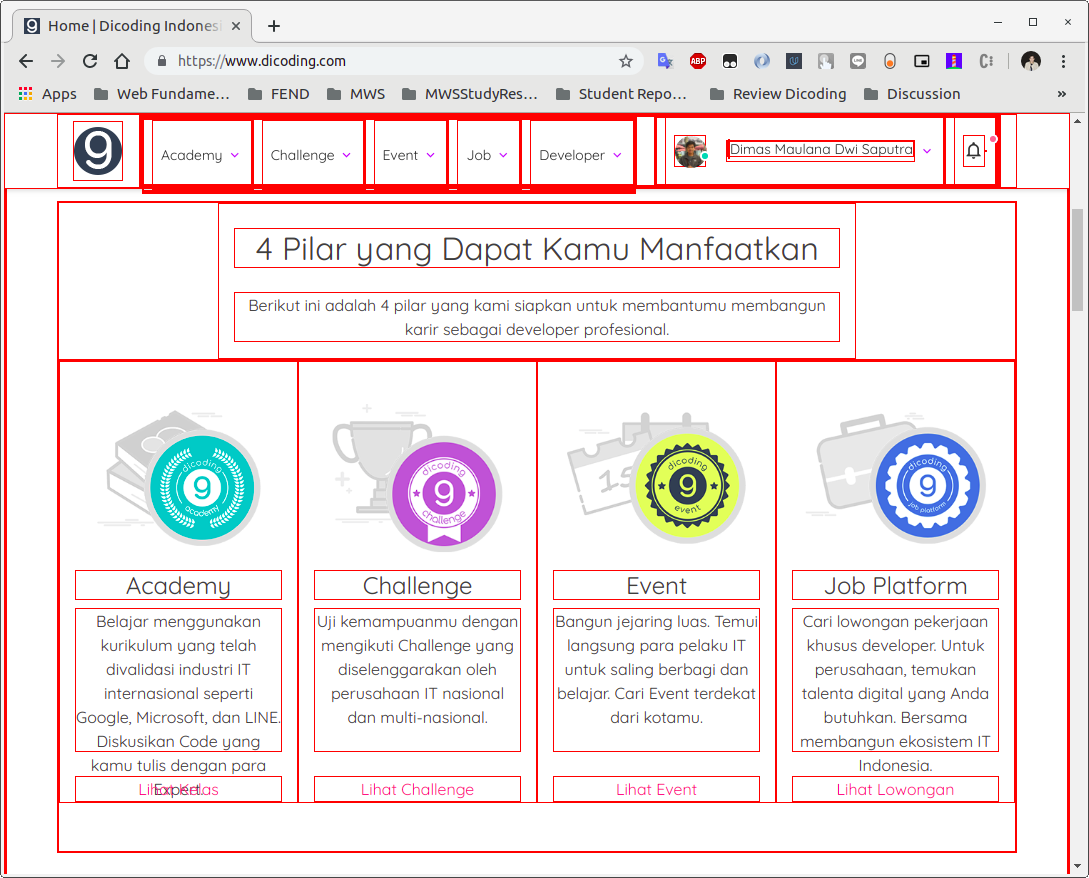
# Box Model

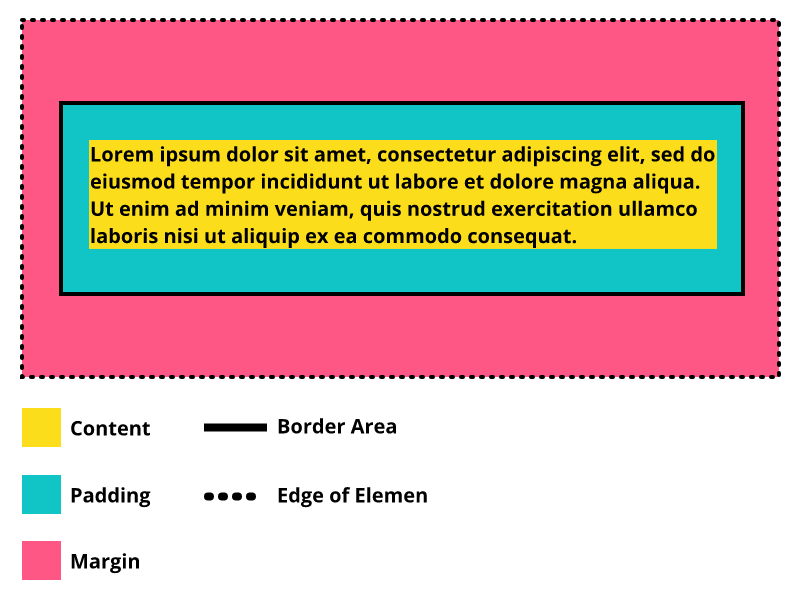
Pada pembahasan background-color disinggung bahwa setiap elemen ditampilkan seperti sebuah boxes atau kotak. Box model merupakan salah satu konsep dari CSS. Setiap elemen yang dibuat pada HTML akan menciptakan sebuah kotak untuk menampung kontennya.  Layaknya bentuk kotak pada umumnya, pasti terdapat nilai seperti width (lebar), height (tinggi), padding (tebal lapisan), border (garis tepi), dan juga jarak. Pada CSS nilai - nilai tersebut dapat kita atur.

