



[Beranda](#) / [Academy](#) / [Belajar Dasar Pemrograman Web](#) / Box Dimensions

Box Dimensions

Secara standar sebuah box yang dihasilkan tiap elemen selalu cukup untuk menampung konten. Tetapi kita dapat mengatur nilai dimensi dari box tersebut dengan properti **width** dan **height**.

Cara yang paling banyak digunakan dalam menentukan dimensi kotak adalah dengan menggunakan *pixel*, persentase, atau *ems*. Secara tradisional, pixel merupakan cara yang paling populer karena kita dapat merancang dan mengontrol ukuran secara akurat.

Berbeda ketika kita menggunakan persentase, ukuran kotak akan *relative* atau menyesuaikan dari ukuran lain, seperti ukuran jendela browser atau ukuran induk yang menaunginya. Sedangkan ketika menggunakan *ems*, nilai dimensi kotak akan menyesuaikan berdasarkan ukuran teks yang ditampilkan pada konten elemen tersebut.

Pada saat ini banyak developer mulai merancang menggunakan persentase dan *ems* untuk menetapkan ukuran *box* agar dapat menyesuaikan dengan berbagai macam ukuran layar.

```
5.      .box {  
6.          height: 300px;  
7.          width: 300px;  
8.          background-color: #11C5C6;  
9.      }  
10.     p {  
11.         height: 75%;  
12.         width: 75%;  
13.         background-color: #FBDD1C;  
14.     }  
15. </style>  
16. </head>  
17. <body>  
18. <div class="box">  
19.     <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Natus officiis perspiciatis quidem ratione? Distinctio eos ex e  
20. </div>  
21. </body>  
22. </html>
```

Berkas di atas jika kita buka melalui browser akan tampak seperti ini:





Lorem ipsum dolor sit amet,
consectetur adipisicing elit. Natus
officiis perspiciatis quidem
ratione? Distinctio eos ex expedita
iusto necessitatibus velit, veritatis.
Aliquid, debitis dignissimos in
iusto magnam nulla sed tempora.

Pada contoh di atas kita dapat melihat elemen `<div>` memiliki dimensi elemen dengan lebar **300px** dan tinggi **300px**. Di dalamnya terdapat elemen `<p>` yang memiliki ukuran elemen 75% dari lebar dan tinggi elemen induknya. Dengan begitu berarti elemen `<p>` memiliki ukuran **225px** untuk panjang dan lebarnya.

Limiting Dimension

Beberapa website yang ada sekarang menampilkan *layout* yang dapat melebar dan menyempit mengikuti ukuran layar pengguna. Pada prinsip tampilan tersebut mungkin kita memerlukan sebuah limitasi ukuran yang harus ditampilkan agar konten selalu dapat ditampilkan secara proporsional. Untuk melakukannya kita memanfaatkan properti `min-width` dan `max-width`.

- `min-width` : merupakan properti yang digunakan untuk menetapkan nilai lebar minimal yang harus dimiliki elemen.
- `max-width` : merupakan properti yang digunakan untuk menetapkan nilai lebar maksimal yang harus dimiliki elemen.

