



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:

- `font-family` : Menetapkan jenis font yang akan diterapkan pada target.
- `font-size` : Menentukan ukuran pada teks.
- `font-weight` : Menentukan ketebalan pada teks.
- `font-style` : Menetapkan styling yang diterapkan pada teks.
- `font-variant` : Menentukan teks untuk menggunakan gaya small caps (huruf kapital kecil).
- `font` : Shorthand dari properti font yang ada.

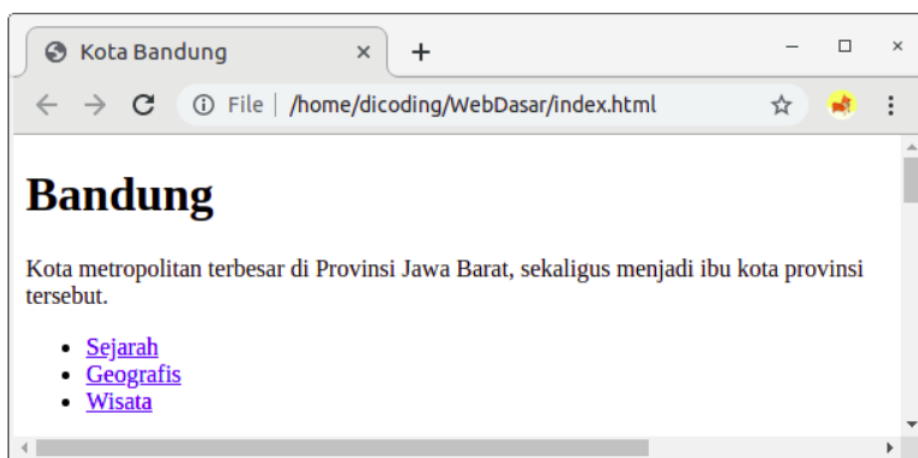
Mari kita bahas properti tersebut satu persatu secara mendetail.

font-family

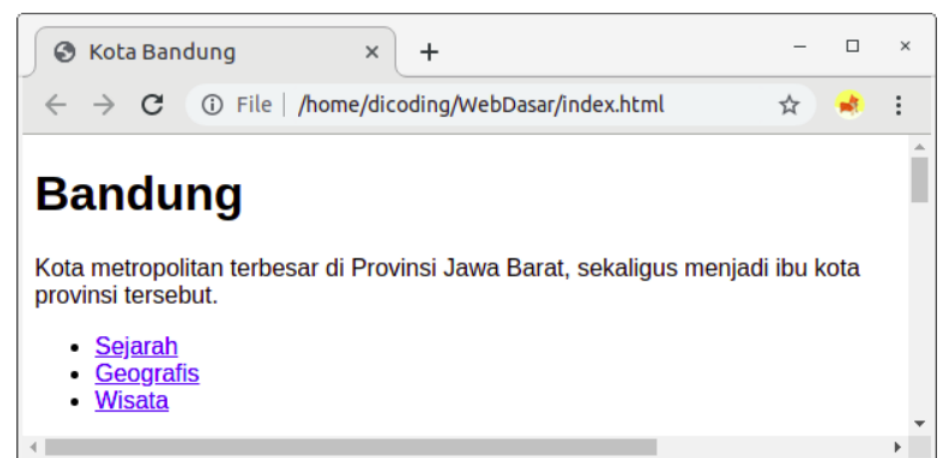
Pada modul pengenalan CSS kita sudah mencoba menggunakan *font properties* ini untuk mengubah standar font yang ditampilkan pada browser dengan menggunakan *font-family* pada elemen *body*.

```
1. body {  
2.     font-family: sans-serif;  
3. }
```

Sehingga menghasilkan tampilan:



Sebelum menerapkan *styling*



Setelah menerapkan *styling*

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.





p { font-family : Arial, Verdana, sans-serif }

Property Name Value 1 Value 2 Value 3

span { font-family : "Open Sans", Verdana, sans-serif }

Property Name Value 1 Value 2 Value 3

- 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.

font-size

Mengubah nilai font pada sebuah dokumen adalah hal yang sangat wajar terjadi, begitu pula pada website. 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:





```
2.     font-size: 1.5em;
3. }
```

