



CSS Box Model

Semua elemen HTML dapat dianggap sebagai kotak. Dalam CSS, istilah “Box Model” digunakan ketika berbicara tentang desain dan tata letak.

Model kotak CSS pada dasarnya adalah kotak yang membungkus setiap elemen HTML. Ini terdiri dari: margin, border, padding, dan konten yang sebenarnya.



Penggunaan Div digunakan untuk membuat Box Model. Sebagai contoh tuliskan deklarasi *style* berikut pada file HTML dengan nama **box_model.html**.

```
<html>
<head>
<title>CSS Box Model</title>
<style type="text/css">
div {
    width: 300px;
    border: 25px solid green;
```

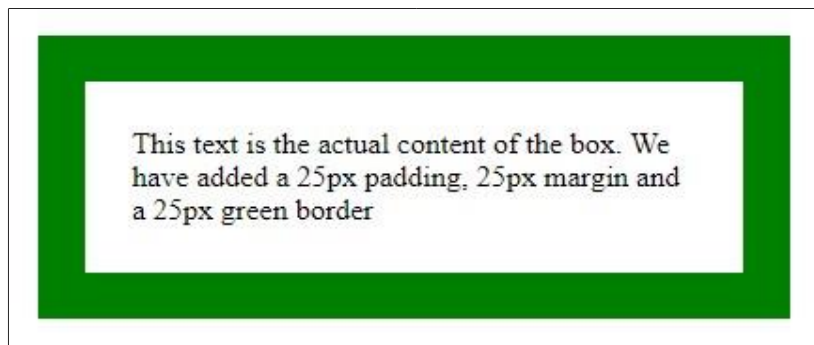
```

padding: 25px;
margin: 25px;
}
</style>
</head>

<body>
  <div>This text is the actual content of the box. We have added a 25px
padding, 25px margin and a 25px green border </div>
</body>
</html>

```

Jalankan file **box_model.html**, hasil tampilan dapat dilihat pada gambar di bawah ini.



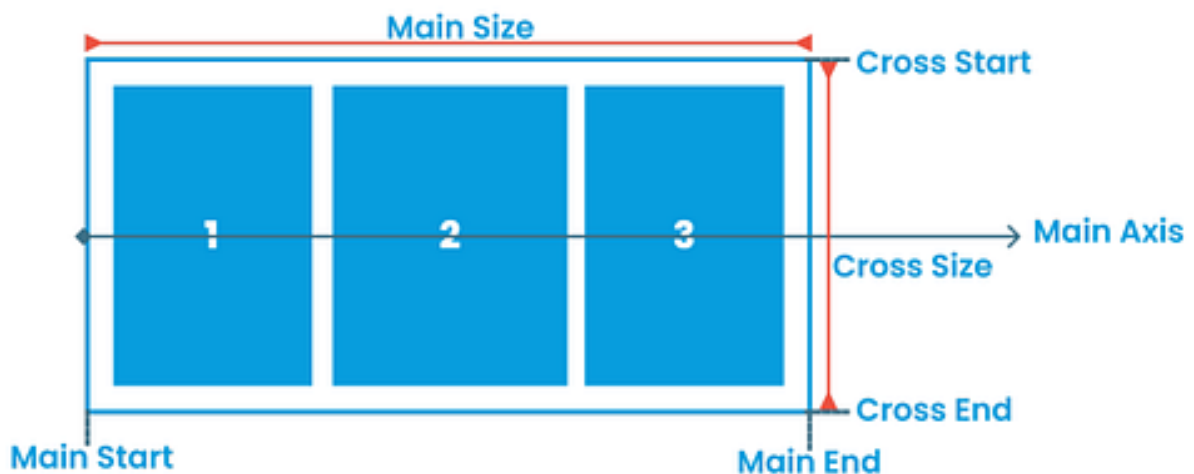
1. Pengenalan Flexbox

Flexbox adalah mode layout 1 dimensi yang terdapat di CSS3 yang digunakan untuk mengatur elemen di suatu halaman Web. Flexbox ini dapat mengatur jarak dan penjumlahan antar item dalam sebuah container. Dalam Flexbox pengertian 1 dimensi yaitu hanya dapat mengatur antara baris atau kolom, jadi tidak bisa kedua nya sekaligus.

Untuk Apa Flexbox?

