Andre Pratama

CSS Uncover

Panduan Belajar CSS untuk Pemula

Duniailkom

CSS Uncover

Panduan Belajar CSS untuk Pemula

Duniailkom

Contents

Terimakasih

Terimakasih sudah mendownload sample buku **CSS Uncover** Duniailkom. Materi yang tersedia diambil dari pembahasan eBook **CSS Uncover**. Ini hanya sekedar perkenalan agar rekan-rekan bisa melihat bagaimana penulisan serta layout buku.

Donasi pembelian buku ini jadi dukungan tak ternilai bagi saya untuk terus bisa berkarya. Kedepannya, saya (dan duniailkom) akan berusaha menghadirkan lebih banyak eBook serta tutorial berkualitas lain terkait web programming.

eBook **CSS Uncover** versi full bisa didapat dengan donasi **Rp 50.000** melalui transfer bank (tersedia Bank Mandiri, BCA, BRI, dan BNI), atau juga bisa melalui transfer pulsa (kartu AS). Untuk pemesanan, tanya-tanya, bisa menghubungi saya di duniailkom@gmail.com.

Salam,

Andre Pratama

www.duniailkom.com

1. Berkenalan Dengan CSS (Bab 1 Full)

CSS adalah dunianya web design. Jika anda ingin mempelajari cara mendesain web, CSS mutlak harus dikuasai. Dalam bab pertama buku **CSS Uncover** ini kita akan membahas pengertian CSS, fungsi CSS, dan melihat sekilas contoh penggunaan CSS dalam proses pembuatan web.

1.1 Pengertian CSS

CSS merupakan singkatan dari **Cascading Style Sheet**. CSS digunakan untuk mengubah tampilan (*style*) dari halaman web. Sebagaimana yang kita ketahui, halaman web modern terdiri dari 3 komponen dasar: **HTML** untuk membuat struktur, **CSS** untuk tampilan, dan **JavaScript** untuk interaksi.

Jika halaman web diibaratkan sebuah bangunan, CSS adalah tampilan luar dari bangunan tersebut, seperti warna dinding atau warna atap. Kerangka dasarnya dibuat dari HTML. Dengan demikian, kita bisa dengan mudah menukar warna dinding bangunan tanpa perlu mengubah struktur dasarnya.

Begitu pula dengan halaman web. Menggunakan CSS, kita bisa mengubah tampilan website tanpa perlu menyentuh kode HTML. Apabila saat ini website kita memiliki warna mayoritas merah, minggu depan bisa menjadi biru hanya dengan menukar beberapa baris kode CSS.

Mengutip dari wikipedia:

> *"Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language".*

Terjemahan bebasnya:

> *"Cascading Style Sheets (CSS) adalah bahasa style sheet yang digunakan untuk mengatur tampilan dan format dari sebuah dokumen yang ditulis dengan bahasa markup".*

Terdapat 2 istilah penting yang perlu penjelasan tambahan, yakni: **bahasa style sheet** (*style sheet language*) dan **bahasa markup** (*markup language*).

Istilah pertama: **style sheet language** adalah format bahasa khusus yang terdiri dari kumpulan kode untuk mengatur tampilan (style) dari sebuah dokumen. Sebagaimana yang akan kita lihat nanti dari sejarah CSS, pada awal perkembangannya terdapat berbagai variasi *style sheet language* yang bisa digunakan, dimana salah satunya adalah CSS.

Istilah kedua, **markup language** merujuk kepada dokumen yang dibuat menggunakan "tanda" atau "mark". Salah satu contoh dari *markup language* ini adalah **HTML** (*Hypertext Markup Language*). Walaupun begitu, CSS tidak hanya digunakan untuk HTML saja, tapi bisa untuk bahasa markup lain seperti **XML** (*Extensible Markup Language*) dan **SVG** (*Scalable Vector Graphics*).

Kata **Cascade** dari kepanjangan CSS juga perlu kita bahas. Dalam bahasa inggris, *cascade* berarti "*air terjun kecil, riam, jeram, mengalir/berpancaran kebawah*". Dimana maknanya adalah: *sesuatu yang mengalir dari atas ke bawah.*

Di dalam CSS, *style* atau aturan tampilan yang dibuat bisa saja saling menimpa satu sama lain, tergantung dari posisinya dan ke-spesifikan kode CSS tersebut.

Sebagai contoh, jika pada baris pertama kode CSS kita membuat perintah untuk mengubah warna paragraf menjadi **biru**, di baris kedua kita bisa menulis kembali perintah yang sama, tetapi kali ini mengubah warna tersebut menjadi **merah**. Konsep **cascading** ini akan kita bahas dengan lebih detail pada bab tersendiri.

Sebagai kesimpulan, dalam pengertian sederhana CSS adalah *"format bahasa khusus yang digunakan untuk mengatur tampilan dari halaman web".*

Sama seperti HTML, CSS bukanlah sebuah *bahasa pemrograman komputer*, tapi bahasa struktur (*structural language*). CSS terdiri dari kode-kode sederhana berupa perintah 'style' tanpa fitur bahasa pemrograman seperti *variabel, logika if, fungsi, dll.* Oleh karena itu, CSS relatif mudah untuk dipahami.

