

## Bab III

### Cascading Style Sheet

*Cascading Style Sheet* (CSS) merupakan bahasa yang dapat digunakan untuk mengatur tampilan dari sebuah dokumen yang ditulis menggunakan bahasa *markup*. Walaupun CSS banyak sekali digunakan untuk mengatur tampilan dari sebuah halaman web yang ditulis menggunakan bahasa HTML, sebenarnya CSS dapat juga diaplikasikan pada dokumen XML (*eXtensible Markup Language*) lain seperti XML standar, SVG (*Scalable Vector Graphic*), maupun XUL (*XML User Interface Language*).

Jika bahasa HTML mengatur struktur dari dokumen web dan memberi tahu browser terkait fungsi dari beberapa elemen (seperti tautan ke dokumen lain, form masukan, ataupun tombol aksi), bahasa CSS memberikan instruksi kepada browser terkait dengan bagaimana sebuah elemen harus ditampilkan. CSS dapat digunakan untuk mengatur *style* (pewarnaan, tipe huruf, ukuran, dll), lebar spasi, serta tata letak elemen dari dokumen web.

CSS mengatur tampilan dari dokumen web menggunakan aturan (*rule*). *Rule* ini yang menentukan elemen dokumen web mana yang akan diatur visualisasinya. Pada setiap *rule*, kita dapat mengatur properti (seperti warna, ukuran, jenis huruf, dll) dari elemen yang dikenakan rule tersebut.

Tujuan utama dari CSS adalah agar isi dan tampilan dari sebuah dokumen dapat dipisahkan. Dengan pemisahan ini dapat meningkatkan performa pembacaan isi dari sebuah dokumen, menyediakan fleksibilitas dan pengelolaan pada spesifikasi dari karakteristik tampilan, memungkinkan beberapa dokumen HTML menggunakan gaya visualisasi yang sama (menggunakan file CSS yang sama), serta dapat mengurangi kompleksitas serta perulangan didalam dokumen HTML.

### 3.1 Sejarah CSS

CSS pertama kali diusulkan oleh Håkon Wium Lie pada 10 Oktober 1994. Pada saat itu Lie bekerja dengan Tim Berners-Lee di CERN (*Conseil Européen pour la Recherche Nucléaire* – sebuah organisasi penelitian bidang nuklir di Eropa). Pada kurun waktu yang sama, terdapat beberapa bahasa style sheet selain CSS yang juga diusulkan untuk digunakan pada dokumen web. Setelah melalui berbagai diskusi, baik di beberapa *mailing list* maupun didalam W3C (*World Wide Web Consortioun*), pada tahun 1996 dikeluarkan rekomendasi dari W3C untuk CSS (CSS1).

#### 3.1.1 CSS level 1

Spesifikasi resmi dari CSS level 1 dipublikasikan pada 17 Desember 1996, dimana Håkon Wium Lie dan Bert Bos dinobatkan sebagai pengembang aslinya. Berikut ini beberapa hal yang dapat diatur dengan CSS1:

- Properti dari huruf seperti bentuk huruf maupun penekanannya (huruf tebal, bergaris bawah, miring, dll).
- Warna teks, latar belakang, dan elemen-elemen lainnya.
- Atribut dari teks seperti lebar spasi antar kata ataupun spasi antar kalimat.
- Alignment dari teks, gambar, tabel, dan elemen-elemen lainnya.
- *Margin* (batasan), *border* (garis tepi), *padding*, dan pengaturan posisi untuk sebagian besar elemen.
- Identifikasi unik dan klasifikasi untuk sekelompok atribut.

#### 3.1.2 CSS level 2

Spesifikasi untuk CSS level 2 dikembangkan oleh W3C dan di publikasikan pada bulan Mei 1998. CSS level 2 merupakan pengembangan dari CSS1 dengan menambahkan beberapa kemampuan seperti pengaturan posisi (*absolute*, *relative*, atau *fixed*) dari elemen dan *z-index*, konsep untuk tipe-tipe media, mendukung

*aural style sheet* (yang kemudian digantikan dengan *speech modules* dari CSS3), dan beberapa properti baru dari huruf seperti shadows (bayangan).

### 3.1.3 CSS level 2 revisi 1

CSS level 2 revisi 1, atau seringkali disebut sebagai CSS2.1, memperbaiki beberapa kesalahan pada CSS2, menghapus beberapa fitur yang tidak banyak didukung oleh browser atau tidak mempunyai interoperable yang bagus, serta menambahkan beberapa ekstensi yang sudah diimplementasikan pada browser kedalam spesifikasi. Setelah melalui proses yang cukup panjang, CSS2.1 akhirnya dipublikasikan sebagai rekomendasi dari W3C pada 7 Juni 2011.

### 3.1.4 CSS level 3

Berbeda dengan CSS level 2 yang didefinisikan sebagai satu kesatuan yang berisi semua fitur yang ada, spesifikasi CSS level 3 dibagi menjadi beberapa dokumen yang disebut sebagai modul. Setiap modul dapat menambahkan kemampuan baru atau mengembangkan kemampuan dari CSS2, dengan tetap menjaga kompatibilitas terhadap versi sebelumnya. Konsep awal dari CSS3 pertama kali dipublikasikan pada Juni 1999.

Karena konsep modular pada CSS3, maka modul yang berbeda memiliki tingkat stabilitas dan status yang berbeda pula. Sampai pada Juni 2012, terdapat lebih dari 50 modul CSS yang telah dipublikasikan dari *CSS Working Group*. 4 diantaranya telah dipublikasikan sebagai *Formal Recommendation*, yaitu:

- 2012-06-19: *Media Queries*
- 2011-09-29: *Namespaces*
- 2012-09-29: *Selector Level 3*
- 2011-06-07: *Color*

Beberapa modul (termasuk modul *Backgrounds and Borders* dan modul *Multi-column Layout*) mempunyai status *Candidate Recommendation* (CR) dan dianggap

sudah cukup stabil. Tabel berikut ini merupakan status dari modul-modul dari CSS3.

**Tabel 3.1** Status dari modul-modul CSS3

Modul	Judul Spesifikasi	Status, Tanggal
css3-background	<a href="#">CSS Backgrounds and Borders Module Level 3</a>	<i>Candidate Recommendation</i> , September 2014
css3-box	<a href="#">CSS basic box model</a>	<i>Working Draft</i> , August 2007
css-cascade-3	<a href="#">CSS Cascading and Inheritance Level 3</a>	<i>Candidate Recommendation</i> , October 2013
css3-color	<a href="#">CSS Color Module Level 3</a>	<i>Recommendation</i> , June 2011
css3-content	<a href="#">CSS3 Generated and Replaced Content Module</a>	<i>Working Draft</i> , May 2003
css-fonts-3	<a href="#">CSS Fonts Module Level 3</a>	<i>Candidate Recommendation</i> , October 2013
css3-gcpm	<a href="#">CSS Generated Content for Paged Media Module</a>	<i>Working Draft</i> , May 2014
css3-layout	<a href="#">CSS Template Layout Module</a>	<i>Working Draft</i> , November 2011
css3-mediaqueries	<a href="#">Media Queries</a>	<i>Recommendation</i> , June 2012
css3-page	<a href="#">CSS Paged Media Module Level 3</a>	<i>Working Draft</i> , March 2013
css3-selectors	<a href="#">Selectors Level 3</a>	<i>Recommendation</i> , September 2011
css3-ui	<a href="#">CSS Basic User Interface Module Level 3 (CSS3 UI)</a>	<i>Working Draft</i> , January 2012

### 3.1.5 CSS level 4

Sama seperti CSS3 yang bersifat modular, tidak ada spesifikasi tunggal untuk CSS4. Dan karena CSS3 didefinisikan secara modular, maka setiap modul memiliki status yang berbeda serta dapat berkembang secara terpisah. Saat ini, sebagian besar modul berada pada level 3, yang dibangun berdasarkan CSS2.1.

Beberapa modul CSS level 4 yang dibangun berdasarkan modul CSS level 3 seperti *Image Values, Backgrounds & Borders*, dan *Selectors*. Terdapat juga beberapa modul yang mendefinisikan fungsi yang benar-benar baru, seperti *Flexbox*, dimana modul ini didesain sebagai “level 1”.

Karena saat ini CSS3 belum sepenuhnya selesai, maka CSS4 juga belum dapat digunakan secara utuh. Meskipun begitu, beberapa modul yang berada pada level 4 dapat disebut sebagai CSS4.

### 3.2 Style Rule

CSS merupakan kumpulan dari *rule* yang digunakan untuk mengatur bagaimana sebuah elemen pada dokumen web divisualisasikan. Berikut ini merupakan bentuk dasar dari *rule* pada CSS:

```
selector {  
  property1: value;  
  property2: value;  
  property3: value;  
}
```

Keterangan:

- selector merupakan identifikasi dari elemen dimana *rule* akan diterapkan. Selector dapat digunakan sebagai identifikasi untuk elemen sebenarnya (seperti elemen **p** atau elemen **body**), atau sebagai identifikasi untuk nilai dari atribut **class**.
- diantara tanda kurung kurawal buka '{' dan tanda kurung kurawal tutup '}' berisi beberapa pasangan *property/value* yang dipisahkan dengan tanda titik koma ';'. *Property* dan *value* dipisahkan dengan tanda titik dua ':'.
- *property* digunakan untuk mendefinisikan apa yang akan dilakukan terhadap elemen yang terpilih. Terdapat banyak sekali *property* yang dapat didefinisikan, seperti warna elemen, warna latar, posisi, *margin*, *padding*, jenis huruf, dan lain sebagainya.
- *value* merupakan nilai yang didefinisikan untuk setiap *property* dari elemen yang terpilih. Jenis nilai untuk satu properti bisa berbeda dibandingkan nilai untuk properti lainnya. Sebagai contoh, nilai untuk properti **color** bisa dalam nilai heksadesimal seperti #3399EE, nilai RGB seperti rgb(13,21134), atau

nama warna seperti *blue*, *red*, atau *green*. Nilai untuk properti **width** bisa berupa nilai dalam piksel, *ems*, prosentase, sentimeter, atau unit lainnya.

Sebagai contoh, lihat rule CSS berikut ini:

```
p {  
  margin: 5px;  
  font-family: arial;  
  color: blue;  
}
```

Elemen HTML yang dipilih oleh *rule* diatas adalah elemen **p**. Sehingga, untuk setiap elemen **p** pada dokumen HTML yang menggunakan CSS diatas akan terkena aturan pada *rule* tersebut. Yaitu, memiliki *margin* untuk paragraf sebesar 5 piksel, menggunakan huruf *Arial*, dan huruf yang digunakan berwarna biru.

### 3.3 Komentar dalam CSS

Sebagaimana bahasa yang lain, anda dapat menambahkan komentar didalam CSS. Anda dapat menambahkan komentar dengan mengapitnya diantara tanda `/*` dan `*/`. Komentar dalam CSS dapat menggunakan satu atau beberapa baris, dan browser akan mengabaikan baris-baris komentar tersebut. Lihat contoh berikut ini:

```
/* These are basic element selectors */  
selector {  
  property1: value;  
  property2: value;  
  property3: value;  
}
```

Pada contoh diatas, kalimat “These are basic element selectors” akan diabaikan oleh browser.

Anda dapat meletakkan komentar diantara rule satu dengan rule lainnya ataupun meletakkan komentar didalam blok untuk *property*. Perhatikan contoh CSS berikut ini:

```
selector {
  property1: value;
  /*
  property2: value;
  property3: value;
  */
}
```

Pada contoh diatas, maka properti kedua dan ketiga akan diabaikan oleh browser karena berada diantara tanda ‘/\*’ dan ‘\*/’.

Berbeda dengan bahasa lain, pada CSS anda hanya dapat menambahkan komentar dalam bentuk blok (komentar berada diantara tanda ‘/\*’ dan ‘\*/’). Anda dapat menuliskan komentar dalam satu baris, tetapi tetap harus menuliskannya diantara tanda ‘/\*’ dan ‘\*/’.

### 3.4 Mengelompokkan selector

Anda dapat mengelompokkan beberapa *selector* berbeda menjadi satu. Sebagai contoh, anda ingin mengatur visualisasi dokumen HTML menggunakan *rule* CSS seperti berikut ini:

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

Walaupun penulisan *rule* seperti diatas tidak salah dan dapat berfungsi sebagaimana mestinya, namun hal tersebut tidak ideal karena terdapat perulangan penulisan, dimana elemen **h1** dan **p** sama-sama memiliki properti *color* bernilai *red*. Untuk kondisi seperti diatas, anda dapat menyingkat penulisan *rule* CSS dengan mengelompokkan beberapa *selector* menggunakan tanda koma ‘,’.

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

Pada contoh diatas, aturan didalam tanda kurung kurawal akan diterapkan untuk kedua *selector* (**h1** dan **p**).

### 3.5 Tipe selector

Anda dapat menggunakan *selector* untuk berbagai tipe. Tiga tipe selector yang paling banyak digunakan adalah *element selector*, *class selector*, dan *ID selector*.

#### 3.5.1 Element Selector

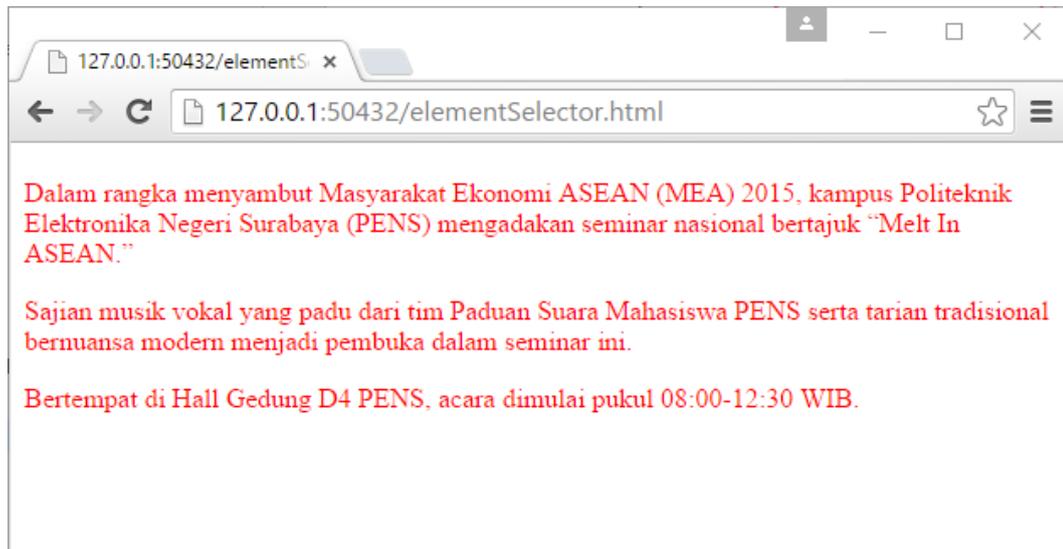
```
p {color: red;}
```

Aturan pada *rule* dengan *element selector* seperti contoh diatas akan diaplikasikan pada semua elemen **p** yang berada didalam dokumen HTML.

Perhatikan contoh dokumen HTML berikut ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
  p {color: red;}
</style>
</head>
<body>
  <p>Dalam rangka menyambut Masyarakat Ekonomi ASEAN (MEA)
2015, kampus Politeknik Elektronika Negeri Surabaya (PENS)
mengadakan seminar nasional bertajuk "Melt In ASEAN." </p>
  <p>Sajian musik vokal yang padu dari tim Paduan Suara
Mahasiswa PENS serta tarian tradisional bernuansa modern
menjadi pembuka dalam seminar ini. </p>
  <p>Bertempat di Hall Gedung D4 PENS, acara dimulai pukul
08:00-12:30 WIB.</p>
</body>
</html>
```

Dokumen HTML diatas menggunakan *rule CSS* berbentuk *embedded style* yang mengatur warna untuk elemen **p**. Didalam dokumen HTML tersebut terdapat 3 elemen **p**. Dokumen HTML diatas akan ditampilkan dalam browser seperti berikut ini:



Pada gambar diatas dapat anda lihat bahwa semua elemen **p** dalam dokumen HTML ditampilkan dengan warna merah, seperti yang diatur dalam *rule* CSS.

### 3.5.2 Class Selector

```
.baru {color: red;}
```

*Class selector* ditulis dengan menambahkan tanda ‘.’ (titik) sebelum nama *selector*. Aturan pada *rule* dengan *class selector* seperti contoh diatas akan diaplikasikan pada semua elemen dalam dokumen HTML yang memiliki atribut “class” dengan nilai “baru”. Perhatikan contoh dokumen HTML berikut ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
    .baru {color: red;}
</style>
</head>
<body>
    <h2 class="baru">Awal</h2>
    <p>Dalam rangka menyambut Masyarakat Ekonomi ASEAN (MEA)
    2015, kampus Politeknik Elektronika Negeri Surabaya (PENS)
    mengadakan seminar nasional bertajuk "Melt In ASEAN." </p>
```

```

<p class="baru">Sajian musik vokal yang padu dari tim
Paduan Suara Mahasiswa PENS serta tarian tradisional
bernuansa modern menjadi pembuka dalam seminar ini. </p>

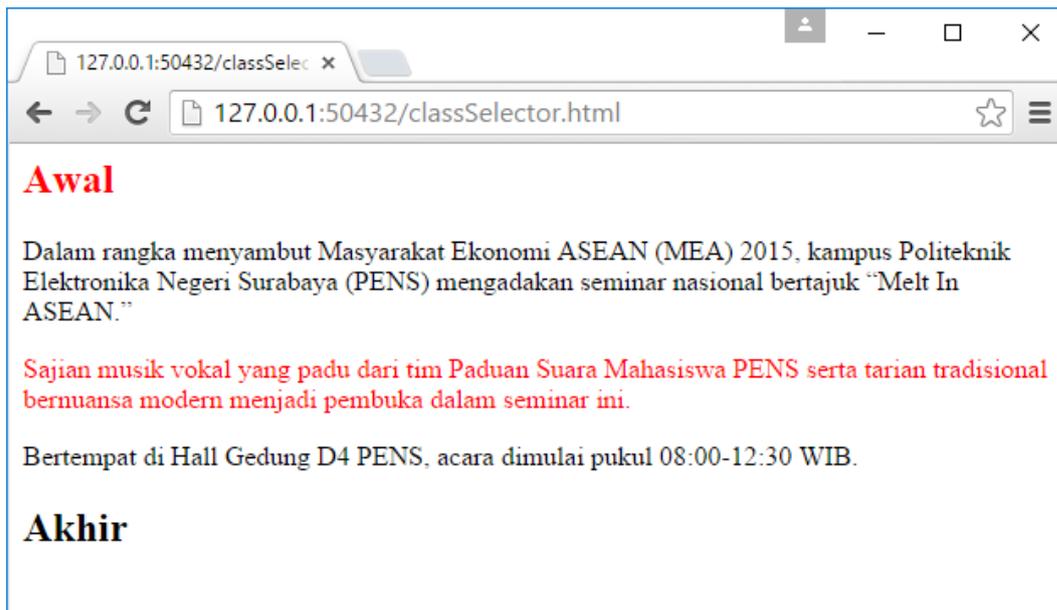
<p>Bertempat di Hall Gedung D4 PENS, acara dimulai pukul
08:00-12:30 WIB.</p>

<h2>Akhir</h2>

</body>
</html>

```

Dokumen HTML diatas menggunakan *rule CSS* yang mengatur warna untuk class 'baru'. Didalam dokumen HTML tersebut terdapat terdapat dua elemen **h2** dimana elemen **h2** yang pertama memiliki atribut 'class' dengan nilai 'baru'. Dalam dokumen tersebut juga terdapat tiga elemen **p** dimana salah satunya memiliki atribut 'class' yang bernilai 'baru' (elemen **p** yang kedua). Visualisasi dari dokumen HTML diatas pada browser adalah seperti berikut ini:



Perhatikan pada gambar diatas, terlihat bahwa hanya elemen yang memiliki atribut 'class' bernilai 'baru' saja yang terkena aturan dari *rule CSS* (ditampilkan dengan warna merah). Sedangkan elemen-elemen lainnya tetap berwarna hitam.

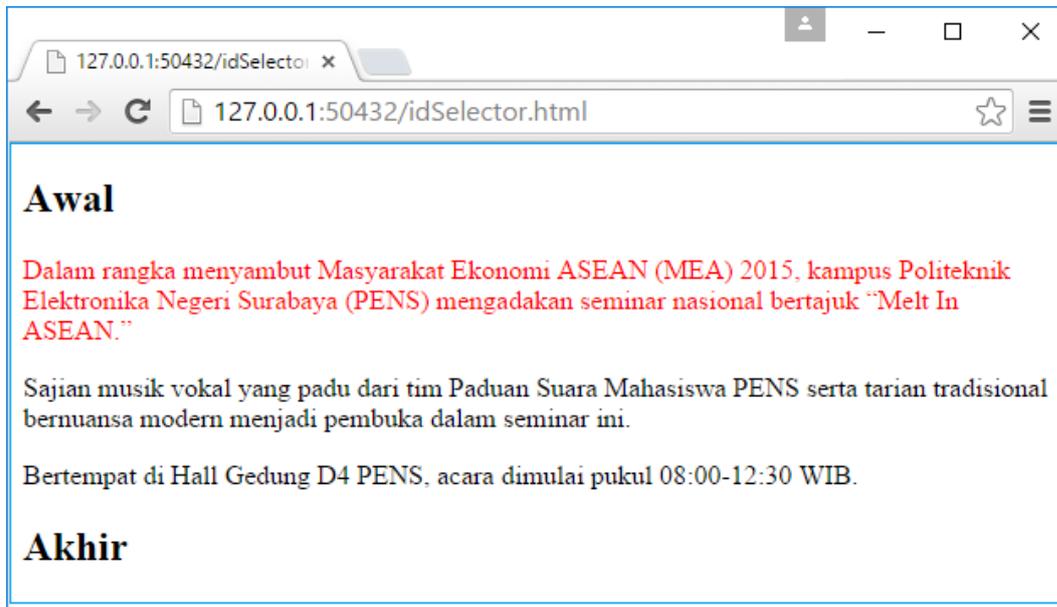
### 3.5.3 ID Selector

```
#ali01 {color: red;}
```

*ID selector* ditulis dengan menambahkan tanda '#' (pagar) sebelum nama *selector*. Aturan pada *rule* dengan *ID selector* seperti contoh diatas akan diaplikasikan pada elemen dalam dokumen HTML yang memiliki atribut "id" dengan nilai "ali01". Berbeda dengan atribut 'class' yang dapat bernilai sama untuk beberapa elemen, properti 'id' bersifat unik, sehingga tidak boleh ada dua atau lebih elemen dalam satu dokumen HTML yang memiliki atribut 'id' yang bernilai sama. Perhatikan contoh dokumen HTML berikut ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
    #ali01 {color: red;}
</style>
</head>
<body>
    <h2 class="baru">Awal</h2>
    <p id="ali01">Dalam rangka menyambut Masyarakat Ekonomi
    ASEAN (MEA) 2015, kampus Politeknik Elektronika Negeri
    Surabaya (PENS) mengadakan seminar nasional bertajuk "Melt In
    ASEAN." </p>
    <p id="ali02" class="baru">Sajian musik vokal yang padu
    dari tim Paduan Suara Mahasiswa PENS serta tarian tradisional
    bernuansa modern menjadi pembuka dalam seminar ini. </p>
    <p id="ali03">Bertempat di Hall Gedung D4 PENS, acara
    dimulai pukul 08:00-12:30 WIB.</p>
    <h2>Akhir</h2>
</body>
</html>
```

Dokumen HTML diatas menggunakan *rule* CSS yang mengatur warna untuk ID 'baru'. Didalam dokumen HTML tersebut terdapat terdapat tiga elemen **p** dimana salah satunya memiliki atribut 'id' yang bernilai 'baru' (elemen **p** yang pertama). Visualisasi dari dokumen HTML diatas pada browser adalah seperti berikut ini:



Perhatikan pada gambar diatas, terlihat bahwa hanya elemen yang memiliki atribut 'class' bernilai 'baru' saja yang terkena aturan dari *rule* CSS (ditampilkan dengan warna merah). Sedangkan elemen-elemen lainnya tetap berwarna hitam.

### 3.6 Mengkombinasikan Selector

Anda dapat menggabungkan beberapa *selector* untuk mendefinisikan aturan yang lebih spesifik. Perhatikan contoh-contoh berikut ini:

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

*Rule* diatas akan diaplikasikan pada elemen **p** yang memiliki atribut 'class' bernilai 'warning'.

```
h3#example {color: red;}
```

*Rule* diatas akan diaplikasikan pada elemen **h3** yang memiliki atribut 'id' bernilai 'example'.

```
p.info, h2.highlight {color: red;}
```

*Rule* diatas akan diaplikasikan pada elemen **p** yang memiliki atribut 'class' bernilai 'info' dan elemen **h2** yang memiliki atribut 'class' bernilai 'highlight'.

### 3.7 Selector Tingkat Lanjut

Selain tipe-tipe *selector* dasar yang sudah dijelaskan pada bagian sebelumnya, terdapat beberapa *selecteor* yang dapat anda gunakan untuk memilih elemen secara lebih detil lagi. Berikut ini daftar tipe *selector* tersebut:

- Universal selector
- Attribute selectors
- Child selectors
- Descendent selectors
- Adjacent sibling selectors
- Pseudo-classes
- Pseudo-elements

#### 3.7.1 Universal Selector

*Universal selector* dapat digunakan untuk memilih semua elemen didalam dokumen HTML. Perhatikan contoh *rule* berikut ini:

```
* {  
  border: 1px solid blue;  
}
```

*Rule* diatas akan mengakibatkan semua elemen dalam dokumen HTML yang menggunakan CSS tersebut memiliki garis tepi setebal 1px dengan tipe solid dan berwarna biru.

#### 3.7.2 Attribute Selector

*Attribute selector* dapat anda gunakan untuk memilih elemen berdasarkan atribut yang dimilikinya. Perhatikan contoh berikut ini:

```
img[alt] {  
  border: 3px solid red;  
}
```

*Rule* diatas akan mengakibatkan semua elemen **img** yang memiliki atribut 'alt' akan memiliki garis tepi setebal 3px dengan tipe solid dan berwarna merah.

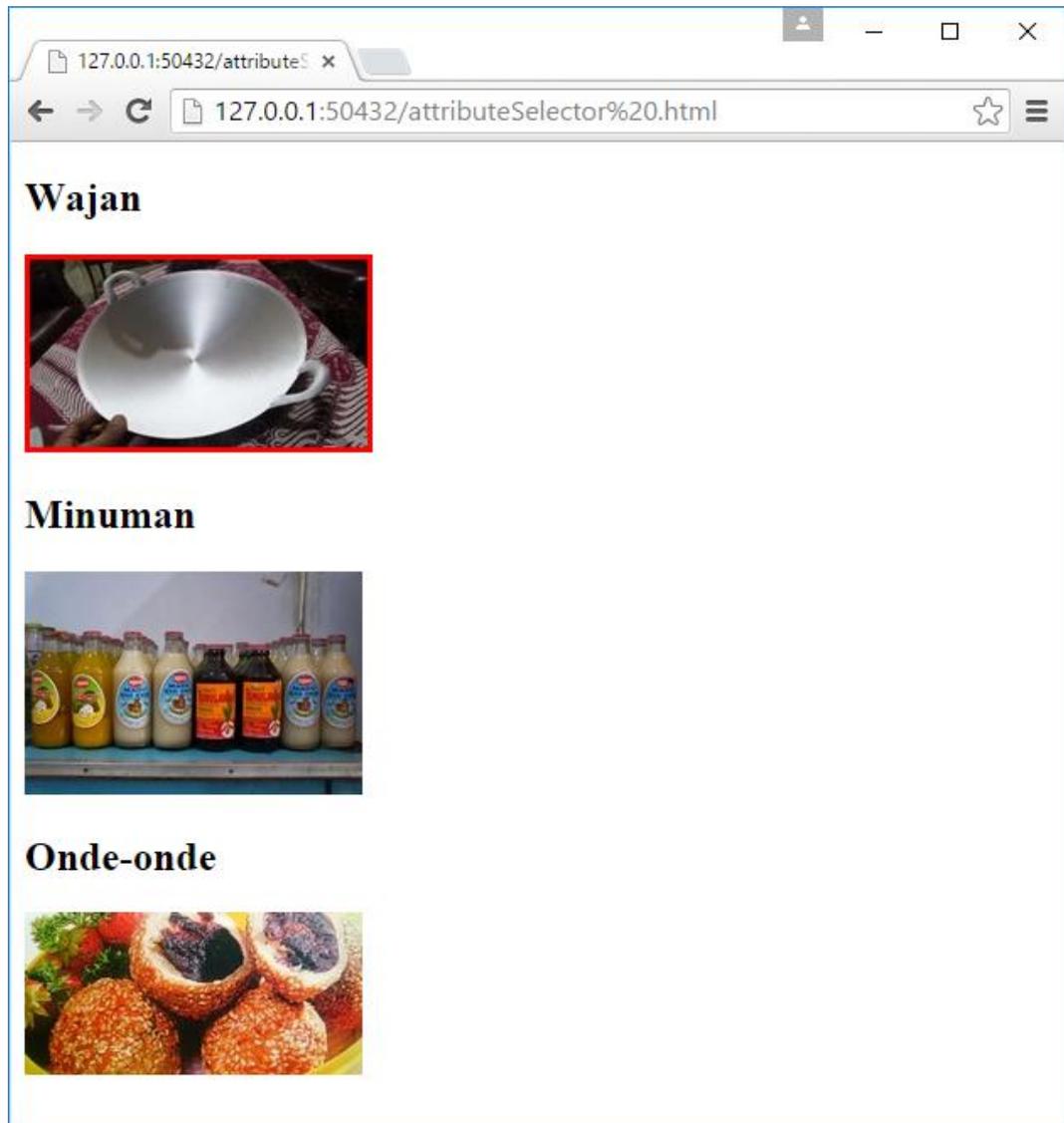
Anda juga dapat menggunakan *attribute selector* untuk melakukan pemilihan elemen secara lebih detail dengan cara memasukkan nilai dari atribut yang dipilih. Lihat contoh berikut ini:

```
img[src="wajan.jpg"] {  
  border: 3px solid red;  
}
```

*Rule* diatas hanya akan diaplikasikan pada elemen **img** yang mempunyai atribut 'src' bernilai 'wajan.jpg'. Perhatikan contoh penggunaan *attribute selector* pada dokumen HTML berikut ini:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
  img[src="wajan.jpg"] {border: 3px solid red;}  
</style>  
</head>  
<body>  
  <h2>Wajan</h2>  
    
  <h2>Minuman</h2>  
    
  <h2>Onde-onde</h2>  
    
</body>  
</html>
```

Visualisasi dokumen HTML diatas pada browser adalah seperti berikut ini:



Perhatikan pada gambar diatas, hanya gambar wajan saja yang memiliki garis tepi berwarna merah. Hal ini dikarenakan *rule* CSS didefinisikan hanya untuk elemen **img** yang memiliki atribut 'src' bernilai 'wajan.jpg' saja.

### 3.7.3 *Child Selector*

Anda dapat menggunakan *child selector* untuk memilih elemen tertentu yang merupakan 'anak' dari dari elemen lainnya. Perhatikan contoh *rule* berikut ini:

```
div > h2 {color: red;}
```

*Rule* diatas hanya akan diaplikasikan pada elemen **h2** yang meenjadi ‘anak’ (berada didalam) elemen **div** saja. Perhatikan contoh penggunaan *child selector* pada dokumen HTML berikut ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div > h2 {color: red;}
</style>
</head>
<body>
  <h2>Wajan Kuat</h2>
  <div>
    <h2>Minuman Segar</h2>
  </div>
  <h2>Onde-onde</h2>
</body>
</html>
```

Dokumen HTML diatas akan menghasilkan tampilan pada browser seperti berikut ini:



Pada gambar diatas, hanya kata “Minuman Segar” saja yang berwarna merah. Hal ini disebabkan karena *rule* didefinisikan hanya untuk elemen **h2** yang menjadi ‘anak’ dari elemen **div** saja. Kata “Minuman Segar” berada dalam elemen **h2** yang

menjadi ‘anak’ dari elemen **div**, sedangkan kata “Wajan Kuat” dan “Onde-onde” walaupun berada didalam elemen **h2** namun elemen **h2** tersebut bukan merupakan anak dari elemen **div**.

### 3.7.4 Descendant Selector

Penulisan *rule* dengan *descendant selector* adalah seperti berikut ini:

```
div em {text-decoration: underline;}
```

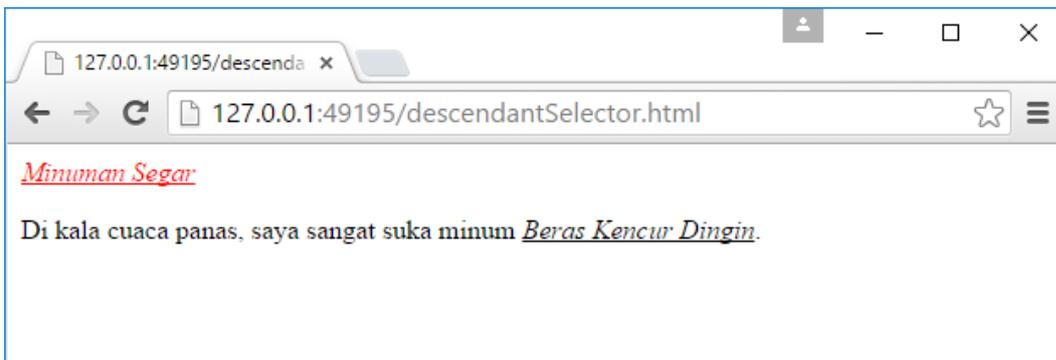
*Rule* diatas hanya akan diaplikasikan pada elemen **em** yang meenjadi turunan dari (berada didalam) elemen **div** saja.

*Descendant selector* mirip dengan *child selector*. Bedanya, jika *child selector* hanya digunakan untuk memilih elemen yang merupakan turunan (anak) langsung dari elemen lainnya, *descendant selector* dapat digunakan untuk memilih semua elemen yang menjadi turunan elemen lainnya, baik secara langsung maupun tidak langsung (tidak memperhitungkan seberapa dalam level hirarkinya). Untuk melihat perbedaan antara *child selector* dan *descendent selector*, perhatikan dokumen HTML berikut ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div > em {color: red;}
  div em {text-decoration: underline;}
</style>
</head>
<body>
  <div>
    <em>Minuman Segar</em>
    <p>Di kala cuaca panas, saya sangat suka minum
    <em>Beras Kencur Dingin</em>.</p>
  </div>
</body>
</html>
```

Pada dokumen HTML diatas terdapat dua *rule*. *Rule* yang pertama menggunakan *child selector* (“div > em {color: red;}”) sedangkan yang kedua menggunakan *descendant selector* (“div em {text-decoration: underline;}”). *Rule* pertama berfungsi untuk merubah warna teks menjadi merah sedangkan *rule* kedua berfungsi untuk merubah teks menjadi bergaris bawah.

Dokumen HTML tersebut akan ditampilkan dalam browser seperti berikut ini:



Perhatikan gambar tampilan browser diatas, aturan dari *rule* pertama (merubah warna teks menjadi merah) hanya dikenakan pada kata “Minuman Segar” saja, sedangkan *rule* kedua (memberi garis bawah pada teks) dikenakan baik pada kata “Minuman Segar” maupun kata “Beras Kencur Dingin”. Kata “Beras Kencur Dingin” tidak dikenakan aturan dari *rule* pertama karena walaupun kata ini berada didalam elemen **em** namun bukan merupakan turunan langsung dari elemen **div**.

### 3.7.5 Adjacent Sibling Selector

Penulisan *rule* dengan adjacent sibling selector adalah seperti berikut ini:

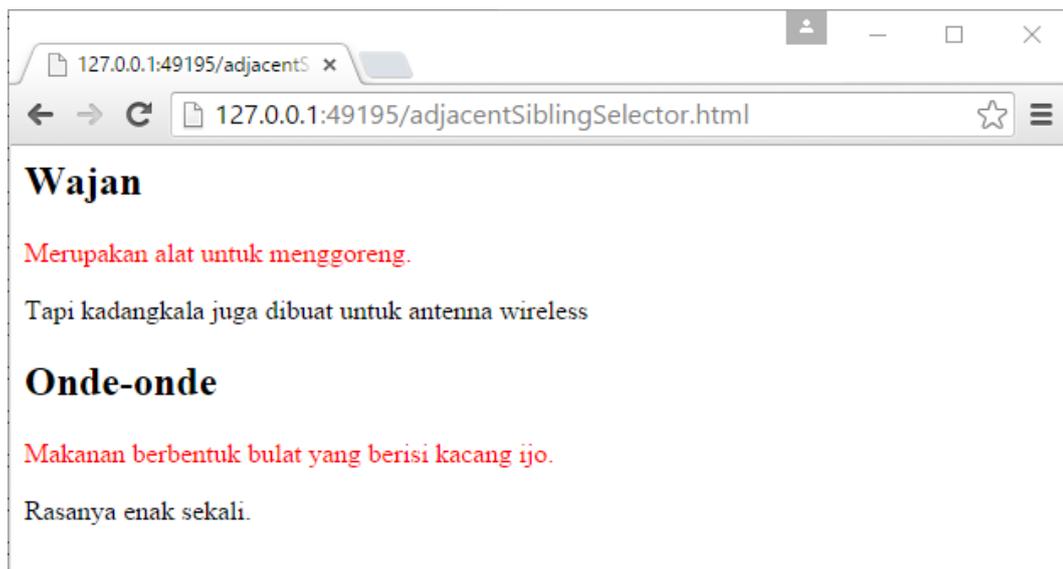
```
h2 + p {color: red}
```

*Rule* diatas akan diaplikasikan pada elemen **p** yang berada setelah (bersebelahan dengan) elemen **h2** pada level hirarki yang sama. Perhatikan contoh dokumen HTML berikut ini:

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

```
h2 + p {color: red}
</style>
</head>
<body>
  <h2>Wajan</h2>
  <p>Merupakan alat untuk menggoreng.</p>
  <p>Tapi kadangkala juga dibuat untuk antenna wireless</p>
  <h2>Onde-onde</h2>
  <p>Makanan berbentuk bulat yang berisi kacang ijo.</p>
  <p>Rasanya enak sekali.</p>
</body>
</html>
```

Dokumen HTML diatas memiliki satu *rule* yang akan merubah warna teks pada elemen **p** yang terletak tepat setelah elemen **h2** menjadi merah. Dokumen ini akan ditampilkan pada browser seperti gambar berikut ini:



Perhatikan gambar diatas, hanya elemen **p** yang berada tepat setelah elemen **h2** saja yang dikenakan aturan dari *rule* (teks berwarna merah).

### 3.7.6 *Pseudo-class*

*Pseudo-class* tidak digunakan untuk memilih elemen, namun untuk memilih status dari sebuah elemen. Penggunaan yang paling umum adalah untuk pewarnaan dari sebuah *hyperlink*. Berikut ini adalah beberapa status dari *hyperlink*:

- **:link** – normal, status awal dari sebuah *link* ketika pertama kali tampil.
- **:visited** – *link* yang sudah pernah anda kunjungi (sudah pernah di-klik).
- **:focus** – *link* (atau kotak isian/form, atau yang lain) dimana kursor *keyboard* sedang berada didalamnya.
- **:hover** – *link* (atau elemen lainnya) dimana pointer mouse sedang berada diatasnya.
- **:active** – *link* yang sedang di-klik.

Berikut ini contoh penggunaan *pseudo-class* pada *selector* untuk penulisan *rule*:

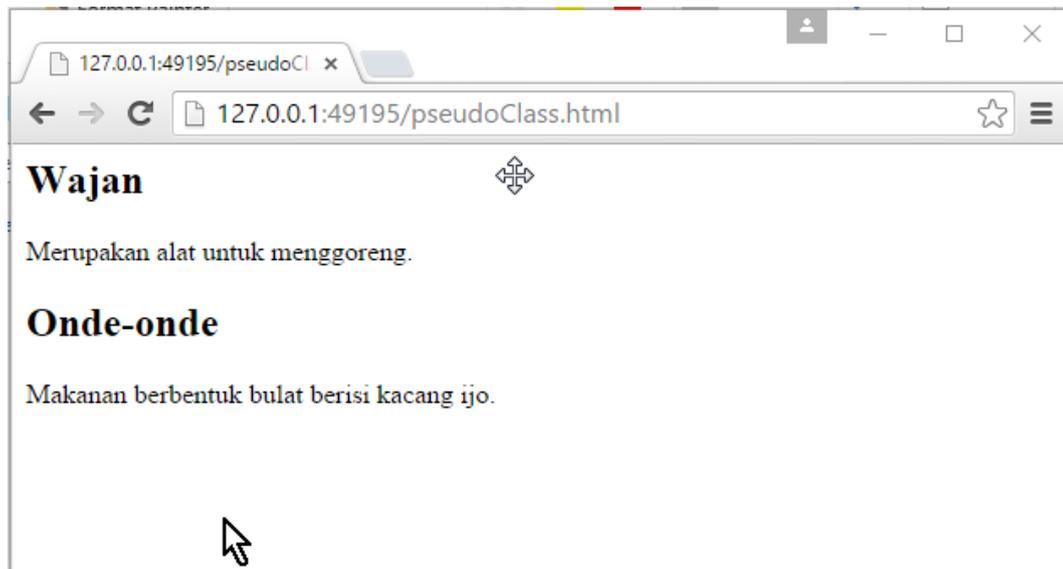
```
a:link {color: blue;}
a:visited {color: gray;}
a:hover, a:focus {text-decoration: none;}
a:active {font-weight: bold;}
```