Pastikan bahwa saat menuliskan nilai dan satuannya, tidak ada jarak (spasi).

```
1.  /* Rule tidak akan diterapkan pada target karena kesalahan penulisan nilai properti */
2.  h1 {
3.      font-size: 1.5 em;
4.  }
```

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

Berikut merupakan nilai satuan yang dapat kita manfaatkan dalam menetapkan ukuran font beserta fungsinya:

Relative unit

Satuan	Relative to	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).
ex	Font height	Satuan relatif terhadap tinggi font saat ini, satuan ini sangat jarang sekali digunakan
rem	Font size	Mirip seperti em, tetapi rem merupakan satuan relatif terhadap ukuran font dari root element.
ch	Font width	Satuan relatif terhadap lebar dari karakter “0” nol.
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.

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





in

Menetapkan nilai font berdasarkan inches

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

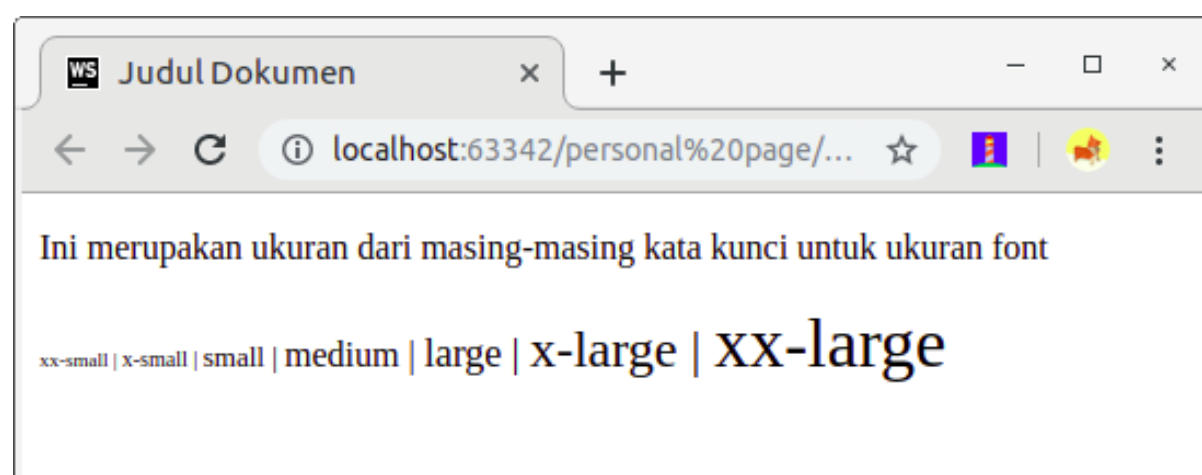
```
1. body {
2.   font-size: 16px;
3. }
4. h1 {
5.   font-size: 150%; /* 150% dari 16 = 24px */
6. }
```

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>`.



Dan yang terakhir kita juga bisa menentukan ukuran font dengan menuliskan kata kunci secara spesifik yang tersedia pada CSS. Kata kunci tersebut adalah: *xx-small*, *x-small*, *small*, *medium*, *large*, *x-large*, dan *xx-large*.

Kata kunci tersebut tidak ada kaitannya dengan pengukuran tertentu (bukan ukuran yang *absolute*) tetapi nilainya diubah secara konsisten satu sama lain.



Pada gambar di atas kita bisa lihat bahwa standarnya browser menampilkan teks dengan nilai medium. Properti *font-size* dapat diaplikasikan ke seluruh elemen yang ada di HTML dan nilainya dapat diturunkan pada elemen turunannya.

[← KEMBALI KE MATERI SEBELUMNYA](#)[LANJUTKAN KE MATERI BERIKUTNYA →](#)

DIBANTU



Dicoding Space
Jl. Batik Kumeli No.50, Sukaluyu,
Kec. Cibeunying Kaler, Kota Bandung
Jawa Barat 40123



DECODE IDEAS

Discover Potential

> [Tentang Kami](#)

[Reward](#)

[Showcase](#)



[FAQ](#)

Penghargaan