1.2 Fungsi CSS

Seperti yang telah kita singgung sebelumnya, **CSS** berfungsi untuk mengatur tampilan (*style*) dari sebuah dokumen. Khusus dalam buku ini dokumen yang dimaksud adalah **HTML**.

Dengan menggunakan CSS kita bisa mengatur (hampir) seluruh tampilan dari HTML, seperti warna teks, gambar backround, besar font, posisi judul, tampilan layout, dll. Ditambah lagi CSS3 membawa banyak fitur lanjutan seperti *color gradient*, *transitions*, dan *animations*.

Keuntungan lain dari CSS, ia bisa digunakan oleh banyak dokumen HTML sekaligus. Sebagai contoh, untuk website yang terdiri dari ribuan halaman, kode CSS yang diperlukan bisa ditempatkan pada 1 file saja. Dengan mengubah beberapa baris kode pada file ini, tampilan seluruh halaman website akan ikut berubah.

Selain itu, CSS memiliki fitur '**media type**' untuk mendeteksi tipe perangkat yang digunakan ketika mengakses halaman web, apakah itu dari layar komputer/smartphone, printer, screen reader, dll. Dengan demikian, kita bisa membuat style yang berbeda-beda untuk masing-masing perangkat ini. Misalnya jika sebuah halaman akan di print, kita bisa menghapus gambar background atau mengubah warna text agar lebih jelas ketika di print.

Pada CSS3, fitur media type disempurnakan lagi dengan '**media query**'. Menggunakan **media query**, kita bisa membuat halaman web yang dapat 'menyesuaikan diri' dengan ukuran layar.

Tipe website seperti ini dikenal dengan **Responsive Web Design (RWD)**, atau cukup disingkat dengan **Web Responsive**.

Selain hal diatas, masih banyak fitur canggih CSS lainnya yang tidak bisa didapat jika menggunakan HTML saja. Dalam buku ini kita akan membahas sebagian besar diantaranya. Mulai dari dasar penulisan CSS, pengertian *selector* dan *property*, tipe-tipe *selector*, hingga cara membuat *web responsive*.

Khusus untuk **web responsive**, Google pun telah memutuskan untuk memberi nilai tambah bagi website '*mobile friendly*'. Dengan demikan, mau tidak mau anda harus menguasai cara membuat *responsive web design* mulai saat ini.

1.3 Contoh Penggunaan CSS

Agar perkenalan kita dengan CSS semakin lengkap, saya akan mengajak anda untuk lihat sekilas contoh penggunaan CSS di dalam dokumen HTML. Anda tidak perlu memahami kode CSS yang digunakan, nantinya akan kita bahas dengan lebih detail sepanjang buku ini.

Perhatikan kode HTML berikut:

01.simple.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Belajar CSS</title>
</head>
<body>
  <h1>Belajar CSS</h1>
  <p>Mohon jangan diganggu, lagi serius belajar CSS.</p>
  <h2>Belajar Web Design</h2>
  <p>CSS adalah dunianya web design.
    Jika anda ingin mempelajari cara mendesain web,
    CSS mutlak harus dikuasai.</p>
</body>
</html>
```

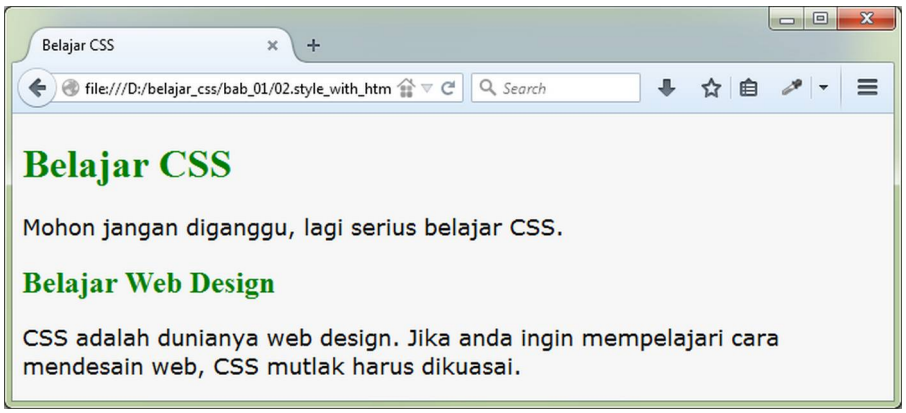
Kode diatas hanyalah file HTML sederhana yang terdiri dari tag `<h1>`, `<h2>` dan 2 buah tag `<p>`. Jika anda telah memahami HTML, tentunya tidak asing dengan fungsi dari tag tersebut. Berikut tampilannya di dalam web browser Mozilla Firefox:

Gambar: Contoh halaman HTML Sederhana

Sekarang saya akan mencoba mengubah *design* halaman HTML diatas. Terdapat 2 alternatif, menggunakan tag HTML atau dengan CSS. Jika menggunakan tag HTML, saya bisa memodifikasinya menjadi seperti berikut ini:

02.style_with_html.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Belajar CSS</title>
</head>
<body bgcolor="#F7F7F7">
  <h1><font color="green">Belajar CSS</font></h1>
  <p><font face="verdana" size="4">Mohon jangan diganggu,
  lagi serius belajar CSS.</font></p>
  <h2><font color="green">Belajar Web Design</font></h2>
  <p><font face="verdana" size="4">CSS adalah dunianya web design.
  Jika anda ingin mempelajari cara mendesain web,
  CSS mutlak harus dikuasai.</font></p>
</body>
</html>
```