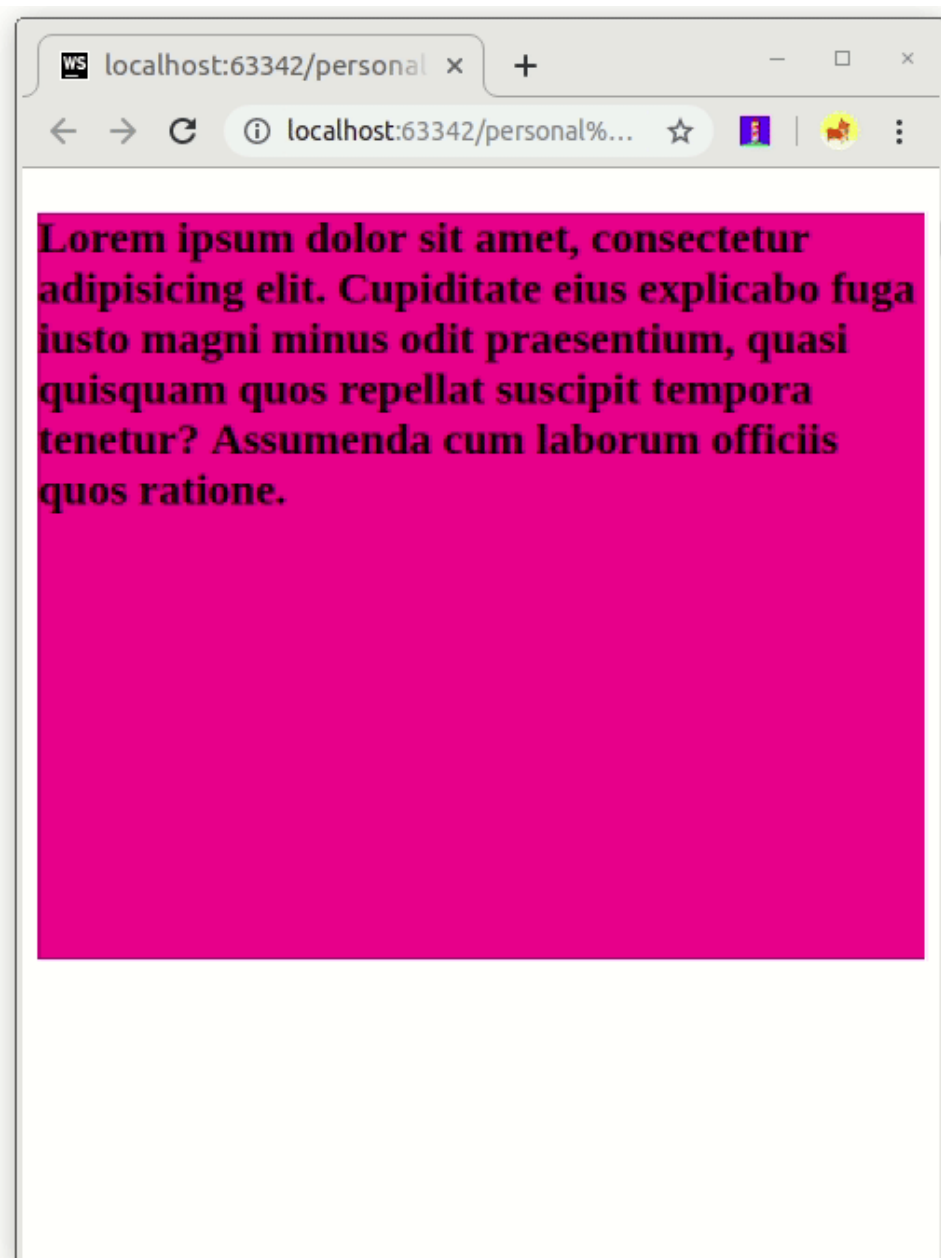
Keduanya merupakan properti yang sangat membantu untuk memastikan konten halaman dapat terbaca oleh pengguna (terutama ketika pengguna menggunakan ponsel). Misalnya, kita dapat menggunakan properti `max-width` untuk memastikan bahwa baris teks yang muncul tidak terlalu lebar.





```
7.         height: 100px;
8.         margin: 0 auto;
9.         background-color: deeppink;
10.    }
11.
12.    p {
13.        font-size: 1.5em;
14.        font-weight: bold;
15.    }
16. </style>
17. </head>
18. <body>
19. <div class="content">
20.     <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Cupiditate eius explicabo fuga iusto magni minus odit praesent:
21. </div>
22. </body>
23. </html>
```

Berkas di atas jika kita buka pada browser akan tampak seperti ini:



Dengan cara yang sama, mungkin kita juga perlu membatasi ukuran panjang. Kita bisa gunakan **min-height** dan **max-height**.