Perhatikan dokumen HTML berikut ini:

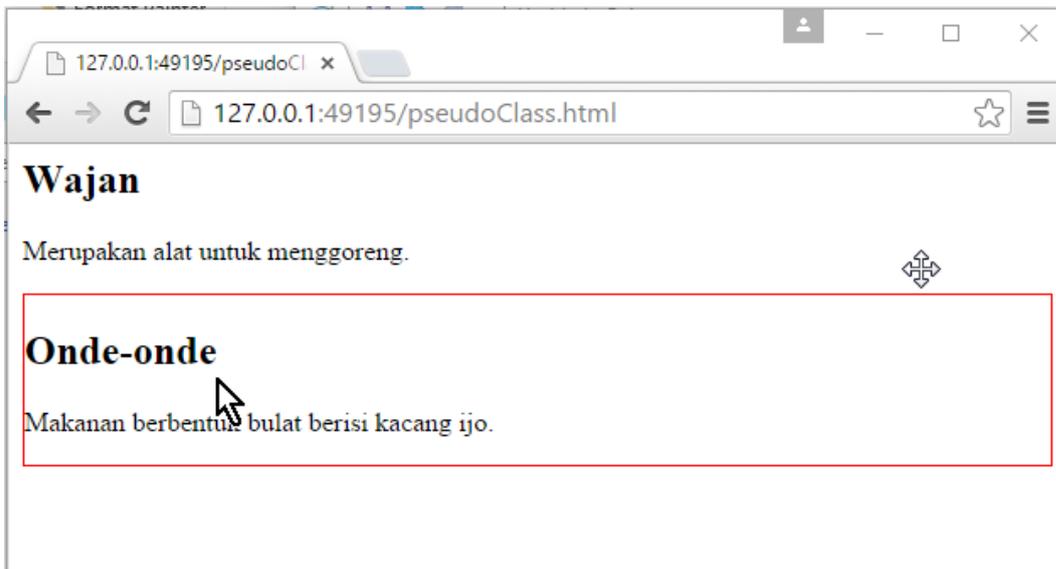
```
<!DOCTYPE html>
<html>
<head>
<style>
  div:hover {border: 1px solid red}
</style>
</head>
<body>
  <div>
    <h2>Wajan</h2>
    <p>Merupakan alat untuk menggoreng.</p>
```

```
</div>
<div>
  <h2>Onde-onde</h2>
  <p>Makanan berbentuk bulat berisi kacang ijo.</p>
</div>
</body>
</html>
```

Dokumen HTML diatas memiliki sebuah *rule* yang akan membuat elemen **div** memiliki garis tepi (*border*) berwarna merah ketika pointer mouse diletakkan diatas elemen tersebut. Perhatikan keluaran dari dokumen HTML pada browser berikut ini:



Gambar diatas memperlihatkan tampilan pada browser ketika pointer mouse tidak berada diatas elemen **div**.



Gambar diatas memperlihatkan tampilan pada browser ketika pointer mouse berada diatas elemen **div**. Perhatikan bahwa elemen **div** yang kedua memiliki garis tepi berwarna merah.

Selain yang telah disebutkan diatas, berikut ini adalah daftar beberapa pseudo-class yang dapat digunakan pada CSS3:

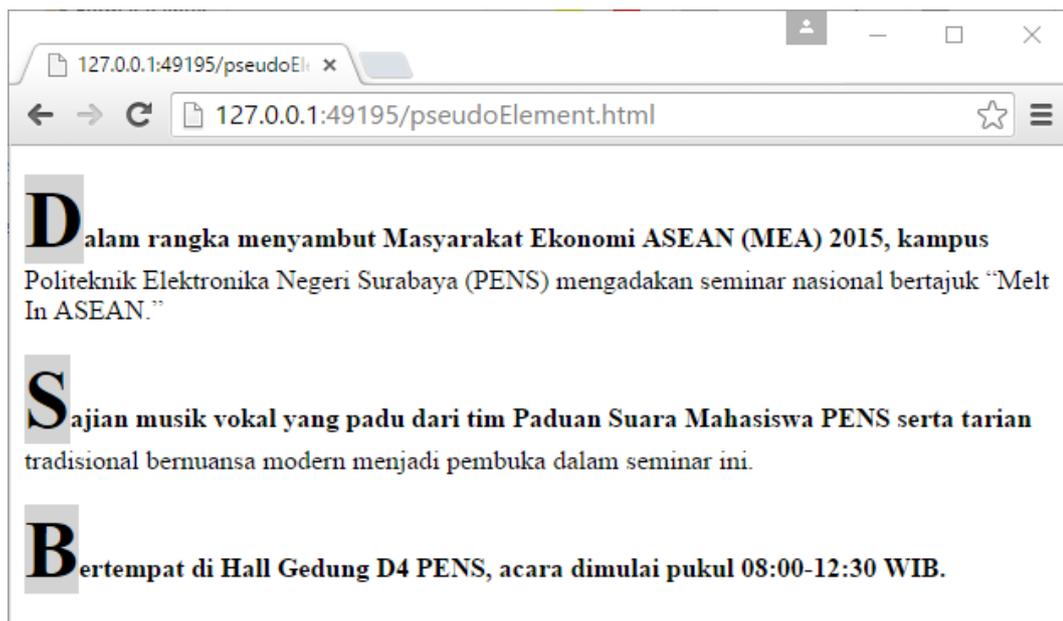
- :checked
- :disabled
- :empty
- :first-child
- :invalid
- dan lain-lain

### 3.7.7 *Pseudo-element*

Pseudo-element dapat digunakan dalam dua hal. Yang pertama, anda dapat menggunakan pseudo-element untuk memilih huruf pertama atau baris pertama dari teks yang berada dalam sebuah elemen HTML. Perhatikan contoh dokumen HTML berikut ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
  p::first-letter {
    font-size: 300%;
    background-color: lightgrey;
  }
  p::first-line {font-weight: bold;}
</style>
</head>
<body>
  <p>Dalam rangka menyambut Masyarakat Ekonomi ASEAN (MEA)
  2015, kampus Politeknik Elektronika Negeri Surabaya (PENS)
  mengadakan seminar nasional bertajuk "Melt In ASEAN." </p>
  <p>Sajian musik vokal yang padu dari tim Paduan Suara
  Mahasiswa PENS serta tarian tradisional bernuansa modern
  menjadi pembuka dalam seminar ini. </p>
  <p>Bertempat di Hall Gedung D4 PENS, acara dimulai pukul
  08:00-12:30 WIB.</p>
</body>
</html>
```

Dokumen HTML diatas akan tampil di browser seperti gambar berikut ini:

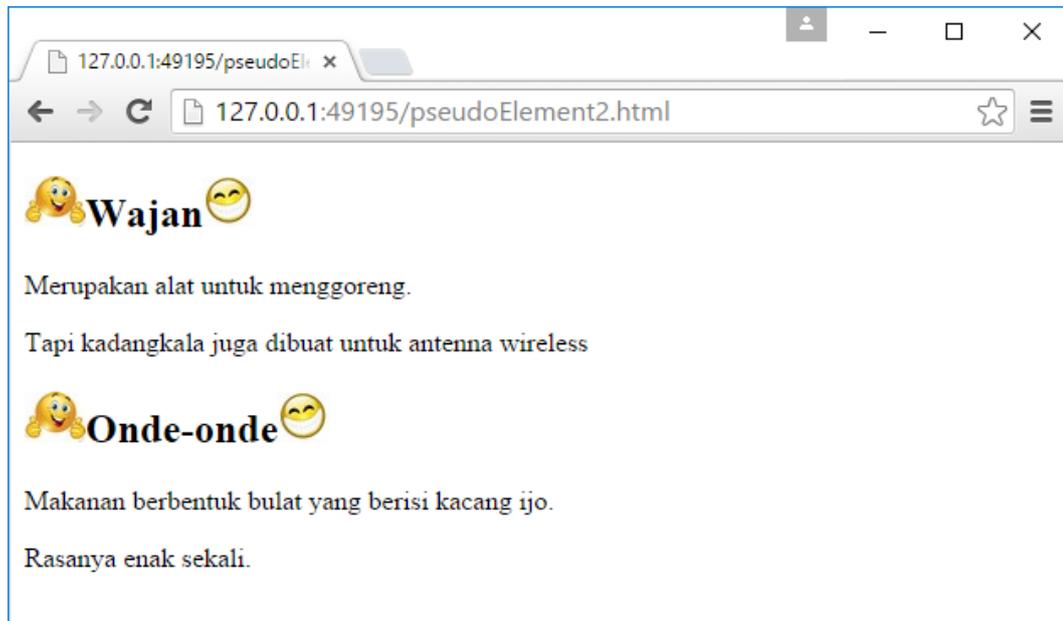


Perhatikan gambar diatas, untuk setiap huruf pertama dari elemen **p** akan ditulis dengan ukuran 300% dan memiliki warna latar abu-abu. Dan setiap baris pertama dari elemen **p** ditulis dengan huruf tebal.

Selain itu, anda dapat pula menggunakan pseudo-element untuk menambahkan isi pada dokumen HTML. Perhatikan contoh dokumen HTML berikut ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
  h2::before {content: url(smile.jpg)}
  h2::after {content: url(laugh.jpg)}
</style>
</head>
<body>
  <h2>Wajan</h2>
  <p>Merupakan alat untuk menggoreng.</p>
  <p>Tapi kadangkala juga dibuat untuk antenna wireless</p>
  <h2>Onde-onde</h2>
  <p>Makanan berbentuk bulat yang berisi kacang ijo.</p>
  <p>Rasanya enak sekali.</p>
</body>
</html>
```