Gambar: Hasil Style dengan HTML

Sekarang tampilan halaman sudah berubah, namun terdapat beberapa masalah dengan kode diatas:

- HTML yang seharusnya berfungsi untuk membuat struktur, juga digunakan untuk menangani tampilan web.
- Apabila saya ingin menambahkan paragraf baru, saya harus ingat untuk selalu menulis tag `<font>` serta atributnya (warna dan ukuran font). Bayangkan jika di dalam halaman tersebut akan ditambah 20 paragraf baru di waktu yang berbeda-beda.
- Akan susah jika dikemudian hari saya ingin mengubah warna teks paragraf. Saya harus mengubah tag `<font>` satu persatu pada setiap halaman.

Dengan CSS, saya bisa menghasilkan design tampilan yang sama menggunakan kode yang lebih fleksibel:

03.style_with_css.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Belajar CSS</title>
  <style>
  body{
    background-color:#F7F7F7;
  }
  h1,h2{
    color:green;
  }
  p{
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 18px;
  }
  </style>
</head>
<body>
  <h1>Belajar CSS</h1>
  <p>Mohon jangan diganggu, lagi serius belajar CSS.</p>
  <h2>Belajar Web Design</h2>
  <p>CSS adalah dunianya web design.
    Jika anda ingin mempelajari cara mendesain web,
    CSS mutlak harus dikuasai.</p>
</body>
</html>
```

Perhatikan tambahan tag `<style>` di bagian `<head>` HTML. Inilah kode CSS yang digunakan pada halaman ini. Dengan CSS, masalah dari kode HTML sebelumnya bisa diatasi:

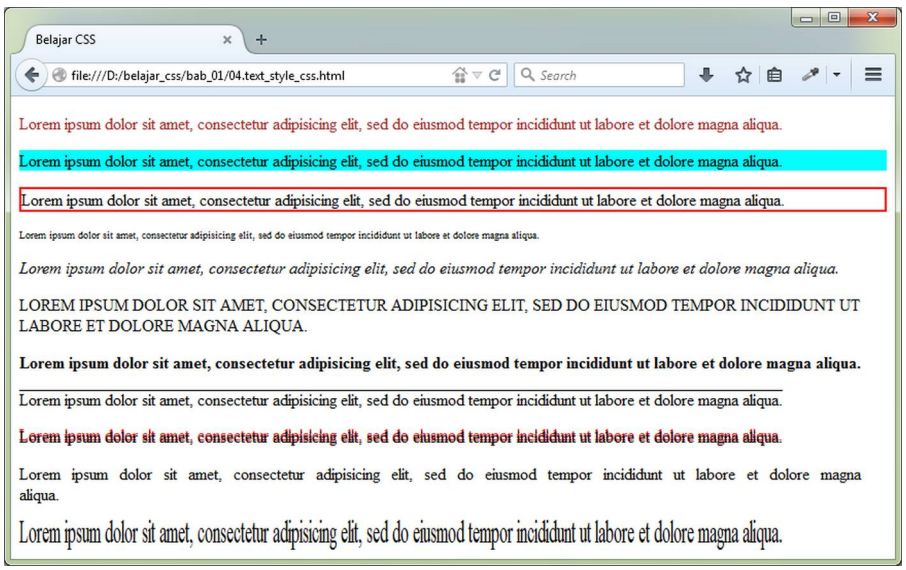
- Kode CSS tersebut sepenuhnya terpisah dari struktur HTML. Sehingga tidak saling 'tercampur'.
- Apabila saya ingin menambahkan 1, 2 atau 1000 paragraf baru menggunakan tag <p>, secara otomatis tag tersebut akan di design menyesuaikan dari kode CSS. Tanpa perlu mengubah paragraf satu persatu.
- Jika di kemudian hari ingin menukar warna paragraf dari hijau menjadi biru, saya tinggal mengubah 1 baris kode CSS saja, dan seluruh teks akan ikut berubah.
- Kode CSS diatas bisa 'diangkat' ke sebuah halaman khusus (**external CSS**), kemudian di-**link** kepada setiap halaman HTML. Dengan cara ini, seluruh design website hanya butuh 1 file CSS saja.

Sebagai contoh tambahan, dalam halaman HTML berikut ini saya tampilkan beberapa design teks yang bisa dihasilkan dari CSS:

04.text_style_css.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Belajar CSS</title>
</head>
<body>
  <p style="color:brown;">
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
  eiusmod tempor incididunt ut labore et dolore magna aliqua.
  </p>
  <p style="background-color:aqua;">
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
  eiusmod tempor incididunt ut labore et dolore magna aliqua.
  </p>
  <p style="border: 2px solid red;">
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
  eiusmod tempor incididunt ut labore et dolore magna aliqua.
  </p>
  <p style="font-size:10px;">
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
  eiusmod tempor incididunt ut labore et dolore magna aliqua.
  </p>
  <p style="font-style:italic;">
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
  eiusmod tempor incididunt ut labore et dolore magna aliqua.
  </p>
  <p style="text-transform: uppercase;">
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
  eiusmod tempor incididunt ut labore et dolore magna aliqua.
```

```
    </p>
    <p style="font-weight:bold;">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua.
    </p>
    <p style="text-decoration: overline;">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua.
    </p>
    <p style="text-shadow: red 0 -2px;">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua.
    </p>
    <p style="word-spacing: 7px;">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua.
    </p>
    <p style="transform: scaleY(2);">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua.
    </p>
</body>
</html>
```

