**CSS**

**CSS comments :** digunakan untuk menjelaskan kode dan dapat membantu ketika Anda mengedit kode sumber di lain waktu. Apa yang tertulis akan diabaikan oleh browser dan tidak ditampilkan di browser, sehingga dapat menjadi opsi baik untuk memberi catatan atau informasi dokumentasi pada kode.

**Embedded Style Sheet-**> Embedded Style Sheet merupakan kumpulan rules yang dituliskan dalam berkas HTML dengan menggunakan elemen <style>. Dengan begitu rules yang dituliskan hanya dapat dicakup oleh satu berkas HTML.

**Inline Style ->** merupakan styling yang diterapkan pada elemen HTML dengan menggunakan atribut style.

Untuk menambahkan styling properties lainnya (multiple properties), kita tuliskan dengan menggunakan semicolon (;) sebagai pemisah antar styling properties-nya.

**Inheritance**

Styling HTML bersifat inheritance yang artinya dapat mewarisi properti style “tertentu” pada elemen yang ada di dalamnya. Contohnya pada rules yang kita tuliskan untuk elemen <body> akan diterapkan pada seluruh elemen yang ada di dalam elemennya (body). Contoh lainnya, pada rules yang diterapkan pada elemen <footer> dengan properti color yang bernilai white, akan diterapkan pada seluruh elemen yang ada di dalam <footer>. Hal ini menjadi alasan mengapa memahami struktur dokumen itu penting.

### Rule Order

Sesuai dengan namanya, cascading artinya “mengalir.” Demikian halnya dengan alur kerja CSS dalam membaca kode yang mengalir dari atas ke bawah. Karena itu kita harus memperhatikan urutan dalam penulisan rules, terutama saat terjadi sebuah konflik.

Konflik dapat terjadi karena kita dapat menerapkan beberapa styling pada satu dokumen HTML. Contohnya, apa yang seharusnya ditampilkan pada browser ketika eksternal css mengharuskan elemen <p> menampilkan warna merah, tetapi pada embedded css <p> harus menampilkan warna biru? Kembali pada alur kerja CSS yang membaca dari atas ke bawah, sehingga warna yang akan diterapkan adalah warna yang paling akhir didefinisikan.

Kita bisa menambahkan keyword !important diakhir nilai propertinya.

Gunakan !important ketika memang benar-benar dibutuhkan saja. Sebaiknya kita pahami aturan urutan pada CSS dengan baik sehingga meminimalisir penggunaan tanda tersebut.

Berikut catatan yang sudah kita pelajari sejauh ini tentang *styling*:

* *Rule* : Sebuah aturan styling yang harus diterapkan pada elemen HTML. Dalam sebuah *rule* terdapat *selector* dan deklarasi properti *styling*.
* *Selector* : Sebuah bagian dari rule, yang mengidentifikasi target elemen untuk menetapkan sebuah rule.
* *Declaration* : Sebuah bagian dari rule, yang terdiri dari pasangan properti dan nilainya.
* *External Style Sheet* : Berkas terpisah yang di dalamnya hanya terdapat sebuah *rules* yang akan digunakan pada website.
* *Embedded Style Sheet* : Kumpulan rules yang dituliskan dalam berkas HTML dengan menggunakan elemen <style>.
* *Inline Style* : styling yang diterapkan pada elemen HTML dengan menggunakan atribut *style*.

**CSS SELECTOR**

### Selector Basic

Pada sub-modul sebelumnya kita sudah mengetahui apa itu selector, dan menggunakannya. Yang kita gunakan tersebut merupakan selector basic. Sebenarnya terdapat beberapa macam dari selector basic, yakni:

* Type Selector
* Class Selector
* ID Selector
* Attribute Selector
* Universal Selector

### Type Selector

Type Selector menggunakan nama elemen sebagai target untuk diterapkannya rule. Dengan kata lain, ketika menggunakan selector ini tentu rule akan diterapkan pada seluruh elemen target yang ada pada dokumen HTML.