Dokumen HTML diatas akan menghasilkan keluaran di browser seperti berikut ini:



Dapat anda lihat pada keluaran dari dokumen HTML diatas, *rule* CSS pada dokumen tersebut akan menambahkan gambar ‘smile.jpg’ sebelum (*before*) elemen **h2** dan menambahkan gambar ‘laugh.jpg’ sesudahnya (*after*).

### 3.8 CSS Shorthand

Salah satu hal yang akan sering anda temui dalam penggunaan CSS adalah *Shorthand*. *Shorthand* dapat diartikan sebagai penggabungan beberapa properti kedalam satu properti untuk mengurangi tempat penyimpanan dan waktu penulisan.

Pada contoh-contoh sebelumnya, sebenarnya beberapa *rule* CSS sudah menggunakan *shorthand*. Perhatikan *rule* berikut ini:

```
div {
  border: 1px solid blue;
}
```

Properti *border* pada *rule* diatas sebenarnya adalah penggabungan dari properti *border-width*, *border-style*, dan *border-color*. Sehingga *rule* diatas akan memiliki efek yang sama dengan *rule* berikut ini:

```
div {
  border-width: 1px;
```

```
border-style: solid;
border-color: blue;
}
```

Perhatikan *rule* berikut ini:

```
div {
  margin-top: 1px;
  margin-right: 2px;
  margin-bottom: 3px;
  margin-left: 4px;
}
```

*Rule* diatas dapat juga anda tuliskan menjadi seperti berikut ini:

```
div {
  margin: 1px 2px 3px 4px;
}
```

Beberapa *shorthand property* (seperti *margin*, *border-width*, atau *padding*) memerlukan 4 nilai. Namun anda mempunyai pilihan untuk memasukkan 1 nilai, 2 nilai, 3 nilai, ataupun 4 nilai sekaligus. Perhatikan penggunaan property *border-width* (lebar garis tepi) pada dokumen HTML berikut ini:

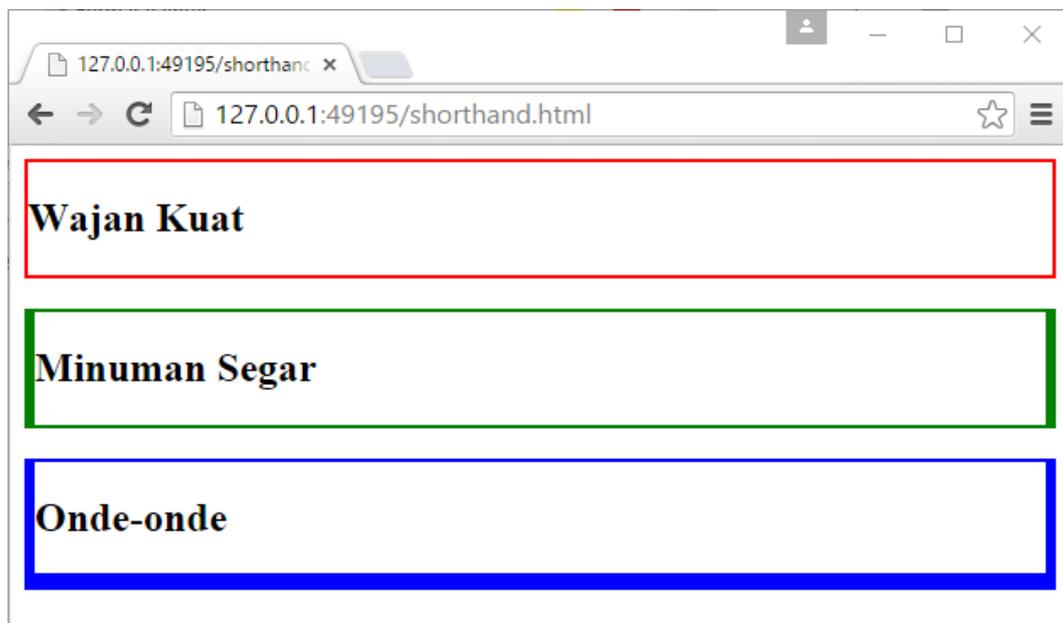
```
<!DOCTYPE html>
<html>
<head>
<style>
  .satu {border-width: 2px; border-color: red;}
  .dua {border-width: 2px 6px; border-color: green;}
  .tiga {border-width: 2px 6px 10px; border-color: blue;}
  div {border-style: solid;}
</style>
</head>
<body>
  <div class="satu">
    <h2>Wajan Kuat</h2>
```

```

</div>
<br>
<div class="dua">
  <h2>Minuman Segar</h2>
</div>
<br>
<div class="tiga">
  <h2>Onde-onde</h2>
</div>
</body>
</html>

```

Pada dokumen HTML tersebut terdapat 4 *rule* CSS: 3 *rule* masing-masing untuk class ‘satu’, ‘dua’, dan ‘tiga’; serta 1 *rule* yang berlaku untuk elemen *div*. Pada dokumen tersebut terdapat 3 elemen **div** dengan atribut ‘class’ yang bernilai ‘satu’, ‘dua’, dan ‘tiga’. Pada browser, dokumen tersebut akan ditampilkan seperti berikut ini:



Perhatikan gambar diatas, elemen **div** pertama dikenakan aturan dari *rule* pertama yang hanya memiliki 1 nilai untuk border-width (“border-width: 2px”). Jika sebuah *shorthand property border-width* berisi satu nilai, maka nilai ketebalan

garis tepi tersebut akan diterapkan pada semua sisi atas/top, kanan/right, bawah/bottom, dan kiri/left.

Elemen **div** kedua dikenakan aturan dari *rule* kedua yang memiliki 2 nilai untuk border-width (“border-width: 2px 6px”). Jika sebuah *shorthand property border-width* berisi dua nilai, maka nilai pertama akan diterapkan pada sisi atas/top dan bawah/bottom, sedangkan nilai kedua akan diterapkan pada sisi kanan/right dan kiri/left.

Elemen **div** ketiga dikenakan aturan dari *rule* ketiga yang memiliki 3 nilai untuk border-width (“border-width: 2px 6px 10px”). Jika sebuah *shorthand property border-width* berisi tiga nilai, maka nilai pertama akan diterapkan pada sisi atas/top, nilai kedua akan diterapkan pada sisi kanan/right dan kiri/left, sedangkan nilai ketiga akan diterapkan pada sisi bawah/bottom.

