# DAY 8 : Pendalaman CSS 2

Setelah kita tahu bagaimana menerapkan CSS pada berkas HTML, mengetahui struktur penulisan CSS, dan menggunakan selector sesuai dengan kebutuhan, apa berikutnya? Kita akan fokus mempelajari secara detail bagaimana menerapkan styling pada sebuah teks.

## 1. Text VS Font Styling

Perlu kita ketahui, CSS memisahkan styling untuk font dan teks. Kita harus tahu kapan seharusnya menetapkan styling pada font atau menetapkannya pada teks. Mengapa kedua hal tersebut dipisah? Padahal teks dan font merupakan satu kesatuan, di mana setiap teks pasti menggunakan font untuk menampilkannya, bukan? Sebabnya, CSS sendiri mempunyai properti untuk kedua hal tersebut secara masing-masing.

Jika kita ingin menetapkan styling pada tampilan teks itu sendiri, gunakanlah properti **font**. Contohnya properti font dapat mengatur tipe font, ukuran, ketebalan, dan lainnya. Sedangkan properti **teks** digunakan untuk mendukung hal lainnya dalam menampilkan teks seperti, menetapkan text alignment, dekorasi, spacing, dan masih banyak lagi. Karena dua hal ini dipisah, mari kita bahas satu persatu.

## 2. Font Styling

Ketika kita membuat sebuah dokumen teks, termasuk dokumen cetak, langkah awal kita biasanya adalah menentukan jenis font yang akan digunakan. Pada pengembangan website pun demikian. Dalam CSS, font ditentukan dengan menggunakan beberapa paket properti font. Kita bisa atur tipe font, ukuran, ketebalan, dan gaya. Berikut ini merupakan properti font yang akan kita pelajari antara lain:

|  |  |
| --- | --- |
| **Nama Properti** | **Fungsi** |
| font-family | Menetapkan jenis font yang akan diterapkan pada target. |
| font-size | Menentukan ukuran pada teks |
| font-weight | Menentukan ketebalan pada teks. |
| font-variant | Menentukan teks untuk menggunakan gaya small caps (huruf kapital kecil). |
| font | Shorthand dari properti font yang ada. |

### 2.1 Font Family (font-family)

Properti font-family merupakan properti yang digunakan untuk menentukan dan merubah jenis font yang digunakan pada teks. Pada sub-modul pengenalan CSS kita sudah mencoba menggunakan font properties ini untuk mengubah standar font yang ditampilkan pada browser dengan menggunakan font-family pada elemen body.

Text

Description automatically generated with medium confidence

Penerapan rule css ini merubah tampilan font yang kita miliki pada website kita yang pada awalnya seperti ini:

Graphical user interface, text, application, email

Description automatically generated

Setelah diterapkan font dengan jenis sans-serif menjadi seperti ini:

Graphical user interface, text

Description automatically generated

Pada rule tersebut kita mengubah standar font yang digunakan browser dengan font ‘sans-serif’. Sebenarnya untuk nilai dari properti ini dapat lebih dari satu (dikenal sebagai font stack). Tujuannya adalah sebagai fallback jika terjadi kegagalan dalam menggunakan font yang kita gunakan.

Graphical user interface

Description automatically generated

Untuk menuliskan lebih dari satu nilai font, berikut aturan yang harus kita perhatikan:

* Seluruh nilai font yang bukan merupakan generic font families, yang mana harus dituliskan secara kapital. Contohnya “Arial” bukan dituliskan “arial”.
* Gunakan tanda koma (,) untuk memisahkan antara nilai font yang digunakan.
* Selalu tanda kutip (“) untuk membungkus nilai font yang memiliki spasi pada namanya (Contohnya “Open Sans”).

Mungkin kita bertanya-tanya mengapa perlu memberikan lebih dari satu nilai pada font-family? Hal tersebut penting karena tidak semua browser mendukung semua jenis font. Memberikan lebih dari satu nilai font dapat menawarkan alternatif tampilan font pada browser. Terutama jika font utama yang diterapkan tidak didukung oleh browser yang digunakan.

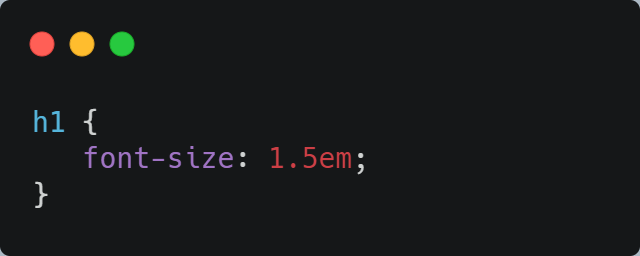
Bagaimana urutan prioritasnya? Mulai dari jenis font yang pertama dituliskan. Jika font pertama didukung oleh browser maka browser akan menggunakannya, jika tidak lantas browser akan memeriksa nilai font yang kedua dan menggunakannya (jika didukung), demikian dan seterusnya.

Pastikan untuk menggunakan generic font families pada akhir nilai properti font-family, karena nilai ini dipastikan didukung oleh seluruh browser saat ini. Lantas apa saja nilai dari generic font families ini? Berikut nilai-nilai generic font families yang dapat kita gunakan untuk fallback mechanism:

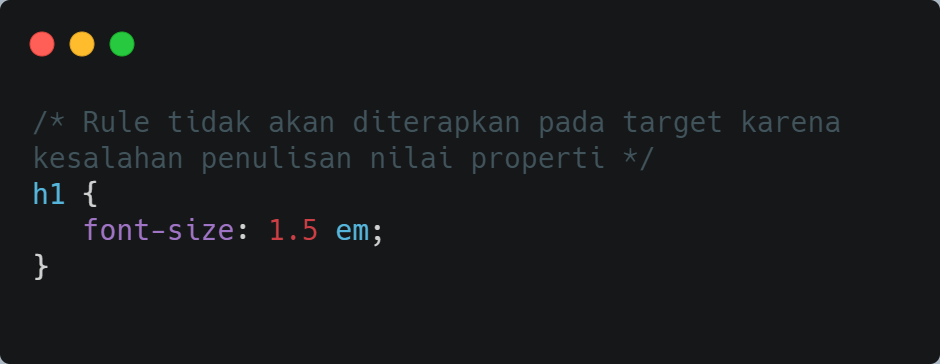
* serif : jenis font yang memiliki runcing pada garis akhir karakternya. Times New Roman merupakan salah satu jenis serif font.
* sans-serif : jenis font yang tidak meruncing pada garis akhir karakternya. Contohnya “Open Sans”, “Fira Sans” dan lainnya.
* monospace : jenis font yang memiliki nilai lebar tiap karakternya sama. Consolas merupakan salah satu jenisnya.
* cursive: jenis font yang tampak seperti handwriting atau hasil tulisan tangan.
* fantasy : jenis font yang merepresentasikan karakteristik yang menyenangkan.
* system-ui : jika menerapkan nilai ini maka font yang diterapkan akan sama seperti font yang digunakan pada sistem operasi kita.
* math : jenis font yang digunakan untuk penulisan rumus-rumus matematika.
* emoji : jenis font yang digunakan untuk menampilkan emoji.
* fangsong : jenis font yang menampilkan gaya penulisan Chinese.

### 2.2 Font Size (font-size)

Properti font-size merupakan property CSS yang digunakan untuk menentukan ukuran font yang digunakan pada teks. Artinya, kita bisa merubah ukuran Teks pada Html yang nantinya akan ditampilkan pada Halaman Web. Untuk menetapkan ukuran font kita kita perlu menerapkan properti font-size. Kita bisa menetapkan nilai dari properti ini dengan menuliskan langsung nilai dan satuannya. Contohnya seperti ini:



Pastikan bahwa saat menuliskan nilai dan satuannya, tidak ada jarak (spasi). Penulisan spasi akan membuat rule tidak diterapkan akibat kesalahan penulisan yang tidak dapat terbaca oleh CSS.



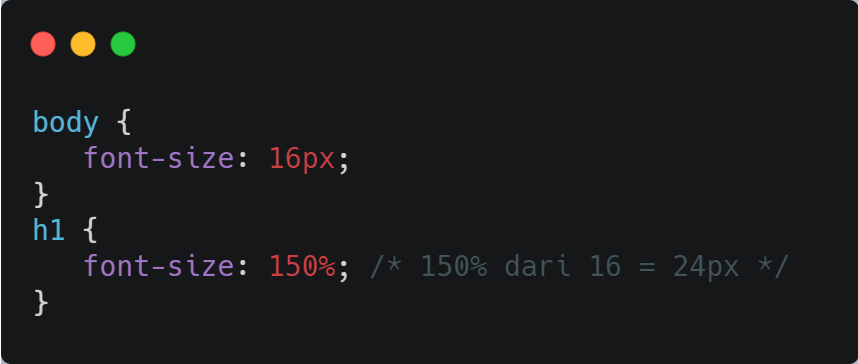
Satuan dalam menetapkan ukuran font terdapat dua jenis. Yang pertama relative, yakni satuan yang nilainya tergantung pada sesuatu hal, contohnya ukuran dari viewport, induk elemen ataupun ukuran teks standar.

|  |  |  |
| --- | --- | --- |
| **Relative Unit** | | |
| **Satuan** | **Relative kepada** | **Fungsi** |
| em | Font size | Satuan relatif terhadap ukuran font yang sedang digunakan pada elemen (contohnya, 2em berarti 2 kali lebih besar dari ukuran font seharusnya). |
| rem | Font size | Mirip seperti em, tetapi rem merupakan satuan relatif terhadap ukuran font dari root element. |
| vw | Viewport width | Satuan relatif terhadap 1% lebar viewport. Contoh 1vw = 1% dari lebar viewport. Satuan ini tidak didukung pada browser IE8 ke bawah. |
| vh | Viewport height | Satuan relatif terhadap 1% tinggi viewport. Contoh 1vh = 1% dari tinggi viewport. Satuan ini tidak didukung pada browser IE8 ke bawah. |

Dan yang kedua adalah absolute, yakni satuan yang nilainya telah ditentukan atau digunakan dalam dunia nyata.

|  |  |
| --- | --- |
| **Absolute Unit** | |
| **Satuan** | **Fungsi** |
| px | Menetapkan nilai font berdasarkan ukuran pixel |
| pt | Menetapkan nilai font berdasarkan points (1/72 inch di CSS2.1) |
| pc | Menetapkan nilai font berdasarkan picas (1 pica = 12 point) |
| mm | Menetapkan nilai font berdasarkan millimeters |
| cm | Menetapkan nilai font berdasarkan centimeters |
| in | Menetapkan nilai font berdasarkan inches |

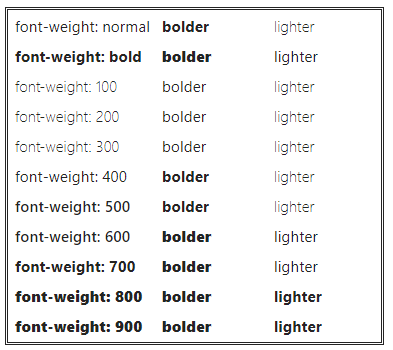
Selain dengan menetapkan nilai dan satuannya secara langsung, untuk mengatur ukuran font kita juga bisa gunakan nilai persentase.



Pada contoh ini ukuran font dari elemen <h1> seharusnya memiliki ukuran 16px karena mewarisi dari induk elemennya (body). Tetapi di bawahnya terdapat rule yang menargetkan secara spesifik untuk elemen <h1> untuk menerapkan ukuran font sebesar 150% dari ukuran induknya. Maka elemen <h1> akan nampak 50% lebih besar dari elemen lain yang ada di dalam <body>.

### 2.3 Font Weight (font-weight)

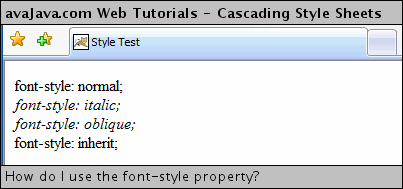
Setelah mengenal font families dan font size, selanjutnya ada juga font-weight yang digunakan untuk mengatur ketebalan dari font yang ditampilkan. Nilai dari properti ini dapat ditentukan dengan menggunakan numeric values (100 sampai 900) atau dengan menggunakan descriptive terms (normal, bold, bolder, dan lighter). Perhatikan contoh penulisan dan hasil implementasi font-weight berikut ini:



Properti font-weight dapat diaplikasikan ke seluruh elemen yang ada di HTML dan nilainya dapat diturunkan pada elemen turunannya.

### 2.4 Font Style (font-style)

Properti selanjutnya adalah font-style. Properti ini digunakan untuk menentukan postur dari teks yang ditampilkan, apakah bentuknya vertikal (normal) atau miring (italic dan oblique).



Italic dan oblique keduanya menampilkan teks yang miring. Perbedaanya adalah italic menerapkan tipe miring (italic font version) dari suatu font sedangkan oblique adalah font normal yang hanya dibuat miring.

Properti font-style dapat diaplikasikan ke seluruh elemen yang ada di HTML dan nilainya dapat diturunkan pada elemen turunannya.

## 3. Shorthand Font Styling

Menspesifikasikan masing-masing nilai properti font akan menghasilkan banyak sekali kode repetitif. Dengan begitu CSS memberikan suatu “jalan pintas” untuk menuliskan properti-properti tersebut ke dalam satu properti yaitu font.

Dengan menggunakan properti font kita dapat menuliskan beberapa properti hanya dalam satu properti pada satu rule.

Logo, company name

Description automatically generated

Nilai dari properti font merupakan nilai dari seluruh properti dari font yang sudah kita bahas. Tiap nilai dipisahkan menggunakan spasi. Pada properti ini urutan nilai merupakan hal yang penting, sehingga jangan sampai salah urutan dalam menuliskannya. Kesalahan penulisan atau urutan menyebabkan seluruh struktur rule menjadi tidak valid.

Walaupun begitu kita bisa tidak menuliskan seluruh nilai properti yang ada, tetapi nilai dari properti font-size dan font-family wajib ada ketika menggunakan properti ini. Berikut contohnya penulisan minimal ketika kita menggunakan properti font:

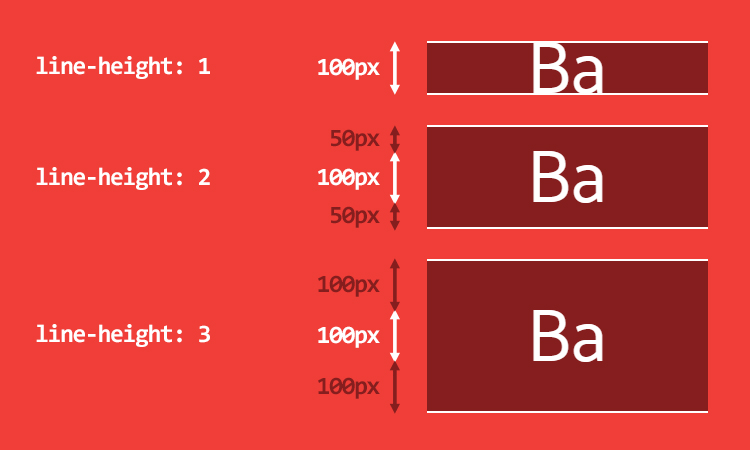


## 4. Text Styling

Pembahasan sebelumnya kita terfokus pada formating bentuk karakter yang ditampilkan dengan menggunakan beberapa properti font yang ada. Sekarang kita akan mempelajari bagaimana seorang developer bisa memberikan formatting pada keseluruhan teks yang ada pada baris paragraf, seperti menetapkan inden, jarak antar baris, kata dan huruf, dan sebagainya. Maka dari itu, mari kita bahas satu persatu.

### 4.1 Line Height

Properti line-height digunakan untuk mengatur jarak minimal dari garis dasar ke garis dasar dalam menampilkannya teks pada halaman.



Berikut merupakan contoh tiga cara berbeda dalam menerapkan tinggi baris dua kali lebih tinggi dari ukuran font:

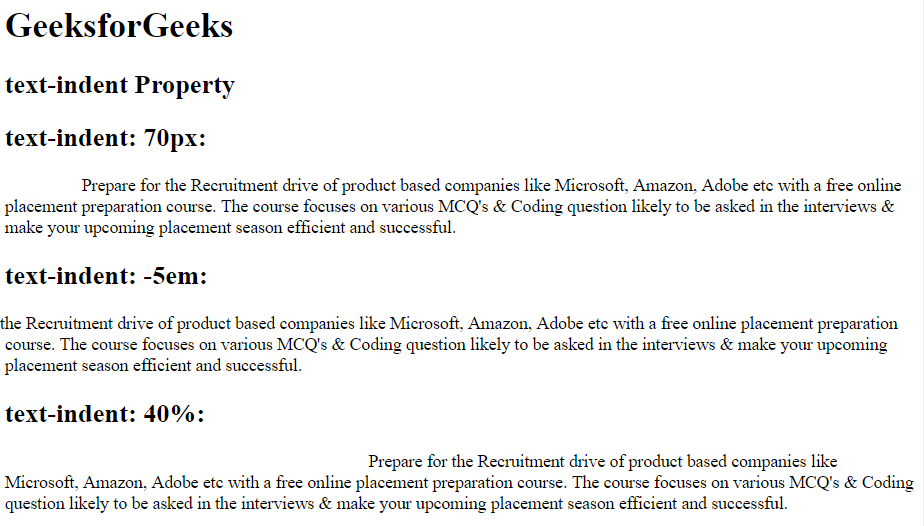


Cara pertama merupakan cara yang paling mudah digunakan, karena kita dapat menentukan nilai hanya dengan satu angka, di mana angka tersebut nantinya dikalikan dengan nilai font-size sebelum diterapkan pada nilai properti line-height. Contohnya, ukuran font standar pada paragraf adalah 16 pixel. ,Kita definisikan properti line-height dengan nilai 2, maka nilai properti line-height seharusnya adalah 16 pixel \* 2 = 32 pixel (dua kali lebih besar dari ukuran font).

Properti line-height dapat diaplikasikan ke seluruh elemen yang ada pada HTML dan nilainya dapat diturunkan pada elemen turunannya.

### 4.2 Text Indent

Dalam membuat sebuah dokumen tidak jarang kita membutuhkan jarak/lekuk di awal paragraf. Hal tersebut dapat dilakukan pada website dengan menerapkan properti text-indent. Kita dapat menentukan nilai properti ini melalui perhitungan panjang dalam px, em, dan in atau bisa menggunakan nilai persentase (%). Nilai persentase dihitung berdasarkan lebar dari induk elemen. Berikut merupakan contoh penggunaannya:



Pada contoh ke dua kita bisa melihat bahwa pada nilai properti ini dapat diberikan nilai negatif. Jika kita menggunakannya, maka baris pertama pada paragraf akan keluar dari batas elemen yang menampungnya (biasa disebut hanging indent).

Perlu diingat kembali, properti ini hanya berpengaruh pada awal baris paragraf. Jika kita ingin menetapkannya untuk seluruh baris kita dapat gunakan margin atau padding, keduanya akan dibahas pada materi box model.

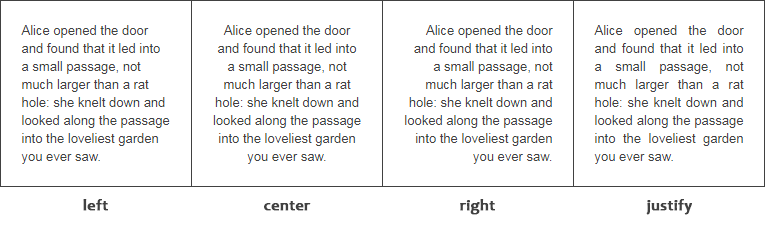
### 4.3 Text Alignment

Kita bisa mengatur text alignment pada website seperti kita melakukannya pada aplikasi Microsoft Word dengan menggunakan properti text-align. Untuk standarnya, properti ini bernilai left atau biasa kita sebut rata kiri. Tetapi jika kita menerapkan atribut language dengan nilai bahasa yang arah bacanya berlawanan, maka standarnya akan menggunakan right atau rata kanan.

Berikut ini nilai yang dapat digunakan pada properti text-align:

|  |  |
| --- | --- |
| **Nilai Properti** | **Fungsi** |
| text-align: **left** | Membuat perataan teks pada ujung kiri |
| text-align: **right** | Membuat perataan teks pada ujung kanan |
| text-align: **center** | Membuat perataan teks secara menengah |
| text-align: **justify** | Membuat perataan teks yang setara pada ujung kiri dan kanannya |

Bagaimana? Pasti kita sudah familiar dengan nilai-nilai tersebut? Berikut contoh penggunaan dari properti text-align:



### 4.4 Text Decoration

Properti ini paling populer digunakan ketika kita ingin membuat garis bawah atau underline pada teks. Tapi tak hanya itu, ada beberapa nilai lain yang dapat kita gunakan untuk properti ini. Detailnya sebagai berikut:

|  |  |
| --- | --- |
| **Nilai Properti** | **Fungsi** |
| text-decoration: **underline** | Memberikan garis bawah (underline) pada teks |
| text-decoration: **overline** | Memberikan garis atas (overline) pada teks |
| text-decoration: **line-through** | Memberikan efek tulisan dicoret (strikethrough) |
| text-decoration: **none** | Menghilangkan dekorasi teks yang ada pada elemen |

### 4.5 Text Transform

Fitur ini sangat membantu di kala kita ingin mengubah kapitalisasi tanpa harus menuliskan kembali teksnya. Pada CSS juga terdapat fitur serupa, yaitu dengan menggunakan properti text-transform. Ketika kita menerapkan properti text-transform pada elemen teks, maka kapitalisasi akan berubah ketika halaman di-render tanpa harus mengubahnya pada dokumen HTML.

Properti ini dapat berisikan nilai sebagai berikut:

|  |  |
| --- | --- |
| **Nilai Properti** | **Fungsi** |
| text-transform: **none** | Teks yang ditampilkan sama seperti yang dituliskan |
| text-transform: **capitalize** | Membuat huruf pertama besar pada tiap katanya |
| text-transform: **lowercase** | Membuat seluruh teks menggunakan huruf kecil |
| Text-transform: **uppercase** | Membuat seluruh teks menggunakan huruf besar |

### 4.6 Text Shadow

Memberikan bayangan pada teks telah menjadi hal yang umum digunakan meskipun tidak memiliki dukungan di semua browser. Pada CSS untuk membuat bayangan pada teks (atau biasa disebut drop shadow) kita dapat gunakan properti text-shadow.

Nilai dari properti ini cukup rumit karena membutuhkan tiga buah nilai dan satu buah nilai warna sehingga membutuhkan empat nilai dalam satu properti untuk menentukan bayangannya:

1. **Nilai pertama** : menunjukkan seberapa jauh ke kiri atau kanan (horizontal) bayangan harus ditampakkan.
2. **Nilai kedua** : menunjukkan jarak ke atas atau ke bawah (vertical) bayangan harus ditampakkan.
3. **Nilai Ketiga (opsional)** : menentukan tingkat keburaman yang harus diterapkan pada bayangan.
4. **Nilai Keempat :** menentukan warna yang digunakan pada bayangan.

Berikut ini contoh penggunaan dari properti drop shadow:

