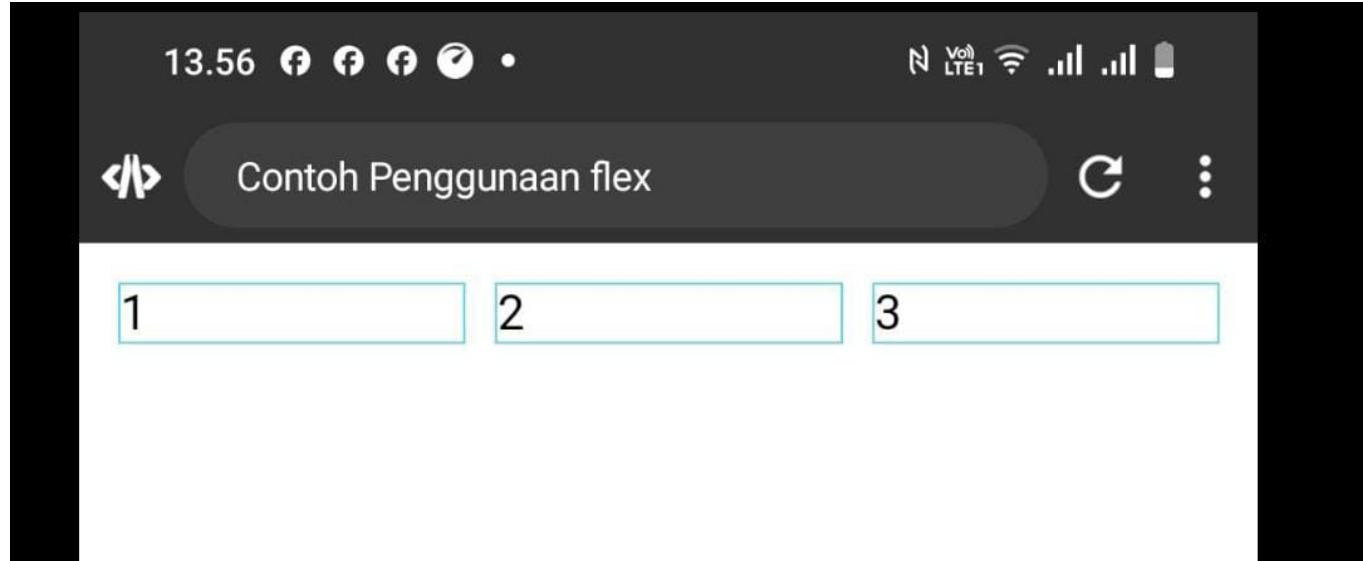
```
        display: flex;
    }

    .flex-item {
        flex: 1;
        border: 1px solid skyblue;
        margin: 5px;
    }
</style>
</head>
<body>

<div class="flex-container">
    <div class="flex-item"> 1</div>
    <div class="flex-item"> 2</div>
    <div class="flex-item"> 3</div>
</div>

</body>
</html>
```

Hasil



Kesimpulan

Flex merupakan gabungan dari `flex grow`, `flex shrink`, `flex basis`

Tantangan Flex box

Code

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS</title>
    <link rel="stylesheet" href="Style.css">
  </head>
  <body>
    <div class="flex-container">
      <div class="containersub">
        <br>
        <span>
          Selamat Datang
        </span>
        <br>
        <span>
          di <b>Web fatir !</b><br>
          <button>Klik disini</button>
        </span>
      </div>
      
    </div>
  </body>
</html>
```

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

.flex-container {
  display: flex;
  align-items: center;
  justify-content: flex-start;
}

span {
  color: white;
  font-size: 30px;
}

button {
  color: orangered;
  background-color: purple;
  border-color: orangered;
  width: 100px;
  height: 50px;
```

```
margin-left: 100px;  
margin-bottom: 1px;  
}  
  
img {  
width: 130px;  
height: 130px;  
border-radius: 100px;  
border-color: black;  
border-style: solid;  
margin-left: 10px;  
}
```

Hasil



CSS



Selamat Datang di Web fatir !

Klik disini



Kesimpulan

1. `body` :

- Latar belakang diatur menjadi warna ungu (`Background-color: Purple`).

2. Kontainer Flex (`flex-container`) :

- Kontainer ini menggunakan Flexbox untuk tata letak elemen-elemen di dalamnya.
- Item dalam kontainer diselaraskan secara vertikal di tengah (`align-items: center`).
- Item juga diselaraskan secara horizontal ke kiri (`justify-content: flex-start`).

3. `span` :

- Warna teks diatur menjadi putih (`color: white`).
- Ukuran font diatur menjadi 30 piksel (`font-size: 30px`).

4. `button` :

- Teks tombol berwarna oranye-merah (`color: orangered`).
- Latar belakang tombol ungu dengan batas oranye-merah.
- Ukuran tombol 100 piksel lebar dan 50 piksel tinggi.
- Margin kiri 100 piksel dan margin bawah 1 piksel.

5. `img` :

- Ukuran gambar 130 piksel untuk lebar dan tinggi.
- Gambar memiliki batas bundar dengan radius 100 piksel.
- Warna batas hitam dan gaya solid.
- Ada jarak antara gambar dan elemen lainnya dengan margin kiri 10 piksel.

Position

Position relative

Penjelasan

position: relative adalah properti CSS yang digunakan untuk menetapkan posisi elemen pada halaman web relatif terhadap posisinya sendiri.

Kode program

```
.box { position: relative; top: 10px; left: 10px; width: 100px; height: 100px; background-color: red; }
```

Hasil

Before

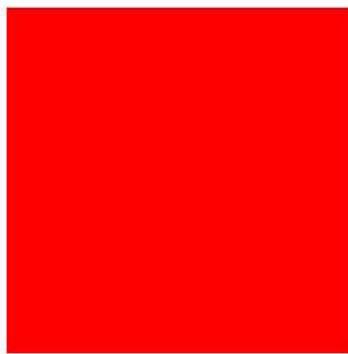
10.39



N VoLTE 4G



cihuy



wkskjwjshsva whshwhhdgag hdhshdhgdb hshdhsha
gsjjsjsjshsh

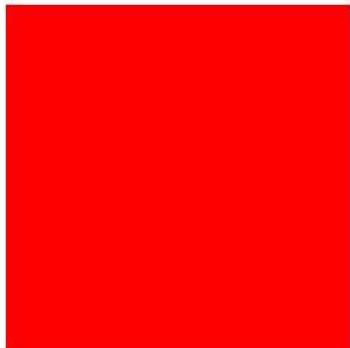
After

10.45

N VoLTE 4G



cihuy



wkskjwjshsva whshwhhdgag hdhshdhgdb hshdhsha
gsjjsjsjshsh

wkskjwjshsva whshwhhdgag hdhshdhgdb hshdhsha
gsjjsjsjshsh

wkskjwjshsva whshwhhdgag hdhshdhgdb hshdhsha
gsjjsjsjshsh

.....

Kesimpulan

memungkinkan pengguna untuk mengatur posisi elemen dengan properti top, right, bottom, atau left

Position absolute

Penjelasan

position: absolute akan dikeluarkan dari normal flow, yang berarti elemen tersebut tidak akan memengaruhi posisi elemen lain di halaman.

Kode program

```
.box { position: absolute; top: 10px; left: 10px; width: 100px; height: 100px;  
background-color: red; }
```

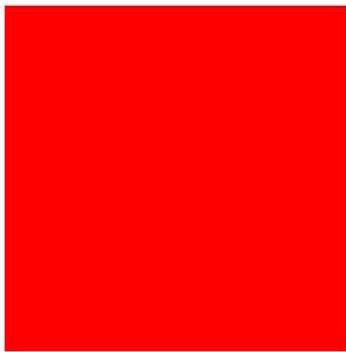
Hasil

10.39

N VoLTE 4G



cihuy



wkskjwjshsva whshwhhdgag hdhshdhgdb hshdhsha
gsjjsjsjshsh

Kesimpulan

hanya berpengaruh pada elemen yang diatur, tidak akan berpengaruh pada posisi elemen lain.

Position fixed

Penjelasan

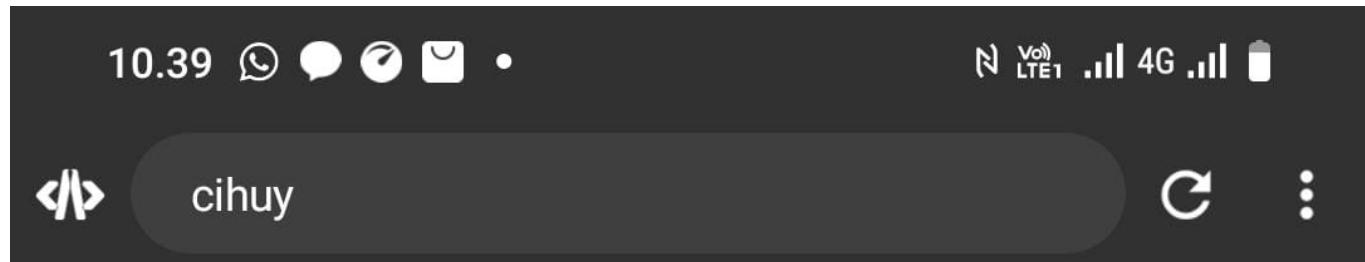
`position: fixed` adalah properti CSS yang digunakan untuk menetapkan posisi elemen pada halaman web tetap dalam posisi tertentu di layar

Kode program

```
.box {  
    position: fixed;  
    width: 100px;  
    height: 100px;  
    background-color: red;  
}
```

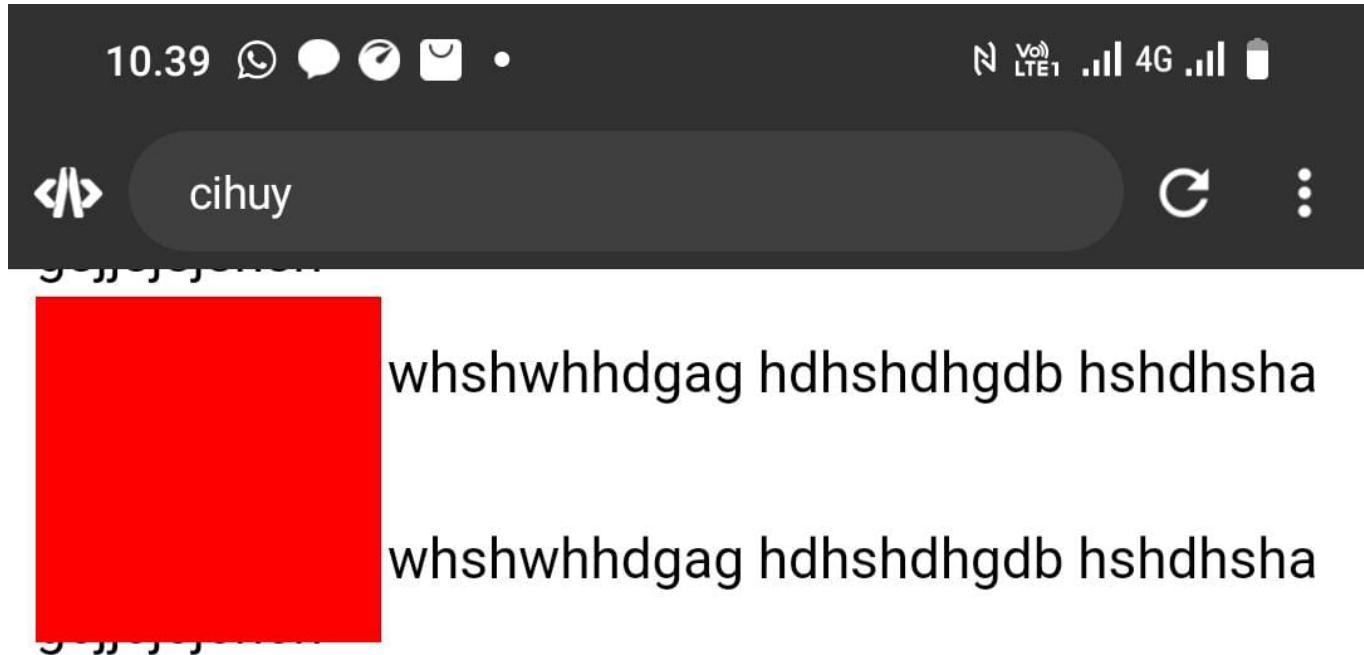
Hasil

Before



wkskjwjshsva whshwhhdgag hdhshdhgdb hshdhsha
gsjjsjsjshsh

After



Kesimpulan

Untuk menetapkan posisi suatu elemen

Position sticky

Penjelasan

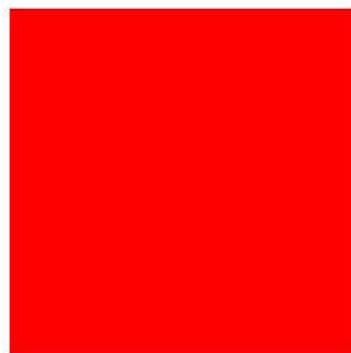
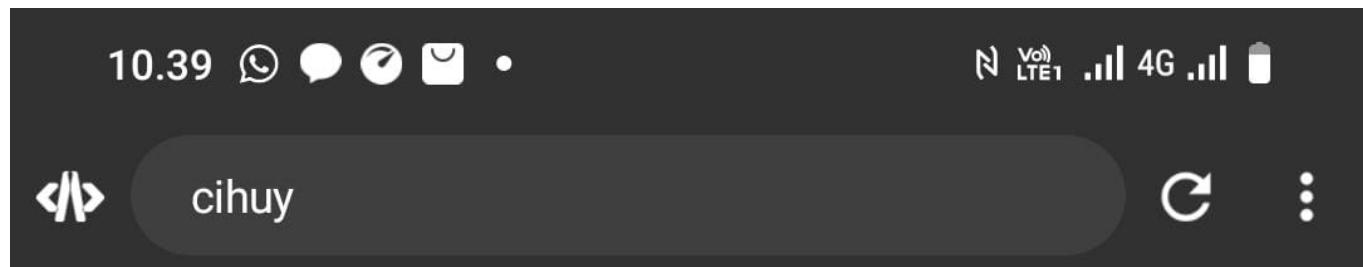
Posisi sticky adalah cara efektif untuk mengubah posisi elemen dengan kecil atau sedikit, seperti menyesuaikan posisi elemen dengan kursor atau mengubah posisi elemen dalam layout.

Kode program

```
.box {  
  position: sticky;  
  top: 10px;  
  width: 100px;  
  height: 100px;  
  background-color: red;  
}
```

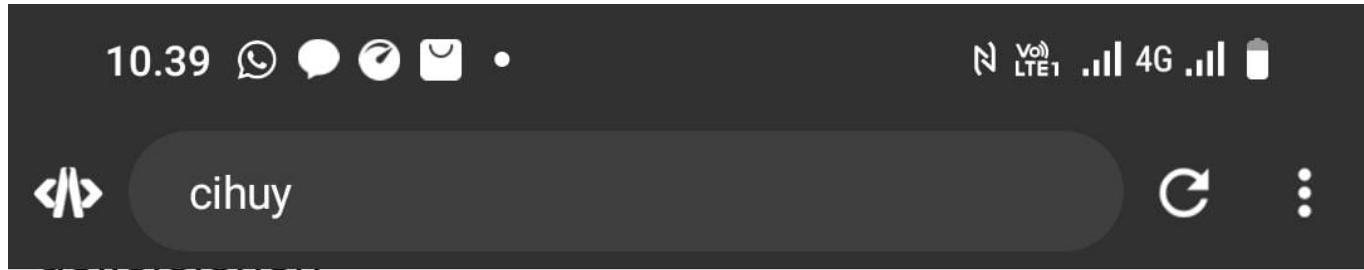
Hasil

Before



wkskjwjshsva whshwhhdgag hdhshdhgdb hshdhsha
gsjjsjsjshsh

After



whshwhhdgag hdhshdhgdb hshdhsha

whshwhhdgag hdhshdhgdb hshdhsha

wkskjwjshsva whshwhhdgag hdhshdhgdb hshdhsha
gsjjsjsjshsh

wkskjwjshsva whshwhhdgag hdhshdhgdb hshdhsha
gsjjsjsjshsh

wkskjwjshsva whshwhhdgag hdhshdhgdb hshdhsha
asiisisishsh

Kesimpulan

position: sticky akan bergulir seperti normal hingga mencapai titik tertentu

Tantangan Position

Code

```
<!DOCTYPE html>
<html>
<head>
    <title>POSITION</title>
    <link rel="stylesheet" href="tantangan_position.css">
```

```

</head>
<body>
    <div class="container">
        <div class="box">
            
            <p> </p>
            <p class="item box-2">Saturday, April 27, 2024</p>
            <h1 class="item box-3">The standard chunk of <br>Lorem Ipsum</h1>
            <p class="item box-4">Sed posuere consectetur est at lobortis.
            <br>Aenean eu leo quam</p>
            <p class="box-5"><b>Read more </b> </p>
        </div>
    </div>
</body>
</html>

```

```

.container {
    position: static;
    display: flex;
    flex-direction: column;
    width: 100%;
    height: 580px;
    background-color: rgba(122, 122, 247, 0.628);
}

.box {
    width: 250px;
    height: 350px;
    background-color: white;
    align-items: center;
    align-content: center;
    border-radius: 10px;
    align-self: center;
    margin-top: 150px;
    margin-bottom: 200px;
}

.item {
    width: 100%;
    color: black;
}

.box-1 {
    height: 225px;
    width: 100%;
}

```

```
border-radius: 10px 10px 0px 0px;

}

.box-2 {
    font-size: 10px;
    margin-left: 20px;
    margin-top: 20px;
    padding-top: 10px;
    font-family: Arial, Helvetica, sans-serif;
}

.box-3 {
    font-size: 18px;
    margin-left: 20px;
    margin-top: 10px;
    padding-top: 10px;
    font-family: Arial, Helvetica, sans-serif;
}

.box-4 {
    font-size: 11px;
    font-family: Arial, Helvetica, sans-serif;
    margin-left: 20px;
}

.box-5 {
    background-color: rgb(193, 193, 193);
    padding-left: 20px;
    padding-bottom: 10px;
    padding-top: 10px;
    margin-bottom: 90px ;
    font-family: Arial, Helvetica, sans-serif;
    border-radius: 0px 0px 10px 10px;
    font-size: small;
}

.item-1 {
    width: 10px;
    padding-left: 130px;
}

.icon {
    background-color: skyblue;
    position: relative;
    left: 200px;
```

```
bottom: 35px;  
width: 30px;  
height: 30px;  
border-radius: 1500px;  
}
```

Hasil



POSITION



Saturday, April 27, 2024

The standard chunk of Lorem Ipsum

Sed posuere consectetur est at lobortis.
Aenean eu leo quam

[Read more](#)





Analisis

1. container

- **Position : static** adalah posisi default di CSS, yang berarti elemen akan muncul sesuai dengan alur dokumen normal.
- **Display : flex** mengaktifkan model layout flexbox, memungkinkan elemen anak untuk diatur dalam arah tertentu dan memberikan fleksibilitas dalam tata letak.
- **Flex-direction: column** berarti elemen anak akan disusun secara vertikal.
- **Width & Height: 100%** untuk lebar dan **580px** untuk tinggi, menunjukkan bahwa elemen ini mengambil seluruh lebar konten, tetapi memiliki tinggi tetap.
- **Background-color:** Warna latar belakang diberi efek transparan dengan nilai RGBA, memberi efek transparansi (0.628).

2. box

- **Width & Height: 250px** lebar dan **350px** tinggi, ukuran ini menentukan ukuran dasar dari elemen **box**.
- **Alignment: align-items** dan **align-content** keduanya berpusat pada tengah, yang berarti elemen dalam box diatur secara horizontal dan vertikal ke tengah.
- **Border-radius:** Memberikan sudut membulat pada box.
- **Margin: margin-top** dan **margin-bottom** memberikan ruang di bagian atas dan bawah **box**, menempatkannya di tengah secara vertikal dalam **container**.

3. item

- **Width & Color: 100%** lebar menunjukkan bahwa elemen ini akan mengambil seluruh lebar kontennya. Warna teks adalah hitam.

4. box-1 hingga .box-5

- Setiap elemen ini memiliki gaya spesifik:
- `box-1` : Memiliki tinggi dan lebar penuh, dan sudut atas membulat.
- `box-2, box-3, box-4` : Berbeda dalam ukuran font dan margin, menunjukkan teks dengan ukuran dan posisi yang bervariasi.
- `box-5`: Bagian bawah dari `box`, dengan latar belakang abu-abu, radius bawah membulat, dan padding untuk ruang dalam.

5. icon

- `Position`: Menggunakan `position: relative` untuk menggeser elemen relatif terhadap posisi default-nya.
- `Background-color`: Warna langit biru dengan radius bulat besar untuk tampilan bundar.
- `Left & Bottom`: Mengatur posisi icon relatif terhadap sudut kanan bawah box.

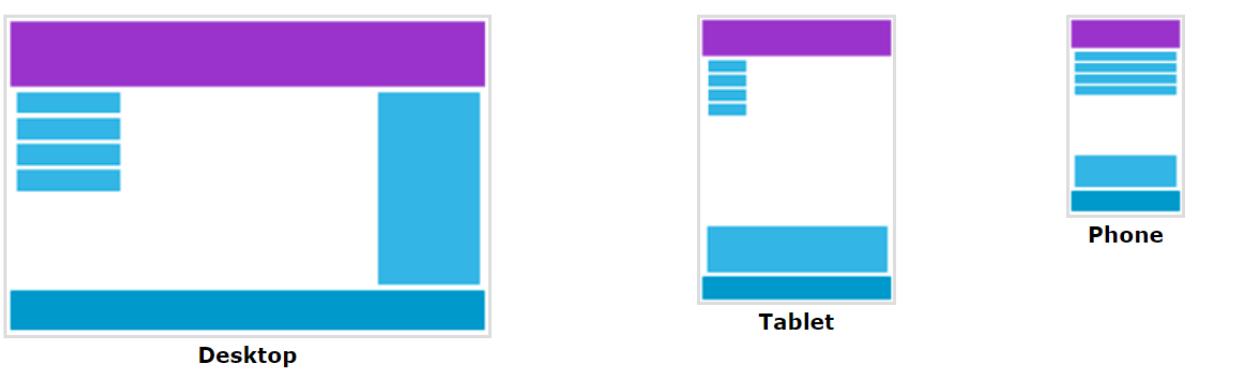
Kesimpulan

elemen utama (`container`), kotak konten (`box`), dan berbagai bagian dalam box (seperti `box-1` hingga `box-5`). Flexbox digunakan untuk pengaturan tata letak, dan elemen-elemen memiliki radius sudut yang memberikan kesan visual yang lebih lembut. Atribut posisi dan margin membantu menempatkan elemen di tempat yang diinginkan dalam konteks tata letak halaman atau aplikasi web.

Pengenalan Responsive Web Design

Responsive web design atau desain web responsif adalah sebuah teknik atau metode bagi web designer untuk membuat suatu layout website yang dapat menyesuaikan diri sesuai dengan ukuran layar pengguna. Baik dari ukuran huruf, user interface, gambar dan tata letak akan menyesuaikan dengan lebar layar dan resolusi device yang digunakan.