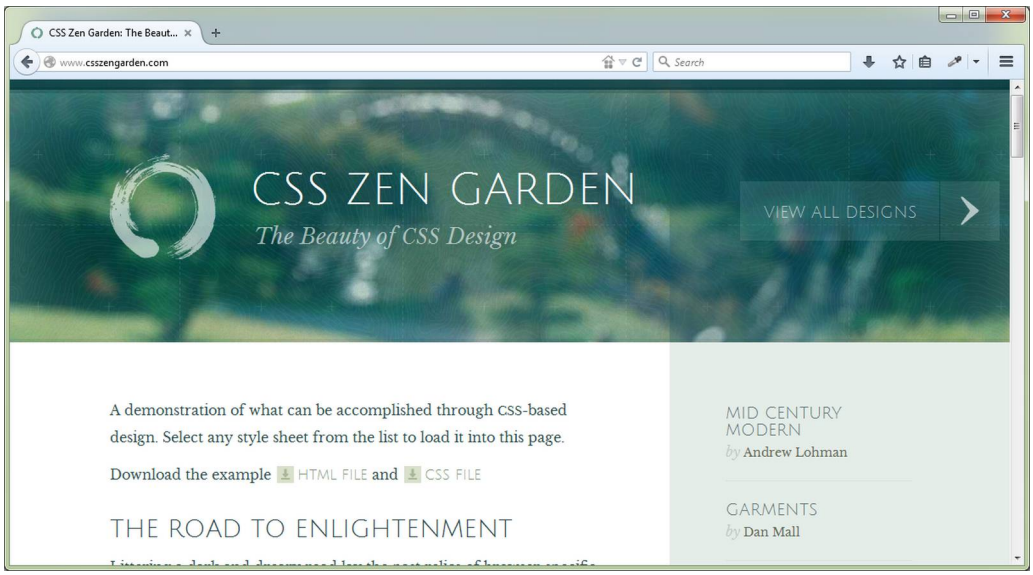


Gambar: Berbagai variasi design text menggunakan CSS

Tampilan design text diatas (dan banyak efek lain) akan kita bahas secara detail dalam bab-bab berikutnya.

1.4 CSS Zen Garden

CSS Zen Garden adalah contoh klasik 'unjuk gigi' kemampuan CSS. Situs CSS Zen Garden beralamat di www.csszengarden.com[1]. Situs ini sengaja dibuat untuk memperlihatkan kreatifitas penggunaan CSS, atau boleh dibilang sebagai 'demo' dari kemampuan CSS.

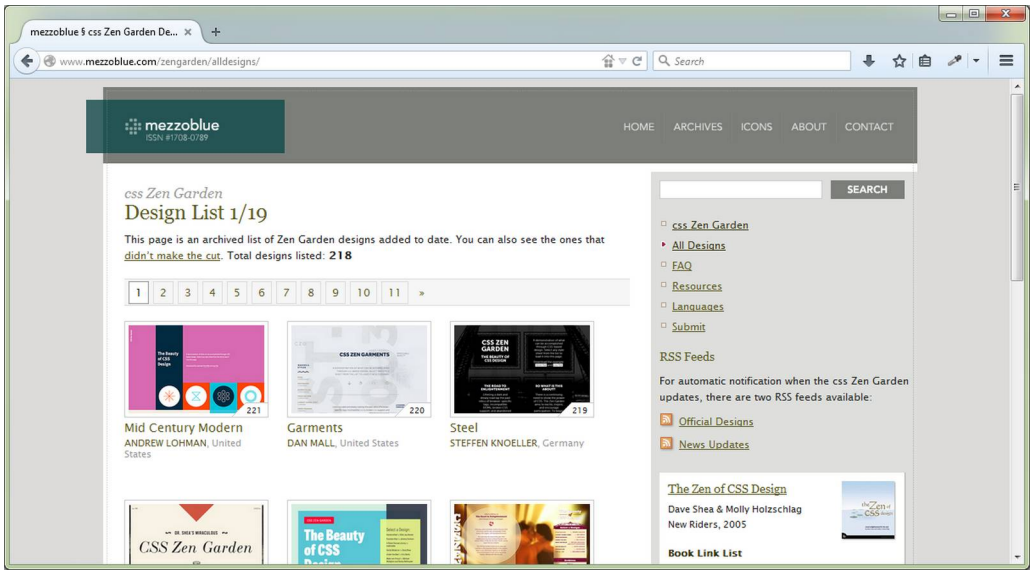


Gambar: Tampilan halaman CSSZenGarden

Di dalamnya disediakan sebuah halaman HTML tanpa style apa-apa dan setiap web designer ditantang untuk mengubah tampilan halaman tersebut menggunakan CSS. Jika berminat, anda pun juga bisa mengajukan design CSS ke situs ini.

Pada saat saya mengakses situs CSS Zen Garden (Juni 2015), terdapat 218 design CSS yang tersedia. Setiap design ini menggunakan kode HTML yang sama.