Tujuan dari flexbox yaitu memberikan container kemampuan untuk mengatur panjang, lebar, dan posisi item-item yang berada di dalamnya agar memaksimalkan ruang yang ada. Pengaturan ini sangat penting bagi seorang frontend developer untuk membuat sebuah website yang nyaman dilihat di berbagai device dengan berbagai macam resolusi.

2. Konsep Dasar Flexbox



3. Istilah Pada Flexbox

- **Main Axis** : Sumbu dari sebuah Container yang menentukan urutan dari penempatan item secara horizontal.
- **Main-start | Main-end**: Mulai dan berakhirnya item yang disimpan di dalam container.
- **Main size** : Ukuran (Width / Height) dari container yang akan membuat dimensi dari item nya relatif terhadap main size.

Press enter or click to view image in full size

```
PENTING !  
untuk membuat flexbox harus menggunakan properties display : flex  
.container {  
    display: flex;  
}
```

4. Properti pada Flexbox

- **Flex — Direction** :

Flex Direction Berfungsi untuk menentukan sumbu arah items. Value pada Property Flex Direction :

- **flex-direction : row**; Nilai Item flex menumpuk secara Horizontal (dari kiri ke kanan).
- **flex-direction : column**; Nilai Item flex menumpuk secara Vertikal (dari atas ke bawah).
- **flex-direction: column-reverse**; Nilai item flex menumpuk secara Vertikal (tetapi dari bawah ke atas).
- **flex-direction: row-reverse**; Nilai Item flex menumpuk secara Horizontal (dari kanan ke kiri)

1. Implementasi Pada Property Flex-direction :

HTML :

```
<div class="container">  
  <div class="box">1</div>  
  <div class="box">2</div>  
  <div class="box">3</div>  
  <div class="box">4</div>  
</div>
```

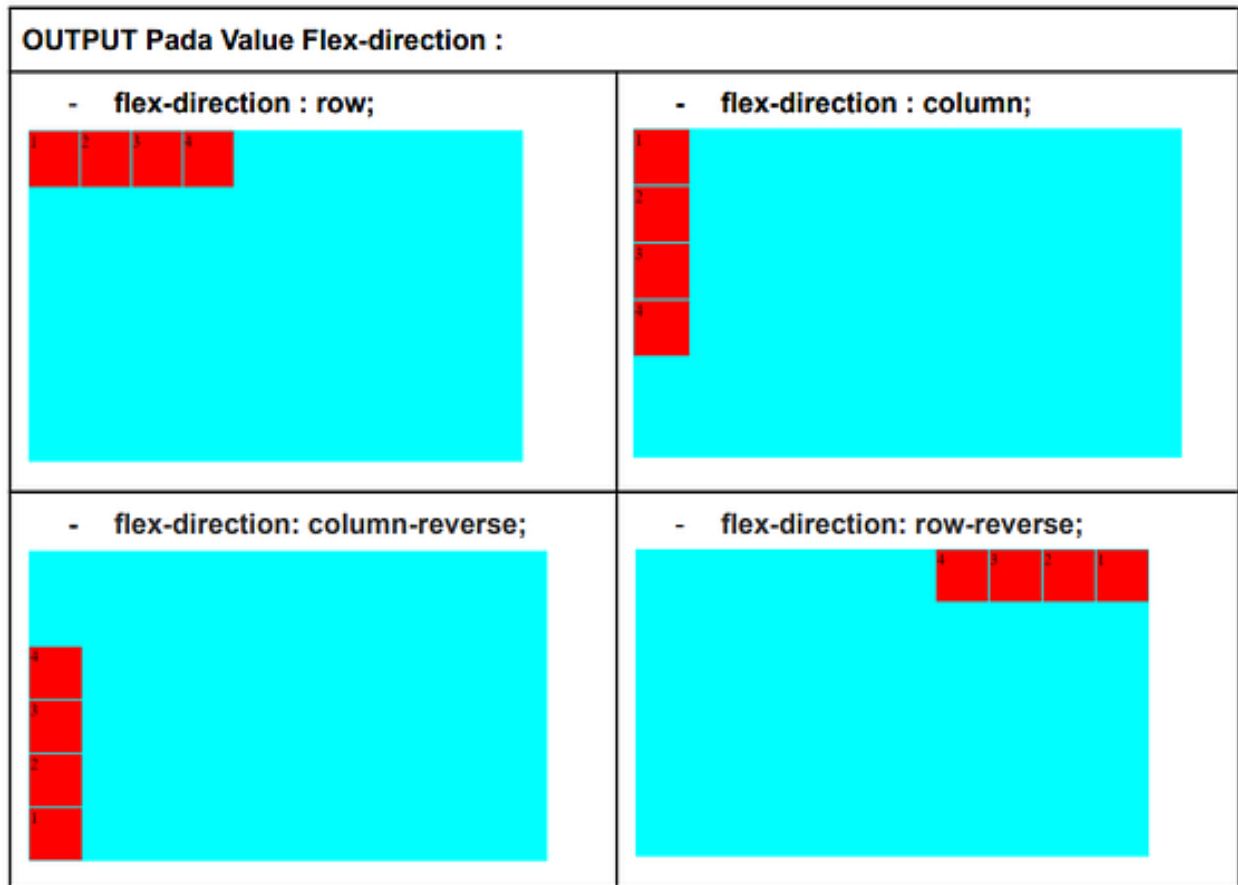
CSS :

```
.container {  
  display: flex;  
  height: 300px;  
  width: 500px;  
  background-color:aqua;  
  flex-direction: row;  
  /*Value : row; / column; / column-reverse; /row-reverse; */  
}  
.box{  
  height: 50px;  
  width: 50px;
```

```

background-color: red;
margin: 1px;
}

```



2. Flex — Wrap

Flex-wrap berfungsi untuk mendefinisikan bahwa elemen item di dalam container flexbox tidak harus disejajarkan dalam satu baris. Artinya, elemen item tersebut digulung (dipindahkan) ke baris baru bila sudah memenuhi lebar container-nya. Value dalam Flex Wrap :

- flex-wrap: wrap; untuk menentukan bahwa nilai item akan dibungkus atau tidaknya.
- flex-wrap: nowrap; (Default); Nilai Item tidak akan dibungkus (Default).
- flex-wrap: wrap-reverse; hampir sama dengan wrap namun urutan akan terbalik.

1) Implementasi Pada Flex Wrap :

HTML :

```

<div class="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
  <div class="box">7</div>
  <div class="box">8</div>

```

```
<div class="box">9</div>
<div class="box">10</div>
<div class="box">11</div>
<div class="box">12</div>
<div class="box">13</div>
<div class="box">14</div>
<div class="box">15</div>
</div>
```

CSS :

```
.container {
    display: flex;
    height: 300px;
    width: 500px;
    background-color:aqua;
    flex-wrap:wrap;
    /* Value : wrap; / nowrap; / wrap-reverse; */
}
.box{
    height: 50px;
    width: 50px;
    background-color: red;
    margin: 1px;
}
```

OUTPUT Pada Property Flex-Wrap :

- flex-wrap: wrap;



- flex-wrap: nowrap;



- flex-wrap: wrap-reverse;



3. Justify-Content

Justify-Content berfungsi untuk menyelaraskan setiap items yang ada pada flex container pada saat items tidak menggunakan semua ruang yang tersedia secara horizontal. Value pada Justify-Content :

- justify-content: center; Untuk menyelaraskan item di tengah container.
- justify-content: flex-start; untuk menyelaraskan item di awal kontainer (default).
- justify-content: flex-end; untuk menyelaraskan item di akhir container.
- justify-content: space-around; menampilkan item dengan jarak sebelum, antara, dan setelah sisi container.
- justify-content: space-between; menampilkan item dengan spasi diantara sisi container.

1. Implementasi pada Properties Justify-Content :

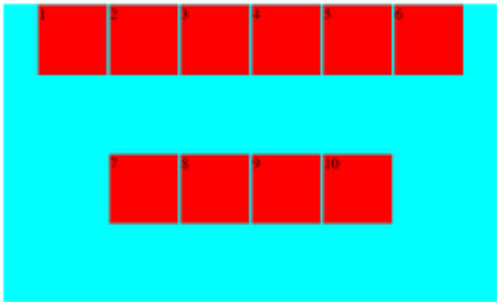
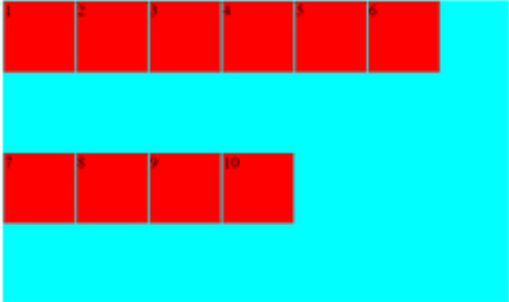
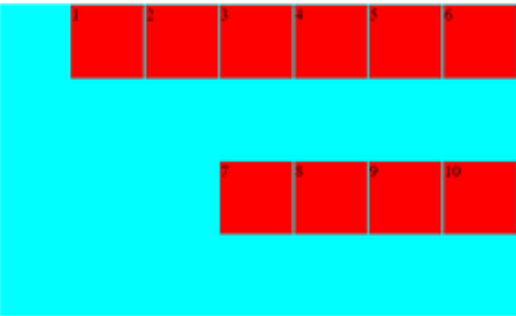
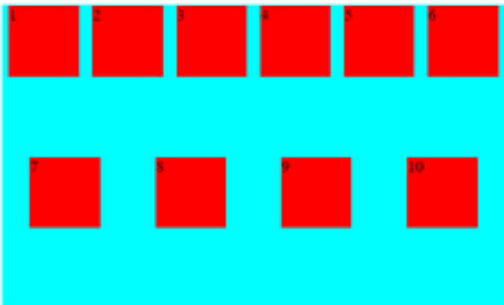
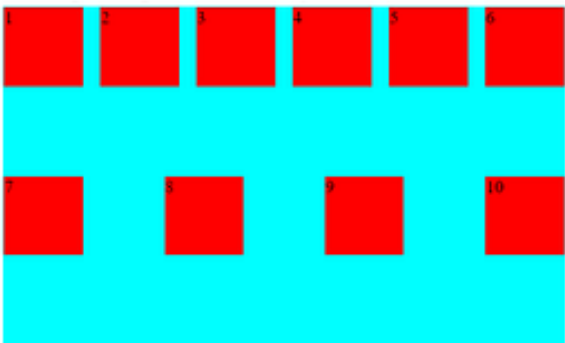
HTML :

```
<div class="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
```

```
<div class="box">7</div>
<div class="box">8</div>
<div class="box">9</div>
<div class="box">10</div>
</div>
```

CSS :

```
.container {
    display: flex;
    height: 300px;
    width: 500px;
    background-color: aqua;
    flex-wrap: wrap;
    justify-content: center;
    /* Value : center; / flex-start; / flex-end / space-around /
    space-between */
}
.box{
    height: 70px;
    width: 70px;
    background-color: red;
    margin: 1px;
}
```

OUTPUT Pada Properties Justify-content :	
<p>- justify-content: center;</p> 	<p>- justify-content: flex-start;</p> 
<p>- justify-content: flex-end;</p> 	<p>- justify-content: space-around;</p> 
<p>- justify-content: space-between;</p> 	

4. align-items

Align-Item berfungsi untuk menyelaraskan setiap items yang pada flex container pada saat items tidak menggunakan semua ruang yang tersedia secara vertikal. Value pada Align-Item :

- align-items: center; //Menyelaraskan Item di tengah Container.
- align-items: flex-start; //Menyelaraskan item di bagian atas container.
- align-items: flex-end; //Menyelaraskan item di bagian bawah kontainer.

1. Implementasi pada Properties Align-Item

HTML :

```
<div class="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
```

```
<div class="box">6</div>
<div class="box">7</div>
<div class="box">8</div>
<div class="box">9</div>
<div class="box">10</div>
```

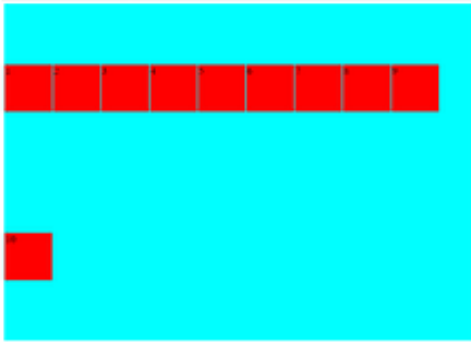
```
</div>
```

CSS :

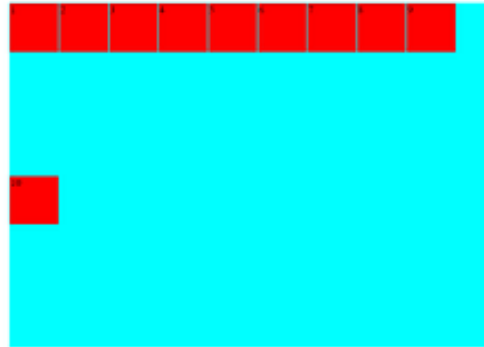
```
.container {
    display: flex;
    height: 500px;
    width: 700px;
    background-color:aqua;
    flex-wrap: wrap;
    align-items:center;
    /* Value : center; / flex-start; / flex-end; */
}
.box{
    height: 70px;
    width: 70px;
    background-color: red;
    margin: 1px;
}
```

OUTPUT Pada Properties Align-Items :

- align-items: center;



- align-items: flex-start;



- align-items: flex-end;



5. Align-content

Align-content berfungsi untuk mengatur setiap items terhadap items lain secara vertikal.

