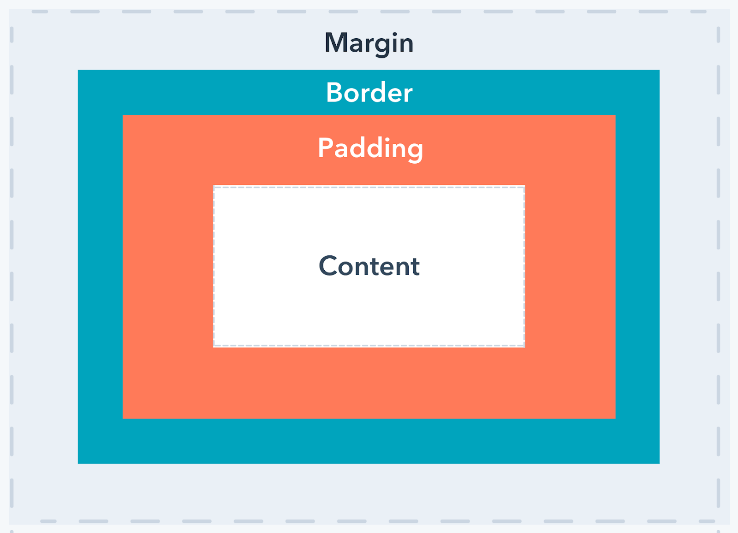
# DAY 10 : Pendalaman CSS 4

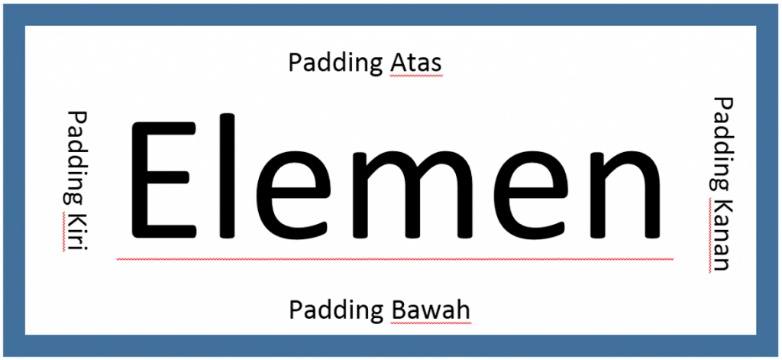
Pada pertemuan sebelumnya kita sudah berkenalan dengan banyak jenis-jenis styling, mulai dari menerapkan Tipe Font dari Google, bermain dengan warna, mengenal dan mengimplementasikan Box Element, sampai mempelajari Border. Pada materi kali ini, kita akan kembali mempelajari banyak materi baru. Kita akan mengenal prinsip dasar yang cukup penting pada CSS yaitu Padding dan Margin. Sebelum mempelajarinya coba kita mulai dengan melihat sekilas ilustrasi dibawah ini:



Pada gambar diatas, kita telah mengenal 2 bagian yakni content dan juga border. Terdapat 2 bagian lagi yang belum kita pelajari, yaitu Padding dan Margin yang baru akan kita pelajari pada materi ini. Tidak hanya itu, kita akan mempelajari beberapa styling lain yang akan membantu mempercantik proyek kita. Ready to learn? Let’s go!

## 1. Padding

Padding merupakan jarak/ruang kosong transparan yang berada di antara elemen dengan border bagian dalam. Padding banyak diterapkan pada elemen jika elemen tersebut menerapkan warna latar atau pun border. Padding memberikan sedikit ruang sehingga konten di dalam elemen dapat lebih nyaman untuk ditampilkan. Penggambaran padding seperti contoh gambar di bawah ini :



Dari gambar di atas dapat kita anggap jika tulisan elemen adalah contoh elemen2 HTML seperti <p>, <h1>, dan sebagainya, lalu garis berwarna biru adalah bordernya, nah space yang berada di antara elemen dengan border bagian dalam inilah yang dinamakan padding.

Jika belum paham mari kita coba menerapkannya langsung pada berkas HTML, perhatikan potongan kode dibawah:



Jika berkas di atas dijalankan pada browser, maka akan tampak seperti ini:

Graphical user interface, text, application

Description automatically generated

Seperti yang kita lihat, paragraf kedua jauh lebih nyaman dibaca karena terdapat jarak antara teks dan border dari kotak. Kotak pun menjadi sedikit lebar karena adanya padding.

Pixel merupakan satuan yang sering digunakan dalam menetapkan nilai properti ini (meskipun kita bisa juga menggunakan persentase atau ems). Jika menetapkan menggunakan persentase, maka nilai akan menjadi relatif pada elemen induk atau jendela browser (jika tidak memiliki induk elemen). Singkatnya, satuan ukuran yang dapat kita berikan dapat berupa px (pixel) , pt (point), cm (centimeter), atau % (persen).

CSS menyediakan konsep shorthand (cara tercepat dalam menggunakan property) pula pada Padding. Nilai dari padding ini bermula dari atas, kanan, bawah, kiri. Contoh :

|  |
| --- |
| **padding: 5px 3px 4px 2px;** |

**Menandakan :**

Atas = 5px.

Kanan = 3px.

Bawah = 4px.

Kiri = 2px.

Apabila kalian mendeklarasikannya dalam 1 nilai property, maka akan berpengaruh global, yakni pada masing – masing sisi padding. Contoh :

|  |
| --- |
| **padding: 5px;** |

Menandakan **atas, kanan, bawah, kiri** sama dengan **5px**.

Lalu apabila kita mengisikan 2 nilai seperti contoh berikut:

|  |
| --- |
| **padding: 5px;** |

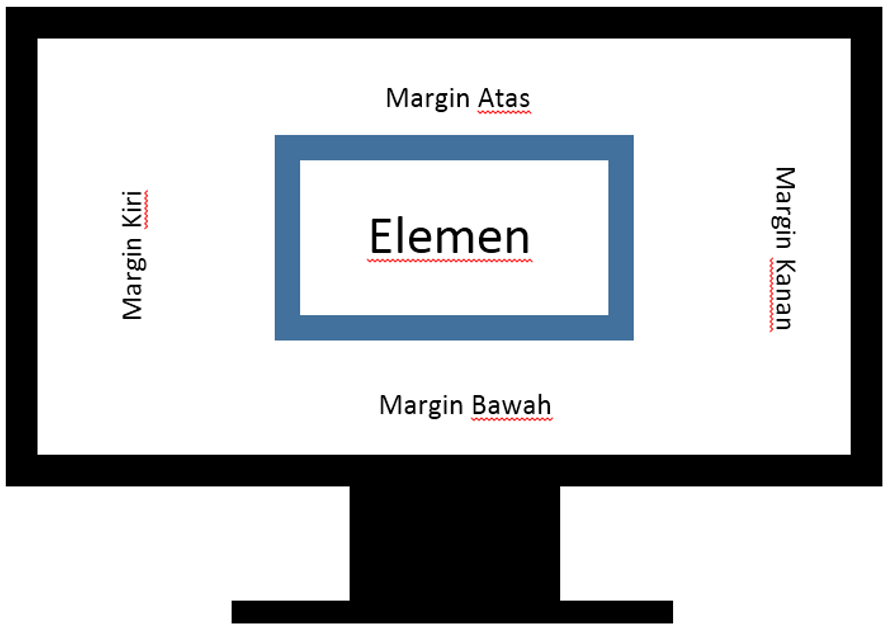
Itu berarti, memberi nilai pada posisi :

Atas + bawah = 5px.

Kanan + kiri = 2px.

## 2. Margin

Margin merupakan area kosong transparan di sekeliling elemen namun berada di luar border, dan berbeda dengan padding yang pengatuan jaraknya berada di dalam border elemen. Margin digunakan untuk menjaga elemen agar tidak bertabrakan satu sama lain atau dari tepi jendela browser. Perhatikan ilutrasi dibawah ini:



Dari gambar di atas dapat diambil kesimpulan, bahwa margin berada di luar border elemen HTML, dan gunanya untuk mengatur jarak antara elemen dengan elemen yang berada di sebelahnya. Namun anda juga dapat membayangkan sebuah padding yang letaknya di dalam border biru dari gambar di atas.

