



Cheat Sheet.

by alamehan.github.io



CSS Layouting

Display • Dimensi • Overflow
Box Model • Box Sizing • CSS Reset
Float • Position • Z-Index

A. Display

- Setiap tag di HTML berada dalam sebuah "kotak"
- Property display pada CSS mengatur perilaku kotak tersebut
- Kotak tersebut bisa di check menggunakan background-color
- Kita dapat mengubah property display dari sebuah tag

display: inline

- Tidak menambah baris baru (saat dibuat, ada dalam sebaris)
- Lebar & tinggi kotak sebesar konten isinya, serta tidak bisa diatur
- Margin & padding hanya mempengaruhi secara horizontal (datar)
- Misal: , <input>, <button>, <a>, , <i>, , , dll
- Ada tag yang default displaynya inline, tetapi lebar & tingginya dapat diatur, seperti dan <button>

```
<a href="#">Link 1</a>
<a href="#">Link 2</a>
<span>Indonesia</span>
```

[Link 1](#) [Link 2](#) Indonesia

css → a, span {
background-color: lightblue;
}

display: inline-block

- Prilakunya sama dengan inline, hanya saja lebar & tinggi kotak dapat diatur (menggunakan property width & height)
- Tidak ada tag yang secara default displaynya inline-block, harus didefinisikan sendiri

```
<a href="#">Link 1</a>
```

[Link 1](#)

a {
display: inline-block;
width: 100px; height: 50px;
background-color: lightblue;
}

display: block

- Menambah baris baru (saat dibuat, ada pada baris baru)
- Lebar kotak secara default akan memenuhi "parent"-nya
- Lebar & tinggi kotak dapat diatur (menggunakan width & height)
- Misal: <div>, <h1>-<h6>, <p>, , , , <form>, <hr>, dll
- Didalam element block, dapat menyimpan tag dengan element inline, inline-block, maupun block lagi (block didalam block)

```
<h1>Judul</h1>
<p>Paragraf</p>
<div>Tambah</div>
```

Judul

Paragraf

Tambah

h1, p, div {
background-color: lightblue;
}

display: none

- Untuk menghilangkan/menghapus element
- Berbeda dengan property CSS visibility: hidden, yang hanya menyembunyikan element (tidak terlihat), bukan menghapus

B. Dimensi

Property width & height dapat menggunakan satuan berikut :

- px (absolut), % (relatif terhadap parent-nya)
- in, cm, mm, pt, pc (satuan di dunia nyata)
- vw, vh, rem (untuk keperluan responsive)

C. Overflow

Untuk mengatur perilaku element (konten) yang melebihi kotaknya

<div>

Saya senang belajar web programming, terutama CSS yang memiliki kemampuan untuk mempercantik website.

</div>

overflow: visible

Konten tetap tampil meski melebihi kotaknya (default)

Saya senang belajar web programming, terutama CSS yang memiliki kemampuan untuk mempercantik website.

div {
width: 150px; height: 50px;
background-color: lightblue;
}

overflow: hidden

Konten yang melebihi kotaknya akan disembunyikan

Saya senang belajar web programming, terutama CSS yang

div {
width: 150px; height: 50px;
background-color: lightblue;
overflow: hidden;

overflow: auto

Jika konten melebihi kotaknya maka munculkan scroll

Saya senang belajar web programming, terutama CSS yang

div {
width: 150px; height: 50px;
background-color: lightblue;
overflow: auto;

overflow: scroll

Muncul scroll sekalipun konten tidak melebihi kotaknya

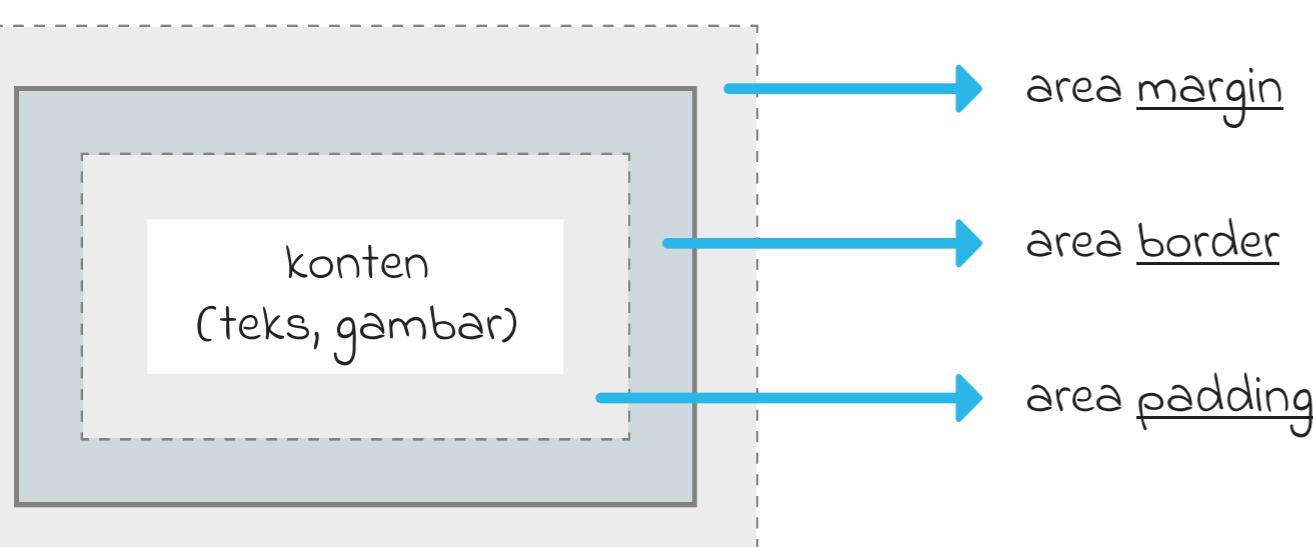
Saya senang belajar web programming,

div {
width: 150px; height: 50px;
background-color: lightblue;
overflow: scroll;

D. Box Model

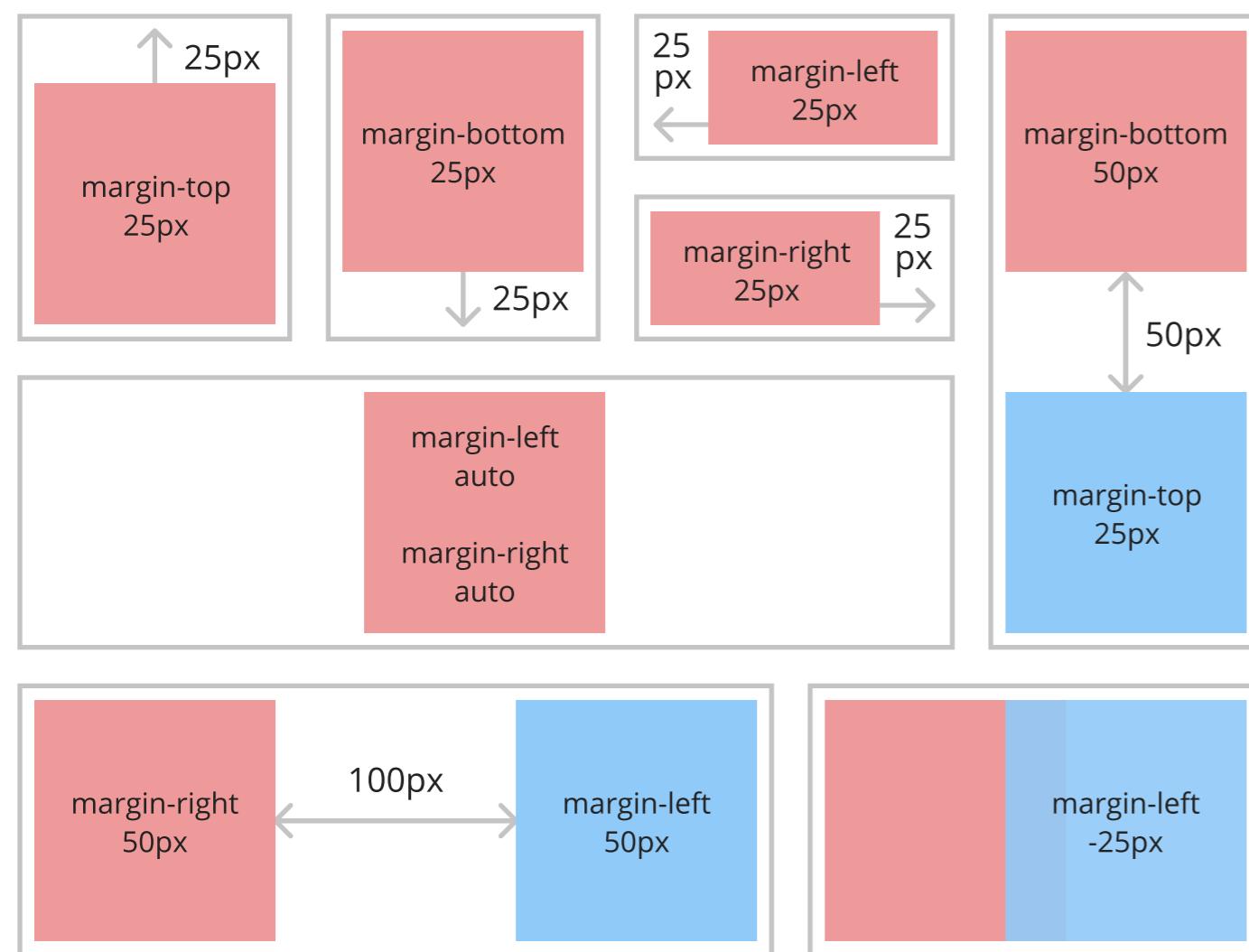
Setiap element (konten) berada di dalam sebuah box (kotak)





margin

- margin-left & margin-right (horizontal) akan saling menambahkan
- margin-top & margin-bottom (vertical) akan diambil nilai terbesar
- margin-left & margin-right (horizontal) bisa bernilai auto, hal ini membuat element berada di tengah dari parent-nya
- margin bisa bernilai negatif, artinya bergerak berlawanan arah



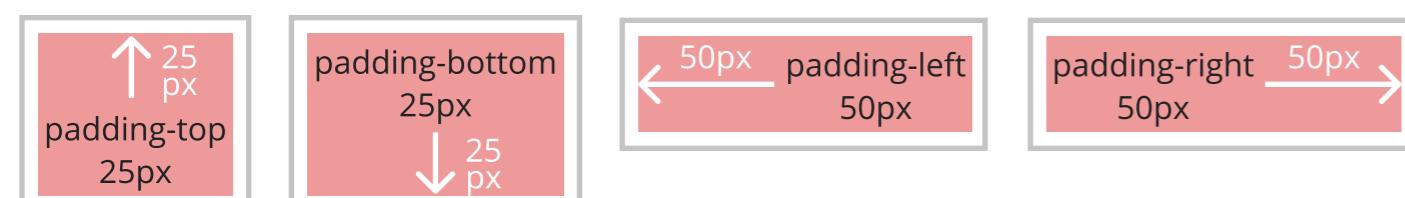
Shorthand:

margin: 50px;
 margin: 50px 25px;
 margin: 50px 25px 10px;
 margin: 50px 25px 10px 5px;

→ top, right, bottom, left : 50px
 → top & bottom : 50px, left & right : 25px
 → top : 50px, left & right : 25px, bottom : 10px
 → top : 50px, right : 25px, bottom : 10px, left : 5px

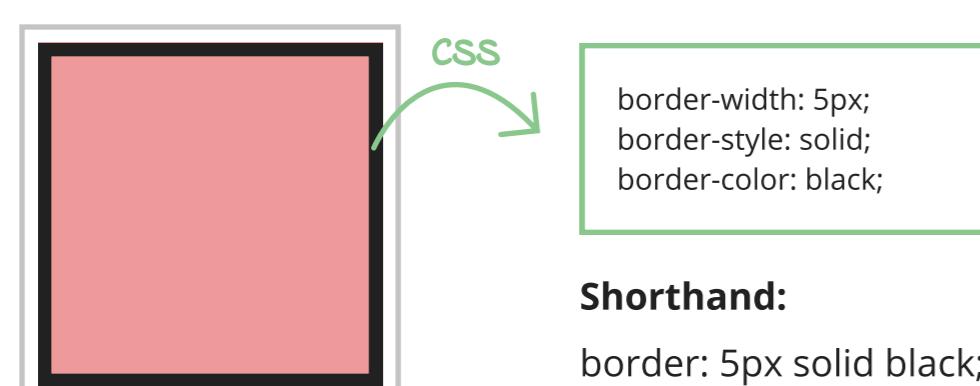
padding

- Margin adalah "space" di luar kotak, sedangkan padding adalah "space" di dalam kotak. Padding dapat mempengaruhi ukuran kotak (bertambah sesuai dengan padding yang ditambahkan).
- Penggunaan padding sama saja seperti margin, namun bedanya padding tidak bisa bernilai auto maupun negatif



border

- Border adalah batas disekeliling konten dan padding
- Sama seperti padding, border pun mempengaruhi ukuran kotak



Style yang tersedia:

- solid • dashed
- dotted • double

Property lainnya:

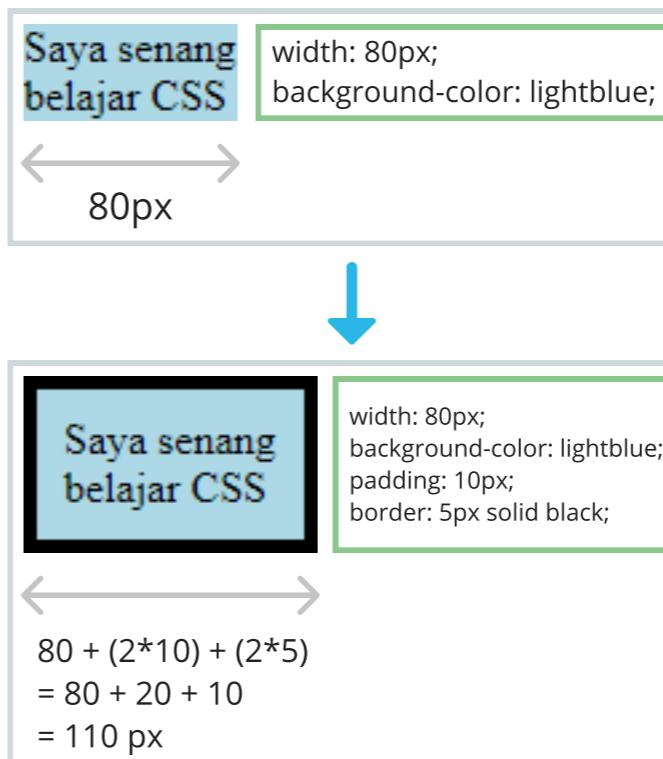
border-radius: ... px;

E. Box Sizing

- box-sizing untuk mengatur kalkulasi total dari dimensi kotak
- Biasanya saat membuat sebuah element (konten) dimensi yang kita atur hanya width-nya saja, untuk height jarang sekali kita atur (karena tinggi kotak menyesuaikan konten isinya)

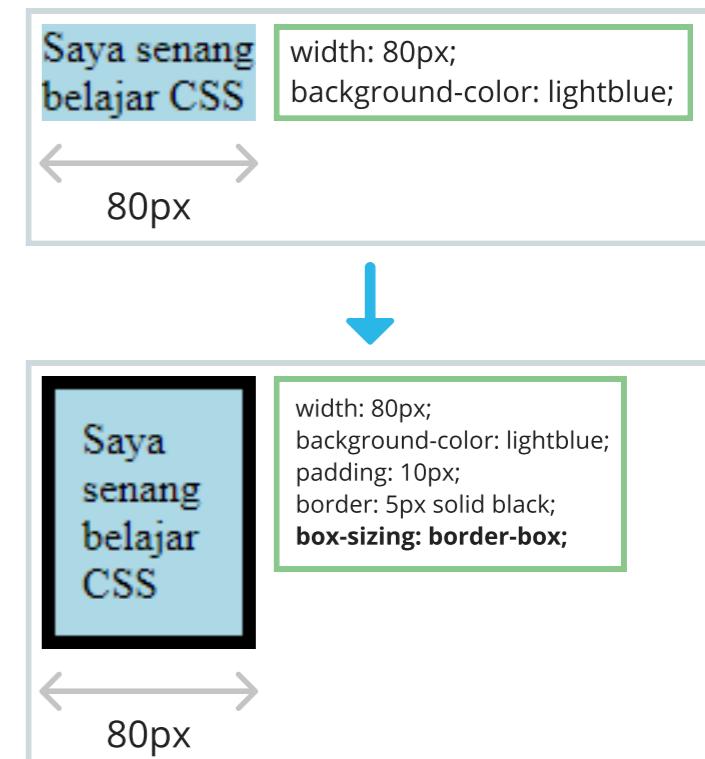
box-sizing: content-box

Padding & border mempengaruhi dimensi kotak (default)



box-sizing: border-box

Padding & border tidak mempengaruhi dimensi kotak



F. CSS Reset

- Untuk mengatur ulang nilai-nilai default dari element HTML
- Diperlukan karena nilai default di setiap browser bisa berbeda
- Biasa digunakan untuk me-reset margin, padding, border, dll
- CSS reset bisa kita definisikan sendiri atau pakai [normalize.css](#)
- Framework CSS seperti [Bootstrap](#) pun menerapkan CSS reset

Tanpa CSS Reset



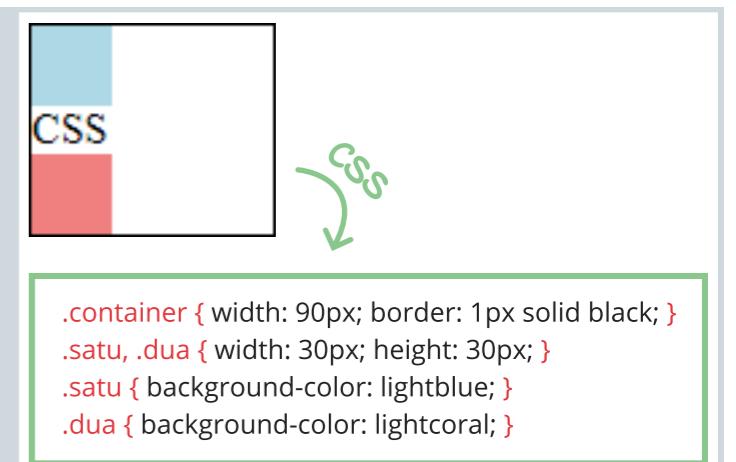
Dengan CSS Reset



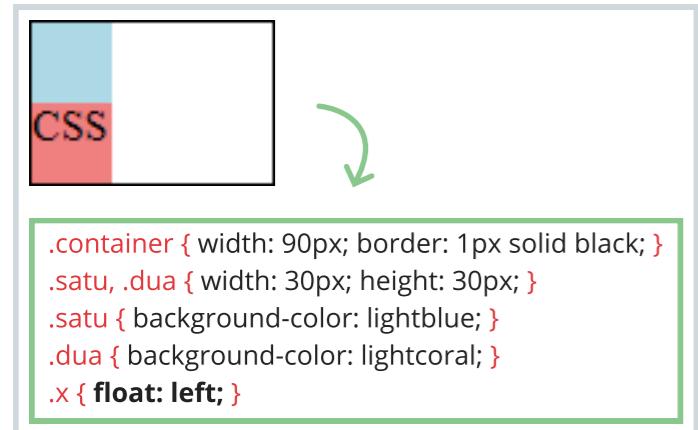
G. Float

- Float membuat sebuah element "dipaksa" untuk berada di posisi sebelah kiri atau kanan dari parent/pembungkusnya
- Float bersifat out-of flow, dimana jika sebuah element diberi property float, element tersebut seolah naik 1 layer keatas, hal ini membuat pembungkusnya menganggap element tersebut tidak ada, agar element tersebut dianggap ada dan tidak mempengaruhi element setelahnya kita harus membersihkan float dengan menggunakan property clear: both atau [micro clearfix](#)

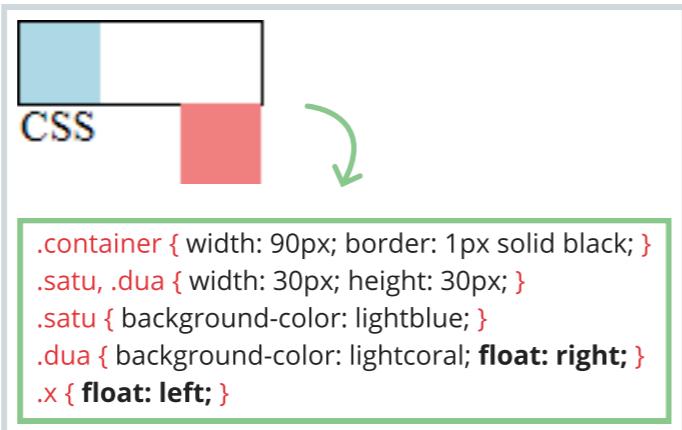
```
<div class="container">
  <div class="satu"></div>
  <div class="x">CSS</div>
  <div class="dua"></div>
</div>
```



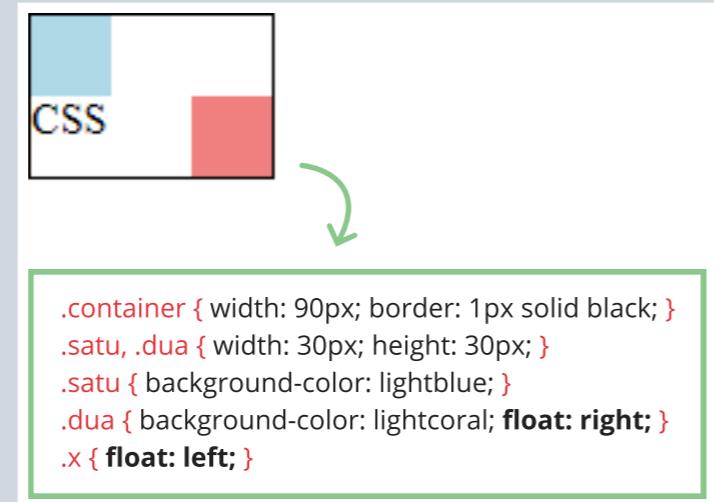
out-of flow, text "CSS" seolah naik 1 layer keatas :



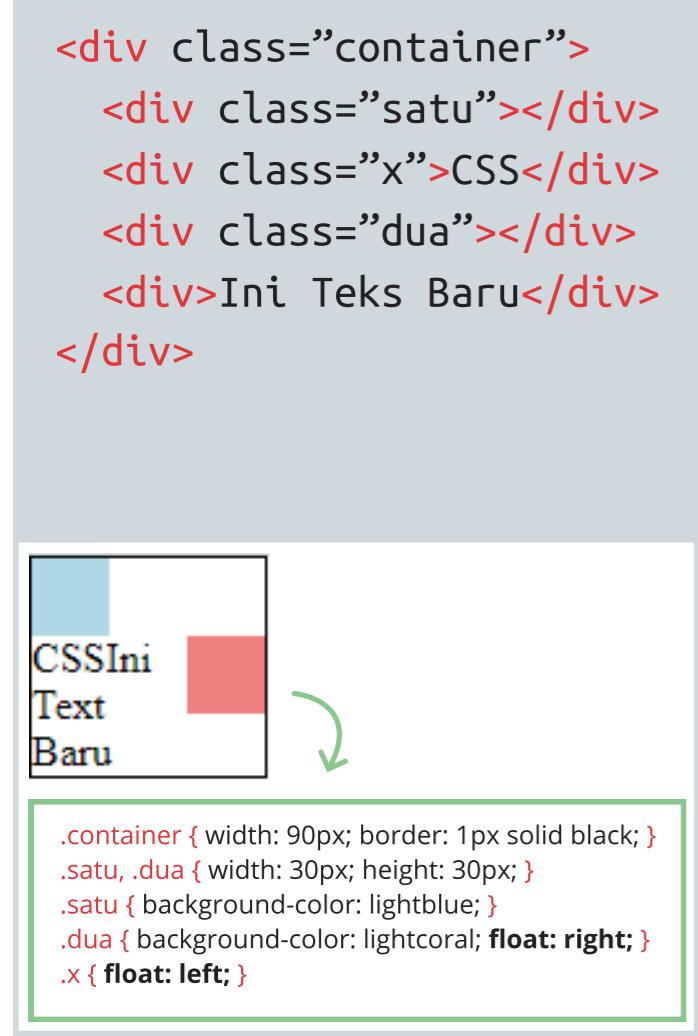
Pembungkus menganggap text "CSS" & kotak merah tidak ada :



Jika sudah selesai memakai float, bersihkan dengan property clear : both, agar text "CSS" & kotak merah dianggap ada :

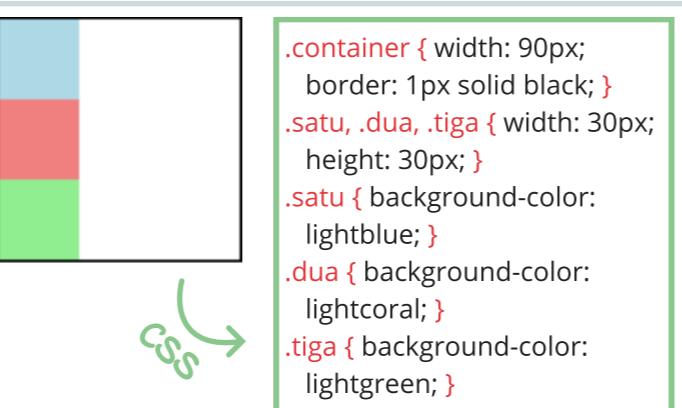
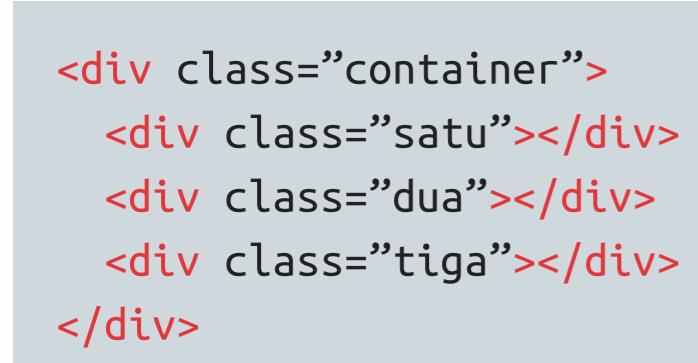


Dengan kata lain, clear : both itu "memberhentikan" float, sehingga element setelahnya tidak terpengaruh (lihat contoh dibawah) :



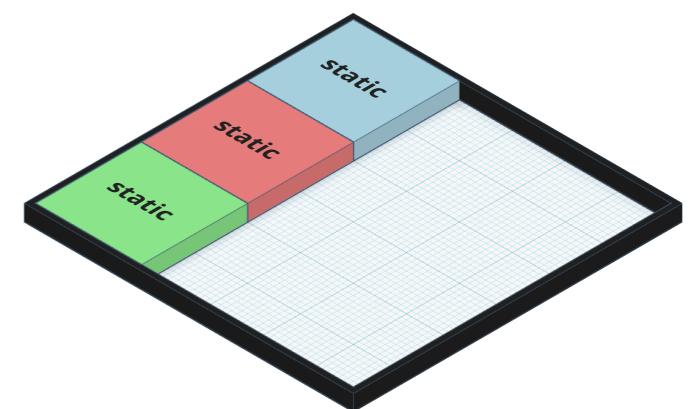
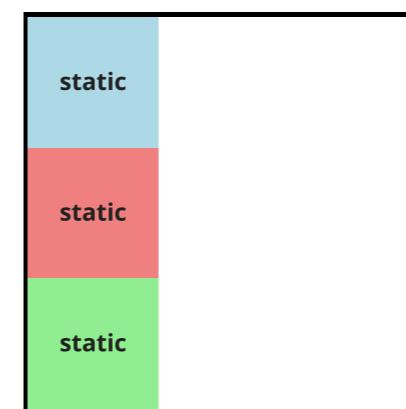
H. Position

- Saat kita menggunakan margin pada sebuah element, maka element setelah/sekitarnya akan terpengaruh (ikut bergeser)
- Property position memungkinkan kita untuk menggeser element tanpa mempengaruhi element setelah/sekitarnya, dengan cara menaikkan element tersebut 1 layer keatas (mendekati pembaca)
- Terdapat 2 kategori position yaitu static (sebagai nilai default) dan non-static (relative, absolute, fixed, dan sticky), yang mana tipe non-static inilah yang menaikkan element 1 layer keatas



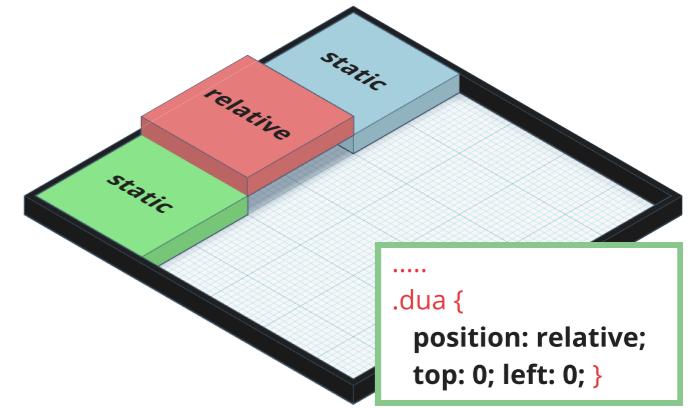
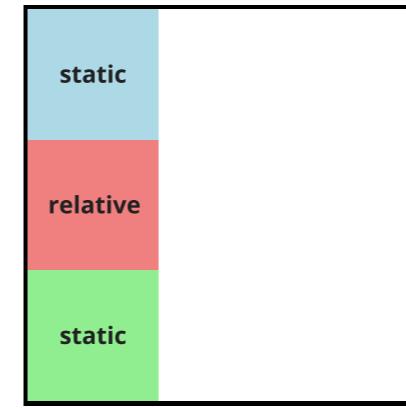
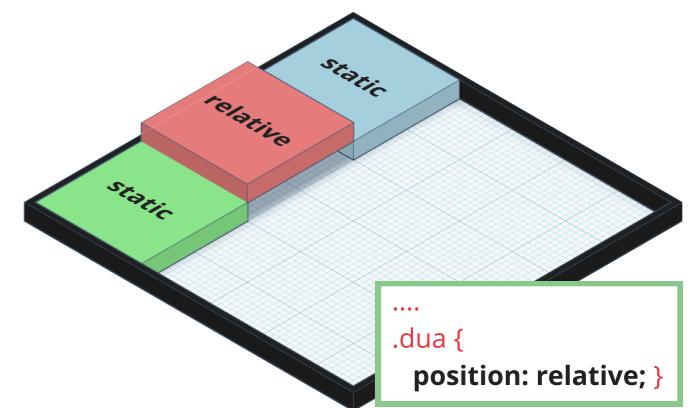
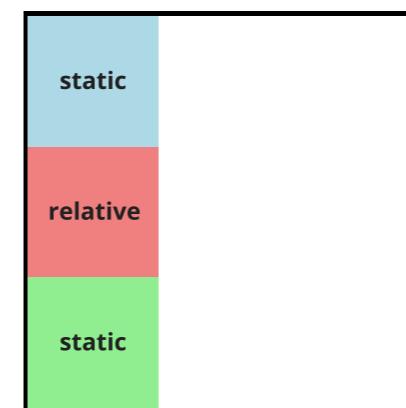
position: static

- Sebagai nilai default dari setiap element yang dibuat
- Jika element digeser, maka akan mempengaruhi element di sekitarnya, karena dengan static, element berada pada layer yang sama dengan element sekitarnya (tidak naik 1 layer keatas)



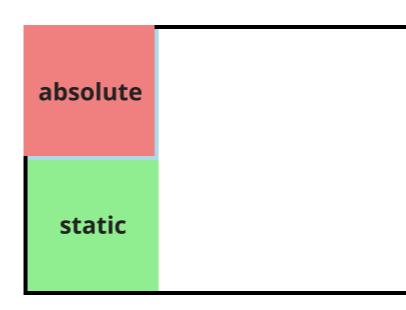
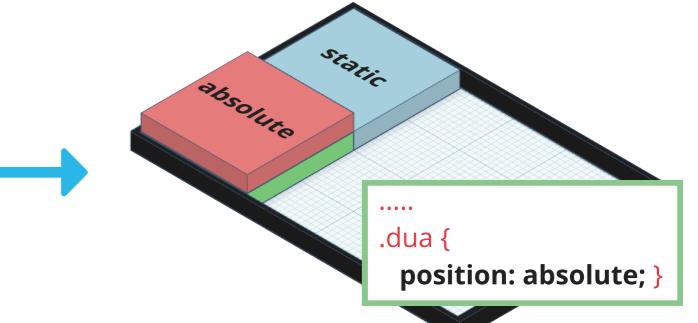
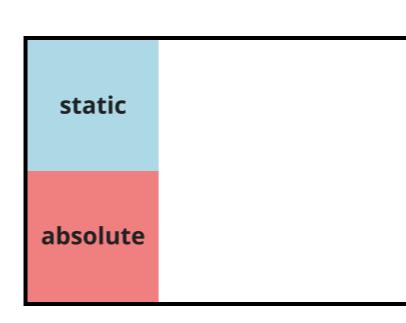
position: relative (non-static)

- Dengan position relative, element naik 1 layer keatas, sehingga dapat bergerak bebas tanpa mempengaruhi element sekitarnya
- Posisi (ruang) yang ditempati sebelumnya masih tetap ada
- Ketika element digeser menggunakan property top, right, bottom & left, element akan bergerak relatif terhadap posisi semulanya
- Jika kita memberi property top: 0 & left: 0, (yang mana digunakan sebagai posisi acuan), maka posisi element tidak berubah



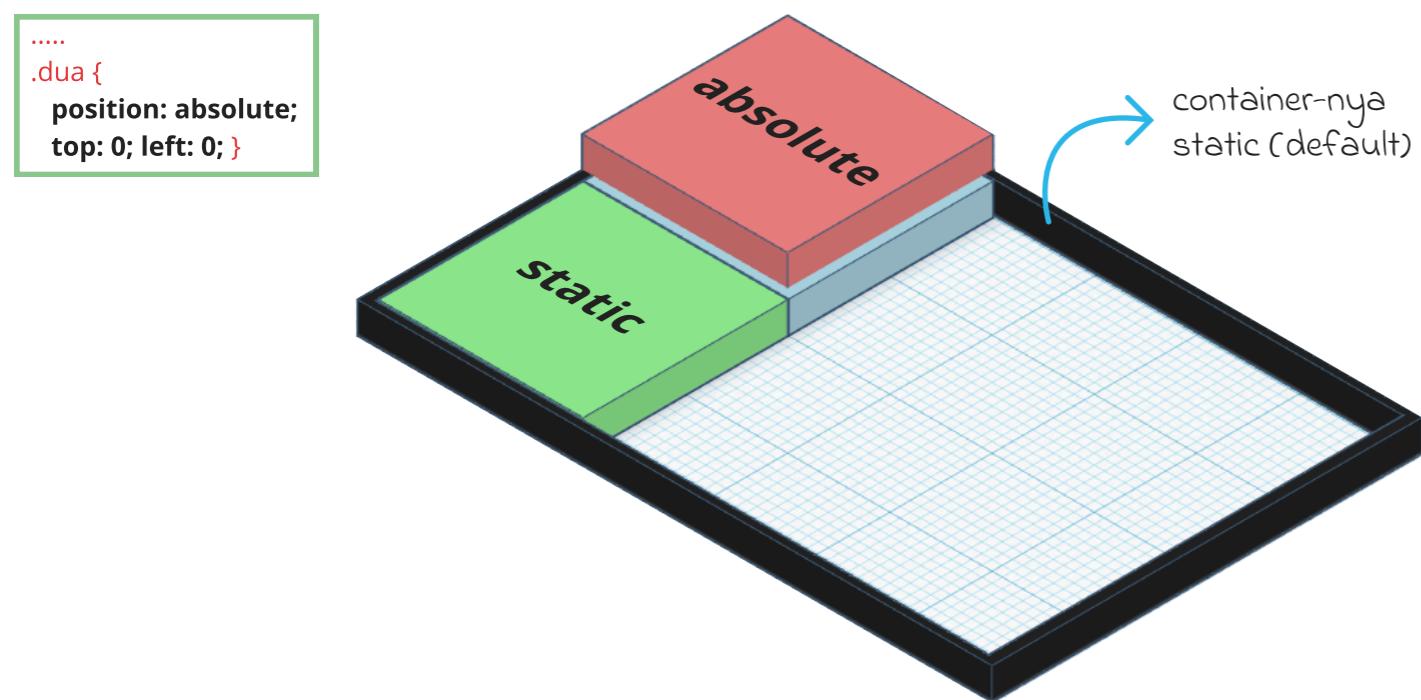
position: absolute (non-static)

- Dengan position absolute, element naik 1 layer keatas, sehingga dapat bergerak bebas tanpa mempengaruhi element sekitarnya
- Posisi (ruang) yang ditempati sebelumnya dianggap tidak ada, dan akan diisi oleh element yang berada tepat setelahnya
- Ketika element digeser menggunakan property top, right, bottom & left, element akan bergerak relatif terhadap pembungkusnya
- Jika kita memberi property top: 0 & left: 0, (posisi acuan), maka posisi element berada di sudut kiri atas pembungkusnya

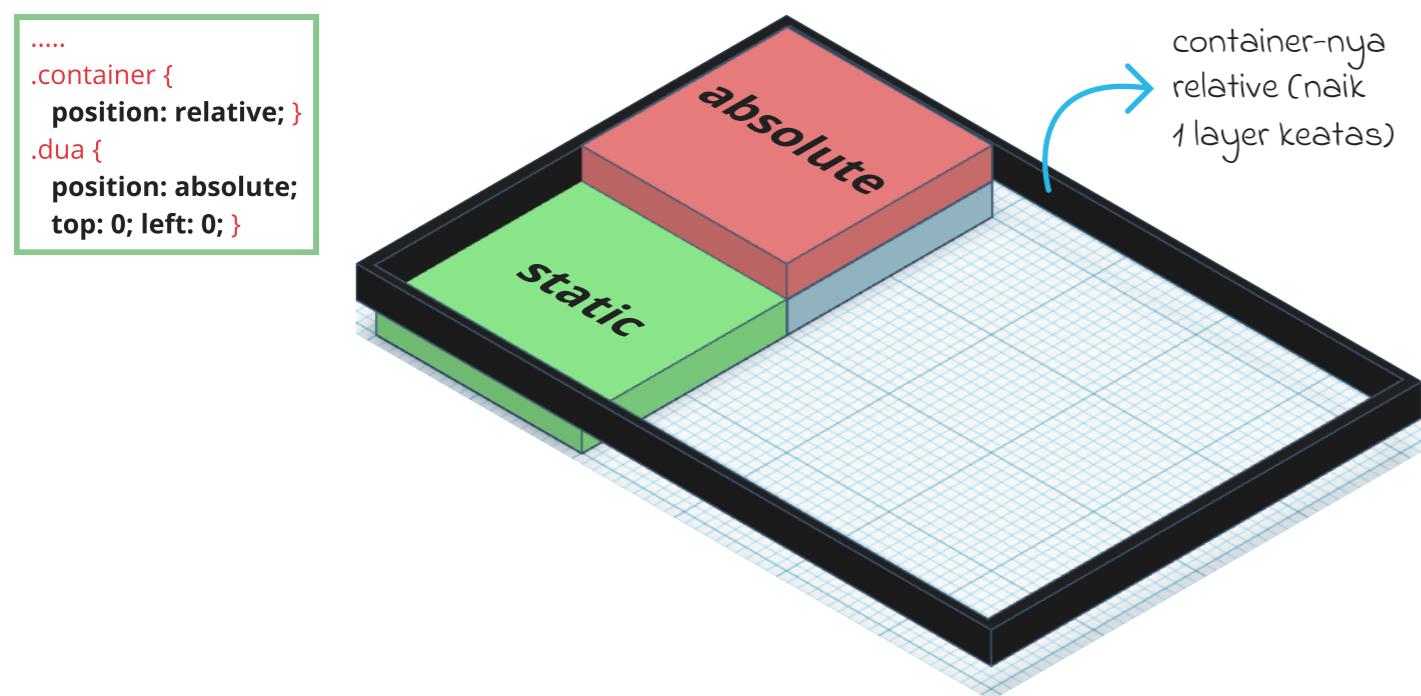


catatan: relative (container) & absolute (element)

Pada contoh position absolute sebelumnya, kita bisa lihat bahwa element kotak merah bergerak relatif terhadap pembungkusnya, tetapi bukan terhadap "container" yang telah kita definisikan, melainkan terhadap `<body>`/windows browser (sebagai pembungkus terluar), hal ini terjadi karena "container"-nya belum kita naikkan 1 layer keatas (masih berada pada layer bawah):



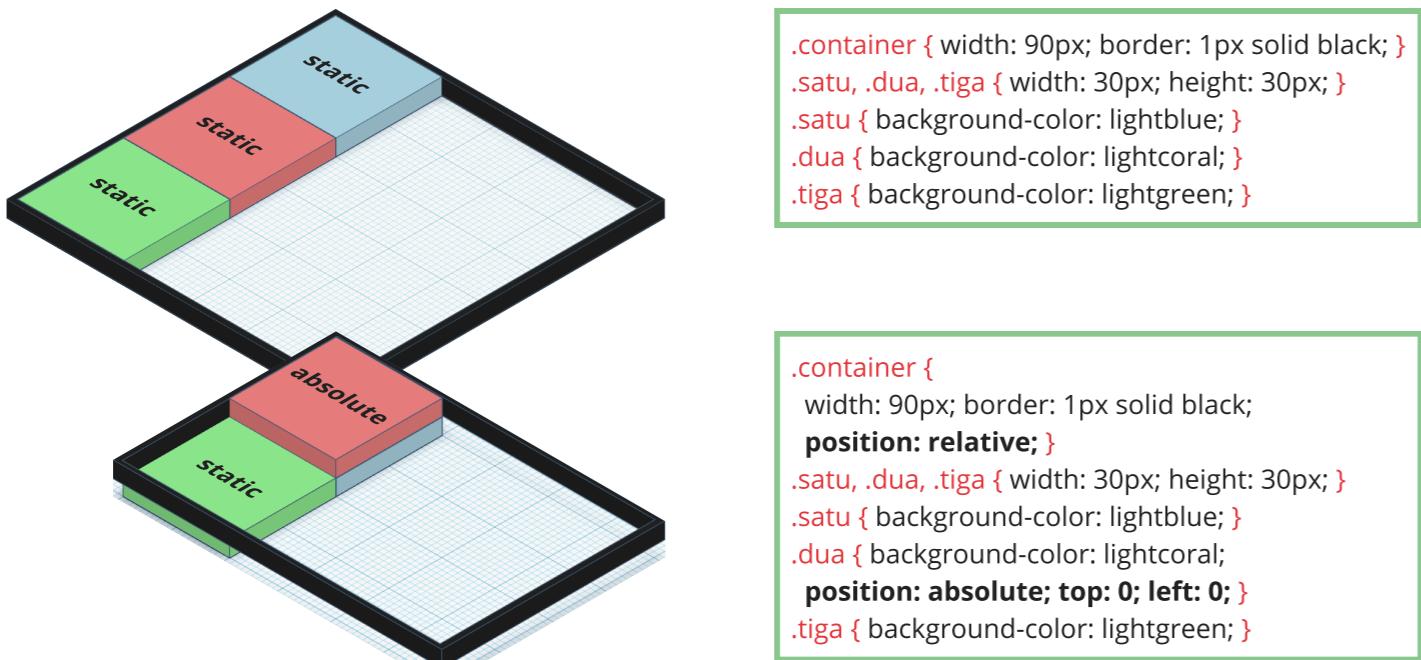
oleh karena itu, agar "container" kita naik 1 layer keatas, sehingga menjadi pembungkus bagi element kotak merah, maka kita harus memberikan position relative terhadap "container":



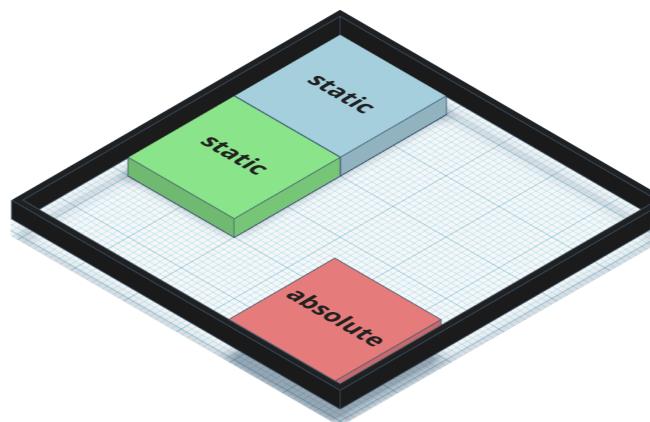
Dengan demikian element kotak merah dapat bergerak relatif terhadap "container" sebagai pembungkusnya, atau dalam kata lain position relative (dalam banyak kasus) digunakan untuk menaikkan layer pembungkusnya saja (jarang digunakan untuk menggeser element), sedangkan untuk menggeser element barulah kita gunakan position absolute

catatan: definisikan width & height untuk container

Pada contoh sebelumnya, "container" hanya kita definisikan property width-nya saja, tanpa height. Bisa dilihat bahwa saat kita menggunakan position absolute terhadap element kotak merah, tinggi dari "container" pun ikut berubah (menyesuaikan isi konten):



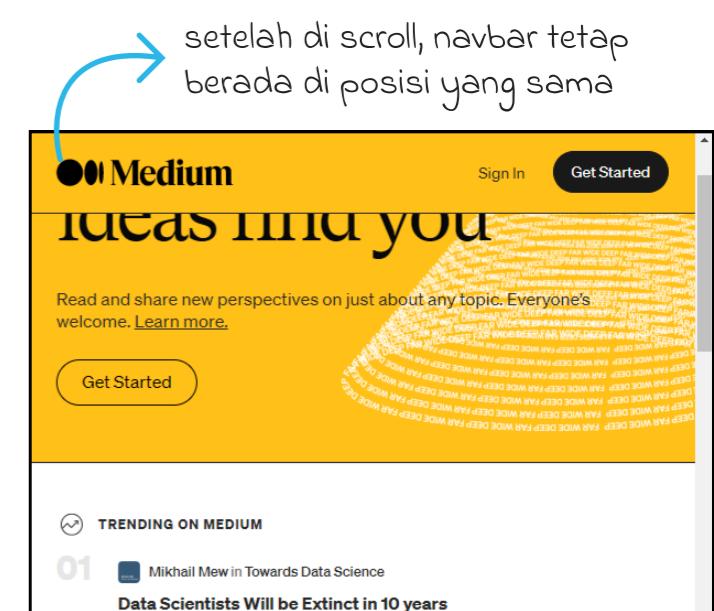
Agar dimensi/ukuran "container" tidak berubah-ubah, kita dapat memberikan property width beserta height sekaligus :



```
.container {  
width: 90px; height: 90px;  
border: 1px solid black;  
position: relative;  
}.satu, .dua, .tiga { width: 30px; height: 30px; }  
.satu { background-color: lightblue; }  
.dua { background-color: lightcoral; }  
.position: absolute; bottom: 0; right: 0; }  
.tiga { background-color: lightgreen; }
```

position: fixed (non-static)

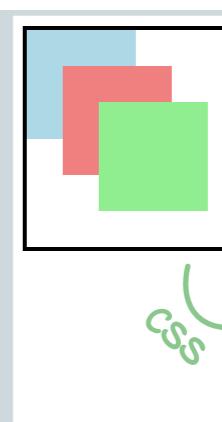
- Prilakunya sama dengan position absolute (naik 1 layer & ruang yang ditempati sebelumnya dianggap tidak ada), hanya saja ketika element digeser menggunakan property top, right, bottom & left, element akan bergerak relatif terhadap windows browser
- Jika kita memberi property top: 0 & left: 0, (posisi acuan), maka posisi element berada di sudut kiri atas windows browser
- Element dengan position fixed akan "terkunci" sekalipun halaman di scroll, contoh pemakaiannya seperti di navbar, banner iklan, dll
- Terdapat satu tipe position yang serupa dengan fixed namun dengan prilaku yang sedikit berbeda, yaitu position : sticky



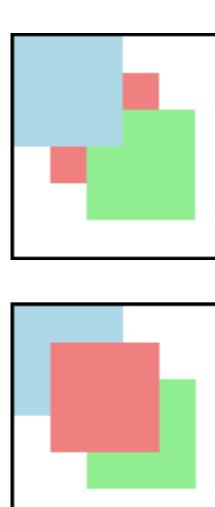
I. Z-Index

- Digunakan untuk mengatur tumpukan (layer) suatu element, apakah itu semakin atas (mendekati pembaca) atau semakin makin bawah (bernilai negatif, artinya menjauhi pembaca)
- z-index hanya berlaku untuk element ber-position non-static

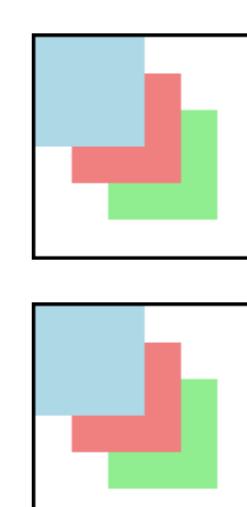
```
<div class="container">  
  <div class="satu"></div>  
  <div class="dua"></div>  
  <div class="tiga"></div>  
</div>
```



```
.container {  
width: 60px; height: 60px;  
border: 1px solid black;  
position: relative;  
}  
.satu, .dua, .tiga {  
width: 30px; height: 30px;  
position: absolute;  
}  
.satu {  
background-color: lightblue;  
top: 0; left: 0; }  
.dua {  
background-color: lightcoral;  
top: 10px; left: 10px; }  
.tiga {  
background-color: lightgreen;  
top: 20px; left: 20px; }
```

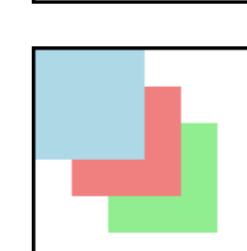


```
....  
.dua {  
z-index: 1; }
```



```
....  
.satu {  
z-index: 2; }  
.dua {  
z-index: 1; }
```

```
....  
.dua {  
z-index: -1; }
```



```
....  
.tiga {  
z-index: -2; }
```