Selain dapat memengaruhi apa yang ditampilkan pada perangkat tertentu, ini juga untuk meningkatkan kenyamanan pengguna dalam mengunjungi suatu website.



Mengapa perlu CSS Responsive?

Pernahkah Anda mengunjungi situs web di smartphone Anda dan halaman webnya berantakan atau teksnya terlalu kecil untuk dibaca? Ini artinya situs web tersebut belum responsif. Dengan memakai css responsif maka web kita memungkinkan tampil optimal di berbagai perangkat, mulai dari desktop, laptop, tablet, hingga smartphone. Berikut alasan beberapa alasan mengapa css responsif penting:

- **Tampilan yang Bagus di Semua Perangkat**

CSS responsif memungkinkan tampilan situs web beradaptasi dengan baik terhadap berbagai perangkat dan ukuran layar, sehingga pengguna akan memiliki pengalaman yang konsisten dan optimal, baik mereka mengakses situs melalui desktop, tablet, atau ponsel.

- **Mudah Diakses**

Situs yang responsif bisa diakses dengan mudah oleh lebih banyak orang, termasuk yang menggunakan ponsel untuk browsing.

- **Interaksi yang Lebih Baik**

Situs yang responsif cenderung membuat orang merasa lebih nyaman dan lebih mudah berinteraksi. Ini bisa meningkatkan kemungkinan orang melakukan hal-hal seperti membeli produk atau menghubungi Anda.

Teknik Dasar CSS Responsif

Konsep Dasar

- **Grid System:** Menggunakan grid system dalam CSS untuk mempermudah penataan elemen pada halaman web.
- **Flexbox:** Memahami penggunaan flexbox untuk menata elemen secara responsif.

Media Query

Media query adalah fitur CSS yang memungkinkan Anda menargetkan perangkat dan ukuran layar yang berbeda dengan aturan CSS yang spesifik. Hal ini sangat penting untuk membuat website yang responsif, yaitu website yang dapat menyesuaikan tata letak dan desainnya secara otomatis agar terlihat optimal di semua perangkat, mulai dari desktop, laptop, tablet, hingga smartphone.

Cara Kerja Media Query

Media query menggunakan aturan `@media` untuk menargetkan perangkat dan ukuran layar yang berbeda. Aturan ini terdiri dari dua bagian:

- **Media type:** Menentukan jenis perangkat yang ingin Anda targetkan, seperti `screen` (untuk desktop, laptop, tablet, dan smartphone) atau `print` (untuk printer).
- **Media feature:** Menentukan karakteristik perangkat yang ingin Anda targetkan, seperti `max-width` (lebar layar maksimum), `min-width` (lebar layar minimum), `device-width` (lebar perangkat), dan `orientation` (orientasi perangkat).

Breakpoints

Breakpoints adalah titik-titik di mana tata letak halaman web akan berubah. Breakpoints biasanya ditentukan berdasarkan lebar layar perangkat.

- **320px:** Lebar minimum untuk smartphone.
- **768px:** Lebar minimum untuk tablet.
- **1024px:** Lebar minimum untuk laptop.
- **1200px:** Lebar minimum untuk desktop.

```
@media screen and (max-width: 768px) {  
    /* Aturan CSS untuk perangkat mobile */  
    body {  
        font-size: 16px;  
    }  
    .container {  
        width: 100%;  
    }  
}
```

Unit Relatif

Unit relatif adalah unit yang digunakan dalam CSS untuk menentukan ukuran elemen secara proporsional. Berbeda dengan unit absolut seperti piksel dan sentimeter, unit relatif tidak memiliki nilai tetap dan dapat berubah tergantung pada konteksnya.

Keuntungan

- **Fleksibilitas:** Unit relatif memungkinkan Anda membuat website yang responsif dan dapat menyesuaikan dengan berbagai ukuran layar dan perangkat.
- **Kemudahan Penggunaan:** Unit relatif lebih mudah digunakan untuk menentukan ukuran elemen yang proporsional dan konsisten di seluruh website.
- **Pemeliharaan:** Website yang menggunakan unit relatif lebih mudah dipelihara dan diubah karena ukuran elemen akan menyesuaikan secara otomatis.

Jenis-jenis Unit Relatif:

- **em**: Ukuran font elemen induk
- **rem**: Ukuran font elemen root (biasanya elemen `<html>`)
- **vw**: Lebar viewport (layar perangkat pengguna)
- **vh**: Tinggi viewport (layar perangkat pengguna)
- **vmin**: Nilai terkecil antara vw dan vh
- **vmax**: Nilai terbesar antara vw dan vh

Viewport Meta Tag

Viewport meta tag adalah sebuah elemen HTML yang memberitahu browser web bagaimana cara menampilkan website Anda pada perangkat pengguna. Tag ini sangat penting untuk website responsif, yaitu website yang dapat menyesuaikan tata letak dan desainnya secara otomatis agar terlihat optimal di semua perangkat, mulai dari desktop, laptop, tablet, hingga smartphone.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- **width=device-width**: Menentukan lebar awal website agar sama dengan lebar perangkat pengguna.
- **initial-scale=1.0**: Mengatur skala awal website menjadi 100%.

Best Practices

Gunakan Mobile-First

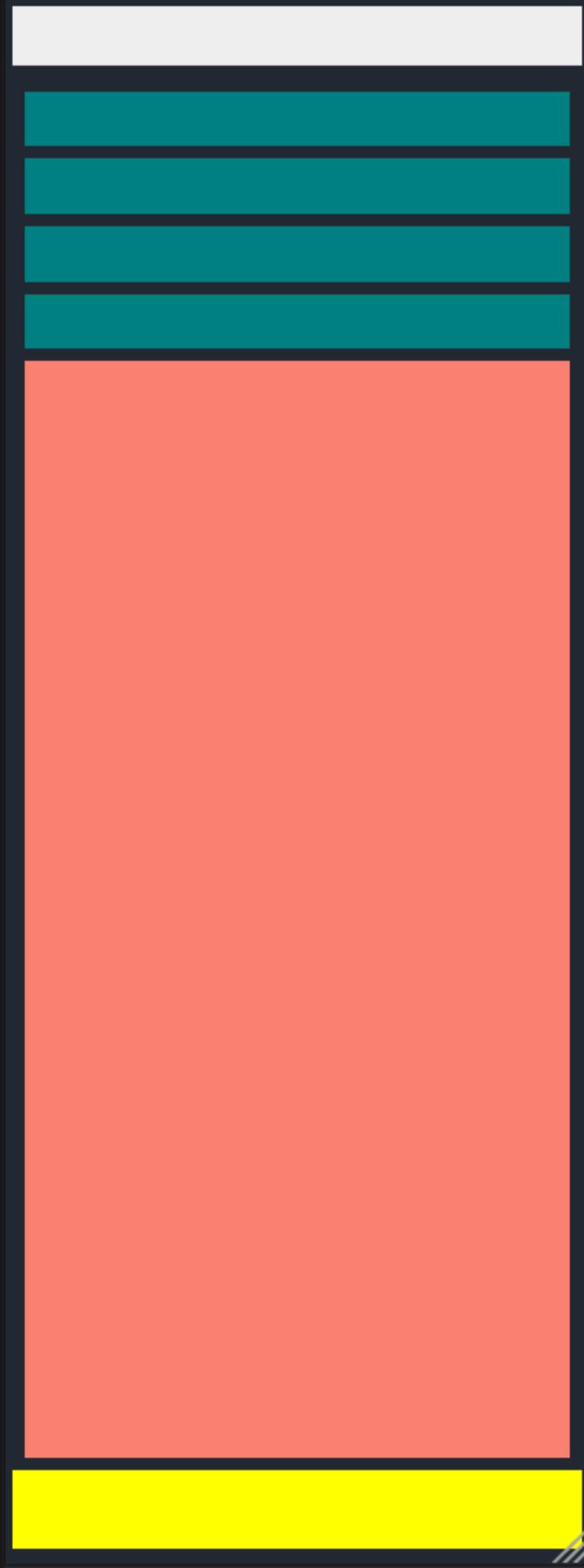
Saat mendesain website, mulailah dengan mempertimbangkan tampilan website pada perangkat mobile. Hal ini akan membantu Anda membuat website yang lebih ringkas dan mudah digunakan di semua perangkat.

Implementasi CSS Responsif

Sebelum Responsif



Sesudah Responsif



Kode

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="style.css" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>CSS Responsif</title>
  </head>
  <body>
    <header></header>
    <section class="container">
      <div class="list">
        <div class="item"></div>
        <div class="item"></div>
        <div class="item"></div>
        <div class="item"></div>
      </div>
      <aside></aside>
    </section>
    <footer></footer>
  </body>
</html>
```

CSS

```
body {
  background-color: #222831;
  color: white;
}

header {
  width: 100%;
  height: 77px;
  background-color: #EEEEEE;
}

.container {
  display: flex;
  justify-content: space-between;
}

.item {
```

```

width: 250px;
height: 70px;
background-color: teal;
margin: 1em 0;
}

aside {
  width: 300px;
  height: 70vh;
  background-color: salmon;
}

footer {
  width: 100%;
  height: 100px;
  background-color: yellow;
}

@media screen and (max-width: 768px) {
  .container {
    padding: 1rem;
    flex-direction: column;
  }
  .item {
    width: 100%;
  }
  aside {
    width: 100%;
  }
}

```

Penjelasan Singkat

```

body {
  background-color: #222831;
  color: white;
}

```

- body : Ini adalah elemen HTML yang menandakan seluruh konten halaman web.
- background-color: #222831 : Ini mengatur warna latar belakang dari elemen "body" menjadi warna gelap dengan kode warna #222831 (hampir hitam).
- color: white : Ini mengatur warna teks di dalam elemen "body" menjadi putih.

```
header {  
    width: 100%;  
    height: 77px;  
    background-color: #EEEEEE;  
}
```

- `header` : elemen HTML yang menandakan bagian atas halaman web.
- `width: 100%` : membuat lebar elemen "header" sejajar dengan lebar area yang mengandungnya, yang mungkin adalah lebar jendela browser atau lebar elemen induk lainnya. Dengan memberikan nilai 100%, elemen "header" akan merentang ke seluruh lebar area tersebut.
- `height: 77px` : menetapkan tinggi elemen "header" menjadi 77 piksel. Jadi, meskipun lebarnya mengikuti lebar area induknya (100%), tingginya tetap diberikan secara spesifik dengan 77 piksel. Ini bertujuan untuk memastikan bahwa tinggi header tetap konsisten meskipun konten di dalamnya berubah.
- `background-color: #EEEEEE` : Ini mengatur warna latar belakang dari elemen "header" menjadi abu-abu muda dengan kode warna # EEEE.

```
.container {  
    display: flex;  
    justify-content: space-between;  
}
```

- `.container` : Ini adalah kelas CSS yang digunakan untuk menargetkan elemen HTML dengan kelas "container".
- `display: flex` : membuat elemen dengan kelas "container" menggunakan model tata letak flexbox.
- `justify-content: space-between` : Ini mengatur jarak antara elemen-elemen di dalam flexbox container agar rata di sepanjang sumbu utama (biasanya sumbu horizontal), dengan menjaga jarak yang sama antara elemen-elemen dan meletakkan mereka pada posisi ujung container.

```
.item {  
    width: 250px;  
    height: 70px;  
    background-color: teal;  
    margin: 1em 0;  
}
```

- `.item` : Ini adalah sebuah selektor CSS yang memilih elemen dengan kelas "item".

- `width: 250px` : Ini mengatur lebar elemen menjadi 250 piksel.
- `height: 70px` : Ini mengatur tinggi elemen menjadi 70 piksel.
- `background-color: teal` : Ini mengatur warna latar belakang elemen menjadi warna teal.
- `margin: 1em 0` : Ini mengatur jarak antara elemen dengan elemen lainnya, dengan 1 em (ukuran relatif terhadap font-size) pada bagian atas dan bawah, dan 0 pada bagian kanan dan kiri.

```
aside {
  width: 300px;
  height: 70vh;
  background-color: salmon;
}
```

- `aside` : Ini adalah sebuah elemen HTML yang biasanya digunakan untuk menunjukkan konten tambahan di samping konten utama.
- `width: 300px` : Ini mengatur lebar elemen "aside" menjadi 300 piksel.
- `height: 70vh` : Ini mengatur tinggi elemen "aside" menjadi 70% dari tinggi viewport.
- `background-color: salmon` : Ini mengatur warna latar belakang elemen "aside" menjadi warna salmon.

```
footer {
  width: 100%;
  height: 100px;
  background-color: yellow;
}
```

- `footer` : Ini adalah sebuah elemen HTML yang biasanya digunakan untuk menampilkan informasi tambahan di bagian bawah halaman web.
- `width: 100%` : Ini mengatur lebar elemen "footer" agar mengisi seluruh lebar dari parent element (biasanya body atau container lainnya).
- `height: 100px` : Ini mengatur tinggi elemen "footer" menjadi 100 piksel.
- `background-color: yellow` : Ini mengatur warna latar belakang elemen "footer" menjadi warna kuning.

```
@media screen and (max-width: 768px) {
  .container {
    padding: 1rem;
    flex-direction: column;
}}
```

```
.item {  
    width: 100%;  
}  
aside {  
    width: 100%;  
}  
}
```

- @media screen and (max-width: 768px) : Ini adalah aturan media query yang diterapkan ketika lebar layar kurang dari atau sama dengan 768 piksel.
- .container : Mengatur tampilan untuk elemen dengan kelas "container" ketika media query aktif.
 - padding: 1rem; : Memberikan padding sebesar 1 rem di sekitar elemen "container".
 - flex-direction: column; : Mengubah arah penataan elemen menjadi vertikal (kolom), sehingga elemen akan ditata dari atas ke bawah.
- .item : Mengatur tampilan untuk elemen dengan kelas "item" ketika media query aktif.
 - width: 100%; : Memberikan lebar sebesar 100% dari parent element, sehingga elemen akan mengisi lebar penuh.
- aside : Mengatur tampilan untuk elemen "aside" ketika media query aktif.
 - width: 100%; : Memberikan lebar sebesar 100% dari parent element, sehingga elemen "aside" akan mengisi lebar penuh.

Pengenalan Bootstrap

Apa Itu Bootstrap?

Bootstrap adalah salah satu dari banyak framework front-end yang ada di web development. Framework-front end menyediakan serangkaian alat dan gaya bawaan untuk mempercepat proses pengembangan web dengan menyediakan komponen-komponen UI siap pakai dan sistem grid yang responsif. Bootstrap khususnya, terkenal dengan kemampuannya dalam menciptakan tata letak yang responsif dan komponen-komponen UI yang seragam.

Instalasi Bootstrap

Cara instalasi Bootstrap secara online / CDN.

sekarang kita akan menjelaskan bagaimana cara menginstall bootstrap secara online. Berikut langkah-langkahnya:

1. Masuklah ke website resmi Bootstrap dengan mengunjungi situs web resmi di <https://getbootstrap.com/>.

2. Di laman utama web Bootstrap, cari dan klik menu "Introduction" yang terletak pada sisi sebelah kiri website.
3. Pada bagian "Starter Template" di halaman Introduction, kita akan melihat kode yang dapat kita gunakan untuk menghubungkan website kita yang mau diterapkan Bootstrap.
4. Klik tombol "Copy" untuk menyalin seluruh kode tersebut.
5. Buatlah file baru dengan nama index.html atau nama yang kita inginkan menggunakan teks editor atau editor HTML yang kita biasa pakai.
6. Tempelkan(paste) kode yang telah kita salin dari langkah sebelumnya ke dalam file html yang baru kita buat. dengan menempelkan kode tersebut maka bootstrap kita sudah terhubung namun harus secara online.
7. kita dapat mulai memberikan gaya pada tag-tag atau elemen dalam file HTML tersebut dengan menggunakan kelas-kelas Bootstrap. kita dapat melihatnya pada website resmi Bootstrap untuk mempelajari lebih lanjut tentang kelas-kelas yang tersedia dan cara penggunaannya.
8. Untuk melihat hasilnya, buka file html tadi menggunakan web browser kita.
9. Halaman web yang ditampilkan akan menggunakan Bootstrap untuk gaya dan fungsionalitasnya

Mengunduh dan menginstal Bootstrap secara lokal.

Untuk menginstal Bootstrap secara offline, kita perlu mengunduh file Bootstrap dan menyimpannya di folder proyek kita. Berikut adalah langkah-langkah untuk menginstal Bootstrap secara offline:

1. **Unduh File Bootstrap:** Kunjungi situs web resmi Bootstrap di <https://getbootstrap.com/> dan cari tautan unduhan untuk versi Bootstrap yang diinginkan. Klik atau ikuti instruksi untuk mengunduh file ZIP Bootstrap.
2. **Ekstrak File Bootstrap:** Setelah mengunduh file ZIP Bootstrap, temukan file tersebut di komputer kita dan ekstrak isi file ZIP ke folder proyek kita. kita dapat menggunakan aplikasi pengarsipan file atau ekstraksi bawaan (Archiver, 7zip, dll) pada sistem operasi kita untuk mengekstrak file ZIP . Setelah diekstrak, kita akan memiliki folder Bootstrap yang berisi berkas-berkas Bootstrap yang diperlukan.
3. **Hubungkan Berkas Bootstrap pada Halaman HTML:** Buka file HTML proyek kita menggunakan teks editor atau editor HTML yang biasa kita gunakan. Di dalam tag pada halaman HTML , tambahkan tautan ke berkas CSS Bootstrap dan skrip JavaScript Bootstrap. Gunakan tag link di dalam tag head untuk tautan CSS dan tag script di dalam tag body untuk javascript. Berikut contoh tautan yang umum di bootstrap:

```
<link href="path/to/bootstrap.min.css" rel="stylesheet">
// tautan bootstrap CSS
```

```
<script src="path/to/bootstrap.bundle.min.js"></script>
// tautan bootstrap Javascript
```

dengan menuliskan script diatas, maka kita telah menghubungkan antara file html kita dengan bootstrap yang kita unduh tadi.

4. Gunakan Kelas Bootstrap: Setelah tautan Bootstrap ditambahkan, Kita dapat menggunakan kelas-kelas Bootstrap dalam elemen HTML proyek Kita untuk menerapkan gaya dan fungsionalitas yang disediakan oleh bootstrap kita. Kita dapat masuk ke web resmi Bootstrap untuk mempelajari lebih lanjut tentang kelas-kelas yang tersedia dan cara penggunaannya.

5. Jalankan File HTML: Setelah Kita selesai menghubungkan tautan dan menggunakan kelas Bootstrap, Kita dapat menjalankan halaman HTML kita di web browser Kita untuk melihat hasilnya. Buka file HTML menggunakan web browser Kita dan lihat hasilnya

Komponen-Komponen Bootstrap

Bootstrap menyediakan berbagai komponen yang siap pakai untuk membangun tampilan website yang responsif dan menarik. Berikut adalah beberapa komponen utama yang disediakan oleh Bootstrap:

Grid System

Grid system Bootstrap adalah sistem layout yang responsif dan fleksibel. kita dapat membagi halaman menjadi baris(rows) dan kolom(columns) yang membentuk grid. Grid terdiri dari 12 kolom, yang dapat kita susun sesuai kebutuhan. Dengan menggunakan kelas CSS yang disediakan oleh Bootstrap, kita dapat dengan mudah menentukan berapa banyak kolom yang akan digunakan oleh setiap elemen di halaman web kita. Grid system ini sangat berguna dalam menciptakan tata letak yang responsif dan dapat menyesuaikan diri dengan berbagai ukuran layar

Typography

Komponen typography Bootstrap menyediakan gaya dan kelas CSS yang konsisten untuk tipografi di halaman web kita. kita dapat dengan mudah mengatur ukuran teks, gaya huruf, dan pengaturan lainnya menggunakan kelas-kelas yang telah ditentukan. Ini memastikan bahwa teks di halaman web kita memiliki tampilan yang konsisten dan mudah dibaca di berbagai perangkat.

Button

Komponen tombol Bootstrap memungkinkan kita dengan mudah membuat tombol dengan tampilan yang menarik dan responsif. kita dapat menggunakan kelas-kelas Bootstrap untuk

mengatur berbagai gaya tombol, termasuk ukuran (large, small), warna (default, primary, secondary, dll.), dan variasi lainnya. Tombol-tombol ini dapat digunakan untuk tindakan seperti mengirim formulir, memuat ulang halaman, atau memicu tindakan lainnya di aplikasi web kita.

Forms

Bootstrap menyediakan komponen form yang mudah digunakan untuk membuat form input. Ini termasuk input teks, area teks, kotak centang (checkbox), tombol radio, dropdown, dan lain-lain. Komponen form Bootstrap telah dirancang dengan tampilan yang responsif dan mudah dikustomisasi. kita dapat dengan mudah menambahkan validasi form dan mengatur tampilan form kita dengan menggunakan kelas-kelas Bootstrap yang telah ditentukan.

Navbar

Komponen navbar Bootstrap memungkinkan kita membuat navigasi yang responsif dan mudah dikustomisasi di halaman web. kita dapat menambahkan logo, menu, tombol, dan komponen lainnya ke navbar dengan mudah. Navbar Bootstrap juga menyediakan fitur seperti menu dropdown, navigasi yang terlipat untuk perangkat mobile, dan tata letak yang fleksibel.

Cards

Cards adalah komponen yang digunakan untuk menampilkan informasi dalam format yang terstruktur. Komponen kartu Bootstrap memungkinkan kita untuk membuat kolom dengan gambar, teks, tombol, dan komponen lainnya. cards ini dapat digunakan untuk menampilkan artikel, produk, profil pengguna, atau konten lainnya dengan tampilan yang menarik.

Modal

Komponen modal Bootstrap digunakan untuk menampilkan jendela pop-up yang tumpang tindih dengan konten utama halaman. Modal ini berfungsi untuk menyoroti konten tambahan, pesan, atau form yang membutuhkan fokus pengguna. kita dapat menyesuaikan tampilan modal, mengatur ukuran, menambahkan judul, dan mengatur perilaku saat modal ditampilkan atau ditutup.

Carousel

Carousel adalah komponen Bootstrap yang digunakan untuk membuat tampilan slide gambar atau konten lainnya. kita dapat menambahkan gambar, teks, tombol navigasi, dan indikator slide untuk membuat tampilan yang menarik dan interaktif. Carousel Bootstrap mendukung navigasi otomatis, kontrol manual, dan animasi transisi yang halus.

Icons

Bootstrap sendiri menggunakan ikon dari Font Awesome, yang merupakan kumpulan ikon vektor yang sangat populer dan kaya akan fitur. Kita dapat dengan mudah menambahkan ikon ke elemen seperti tombol, tautan, dan elemen lainnya menggunakan class-class ikon Bootstrap. Ini berfungsi untuk memperindah halaman web kita.

Jumbotron

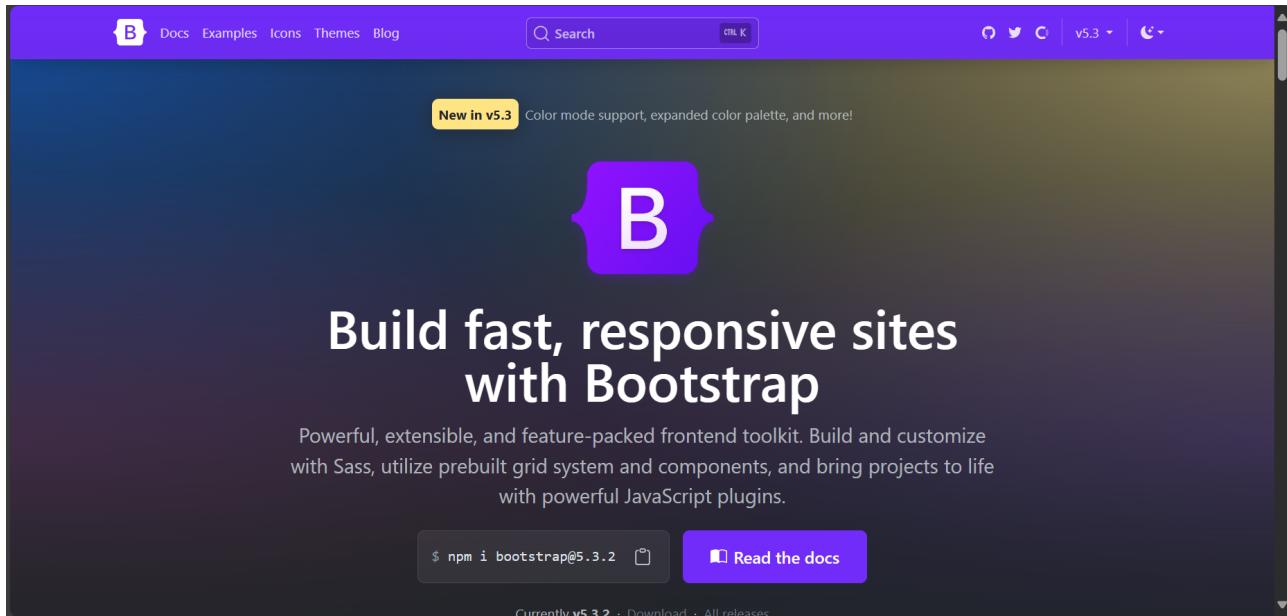
Jumbotron adalah komponen Bootstrap yang digunakan untuk menyoroti konten utama di halaman web. Biasanya ditempatkan di bagian atas halaman dengan judul besar dan terima kasih atas klarifikasinya. Komponen carousel Bootstrap digunakan untuk membuat tampilan slide yang interaktif. Kita dapat menambahkan gambar, teks, dan tombol navigasi ke dalam carousel. Carousel Bootstrap mendukung navigasi otomatis, kontrol manual, dan animasi transisi.

Kita dapat menemukan daftar komponen lengkap dan dokumentasi resmi di situs web Bootstrap (<https://getbootstrap.com/docs/>).

Contoh Penerapan Komponen Bootstrap

Typography

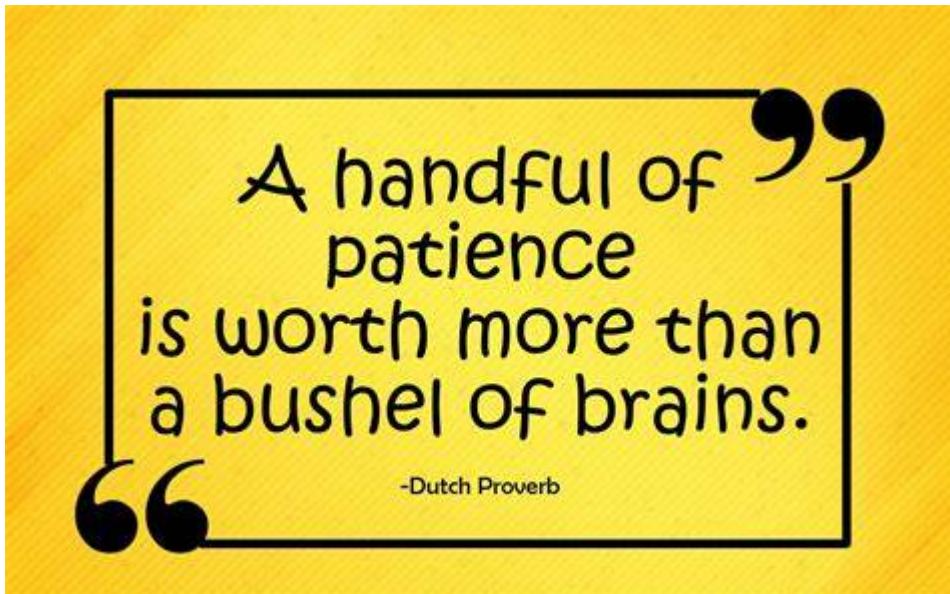
1. Buka situs resmi Bootstrap di getbootstrap.com



2. Klik "Read the docs". Maka kita akan diarahkan pada halaman web yang berisi berbagai macam Typografi (class Bootstrap).

The screenshot shows the Bootstrap documentation website. On the left, there's a sidebar with sections like 'Getting started' (Introduction, Download, Contents, Browsers & devices, JavaScript, Webpack, Parcel, Vite, Accessibility, RFS, RTL, Contribute) and 'Customize' (Overview, Sass, Options, Color, Color modes, Components, CSS variables). The main content area features a large heading 'Get started with Bootstrap'. Below it, a paragraph says 'Bootstrap is a powerful, feature-packed frontend toolkit. Build anything—from prototype to production—in minutes.' There's also an 'Adobe Stock' advertisement for free images. To the right, there's a 'View on GitHub' button and a 'On this page' sidebar with links to 'Quick start', 'CDN links', 'Next steps', 'JS components', 'Important globals', 'HTML5 doctype', 'Viewport meta', 'Box-sizing', 'Reboot', and 'Community'.

3. Misalnya kita akan membuat sebuah teks quotes seperti di bawah ini:



4. Ketikan "Typography" di kolom search lalu enter. Maka kita akan masuk ke halaman berikut:

The screenshot shows the Bootstrap documentation website with the following details:

- Top Bar:** Includes links for Docs, Examples, Icons, Themes, Blog, a search bar, and version v5.3.
- Sidebar (Left):** A navigation menu with sections like Breakpoints, Containers, Grid, Columns, Gutters, Utilities, Z-index, CSS Grid, Content (selected), Typography (selected), Images, Tables, Figures, and Forms (selected). Under Forms, there are sub-links for Overview, Form control, Select, Checks & radios, Range, and Input group.
- Main Content Area:** The title "Typography" is displayed in a large, bold font. Below it is a subtitle: "Documentation and examples for Bootstrap typography, including global settings, headings, body text, lists, and more." To the right of the title is a "View on GitHub" button. A dark callout box contains icons for navigating between pages and a message: "Design and Development tips in your inbox. Every weekday." followed by "ads via Carbon".
- Right Sidebar:** Titled "On this page", it lists various Bootstrap components and utility classes, each with a small icon:
 - Global settings (selected)
 - Headings
 - Customizing headings
 - Display headings
 - Lead
 - Inline text elements
 - Text utilities
 - Abbreviations
 - Blockquotes
 - Naming a source
 - Alignment
 - Lists
 - Unstyled
 - Inline
 - Description list alignment
 - Responsive font sizes
 - CSS
 - Sass variables
 - Sass mixins

5. Carilah Subbab "Alignment":

Alignment

Use text utilities as needed to change the alignment of your blockquote.

A well-known quote, contained in a blockquote element.

— Someone famous in *Source Title*

HTML

```
<figure class="text-center">
  <blockquote class="blockquote">
    <p>A well-known quote, contained in a blockquote element.</p>
  </blockquote>
  <figcaption class="blockquote-footer">
    Someone famous in <cite title="Source Title">Source Title</cite>
  </figcaption>
</figure>
```

6. Salinlah kode yang tertera, lalu tempelkan di tag body pada halaman html yang telah kita hubungkan dengan Bootstrap kita

```
<figure class="text-center">
<blockquote class="blockquote">
<p>A well-known quote, contained in a blockquote element.</p>
</blockquote>
<figcaption class="blockquote-footer">
Someone famous in
<cite title="Source Title">Source Title </cite>
```

```
</figcaption>  
</figure>
```

```
◇ tes.html > ◇ html > ◇ body  
1  <!DOCTYPE html>  
2  <html lang="en">  
3  <head>  
4  |   <meta charset="UTF-8">  
5  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6  |   <title>Document</title>  
7  <link rel="stylesheet" href="css/bootstrap.min.css">  
8  </head>  
9  <body>  
10 |   <figure class="text-center">  
11 |       <blockquote class="blockquote">  
12 |           |   <p>A well-known quote, contained in a blockquote element.</p>  
13 |       </blockquote>  
14 |       <figcaption class="blockquote-footer">  
15 |           |   Someone famous in <cite title="Source Title">Source Title</cite>  
16 |       </figcaption>  
17 |   </figure>  
18 </html>
```

7. Hasilnya akan terlihat seperti berikut:

```
A well-known quote, contained in a blockquote element.  
— Someone famous in Source Title
```

8. Kita bisa mengganti kata kata nya dengan cara mengganti teks yang ada di dalam tag p dan untuk teks kecil bagian bawah kita ganti pada bagian dalam tag figcaption

```
<figure class="text-center">  
  <blockquote class="blockquote">  
    <p>Sebenarnya otak kita sama aja, yang <br>  
      membedakan hanya siapa yang <br>
```

```
mulai belajar duluan dan siapa yang <br>
belajar terus menerus </p>
</blockquote>
<figcaption class="blockquote-footer">
  Dea Afrizal Doroboka<cite title="Source Title">Doroboka</cite>
</figcaption>
</figure>
```

Hasilnya adalah sebagai berikut:

Sebenarnya otak kita sama aja, yang
membedakan hanya siapa yang
mulai belajar duluan dan siapa yang
belajar terus menerus
— Dea Afrizal Doroboka

Keterangan :

1. `text-center` pada `<figure>` : Kelas ini diterapkan pada elemen `<figure>`, yang mengakibatkan kontennya, termasuk elemen `<blockquote>` dan `<figcaption>`, akan diatur menjadi ketengah halaman maupun kontainer.
2. `blockquote` pada `<blockquote>` : Kelas ini memberikan gaya khusus pada elemen `<blockquote>`. Elemen ini digunakan untuk merinci sebuah kutipan atau teks yang dianggap signifikan. Penggunaan kelas ini dari Bootstrap mungkin memberikan tampilan tertentu, seperti memodifikasi gaya margin atau padding, untuk memberikan estetika yang lebih baik.
3. `blockquote-footer` pada `<figcaption>` : Kelas ini memberikan gaya khusus pada elemen `<figcaption>` yang berada di dalam `<figure>`. Elemen ini kemungkinan berisi informasi tambahan atau keterangan terkait elemen-elemen lain dalam `<figure>`.
 - `Dea Afrizal Doroboka` : Ini adalah teks yang berada di dalam elemen `<figcaption>`, memberikan informasi tambahan atau keterangan terkait dengan kutipan atau elemen-elemen lain dalam `<figure>`.
 - `<cite title="Source Title">Doroboka / cite>` : Ini adalah elemen `<cite>` yang memberikan judul sumber ("Source Title"). Penggunaan kelas `blockquote-footer` dari Bootstrap mungkin menyusun elemen ini dengan tata letak dan gaya tertentu, seperti menetapkannya sebagai teks kaki atau memberikan gaya yang konsisten dengan elemen

lain dalam kelompok `<figure>`.

Keterangan:

Untuk bagian "Dea Afrizal" adalah nama seseorang yang mengatakan quotes tersebut dan "Doroboka" adalah sumber dimana orang tersebut mengatakan kata-kata itu. Kata-kata yang dituliskan yaitu berada dalam tag `p`.

Navbar

1. Buka halaman getbootstrap.com lalu ketik "navbar" di kolom "search" dan klik enter

The screenshot shows the Bootstrap v5.3 documentation for the 'Navbar' component. The left sidebar has 'Navbar' selected. The main content area features the title 'Navbar' and a brief description of its purpose. Below this is a section titled 'How it works' with a list of guidelines for using navbars. A note at the bottom mentions new features in v5.2.0. On the right, there's a sidebar titled 'On this page' with a list of related topics.

2. Carilah jenis navbar yang diinginkan.

The screenshot shows the 'Image and text' section of the 'Image' documentation. It includes an example of HTML code for a navbar with an image and text. Below the code, there's a note about using additional utilities like `.d-inline-block` and `.align-text-top`. The right sidebar lists other documentation categories.

3. Jika misalkan kita telah menemukan jenis navbar yang kita inginkan, maka, salinlah kode program yang ada dibawah gambar contoh navbar tersebut

Navbar Home Features Pricing Dropdown link ▾

HTML



```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavDropdown">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Features</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Pricing</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" ,>
            Dropdown link
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Action</a></li>
```

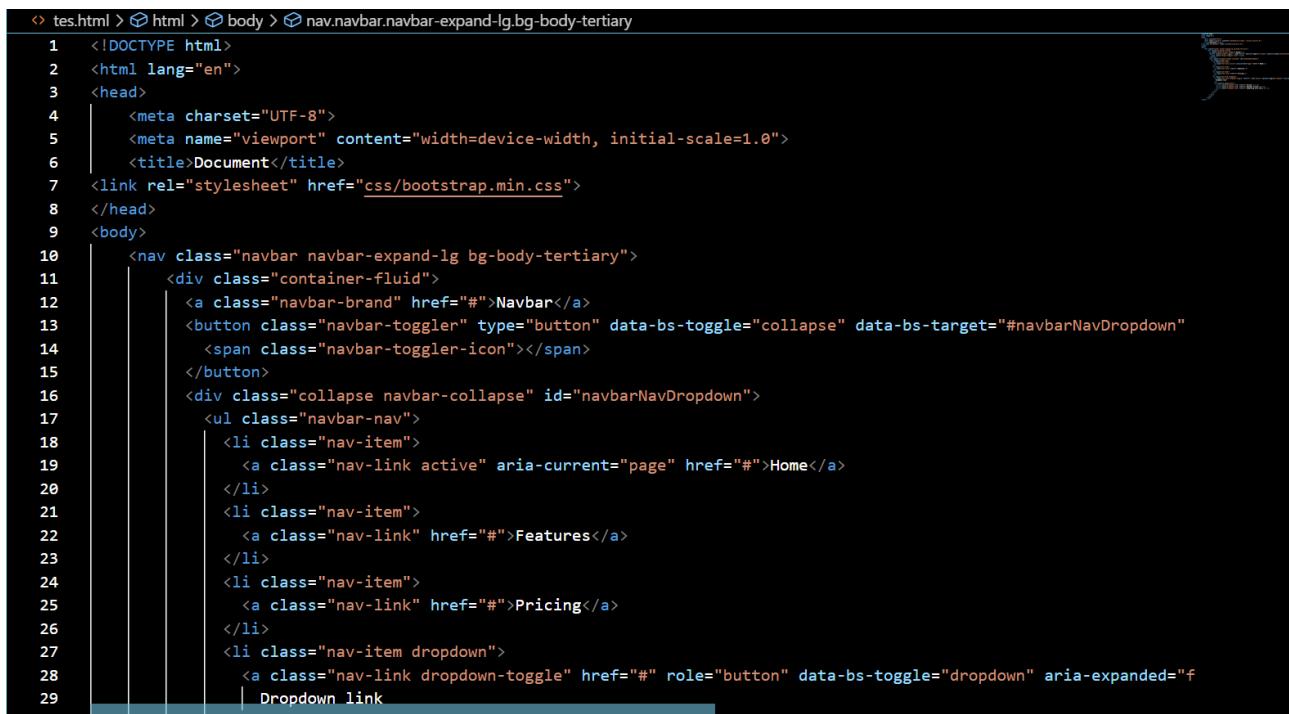
```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavDropdown">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Features</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Pricing</a>
        </li>
        <li class="nav-item dropdown">
```

```

        <a class="nav-link dropdown-toggle" href="#" role="button"
data-bs-toggle="dropdown" aria-expanded="false">
            Dropdown link
        </a>
        <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Action</a></li>
            <li><a class="dropdown-item" href="#">Another
action</a></li>
            <li><a class="dropdown-item" href="#">Something else
here</a></li>
        </ul>
    </li>
</ul>
</div>
</div>
</nav>

```

4. Tempelkanlah kode program yang terlah kita salin ke dalam tag body di file html kita. Jangan lupa untuk menautkan file html kita dengan bootstrap seperti pada langkah-langkah sebelumnya



```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
</head>
<body>
    <nav class="navbar navbar-expand-lg bg-body-tertiary">
        <div class="container-fluid">
            <a class="navbar-brand" href="#">Navbar</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNavDropdown">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link active" aria-current="page" href="#">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Features</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Pricing</a>
                    </li>
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="f


```

5. Jalankanlah file html yang sudah ditempelki kode program bootstrap tadi melalui web browser. Maka lihatlah hasilnya

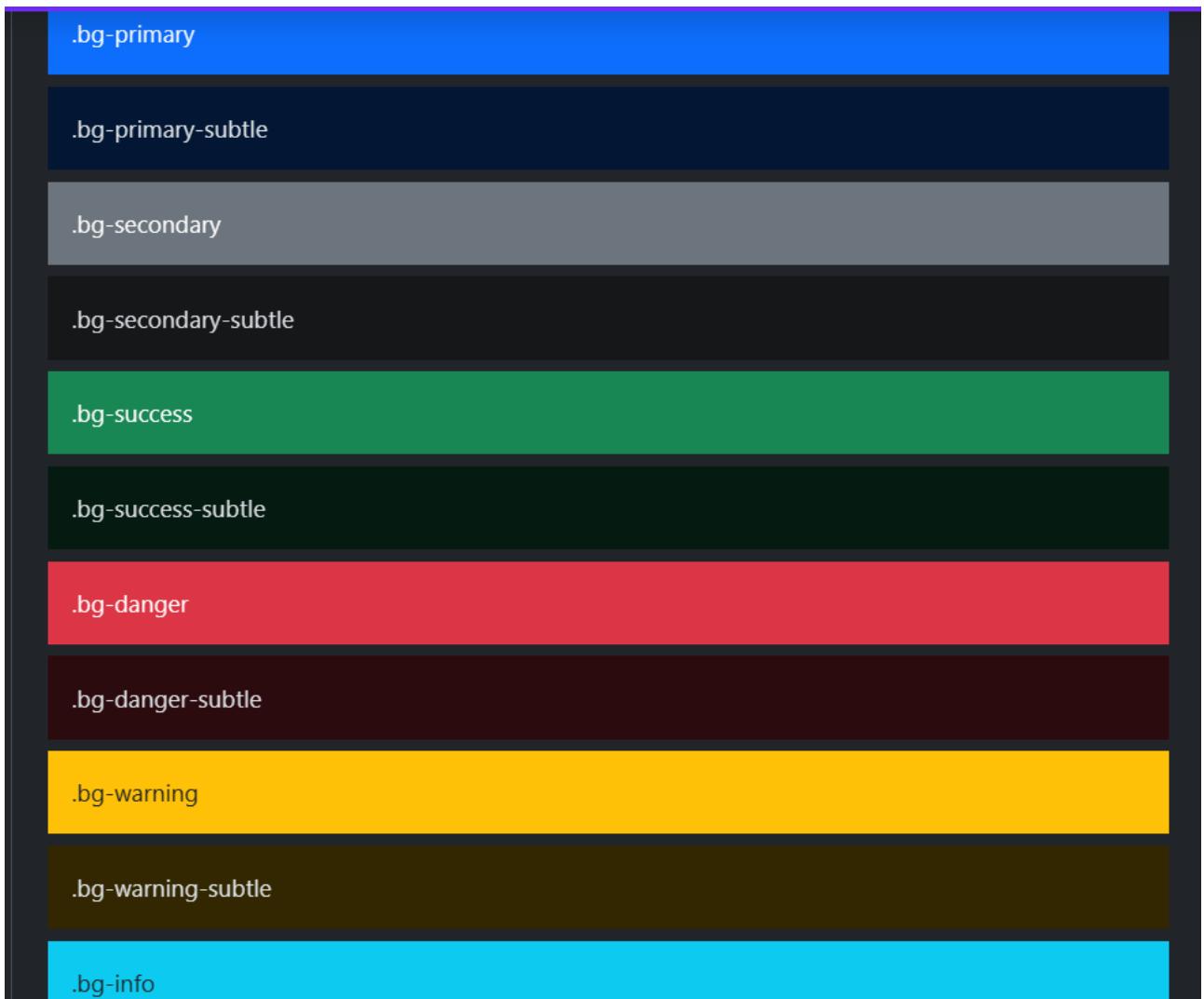


6. Misalnya kita ingin memberikan warna hijau pada background navbar kita. Pertama ketiklah "background" di kolom search

The screenshot shows the Bootstrap v5.3 documentation for the 'Background' section. The sidebar on the left has 'Background' selected under the 'Utilities' category. The main content area features a green background with the class `.bg-success` applied to a button. A sidebar on the right lists related topics like 'Background color', 'Background gradient', and 'Opacity'.

7. Carilah Warna yang kita inginkan. Misalnya dalam praktek ini adalah warna hijau





Terdapat banyak jenis warna dalam bootstrap.

8. Karena kita ingin mengkostumisasi warna background kita menjadi hijau maka kita akan memakai bg-success . ketikkan bg-success pada class tag pembuka seperti di bawah ini

```
↳ tes.html > ⌂ html > ⌂ body > ⌂ nav.navbar.navbar-expand-lg.bg-success
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Document</title>
7       <link rel="stylesheet" href="css/bootstrap.min.css">
8     </head>
9     <body>
10      <nav class="navbar navbar-expand-lg bg-success">
11        <div class="container-fluid">
```

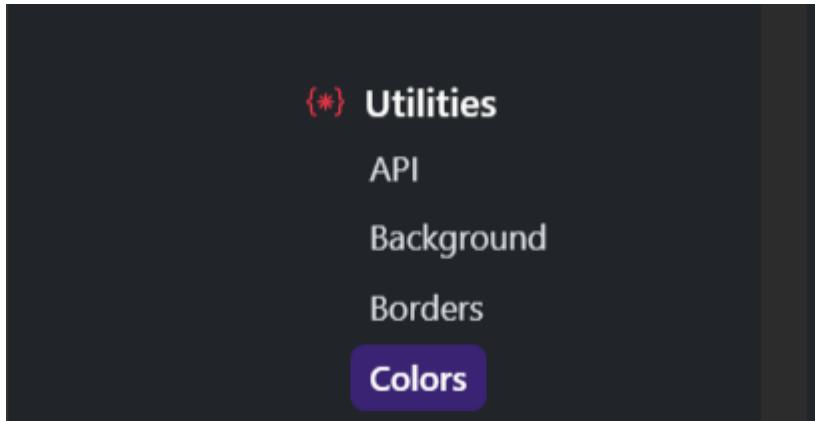
Keterangan:

Karena kita akan memberikan warna background pada navbar maka untuk menerapkannya, langsung terapkan class tag yang membungkus semua eleme-elemen yang ada pada tag tersebut.

9. Jika sudah maka hasilnya akan seperti ini:



10. Sekarang kita akan mengubah warna font yang ada pada navbar. Langkah pertama klik "Colors" pada bagian Utilities



11. Maka kita akan diarahkan pada halaman yang memuat berbagai macam warna font

```
.text-secondary-emphasis  
.text-success  
.text-success-emphasis  
.text-danger  
.text-danger-emphasis  
.text-warning  
.text-warning-emphasis  
.text-info  
.text-info-emphasis  
.text-light  
.text-light-emphasis  
.text-dark
```

```
.text-dark-emphasis  
.text-body  
.text-body-emphasis  
.text-body-secondary  
.text-body-tertiary
```

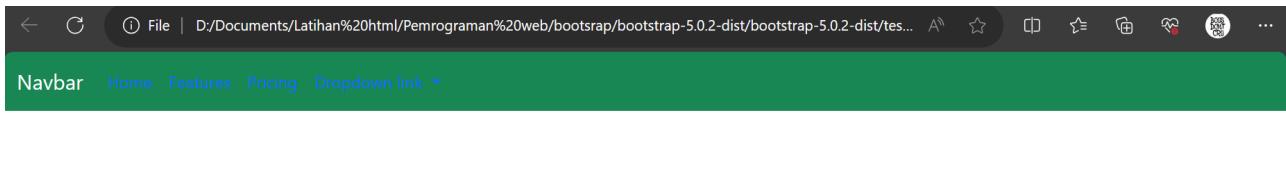
12. Carilah warna font yang kita inginkan. Misalnya pada kostumisasi navbar ini. Kita akan menggunakan warna font putih

```
.text-black  
.text-white  
.text-black-50  
.text-white-50
```

13. Karena kita ingin menggunakan warna font putih maka perintah yang digunakan adalah "text-white"
14. Masuklah kembali ke dalam file html kita
15. pada bagian class sebuah tag yang membungkus text yang akan kita ganti warna, ketiklah text-white . Contohnya ada pada gambar berikut:

```
<nav class="navbar navbar-expand-lg bg-success">
|   <div class="container-fluid">
|       <a class="navbar-brand text-white" href="#">Navbar</a>
```

16. Hasilnya akan terlihat seperti di gambar ini:



17. Terapkanlah Text-White diseluruh class pada tag yang membungkus text-text itu seperti halnya pada text navbar tadi

```
19 |           <a class="nav-link active text-white" aria-current="page" href="#">Home</a>
20 |       </li>
21 |       <li class="nav-item">
22 |           <a class="nav-link text-white" href="#">Features</a>
23 |       </li>
24 |       <li class="nav-item">
25 |           <a class="nav-link text-white" href="#">Pricing</a>
26 |       </li>
27 |       <li class="nav-item dropdown">
28 |           <a class="nav-link dropdown-toggle text-white" href="#" role="button" data-bs-toggle="dropdown" aria-
29 |               |   Dropdown link
30 |           </a>
```

18. Jika file html dijalankan, hasilnya adalah sebagai berikut



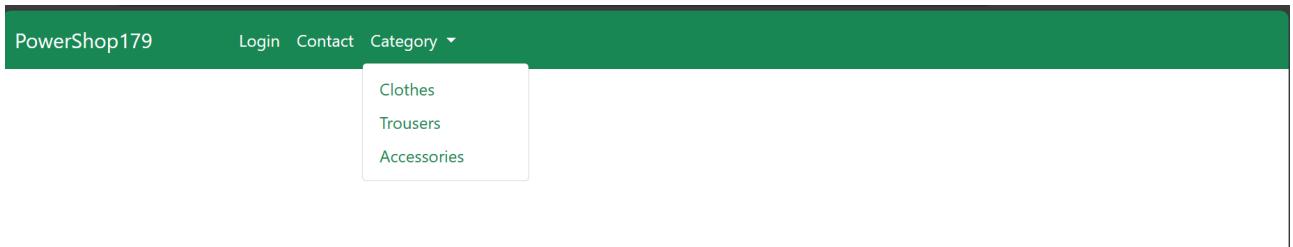
19. Sekarang kita akan mengganti text yang ada di navbar tersebut. Caranya sangat mudah, yaitu cukup kita ganti text-text tadi dengan kata kata yang kita inginkan. Misalnya adalah sebagai berikut:

```

10   <nav class="navbar navbar-expand-lg bg-success" >
11     <div class="container-fluid" >
12       <a class="navbar-brand text-white" href="#">PowerShop179</a>
13       <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown" >
14         <span class="navbar-toggler-icon"></span>
15       </button>
16       <div class="collapse navbar-collapse" id="navbarNavDropdown" >
17         <ul class="navbar-nav" >
18           <li class="nav-item" >
19             <a class="nav-link active text-success" aria-current="page" href="#">Home</a>
20           </li>
21           <li class="nav-item" >
22             <a class="nav-link text-white" href="#">Login</a>
23           </li>
24           <li class="nav-item" >
25             <a class="nav-link text-white" href="#">Contact</a>
26           </li>
27           <li class="nav-item dropdown" >
28             <a class="nav-link dropdown-toggle text-white" href="#" role="button" data-bs-toggle="dropdown" aria-
29               | Category
30             </a>
31             <ul class="dropdown-menu" >
32               <li><a class="dropdown-item text-success" href="#">Clothes</a></li>
33               <li><a class="dropdown-item text-success" href="#">Trousers</a></li>
34               <li><a class="dropdown-item text-success" href="#">Accessories</a></li>
35             </ul>
36           </li>
37         </ul>
38       </div>

```

20. Maka hasilnya adalah sebagai berikut:



21. Sekarang kita akan menambahkan kolom search pada navbar kita. pada bagian navbar di getbootstrap.com Scroll ke bawah hingga menemukan seperti pada gambar

```

<nav class="navbar bg-body-tertiary">
  <div class="container-fluid">
    <form class="d-flex" role="search">
      <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
      <button class="btn btn-outline-success" type="submit">Search</button>
    </form>
  </div>
</nav>

```

22. Salin kodennya lalu tempelkan di bagian bawah kode navbar kita tadi. Jadi jangan disatukan ke dalam kode navbar tadi.

```
<nav class="navbar bg-body-tertiary">
  <div class="container-fluid">
    <form class="d-flex" role="search">
      <input class="form-control me-2" type="search"
placeholder="Search" aria-label="Search">
      <button class="btn btn-outline-success"
type="submit">Search</button>
    </form>
  </div>
</nav>
```

```
25   <a class="nav-link text-white" href="#">Contact</a>
26   </li>
27   <li class="nav-item dropdown">
28     <a class="nav-link dropdown-toggle text-white" href="#" role="button" data-bs-toggle="dropdown" aria-
29       | Category
30       </a>
31       <ul class="dropdown-menu">
32         <li><a class="dropdown-item text-success" href="#">Clothes</a></li>
33         <li><a class="dropdown-item text-success" href="#">Trousers</a></li>
34         <li><a class="dropdown-item text-success" href="#">Accessories</a></li>
35       </ul>
36     </li>
37   </ul>
38 </div>
39 </div>
40 </nav>
41
42 <nav class="navbar bg-body-tertiary">
43   <div class="container-fluid">
44     <form class="d-flex" role="search">
45       <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
46       <button class="btn btn-outline-success" type="submit">Search</button>
47     </form>
48   </div>
49 </nav>
50 </html>
```

Keterangan:

- [navbar] : Kelas ini menunjukkan bahwa elemen [<nav>] adalah bagian dari komponen navbar. Navbar adalah bagian dari Bootstrap yang menyediakan navigasi di bagian atas halaman web. Dengan memberikan kelas navbar , kita memberitahu Bootstrap untuk menerapkan gaya dan tata letak khusus yang terkait dengan navigasi.
- [bg-body-tertiary] : Kelas ini memberikan warna latar belakang pada navbar. Dalam hal ini, warna latar belakangnya diberi warna sesuai dengan warna "hijau" Warna latar belakang ini dapat disesuaikan sesuai kebutuhan desain.
- [container-fluid] : Kelas ini diterapkan pada elemen [<div>] yang membungkus elemen-elemen dalam navbar. Kelas ini memberikan padding yang sesuai dan membuat elemen-elemen di dalamnya menjaga lebar penuh dari layar (menggunakan grid system

Bootstrap). Dengan kata lain, kontennya akan merespons secara baik pada berbagai lebar perangkat.

4. `[d-flex]` : Kelas ini mengubah elemen `<form>` menjadi kontainer flex. Flexbox adalah teknik tata letak yang kuat di CSS yang memudahkan pengaturan dan penataan elemen dalam satu atau dua dimensi. Dengan memberikan kelas `d-flex` , elemen form dan anak-anaknya dapat disusun secara fleksibel.
5. `[form-control]` : Kelas ini diterapkan pada elemen `<input>` untuk memberikan gaya yang konsisten pada elemen formulir. Di sini, elemen input diberi gaya Bootstrap standar untuk mengubahnya menjadi kontrol formulir yang responsif dan berada dalam baris yang sesuai.
6. `[me-2]` : Ini adalah kelas Bootstrap yang memberikan margin kanan sebesar 2 pada elemen yang memilikinya. Dalam codingan ini, kelas ini diterapkan pada elemen input untuk memberikan jarak margin kanan.
7. `btn` dan `[btn-outline-success]` : Kelas ini memberikan gaya pada elemen `<button>` . Kelas `btn` mengindikasikan bahwa ini adalah elemen tombol Bootstrap, sedangkan `btn-outline-success` memberikan gaya tombol dengan warna tepi hijau ("success" dalam Bootstrap).
8. `type="submit"` pada tombol: Ini menentukan bahwa tombol tersebut bertindak sebagai tombol submit dalam formulir. Ketika ditekan, formulir akan dikirim.
9. Maka hasilnya adalah seperti berikut:



10. Untuk membuat agar navbar tersebut terlihat menyatu, berikan `background-color` yang sama pada navbar diatasnya, yaitu menggunakan kode `bg-success` . Letakkan di class tag yang membungkus seluruh elemen search tadi.

```
<nav class="navbar bg-success">
  <div class="container-fluid">
    <form class="d-flex" role="search">
      <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
      <button class="btn btn-outline-success" type="submit">Search</button>
    </form>
  </div>
</nav>
```

11. Maka hasil yang muncul akan seperti ini

Buttons

Pada praktik sebelumnya kita sudah mengatur navbar dan juga menambahkan kolom search. Namun ada masalah dengan tombolnya. Karena tombolnya memiliki warna yang sama dengan warna background yang digunakan yaitu hijau, Maka tombol tersebut tidak terlihat akibat warnanya menyatu. Pada bagian kali kita akan membahas cara mengatur sebuah tombol di bootstrap. Berikut caranya:

1. Permasalahan tadi ialah background color tombolnya yang menyatu dengan warna background navbarnya. Untuk mengatur warna tombol, pertama di situs getbootstrap.com, carilah bagian "buttons" di Components

Components

Accordion

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Close button

Collapse

Dropdowns

List group

Modal

Navbar

Navs & tabs

Offcanvas

2. Jika sudah menemukan klik bagian tersebut, maka kita akan diarahkan pada bagian buttons seperti pada gambar

Buttons

Use Bootstrap's custom button styles for actions in forms, dialogs, and more with support for multiple sizes, states, and more.



I've spent 2 years learning DNS while building NSLookup.io. Now, I'm teaching everything I know.
ads via Carbon

[View on GitHub](#)

On this page

Base class

Variants
Disable text wrapping

Button tags

Outline buttons

Sizes

Disabled state

Link functionality caveat

Block buttons

Button plugin

Toggle states

Methods

CSS

Variables

Sass variables

Sass mixins

Sass loops

Base class

Bootstrap has a base `.btn` class that sets up basic styles such as padding and content alignment. By default, `.btn` controls have a transparent border and background color, and lack any explicit focus and hover styles.

Base class

HTML



```
<button type="button" class="btn">Basic class</button>
```

3. Scroll lah ke bawah hingga menemukan bagian "Outlines Buttons"

Outline buttons

In need of a button, but not the hefty background colors they bring? Replace the default modifier classes with the `.btn-outline-*` ones to remove all background images and colors on any button.

Primary

Secondary

Success

Danger

Warning

Info

Light

HTML



```
<button type="button" class="btn btn-outline-primary">Primary</button>
<button type="button" class="btn btn-outline-secondary">Secondary</button>
<button type="button" class="btn btn-outline-success">Success</button>
<button type="button" class="btn btn-outline-danger">Danger</button>
<button type="button" class="btn btn-outline-warning">Warning</button>
<button type="button" class="btn btn-outline-info">Info</button>
<button type="button" class="btn btn-outline-light">Light</button>
<button type="button" class="btn btn-outline-dark">Dark</button>
```

4. Saat ini kita akan menggunakan jenis warna putih agar tidak lagi menyatu dengan warna background navabrnya. Pada outline buttons yang berwarna putih, tertulis "Light". Oleh karena itu salinlah kode program yang ada tulisan "Light".

```
<button type="button" class="btn btn-outline-light">Light</button>
```

5. Blok lah terlebih dahulu kode program yang mengatur tombol pada bagian search tadi

```
<nav class="navbar bg-success">
  <div class="container-fluid">
    <form class="d-flex" role="search">
      <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
      <button class="btn btn-outline-success" type="submit">Search</button>
    </form>
  </div>
</nav>
```

6. Tempelkanlah kode program outline light, untuk menggantikan kode program button yang lama

```
<nav class="navbar bg-success">
  <div class="container-fluid">
    <form class="d-flex" role="search">
      <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
      <button type="button" class="btn btn-outline-light">Light</button>
    </form>
  </div>
</nav>
```

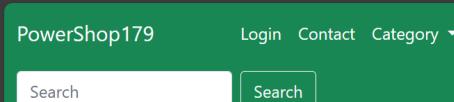
7. Maka hasilnya akan bagus seperti pada gambar dibawah ini:



8. Ubahlah Text "Light" menjadi "Search"

```
<button type="button" class="btn btn-outline-light">Search</button>
```

9. Maka hasilnya akan seperti ini:



Cards dan Form

Sekarang kita akan membuat sebuah form, dimana form tersebut kita asumsikan sebagai login page dari web kita tadi. Dalam pembuatan form login ini, kita akan menggunakan cards sebagai layout/tata letak yang akan mengatur form login tadi. Berikut langkah-langkahnya:

1. Yang pertama kita tentunya harus menentukan layout cards nya terlebih dahulu. Hal ini bertujuan agar jika tata letak card sudah ada, maka kita tinggal mengatur form nya di dalam card tadi. Masuk ke situs bootstrap tadi lalu pilih cards

Cards

[View on GitHub](#)

Bootstrap's cards provide a flexible and extensible content container with multiple variants and options.



Get 10 free Adobe Stock photos. Start downloading amazing royalty-free stock photos today.

ads via Carbon

About

A **card** is a flexible and extensible content container. It includes options for headers and footers, a wide variety of content, contextual background colors, and powerful display options. If you're familiar with Bootstrap 3, cards replace our old panels, wells, and thumbnails. Similar functionality to those components is available as modifier classes for cards.

Example

Cards are built with as little markup and styles as possible, but still manage to deliver a ton of control and customization. Built with flexbox, they offer easy alignment and mix well with other Bootstrap components. They

On this page

About

Example

Content types

Body

Titles, text, and links

Images

List groups

Kitchen sink

Header and footer

Sizing

Using grid markup

Using utilities

Using custom CSS

Text alignment

Navigation

Images

Image caps

Image overlays

Horizontal

Card styles

Background and color

Border

Mixins utilities

- Scrol ke bawah hingga menemukan layout yang kita inginkan, misalnya seperti pada gambar:

Featured

Special title treatment

With supporting text below as a natural lead-in to additional content.

[Go somewhere](#)

HTML

```
<div class="card">
  <div class="card-header">
    Featured
  </div>
  <div class="card-body">
    <h5 class="card-title">Special title treatment</h5>
    <p class="card-text">With supporting text below as a natural lead-in to additional content...
      <a href="#" class="btn btn-primary">Go somewhere</a>
    </p>
  </div>
</div>
```

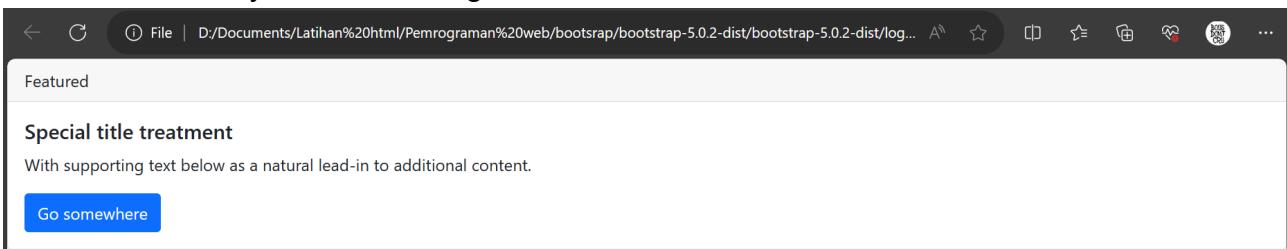
- Pada bagian bawah contoh, terdapat kode program. Salinlah kode program tersebut

```

<div class="card">
    <div class="card-header">
        Featured
    </div>
    <div class="card-body">
        <h5 class="card-title">Special title treatment</h5>
        <p class="card-text">With supporting text below as a natural lead-in to additional content.</p>
        <a href="#" class="btn btn-primary">Go somewhere</a>
    </div>
</div>

```

4. Tempelkanlah kode program itu pada halaman html terkait. Tentu saja dengan bootstrap yang sudah ditautkan, baik offline, maupun online.
5. Maka hasil awalnya adalah sebagai berikut:



6. karena jarak atas antara card dengan batas website terlalu dekat maka pada class card tambahkan mt-2 .

```

<div class="card mt-2">
    <div class="card-header">
        Featured
    </div>
    <div class="card-body">
        <h5 class="card-title">Special title treatment</h5>
        <p class="card-text">With supporting text below as a natural lead-in to additional content.</p>
        <a href="#" class="btn btn-primary">Go somewhere</a>
    </div>
</div>

```

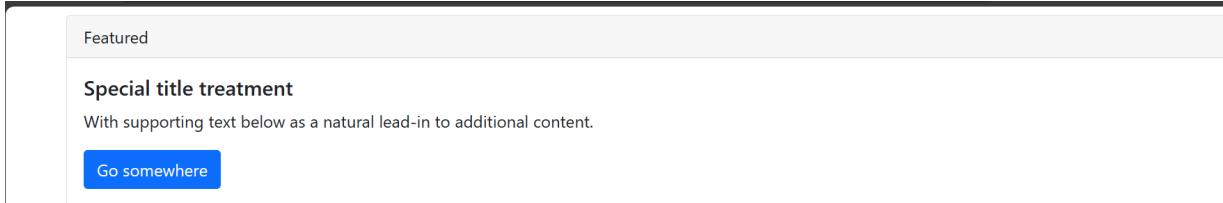
7. Maka hasilnya adalah seperti ini:



8. Sekarang kita ingin membuatnya terlihat ke tengah. Maka buatlah sebuah div dengan class yaitu "container", lalu salin masuklah kode program card tadi. Dengan kata lain, bungkuslah kode program card dengan sebuah div dengan class="container".

```
<div class="container">
  <div class="card mt-2">
    <div class="card-header">
      Featured
    </div>
    <div class="card-body">
      <h5 class="card-title">Special title treatment</h5>
      <p class="card-text">With supporting text below as a natural lead-in to additional content.</p>
      <a href="#" class="btn btn-primary">Go somewhere</a>
    </div>
  </div>
</div>
```

9. Berikut hasil dari di bungkusnya cards tadi.



10. Sekarang kita akan membuat form nya. Pertama-tama, kita perlu membuat judul form. Caranya adalah ganti tulisan "Featured" di html kita dengan kata yang kita inginkan. misalnya pada program ini kita akan mengganti tulisan menjadi "Form Login".

```
<div class="container">
  <div class="card mt-2">
    <div class="card-header">
      | <h5>Form Login</h5>
    </div>
    <div class="card-body">
      | <h5 class="card-title">Special title treatment</h5>
      | <p class="card-text">With supporting text below as a natural lead-in to additional content.</p>
      | <a href="#" class="btn btn-primary">Go somewhere</a>
    </div>
  </div>
</div>
```

```
<div class="container">
  <div class="card mt-2">
    <div class="card-header">
      <h5>Form Login</h5>
    </div>
    <div class="card-body">
      <h5 class="card-title">Special title treatment</h5>
```

```

<p class="card-text">With supporting text below as a natural lead-in to additional content.</p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
</div>
</div>
</div>

```

Keterangan:

- [container] pada [<div class="container">] : Kelas ini memberikan tata letak yang terkandung pada lebar tertentu (dalam pixel) dan ditengahkan di tengah halaman atau elemen yang memuatnya. Ini membantu dalam mengatur konten agar sesuai dengan standar desain Bootstrap.
- [card] pada [<div class="card mt-2">] : Kelas ini memberikan gaya dan tata letak khusus untuk elemen [<div>] yang merupakan kartu (card). Kartu adalah elemen Bootstrap yang sering digunakan untuk menampilkan konten atau informasi dalam satu unit terpisah dengan gaya yang konsisten.
 - mt-2 : Kelas ini memberikan margin atas (margin-top) sebesar 2 unit. Ini membantu memberikan ruang di bagian atas kartu dan memisahkannya dari elemen-elemen sekitarnya.
- [card-header] pada [<div class="card-header">] : Kelas ini memberikan gaya khusus untuk elemen [<div>] yang berfungsi sebagai header kartu. Header kartu biasanya berisi judul atau informasi lain yang menandai atau menjelaskan konten kartu.
 - [<h5>Form Login / h5>] : Ini adalah elemen judul level 5 (h5) di dalam header kartu yang memberikan judul "Form Login" pada kartu.
- [card-body] pada [<div class="card-body">] : Kelas ini memberikan gaya khusus untuk elemen [<div>] yang berisi tubuh atau konten utama kartu. Ini membantu memisahkan dan memvisualisasikan konten utama kartu.
 - [<h5 class="card-title">Special title treatment / h5>] : Ini adalah elemen judul level 5 (h5) di dalam tubuh kartu yang memberikan judul khusus.
 - [<p class="card-text">With supporting text below as a natural lead-in to additional content. / p>] : Ini adalah elemen paragraf di dalam tubuh kartu yang memberikan teks pendukung atau keterangan untuk konten utama kartu.
 - [Go somewhere / a>] : Ini adalah elemen anchor (tautan) yang merupakan tombol dengan kelas Bootstrap "btn" dan "btn-primary", memberikan tampilan dan warna tertentu sesuai dengan desain Bootstrap.

Perbedaan Container dengan Container-fluid:

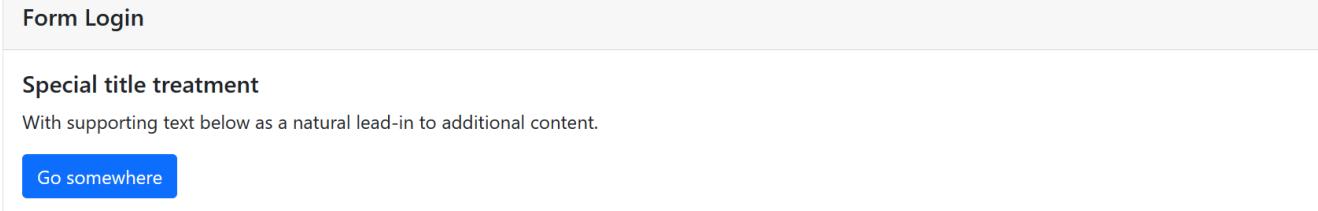
container:

- [container] memberikan tata letak yang terpusat dan memiliki lebar yang tetap (fixed-width).
- Lebar [container] diatur dalam satuan piksel dan tetap konstan terlepas dari ukuran layar pengguna.
- Digunakan untuk membuat tata letak yang terkonsentrasi di tengah halaman dan memberikan batasan pada lebar kontennya.

container-fluid:

- `[container-fluid]` memberikan tata letak yang penuh lebar, mengisi seluruh lebar layar.
- Lebar `[container-fluid]` disesuaikan dengan lebar layar pengguna, sehingga konten dapat memanfaatkan seluruh lebar tampilan tanpa batasan piksel tertentu.
- Digunakan ketika Anda ingin membuat tata letak yang menyesuaikan diri dengan lebar layar dan memberikan tampilan responsif.

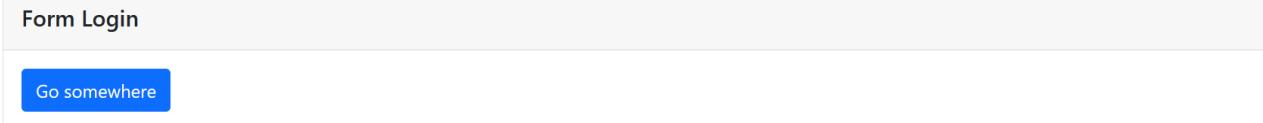
11. Hasilnya adalah sebagai berikut



12. Sekarang pada bagian dalam div card-body hapus semua elemen yang ada didalamnya kecuali button:

```
<div class="container">
  <div class="card mt-2">
    <div class="card-header">
      <h5>Form Login</h5>
    </div>
    <div class="card-body">
      <a href="#" class="btn btn-primary">Go somewhere</a>
    </div>
  </div>
</div>
```

13. Hasilnya akan seperti ini



14. Langkah selanjutnya adalah menambahkan elemen-elemen form ke dalam card-body untuk membuat formulir login. Anda dapat menggunakan elemen-elemen HTML seperti form , input , dan button untuk ini. bukalah kembali getbootstrap.com dan pilihlah form yang kita inginkan:

Email address

We'll never share your email with anyone else.

Password

Check me out

Submit

HTML

```
<form>
  <div class="mb-3">
    <label for="exampleInputEmail1" class="form-label">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
    <div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>
  </div>
```

Copy to clipboard

15. Salinlah kode program dibawah contoh forms tadi:

```
<form>
  <div class="mb-3">
    <label for="exampleInputEmail1" class="form-label">Email
address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-
describedby="emailHelp">
    <div id="emailHelp" class="form-text">We'll never share your email
with anyone else.</div>
  </div>
  <div class="mb-3">
    <label for="exampleInputPassword1" class="form-label">Password</label>
    <input type="password" class="form-control"
id="exampleInputPassword1">
  </div>
  <div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me
out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

16. Tempelkanlah didalam cards-body tadi

```

<body>
  <div class="container">
    <div class="card mt-2">
      <div class="card-header">
        <h5>Form Login</h5>
      </div>
      <div class="card-body">
        <form>
          <div class="mb-3">
            <label for="exampleInputEmail1" class="form-label">Email address</label>
            <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
            <div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>
          </div>
          <div class="mb-3">
            <label for="exampleInputPassword1" class="form-label">Password</label>
            <input type="password" class="form-control" id="exampleInputPassword1">
          </div>
          <div class="mb-3 form-check">
            <input type="checkbox" class="form-check-input" id="exampleCheck1">
            <label class="form-check-label" for="exampleCheck1">Check me out</label>
          </div>
          <button type="submit" class="btn btn-primary">Submit</button>
        </form>
      </div>
    </div>
  </div>

```

17. Maka hasilnya adalah seperti ini:

Dengan demikian kita sudah berhasil membuat layout cards dimana form login sebagai kontennya

Grid For Responsive Web/Layouting

Bootstrap memiliki sistem tata letak yang kuat yang bernama grid. Grid berguna untuk membuat tata letak yang responsif di halaman web kita. Sistem grid Bootstrap berdasarkan konsep kolom yang dapat diatur dalam baris. Berikut adalah beberapa hal penting yang perlu kita ketahui tentang grid dan layouting di Bootstrap:

1. KONTAINER (Container)

Grid Bootstrap harus ditempatkan dalam elemen kontainer. Terdapat dua jenis kontainer yang tersedia, yaitu .container dan .container-fluid. .container memiliki lebar terbatas dan

akan disesuaikan dengan lebar layar. .container ini digunakan ketika kita ingin membuat tata letak yang terpusat dan terbatas pada lebar tertentu. .container-fluid memiliki lebar penuh dan akan mengisi seluruh lebar layar. Kontainer ini digunakan ketika kita ingin membuat tata letak yang menyesuaikan dengan lebar layar penuh.

2. BARIS (Row)

Baris digunakan untuk mengelompokkan kolom-kolom dalam tata letak. kita dapat menambahkan class .row pada elemen yang berfungsi sebagai wadah kolom. Baris ini akan memastikan bahwa kolom-kolom di dalamnya akan diatur secara horizontal.

3. KOLOM (Column):

Kolom adalah bagian dasar dari sistem grid Bootstrap. Kolom- kolom ditempatkan di dalam baris dan digunakan untuk membagi horizontal ruang dalam grid. Setiap baris dipecah menjadi 12 kolom, yang dapat kita bagi sesuai kebutuhan.

Catatan: kita dapat menggunakan kelas seperti .col- , .col-sm- , .col-md- , .col-lg- , atau .col-xl- untuk menentukan bagaimana kolom akan berperilaku di berbagai ukuran perangkat. Misalnya, .col-sm-6 akan membuat kolom tersebut memiliki lebar setengah dari baris pada ukuran layar kecil. kita dapat menggabungkan kelas kolom untuk ukuran layar yang berbeda untuk menciptakan tata letak yang responsif. Misalnya, .col-md-6 .col-lg-4 akan membuat kolom tersebut memiliki lebar setengah dari baris pada ukuran layar medium, dan memiliki lebar sepertiga dari baris pada ukuran layar besar.

4. OFFSET

Kita dapat menggunakan kelas offset untuk memindahkan kolom ke samping. Misalnya, .offset-md-2 akan memindahkan kolom 2 satuan ke kanan pada layar dengan ukuran medium. Offset digunakan ketika Kita ingin membuat ruang kosong di antara kolom-kolom.

5. Perilaku pada Ukuran Layar yang Berbeda

Kita dapat menggunakan kelas-kelas grid Bootstrap yang berbeda untuk mengontrol tampilan kolom pada ukuran layar yang berbeda. Misalnya, Kita dapat menggunakan .colsm- untuk ukuran layar kecil, .col-md- untuk ukuran layar medium, dan sebagainya. Dengan cara ini, Kita dapat membuat tata letak yang responsif untuk berbagai perangkat.

6. NESTING

Nesting memungkinkan Kita menempatkan baris dan kolom di dalam kolom lainnya. Dengan cara ini, Kita dapat membuat tata letak yang lebih kompleks dengan komponen-komponen yang terkait. Misalnya, Kita dapat menempatkan sebuah baris di dalam kolom yang ada di dalam baris lainnya

7. ORDER

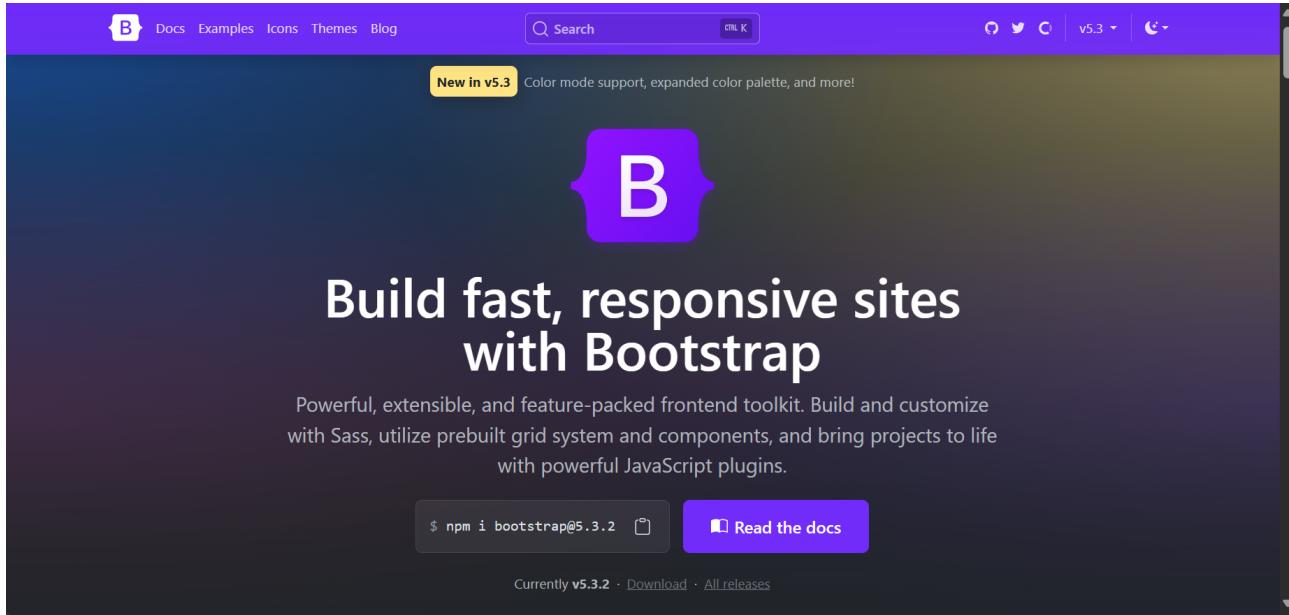
Kita dapat menggunakan kelas .order- untuk mengubah urutan kolom pada ukuran layar tertentu. Misalnya, .order-first akan memindahkan kolom ke posisi pertama, dan .orderlast akan memindahkan kolom ke posisi terakhir. Dengan menggunakan kelas ini, Kita dapat mengatur ulang urutan kolom untuk mencapai tata letak yang diinginkan pada berbagai ukuran layar.

Itu adalah beberapa konsep penting dalam sistem grid Bootstrap. Dengan menggunakan sistem grid ini, kita dapat membuat tata letak yang responsif dan menyesuaikan tampilan elemen-elemen pada berbagai ukuran layar dengan mudah.

Contoh Penggunaan dan Penjelasannya

Sekarang kita akan mencoba penggunaan grid yang dengan bantuan Bootstrap. Berikut langkah-langkahnya:

1. Masuklah kembali ke web getbootstrap.com.



2. carilah "Grid" di Kolom search lalu klik enter. Maka kita akan diarahkan pada halaman web yang berisi tentang grid.

A screenshot of the Bootstrap Grid system documentation page. The left sidebar has a dark theme with categories like Sass, Options, Color, Color modes, Components, CSS variables, Optimize, Layout (Breakpoints, Containers, Grid), Content (Reboot, Typography, Images, Tables), and a search bar. The main content area has a light background. It features a section titled "Grid system" with a sub-section "Example #". The "Example #" section contains text about the grid system's purpose and how it works, along with a callout for "Get 10 Free Images From Adobe Stock". On the right side, there's a sidebar titled "On this page" with a list of topics including Example, How it works, Grid options, Auto-layout columns, Equal-width, Setting one column width, Variable width content, Responsive classes, All breakpoints, Stacked to horizontal, Mix and match, Row columns, Nesting, CSS, Sass variables, Sass mixins, Example usage, Customizing the grid, Columns and gutters, and Grid tiers.

3. Carilah layout atau tata letak grid yang kita inginkan. Misalnya pada praktek kali ini, kita akan menggunakan layout "Row columns".

Row columns

Use the responsive `.row-cols-*` classes to quickly set the number of columns that best render your content and layout. Whereas normal `.col-*` classes apply to the individual columns (e.g., `.col-md-4`), the row columns classes are set on the parent `.row` as a shortcut. With `.row-cols-auto` you can give the columns their natural width.

Use these row columns classes to quickly create basic grid layouts or to control your card layouts.

Column	Column
Column	Column

HTML



```
<div class="container text-center">
  <div class="row row-cols-2">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

4. Salinlah kode program yang terletak di kode bawah contohnya.

```
<div class="container text-center">
  <div class="row row-cols-2">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

5. Tempelah pada file html yang tentunya telah dihubungkan dengan tautan bootstrap, baik secara offline maupun online.

```

41 </nav>
42
43 <nav class="navbar bg-success">
44   <div class="container-fluid">
45     <form class="d-flex" role="search">
46       <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
47       <button type="button" class="btn btn-outline-light">Search</button>
48     </form>
49   </div>
50 </nav>
51
52 <div class="container text-center">
53   <div class="row row-cols-2">
54     <div class="col">Column</div>
55     <div class="col">Column</div>
56     <div class="col">Column</div>
57     <div class="col">Column</div>
58   </div>
59 </div>
60
61 </body>
62 </html>

```

6. Maka hasilnya akan seperti berikut:



Keterangan:

Jadi tulisan "Column" adalah isi konten yang sudah teratur dalam grid ini. kita bisa mengganti "konten" ini dengan konten yang kita inginkan

7. Untuk mengganti kontennya hapuslah tulisan "Column" lalu gantilah dengan konten yang kita inginkan.

```

<div class="container text-center">
  <div class="row row-cols-2">
    <div class="col">
      <div>
        
        <h3>HTML</h3>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
        <br>Nobis totam assumenda unde quod et vitae minus beatae, <br> quibusdam
        impedit laboriosam earum, <br> possimus consequatur quisquam dolore ex ipsa
        eaque cupiditate neque.</p>
      </div>
    </div>
    <div class="col">
      <div>
        
      </div>
    </div>
  </div>
</div>

```

```

<h3>CSS</h3>
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
<br>Nobis totam assumenda unde quod et vitae minus beatae, <br> quibusdam
impedit laboriosam earum, <br> possimus consequatur quisquam dolore ex ipsa
eaque cupiditate neque.</p>
</div>
</div>
<div class="col">
<div>

<h3>JavaScrip</h3>
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
<br>Nobis totam assumenda unde quod et vitae minus beatae, <br> quibusdam
impedit laboriosam earum, <br> possimus consequatur quisquam dolore ex ipsa
eaque cupiditate neque.</p>
</div>
</div>
<div class="col">
<div>

<h3>JQuery</h3>
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
<br> Nobis totam assumenda unde quod et vitae minus beatae, <br> quibusdam
impedit laboriosam earum, <br> possimus consequatur quisquam dolore ex ipsa
eaque cupiditate neque.</p>
</div>
</div>
</div>
</div>

```

8. Hasil nya akan menjadi seperti berikut:



Lorem ipsum dolor sit amet consectetur adipisicing elit.
 Nobis totam assumenda unde quod et vitae minus beatae,
 quibusdam impedit laboriosam earum,
 possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque.



Lorem ipsum dolor sit amet consectetur adipisicing elit.
 Nobis totam assumenda unde quod et vitae minus beatae,
 quibusdam impedit laboriosam earum,
 possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque.



Lorem ipsum dolor sit amet consectetur adipisicing elit.
 Nobis totam assumenda unde quod et vitae minus beatae,
 quibusdam impedit laboriosam earum,
 possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque.

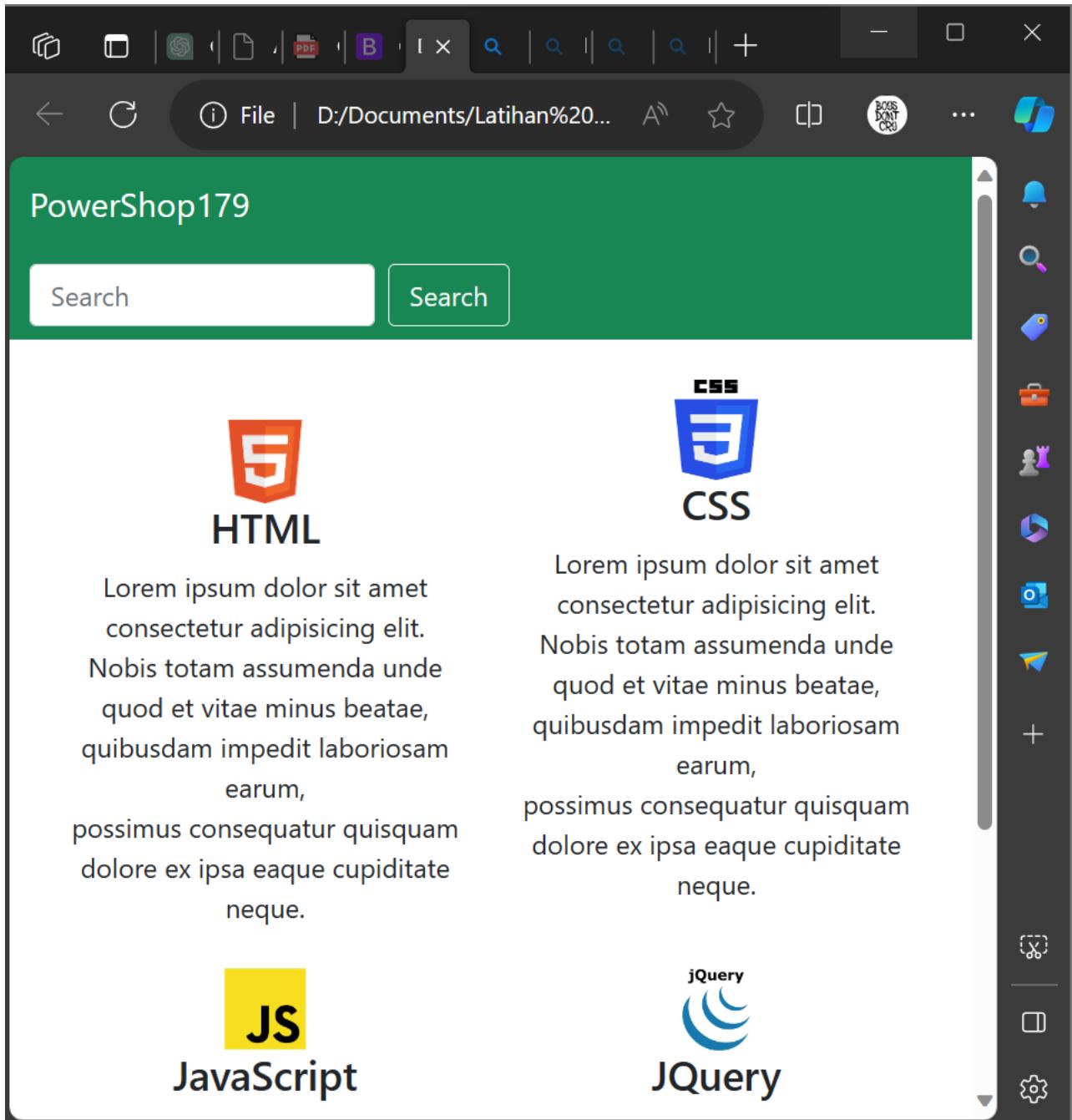


Lorem ipsum dolor sit amet consectetur adipisicing elit.
 Nobis totam assumenda unde quod et vitae minus beatae,
 quibusdam impedit laboriosam earum,
 possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque.

Keterangan:

Jadi Grid tadi mengatur Row dan Column dimana terdapat 2 baris berisi 4 kolom sebagai kontennya. Kontennya tadi kita ganti dengan konten yang kita inginkan

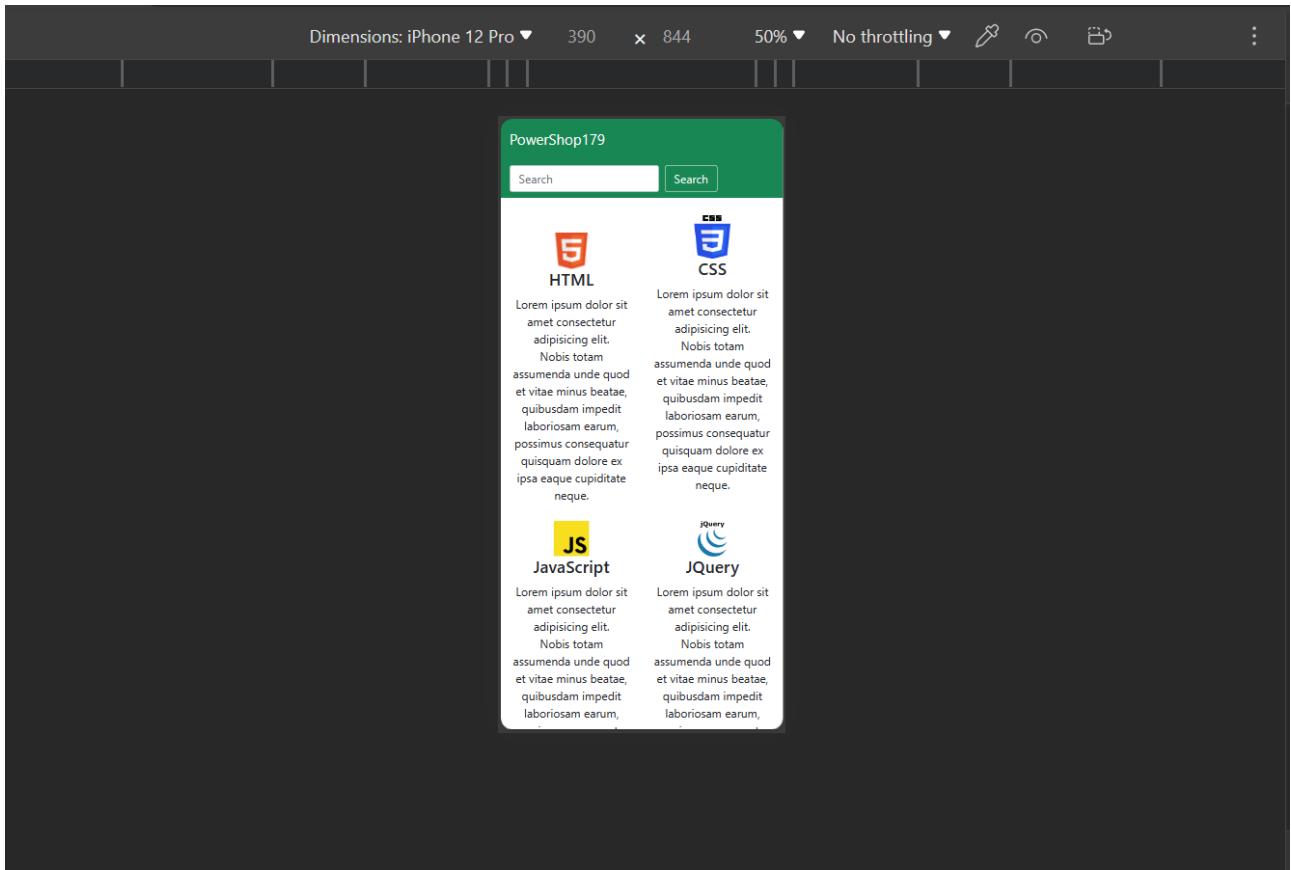
- Apabila layar yang kita kecilkan, maka grid akan menyesuaikan ukurannya seperti pada gambar ini



Keterangan:

Jadi grid ini sudah mengatur mengenai penyesuaian ukuran layar kita, sehingga, dapat menyesuaikan di segala ukuran monitor.

- Apabila kita melihatnya di perangkat, maka akan terlihat seperti ini:



Keterangan:

Jadi karena grid yang disediakan oleh bootstrap telah mendukung fitur responsive, maka ketika user melihat tampilan web kita di ponsel, maka tata letak akan mengikutinya sesuai dengan ukuran ponsel, seperti yang terlihat pada gambar di atas

Contoh Penggunaan `col-md` , `col-lg` dan `col-sm`

Kelas `col-md` , `col-lg` , dan `col-sm` adalah kelas-kelas kolom dalam Bootstrap yang digunakan untuk meresponsifkan tata letak halaman web sesuai dengan ukuran layar perangkat. Bootstrap menggunakan sistem grid yang terdiri dari 12 kolom, dan kelas-kelas ini memungkinkan Anda mengontrol sejauh mana elemen-elemen tersebut harus meluas pada layar yang berbeda.

- `[col-sm]` : Digunakan untuk layar kecil (small), seperti pada perangkat mobile atau tablet dalam orientasi potret. Kolom ini akan berlaku untuk layar dengan lebar 576px atau lebih.
- `[col-md]` : Digunakan untuk layar sedang (medium), yang mencakup perangkat seperti tablet dalam orientasi landscape. Kolom ini akan berlaku untuk layar dengan lebar 768px atau lebih.
- `[col-lg]` : Digunakan untuk layar besar (large), yang mencakup desktop. Kolom ini akan berlaku untuk layar dengan lebar 992px atau lebih.

Contoh:

Sekarang kita akan coba untuk membuat sebuah tampilan dengan menggunakan class diatas tadi. Berikut contohnya:

1. Bukalah kembali web getbootstrap lalu di kolom pencarian carilah "grid".

The screenshot shows the Bootstrap website's search interface. A search bar at the top contains the word 'grid'. Below it, a dropdown menu titled 'Grid system' is open, showing a list of items: 'Example', 'How it works', 'Grid options', 'Auto-layout columns', and 'Responsive classes'. To the right of this dropdown, another section titled 'CSS Grid' is visible with its own list: 'How it works' and 'Key differences'. The background of the page shows the 'Get started with Bootstrap' landing page.

2. Cari dan copylah kembali codingan grid yang kita gunakan pada praktik sebelumnya.

The screenshot shows the Bootstrap documentation for 'Row columns'. The main heading is 'Row columns'. Below it, a text block explains: 'Use the responsive `.row-cols-*` classes to quickly set the number of columns that best render your content and layout. Whereas normal `.col-*` classes apply to the individual columns (e.g., `.col-md-4`), the row columns classes are set on the parent `.row` as a shortcut. With `.row-cols-auto` you can give the columns their natural width.' Below this, another text block says: 'Use these row columns classes to quickly create basic grid layouts or to control your card layouts.' At the bottom, there is a code preview area showing the HTML and CSS for creating a 2-column grid. The HTML code is:

```
<div class="container text-center">
  <div class="row row-cols-2">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

```

<div class="container text-center">
    <div class="row row-cols-2">
        <div class="col">Column</div>
        <div class="col">Column</div>
        <div class="col">Column</div>
        <div class="col">Column</div>
    </div>
</div>

```

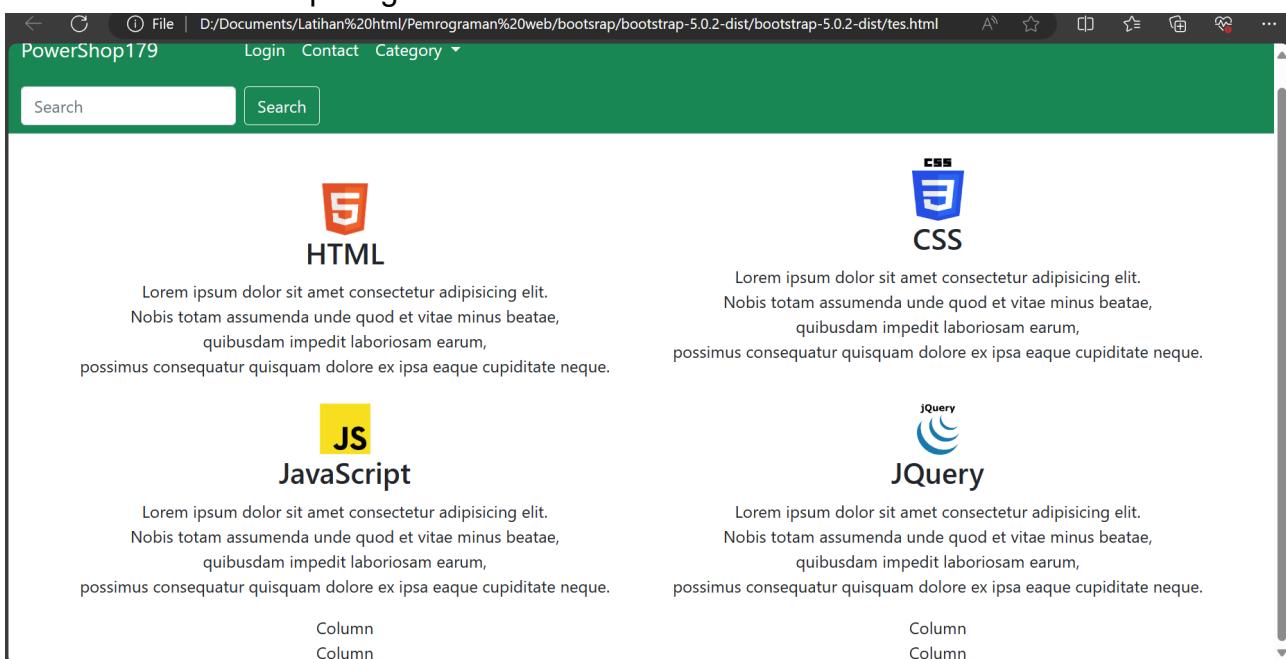
3. Tempelkanlah codingan tadi di visual studio kode pada file tempat kita mempraktekkan grid pada praktikum sebelumnya, agar kita bisa menggunakan satu file saja untuk praktek.

```

<div class="container text-center">
    <div class="row row-cols-2">
        <div class="col">Column</div>
        <div class="col">Column</div>
        <div class="col">Column</div>
        <div class="col">Column</div>
    </div>
</div>

```

4. Maka akan terlihat seperti gambar di bawah ini:



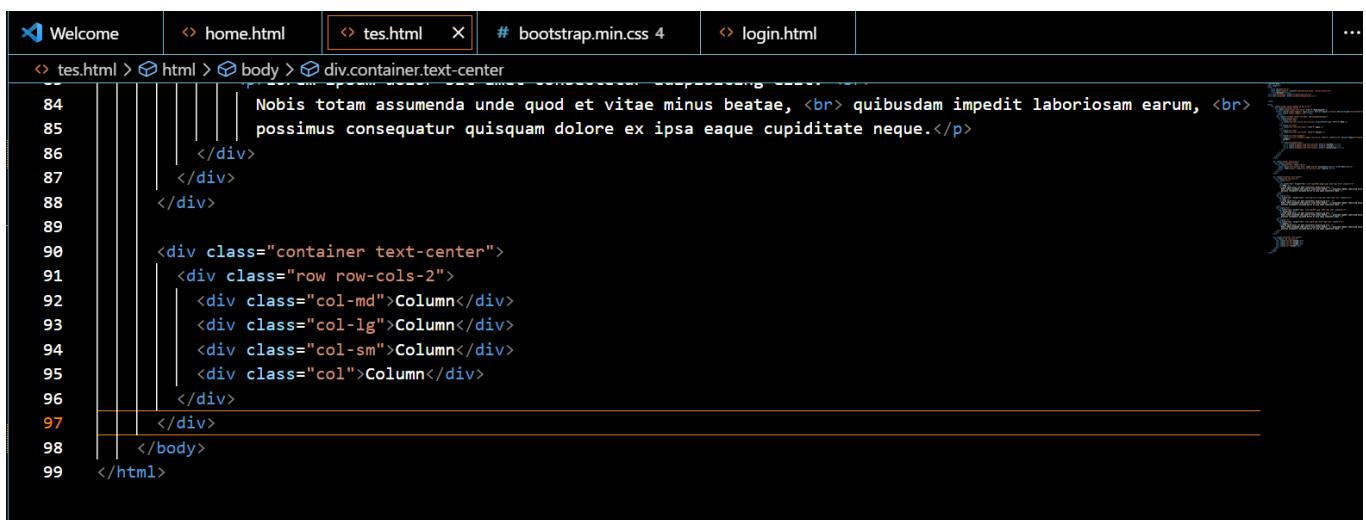
Keterangan:

Perhatikanlah di bawah kolom JavaScript dan JQuery, terdapat tulisan column sebanyak 4 dan bersusun hal ini dikarenakan pada kodingan tadi, teks yang dipakai sebagai template

di bootstrap adalah "Column", nanti pada tahap selanjutnya kita akan mengganti "Column" itu dengan text atau item yang kita inginkan.

5. Sekarang, pada codingan yang kita ambil dari web bootstrap tadi pada bagian `<div class="col">` yang pertama, tambahkanlah -md sehingga menjadi `<div class="col-md">`. Kemudian pada `<div class="col">` yang kedua, tambahkanlah - lg sehingga menjadi `<div class="col-lg">`, dan terakhir pada `<div class="col">` yang ketiga, tambahkanlah -sm sehingga menjadi `<div class="col-sm">`, sementara pada `<div class="col">` yang terakhir kita biarkan saja seperti bawaan.

```
<div class="container text-center">
  <div class="row row-cols-2">
    <div class="col-md">Column</div>
    <div class="col-lg">Column</div>
    <div class="col-sm">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```



The screenshot shows a code editor interface with several tabs at the top: "Welcome", "home.html", "tes.html", "# bootstrap.min.css 4", "login.html", and "...". The "tes.html" tab is active. Below the tabs, the file structure is shown as a tree: "tes.html" > "html" > "body" > "div.container.text-center". The code itself is a snippet from Bootstrap's grid system. It starts with a container and a row with two columns. The first column contains a paragraph of Latin text. The second column contains a nested container with a row of four columns. The first three columns in this row have specific Bootstrap classes: col-md, col-lg, and col-sm, while the fourth column is a standard col. The entire code block is numbered from 84 to 99.

```
84 |   Nobis totam assumenda unde quod et vitae minus beatae, <br> quibusdam impedit laboriosam earum, <br>
85 |   possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque.</p>
86 |
87 |
88 |
89 |
90 |   <div class="container text-center">
91 |     <div class="row row-cols-2">
92 |       <div class="col-md">Column</div>
93 |       <div class="col-lg">Column</div>
94 |       <div class="col-sm">Column</div>
95 |       <div class="col">Column</div>
96 |
97 |     </div>
98 |
99 |   </div>
  </body>
</html>
```

6. Maka hasilnya akan menjadi seperti ini:

Sebelum:

Column

Column

Column

Column

Sesudah:

Column

Column

Column

Column

Keterangan:

- `[col-md]` : Dengan menambahkan -md pada codingan tadi akan membuat kolom tersebut mengambil sebagian besar lebar tersedia (12 kolom) pada layar dengan lebar medium (768px atau lebih).
- `[col-lg]` : Sama seperti col-md , tetapi ditujukan untuk layar lebar (large, 992px atau lebih).
`[col-sm]` : Sama seperti col-md , tetapi ditujukan untuk layar kecil (small, 576px atau lebih).

7. Tambahkanlah teks ataupun item dari yang kita inginkan dengan cara hapuslah teks "Column" tadi, dan ditempat teks "Column" yang sudah dihapus, tempelkan lah item atau teks yang kita inginkan

```
<div class="container text-center">
    <div class="row row-cols-2">
        <div class="col-md">
            <div>
                
                <h3>HTML</h3>
                <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
<br>
                    Nobis totam assumenda unde quod et vitae minus beatae,
<br> quibusdam impedit laboriosam earum, <br>
                    possimus consequatur quisquam dolore ex ipsa eaque
cupiditate neque.</p>
            </div>
        </div>
        <div class="col-lg">
            <div>
                
                <h3>CSS</h3>
                <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
<br>
                    Nobis totam assumenda unde quod et vitae minus beatae,
<br> quibusdam impedit laboriosam earum, <br>
                    possimus consequatur quisquam dolore ex ipsa eaque
cupiditate neque.</p>
            </div>
        </div>
        <div class="col-sm">
            <div>
                
                <h3>JavaScript</h3>
                <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
<br>
```

```

        Nobis totam assumenda unde quod et vitae minus beatae,
<br> quibusdam impedit laboriosam earum, <br>
            possimus consequatur quisquam dolore ex ipsa eaque
cupiditate neque.</p>
        </div>
    </div>
<div class="col">
    <div>
        
            <h3>JQuery</h3>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
<br>
        Nobis totam assumenda unde quod et vitae minus beatae,
<br> quibusdam impedit laboriosam earum, <br>
            possimus consequatur quisquam dolore ex ipsa eaque
cupiditate neque.</p>
        </div>
    </div>
</div>
</div>

```

```

<div class="container text-center">
    <div class="row row-cols-2">
        <div class="col-md">
            <div>
                
                <h3>HTML</h3>
                <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. <br>
                    Nobis totam assumenda unde quod et vitae minus beatae, <br> quibusdam impedit laboriosam earum, <br>
                    possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque.</p>
            </div>
        </div>
        <div class="col-lg">
            
            <h3>CSS</h3>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. <br>
                Nobis totam assumenda unde quod et vitae minus beatae, <br> quibusdam impedit laboriosam earum, <br>
                possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque.</p>
        </div>
        <div class="col-sm">
            
            <h3>JavaScript</h3>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. <br>
                Nobis totam assumenda unde quod et vitae minus beatae, <br> quibusdam impedit laboriosam earum, <br>
                possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque.</p>
        </div>
        <div class="col">
            
            <h3>JQuery</h3>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. <br>

```

8. Maka hasilnya akan seperti ini:

 HTML	 CSS	 JavaScript	 JQuery
<p> Lorem ipsum dolor sit amet consectetur adipisicing elit. Nobis totam assumenda unde quod et vitae minus beatae, quibusdam impedit laboriosam earum, possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque.</p>	<p> Lorem ipsum dolor sit amet consectetur adipisicing elit. Nobis totam assumenda unde quod et vitae minus beatae, quibusdam impedit laboriosam earum, possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque.</p>	<p> Lorem ipsum dolor sit amet consectetur adipisicing elit. Nobis totam assumenda unde quod et vitae minus beatae, quibusdam impedit laboriosam earum, possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque.</p>	<p> Lorem ipsum dolor sit amet consectetur adipisicing elit. Nobis totam assumenda unde quod et vitae minus beatae, quibusdam impedit laboriosam earum, possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque.</p>

Keterangan:

Jadi pada praktek sebelumnya tanpa adanya tambahan pada class bootstrap tadi maka elemen akan menyesuaikan lebar pada pembungkus menjadi full, sedangkan untuk praktek yang saat ini kita tampilkan, item dan teks nya menyesuaikan ukurannya sesuai dengan tambahannya. Untuk penjelasan ukurannya **Penjelasannya telah kami sediakan di langkah ke-6.**

Penggunaan Tema

Dalam Bootstrap, tema dasarnya terdiri dari sejumlah variabel CSS yang mengontrol aspek-aspek seperti warna, ukuran teks, jarak, dan banyak lagi. Untuk melakukan kostumisasi tema, kita dapat mengganti nilai-nilai variabel ini sesuai dengan preferensi kita.

Berikut adalah langkah-langkah umum untuk melakukan kostumisasi tema Bootstrap:

- Menentukan gaya desain:** Pertama, tentukan gaya desain yang ingin kita terapkan pada situs web kita. Apakah kita ingin tampilan yang lebih minimalis, warna yang cerah, atau mungkin tampilan yang lebih kustom dengan elemen desain yang unik? Menentukan gaya desain akan membantu kita memutuskan bagaimana mengubah tema Bootstrap.
- Menggunakan Sass atau CSS:** Bootstrap menyediakan versi Sass (Syntactically Awesome Style Sheets) dari file sumbernya. Sass adalah bahasa pemrograman yang memungkinkan kita menulis CSS dengan sintaks yang lebih kuat dan fleksibel. Jika kita memiliki pengetahuan tentang Sass, kita dapat mengunduh versi Sass Bootstrap dan mengedit variabel-variabelnya. Jika kita tidak familiar dengan Sass , kita masih dapat mengedit file CSS Bootstrap langsung.
- Membuat file kustom:** Buat file kustom yang akan berisi kostumisasi tema kita. Dalam file ini, kita dapat menetapkan nilai-nilai variabel yang ingin kita ubah. Misalnya, kita dapat mengubah warna primer, warna latar belakang, ukuran huruf, dan lain sebagainya.
- Mengganti variabel:** Temukan variabel yang relevan dalam file sumber Bootstrap dan ubah nilainya sesuai dengan preferensi kita. Misalnya, jika kita ingin mengubah warna

primer, cari variabel `$primary-color` atau sejenisnya dalam file sumber dan ubah nilainya menjadi warna yang diinginkan.

5. **Mengompilasi tema kustom:** Setelah kita selesai mengedit file kustom, kita perlu mengompilasinya ke dalam file CSS yang dapat digunakan pada situs web kita. Jika kita menggunakan Sass, kita perlu mengompilasi file Sass menjadi CSS . Jika kita mengedit file CSS langsung, kita dapat menggunakan file tersebut langsung.
6. **Menerapkan tema kustom:** Setelah kita memiliki file CSS tema kustom, kita perlu menggantikan file Bootstrap default dengan file tema kustom kita pada situs web kita. Pastikan untuk menghubungkan file CSS tema kustom kita setelah file Bootstrap default sehingga kostumisasi kita akan ditimpa pada tema default. Kita dapat membuat tema kostuminasi Bootstrap yang sesuai dengan kebutuhan dan gaya desain kita. Penting untuk diingat bahwa jika kita menggunakan versi Bootstrap yang lebih baru, beberapa variabel atau struktur file mungkin telah berubah.

Kita dapat membuat tema kostuminasi Bootstrap yang sesuai dengan kebutuhan dan gaya desain kita. Penting untuk diingat bahwa jika kita menggunakan versi Bootstrap yang lebih baru, beberapa variabel atau struktur file mungkin telah berubah.

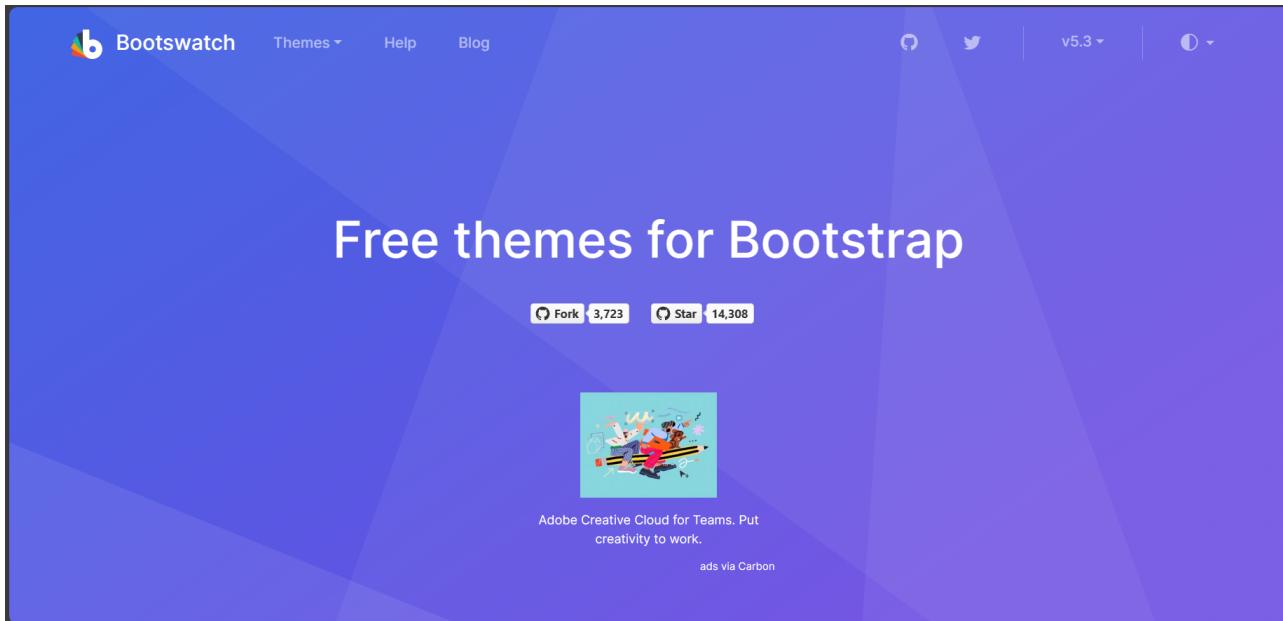
Contoh Penggunaan Tema dalam Bootstrap

1. Jadi misalnya kita ingin memberikan tema pada website yang telah kita buat sebelumnya

The screenshot shows a website layout with a green header bar. The header contains the text "PowerShop179" and navigation links for "Login", "Contact", and "Category". Below the header, there are four main content sections, each featuring a logo and text. The first section is for "HTML" (yellow square icon), the second for "CSS" (blue square icon), the third for "JS" (yellow square icon), and the fourth for "JQuery" (blue circular icon). Each section contains placeholder text in Latin.