Overflowing Content



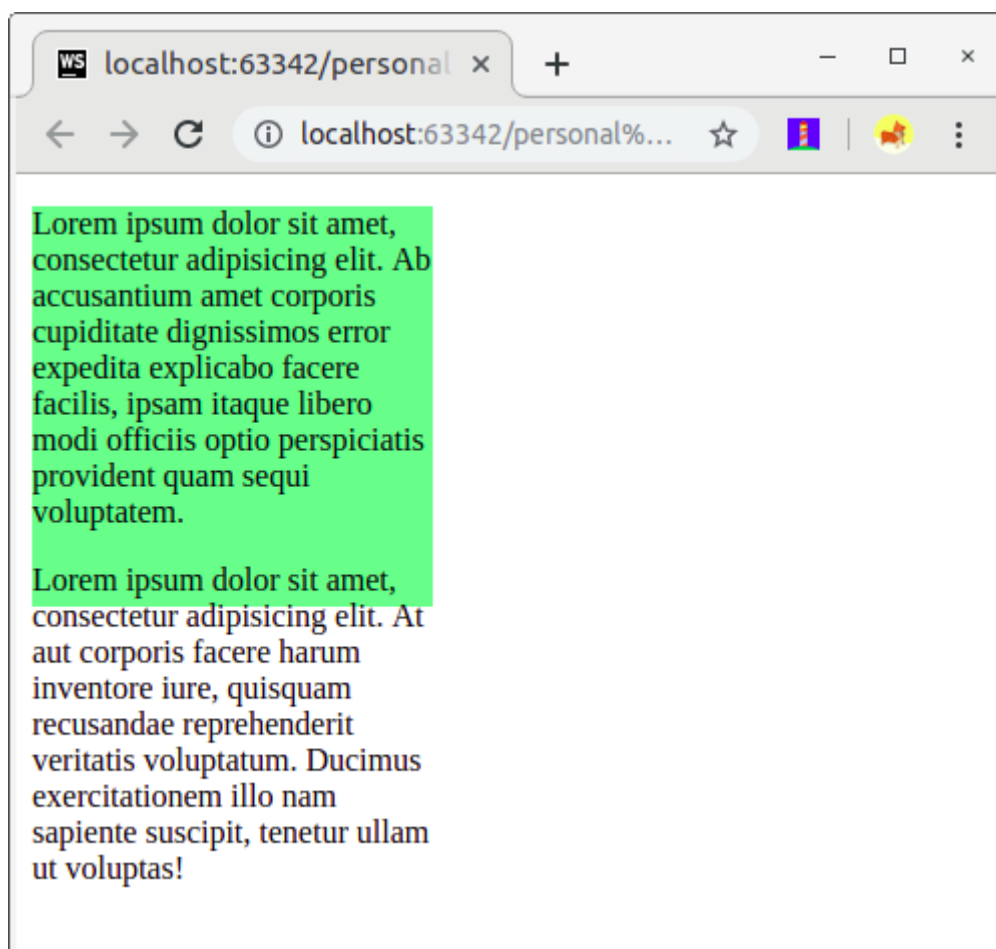
DIBANTU



adanya begitu banyak. Contohnya seperti berikut.

```
1. <!doctype html>
2. <html lang="en">
3. <head>
4.   <style>
5.     div {
6.       height: 200px;
7.       width: 200px;
8.       background-color: lightgreen;
9.     }
10.  </style>
11. </head>
12. <body>
13. <div>
14.   <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ab accusantium amet corporis cupiditate dignissimos error expedit
15.   <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. At aut corporis facere harum inventore iure, quisquam recusandae
16. </div>
17. </body>
18. </html>
```

Berkas di atas jika kita buka pada browser akan tampak seperti ini:



Untuk menangani kasus seperti ini kita bisa gunakan properti **overflow**. Properti ini dapat bernilai:

- *visible* : Merupakan nilai default pada properti ini. Konten yang tidak tertampung (overflow) akan tetap ditampilkan seperti pada standarnya.
- *hidden* : Jika terjadi overflow, maka konten yang tidak tertampung akan disembunyikan.
- *scroll* : Memunculkan scroll bar pada pinggir elemen, sehingga konten yang tidak tertampung akan ditampilkan dengan scroll bar. Jika menggunakan nilai ini, scroll bar akan tetap muncul walaupun konten tidak terjadi overflow.
- *auto* : Sama seperti scroll, bedanya jika tidak terjadi overflow, maka nilai visible yang akan diterapkan.





div { overflow: visible }

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ab accusantium amet corporis cupiditate dignissimos error expedita explicabo facere facilis, ipsam itaque libero modi officiis optio perspiciatis provident quam sequi voluptatem.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. At aut corporis facere harum inventore iure, quisquam recusandae reprehenderit veritatis voluptatum. Ducimus exercitationem illo nam sapiente suscipit, tenetur ullam ut voluptas!

div { overflow: hidden }

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ab accusantium amet corporis cupiditate dignissimos error expedita explicabo facere facilis, ipsam itaque libero modi officiis optio perspiciatis provident quam sequi voluptatem.