Pixel merupakan satuan yang sering digunakan dalam menetapkan nilai properti ini (meskipun kita bisa juga menggunakan persentase atau ems). Jika menetapkan menggunakan persentase, maka nilai akan menjadi relatif pada elemen induk atau jendela browser (jika tidak memiliki induk elemen). Singkatnya, satuan ukuran yang dapat kita berikan dapat berupa px (pixel) , pt (point), cm (centimeter), atau % (persen).

Kita pun juga dapat mengatur Margin secara bebas, ingin kita atur langsung ke 4 sisinya atau hanya sisi atas saja, dan atau hanya sisi kiri saja.

|  |
| --- |
| **margin: 20px;** |

**Fungsi :** Mengatur seluruh sisi Margin 🡪 Atas 20px, Bawah 20px, Kanan 20px, Kiri 20px.

|  |
| --- |
| **margin: 20px 10px;** |

**Fungsi :** Mengatur 2 sisi Margin 🡪 Atas – Bawah 20px, Kanan – Kiri 20px.

|  |
| --- |
| **margin: 10px 20px 30px;** |

**Fungsi :** Mengatur 3 sisi Margin 🡪 Atas 10px, Kanan – Kiri 20px, Bawah 30px.

|  |
| --- |
| **margin: 10px 20px 30px 40px;** |

**Fungsi :** Mengatur masing – masing sisi Margin 🡪 Atas 10px, Kanan 20px, Kiri 30px, dan Bawah 40px.

Selain itu kita juga dapat mengatur masing – masing Margin sesuai sisinya:

* **Margin-left**: nilai-property;
* **Margin-right**: nilai-property;
* **Margin-top**: nilai-property;
* **Margin-bottom**: nilai-property;

## 3. Centering Content

Bagaimana meletakkan sebuah kotak tepat pada tengah sebuah halaman atau di dalam elemen induknya? Aturlah margin kanan dan kiri dengan nilai auto. Lebih lanjut, kita juga harus menentukan lebar dari kotak tersebut (menggunakan properti width). Jika tidak, kotak akan mengambil lebar penuh pada halaman atau induk elemen.

Setelah kita menentukan lebar kotak dan mengatur margin kiri dan kanan menjadi auto, maka secara otomatis browser akan memberi jarak yang sama di setiap sisi horizontal kotak, sehingga membuat kotak berada di tengah halaman. Perhatikan contoh potongan kode dibawah:

Text

Description automatically generated

Jika berkas tersebut kita buka pada browser, maka akan tampak seperti berikut:

A screenshot of a computer

Description automatically generated

## 4. Display Roles

Kita pernah membahas mengenai inline dan block pada sub-modul Pendalaman HTML. Kita telah paham bahwa standarnya, tiap elemen HTML memiliki dua tipe yaitu block dan inline:

a. Inline Element

* Elemen HTML yang secara default tidak menambahkan baris baru ketika dibuat.
* Nilai lebar dan tinggi elemen inline sebesar konten di dalamnya, dan tidak dapat diubah.
* Margin dan padding hanya mempengaruhi elemen secara horizontal, tidak vertikal.

b. Block Element

* Elemen HTML secara default menambahkan baris baru ketika dibuat.
* Jika tidak diatur lebarnya, lebar dari elemen block akan memenuhi lebar dari browser atau elemen yang menaunginya.
* Kita dapat mengatur dimensi dari elemen block.
* Di dalam elemen block, kita dapat menyimpan tag elemen HTML lainnya.

Dengan menggunakan properti **display**, kita dapat **mengubah sebuah elemen inline menjadi block, begitupun sebaliknya**. Dalam hal ini, properti ini juga dapat digunakan untuk menyembunyikan elemen yang ditampilkan. Terdapat beberapa nilai display yang dapat kita gunakan, diantaranya:

|  |  |
| --- | --- |
| **Properti** | **Fungsi** |
| inline | Digunakan untuk mengubah elemen block berperilaku seperti elemen inline |
| block | Digunakan untuk mengubah elemen inline berperilaku seperti elemen block |
| inline-block | Membuat elemen block tidak menambahkan baris baru ketika dibuat, namun tetap mempertahankan sifat lain dari elemen block |
| none | Digunakan untuk menyembunyikan elemen dari halaman |

Properti ini banyak sekali digunakan dalam kasus dalam **pembuatan navigasi**. Biasanya navigasi dibuat menggunakan elemen list yang memiliki sifat block, sehingga item list selalu ditampilkan dalam baris baru. Dengan menggunakan properti display, kita dapat mengubah perilaku elemen list tersebut dengan menerapkan inline, sehingga item list dapat ditampilkan secara horizontal. Perhatikan contoh dibawah ini:



Jika berkas tersebut dibuka pada browser, maka akan tampak seperti berikut:

A picture containing text

Description automatically generated

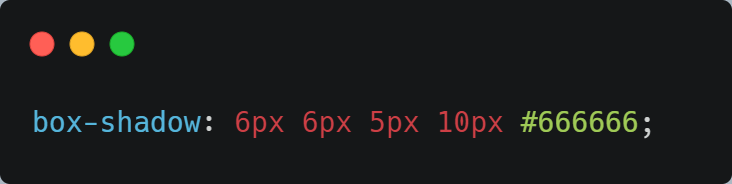
## 5. Box-Shadow

Pada materi formatting text kita sudah belajar cara menambahkan drop shadow pada teks. Pada CSS3 kita juga dapat menambahkan drop shadow di sekitar kotak elemen (tidak termasuk margin) dengan menggunakan properti box-shadow.

Nilai dan cara kerja dari properti box-shadow mirip seperti text-shadow, yaitu nilainya menentukan jarak vertikal dan horizontal, tingkat keburaman, dan warna pada bayangan. Pada box shadow, kita juga dapat menentukan tingkat sebaran (spread) bayangan. Semakin besar nilai, bayangan yang nampak akan semakin luas. Berikut contoh penerapan box shadow pada CSS:



Perhatikan kode yang menyatakan baris sebagai barikut:



Jika rules tersebut diterapkan pada box, maka akan tampak seperti ini:

Graphical user interface, text, application

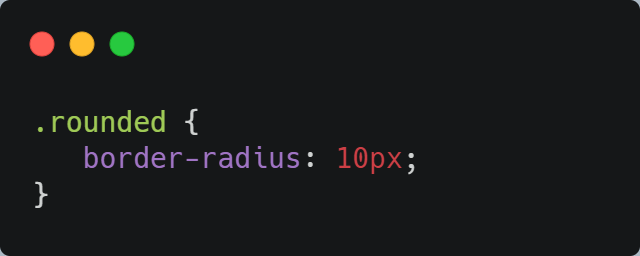
Description automatically generated

Berikut penjelasan tiap-tiap nilai dari propertinya:

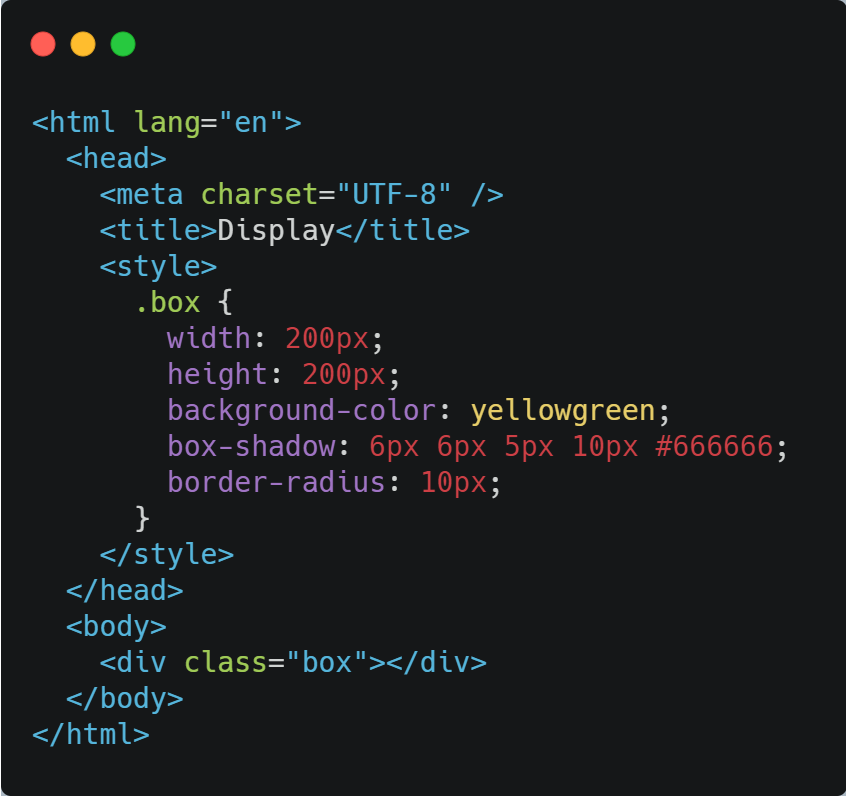
|  |  |
| --- | --- |
| **Nilai ke-** | **Fungsi** |
| Satu | menunjukkan seberapa jauh ke kiri atau kanan (horizontal) bayangan harus ditampakkan. |
| Dua | menunjukkan jarak ke atas atau ke bawah (vertical) bayangan harus ditampakkan. |
| Tiga (opsional) | menentukan tingkat keburaman yang harus diterapkan pada bayangan. |
| Empat (opsional) | menentukan tingkat sebaran (spread) bayangan. Semakin besar nilai yang ditentukan, bayangan yang nampak pun semakin luas. |
| Lima | menentukan warna yang digunakan pada bayangan. |

## 6. Rounded Corner

CSS3 memperkenalkan kemampuan untuk membuat rounded corner atau sudut bundar pada box dengan menggunakan properti **border-radius**. Nilai dari properti ini merupakan tingkat lengkungan border dalam pixel. Perhatikan penulisan *rule declaration* dibawah ini:



Apabila kita implementasikan pada berkas HTML pada contoh sebelumnya, akan menjadi sebagai berikut:

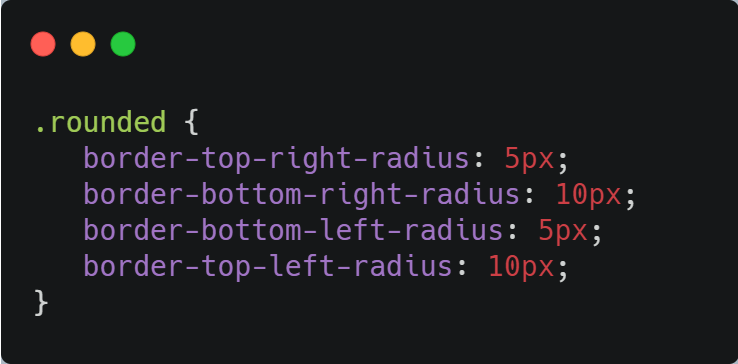


Apabila ditampilkan pada browser akan tampil sebagai berikut:

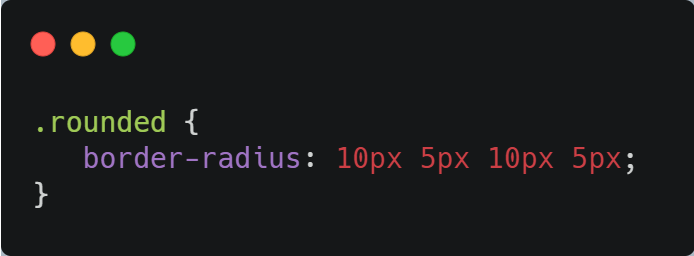
Graphical user interface, text, application

Description automatically generated

Kita bisa menetapkan nilai pada individu siku kotak dengan menggunakan properti yang terpisah, seperti ini:



Atau kita dapat menggunakan shorthand sehingga dapat menetapkan keempat nilainya dalam satu properti.



Maka tiap sisi dari border memiliki ketajaman yang berbeda seperti ini.

Graphical user interface, text, application

Description automatically generated