Section	Icon	Text Placeholder
HTML	Yellow square icon	Placeholder text: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Nobis totam assumenda unde quod et vitae minus beatae, quibusdam impedit laboriosam earum, possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque."
CSS	Blue square icon	Placeholder text: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Nobis totam assumenda unde quod et vitae minus beatae, quibusdam impedit laboriosam earum, possimus consequatur quisquam dolore ex ipsa eaque cupiditate neque."
JS	Yellow square icon	Placeholder text: "Placeholder text for JS section"
JQuery	Blue circular icon	Placeholder text: "Placeholder text for JQuery section"

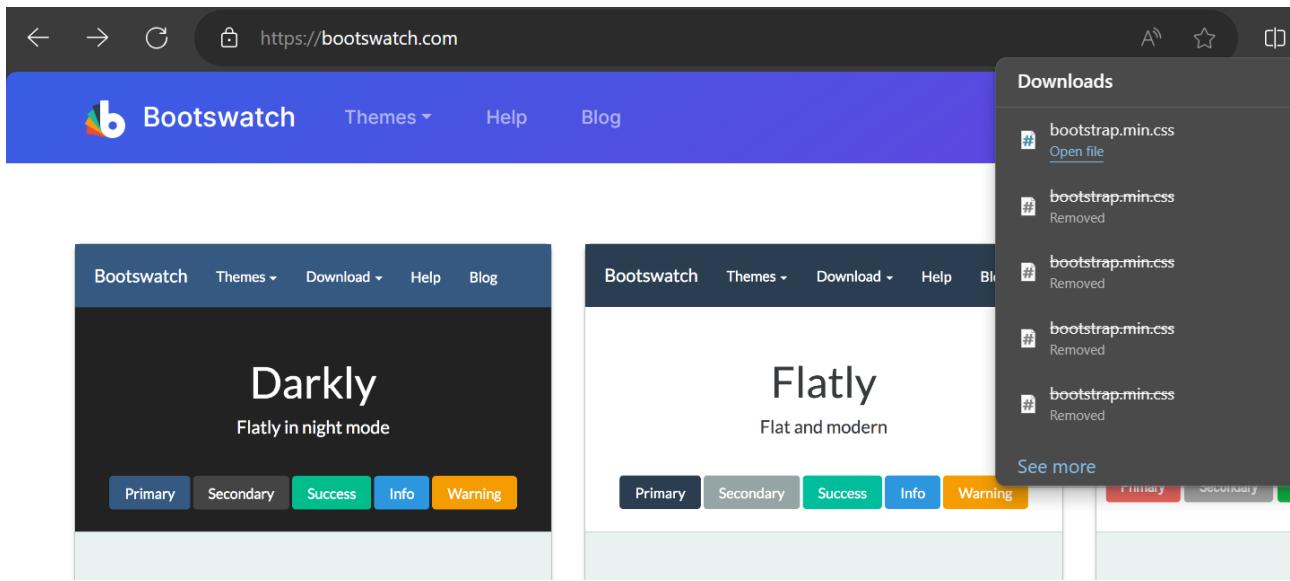
2. Bukalah website bootswatch.com dimana website ini menyediakan tema tema yang dapat digunakan untuk web bootstrap kia.



3. Scroll lah ke bawah hingga kita menemukan kumpulan tema tema yang telah disediakan oleh website ini

This screenshot shows a scroll view of the Bootswatch theme library. The top navigation bar is identical to the homepage. Below it, three theme cards are visible: 'Darkly' (dark mode), 'Flatly' (flat and modern), and 'Journal' (crisp like new paper). Each card has a preview image, a title, a subtitle, a color palette, and download buttons for 'Preview' and 'Download'. The URL 'https://bootswatch.com/flatly/' is visible at the bottom of the scroll area.

4. Misalnya kita akan memilih tema "Darkly", oleh karena itu, klik download pada bagian bawah contoh tema tersebut.



5. Jika sudah mendownloadnya maka kita telah mendapatkan file css untuk tema tersebut

Name	Date modified	Type	Size
css	10/22/2023 5:04 PM	File folder	
img	11/15/2023 4:38 PM	File folder	
js	10/22/2023 5:04 PM	File folder	
home	11/15/2023 1:01 AM	Microsoft Edge HTM...	12 KB
login	11/15/2023 4:04 PM	Microsoft Edge HTM...	1 KB
tes	11/16/2023 4:13 AM	Microsoft Edge HTM...	6 KB
# bootstrap.min	11/16/2023 4:16 AM	CSS Source File	229 KB

6. Pindahkanlah file css tersebut ke dalam folder yang berisi file html yang akan kita berikan tema

Name	Date modified	Type	Size
css	10/22/2023 5:04 PM	File folder	
img	11/15/2023 4:38 PM	File folder	
js	10/22/2023 5:04 PM	File folder	
home	11/15/2023 1:01 AM	Microsoft Edge HTM...	12 KB
login	11/15/2023 4:04 PM	Microsoft Edge HTM...	1 KB
tes	11/16/2023 4:13 AM	Microsoft Edge HTM...	6 KB
# bootstrap.min	11/16/2023 4:16 AM	CSS Source File	229 KB

7. Sekarang, masuklah ke dalam file html yang akan kita berikan tema bootstrap, lalu panggilah secara eksternal file css tema tadi ke dalam file html tersebut

```

<tes.html> <html> <head> <link href="#bootstrap.min.css" rel="stylesheet"/>
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Document</title>
7       <link rel="stylesheet" href="css/bootstrap.min.css">
8       <link rel="stylesheet" href="css/bootstrap-grid.min.css">
9       <link rel="stylesheet" href="b">
10      # bootstrap.min.css
11    </head>

```

8. Maka hasilnya adalah sebagai berikut dengan demikian kita sudah berhasil memberikan tema gelap pada tampilan html kita menggunakan tema dark pada bootstrap

The screenshot shows a dark-themed website layout. At the top, there's a green header bar with the text "PowerShop179" and navigation links for "Login", "Contact", and "Category". Below the header is a search bar with a placeholder "Search" and a "Search" button. The main content area has four dark cards arranged in a grid. The first card on the left is titled "HTML" with an orange "5" icon and contains placeholder text. The second card on the right is titled "CSS" with a blue "3" icon and contains placeholder text. The third card at the bottom left is titled "JS" with a yellow "S" icon and contains placeholder text. The fourth card at the bottom right is titled "JQuery" with a blue circular logo and contains placeholder text. Each card has a small "Lorem ipsum" placeholder text above the main content.

dengan demikian kita sudah berhasil memberikan tema gelap pada tampilan html kita menggunakan tema dark pada bootstrap

Modifikasi Template Bootstrap

Kostumasisasi dalam bootstrap adalah menggunakan template bootstrap kemudian kita akan memodifikasi template tersebut sesuai dengan keinginan kita.

Template dalam Bootstrap adalah struktur dasar atau kerangka kerja yang telah dirancang sebelumnya dan dapat digunakan sebagai dasar untuk membangun halaman web. Bootstrap adalah sebuah framework front-end yang populer digunakan untuk pengembangan web responsif. Dalam konteks Bootstrap, template menyediakan struktur HTML dan gaya CSS dasar yang dapat digunakan sebagai dasar untuk membangun halaman web yang responsif dan mudah diatur.

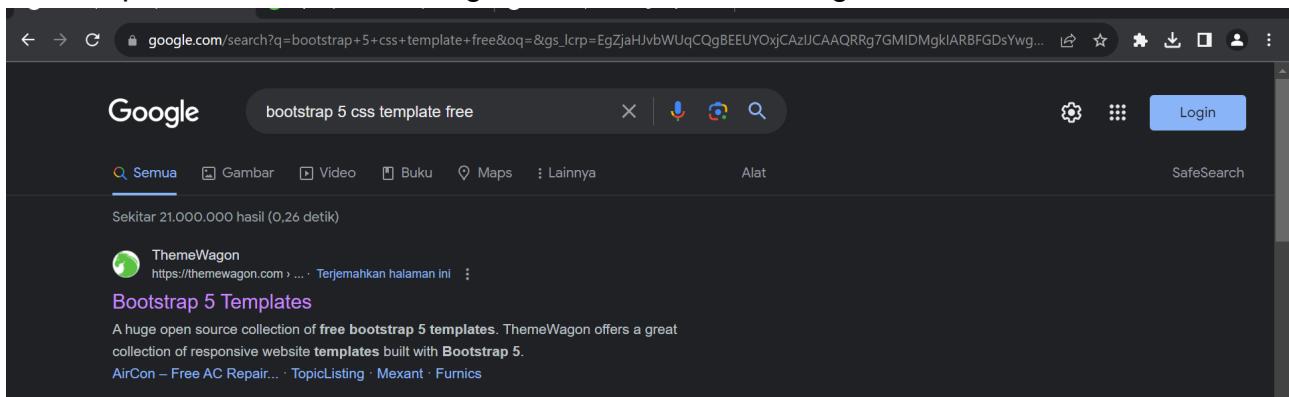
Dengan menggunakan template Bootstrap, kita dapat menghemat waktu dan usaha dalam merancang tata letak dan gaya dasar dari suatu proyek web. Template ini biasanya sudah mencakup elemen-elemen umum seperti navigasi, grid system, tombol, formulir, dan banyak

lagi. Dengan cara ini, kita dapat fokus pada konten dan fitur unik dari situs web kita tanpa harus memulai dari awal.

Template dalam Bootstrap juga mendukung konsep responsivitas, yang berarti halaman web atau aplikasi yang dibangun dengan menggunakan template ini dapat menyesuaikan diri dengan berbagai ukuran layar, mulai dari perangkat seluler hingga desktop, tanpa perlu penyesuaian tambahan. Ini membuat pengembangan web lebih efisien dan memastikan pengalaman pengguna yang konsisten di berbagai perangkat.

Berikut cara mengkostumisasi template bootstrap:

1. Pertama adalah kita akan membuka sebuah website yang menyediakan template bootstrap. Kali ini web akan kita gunakan adalah themewagon.com



2. Masuklah ke dalam website tersebut

The screenshot shows the homepage of themewagon.com. At the top, there's a navigation bar with links like 'Admin & Dashboard', 'Bootstrap 5', 'eCommerce', 'Tailwind CSS', 'Landing Pages', 'Business & Corporate', 'Portfolio', 'Educational', and a prominent 'Bundle — Save 80%' button. Below the navigation, there's a search bar with the placeholder 'shop'. On the left, there's a sidebar with 'Browse Themes' and a 'Themes' section. The main content area displays a grid of five free Bootstrap templates. Each template card includes a preview image, the template name, a download count (e.g., 4931 Downloads), and a rating (e.g., ★★★★☆). The templates shown are: MultiShop (Free), CAKEZONE (Free), E Shopper (Free), Zay (Free), and Aroma (Free).

3. Carilah template yang ingin kita sesuaikan dengan cara mensearch ataupun menscroll hingga menemukan tema yang sesuai



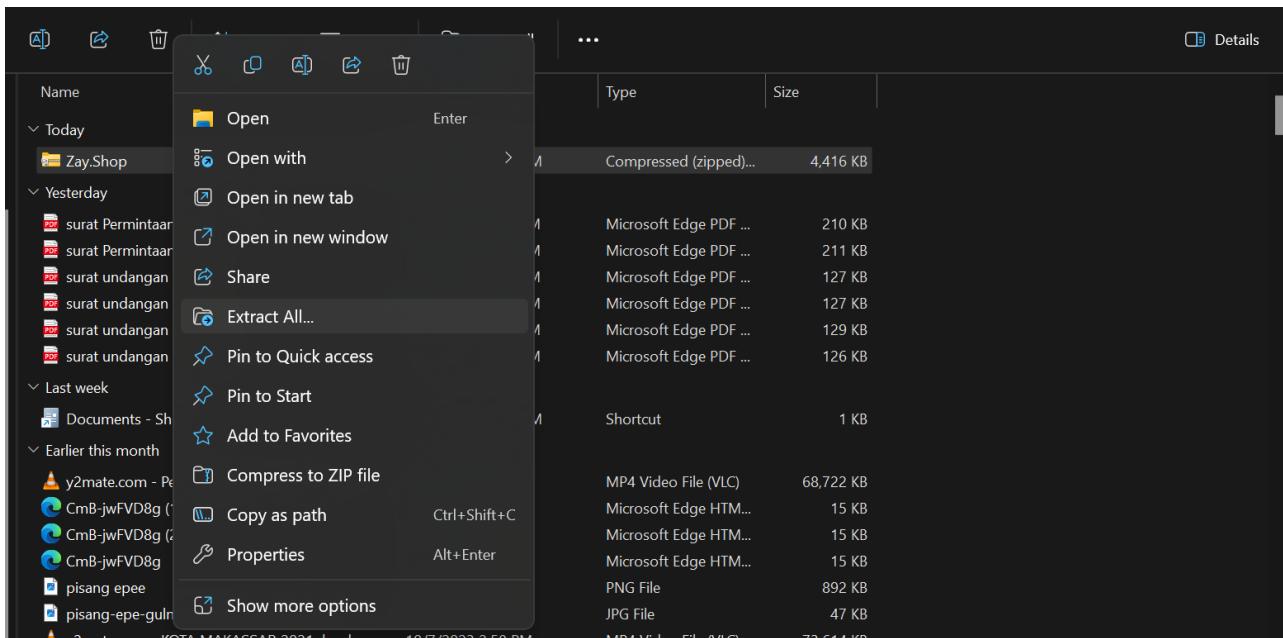
Browse Themes

Clear filter | Free (20) | Premium (0) | Filter (3) | Relevance

4. Jika sudah menemukan template yang kita inginkan, masuk ke template tersebut lalu klik download pada bagian kanan sebelah, contoh gambar dibawah ini.

5. Jika sudah mendownloadnya maka kita telah mendapatkan file template bootstrap kita dalam bentuk zip. File ini berisi asset gambar, style css, index tml, maupun javascript yang menyusun komponen template ini

6. Ekstraklah file tersebut hingga berbentuk seperti file biasa dengan cara klik kanan pada file zip tadi lalu pilih "Extract All".



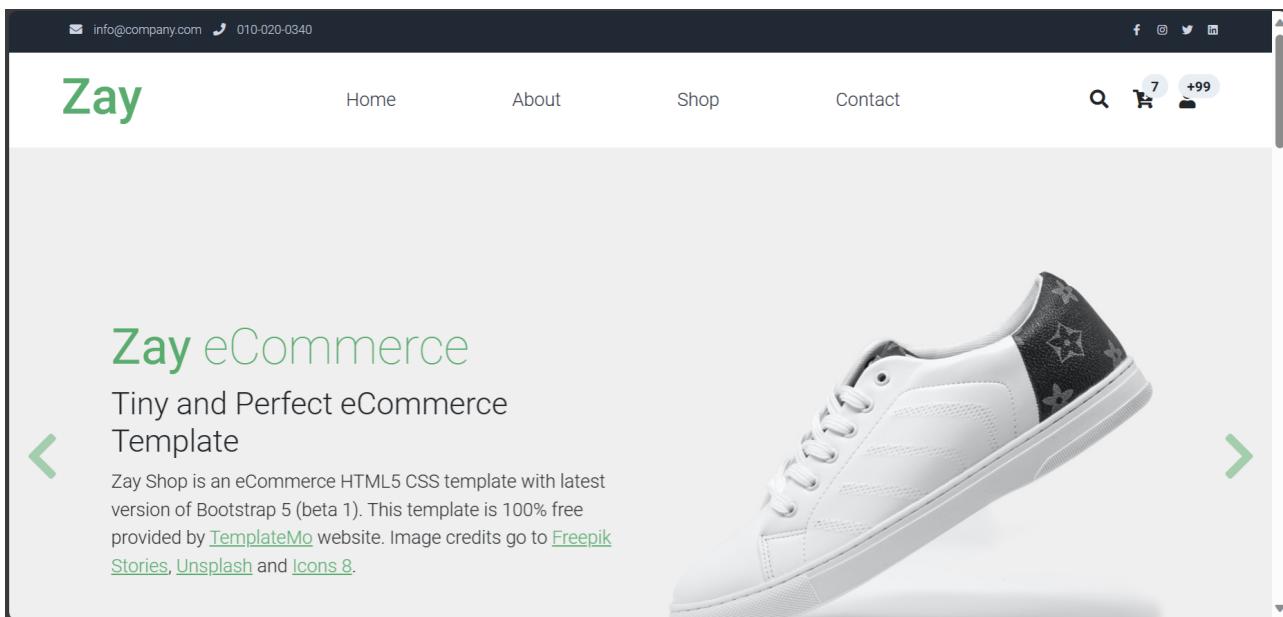
7. Jika proses ekstrak sudah selesai maka kita telah mendapatkan folder template tersebut dalam bentuk folder biasa



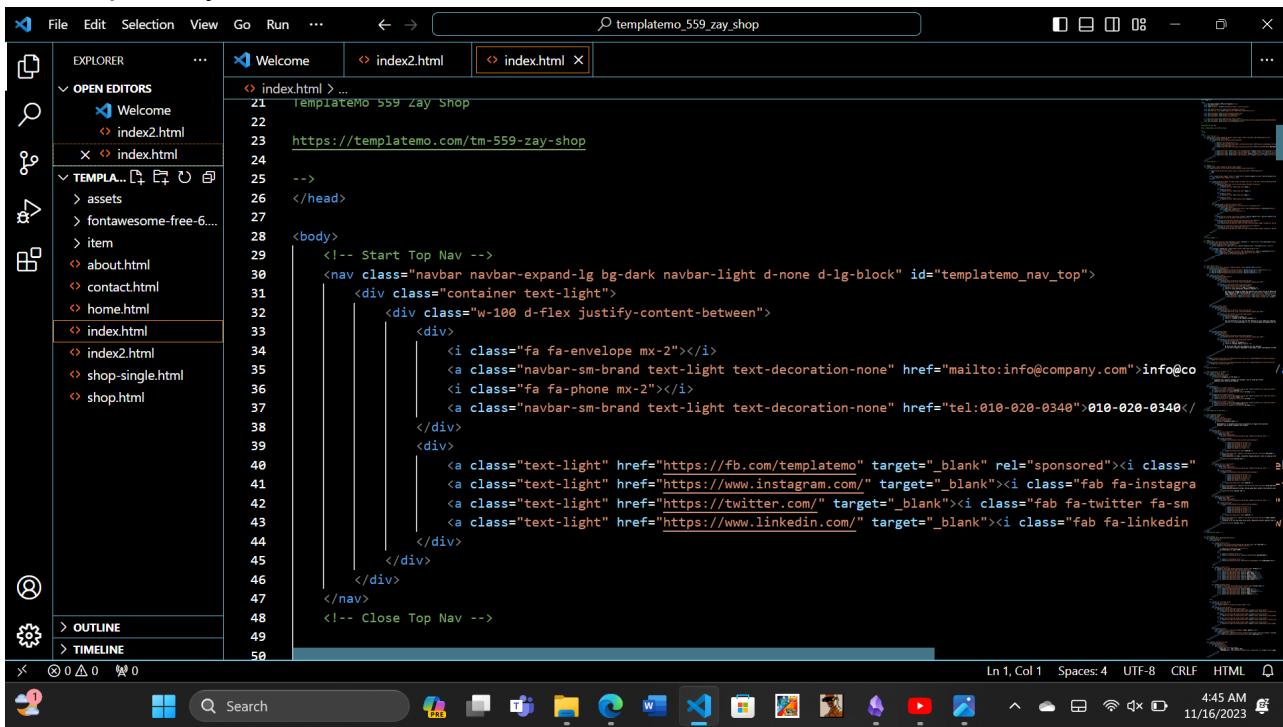
8. Masuklah ke dalam folder template yang telah diekstrak tadi

Name	Date modified	Type	Size
assets	10/18/2023 4:42 AM	File folder	
fontawesome-free-6.4.2-web	10/18/2023 9:42 AM	File folder	
item	10/18/2023 8:50 AM	File folder	
about	10/18/2023 4:42 AM	Microsoft Edge HTM...	21 KB
contact	10/18/2023 4:42 AM	Microsoft Edge HTM...	16 KB
home	10/23/2023 4:20 AM	Microsoft Edge HTM...	8 KB
index	10/18/2023 4:42 AM	Microsoft Edge HTM...	24 KB
index2	10/18/2023 8:58 AM	Microsoft Edge HTM...	1 KB
shop	10/18/2023 4:42 AM	Microsoft Edge HTM...	49 KB
shop-single	10/18/2023 4:42 AM	Microsoft Edge HTM...	58 KB

9. Masuklah ke dalam folder index.html. didalam sana terdapat tampilan template kita



10. Sekarang kita akan mengkostumasisasi tampilan web template ini. Caranya adalah buka file template nya di visual studio code



11. Mulailah mengganti konten-konten seperti teks, gambar dan lainnya seperti yang kita inginkan. Misalnya saya akan mengganti teks yang ada di gambar ini



12. Buka kembali visual studio dan masuk ke file index.html tersebut lalu carilah penyusun konten itu

```
:class="fa fa-envelope mx-2">>/i>
:class="navbar-sm-brand text-light text-decoration-none" href="mailto:info@company.com">info@company.com</a>
:class="fa fa-phone mx-2">>/i>
:class="navbar-sm-brand text-light text-decoration-none" href="tel:010-020-0340">010-020-0340</a>
```

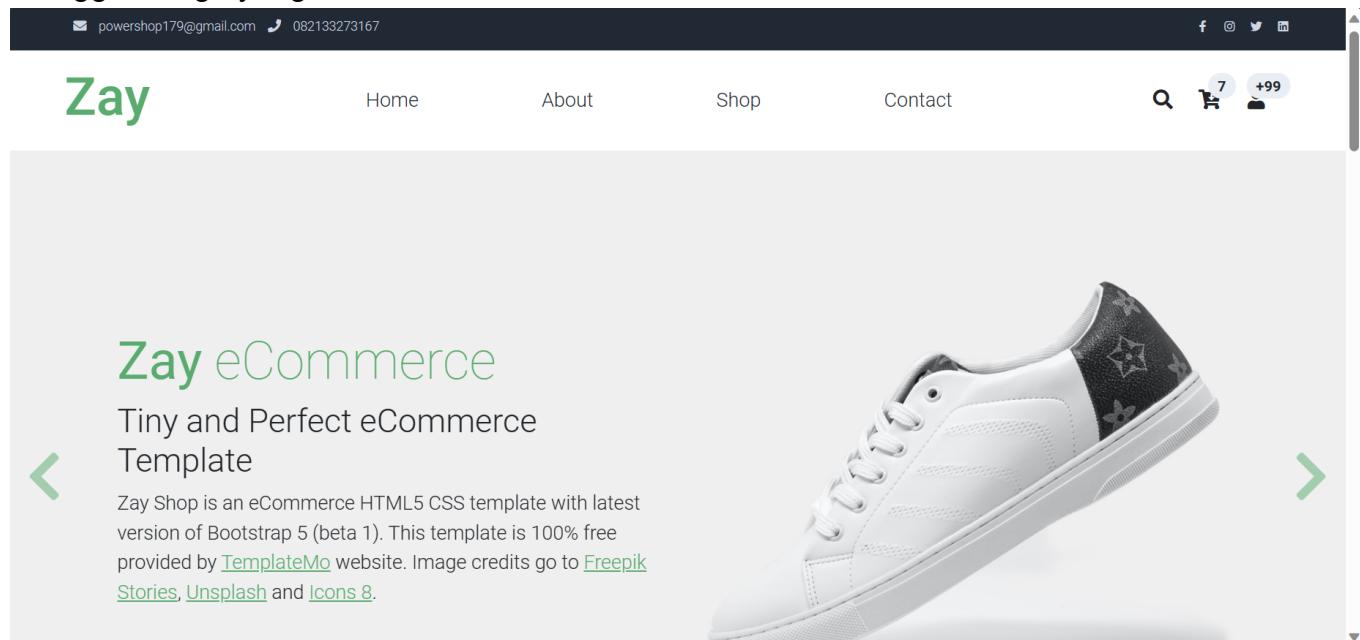
13. Gantilah kalimat dan tujuan email dan nomor telp itu dengan yang kita inginkan

```
<a class="navbar-sm-brand text-light text-decoration-none"  
href="mailto:powershop179@gmail.com">powershop179@gmail.com</a>  
<i class="fa fa-phone mx-2"></i>  
<a class="navbar-sm-brand text-light text-decoration-none"  
href="tel:082133273167">082133273167</a>
```

Maka hasilnya akan terlihat seperti dibawah ini:



14. Mulailah mengganti elemen-elemen lainnya sesuai keinginan kita. Misalnya kita ingin juga mengganti logo yang ada di website ini



```
<a class="navbar-brand text-success logo h1 align-self-center"  
href="index.html">  
Powershop179  
</a>
```

15. Hasil setelah diganti adalah seperti ini

Zay eCommerce

Tiny and Perfect eCommerce Template

Zay Shop is an eCommerce HTML5 CSS template with latest version of Bootstrap 5 (beta 1). This template is 100% free provided by [TemplateMo](#) website. Image credits go to [Freepik Stories](#), [Unsplash](#) and [Icons 8](#).



