# 1 Introduction to CSS dan external stylesheet

Cara menggunakan CSS di dalam HTML adalah dengan menggunakan 3 macam cara, yaitu: Inline Style,

Internal Style, dan External Style Sheet.

**1. Inline Style**: memasukan style tanpa style sheet, langsung ke html tagnya.

*Contoh:*

|  |
| --- |
| <p **style="color:blue;margin-left:30px;"**>Membuat text menjadi biru dan memakai margin.</p> |

Inline style bukanlah cara yang terbaik untuk menerapkan style kepada HTML yang dituju. Karena beberapa penyebab, kurang lebih seperti berikut ini:

* Membuat HTML lebih sulit dibaca karena attribute style yang panjang sekali.
* Tidak dapat memberikan style sheet tanpa memberikan html sheet kepada developer lain.
* Mempersulit kerja sama antara frontend developer dan backend developer, karena mereka harus bekerja pada document yang sama.
* Sulitnya membuat tampilan responsive dan adaptive. Sulitnya melakukan style debugging.
* Inline style adalah "selector" terkuat, sehingga tidak ada selector jenis apa pun yang dapat melakukan override terhadap inline style, terkecuali dengan menggunakan "important property".
* Mengenai "selector" dan "important property" akan dijelaskan kemudian pada section selector dan property.

**2. Internal Style**: Dipakai kalau benar-benar dalam keadaan terpaksa. Dengan membuat semua css code dalam tag <style></style>

*Contoh:*

|  |
| --- |
| <p> Membuat text menjadi biru dan memakai margin.</p>  **<style>**  **p{color:blue;margin-left:30px;}**  **</style>** |

Internal style juga bukan cara terbaik untuk melakukan penerapan style kepada html content anda. Internal biasanya digunakan apabila frontend developer yakin bahwa style ini hanya akan digunakan pada halaman ini saja, dan adanya kondisi special yang memungkinkan client atau implementator hanya akan menerima html document saja (tidak mau menerima stylesheet atau css document).

Beberapa hal yang masih menjadi kekurangan dan kendala pada penerapan inline style:

* Tidak dapat memberikan style sheet tanpa memberikan html sheet kepada developer lain.
* Mempersulit kerja sama antara frontend developer dan backend developer, karena mereka harus bekerja pada document yang sama.
* Sulitnya melakukan style debugging.

**3. External Style:** Membuat CSS dalam file terpisah, dan semua codenya ditulis dalam file.css format. Sama seperti inline cara menulisnya, tapi yang ini di file terpisah dan tanpa menggunakan <style></style> tag.

*Contoh:*

|  |
| --- |
| <head>  **<link rel="stylesheet" type="text/css" href="mystyle.css">**  </head>  Lalu di dalam file mystyle.css, ketik  **p{color:blue;margin-left:30px;}** |

External adalah cara terbaik untuk menerapkan style kepada html. Keuntungan-keuntungan

Menerapkan external style kurang lebih seperti di bawah ini:

* Membuat HTML lebih mudah dibaca, karena tidak adanya penulisan code style di dalamnya. Perubahan pada style sheet akan independent dari html documentnya.
* Mempermudah kerja sama antar frontend developer dan backend developer.
* Mudahnya untuk membuat responsive dan adaptive style.
* Dengan menggunakan web browser yang memiliki inspecting tools atau debugger, external style dapat di debug dengan melacak line number dan document name dari style sheetnya.
* Selalu ada cara untuk melakukan override sebuah style dengan menggunakan selector yang tepat. Sehingga apa pun style yang diaplikasikan ke dalam aplikasi, selalu bisa diubah tanpa harus menghapus code style sebelumnya. Untuk keterangan lebih lanjut bisa dibaca di section selector.
* Tidak perlu menggunakan important property dalam penerapannya, karena selectornya selalu bisa di override.
* Dapat menggunakan macam-macam preprocessor document seperti SASS dan LESS. Keduanya akan dibahas di section berikutnya.

# 2 Introduction to CSS Components

Component CSS tidaklah rumit, karena CSS hanya punya 3 components, yaitu: **Selector**, **Properties** dan **Values**. Contoh:

|  |
| --- |
| **div.main-container**  **{**  **color:white;**  **background-color: #106fbe;**  **width:300px;**  **margin-left:10px;**  **}** |

**1. Selector (Pemilih atau Penunjuk nodes)**

Ini adalah component terpenting dalam CSS! Mengerti dan menguasai selector maka kurang lebih anda sudah menguasai 40% dari CSS! Selector berperan untuk menunjuk nodes di dalam sebuah DOM. Oleh karena itu developer harus mengerti konsep DOM sebelum memahami selector lebih lanjut.

**Jangan meremehkan selector! Karena kamu akan pakai ini lagi di native javascript dan jquery di hari berikutnya. Tidak memahami selector akan menyulitkan mu belajar javascript juga.**

**2. Properties (Nama variable-variable pada CSS)**

Properties adalah fasilitas yang disediakan css, dimana fasilitas itu adalah hal-hal yang bisa kamu rubah pada tag tersebut! Fasilitas tersebut berupa css library yang disediakan oleh versi CSS tersebut. Properties bukanlah hal sulit dimengerti namun sulit dikuasai bagi yang CSSnya kurang jitu! The best way to master properties adalah lewat latihan dan pengalaman.

Setiap properties memiliki 2 karakteristik utama, yaitu short-hand notation properties dan dependent properties:

Short-hand properties adalah satu properties yang merupakan kombinasi dari banyak properties lainnya. Contohnya akan diberikan pada section berikutnya.

Dependent properties artinya adalah dimana properties tersebut bergantung pada properties. Lebih jelasnya mengenai properties akan dapat di lihat di section css properties and value. Pada contoh di atas properties adalah color dan margin-left.

**3. Values (Nilai dari properties itu)**

Values adalah nilai dari satu property. Values yang tersedia atau yang bisa dipakai tergantung dari macam propertiesnya. Sama seperti properties, values mudah dimengerti tapi sulit dikuasai.

Pilihan values yang dapat digunakan tergantung dari propertiesnya. Sama seperti properties values juga dapat bersifat dependent dengan value lainnya. Setiap macam values akan dijelaskan di dalam section css properties and value. Pada contoh di atas value adalah white dan 300px.

# 3 The Science of Web User Interface

Secara default semua html tag memiliki css property dan value yang diberikan oleh browser render engine, user agent stylesheet, dan initial value.

**Browser render engine**, adalah technology yang digunakan setiap web browser untuk menerjemahkan suatu code markup dalam suatu specific style menjadi tampilan yang anda bisa lihat.

<https://en.wikipedia.org/wiki/Web_browser_engine>

Perbedaan berbagai macam rendering engine ini yang menyusahkan hidup kita. Bukan hanya default viewnya saja yang berbeda, ada code-code di dalam css pula yang di satu browser bisa bekerja di browser lain tidak, misalnya border-radius, di beberapa versi mozilla fire fox dan IE yang lama property border radius tidak bisa ditulis seperti biasa.

|  |
| --- |
| -webkit-border-radius (untuk safari dulu)  -moz-border-radius (untuk mozilla ff)  -ms-border-radius (untuk IE)  border-radius (normalnya) |

Tetapi browser versi-versi terbaru kebanyakan sudah bekerja dengan border-radius normal. Biasanya untuk jaga-jaga, developer menulis ke-4nya, sehingga kalau salah satu tidak jalan masih ada yang lain.

**User agent style sheet** bisa dilihat pada saat inspecting element pada browser. User agent stylesheet suka disebut sebagai Default CSS setting. Default CSS Setting adalah property dan attribute yang secara default diberikan ke sebagian markup dengan tujuan mempermudah programmer malas.

Sebagian besar tag yang exist pada saat HTML 5, bersifat “Semantic”. Yang artinya, tag tersebut sudah dibanjiri banyak setting untuk front-end programmer yang malas membuat sesuatu hanya dari tag-tag primitif seperti div dan span. Check semantic tag yang muncul pada html 5 lewat link dibawah ini.

<http://www.w3schools.com/html/html5_semantic_elements.asp>

Untuk mengetahui setiap default setting dari setiap tag, lihat keterangan setiap tag di W3C school. Contohnya untuk div, default settingnya adalah display: block;

Untuk lebih jelasnya check link dibawah.

<http://www.w3schools.com/tags/tag_div.asp>

User agent stylesheet/ Default value adalah sebuah selector paling lemah, sehingga lewat selector apapun anda memberikan style pada suatu tag, dia akan ter-replace.

**Initial value** adalah value yang diberikan pada suatu tag sebelum user agent stylesheet memberikannya. Initial value tidak bisa dilihat didalam inspect element, initial value bisa didapat dengan memasukan attribute initial ke dalam css propertynya.

Sebagai contoh sebuah div memiliki user agent stylesheet display:block, tetapi initial valuenya adalah display:inline.