#### Class Selector

Class selector menetapkan target elemen berdasarkan nilai dari atribut class yang diterapkan pada elemennya. Untuk penulisan selector, awali dengan tanda titik (.) kemudian lanjutkan dengan nama class-nya. Class bersifat shareable, sehingga dapat diterapkan pada banyak elemen dengan tipe elemen yang berbeda-beda. Misalkan sebuah class red dapat diterapkan pada elemen paragraf dan juga heading untuk menampilkan teks berwarna merah.

sebuah elemen juga mungkin memiliki banyak nilai class, sehingga kita dapat menerapkan lebih dari satu rule atau gabungan rule pada elemen target. Untuk menggunakannya, pada atribut class kita cukup tuliskan nama kelasnya dan pisahkan tiap nilai kelasnya dengan spasi.

Kita juga bisa menargetkan elemen secara spesifik yang memiliki sebuah class. Bagaimana jika kita ingin menerapkan gaya yang berbeda ketika menggunakan kelas bergaya keren alias fancy pada sebuah paragraf? untuk melakukannya, pada selector kita tuliskan nama elemen target diikuti dengan titik (.) kemudian nama kelasnya

### ID Selector

ID selector menetapkan target elemen berdasarkan nilai dari atribut id yang diterapkan pada elemennya. Sama seperti class, atribut id dapat diterapkan pada seluruh elemen HTML, dan kebanyakan atribut ini digunakan untuk memberikan sebuah arti pada generic element seperti <div> dan <span>. Namun atribut id ini tidak bersifat shareable, yang artinya nilai id ini harus unik dan digunakan pada satu elemen saja.

Untuk menetapkan selector dengan menggunakan id, kita gunakan tanda octothorpe (#) atau lebih familiar disebut dengan hash.

### Attribute Selector

Attribute selector merupakan cara menetapkan target elemen berdasarkan sebuah atribut yang digunakan atau bahkan bisa lebih spesifik dengan nilainya

|  |  |
| --- | --- |
| **Syntax** | **Description** |
| [attr] | Menargetkan elemen yang menerapkan atribut attr. |
| [attr=value] | Menargetkan elemen yang menerapkan atribut attr dengan nilai value. |
| [attr~=value] | Menargetkan elemen yang menerapkan atribut attr dengan salah satu nilainya adalah value. |
| [attr^=value] | Menargetkan elemen yang menerapkan atribut attr yang nilainya diawali dengan nilai value. |
| [attr$=value] | Menargetkan elemen yang menerapkan atribut attr yang nilainya diakhiri dengan value. |
| [attr\*=value] | Menargetkan elemen yang menerapkan atribut attr yang nilainya mengadung value. |

### Universal Selector

Universal selector digunakan untuk diterapkan pada seluruh elemen. Tetapi selector ini juga bisa secara spesifik menargetkan sebuah elemen dengan menggabungkan bersama selector yang lainnya. Berikut ini contohnya

## Combinators

Kita sudah mengetahui beberapa basic selector yang ada pada CSS. Tidak hanya sampai disitu, CSS Selector dapat berisi lebih dari satu basic selector. Di antara basic selector, kita dapat menyertakan sebuah kombinator.

Ada empat kombinator yang dapat kita gunakan yaitu: Adjacent Sibling Selector, General Sibling Selector, Child Selector, dan Descendant Selector.

### Adjacent Sibling Selector (+)

Adjacent Sibling Selector menggabungkan dua buah basic selector dengan menggunakan tanda + di antara keduanya.

Adjacent Sibling Selector terdiri dari dua buah target elemen, namun hanya elemen kedua yang menerapkan rule selama elemen tersebut dituliskan langsung setelah elemen pertama pada berkas HTML. Selain itu kedua elemen tersebut harus berada di dalam induk elemen yang sama. Pada contoh di atas rule akan diterapkan pada elemen paragraf yang berada tepat setelah elemen gambar.