16. Sekarang kita akan mengganti Nama web kita yang ada di slide carousel

Zay eCommerce

Tiny and Perfect eCommerce Template

Zay Shop is an eCommerce HTML5 CSS template with latest version of Bootstrap 5 (beta 1). This template is 100% free provided by [TemplateMo](#) website. Image credits go to [Freepik Stories](#), [Unsplash](#) and [Icons 8](#).



```
<div class="col-lg-6 mb-0 d-flex align-items-center">
  <div class="text-align-left align-self-center">
    <h1 class="h1 text-success"><b>PowerShop179</b> eCommerce</h1>
    <h3 class="h2">Tiny and Perfect eCommerce Template</h3>
    <p>
      PowerShop179 is an eCommerce HTML5 CSS template with the latest version of Bootstrap 5 (beta 1).
      This template is 100% free provided by <a rel="sponsored" class="text-success" href="https://templatemo.com" target="_blank">TemplateMo</a> website. Image credits go to <a rel="sponsored" class="text-success" href="https://stories.freepik.com/" target="_blank">Freepik Stories</a>, <a rel="sponsored" class="text-success" href="https://unsplash.com/" target="_blank">Unsplash</a> and <a rel="sponsored" class="text-success" href="https://icons8.com/" target="_blank">Icons 8</a>.
    </p>
  </div>
```

```
</div>
```

Hasilnya akan menjadi seperti ini:

The screenshot shows the homepage of the Powershop179 eCommerce template. At the top, there is a navigation bar with links for Home, About, Shop, and Contact, along with a search icon, a shopping cart icon showing 7 items, and a user icon showing +99 notifications. The main content area features a large image of a white and black sneaker on the right. To the left of the sneaker, there is a title section with "Powershow179" in a large green font and "eCommerce" in a smaller green font. Below the title is a subtitle "Tiny and Perfect eCommerce Template" with left and right navigation arrows. A text block below the subtitle provides information about the template being an HTML5 CSS template with Bootstrap 5 (beta 1), 100% free, and credits to TemplateMo, Freepik Stories, Unsplash, and Icons 8.

17. bagaimana jika kita ingin mengganti gambar? Misalnya kita akan mengganti gambar yang ada di card ini

The screenshot shows three product cards from the Powershop179 template. Each card has a large image at the top, followed by a rating, price, title, a short description, and a reviews link. The first card is for a "Gym Weight" priced at \$240.00. The second card is for "Cloud Nike Shoes" priced at \$480.00. The third card is for "Summer Addides Shoes" priced at \$360.00. The products shown in the cards are a grey cylindrical weight, a black Apple Watch with a red woven band, and a Polaroid One Step camera.

Product	Rating	Price	Description	Reviews
Gym Weight	★★★★★	\$240.00	Lorem ipsum dolor sit amet, consectetur adipisicing elit. Sunt in culpa qui officia deserunt.	Reviews (24)
Cloud Nike Shoes	★★★★★	\$480.00	Aenean gravida dignissim finibus. Nullam ipsum diam, posuere vitae pharetra sed, commodo ullamcorper.	Reviews (48)
Summer Addides Shoes	★★★★★	\$360.00	Curabitur ac mi sit amet diam luctus porta. Phasellus pulvinar sagittis diam, et scelerisque ipsum lobortis nec.	Reviews (74)

18. Masuklah ke visual studio code dan ke index.html tadi lalu carilah kode penyusun untuk gambar ini, kemudian gantilah di bagian src gambar yang sudah ada dengan gambar yang kita inginkan

```
<a href="shop-single.html">  

```

```
</a>
```

Maka hasilnya akan menjadi seperti ini:

```
"assets/btc95.png" could not be found.
```

Demikianlah kostumisasi template bootstrap ini. Kesimpulanya kita dapat mengganti text, gambar, link tujuan, atau menambahkan elemen tertentu sesuai dengan kebutuhan kita. Ini tentunya sangat menghemat waktu dan tenaga, dimana kita hanya fokus pada konten yang kita sajikan dibanding harus menyusun grid, responsif, dan lainnya dari awal

Studi Kasus Pengaplikasian Bootstrap dalam Desain Web

Contoh Desain Web Yang Akan di Implementasikan:

```
"assets/bf-1.png" could not be found.
```

1. dalam desain web tersebut terdapat beberapa komponen, yaitu
 - Navbar, Hero Section, Button dll
2. Pertama-tama kita perlu Inisialisasikan class `container` agar dapat memberikan tata letak konten yang sesuai dengan desainnya.

```
<div class="container">  
</div>
```

Keterangan:

- pada `<div class="container">`: Kelas ini memberikan tata letak yang terkandung pada lebar tertentu (dalam pixel) dan ditengahkan di tengah halaman atau elemen yang memuatnya. Ini membantu dalam mengatur konten agar sesuai dengan standar desain Bootstrap.
3. Selanjutnya kita akan implementasikan navbarnya, bukalah kembali web getbootstrap lalu di kolom pencarian carilah "navbar".

The screenshot shows the Bootstrap documentation homepage. A search bar at the top has 'Navbar' typed into it. Below the search bar, a modal window titled 'Navbar' is open, listing several sub-sections: 'How it works', 'Supported content', 'Color schemes', 'Containers', and 'Placement'. To the left of the modal is a sidebar containing a list of Bootstrap components, and to the right is a sidebar titled 'On this page' with various links.

4. Cari dan copylah codingan komponen navbar yang mungkin sesuai dengan desainnya.

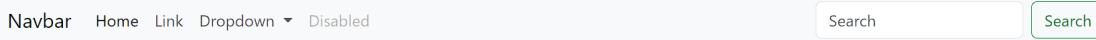
The screenshot shows the Bootstrap documentation for the 'Navbar' component. On the left is a sidebar with the 'Navbar' item selected. The main content area displays the 'HTML' code for a navbar. The code includes a navigation bar with a brand link, a toggler button, and a dropdown menu. The dropdown menu contains several items, including one that is active. The code uses Bootstrap's utility classes like 'bg-body-tertiary' and 'nav-item'.

```
<nav class="navbar navbar-expand-lg bg-body-tertiary"> <div class="container-fluid"> <a class="navbar-brand" href="#">Navbar</a> <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"> <span class="navbar-toggler-icon"></span> </button> <div class="collapse navbar-collapse" id="navbarSupportedContent"> <ul class="navbar-nav me-auto mb-2 mb-lg-0"> <li class="nav-item"> <a class="nav-link active" aria-current="page" href="#">Home</a> </li> <li class="nav-item"> <a class="nav-link" href="#">Link</a> </li> <li class="nav-item dropdown"> <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false"> Dropdown </a> <ul class="dropdown-menu"> <li><a class="dropdown-item" href="#">Action</a></li> <li><a class="dropdown-item" href="#">Another action</a></li> <li> <hr class="dropdown-divider"> </li> <li> <a class="dropdown-item" href="#">Something else here</a></li> </ul> </li> <li class="nav-item"> <a class="nav-link disabled" aria-
```

```
disabled="true">Disabled</a> </li> </ul> <form class="d-flex" role="search">  
<input class="form-control me-2" type="search" placeholder="Search" aria-  
label="Search"> <button class="btn btn-outline-success"  
type="submit">Search</button> </form> </div> </div> </nav>
```

Keterangan:

1. [navbar] : Kelas ini menunjukkan bahwa elemen [<nav>] adalah bagian dari komponen navbar. Navbar adalah bagian dari Bootstrap yang menyediakan navigasi di bagian atas halaman web. Dengan memberikan kelas navbar , kita memberitahu Bootstrap untuk menerapkan gaya dan tata letak khusus yang terkait dengan navigasi.
2. [bg-body-tertiary] : Kelas ini memberikan warna latar belakang pada navbar. Dalam hal ini, warna latar belakangnya diberi warna sesuai dengan warna "hijau" Warna latar belakang ini dapat disesuaikan sesuai kebutuhan desain.
3. [container-fluid] : Kelas ini diterapkan pada elemen [<div>] yang membungkus elemen-elemen dalam navbar. Kelas ini memberikan padding yang sesuai dan membuat elemen-elemen di dalamnya menjaga lebar penuh dari layar (menggunakan grid system Bootstrap). Dengan kata lain, kontennya akan merespons secara baik pada berbagai lebar perangkat.
4. [d-flex] : Kelas ini mengubah elemen [<form>] menjadi kontainer flex. Flexbox adalah teknik tata letak yang kuat di CSS yang memudahkan pengaturan dan penataan elemen dalam satu atau dua dimensi. Dengan memberikan kelas d-flex , elemen form dan anak-anaknya dapat disusun secara fleksibel.
5. [me-2] : Ini adalah kelas Bootstrap yang memberikan margin kanan sebesar 2 pada elemen yang memilikinya. Dalam codingan ini, kelas ini diterapkan pada elemen input untuk memberikan jarak margin kanan.
6. Tempelkanlah kode program itu didalam class [container] sebelumnya pada halaman html terkait. Tentu saja dengan bootstrap yang sudah ditautkan, baik offline, maupun online.
7. Maka hasilnya akan seperti ini:



7. Sekarang pada bagian dalam div navbar-collapse hapus elemen form:

```
<form class="d-flex" role="search"> <input class="form-control me-2"  
type="search" placeholder="Search" aria-label="Search" /> <button class="btn  
btn-outline-success" type="submit">Search</button> </form>
```

```

<nav class="navbar navbar-expand-lg bg-body-tertiary">
    <div class="container-fluid"> <a class="navbar-brand" href="#">Navbar</a> <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"> <span class="navbar-toggler-icon"></span> </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav me-auto mb-2 mb-lg-0">
            <li class="nav-item"> <a class="nav-link active" aria-current="page" href="#">Home</a> </li>
            <li class="nav-item"> <a class="nav-link" href="#">Link</a> </li>
            <li class="nav-item dropdown"> <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false"> Dropdown </a>
                <ul class="dropdown-menu">
                    <li><a class="dropdown-item" href="#">Action</a></li>
                    <li><a class="dropdown-item" href="#">Another action</a></li>
                    <li>
                        <hr class="dropdown-divider">
                    </li>
                    <li><a class="dropdown-item" href="#">Something else here</a></li>
                </ul>
            </li>
            <li class="nav-item"> <a class="nav-link disabled" aria-disabled="true">Disabled</a> </li>
        </ul>

        <form class="d-flex" role="search"> <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search"> <button class="btn btn-outline-success" type="submit">Search</button> </form>
    </div>
</div>
</nav>

```

8. Hasilnya akan seperti ini:

Navbar Home Link Dropdown ▾ Disabled

9. Langkah Selanjutnya kita perlu mengganti list list serta nama logo pada navbarnya

```

<ul class="navbar-nav me-auto mb-2 mb-lg-0"> <li class="nav-item"> <a class="nav-link active fw-bold" aria-current="page" href="#">Home</a> </li>
<li class="nav-item"> <a class="nav-link text-muted" href="#">Discover</a> </li>
<li class="nav-item"> <a class="nav-link text-muted" href="#">Post a Job</a> </li>
<li class="nav-item"> <a class="nav-link text-muted" href="#">Contact Us</a> </li>
</ul>

```

Keterangan:

1. Disini kita mengganti text list item, dari sebelumnya default dari bootstrap ke desain yang akan diimplementasikan seperti: home,discover,post a job dan contact us
2. `fw-bold` dalam Bootstrap digunakan untuk memberikan tebal pada teks. Kelas ini merupakan bagian dari utilitas `font-weight` yang disediakan oleh Bootstrap, yang memungkinkan Anda untuk dengan cepat mengubah berat font teks dengan kelas `.fw-*`. Kelas `fw-bold` setara dengan berat font 700
3. `text-muted` digunakan untuk memberi gaya pada teks dengan warna yang redup. Kelas ini merupakan bagian dari utilitas `text-color` yang disediakan oleh Bootstrap, yang

memungkinkan Anda untuk dengan mudah mengubah warna teks dengan kelas `.text-*`. Kelas `[text-muted]` setara dengan warna `[$gray-600]`, yang merupakan warna abu-abu yang redup.

Jobless Home Discover Post a Job Contact Us

10. Kemudian tambah tombol `sign-up` di sebelah kanan, sama seperti form namun kita berikan tag anchor

```
<a href="/sign-up" class="btn btn-outline-dark rounded-0 fw-bold"> Sign Up </a>
```

Keterangan:

1. class `btn` digunakan untuk mengatur tampilan tombol. Class ini menambahkan beberapa stil ke tampilan tombol, seperti padding, margin, dan warna latar belakang.
2. `btn-outline-dark` digunakan untuk mengatur tampilan tombol dengan latar belakang transparan dan warna teks yang hitam.
3. `rounded-0` digunakan untuk mengatur bentuk tombol dengan sudut yang tidak ada (0).
4. `fw-bold` digunakan untuk mengatur font teks dengan berat font yang tebal.
5. `[text-muted]` digunakan untuk mengatur warna teks dengan warna yang redup.

Jobless Home Discover Post a Job Contact Us

Sign Up

11. Selanjutnya ganti background navbarnya dari `bg-body-tertiary` menjadi transparan dengan menggunakan class `bg-transparent`

```
<nav class="navbar navbar-expand-lg bg-transparent">
  <div class="container-fluid"> <a class="navbar-brand" href="#">Navbar</a> <button class="navbar-toggler"
    type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
    aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"> <span
      class="navbar-toggler-icon"></span> </button>
  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
      <li class="nav-item"> <a class="nav-link active fw-bold" aria-current="page" href="#">Home</a>
      </li>
      <li class="nav-item"> <a class="nav-link text-muted" href="#">Discover</a> </li>
      <li class="nav-item"> <a class="nav-link text-muted" href="#">Post a Job</a> </li>
      <li class="nav-item"> <a class="nav-link text-muted" href="#">Contact Us</a> </li>
    </ul>

    <a href="/sign-up" class="btn btn-outline-dark rounded-0 fw-bold"> Sign Up </a>
  </div>
</div>
</nav>
```

12. Sekarang kita akan menambahkan list navbarnya dengan menggunakan class `m-auto` sebelum

```
<div class="container">

    <nav class="navbar navbar-expand-lg bg-transparent">
        <div class="container-fluid">
            <a class="navbar-brand" href="#">Jobless</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
                aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item"> <a class="nav-link active" aria-current="page" href="#">Home</a> </li>
                    <li class="nav-item"> <a class="nav-link" href="#">Discover</a> </li>
                    <li class="nav-item"> <a class="nav-link" href="#">Post a Job</a> </li>
                    <li class="nav-item"> <a class="nav-link" href="#">Contact Us</a> </li>
                </ul>
                <a href="/sign-up" class="btn btn-outline-dark rounded-0">
                    Sign Up
                </a>
            </div>
        </div>
    </nav>

</div>
```

sesudah

```
<div class="container">

    <nav class="navbar navbar-expand-lg bg-transparent">
        <div class="container-fluid">
            <a class="navbar-brand" href="#">Jobless</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
                aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav m-auto mb-2 mb-lg-0">
                    <li class="nav-item"> <a class="nav-link active" aria-current="page" href="#">Home</a> </li>
                    <li class="nav-item"> <a class="nav-link" href="#">Discover</a> </li>
                    <li class="nav-item"> <a class="nav-link" href="#">Post a Job</a> </li>
                    <li class="nav-item"> <a class="nav-link" href="#">Contact Us</a> </li>
                </ul>
                <a href="/sign-up" class="btn btn-outline-dark rounded-0">
                    Sign Up
                </a>
            </div>
        </div>
    </nav>

</div>
```

13. Hasilnya akan seperti ini:

14. Sekarang kita akan implementasikan komponen hero section, tuliskan kode berikut.

```
<section id="hero"></section>
```

15. dapat kita lihat pada komponen hero terdapat text dan tombol posisinya ditengah, nah disini kita bisa menggunakan grid system bootstrap agar dapat mengatur posisinya.

```
<div class="row h-100 justify-content-center align-items-center text-center">
<div class="col-12"> <h1 class="fw-bold">Get Connected to the Best Remote <br
/>Jobs in Your Field </h1> <p class="text-muted">Discover a wide range of
remote opportunities on our platform and<br />take control of your career</p>
</div> </div>
```

Keterangan:

1. [row] digunakan untuk mengatur tata letak konten dalam satu baris. Class ini menambahkan beberapa stil ke tata letak konten, seperti mengatur margin dan padding untuk konten yang berada di sisi.
2. [h-100] adalah class Bootstrap yang digunakan untuk mengatur tinggi konten dalam sebuah halaman web. Class ini menambahkan stil yang memastikan konten berada pada tinggi penuh dari bagian atas ke bawah.d
3. [justify-content-center] adalah class Bootstrap yang digunakan untuk mengatur posisinya konten di tengah bagian horizontal.
4. class [align-items-center] adalah class Bootstrap yang digunakan untuk mengatur posisinya konten di tengah bagian vertikal.
5. class [text-center] adalah class Bootstrap yang digunakan untuk mengatur posisi teks di tengah bagian konten.
6. class [col-12] adalah class Bootstrap yang digunakan untuk mengatur lebar konten. Class ini

menambahkan stil yang memastikan konten berada pada lebar penuh dari bagian atas ke bawah.

16. Hasilnya akan seperti ini:



17. Disini text nya hanya ke tengah secara horizontal, agar dapat ke tengah secara vertikal perlu kita atur tinggi pada hero sectionnya dengan external style

```
<style> #hero { height: 85vh; } </style>
```

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

  <meta name="viewport" content="width=device-width, initial-s<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pI" />

  <title>Bootstrap Implementasi Web Desain</title>

  <style>
    #hero {
      height: 85vh;
    }
  </style>
</head>
```

Keterangan:

ini mengatur tinggi dari elemen yang memiliki ID `hero` menjadi 85% dari tinggi viewport (viewport height).

18. Hasilnya akan seperti ini:

Get Connected to the Best Remote Jobs in Your Field

Discover a wide range of remote opportunities on our platform and take control of your career

19. Sekarang kita akan implementasikan komponen button nya, dapat kita lihat pada desainnya terdapat dua button sejajar secara horizontal, maka kita perlu memakai flexbox yang disediakan oleh bootstrap.

```
<div class="d-flex gap-4 justify-content-center align-items-center"> <a href="#" class="">Explore Remote Jobs</a> <a href="#" class="">How it works? </a> </div>
```

Keterangan:

1. [d-flex] digunakan untuk mengatur tata letak konten dalam satu baris. Class ini menambahkan stil yang memastikan konten berada pada satu baris dan tidak berpindah ke baris baru.
2. [gap-4] digunakan untuk mengatur jarak antara konten dalam satu baris. Class ini menambahkan stil yang membuat jarak antara konten berada 4 unit.
3. [justify-content-center] digunakan untuk mengatur posisinya konten di tengah bagian horizontal.
4. [align-items-center] digunakan untuk mengatur posisinya konten di tengah bagian vertikal.

Get Connected to the Best Remote Jobs in Your Field

Discover a wide range of remote opportunities on our platform and take control of your career

[Explore Remote Jobs](#) [How it works?](#)

20. Selanjutnya kita perlu beri space dari setiap buttonnya, dan beri warna pada button sebelah kiri

```
<div class="d-flex gap-4 justify-content-center align-items-center"> <a href="#" class="p-2 px-3 text-decoration-none bg-primary rounded-5 text-white shadow-sm ">Explore Remote Jobs</a> <a href="#" class="p-2 px-3 text-primary rounded-5 fw-semibold">How it works?</a> </div>
```

Keterangan:

- [p-2] digunakan untuk mengatur padding pada tautan. Class ini menambahkan stil yang membuat padding pada tautan berada 0.5 unit pada sisi atas dan bawah, serta 0.75 unit pada sisi kiri dan kanan.
- [px-3] digunakan untuk mengatur padding pada tautan. Class ini menambahkan stil yang membuat padding pada tautan berada 0.75 unit pada sisi kiri dan kanan.
- [text-decoration-none] digunakan untuk mengatur dekorasi teks pada tautan. Class ini menghilangkan dekorasi garis bawah yang biasanya muncul pada tautan.
- [bg-primary] digunakan untuk mengatur warna latar belakang pada tautan. Class ini menambahkan stil yang membuat latar belakang tautan berwarna hitam.
- [rounded-5] digunakan untuk mengatur bentuk tautan. Class ini menambahkan stil yang membuat bentuk tautan berbentuk lingkaran dengan jari-jari 5 unit.
- [text-white] digunakan untuk mengatur warna teks pada tautan. Class ini menambahkan stil yang membuat warna teks tautan berwarna putih.
- [shadow-sm] digunakan untuk mengatur gaya bagian atas tautan. Class ini menambahkan stil yang membuat bagian atas tautan berwarna abu-abu dengan efek shadow.
- [fw-semibold] digunakan untuk mengatur font teks dengan berat font yang semibold.
- [text-primary] digunakan untuk mengatur warna teks pada tautan. Class ini menambahkan stil yang membuat warna teks tautan berwarna hitam.

Get Connected to the Best Remote Jobs in Your Field

Discover a wide range of remote opportunities on our platform and take control of your career

[Explore Remote Jobs](#)[How it works?](#)

21. Sekarang kita akan mengganti background body nya

```
<style> body { background-color: #F6FAFF; background-image: linear-gradient(90deg, #0500FF 0%, #F8FB76 35%, #4EFF75 64%, #00FFD1 100%); background-blend-mode: overlay; } </style>
```

```
<style>
  body {
    background-color: #f6fafaf;
    background-image: linear-gradient(120deg, #0500FF 0%, #F8FB76 35%, #4EFF75 64%, #00FFD1 100%);
    background-blend-mode: overlay;
  }

  #hero {
    height: 85vh;
  }
</style>
</head>
```

22. Hasilnya akan seperti ini:

Get Connected to the Best Remote Jobs in Your Field

Discover a wide range of remote opportunities on our platform and take control of your career

[Explore Remote Jobs](#)

[How it works?](#)

Best Practices dan Tips

Gunakan Grid Sistem:

Manfaatkan grid sistem Bootstrap untuk mengatur tata letak (layout) situs Anda dengan baik. Gunakan kelas-kelas seperti `container`, `row`, dan `col` untuk membagi konten Anda menjadi bagian-bagian yang sesuai.

Gunakan Komponen Bootstrap:

Manfaatkan komponen-komponen yang disediakan oleh Bootstrap seperti navbar, card, form, dll. Komponen-komponen ini sudah dirancang dan diuji untuk memastikan konsistensi dan fungsionalitas yang baik.

Uji pada Berbagai Browser:

Pastikan situs Anda berfungsi dengan baik di berbagai browser, termasuk versi lama dan baru, dengan melakukan pengujian silang browser.