[1] http://www.csszengarden.com/



Gambar: Berbagai style halaman pada CSSZenGarden

Situs CSS Zen Garden sempat tidak lagi di update sejak tahun 2008, namun mulai tahun 2013, situs ini kembali 'ramai' dengan menghadirkan design baru dari CSS3.

Dalam bab pembuka buku **CSS Uncover** ini kita telah membahas pengertian CSS dan melihat sekilas fungsi dan fitur dari CSS. Selanjutnya, saya akan mengajak anda untuk mengenal lebih dalam tentang sejarah lahirnya CSS serta melihat bagaimana perkembangan CSS hingga saat ini.

2. Aturan Dasar Penulisan CSS (Contoh Bab 4)

Dalam 3 bab awal buku ini kita telah membahas pengertian CSS, sejarah CSS, serta menyiapkan 'perangkat perang', yakni web browser dan text editor. Mulai dari bab ini ke depan kita akan masuk ke materi coding CSS, yang dibuka dengan pembahasan mengenai aturan dasar penulisan kode CSS.

2.1 Pengertian Selector, Declaration, Property dan Value CSS

Selector, **declaration**, **property** dan **value** adalah inti dari CSS. 90% kode CSS yang kita buat hanya terdiri dari ke-4 element ini. Agar lebih mudah dipahami, perhatikan kode CSS berikut:

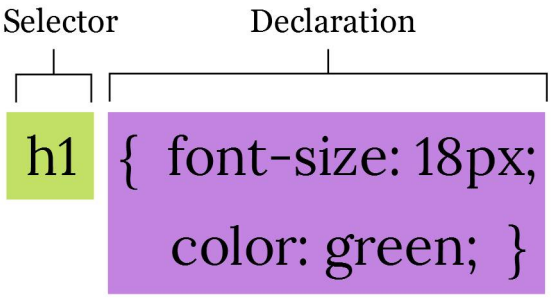
```
h1  { font-size: 18px;
      color: green;  }
```

Arti dari kode diatas adalah: *temukan seluruh tag `<h1>` di dalam halaman HTML, kemudian set ukuran font sebesar 18 pixel dan set warna teks menjadi hijau.*

Kode "**h1**" dari contoh diatas adalah *selector* CSS. **Selector** digunakan untuk mencari bagian mana dari HTML yang ingin di-*style*. CSS menyediakan beragam jenis selector, mulai dari yang sederhana seperti *element selector*, hingga yang cukup rumit seperti *pseudo selector*.

Dalam contoh diatas, "h1" dikenal sebagai **element selector** atau **tag selector**. Element selector digunakan untuk 'mencari' seluruh element HTML yang menggunakan tag tertentu, dimana dalam contoh kita adalah tag `<h1>`. Lebih jauh tentang selector akan saya bahas pada bab berikutnya.

Setelah penulisan *selector*, berikutnya: *declaration*. **Declaration** adalah kumpulan aturan *style* CSS yang berada di antara tanda kurung kurawal. Gambar berikut akan memperjelas pengertian *selector* dan *declaration*:

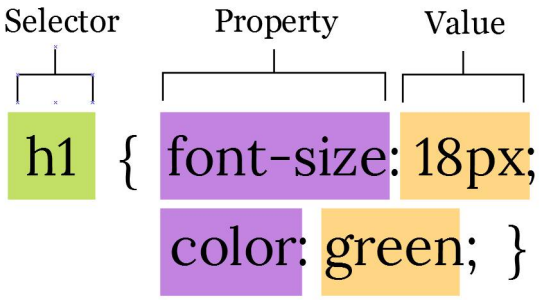


Gambar: Selector dan Declaration Block CSS

Di dalam *declaration*, terdapat pasangan *property* dan *value*. **Property** adalah jenis *style*, atau bagian apa yang akan diubah dari sebuah tag HTML. Misalnya jenis font, ukuran huruf, warna background, dll. Terdapat ratusan *property* di dalam CSS dan terus bertambah (terutama untuk property CSS3).

Agar bisa digunakan, sebuah *property* memiliki nilai satuan atau *value*. **Value** ini sangat bergantung kepada jenis property, misalnya untuk ukuran font bisa menggunakan satuan *pixel, point*, dll. Tapi untuk property warna (*color*) kita harus menggunakan value berupa nama warna atau kode **RGB**.

Keempat elemen inilah yang membangun CSS. Sepanjang buku ini kita akan membahas dengan detail tentang selector, property dan value CSS. Gambar berikut menampilkan bagian mana yang disebut selector, property dan value CSS:



Gambar: Selector, Property dan Value dalam CSS

2.2 Cara Penulisan Selector, Property dan Value

Seperti yang telah kita lihat sebelumnya, penulisan kode CSS cukup sederhana:

- Sebuah style di dalam CSS diawali dengan penulisan **selector**, yakni bagian apa dari HTML yang ingin diubah tampilannya.
- Setelah penulisan *selector*, seluruh **property** dan **value** (*declaration*) untuk selector tersebut harus berada di dalam tanda kurung kurawal.
- Penulisan property dengan value dipisah oleh tanda titik dua “ : ”.
- Antara satu property dengan property lain dipisah dengan tanda titik koma “ ; ”.

- Khusus untuk penulisan property terakhir, tanda titik koma “ ; ” boleh tidak ditulis, tapi sangat disarankan untuk tetap menambahkannya.

---

3. CSS Typography (Contoh Bab 7)

Typography adalah sebuah seni mengatur huruf/teks untuk membuatnya lebih mudah dibaca dan menarik secara visual. Teks merupakan konten utama dari hampir semua website. Walaupun anda membuat web yang fokus kepada multimedia (gambar dan video), namun tetap memerlukan teks untuk bagian keterangan/deskripsi. Menu navigasi sebuah website juga hampir semuanya terdiri dari teks.

Dalam bab ini kita akan fokus membahas cara penggunaan property CSS yang berkaitan dengan *typography* atau teks. Selain itu, saya juga akan membahas konsep-konsep dasar property CSS seperti nilai **length** (pixel, em, %, dll), nilai **color**(#RBG, HSL) dan **CSS3 vendor prefix**.

Kecuali dinyatakan lain, hampir semua property *typography* di dalam CSS akan diturunkan ke *child element*, atau dikenal sebagai **inherit property**.

3.1 Property font-family

Pembahasan mengenai property CSS kita mulai dengan **font-family**. Property ini digunakan untuk mengatur jenis font. Nilai dari property **font-family** adalah nama font yang ingin digunakan. Berikut contoh penulisannya:

```
p {
  font-family: Arial;
}
```

Kode CSS diatas akan men-set jenis font **Arial** pada semua paragraf. Akan tetapi, dari manakah font ini diambil?

Property **font-family** akan mencari font di dalam **komputer client**, yakni di dalam komputer pengunjung web browser, bukan di dalam komputer server tempat website berada.

Ini berarti kode CSS diatas hanya akan ditampilkan dengan font **Arial** apabila di dalam komputer pengunjung terdapat font bernama "Arial". Jika tidak ditemukan, web browser akan menentukan sendiri font yang akan digunakan (tergantung settingan default web browser tersebut).

Bagi pengguna Windows, seluruh file font yang tersedia di komputer bisa dilihat dari **Control Panel -> Appearance and Personalization -> Fonts**.

Untuk mengantisipasi tidak tersedianya sebuah font, kita bisa memberikan alternatif font pengganti, seperti contoh berikut:

```
p {
  font-family: Arial, Helvetica, sans-serif;
}
```

Kali ini, kode CSS diatas memberikan pilihan kepada web browser: *Cari seluruh tag <p>, kemudian gunakan font **Arial**. Apabila tidak ditemukan, gunakan font **Helvetica**. Apabila tidak ditemukan juga, gunakan font generic dengan tipe **sans-serif**.*

CSS tidak membatasi seberapa banyak font alternatif yang bisa dipersiapkan, bisa 1, 3, bahkan 10 font:

```
body {
font-family: Arial, Helvetica, Verdana, Geneva, Impact, Charcoal, sans-serif;
}
```

Kode CSS diatas memberikan 7 alternatif font yang bisa dipilih, mulai dari **Arial** sebagai prioritas utama, kemudian **Helvetica** (jika Arial tidak tersedia), lalu **Verdana** (jika Arial dan Helvetica tidak ditemukan), dst. Walaupun kita bisa mempersiapkan banyak font alternatif, umumnya web designer hanya menulis maksimal 3 jenis font (2 font dan 1 font generic).

Perhatikan untuk setiap nama font dipisah dengan tanda koma: “ , ”. Khusus untuk nama font yang terdiri dari beberapa kata atau mengandung spasi, penulisannya harus berada diantara tanda kutip, seperti *“Times New Roman“*:

```
p {
  font-family: "Times New Roman", Georgia, serif;
}
```

Property **font-family** termasuk ke dalam properti yang diturunkan ke dalam *child element*, atau dikenal sebagai **inherit property**. Oleh karena itu, jika kita ingin mengunakan sebuah jenis font untuk seluruh halaman, tinggal menempatkan property ini di dalam selector body:

```
body {
  font-family: "Times New Roman", Georgia, serif;
}
```

Kode CSS diatas akan membuat seluruh element HTML yang berada di dalam tag <body> mengunakan font *“Times New Roman“*.

Mengenal ‘Generic Font’

Jika anda perhatikan, selain menulis nama font, pada pilihan font terakhir terakhir saya menambahkan kata ‘**serif**’ dan ‘**sans-serif**’. Ini sebenarnya bukanlah nama font, tetapi sebuah *jenis font*. Di dalam CSS, ini dikenal dengan sebutan ‘*generic font*’.

Generic font berfungsi sebagai alternatif terakhir ketika font yang kita inginkan tidak tersedia. CSS menyediakan 5 jenis *generic font*:

- **serif**
- **sans-serif**
- **monospace**
- **cursive**
- **fantasy**

Generic Font: serif

Dalam ilmu *typography*, **serif** adalah istilah untuk menyebut 'kaki'/garis tambahan pada ujung setiap huruf. Serif akan membuat sebuah teks lebih mudah dibaca, terutama dalam format media cetak seperti kertas atau buku.

Beberapa contoh font yang termasuk kategori serif adalah *Times New Roman, Georgia, Lucida Bright, Lucida Fax, Palatino, "Palatino Linotype"* dan *Palladio.*

Generic Font: sans-serif

Sans-serif adalah sebutan untuk kelompok font yang tidak memiliki kaki/garis tambahan. Kata '*sans'* berasal dari bahasa perancis yang berarti '*tanpa'*, sehingga *sans-serif* adalah '*tanpa-serif'*. Font dengan jenis *sans-serif* banyak digunakan sebagai tipe font utama di dalam website, karena dianggap lebih mudah dibaca di media elektronik.

Contoh font sans-serif adalah *Arial, Verdana, Trebuchet MS, Helvetica,* dan *Calibri.*

Gambar: Perbedaan font serif dan sans-serif (sumber: wikipedia.org)

Generic Font: monospace

Font berjenis **monospace** adalah tipe font khusus dimana lebar setiap karakternya sama panjang. Sebagai contoh, di dalam font *reguler*, huruf 'i' akan mengambil tempat lebih sedikit daripada huruf 'w'. Namun di dalam font *monospace*, kedua huruf ini menggunakan ruang yang sama besar.

Font jenis *monospace* cocok digunakan untuk tulisan teknis yang membutuhkan ketelitian. Hampir semua teks editor programming menggunakan font *monospace*. Contoh dari font monospace adalah *Courier, Courier New,* dan *Andale Mono.*

Gambar: Tampilan font monospace: *Courier* (sumber: wikipedia.org)

Generic Font: cursive

Font **cursive** adalah jenis font yang meniru tulisan tangan atau kaligrafi. Font jenis ini cukup beragam dengan berbagai variasi huruf. Biasanya font *cursive* digunakan untuk bagian judul, sub judul, atau teks lain yang bukan bagian dari konten utama.

Contoh dari *font cursive* adalah *"Comic Sans", "Brush Script MT", "Lucida Calligraphy'* dan *"Lucida Handwriting".*

Generic Font: fantasy

Font dengan jenis **fantasy** adalah font yang bersifat visual dengan karakter font khusus seperti font *disney, matrix*, dll. Sama seperti *cursive*, font jenis ini relatif jarang digunakan sebagai text utama, tetapi biasanya digunakan untuk bagian teks yang butuh perhatian lebih seperti judul.

Contoh dari font *fantasy* adalah *Papyrus, Herculanum, Party LET, Curlz MT,* dan *Harrington.*

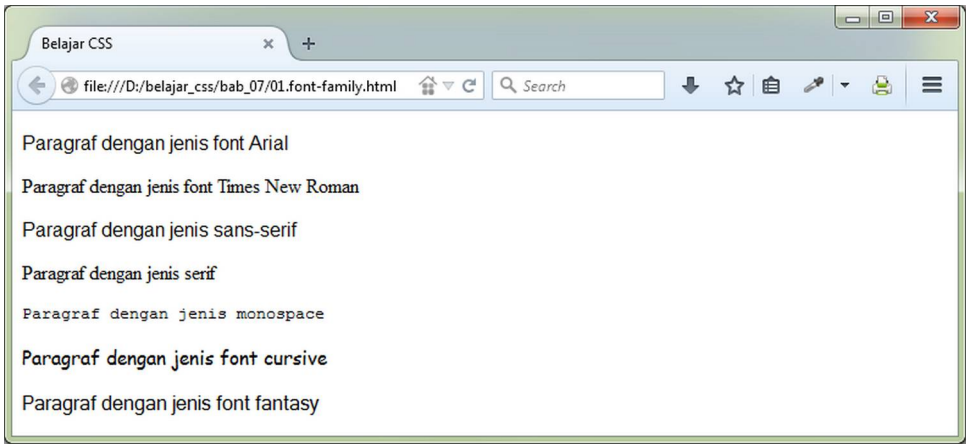
Kelima generic font ini bisa menjadi patokan seperti apa jenis font yang ingin digunakan. Sebagai contoh, jika memilih menggunakan font **Arial** sebagai font utama, anda bisa menambahkan **sans-serif** sebagai generic font, sehingga apabila di komputer client tidak terdapat font **Arial**, web browser akan mencari font lain dengan jenis *sans-serif* (yang sejenis dengan Arial).

Agar bisa berfungsi sebagaimana mestinya, font generic harus ditempatkan di pilihan terakhir **font-family**. Penulisan font generic juga harus ditulis dengan huruf kecil dan tanpa tanda kutip.

Berikut adalah contoh kode CSS dengan menggunakan berbagai variasi font:

01.font-family.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Belajar CSS</title>
  <style>
  p.arial{
    font-family: Arial, Helvetica, sans-serif;
  }
  p.times{
    font-family: "Times New Roman", Georgia, serif;
  }
  p.sans-serif{
    font-family: sans-serif;
  }
  p.serif{
    font-family: serif;
  }
  p.monospace{
    font-family: monospace;
  }
  p.cursive{
    font-family: cursive;
  }
  p.fantasy{
    font-family: fantasy;
  }
  </style>
</head>
<body>
  <p class="arial">Paragraf dengan jenis font Arial</p>
  <p class="times">Paragraf dengan jenis font Times New Roman</p>
  <p class="sans-serif">Paragraf dengan jenis sans-serif</p>
  <p class="serif">Paragraf dengan jenis serif</p>
  <p class="monospace">Paragraf dengan jenis monospace</p>
  <p class="cursive">Paragraf dengan jenis font cursive</p>
  <p class="fantasy">Paragraf dengan jenis font fantasy</p>
</body>
</body>
</html>
```



Gambar: Tampilan berbagai jenis font serta generic font CSS

4. CSS Box Model (Contoh Bab 8)

Salah satu konsep dasar yang wajib dipahami di dalam CSS adalah tentang **box model**. Dalam bab ini kita akan membahas dengan detail tentang CSS box model, yang terdiri dari lebar dan tinggi konten (**width** & **height**), **padding**, **border**, dan **margin**.

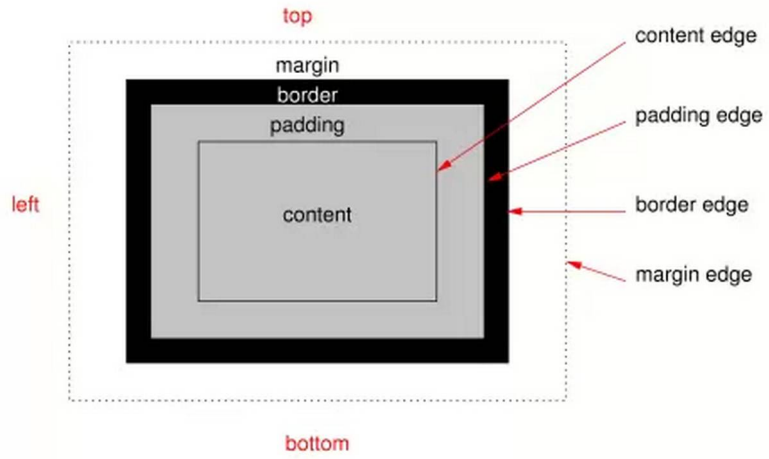
Selain itu juga akan dibahas tentang property *overflow*, *rounding-corner*, serta *CSS Reset*.

4.1 Pengertian CSS Box Model

CSS Box Model adalah sebuah konsep dimana setiap element yang terdapat pada halaman web diproses sebagai kotak (box). Mulai dari paragraf, header, form, gambar, logo hingga video, sebenarnya di tampilkan oleh web browser sebagai 'box'.

Sebagaimana layaknya 'kotak', masing-masing element HTML ini terdiri dari 4 lapisan, yakni: **konten** (isi), **padding**, **border** dan **margin**. Keempat 'lapisan' inilah yang membangun *CSS Box Model*.

Agar lebih mudah dipahami, perhatikan gambar berikut:



Gambar: CSS Box Model (sumber: www.w3c.org)

Konten atau teks dari sebuah element berada di bagian tengah. Kita bisa mengatur lebar dan tinggi konten ini menggunakan property **width** dan **height**.

Selanjutnya, terdapat *padding*. **Padding** adalah jarak antara konten dengan garis tepi (*border*) element. Sebagai contoh, jika kita membuat sebuah teks tanpa padding, maka teks tersebut akan mulai persis setelah garis tepi. *Padding* biasa ditambahkan supaya teks tidak menyentuh sisi dalam dari sebuah sel tabel.

Setelah *padding*, berikutnya: *border*. **Border** merupakan pembatas element. Kita bisa mengatur berbagai hal tentang *border*, seperti ketebalan, warna, dan jenis garis yang digunakan.

Di lapisan terakhir terdapat *margin*. **Margin** adalah 'spasi' dari sebuah element dengan element lain di sekelilingnya. *Margin* bersifat transparan dan digunakan agar setiap element tidak saling menempel satu sama lain.

Konsep **box model** sebenarnya cukup mudah dipahami. Namun dalam prakteknya, terdapat beberapa hal yang perlu penjelasan lebih lanjut, terutama tentang bagaimana keempat element ini saling berhubungan dengan element lain.

4.2 Property width dan height

Pembahasan mengenai *Box Model* akan kita mulai dari konten utama sebuah element. Untuk mengatur lebar dan tinggi element, bisa menggunakan property **width** dan **height**. Kedua property ini mendukung seluruh satuan nilai 'length' yang telah kita bahas dalam bab sebelumnya, seperti *px*, *em*, *%*, dan *rem*.

Berikut contoh penggunaan property *width* dan *height*:

01.width_and_height.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Belajar CSS</title>
  <style>
    h3 {
      width: 300px;
      height: 100px;
      background-color: yellow;
    }
    p {
      width: 800px;
      height: 150px;
      background-color: aqua;
    }
  </style>
</head>
<body>
  <h3>Belajar Box Model</h3>
  <p>Sebuah paragraf sederhana</p>
</body>
</html>
```

Gambar: Contoh pengggunaan property width dan height

Sesuai dengan property yang ditulis, tag `<h3>` ditampilkan dengan lebar **300 pixel** dan tinggi **100 pixel**, sedangkan tag `<p>` ditampilkan dengan lebar **800 pixel** dan tinggi **120 pixel**.

Selain property *width* dan *height*, dalam kode CSS diatas saya juga menambahkan property **background-color**. Property ini berfungsi untuk membuat warna latar belakang (*background*) dari sebuah element. Dengan demikian, kita bisa dengan mudah melihat ukuran sebenarnya dari element tersebut. Saya akan membahas mengenai property **backround** pada bab berikutnya.

Warna putih diantara kedua element ini merupakan *margin* bawaan web browser, kita akan membahas tentang margin beberapa saat lagi.