

4. Stylesheet

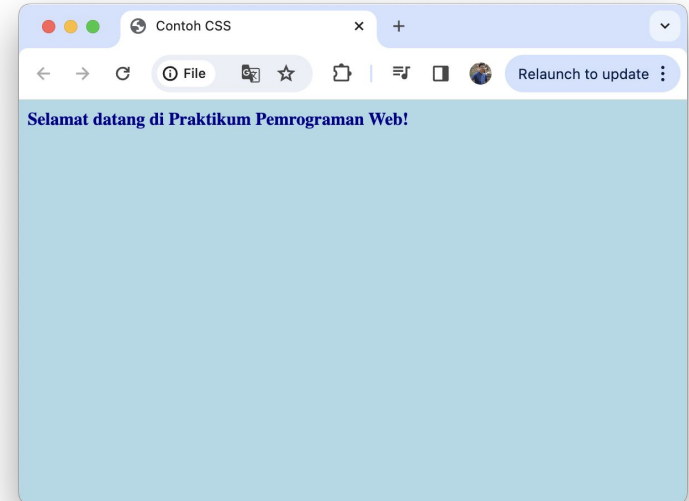
CSS

CSS = Cascading Style Sheets, bahasa yang digunakan untuk menghias website yang dibuat dengan HTML. Dengan CSS maka website kita akan menjadi lebih menarik dan tidak membosankan.

Contoh penggunaan:

```
<html>
<head>
  <title>Contoh CSS</title>
  <style>
    body {
      background-color: lightblue;
    }
    h4 {
      Text-align: center;
      color: darkblue;
    }
  </style>
</head>

<body>
  <h4>Selamat datang di Praktikum Pemrograman Web!</h4>
</body>
</html>
```



Implementasi CSS

Demo CSS https://www.w3schools.com/css/demo_default.htm

Penulisan CSS

CSS bisa ditulis dengan beberapa cara:

- Pada file terpisah dengan ekstensi `.css`
- Bersamaan di file HTML tersebut dengan tag `<style>`
- Sebagai atribut di setiap elemen HTML

Penulisan CSS dengan File Terpisah

- CSS bisa diletakkan pada file terpisah dari HTML.
- Beri nama file dengan ekstensi `.css`.

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="namafile.css">
</head>
<body>
  <h1>Hallo!</h1>
</body>
</html>
```

- Lokasi file CSS diletakkan pada atribut href di tag link

```
<link rel="stylesheet" href="namafile.css">
```

- Umumnya diletakkan di antara tag head pada HTML.

Penulisan CSS dengan File yang Sama

Menulis CSS juga boleh di file HTML yang sama. Kita dapat meletakkannya di antara tag `<style>`.

```
<html>
<head>
  <title>Contoh CSS</title>
  <style>
    body {
      background-color: lightblue;
    }
    h4 {
      Text-center;
      color: darkblue;
    }
  </style>
</head>

<body>
  <h4>Selamat datang di Praktikum Pemrograman Web!</h4>
</body>
</html>
```

Penulisan CSS pada Elemen HTML

CSS juga bisa dimuat pada elemen HTML yang ingin dihias dengan atribut style. Cara ini disebut juga sebagai inline-style

```
<html>
<head>
  <title>Contoh CSS</title>
</head>
<body style="background:lightblue;">
  <p style="color:blue;text-align:center;">Hallo good people</p>
</body>
</html>
```

Ukuran, Batas dan Jarak

Elemen-elemen HTML bisa diberikan batas dan jarak. Ada berbagai jenis batas dan jarak yang bisa digunakan pada CSS. Untuk membantu secara visual, kita akan melihat box model pada CSS.

Jarak dan batas pada CSS:

- `Border` : border adalah garis batas pada setiap elemen
- `Padding` : padding adalah jarak antara elemen dan garis border
- `Margin` : jarak antara border dengan elemen lain



Border

Untuk memperlihatkan garis batas dari sebuah elemen, dapat menggunakan properti border. Properti border yang dapat digunakan:

- border
- border-style
- border-ARAH (top/bottom/right/left)
- border-radius
- border-width
- border-color

Border

Contoh penggunaan border

```
<p style="border: 1px solid white;">Aku adalah tag p dengan border </p>
```

Properti border sendiri terdiri dari tiga bagian:

```
border: A B C;
```

- A adalah ukuran ketebalan border
- B adalah jenis dari border
- C adalah warna dari border

Jika `border: 1px solid white;` berarti:

- Ukuran garis sebesar 1px
- Jenis border garis lurus (solid)
- Warna border putih

Border

Jenis Garis pada Border:

- dashed
- dotted
- Solid
- double
- groove
- ridge
- inset
- outset

Padding

Padding adalah jarak antara garis batas (border) dengan elemennya

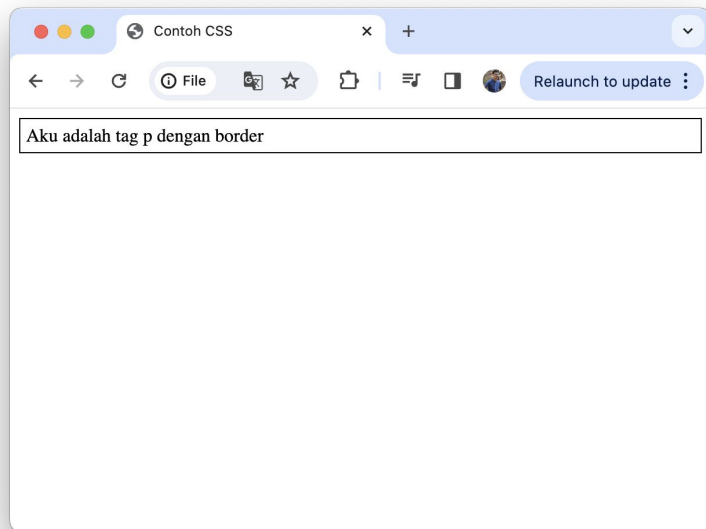
Contoh penggunaan

```
<html>

<head>
  <title>Contoh CSS</title>
  <style>
    .padding {
      border: 1px solid black;
      padding: 5px;
    }
  </style>
</head>

<body>
  <p class="padding">Aku adalah tag p dengan border </p>
</body>

</html>
```



Margin

Margin adalah jarak antara garis batas (border) dengan elemen luar atau elemen lainnya.

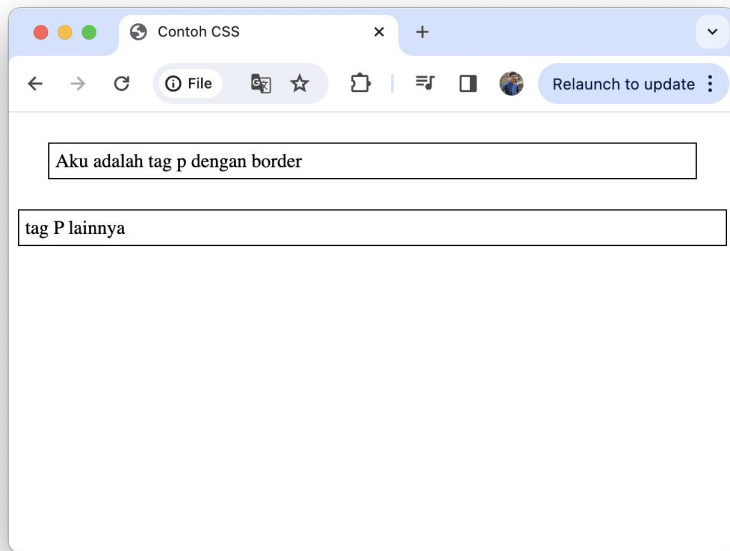
```
<html>

<head>
  <title>Contoh CSS</title>
  <style>
    .p {
      border: 1px solid black;
      padding: 5px;
    }

    .margin {
      margin: 25px;
    }
  </style>
</head>

<body>
  <p class="p margin">Aku adalah tag p dengan border </p>
  <p class="p"> tag P lainnya </p>
</body>

</html>
```



Text CSS

Properti dari CSS yang dapat digunakan untuk menghias HTML, seperti:

- `color`: untuk mengubah warna
- `background-color`: mengubah warna latar text
- `text-align`: mengatur perataan text secara horizontal
- `vertical-align`: perataan text secara vertikal
- `direction`: mengatur arah text kiri-kanan atau sebaliknya
- `text-decoration-line`: memberi garis pada text
- `text-transform`: mengubah huruf besar/kecil untuk sebuah text
- `text-shadow`: memberi bayangan pada sebuah text

Text CSS

Jarak antar properti juga dapat diatur dengan:

- `text-indent`: jarak indentasi
- `letter-spacing`: jarak antar huruf
- `word-spacing`: jarak antar huruf
- `line-height`: jarak antar baris
- `white-space`: bagaimana white-space diatasi di text tersebut

Text CSS

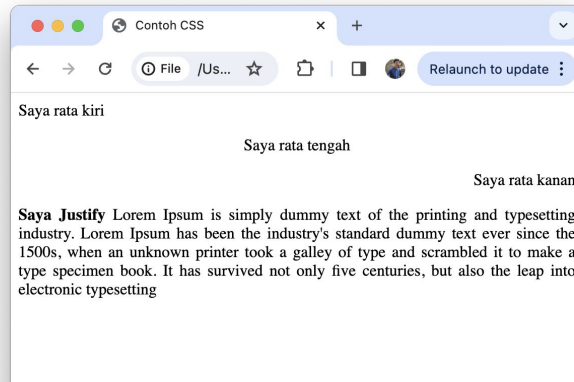
Text align digunakan untuk mengatur posisi text, penggunaannya seperti contoh dibawah ini:

```
<p style="text-align:left;"> Saya rata kiri </p>
```

```
<p style="text-align:center;"> Saya rata tengah </p>
```

```
<p style="text-align:right;"> Saya rata kanan </p>
```

```
<p style="text-align:justify;"><strong>Saya Justify </strong>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting</p>
```



Font CSS

Ada banyak jenis font yang bisa digunakan untuk menampilkan text di website. Jenis font yang ada website dapat memiliki lebih dari satu jenis font. Untuk menyajikan font yang mudah dibaca dan sesuai dengan karakter website merupakan salah satu hal penting.

Terdapat 5 font umum yang bisa digunakan pada CSS atau dikenal sebagai generic font, yaitu:

- Serif
- Sans-serif
- Monospace
- Cursive
- Fantasy

Font CSS

Contoh penggunaan Font dalam CSS

```
.p1 {  
    font-family: "Times New Roman", Times, serif;  
}  
.p2 {  
    font-family: Arial, Helvetica, sans-serif;  
}  
.p3 {  
    font-family: "Lucida Console", "Courier New", monospace;  
}
```

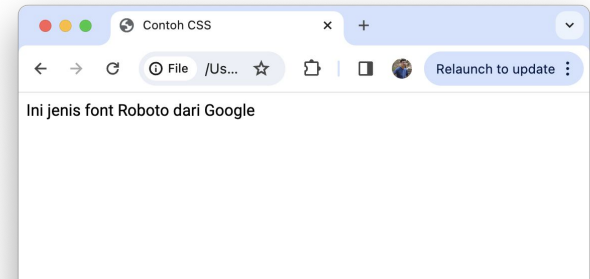
Font CSS

Ada beberapa website yang menawarkan variasi font yang beragam, salah satunya yang sangat populer digunakan adalah [Google Fonts](https://fonts.googleapis.com/css2?family=Roboto&display=swap). Kita bisa mencari berbagai jenis font yang cocok dengan website kita. Untuk menggunakannya perlu memasang kode di bagian CSS atau HTML.

Contoh penggunaannya:

```
<style>
@import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');
.p1 {
  font-family: 'Roboto', sans-serif;
  font-size: 30px;
}
</style>
```

```
<p class="p1"> Ini jenis font Roboto dari Google </p>
```



Tata Letak

Dalam CSS, display memiliki beberapa nilai, seperti block dan inline yang mengubah bagaimana elemen ditampilkan di halaman web. Nilai ini mengubah bagaimana elemen berinteraksi dengan elemen lain di sekitarnya.

Display Block

Elemen block selalu memulai baris baru di halaman web. Elemen ini akan mengambil seluruh lebar yang tersedia, dengan margin atas dan bawah

```
p {  
  display: block;  
}
```

elemen `<p>` akan ditampilkan sebagai elemen block. Artinya paragraf akan memulai pada baris baru dan membentang sejauh mungkin ke kanan dan ke kiri.

Tata Letak

Display Inline

Berbeda dengan elemen block, elemen inline tidak memulai baris baru dan hanya mengambil lebar sebanyak yang diperlukannya saja.

```
span {  
    display: inline;  
}
```

Elemen `` di atas akan ditampilkan sebagai elemen inline. Artinya elemen tersebut tidak akan memulai baris baru dan hanya akan mengambil ruang yang diperlukan.

Tata Letak

Perbedaan utama antara block dan inline adalah cara berinteraksi dengan elemen lain di sekitarnya. Elemen block akan memulai baris baru dan mengambil seluruh lebar yang tersedia, sedangkan elemen inline akan berdampingan dengan elemen lain dan hanya mengambil ruang yang diperlukan.

Tata Letak

Inline-block

untuk menampilkan elemen di baris yang sama, tetapi dengan dapat mengontrol layout seperti tinggi, lebar, dan margin

```
element {  
  display: inline-block;  
}
```

Kelebihan menggunakan inline-block:

- Dapat menetapkan tinggi dan lebar seperti elemen block.
- Elemen ini tetap dalam aliran inline normal, seperti elemen inline.
- Proses penataan antara elemen menjadi lebih mudah dan lebih fleksibel.

Tata Letak

Display None

digunakan untuk menghilangkan elemen dari tampilan tanpa mengubah tata letak halaman.

```
p.hidden {  
  display: none;  
}
```

elemen paragraf `<p>` dengan kelas `.hidden` akan dihilangkan dari tampilan, seolah-olah tidak pernah ada pada halaman tersebut.

Tata Letak

Perbedaan antara 'display: none' dan 'visibility: hidden'

Saat menggunakan `display: none`, elemen tersebut secara efektif dihapus dari flow halaman. Artinya, ruang yang sebelumnya ditempati oleh elemen tersebut akan diambil alih oleh elemen lain jika ada.

Sebaliknya, `visibility: hidden` akan membuat elemen tidak tampak, namun tetap mempertahankan ruang yang ditempati elemen tersebut di halaman. Elemen lain tidak akan mengambil alih ruang tersebut.

Tata Letak

Position Static

Secara default, semua elemen HTML memiliki nilai `position: static;`. Artinya, elemen akan ditempatkan dalam urutan alami dalam halaman web, sebagaimana mereka muncul dalam kode HTML. Dalam kasus ini, top, right, bottom, dan left tidak berfungsi.

```
div {  
    position: static;  
}
```

elemen `div` akan muncul secara normal dalam alur dokumen. Properti lain seperti top, bottom, right, dan left tidak akan memberikan efek apapun pada elemen ini.

Tata Letak

Position Static vs Position Lainnya

`position: relative;` akan membuat elemen berlaku relatif terhadap posisi normalnya.

`position: fixed;` akan membuat elemen ditempatkan relatif terhadap jendela browser, dan tidak akan bergerak meskipun halaman di-scroll.

`position: absolute;` akan membuat elemen berada dalam posisi absolut terhadap elemen terdekat yang bukan `position: static`.

`position: sticky;` adalah kombinasi antara `position: relative;` dan `position: fixed;`, dan akan berubah berdasarkan posisi scroll pengguna.

Tata Letak

Float

properti `float` digunakan untuk mendorong elemen sejauh mungkin ke kiri atau ke kanan, agar elemen lain dapat membungkus di sekitarnya. Ini berguna saat kita ingin konten seperti gambar untuk berdampingan dengan teks.

```
img {  
  float: right;  
}
```

Gambar akan dipindahkan sejauh mungkin ke kanan halaman, dengan teks akan ditampilkan di sebelah kirinya.

Pseudo-Class link

Pseudo-Class link merupakan salah satu fitur dalam CSS yang memungkinkan untuk memodifikasi tampilan link. Dengan menggunakan Pseudo-Class link, kita bisa merubah warna, ukuran teks, dan berbagai properti lainnya pada sebuah link.

Untuk menggunakan Pseudo-Class link, kita perlu menambahkan `:link` setelah selector. Berikut ini adalah contoh penggunaannya:

```
a:link {  
  color: red;  
}
```

semua link tidak aktif (belum diklik) akan berwarna merah.

Pseudo-Class link

`:hover` akan memodifikasi tampilan link saat mouse diarahkan ke link tersebut.

```
a:hover {  
  color: blue;  
}
```

`:active` akan memodifikasi tampilan link saat link tersebut diklik.

```
a:active {  
  color: green;  
}
```

`:visited` akan memodifikasi tampilan link yang telah diklik.

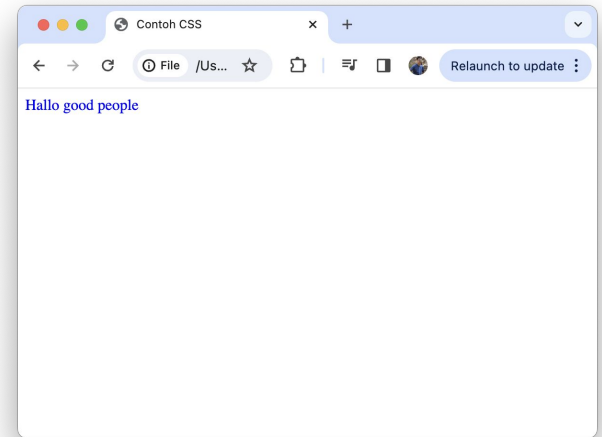
```
a:visited {  
  color: purple;  
}
```

Langkah Praktikum

1. Mengganti warna pada tulisan di CSS

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

- `color` adalah kata kunci (property) yang disediakan CSS untuk mengganti warna
- `blue` adalah nilai warna yang kita berikan

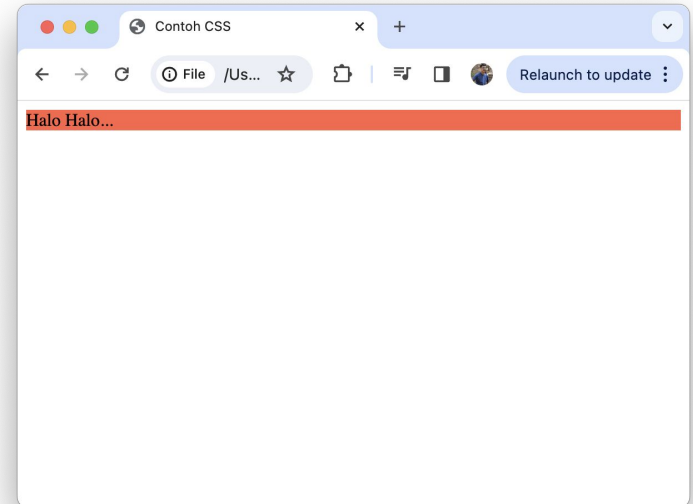


Langkah Praktikum

2. Mengganti warna background di CSS

Properti yang digunakan untuk mengganti warna latar adalah background-color

```
<p style="background-color:Tomato;">Halo Halo...</p>
```



Langkah Praktikum

3. Mengganti background dengan gambar di CSS

```
body {  
  background-image: url('link_image.jpg');  
}
```

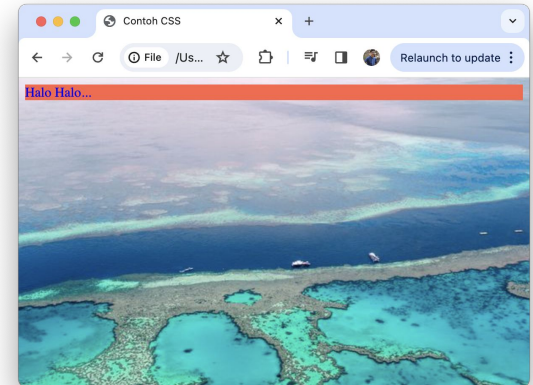
seluruh latar belakang halaman web akan ditutupi oleh gambar yang ditunjuk oleh URL di dalam url()

Mengulang gambar background

```
background-repeat: no-repeat;
```

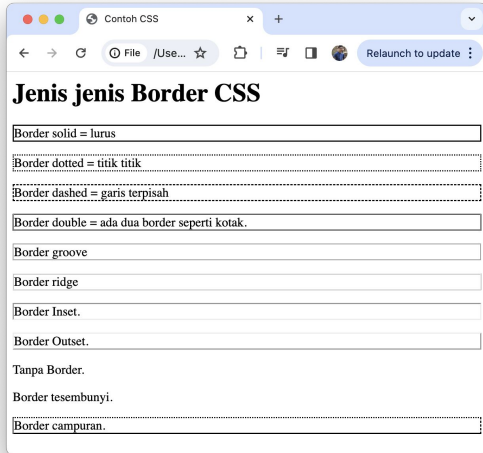
Mengatur posisi

```
background-position: right top;
```



Langkah Praktikum

4. Mengimplementasikan jenis Border



```
<html>

<head>
  <title>Contoh CSS</title>
  <style>
    p.solid {border-style: solid;}
    p.dotted {border-style: dotted;}
    p.dashed {border-style: dashed;}
    p.double {border-style: double;}
    p.groove {border-style: groove;}
    p.ridge {border-style: ridge;}
    p.inset {border-style: inset;}
    p.outset {border-style: outset;}
    p.none {border-style: none;}
    p.hidden {border-style: hidden;}
    p.mix {border-style: dotted dashed solid double;}
  </style>
</head>

<body>
  <h1>Jenis jenis Border CSS </h1>

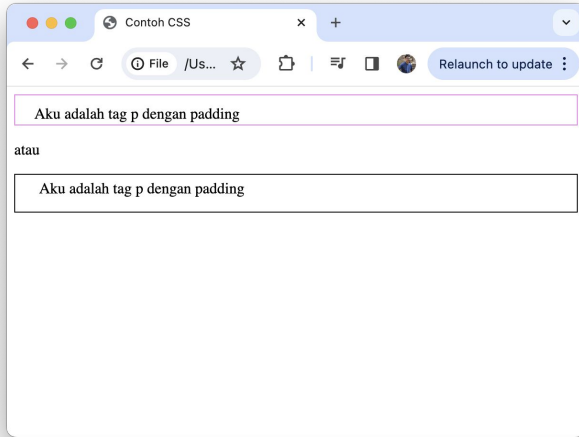
  <p class="solid">Border solid = lurus</p>
  <p class="dotted">Border dotted = titik titik</p>
  <p class="dashed">Border dashed = garis terpisah</p>
  <p class="double">Border double = ada dua border seperti kotak.</p>
  <p class="groove">Border groove </p>
  <p class="ridge">Border ridge</p>
  <p class="inset">Border Inset.</p>
  <p class="outset">Border Outset.</p>
  <p class="none">Tanpa Border.</p>
  <p class="hidden">Border tesembunyi.</p>
  <p class="mix">Border campuran.</p>

</body>

</html>
```

Langkah Praktikum

5. Membuat padding pada tiap sisi



```
<html>

<head>
  <title>Contoh CSS</title>
  <style>
    .padding-all {
      border: 1px solid violet;
      padding: 10px 5px 2px 20px;
    }

    .padding-each {
      border: 1px solid black;
      padding-top: 5px;
      padding-right: 10px;
      padding-bottom: 15px;
      padding-left: 25px;
    }
  </style>
</head>

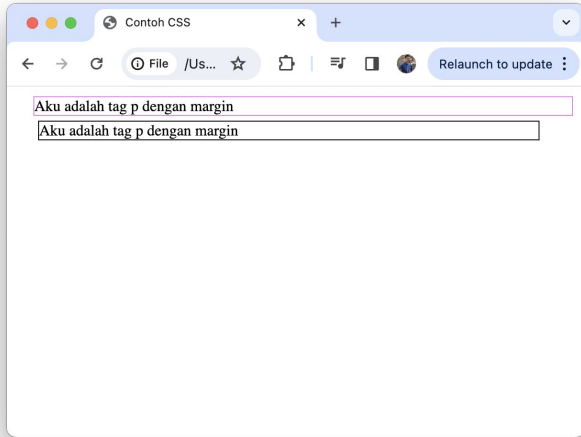
<body>
  <p class="padding-all">Aku adalah tag p dengan padding </p>
  atau
  <p class="padding-each">Aku adalah tag p dengan padding </p>

</body>

</html>
```

Langkah Praktikum

6. Memberi margin pada tiap sisi



```
<html>

<head>
  <title>Contoh CSS</title>
  <style>
    .margin-all {
      border: 1px solid violet;
      margin: 10px 5px 2px 20px;
    }

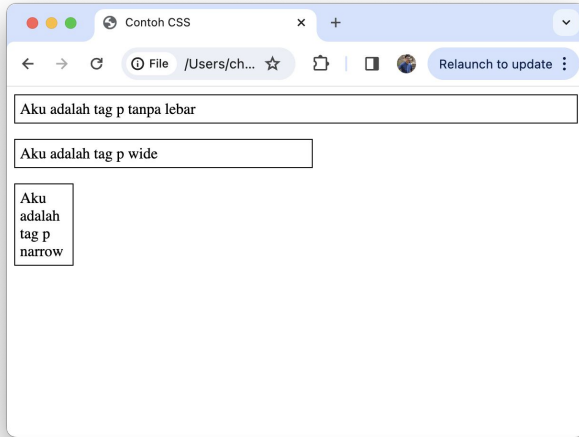
    .margin-each {
      border: 1px solid black;
      margin-top: 5px;
      margin-right: 40px;
      margin-bottom: 15px;
      margin-left: 25px;
    }
  </style>
</head>

<body>
  <p class="margin-all">Aku adalah tag p dengan margin </p>
  <p class="margin-each">Aku adalah tag p dengan margin </p>
</body>

</html>
```

Langkah Praktikum

7. Mengatur Lebar Elemen



tag p pertama otomatis punya lebar 100% karena ia bersifat block
tag p kedua punya lebar 300px
tag p ketiga lebarnya lebih kecil dari isi konten, maka otomatis konten akan terbungkus

```
<html>

<head>
  <title>Contoh CSS</title>
  <style>
    .p {
      border: 1px solid black;
      padding: 5px;
    }

    .wide {
      width: 300px;
    }

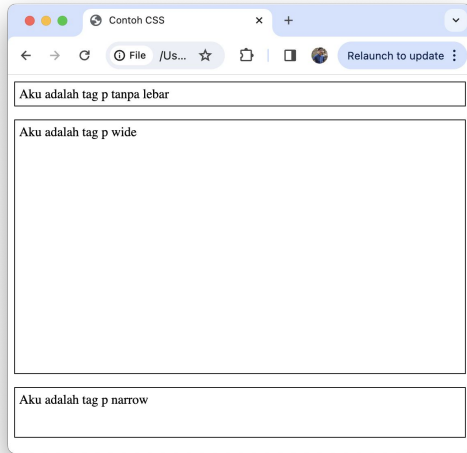
    .narrow {
      width: 50px;
    }
  </style>
</head>

<body>
  <p class="p">Aku adalah tag p tanpa lebar </p>
  <p class="p wide">Aku adalah tag p wide </p>
  <p class="p narrow"> Aku adalah tag p narrow </p>
</body>

</html>
```

Langkah Praktikum

8. Mengatur Tinggi Elemen



tag p pertama otomatis punya tinggi sesuai kontennya
tag p kedua punya tinggi, sehingga ada jarak terlihat
tag p ketiga tingginya lebih kecil dari isi konten, maka otomatis konten akan terbungkus

```
<html>

<head>
  <title>Contoh CSS</title>
  <style>
    .p {
      border: 1px solid black;
      padding: 5px;
    }

    .high {
      height: 300px;
    }

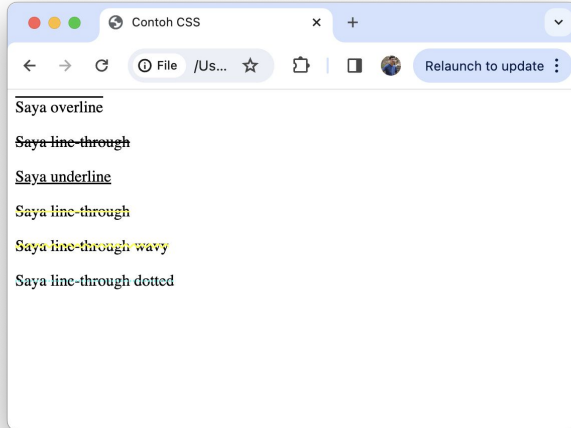
    .low {
      height: 50px;
    }
  </style>
</head>

<body>
  <p class="p">Aku adalah tag p tanpa lebar </p>
  <p class="p high">Aku adalah tag p wide </p>
  <p class="p low"> Aku adalah tag p narrow </p>
</body>

</html>
```

Langkah Praktikum

9. Mendekorasi Text



```
<html>

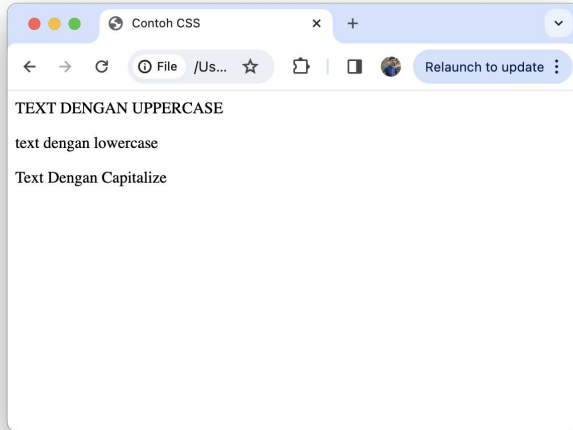
<head>
  <title>Contoh CSS</title>
</head>

<body>
  <p style="text-decoration:overline;"> Saya overline </p>
  <p style="text-decoration:line-through;"> Saya line-through </p>
  <p style="text-decoration:underline;"> Saya underline </p>
  <p style=" text-decoration:line-through; text-decoration-color:
yellow; "> Saya line-through </p>
  <p style=" text-decoration:line-through; text-decoration-color:
yellow; text-decoration-style: wavy; "> Saya
    line-through wavy </p>
  <p style=" text-decoration:line-through; text-decoration-color: aqua;
text-decoration-style: dotted; "> Saya
    line-through dotted </p>
</body>

</html>
```

Langkah Praktikum

10. Mentransformasikan Text



```
<html>

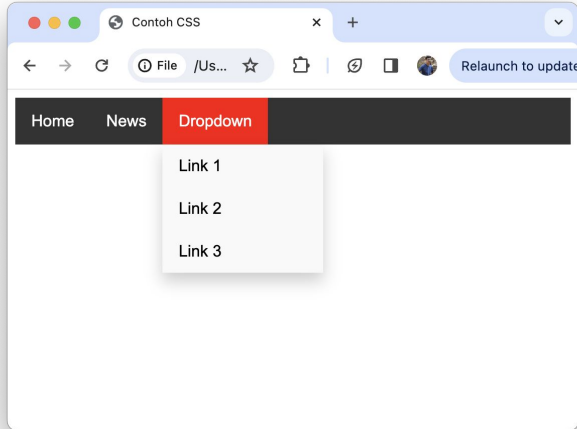
<head>
  <title>Contoh CSS</title>
</head>

<body>
  <p style="text-transform: uppercase;"> Text dengan uppercase </p>
  <p style="text-transform: lowercase;"> Text dengan lowercase </p>
  <p style="text-transform: capitalize;"> Text dengan Capitalize </p>
</body>

</html>
```


Langkah Praktikum

11. Membuat List Menu



```
<html>
<head>
  <title>Contoh CSS</title>
  <style>
    body { font-family: Arial, Helvetica, sans-serif; }
    .navbar { overflow: hidden; background-color: #333; }
    .navbar a { float: left; font-size: 16px; color: white; text-align: center; padding: 14px
16px; text-decoration: none; }
    .dropdown { float: left; overflow: hidden; }
    dropdown .dropbtn { font-size: 16px; border: none; outline: none; color: white; padding: 14px
16px; border-color: inherit; font-family: inherit; margin: 0; }
    .navbar a:hover, .dropdown:hover .dropbtn { background-color: red; }
    .dropdown-content { display: none; position: absolute; background-color: #f9f9f9; min-width:
160px; box-shadow: 0px 8px 16px 0px rgba(0, 0, 0, 0.2); z-index: 1; }
    .dropdown-content a { float: none; color: black; padding: 12px 16px; text-decoration: none;
display: block; text-align: left; }
    .dropdown-content a:hover { background-color: #ddd; }
    .dropdown:hover .dropdown-content { display: block; }
  </style>
</head>

<body>
  <div class="navbar">
    <a href="#home">Home</a>
    <a href="#news">News</a>
    <div class="dropdown">
      <button class="dropbtn">Dropdown</button>
      <div class="dropdown-content">
        <a href="#">Link 1</a>
        <a href="#">Link 2</a>
        <a href="#">Link 3</a>
      </div>
    </div>
  </div>
</body>
</html>
```

Tugas Praktikum

- Silahkan pelajari tentang kegunaan dari fungsi CSS: `text-indent`, `letter-spacing`, `word-spacing`, `line-height`, `white-space`
- Jelaskan perbedaan masing-masing disertai dengan capture code dan hasil
- Buatlah sebuah halaman website dengan Semantic HTML dan menerapkan styling CSS
- Hasil dikumpulkan dalam bentuk file berformat .pdf