



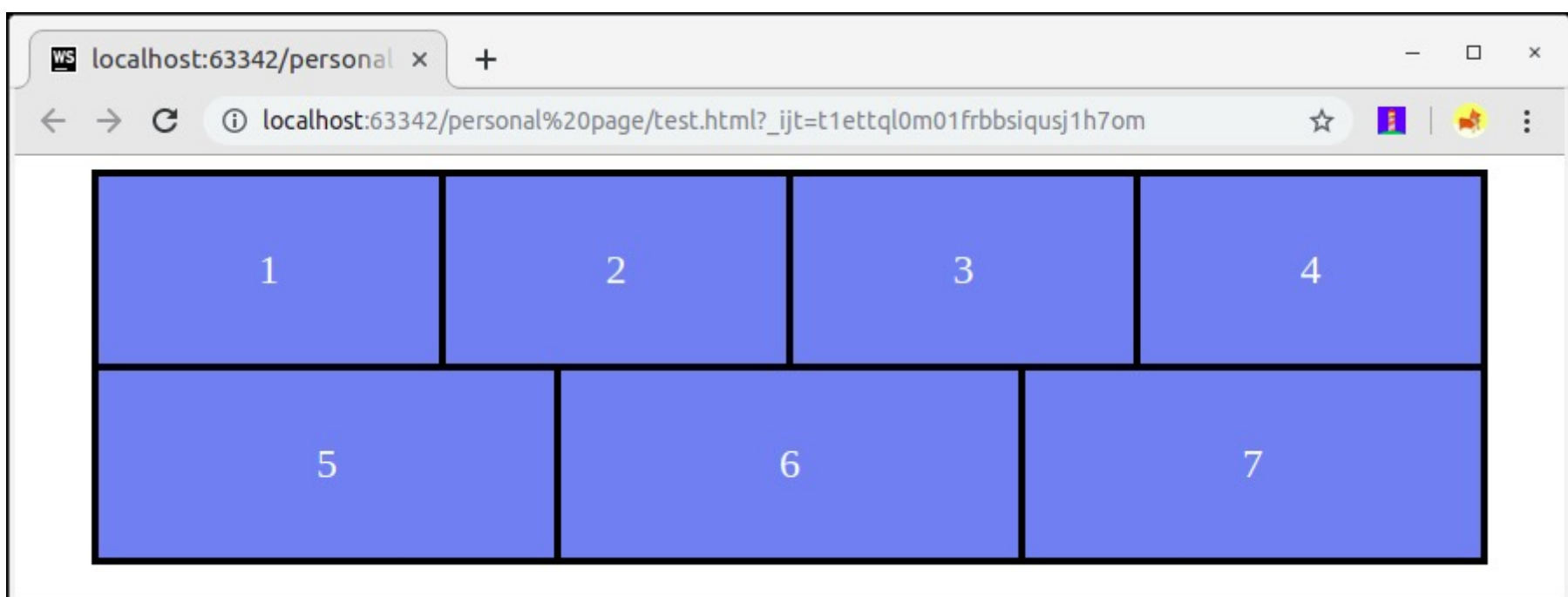
## Flex Basis

Selain menggunakan flex-grow, untuk menentukan ukuran flex item, kita bisa gunakan properti flex-basis. Properti ini mirip seperti width dalam menentukan dimensi box. Kita bisa menggunakan nilai satuan tetap seperti px, pt, pc, cm dll, atau kita juga bisa menggunakan satuan persentase %.

Flex-grow ini digunakan ketika kita ingin menetapkan ukuran awal pada sebuah flex-item. Alhasil, kita dapat mengatur ukuran dengan lebih leluasa. Flex-grow biasa digunakan ketika kita menerapkan nested flex-container dan terdapat perbedaan jumlah child pada container-nya. Untuk lebih mudah menggambarkannya, perhatikan contoh berikut:

```
1. <!doctype html>
2. <html lang="en">
3. <head>
4.   <style>
5.     * {
6.       box-sizing: border-box;
7.     }
8.     .container {
9.       width: 800px;
10.      border: 2px solid black;
11.      margin: 0 auto;
12.    }
13.
14.    .flex-column {
15.      display: flex;
16.      flex-direction: column;
17.    }
18.
19.    .flex-row {
20.      display: flex;
21.    }
```

Output dari kode di atas adalah:



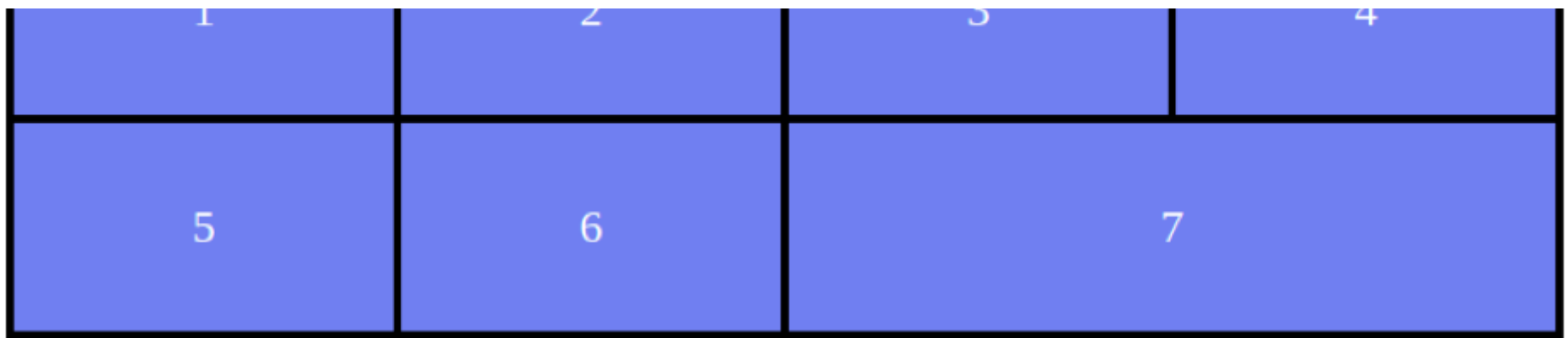
Pada contoh di atas kita memiliki satu buah flex-container column yang di dalamnya terdapat dua buah flex-container row. Kemudian flex-container row yang pertama memiliki empat buah flex-item, dan yang kedua memiliki tiga buah flex-item.

Standarnya ketika kita menggunakan nilai flex-grow: 1 pada seluruh flex-item yang ada, maka tampilan akan seperti di atas. Namun bagaimana jika kita ingin mengatur flex-item di row kedua dengan tampilan berbeda seperti di bawah ini?



DIBANTU

9

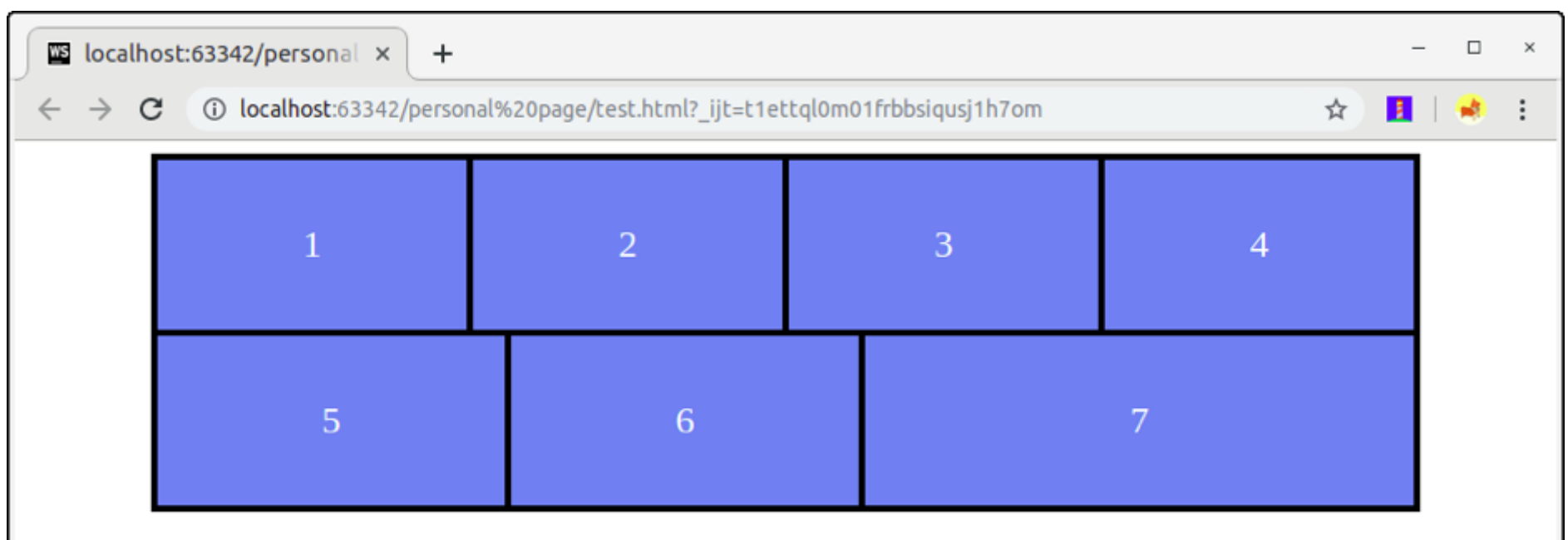


Jika kita menggunakan flex-grow, mungkin kita dapat memberikan nilai **flex-grow: 2** pada box ke tujuh.

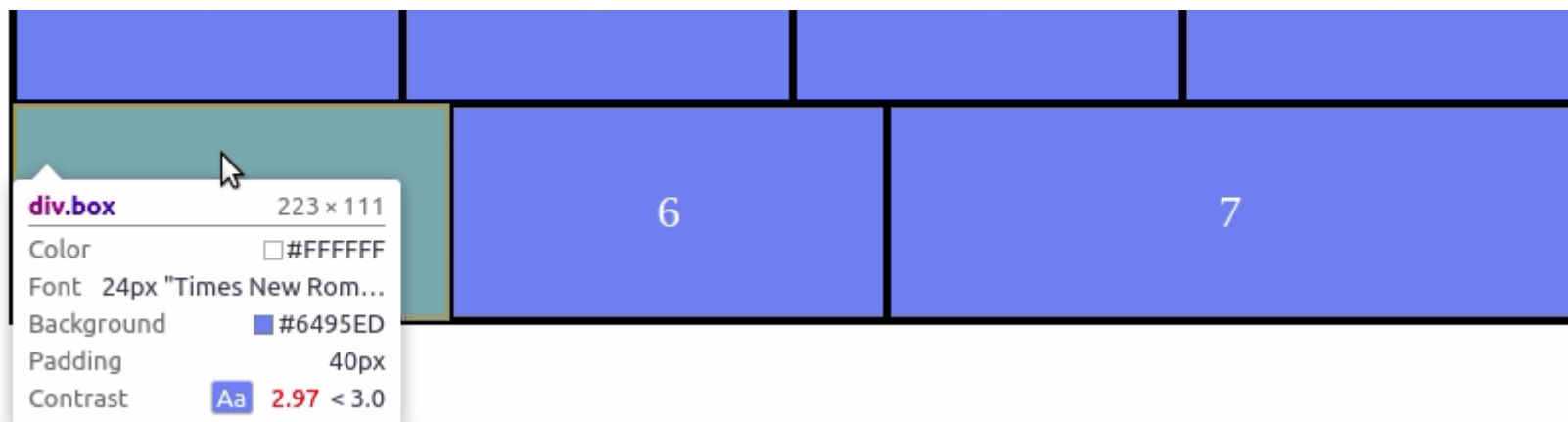
```
1. ....
2. <div class="box double">7</div>
3. ...
```

```
1. ....
2. .double {
3.   flex-grow: 2;
4. }
```

Namun hasilnya akan tampak seperti ini:



Sebabnya, properti **flex-grow** tidak menjamin elemen yang menerapkan nilai 2 akan lebih besar 2 kali lipat dari nilai 1.



Flex-grow akan mencari nilai yang sesuai yang dapat dibagi pada flex-items. Agak terdengar aneh, bukan? Nah, temukan artikel yang cukup menarik yang membahas permasalahan ini pada tautan [berikut](#).

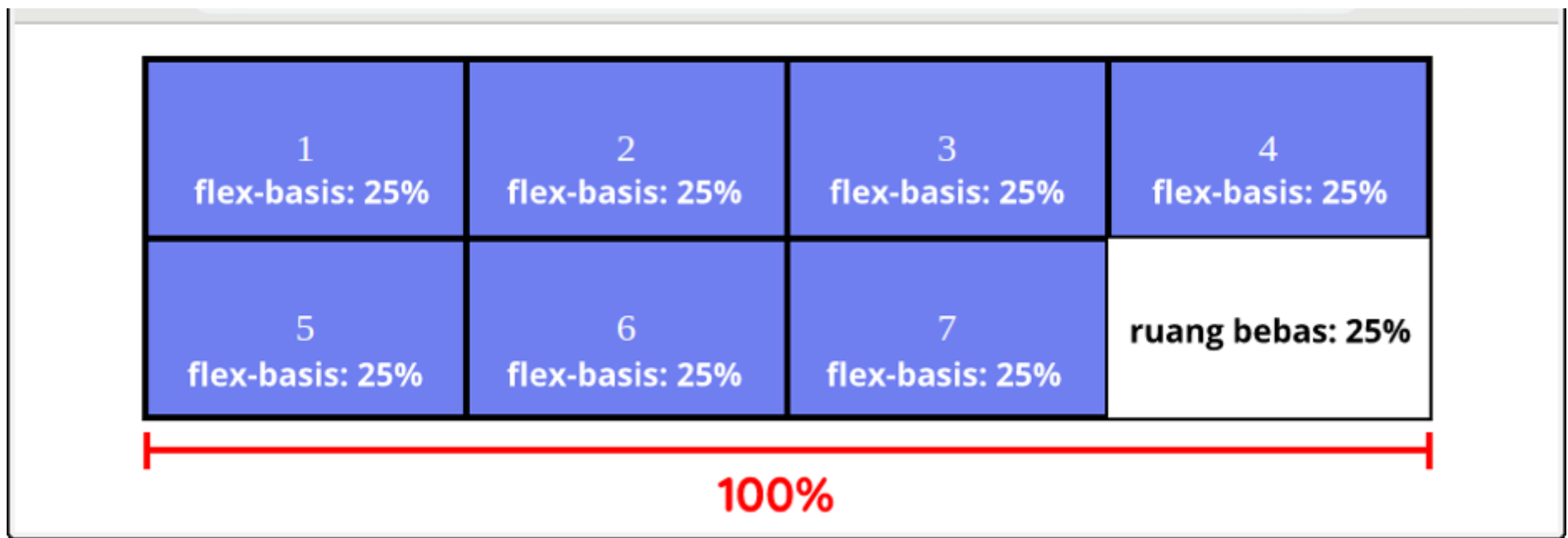
Sebenarnya bisa saja kita menggunakan flex-grow untuk mendapatkan hasil yang diinginkan namun kita harus mencari nilai yang pas secara manual. Ini tentunya akan memakan waktu lebih.

Pada kasus seperti ini, solusinya adalah menggunakan properti flex-basis. Dengan properti ini kita dapat asumsikan bahwa total ruang kosong pada flex-container adalah 100%. Jika dibagi rata terhadap empat buah flex-item, maka tiap itemnya harus memiliki nilai 25%.



```
1. <!doctype html>
2. <html lang="en">
3. <head>
4.   <style>
5.     * {
6.       box-sizing: border-box;
7.     }
8.
9.     .container {
10.      width: 800px;
11.      border: 2px solid black;
12.      margin: 0 auto;
13.    }
14.
15.    .flex-column {
16.      display: flex;
17.      flex-direction: column;
18.    }
```

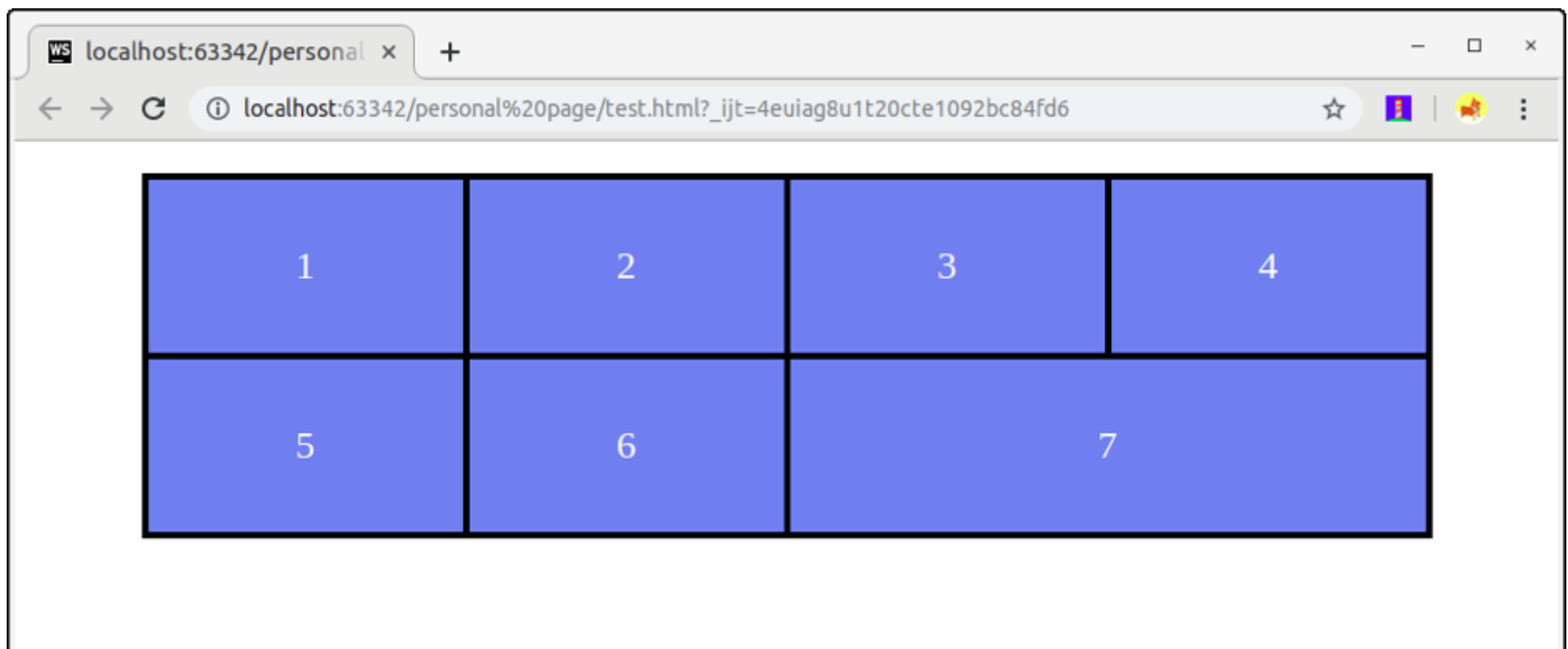
Output kode di atas adalah:



Maka seluruh flex-items akan memiliki ukuran yang sama. Tetapi pada flex-container baris kedua terdapat 25% ruang bebas, karena jumlah flex-itemnya hanya tiga. Kita bisa menggunakan ruang kosong tersebut untuk membuat box tujuh lebih besar dua kali lipat dengan menggunakan nilai **flex-basis: 50%**;

```
1. ....
2. <div class="box double">7</div>
3. ...
```

```
1. .....
2. .double {
3.   flex-basis: 50%;
4. }
```

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



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

> [Tentang Kami](#)

[Showcase](#)



Penghargaan