Kita mungkin sudah sedikit paham bagaimana konsep box model ini bekerja dengan mengikuti latihan sebelumnya. Di sini kita akan membahas lebih lanjut mengenai properti apa saja yang dapat memanipulasi boxes tersebut. Tapi sebelum itu, kita pahami lebih tentang konsep box atau kotak pada elemen HTML agar kita dapat gambaran lebih tentang hal tersebut.

## Box Element



Seperti yang kita lihat pada gambar di atas, setiap elemen pada HTML baik itu block-level atau inline elemen, akan menghasilkan sebuah kotak elemen. Perhatikan ilustrasi berikut agar lebih jelasnya:



Content : Merupakan sebuah inti dari kotak elemen yang merupakan konten dari elemen itu sendiri

* Padding : Area yang menjadi jarak antara border elemen dengan konten yang ditampilkan. Ketika kita menerapkan background-color maka warna tersebut diterapkan pada area padding. Penerapan ukuran padding pada elemen merupakan opsional.
* Border : Merupakan garis terluar yang mengelilingi konten dan padding. Penerapan border pada elemen merupakan opsional.
* Margin : Berbeda dengan padding, margin merupakan area jarak diluar dari border. Pada ilustrasi diatas, margin ditunjukkan dengan warna merah muda namun pada aslinya margin selalu transparan dan penerapan margin pada elemen merupakan opsional.
* Edge of element : merupakan batas dari suatu elemen. Di dalamnya terdapat margin, border, padding dan konten. Pada ilustrasi di atas, batas elemen ditunjukkan dengan garis putus tetapi pada aslinya batas elemen selalu tidak terlihat.

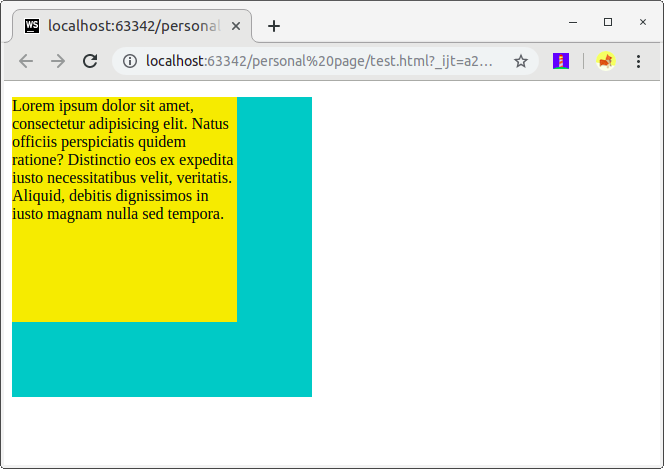
## Box Dimensions

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.



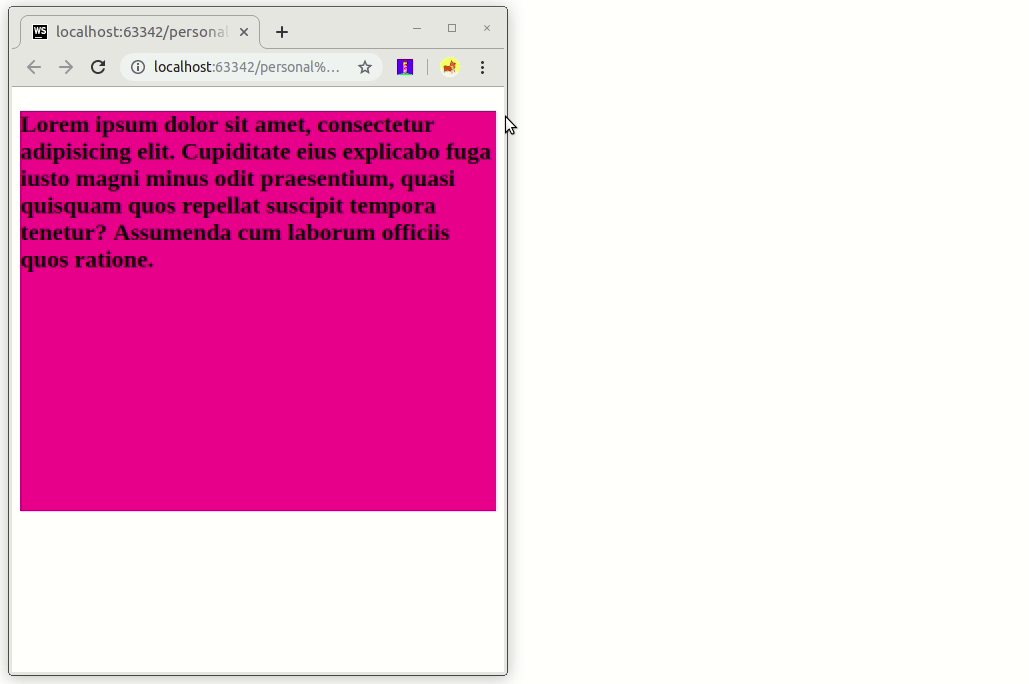
Pada contoh di atas kita dapat melihat elemen <div> memiliki dimensi elemen dengan lebar 300px dan tinggi 300px. Di dalamnya terdapat elemen <p> yang memiliki ukuran elemen 75% dari lebar dan tinggi elemen induknya. Dengan begitu berarti elemen <p> memiliki ukuran 225px untuk panjang dan lebarnya.