### 3.9 Mengaplikasikan CSS kedalam dokumen HTML

Terdapat tiga cara untuk mengaplikasikan CSS ke dalam dokumen HTML: *inline style*, *embedded style*, dan *external style*. Jika anda tidak memiliki alasan yang sangat kuat, jangan gunakan cara pertama dan kedua. Selalu gunakan cara ketiga, *external style*, untuk mengaplikasikan CSS kedalam dokumen HTML.

#### 3.9.1 Inline Style

Anda dapat mengaplikasikan *CSS style* pada elemen dalam dokumen HTML menggunakan atribut ‘style’, seperti pada contoh berikut ini:

```
<p style="background:blue; color:yellow; padding:5px;">
  Lorem Ipsum is simply dummy text of the printing and
  typesetting industry.
</p>
```

Didalam atribut ‘style’, anda dapat menuliskan semua properti CSS beserta nilainya (*property/value pair*). Antara pasangan properti/nilai satu dengan pasangan lainnya dipisahkan dengan menggunakan tanda titik koma ‘;’.

### 3.9.2 *Embedded Style*

Dengan *embedded style*, *rule* CSS ditempatkan pada bagian `<head>` dari dokumen HTML didalam elemen *style*, seperti pada contoh berikut ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
  p (color: blue;)
  h1 (font-weight: bold )
  div > h2 {color: red;}
</style>
</head>
<body>
  . . . . .
  . . . . .
  . . . . .
</body>
</html>
```

Hampir semua contoh penggunaan CSS pada bab ini menggunakan *embedded style*. Dengan cara ini anda tidak perlu menuliskan atribut *style* untuk setiap elemen dalam dokumen HTML anda.

### 3.9.3 *External Style*

*External style* merupakan cara penggunaan CSS yang direkomendasikan. Dengan cara ini, rule CSS ditulis didalam sebuah file terpisah yang berekstensi '.css'. Kemudian, file CSS tersebut diaplikasikan ke dalam dokumen HTML dengan menggunakan elemen *link*, seperti contoh berikut ini:

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

Dengan cara ini memungkinkan anda untuk membuat aturan-aturan terkait visualisasi dalam sebuah file CSS, kemudian menggunakan file tersebut untuk semua dokumen HTML dalam sebuah website. Anda dapat merubah tampilan dari keseluruhan dokumen hanya dengan merubah satu file saja. Cara ini juga akan

mempercepat waktu akses, karena browser hanya perlu mengunduh satu file CSS saja untuk dapat digunakan oleh keseluruhan dokumen.

Contoh berikut ini adalah isi dari sebuah file CSS dengan nama style.css:

```
.satu {border-width: 2px; border-color: red;}  
.dua {border-width: 2px 6px; border-color: green;}  
.tiga {border-width: 2px 6px 10px; border-color: blue;}  
div {border-style: solid;}
```

Dan berikut ini adalah contoh dokumen HTML yang menggunakan file CSS style.css:

```
<!DOCTYPE html>  
<html>  
<head>  
  <link rel="stylesheet" href="style.css" />  
</head>  
<body>  
  <div class="satu">  
    <h2>Wajan Kuat</h2>  
  </div>  
  <br>  
  <div class="dua">  
    <h2>Minuman Segar</h2>  
  </div>  
  <br>  
  <div class="tiga">  
    <h2>Onde-onde</h2>  
  </div>  
</body>  
</html>
```

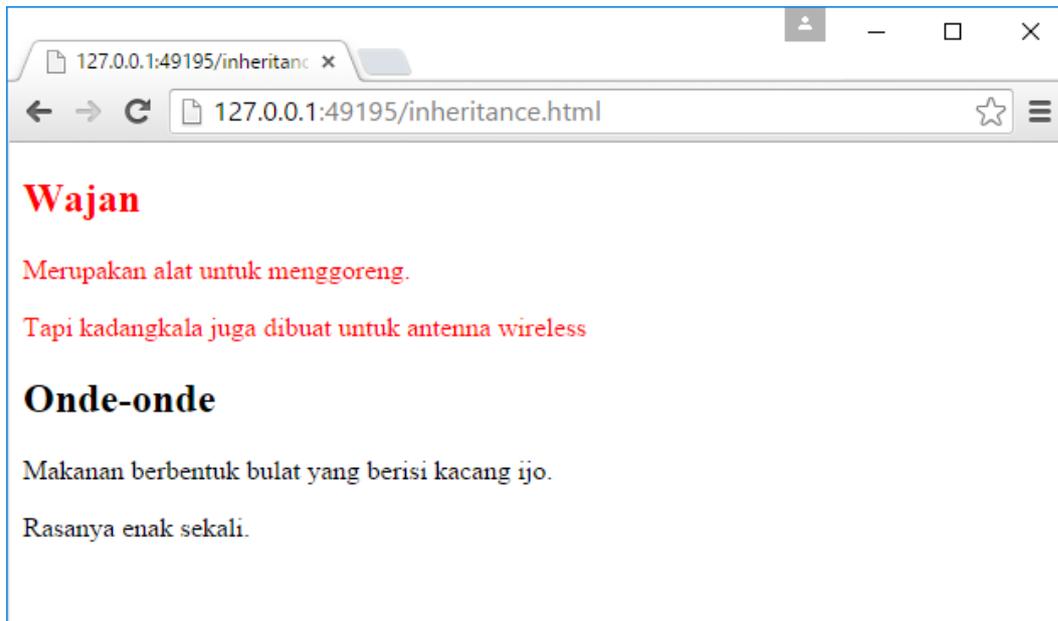
Contoh ini akan menghasilkan keluaran yang sama persis seperti contoh dokumen HTML pada sub bab *CSS Shorthand*.

### 3.10 Inheritance

Didalam CSS, *inheritance* (warisan) merupakan suatu mekanisme untuk mewariskan properti dari elemen induk ke elemen lain yang menjadi anak/turunannya. Hal ini mirip dengan *inheritance* di dunia genetika, seperti, jika orang tua memiliki mata berwarna biru maka anaknya juga akan memiliki mata berwarna biru juga. Perhatikan contoh dokumen HTML berikut ini terkait dengan *inheritance*:

```
<!DOCTYPE html>
<html>
<head>
<style>
    .satu {color: red;}
</style>
</head>
<body>
    <div class="satu">
        <h2>Wajan</h2>
        <div>
            <p>Merupakan alat untuk menggoreng.</p>
            <p>Tapi kadangkala juga dibuat untuk antenna
wireless</p>
        </div>
    </div>
    <div class="dua">
        <h2>Onde-onde</h2>
        <p>Makanan berbentuk bulat yang berisi kacang
ijo.</p>
        <p>Rasanya enak sekali.</p>
    </div>
</body>
</html>
```

Terdapat satu *rule* pada dokumen HTML diatas yang mengatur warna teks untuk elemen **div** dengan atribut *class* yang bernilai 'satu'. Berikut ini visualisasi dari dokumen diatas pada browser:



Perhatikan gambar diatas, *rule* yang mengatur warna teks menjadi merah tidak diterapkan pada elemen **div** dengan atribut *class* bernilai 'satu' saja, melainkan juga diterapkan pada semua elemen yang berada didalam elemen **div** dengan atribut *class* bernilai 'satu' tersebut.

Tidak semua properti di CSS dapat diwariskan, karena ada beberapa properti yang memang tidak perlu untuk diwariskan. Sebagai contoh, properti *margin* tidak diwariskan, karena jarang sekali elemen anak menggunakan *margin* yang sama dengan induknya. Anda dapat mencari daftar lengkap properti apa saja yang dapat diwariskan maupun tidak dari website World Wide Web Consortium (W3C) di alamat [www.w3c.org](http://www.w3c.org).

### 3.11 Cascade

CSS merupakan singkatan dari *Cascading Style Sheet*, sehingga sudah lumrah jika *cascade* ini menjadi hal yang penting disini. Konsep *cascade* pada CSS adalah mekanisme yang mengatur hasil akhir dari visualisasi jika terdapat beberapa deklarasi CSS yang mengatur elemen yang sama. Terdapat tiga konsep utama yang akan mengatur urutan dari beberapa deklarasi CSS yang akan diaplikasikan:

1. Importance --> tingkat kepentingan

2. Specificity --> seberapa spesifik sebuah *selector* dari *rule*
3. Source order --> urutan kemunculan

Jika sebuah elemen pada dokumen HTML dikenakan dua *rule*, maka *rule* dengan *importance* paling tinggi yang akan diterapkan. Jika nilai *importance* sama, maka akan dilihat level *specificity*-nya, dan jika level *specificity*-nya sama akan dilihat urutan kemunculannya.