

CSS (Cascading Style Sheet)

Pemrograman Web
Maryam S.Kom.,M.Eng.

- **CSS = Cascading Style Sheets**
- Styles mendefinisikan **bagaimana menampilkan** elemen HTML
- Styles telah diintegrasikan dengan HTML 5 **untuk mengatasi permasalahan style**

- HTML tidak pernah dimaksudkan berisi tag untuk memformat sebuah dokumen.
- HTML hanya dimaksudkan untuk mendefinisikan konten sebuah dokumen, seperti:
 - `<h1>This is a heading</h1>`
 - `<p>This is a paragraph.</p>`

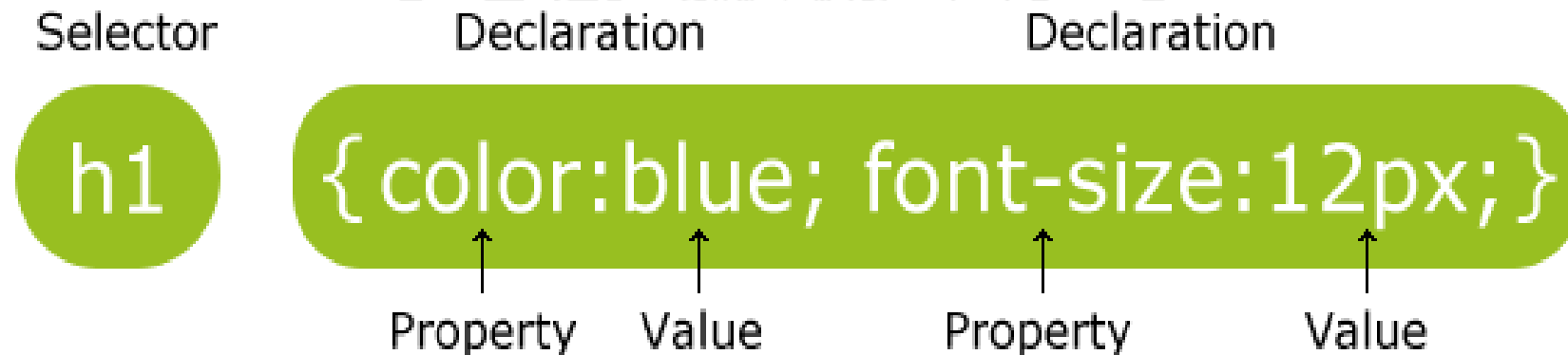
- Jika tag seperti , dan atribut color ditambahkan dalam HTML ini akan mempersulit bagi pengembang web.
- Pengembangan website yang besar, jika font dan color dimasukkan dalam setiap halaman, proses akan menjadi lama dan mahal.
- Untuk mengatasi masalah ini, World Wide Web Consortium (W3C) telah menciptakan CSS.
- Pada HTML 5.0, semua formatting dihilangkan dari dokumen HTML, dan disimpan dalam file CSS yang terpisah.
- Kini, semua browser mendukung CSS.

CSS Saves a Lot of Work!

- CSS mendefinisikan BAGAIMANA elemen HTML akan ditampilkan.
- Style secara umum disimpan dalam file .css eksternal.
- External style sheets memungkinkan untuk mengubah tampilan dan layout semua halaman dalam sebuah Web site, hanya dengan mengubah sebuah file!

- Satu set CSS terdiri atas sebuah selector dan sebuah blok deklarasi.

1. `selector { property:value }`
2. `selector { property:value; property:value }`



```
<!DOCTYPE html>
<html>
  <head>
    <title>Contoh CSS</title>
    <style>
      h1 {
        font-size: 40px;
        color: orange;
      }
    </style>
  </head>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```



Hello World

1. Inline Style

Kode css berada dalam tag HTML dengan menambah atribut style

```
<h1 style="font-size: 40px; color: ■ orange">Teks Judul</h1>
```

2. Internal Style

Kode CSS dikumpulkan dalam elemen style pada tag <head>

```
<style>
  h1 {
    font-size: 40px;
    color: ■ orange;
  }
</style>
```


3. Eksternal Style

Kode CSS berada pada file yang terpisah dengan kode HTML

File : style.css

```
h1 {
    font-size: 40px;
    color: orange;
}
```

- Property pada CSS digunakan untuk menentukan jenis style apa yang akan diterapkan pada tag, class, atau id yang kita pilih pada selector.
- Macam-macam property pada CSS antara lain:
 1. background
 2. border
 3. box model
 4. layouting
 5. font & text, dll.



- Element selector digunakan untuk memilih elemen berdasarkan pada nama elemen
- Dalam style sheets sering terjadi ada beberapa elemen dengan style yang sama, sehingga dapat dikelompokkan

```
h1 {  
    text-align: center;  
    color: red;  
}  
h2 {  
    text-align: center;  
    color: red;  
}  
p {  
    text-align: center;  
    color: red;  
}
```

```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```



Class dan id pada CSS

- Selain penggunaan elemen, juga dapat menargetkan class dan id pada halaman HTML. Pada id hanya boleh dipanggil satu kali saja, sedangkan class bisa dipanggil berkali-kali pada satu halaman HTML
- Penulisan id diawali dengan tanda # dan id="nama-id"
- Penulisan class diawali dengan tanda titik dan class="nama-class"

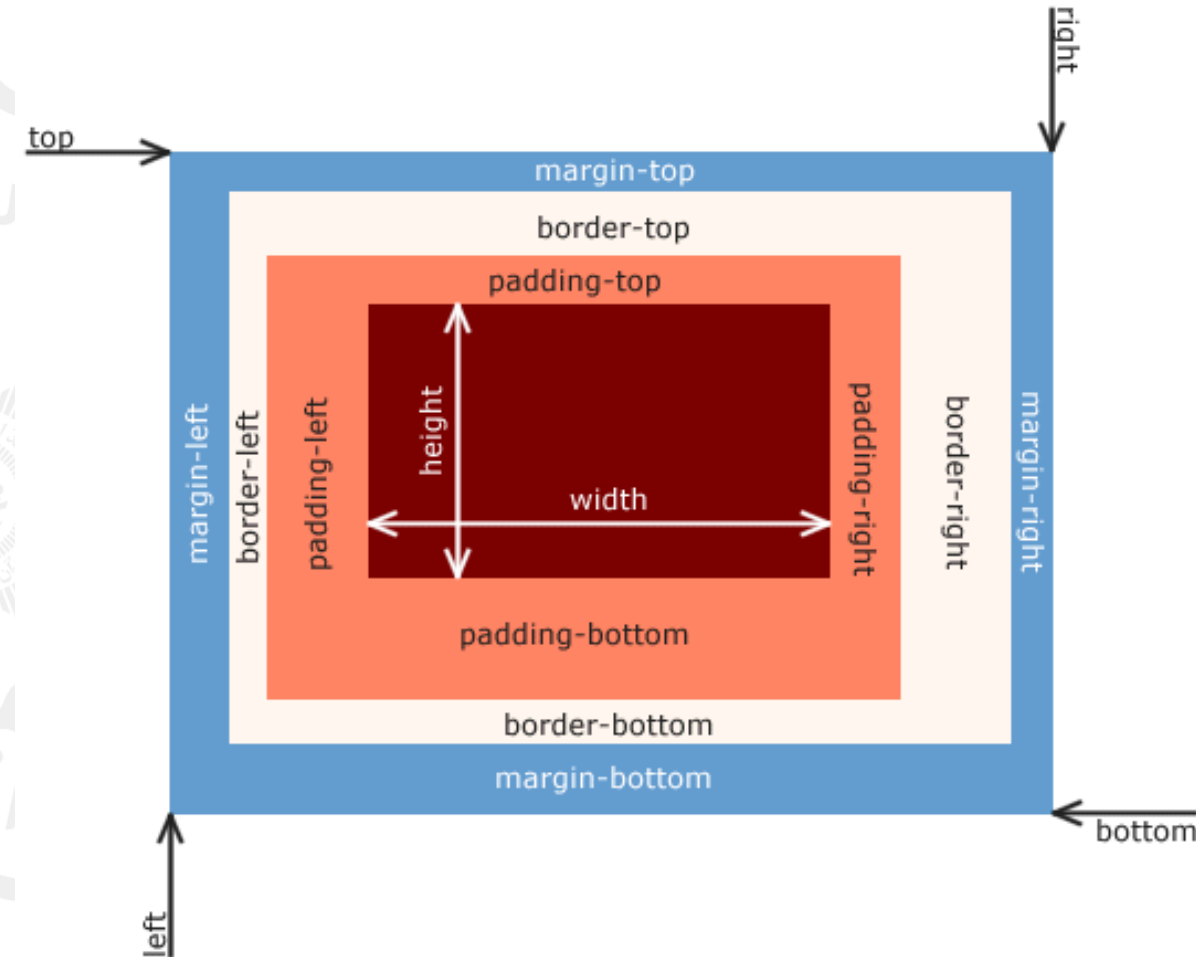
```
<html>
<head>
  <title>Contoh CSS</title>
  <style>
    h1 {
      font-size: 40px;
      color: orange;
    }
    #highlight {
      background-color: yellow;
    }
    .red {
      color: red;
    }
  </style>
</head>
<body>
  <h1>Penggunaan Selector Class dan Id</h1>
  <p class="red">Ini selector class</p>
  <p id="highlight">Ini selector id</p>
</body>
</html>
```

- Struktur dari border seperti berikut:

```
border : width style color;
```

- Jika kita ingin membuat border garis lurus dengan warna kuning maka kita dapat menulis seperti berikut:

```
border : 2px solid yellow;
```



- Margin : jarak spasi antar elemen. Memiliki properti yaitu margin, margin-top, margin-right, margin-bottom dan marginleft. Nilai dari margin boleh memiliki nilai negatif. Jika ingin menempatkan elemen berada di tengah maka gunakan nilai margin = auto.
- Padding : jarak antara border dengan isi html. Memiliki property antara lain padding, padding-top, padding-right, paddingbottom, dan padding-left. Nilai dari padding tidak boleh negatif.
- Border : batas garis yang mengelilingi padding dan isi dari html. Memiliki properti antara lain border, border-top, borderright, border-bottom, dan border-left

Float sendiri memiliki beberapa property, yaitu: **left, right, none**

- **Float : none**

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS Float</title>
  </head>
  <body>
    
    <p>TEKNIK INFORMATIKA, UNIVERSITAS MUHAMMADIYAH SURAKARTA</p>
  </body>
</html>
```



TEKNIK INFORMATIKA, UNIVERSITAS
MUHAMMADIYAH SURAKARTA

- Float : **left**

```
<head>
  <title>CSS Float</title>
  <style>
    |  img {
      |    float: left;
    |  }
  </style>
</head>
```



TEKNIK INFORMATIKA,
UNIVERSITAS
MUHAMMADIYAH
SURAKARTA

- Float : right

```
<head>
  <title>CSS Float</title>
  <style>
    img {
      float: right;
    }
  </style>
</head>
```

TEKNIK INFORMATIKA,
UNIVERSITAS
MUHAMMADIYAH
SURAKARTA



menentukan posisi suatu elemen terhadap elemen yang lain

- **Static** : posisi elemen tidak terpengaruh oleh properti top, bottom, left, dan right.
- **Relative** : posisi elemen terpengaruh oleh properti top, bottom, left, dan right.
- **Fixed** : posisi elemen tidak berubah walaupun kita melakukan scroll pada layar.
- **Absolute** : posisi elemen dapat kita atur secara bebas terhadap elemen lain yang diberi posisi relatif.

- Properties background yang dapat digunakan dalam CSS, yaitu:
- Background-color: digunakan untuk menentukan warna background dari suatu elemen.
- Nama warna – seperti “red” untuk merah atau white untuk putih
- HEX value – seperti #000000
- RGB value – seperti rgb(255,255,0)

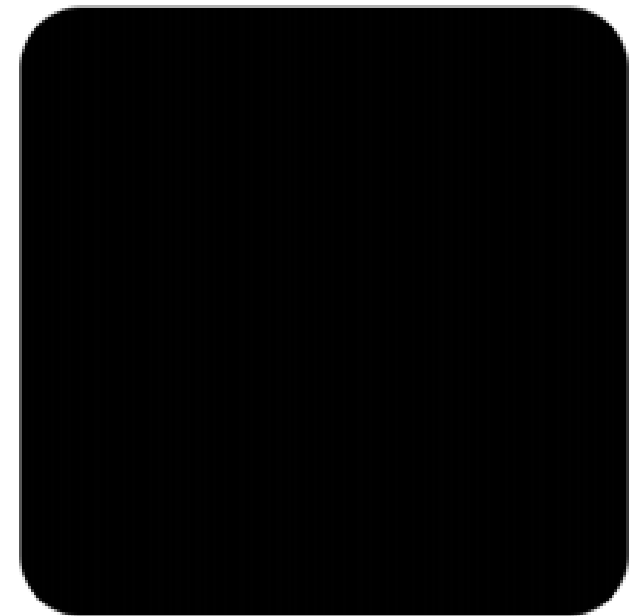
```
body{  
    background-color: #b0c53e;  
    background-image: url("ums.png");  
    background-size: 100px;  
    background-repeat: no-repeat;  
    background-position: center;  
}
```

- **Color** : memberi warna pada tulisan.
- **Text-align** : mengatur format text secara horizontal, yaitu rata kanan (right), rata kiri (left), rata tengah (center), dan rata kanan-kiri (justify).
- **Text-indent** : memberi tambahan jarak/indent pada paragraph (px dan %)
- **Text-decoration**: memberi dekorasi pada text, yaitu garis bawah (underline), garis atas (overline), atau garis tengah (line-through).
- **Text-transform**: mengubah jenis huruf., yaitu uppercase untuk mengubah text menjadi huruf besar semua, lowercase untuk mengubah text menjadi huruf kecil, capitalize untuk mengubah huruf pertama text menjadi huruf besar.
- **Letter-spacing dan word spacing**: untuk mengatur jarak antarkata sedang word spacing digunakan untuk mengatur jarak antarkata

- **Font-family:** mengatur jenis font yang akan digunakan. Fontfamily mengharuskan kita untuk menulis beberapa nama font sebagai sistem fallback, yaitu kondisi jika browser tidak mendukung font pertama, maka akan menggunakan font kedua.
- **Font-size:** mengatur besar dan kecilnya ukuran font (px, % dan em)
- **Font-weight:** mengatur ketebalan font(lighter, normal, bold, dan bolder)
- **Font-variant:** mengubah font menjadi bentuk small-caps.
- **Font-style:** mengubah bentuk font menjadi bercetak miring (normal, italic, dan obliqu)

- membuat sudut pada box menjadi melengkung. Property yang digunakan adalah border radius. Semakin besar nilai pada property ini maka sudut lingkaran akan menjadi semakin besar.

```
#box{  
    background-color: black;  
    width: 100px; height: 100px;  
    border-radius: 20px;  
}
```



- memberikan bayangan pada text maupun elemen lain. Untuk membuat bayangan pada text maka menggunakan text-shadow, sedangkan untuk box menggunakan box-shadow.
- Kedua property ini memiliki nilai dengan urutan:
 - H-offset : jarak bayangan secara horizontal.
 - V-offset : jarak bayangan secara vertikal.
 - Blur : tingkat bayangan semakin tinggi nilai semakin blur.
 - Spread : jarak penyebaran bayangan.
 - Color : warna bayangan



```
#box {  
  background-color: black;  
  width: 200px; height: 200px;  
  border-radius: 20px;  
  box-shadow: 10px 10px 5px grey;  
}
```

- membuat gradasi warna, sering digunakan dalam desain suatu tombol.
- Linear gradient : gradasi warna lurus

```
background : linear-gradient(direction, warna-  
pertama, warna-kedua, warna-dst);
```

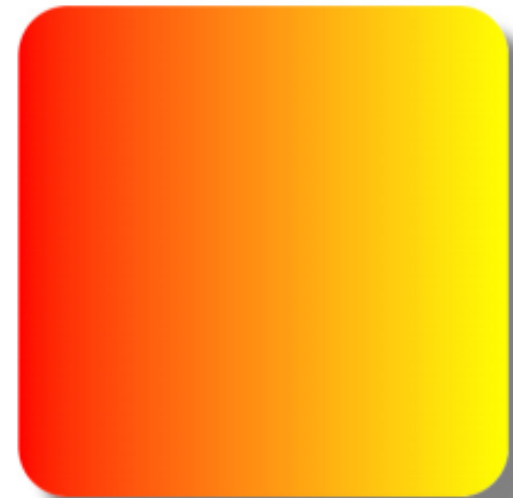
- Radial gradient : gradasi warna melingkar

```
background : radial-gradient(shape, warna-  
pertama, warna-kedua);
```

- Nilai *direction* diisi dengan arah *gradient* seperti *to left*, *to bottom right* atau menggunakan nilai sudut seperti 90deg, 240 deg dan seterusnya

```
#box {  
  background: linear-gradient(90deg, red, yellow);  
  width: 200px; height: 200px;  
  border-radius: 20px;  
  box-shadow: 10px 10px 5px grey;
```

```
}
```



Flexbox layout lebih mudah karena tanpa menggunakan *float* atau *position*. Parent element (display:flex) sebagai pembungkus , child element sebagai elemen yang akan diatur layout-nya.

PROPERTI :

- **Flex-direction:** bagaimana child element akan disusun(row, row-reverse, column, dan column-reverse)
- **Flex-wrap:** bagaimana child element akan disusun jika lebar total melebihi lebar parent element (nowrap, wrap, dan wrap reverse)
- **Flex-flow:** mengatur flex-direction dan flex-wrap sekaligus.
- **Justify-content:** mengatur align item secara horizontal (center, flex-start, flex-end, space-around, dan space-between)
- **Align-items:** mengatur align item secara vertical (center, flex-start, flex-end, stretch, dan baseline)
- **Align-content:** mengatur align dari konten (space-between, space-around, stretch, center, flex-start, dan flex-end)

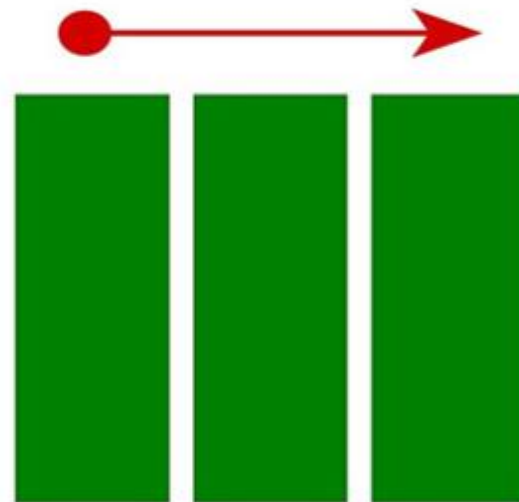
Sedangkan child element memiliki property sendiri, yaitu:

- **Order**: mengatur urutan susunan item
- **Flex-grow**: menambah ukuran item berdasar tempat kosong yang masih bisa diambil.
- **Flex-shrink**: mengurangi ukuran item jika tidak ada lagi tempat kosong yang tersisa.
- **Flex-basis**: mengatur lebar item dengan satuan px.
- **Align-self**: mengatur align dari suatu item menggantikan align yang ditetapkan pada parent element.
- **Flex**: shorthand untuk mengatur flex-grow, flex-shrink, dan flex-basis sekaligus.

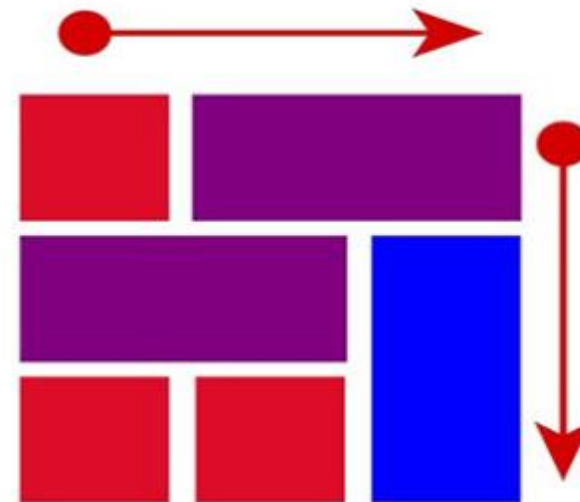
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Flex Layout</title>
    <style>
      .row {
        display: flex;
        background: grey;
        height: 200px;
        align-items: center;
      }
      .col {
        background: black;
        width: 100px;
        height: 100px;
        margin: 10px;
        flex-wrap: wrap;
      }
    </style>
  </head>
  <body>
    <div class="row">
      <div class="col" style="flex-grow: 2"></div>
      <div class="col" style="flex-grow: 4"></div>
      <div class="col" style="order: 4"></div>
      <div class="col" style="order: 5"></div>
      <div class="col" style="order: 3"></div>
    </div>
  </body>
</html>
```



- CSS Grid digunakan untuk membuat layout suatu halaman website
- Perbedaan utamanya adalah jika flexbox hanya bisa membuat layout untuk satu dimensi, CSS Grid dapat membuat layout untuk dua dimensi (sekaligus mengatur baris dan kolom)



Flexbox
One Dimensions



CSS Grids
Two Dimensions

CSS Grid terdiri dari parent element yang harus diset **display:grid** dan beberapa child elemen sebagai isinya.

PROPERTI parent element:

- **Grid-column-gap**: jarak antar kolom pada layout.
- **Grid-row-gap**: jarak antar baris pada layout.
- **Grid-gap**: mengatur jarak kolom dan baris sekaligus.
- **Grid-template-columns**: mengatur jumlah kolom pada grid layout.
- **Grid-template-rows**: mengatur tinggi dari masing-masing baris.
- **Grid-template-area**: sebagai referensi dari grid area pada child elemen.

PROPERTI child element:

- **Grid-column:** mengatur pada kolom seberapa akan diisi dengan konten.
- **Grid-row:** mengatur pada baris seberapa akan diisi dengan konten.
- **Grid-area:** digunakan untuk memberi nama ke grid item.

Buat halaman homepage yang memiliki struktur layout seperti berikut.
Gunakan CSS Grid untuk membuat halaman tersebut.

