

Cascading Style Sheets

[3.0 Pendahuluan](#)

[3.1 Pemilih & Deklarasi](#)

[3.2 Properti Font](#)

[3.3 Properti Warna](#)

[3.4 Daftar](#)

[3.5 Perataan Teks](#)

[3.6 Gambar Latar](#)

[3.7 Model Kotak](#)

[3.7.1 Border](#)

[3.7.2 Margin dan Padding](#)

3.0 Pendahuluan

MENGAPA

CSS adalah singkatan dari Cascading Style Sheets. Meski HTML digunakan untuk menambahkan makna dan struktur ke konten halaman web, CSS mendeskripsikan *cara* konten tersebut ditampilkan. "Cara" ini disebut sebagai *gaya* halaman.

Meski pembuatan gaya dan tata letak dapat dilakukan menggunakan HTML (tabel untuk tata letak, elemen font untuk gaya, dll.), penggunaan Cascading Style Sheets merupakan opsi yang jauh lebih baik, karena fitur ini memisahkan kode untuk menambahkan makna (HTML) dari kode untuk presentasi (CSS).

BAGAIMANA

Terdapat 3 cara untuk menggunakan CSS:

- **Inline**
 - Style sheet inline berlaku untuk konten elemen tunggal. Style sheet inline muncul di pembukaan dan penutupan tag dan hanya berlaku untuk konten tag khusus tersebut.
 - Spesifikasi gaya inline muncul sebagai nilai atribut *gaya* tag seperti berikut:
`<p style="color: red;">Selamat datang di CSS!</p>`

Contoh

```
<!DOCTYPE html>
<html>

<head>
<title>Halaman saya</title>
</head>

<body>
<p style="color:red">teks</p>
</body>

</html>
```

- Biasanya cara ini tidak bagus untuk digunakan dalam menerapkan gaya, karena CSS dan HTML bercampur sehingga sulit dikelola.

- **Tingkat dokumen**

- Style sheet tingkat dokumen berlaku untuk seluruh dokumen.
- Tambahkan elemen gaya di <head> dokumen seperti berikut:

```
<style>
p {
color: red;
}
</style>
```

Contoh

```
<!DOCTYPE html>
<html>

<head>
<title>Halaman saya</title>

<style type="text/css">
p {
color: red;
}
</style>
</head>
```

```
<body>
<p>teks</p>
</body>

</html>
```

- Cara ini memisahkan CSS dari kode HTML dan mempertahankan semua aturan CSS di satu tempat, dan itu bagus! Namun, dengan metode ini, kode CSS tetap disertakan dalam file HTML dan tidak dapat dibagikan antardokumen.

- **Eksternal**

- Style sheet eksternal dapat diterapkan ke dokumen, berapa pun banyaknya.
- Tag <link> digunakan untuk menentukan style sheet eksternal. Tambahkan tag link ke <head>:

```
<head>
...
<link rel="stylesheet" href="/css/global.css">
...
</head>
```

- Anda dapat menautkan ke lebih dari satu style sheet:

```
<head>
...
<link rel="stylesheet" href="/css/global.css">
<link rel="stylesheet" href="css/local.css">
...
</head>
```

Contoh

```
<!DOCTYPE html>
<html>

<head>
<title>Halaman saya</title>
<link rel="stylesheet" href="my_styles.css">
</head>

<body>
```

```
<p>teks</p>
</body>

</html>
```

- Menggunakan CSS dari file eksternal membuat kode CSS tetap terpisah dari HTML. Tindakan tersebut membuat CSS lebih mudah dipahami dan dikelola serta memungkinkan Anda berbagi gaya antardokumen — dan memungkinkan penyimpanan cache dengan browser, yang hanya perlu memuat setiap file CSS satu kali.

TIPS: Setiap browser memiliki style sheet default: misalnya, untuk menentukan ukuran judul, dan margin default sebelum dan setelah elemen paragraf. Anda dapat menemukan daftar lengkap style sheet browser [di sini](#).

GLOSARIUM

Cascading Style Sheets sangat cocok jika digunakan untuk menentukan urutan prioritas ketika terjadi bentrok elemen gaya.

3.1 Pemilih & Deklarasi

BAGAIMANA

Setiap aturan CSS memiliki 3 bagian:

1. Pemilih
2. Properti
3. Nilai

Pemilih menunjukkan elemen yang dipengaruhi oleh aturan.

Properti ('color') dan nilai ('red') membentuk deklarasi.

Misalnya, Anda dapat membuat teks untuk semua elemen `p` berwarna merah dengan aturan CSS berikut:

```
p { color: red; }
```

Di sini, `p` merupakan pemilih.

`color` adalah properti. Properti selalu diikuti dengan titik dua (:)

`red` adalah nilai. Nilai selalu diikuti dengan titik koma (;)

Anda dapat menambahkan beberapa deklarasi ke setiap aturan:

```
p {color: red; font-size: 12px;}
```

Tempatkan setiap deklarasi di baris terpisah agar lebih mudah dibaca:

```
p {  
  color: red;  
  font-size: 12px;  
}
```

TIPS: Sebaiknya tambahkan titik koma ke setiap deklarasi, meski hanya ada satu deklarasi, untuk menghindari masalah jika kemudian Anda menambahkan deklarasi lagi.

BAGAIMANA

Bentuk pemilih bisa bervariasi:

COBALAH

1. Nama elemen

Nilai properti berlaku untuk semua kekekrapan elemen yang diberi nama.

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

Anda dapat menerapkan nilai yang sama ke beberapa elemen:

```
h2, h3 {color: green;}
```

2. Pemilih Kelas

Pilih elemen dengan atribut kelas seperti berikut:

HTML:

```
<p class="highlight">Yah!</p>
```

CSS:

```
p.highlight {  
  color: red;  
  background-color: yellow;  
}
```

Biarkan nama elemen untuk menerapkan nilai ke beberapa elemen dengan kelas yang sama:

HTML:

```
<p class="highlight">Yah!</p>  
...
```

```
<blockquote class="highlight">Sisipkan kutipan apa pun.</blockquote>
```

CSS:

```
.highlight {  
  color: red;  
  background-color: yellow;  
}
```

3. Pemilih ID

Terapkan gaya ke elemen tunggal.

```
#intro {  
  font-size: 18pt;  
}
```

```
<p id="intro">Ukuran di sini adalah 18. </p>
```

4. Pemilih Universal

Pemilih universal berlaku untuk semua elemen di dokumen dan ditandai dengan bintang (*).

```
* {  
  color: yellow;  
}
```

PERINGATAN: pemilih universal tidak disarankan. Pemilih tersebut dapat mengurangi performa dan membuat CSS sulit melakukan debug karena aturan diterapkan untuk setiap elemen.

GLOSARIUM

Pemilih digunakan untuk menemukan elemen HTML yang ingin diberi gaya. Hal tersebut dapat berupa nama elemen, ID, atau nama kelas: `h1`, `#intro`, `.warning`, dll.

Deklarasi memiliki dua bagian, properti dan nilai, misalnya: `color: red;`.

TIPS: Jangan lupa menggunakan komentar untuk meningkatkan keterbacaan kode CSS Anda! Komentar ditulis dalam format berikut:

```
/* Ini adalah komentar, format tidak menjadi masalah! */
```

3.2 Properti Font

MENGAPA

Dalam dokumen web apa pun, Anda akan melihat berbagai macam ukuran font, jenis font (Arial, Helvetica dll), gaya font (**Cetak tebal**, *Miring*, Garis bawah), dan properti font lain.

COBALAH

1. Tentukan urutan preferensi font

```
font-family: Arial, Helvetica, Futura, sans-serif
```

2. Tentukan ukuran teks

```
font-size: 14pt
```

3. Buat teks jadi cetak miring:

```
font-style: italic
```

4. Tentukan teks cetak tebal:

```
font-weight: bold
```

Properti disingkat

Anda dapat menentukan lebih dari satu nilai dengan menggunakan properti `font`:

```
font: bold 12pt Arial
```

GLOSARIUM

Font-family adalah jenis huruf (Helvetica, Arial, dll).