### 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 manfaatkan properti min-width dan max-width.

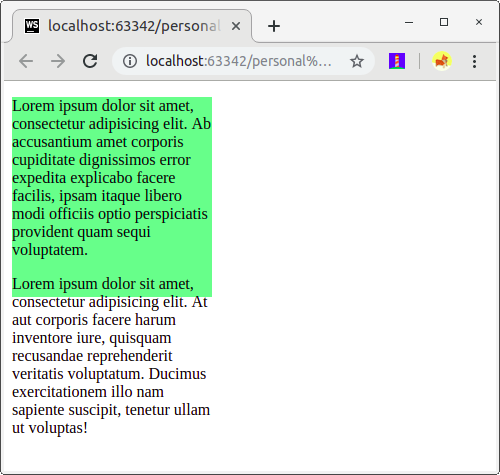
* min-width : merupakan properti yang digunakan untuk menetapkan nilai lebar minimal yang harus dimiliki elemen.
* max-width : merupakan properti yang digunakan untuk menetapkan nilai lebar maksimal yang harus dimiliki elemen.

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.



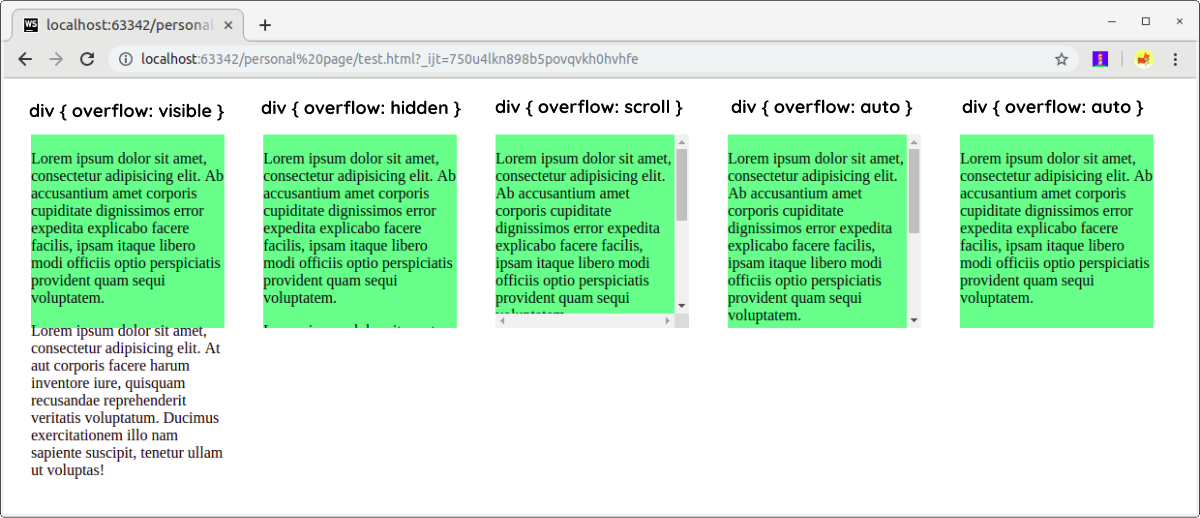
Dengan cara yang sama, mungkin kita juga perlu membatasi ukuran panjang. Kita bisa gunakan min-height dan max-height.

### Overflowing Content

Dimensi box yang dihasilkan elemen selalu cukup untuk menampung konten tetapi hal ini tidak berlaku jika kita tetapkan secara manual panjang dan lebarnya. Tak jarang terjadi overflow ketika kita menerapkan ukuran pada elemen tetapi konten di dalamnya begitu banyak. Contohnya seperti berikut:

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.



### 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.