### General Sibling Selector (~)

Mirip seperti Adjacent Sibling Selector namun rules akan diterapkan pada seluruh elemen kedua yang berada setelah elemen pertama. Dengan catatan, induknya sama, walaupun posisi dari elemen kedua tidak berada tepat setelahnya. General Sibling Selector menggunakan tanda ~ untuk menetapkan elemenya.

### Child Selector (>)

Child Selector menggabungkan dua buah basic selector dengan menggunakan tanda greater than (>) di antara basic selector-nya

### Descendant Selector (space)

Descendant Selector mirip seperti child selector namun hierarkinya lebih luas, karena rule akan diterapkan pada seluruh elemen yang menjadi turunannya walaupun secara tidak langsung. Basic selector pertama yang dituliskan pada selector ini menjadi induknya dan basic selector yang kedua merupakan elemen yang akan menerapkan rule. Selector ini menggunakan spasi dalam menggabungkan dua basic selector.

## Pseudo Selector

Selain beberapa selector yang sudah kita pelajari, CSS masih memiliki dua selector lagi yang dapat kita manfaatkan membantu menyeleksi elemen dalam menerapkan sebuah rule, yakni Pseudo-class dan Pseudo-element.

Sebelum menjelaskan lebih detail tentang kedua selector tersebut, perlu kita ketahui pseudo selector berbeda dari selector yang sudah dibahas sebelumnya, selector ini menargetkan elemen pada bagian yang “tidak terlihat” seperti sifat pada elemen, sehingga untuk menetapkannya, kita tidak bisa menggunakan selector biasa. Salah satu contoh yang paling sering kita terapkan adalah :hover, Pseudo Selector tersebut kita gunakan untuk menetapkan rule ketika cursor diarahkan ke target elemen.

### Pseudo-class Selector

Pseudo-class merupakan sebuah class “semu” yang sebenarnya ada pada tiap elemen HTML. Pada contoh sebelumnya kita sudah mengetahui salah satu pseudo-class selector, dengan menggunakan selector ini kita dapat memilih elemen berdasarkan class yang tidak nampak pada dokumen. Kita bisa menetapkan rule hanya ketika sebuah tautan telah dikunjungi (:visited) atau ketika sebuah elemen diarahkan dengan kursor (:hover).

Untuk menggunakan pseudo-class kita gunakan tanda titik dua (:) pada basic selector kemudian diikuti dengan pseudo-class nya.

Tidak hanya yang dicontohkan di atas, ada banyak sekali class semu yang menjadi standar pseudo-classes dalam CSS. Kita pun bisa melihat indeks standar yang ada pada pseudo-class pada tautan resmi Mozilla berikut: <https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes>

### Pseudo-elemen Selector

Sama seperti pseudo-class, pseudo-elemen merupakan sebuah elemen “semu” yang sebenarnya ada tetapi tidak tampak secara tertulis pada berkas HTML. Selector ini biasa digunakan ketika kita ingin menambahkan sebuah konten tepat sebelum dan setelah sebuah elemen paragraf. Alhasil kita tidak perlu menuliskan struktur elemen tersebut pada berkas HTML. Cukup manfaatkan pseudo-elemen ::before dan ::after kemudian tuliskan konten tersebut cukup pada CSS. Contohnya kita ingin menambahkan tanda kutip sebelum dan sesudah elemen blockquote

Untuk menggunakan pseudo-elemen kita gunakan dua buah tanda titik dua (::) kemudian diikuti dengan pseudo-elemen nya. Sebenarnya kita bisa menggunakan satu buah tanda tanda titik dua (:) namun kita perlu membedakannya dengan pseudo-class. Karena itulah baiknya ketika menggunakan pseudo-elemen, gunakanlah dua buah titik dua (::).

Pseudo-element tidak hanya ::before dan ::after. Dengan pseudo-elemen kita juga dapat menentukan rule pada awal karakter konten elemen.

Untuk mengetahui lebih lengkap mengenai apa saja pseudo-element yang dapat kita manfaatkan, kita bisa melihat dokumentasi resmi yang diberikan Mozilla pada tautan berikut:<https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements>