# DAY 7 : Pendalaman CSS

Pada pembelajaran sebelumnya kita sudah mengenal dasar-dasar CSS seperti membuat berkas CSS, rule/aturan penulisan dalam CSS, dan juga melampirkan berkas CSS kita kedalam berkas HTML. Pada pembelajaran kali ini kita akan mendalami CSS lebih dalam lagi. Beberapa materi pendalaman CSS yang kita bahas diantaranya CSS Conception, Selectors, Pseudo Selectors, dan juga Font Styling. Sudah tidak sabar? Mari kita langsung bahas materinya!

## 1. CSS Conception

### 1.1 Inheritance

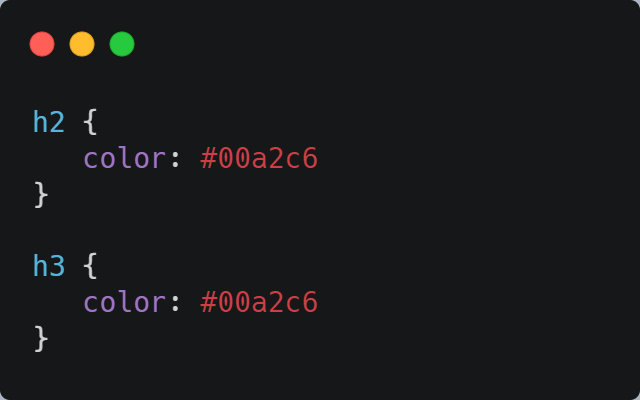
Styling HTML bersifat inheritance yang artinya dapat mewarisi properti style “tertentu” pada elemen yang ada di dalamnya. Contohnya pada rules yang kita tuliskan untuk elemen <body></body> akan diterapkan pada seluruh elemen yang ada di dalam elemennya (body). Contoh lainnya, pada rules yang diterapkan pada elemen <footer></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.

Diagram

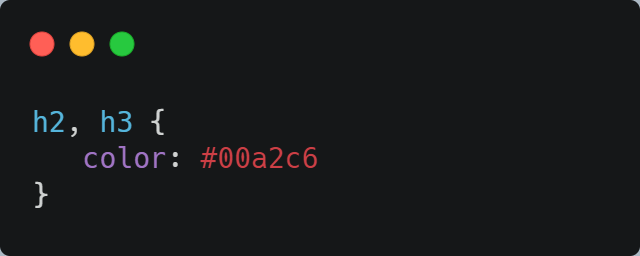
Description automatically generated

### 1.2 Group Selectors

Jika kita menerapkan rule yang sama pada beberapa selector yang berbeda, di CSS kita dapat menggabungkan selector tersebut sehingga dapat meminimalisir penulisan kode yang berulang.



Jika terdapat kasus seperti ini, kita dapat menuliskan dua selector sekaligus dalam satu struktur rule. Gunakan tanda koma (,) untuk memisahkan tiap selector-nya. Silakan kita buka kembali berkas style.css dan gabungkan rule untuk elemen <h2> dan <h3> menjadi seperti ini:



### 1.3 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. Untuk lebih jelasnya, kita bisa lihat contoh berikut:



Tetapi kita bisa membuat sebuah property styling agar dianggap penting oleh browser untuk diterapkan dan tidak memperhatikan urutan. Kita bisa menambahkan keyword !important pada akhir nilai propertinya.



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

## 2. Selectors

Ada banyak jenis selector untuk menargetkan aturan ke elemen tertentu dalam dokumen HTML. Pada sub-modul sebelumnya, kita sudah mengetahui salah satu cara dasar dalam menggunakan selector. Sekarang, mari kita bahas lebih detail mengenai macam macam selector yang ada. Sebenarnya terdapat beberapa macam dari selector basic, yakni:

### 2.1 Type

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. Contohnya sebagai berikut:

Text

Description automatically generated

Jika berkas tersebut dibuka pada browser, maka teks yang berada pada setiap elemen <span></span> akan berwarna merah.

Graphical user interface, text, application

Description automatically generated

### 2.2 Class

Selector digunakan untuk menyeleksi element berdasarkan nama class atau nilai (value) dari attribute class yang sama pada html element yang sebelumnya telah ditentukan.

Penulisan Class Selector diawali dengan tanda titik (.) atau period, kemudian diikuti dengan nama class. Kita bisa menentukan sendiri nama class apa saja yang kita inginkan, asalkan mengikuti aturan berikut:

* Nama Class hanya bisa ditulis dengan letters (abjad [A-Z, a-z]), numbers (angka), hyphens (tanda penghubung), dan underscores (tanda garis bawah).
* Setelah titik (.) nama class harus diawali dengan letters (huruf alphabet) pada awal karakter. contoh: .wali9 adalah benar, tetapi, .9wali adalah salah.
* Nama Class adalah case-sensitive. Artinya perbedaan huruf besar dengan huruf kecil akan berpengaruh. contoh: .content berbeda dengan .Content karena pebedaan huruf besar C pada awal karakter.

Class Selector biasanya digunakan untuk memberikan style pada element berdasarkan group tertentu. Satu nama class bisa disisipkan pada beberapa element agar memiliki style yang sama. Jadi, bisa digunakan lebih dari satu deklarasi. Ini, jelas berbeda dengan ID Selector yang hanya bisa digunakan sekali dalam satu halaman.

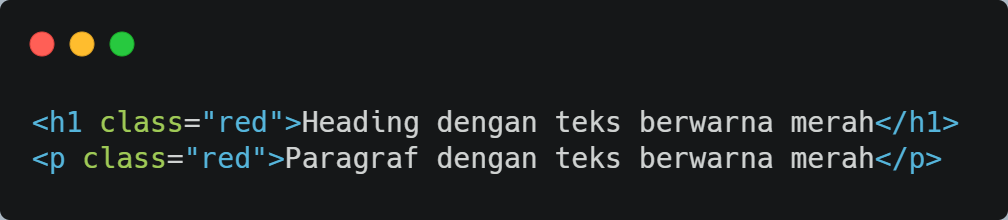


Jika berkas tersebut dibuka pada browser, maka akan nampak seperti ini:

Graphical user interface, text, application, website

Description automatically generated

Seperti sudah disinggung sebelumnya bahwa 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.



Tidak hanya itu, 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.

Text

Description automatically generated

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. Perhatikan contoh berikut:



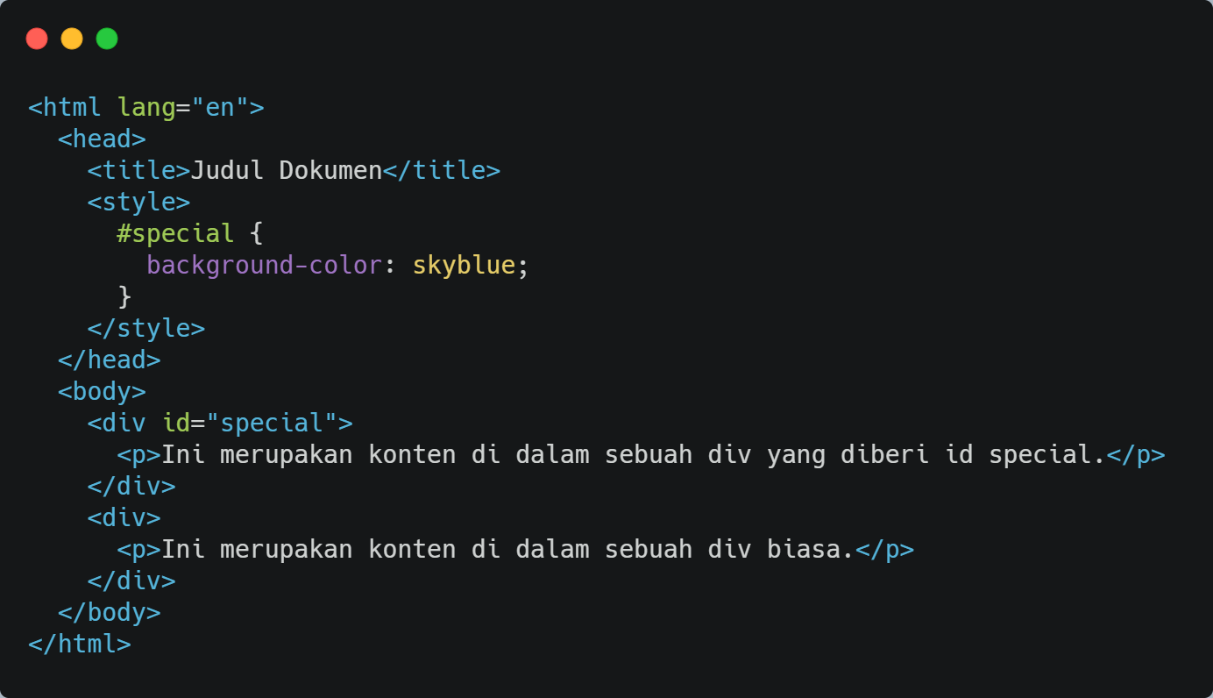
Jika berkas tersebut dibuka pada browser, maka gaya fancy yang diterapkan pada elemen heading dan elemen paragraf akan berbeda.

Graphical user interface, text, application

Description automatically generated

### 2.3 Id

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. Dalam penggunaannya, ID selector diawali dengan tanda pagar (#) atau hash. Berikut ini contoh penggunaan selector menggunakan Id:



Jika kode di atas dibuka pada browser, maka tampilan akan tampak seperti ini:

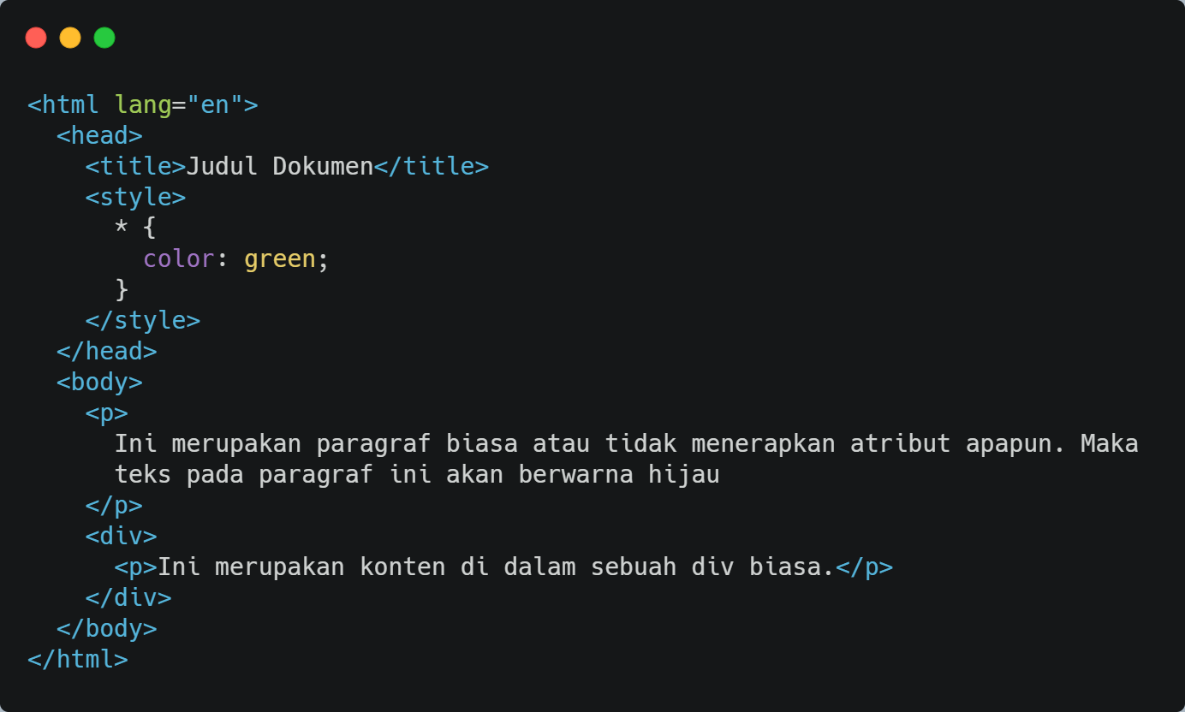
Graphical user interface, text, application

Description automatically generated

Yang harus kita perhatikan kembali adalah id ini bersifat unik. Jika kita ingin menerapkan sebuah rule pada banyak elemen, sebaiknya gunakan atribut class, bukan dengan id.

### 2.4 Universal

CSS Universal Selector (\*) digunakan untuk menyeleksi setiap single element dari tipe element apapun. Artinya, penyeleksian berlaku secara keseluruhan untuk semua single element. Penggunaan Universal Selector ditandai dengan karakter arterisk \* yang dapat digunakan secara berdiri sendiri ataupun digabungkan dengan tipe selektor lainnya.



Jika kode di atas dibuka pada browser, maka tampilan akan tampak seperti ini:

Graphical user interface, text, application

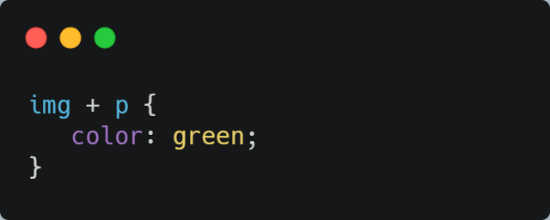
Description automatically generated

## 3. 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 tiga jenis kombinator yang akan kita pelajari yaitu: Adjacent Sibling Selector, General Sibling Selector, Child Selector, dan Descendant Selector.

### 3.1 Adjacent Sibling

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



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. Berikut contoh penerapannya:



Jika berkas di atas dijalankan pada browser, maka akan menghasilkan tampilan seperti berikut:

Graphical user interface, text, application

Description automatically generated

### 3.2 Child

Child Selector menggabungkan dua buah basic selector dengan menggunakan tanda greater than (>) di antara basic selector-nya. Contohnya sebagai berikut:



Pada contoh di atas, rule akan diterapkan pada seluruh elemen paragraf yang berada di dalam elemen div secara langsung. Dalam arti lain, elemen paragraf merupakan anak dari **elemen div secara langsung**. Untuk lebih jelas, perhatikan contoh penerapannya berikut:



Jika berkas di atas dijalankan pada browser, maka akan menghasilkan tampilan seperti berikut:

Graphical user interface, text, application, website

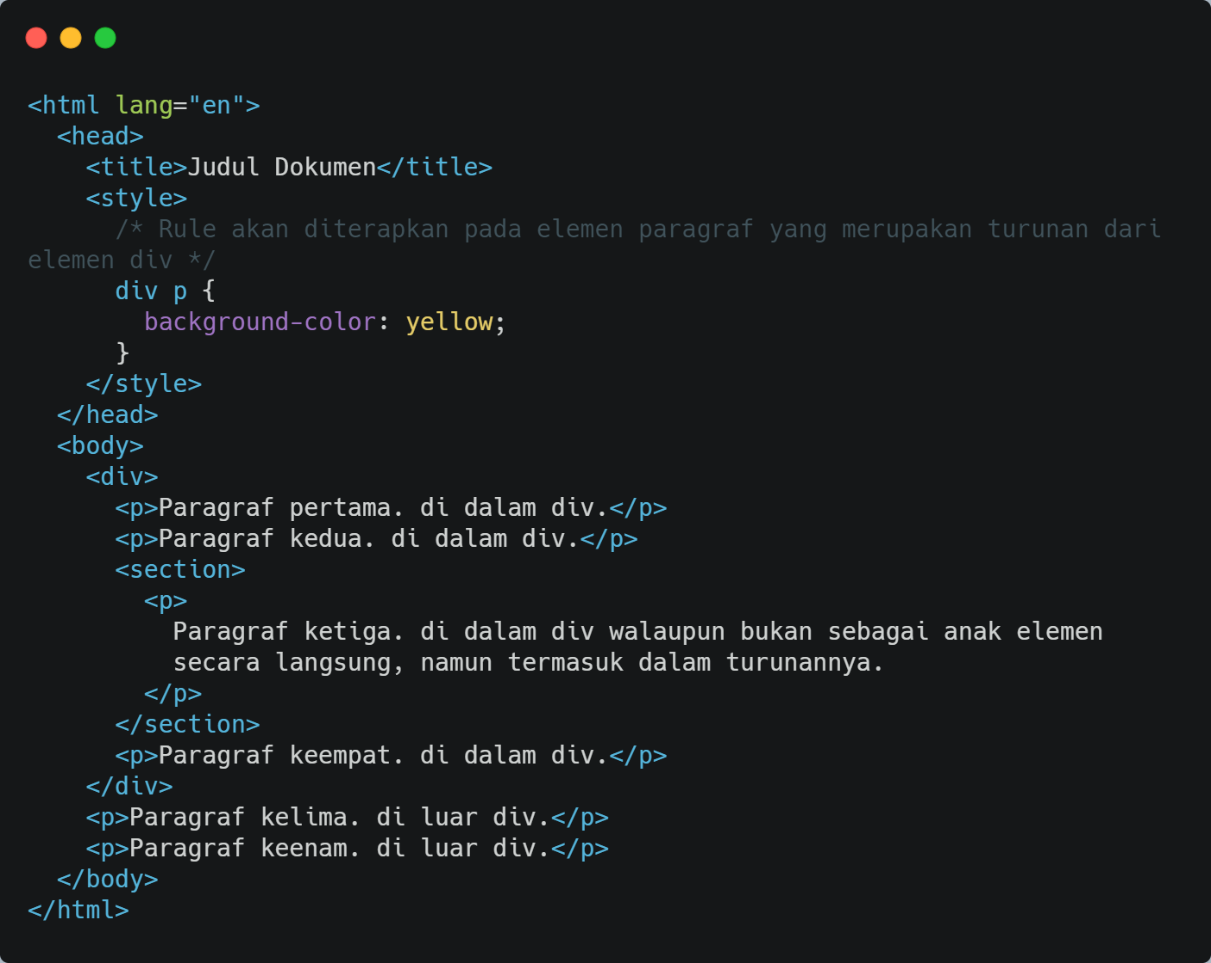
Description automatically generated

### 3.3 Descendant

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. Contohnya sebagai berikut:



Pada contoh di atas rule akan diterapkan pada seluruh elemen paragraf yang merupakan turunan dari elemen div. Berikut contoh penerapannya:



Jika berkas di atas dijalankan pada browser, maka akan menghasilkan tampilan seperti berikut:

Graphical user interface, text, application, email

Description automatically generated

## 4. Pseudo Selector

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.



Berikut contoh beberapa Psedo Selector lain yang dapat kita manfaatkan pada website kita