Font-size adalah ukuran teks.

Font-style adalah gaya teks (*miring*, **cetak tebal**, atau normal).

3.3 Properti Warna

MENGAPA

Tentukan warna menggunakan salah satu dari 16 nama warna: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, dan yellow. Sebagai ganti menggunakan nama warna, Anda dapat menggunakan nilai numerik untuk menentukan kadar warna merah, hijau, dan biru yang akan digunakan. Tiga aturan berikut memiliki pengaruh yang sama. Aturan pertama menggunakan angka desimal dan yang kedua menggunakan angka heksadesimal: FF artinya 255 dalam heksadesimal (basis 16).

```
h1 {color: rgb(255, 0, 0);}
h1 {color: #FF0000;}
h1 {color: red;}
```

Anda juga dapat menyetel warna latar belakang elemen. Dalam contoh berikut, teks akan berwarna merah dan warna latar belakang elemen H1 akan menjadi kuning.

```
h1 {
  background-color: yellow;
  color: red;
}
```

COBALAH

```
p.bright {color: red; background-color: yellow}
    <p> Bagian ini akan berwarna merah dengan latar
belakang kuning. </p>

    h1 {
                                color: red;
    }
    <h1>Teks ini berwarna merah.</h1>
```

GLOSARIUM

Color adalah warna elemen. Untuk elemen yang berisi teks, color adalah warna teks latar depan. Nilainya dapat berupa nama warna (misalnya `black`, `red`), atau nilai heksadesimal atau desimal warna (misalnya `#ff0000` atau `rgb(255, 0, 0)`).

Background-color adalah warna latar belakang elemen.

3.4 Perataan Teks & Posisi

Perataan Teks

MENGAPA

Anda dapat meratakan teks ke kiri (default), ke kanan, teks tengah, atau menggunakan teks yang diratakan (rata kanan-kiri). Anda juga dapat melakukan inden untuk baris pertama blok teks.

CSS menawarkan sejumlah unit untuk menerangkan panjang yang menentukan tampilan teks di media keluaran seperti layar atau di kertas. Anda dapat menggunakan piksel (px), sentimeter (cm), ataupun inci (in).

BAGAIMANA

Ratakan teks menggunakan properti `text-align`; lakukan inden pada teks dengan `text-indent`

COBALAH

```
<style>
        .body {
            text-align: justified;
            text-indent: 10px; }
</style>

<p class="body">
    Paragraf ini akan dimulai dengan jarak 10 piksel dari
margin. Baris selanjutnya akan dimulai dari margin seperti biasa dan
semua teks dalam paragraf akan diratakan, sehingga tepi kanan dan
kiri akan lurus.
</p>
```

Posisi

MENGAPA

Properti `position` menentukan jenis metode penetapan posisi yang digunakan untuk elemen. Terdapat 4 jenis:

1. **Static**
Elemen diposisikan sesuai alur normal halaman. Ini adalah setelan default.
2. **Fixed**
Elemen tetap berada di posisi yang sama, meski halaman di-scroll.
3. **Relative**
Elemen diposisikan relatif terhadap posisi normalnya.
4. **Absolute**

Elemen diposisikan relatif terhadap elemen awal yang ditempatkan paling dekat (elemen dengan nilai `position` yang telah ditentukan). Jika tidak memiliki elemen awal yang ditempatkan paling dekat, isi dokumen akan digunakan dan elemen akan bergerak saat halaman di-scroll.

COBALAH

```
<style>
        .box1 {
            position: relative;
            left: 10px; }
        .box2 {
            position: fixed;
            bottom: 20px; }
</style>
```

```
<p class="box1">
    Paragraf ini akan dimulai dengan jarak 10 piksel
dari margin.
</p>
<p class="box2">
    Paragraf ini akan dimulai dengan jarak 20 piksel
dari margin dan akan selalu berada di tempat yang sama meskipun
halaman di-scroll.
</p>
```

GLOSARIUM

Text-indent adalah ruang kosong di sisi kiri untuk baris pertama blok teks.

Text-align adalah perataan untuk setiap blok teks secara keseluruhan.

3.5 Gambar Latar

MENGAPA

Properti `background-image` digunakan untuk menyetel gambar sebagai latar belakang elemen.

BAGAIMANA

Gunakan properti `background-image` dengan jalur file:

```
h1 {background-image: url(/images/gradient.png);}
```

TIPS: Jalur untuk gambar latar bersifat relatif terhadap dokumen yang berisi aturan, sehingga jika Anda menggunakan `background-image: url (gradient.png)` di file CSS pada `/css/global.css`, browser akan mencari `gradient.png` dalam direktori CSS yang sama.

COBALAH

Terapkan gambar latar ke elemen isi dokumen web.

3.6 Cascade

MENGAPA

Terkadang, lebih dari satu aturan stylesheet dapat diterapkan ke satu elemen CSS tertentu. Misalnya, satu aturan dapat digunakan untuk membuat semua elemen di tag `<p>` berwarna pink, sementara aturan lain membuatnya berwarna biru. Dalam kasus semacam ini, harus ada

cara untuk menentukan aturan stylesheet mana yang diterapkan ke elemen.
Cascade pada dasarnya menentukan deklarasi mana yang diterapkan ke elemen.

BAGAIMANA

Karena satu elemen berpotensi memiliki satu deklarasi atau lebih yang diterapkan ke elemen tersebut, 'Cascading' adalah metode yang digunakan untuk menentukan deklarasi mana yang akhirnya diterapkan. Dalam kasus deklarasi yang bentrok, deklarasi terakhir dan paling spesifik yang akan digunakan.

COBALAH

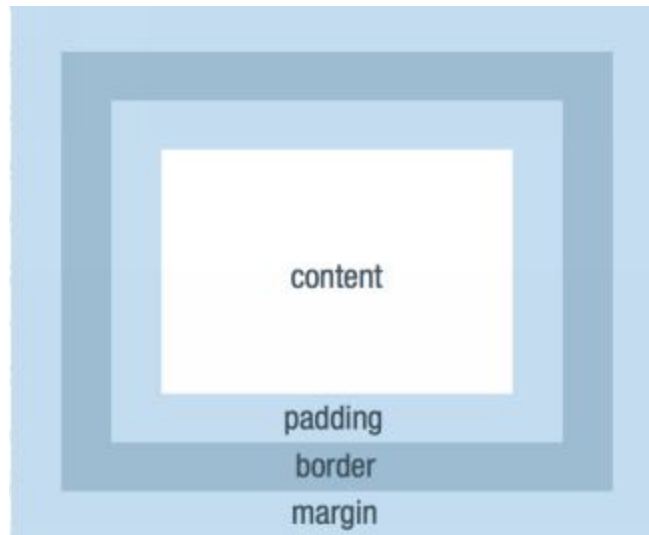
```
<style>
    div {
        color: blue;
        font-style: italic; }
    p {
        color: pink; }
</style>

<div>
<p> Bagian ini akan berwarna Pink. </p>
</div>
```

3.7 Model Kotak

APA

Dalam dokumen, setiap elemen diwakili dengan kotak persegi. Kotak ini disebut 'Model Kotak'. Pada dasarnya ini adalah kotak yang berada mengelilingi elemen HTML. Kotak ini terdiri dari 4 area tepian: margin, border, padding, dan konten.
Gambar di bawah mengilustrasikan model kotak:



Konten adalah area yang berisi konten elemen HTML sebenarnya: misalnya teks atau gambar.

Padding adalah area di sekitar konten.

Border adalah border! Border ini mengelilingi padding. Border dapat memiliki properti seperti `border-style`, `border-width`, dan `border-color`.

Margin adalah ruang kosong antara dua elemen yang berdekatan: di atas, kanan, bawah, atau kiri.

3.7.1 Border

MENGAPA

Properti yang mengontrol apakah elemen memiliki border, serta gaya border, disebut properti border.

Border berikut memiliki berbagai fitur seperti gaya, warna, lebar, dll.

BAGAIMANA

Setiap sisi border (atas, bawah, kiri, kanan) dapat memiliki gaya, lebar, atau warna yang berbeda. Border memiliki 3 properti utama:

- **border-style**
 - Properti ini menentukan gaya border, apakah itu *dotted*, *dashed*, atau *double*.
 - Nilai default untuk `border-style` adalah *none*.
 - Gaya salah satu dari empat sisi elemen dapat disetel dengan `border-top-style`, `border-bottom-style`, `border-left-style`, dan `border-right-style`.
- **border-width**
 - Properti ini menentukan ketebalan border. Nilainya yang memungkinkan adalah *thin*, *medium*, *thick*, atau nilai panjang dalam piksel.
 - Nilai default untuk `border-width` adalah *medium*.
 - Ketebalan salah satu dari empat sisi elemen dapat disetel dengan

`border-top-width`, `border-bottom-width`, `border-left-width` ,
dan `border-right-width` .

- **border-color**
 - Properti ini menentukan warna border. Nilainya dapat berupa nama warna atau nilai heksadesimal.
 - Warna defaultnya adalah *black*.
 - Warna salah satu dari empat sisi elemen dapat disetel dengan `border-top-color`, `border-bottom-color`, `border-left-color` , dan `border-right-color` .

COBALAH

```
<head>
<link rel="stylesheet" type="text/css" href="boxmodel.css">
</head>
```

boxmodel.css:

```
table {
border-top-width: thin;
border-bottom-width: thick;
border-top-color: red;
border-bottom-color: yellow;
border-top-style: dotted;
}

p {
border-width: thick;
border-color: aqua;
border-style: dashed;
}
```

Kini tambahkan tabel dan elemen para, simpan, dan jalankan. Seharusnya Anda melihat properti border di keluaran.

GLOSARIUM

Border adalah area antara padding dan margin.

Border-style adalah cara border ditampilkan (atau tampilan border): misalnya titik-titik atau garis putus-putus

Border-width adalah ketebalan border.

Border-color adalah warna border.

3.7.2 Margin dan Padding

MENGAPA

Jarak antara konten elemen dan border disebut padding.

Ruang antara border elemen dan bagian di dekatnya (ke atas, kanan, bawah, atau kiri) disebut margin.

BAGAIMANA

Properti margin disebut *margin*, sementara properti padding disebut *padding*. Properti ini dapat diterapkan ke empat sisi gambar: atas, bawah, kiri, dan kanan.

Karena itu, properti margin suatu elemen adalah: `margin-top`, `margin-bottom`, `margin-left`, dan `margin-right`; sementara properti padding-nya adalah `padding-top`, `padding-bottom`, `padding-left`, dan `padding-right`.

Nilai untuk margin dan padding dapat diberikan dalam inci atau piksel (atau satuan ukuran lain).

COBALAH

```
<head>
<link rel="stylesheet" type="text/css" href="margins.css">
</head>
```

margins.css:

```
    p.one {
margin: 0.1in;
padding: 0.1in;
background-color: red;
    }
    p.two {
margin: 0.2in;
padding: 0.2in;
background-color: blue;
    }
    p.three {
margin: 20px;
padding: 10px;
background-color: yellow;
    }
    p.four {
margin: 40px;
background-color: red;
    }
```

Kini tambahkan 4 elemen para dengan nama kelas seperti *one*, *two*, *three*, dan *four* secara berurutan; dan jalankan. Seharusnya properti margin dan padding akan muncul di keluaran. Ingat untuk menyimpan pekerjaan Anda sebelum melanjutkan ke sesi berikutnya.

GLOSARIUM

Margin adalah ruang antara konten elemen dan elemen di dekatnya.

Padding adalah jarak antara elemen dan border-nya.