div { overflow: scroll }

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ab accusantium amet corporis cupiditate dignissimos error expedita explicabo facere facilis, ipsam itaque libero modi officiis optio perspiciatis provident quam sequi voluptatem.

div { overflow: auto }

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ab accusantium amet corporis cupiditate dignissimos error expedita explicabo facere facilis, ipsam itaque libero modi officiis optio perspiciatis provident quam sequi voluptatem.

div { overflow: auto }

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ab accusantium amet corporis cupiditate dignissimos error expedita explicabo facere facilis, ipsam itaque libero modi officiis optio perspiciatis provident quam sequi voluptatem.

Box-sizing

Pada CSS2 ukuran lebar dan panjang elemen mengacu pada konten elemen (*content-box*). Itu berarti ukuran elemen seluruhnya merupakan nilai panjang dan lebar yang kita spesifikasikan ditambah dengan nilai *padding* dan *border* yang diterapkan pada elemen. Hal tersebut membuat sebagian developer menjadi sulit menetapkan ukuran dimensi.

Setelah CSS3 kita dapat memilih tipe pengukuran lain dalam menentukan dimensi elemen. Dengan menggunakan properti **box-sizing** kita dapat menentukannya berdasarkan *border box*, di mana ukuran elemen sudah termasuk content, padding dan border. Dengan metode ini, hasil elemen yang ditampilkan (termasuk *padding* dan *border*) akan memiliki dimensi yang sama persis seperti yang kita tentukan.

```

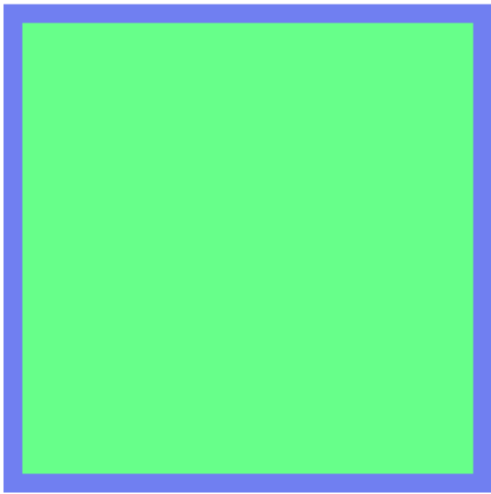
14.         box-sizing: content-box;
15.     }
16.
17.     .box {
18.         box-sizing: border-box;
19.     }
20. </style>
21. </head>
22. <body>
23. <div class="content">
24. </div>
25. <p>Elemen menerapkan <code>box-sizing: content-box;</code> Ukuran box secara keseluruhan akan menjadi 260px lebar, 260px tinggi:
26. <br>
27. <div class="box">
28. </div>
29. <p>Elemen menerapkan <code>box-sizing: border-box;</code> Ukuran box akan tetap 200px lebar, 200px tinggi meskipun menerapkan |
30. </body>
31. </html>

```

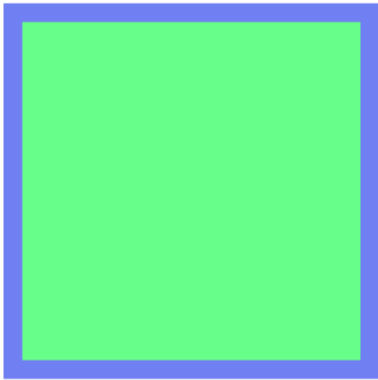
Berkas di atas jika kita buka pada browser akan tampak seperti ini:



DIBANTU



Elemen menerapkan `box-sizing: content-box`; Ukuran box secara keseluruhan akan menjadi 260px lebar, 260px tinggi; $260 = 200 + 20 + 20 + 10 + 10$



Elemen menerapkan `box-sizing: border-box`; Ukuran box akan tetap 200px lebar, 200px tinggi meskipun menerapkan padding dan border

← KEMBALI KE MATERI SEBELUMNYA

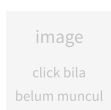
LANJUTKAN KE MATERI BERIKUTNYA →



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



Penghargaan



Decode Ideas Discover Potential

➤ [Tentang Kami](#)

[Blog](#)

[Reward](#)

[Showcase](#)

[Hubungi Kami](#)

[FAQ](#)