Value pada Align-Content :

- align-content: center; //item flex diselaraskan di bagian tengah dari cross axis.
- align-content: flex-start; //item flex diselaraskan ke arah cross axis dimulai.
- align-content: flex-end; //item flex diselaraskan ke arah cross axis berakhir.
- align-content: space-between; //item flex diletakkan secara merata dalam cross axis, dari flex start sampai flex-end.
- align-content: space-around; //item flex diletakkan secara merata dalam cross axis, tetapi menyisakan setengah ruang kosong di setiap ujungnya.
- align-content: stretch; //item flex akan direng

1. Implementasi Properties pada Align-Content :

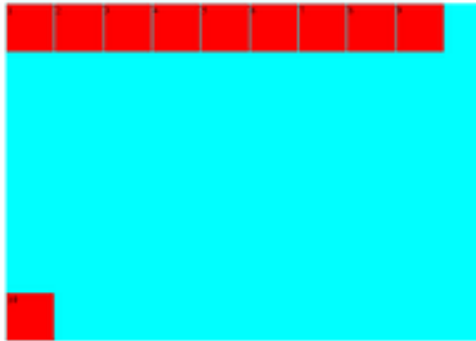
HTML :

```
<div class="container">  
  <div class="box">1</div>  
  <div class="box">2</div>  
  <div class="box">3</div>  
  <div class="box">4</div>  
  <div class="box">5</div>  
  <div class="box">6</div>  
  <div class="box">7</div>
```

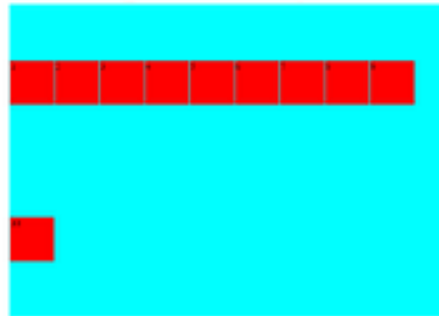
```
<div class="box">8</div>
<div class="box">9</div>
<div class="box">10</div>
</div>
CSS :
.container {
  display: flex;
  height: 500px;
  width: 700px;
  background-color:aqua;
  flex-wrap: wrap;
  align-content: center;
  /* Value : space-between; / tretch; / center; / flex-start; /
  flex-end;*/
}
.box{
  height: 70px;
  width: 70px;
  background-color: red;
  margin: 1px;
}
```

OUTPUT Pada Properties Align-Content :

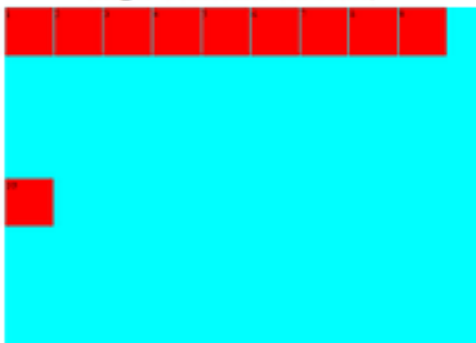
- **align-content: space-between;**



- **align-content: space-around;**



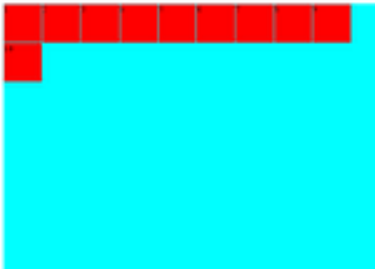
- **align-content: stretch;**



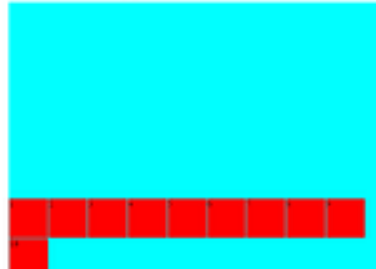
- **align-content: center;**



- **align-content: flex-start;**



- **align-content: flex-end;**



1. Pengenalan CSS GRID

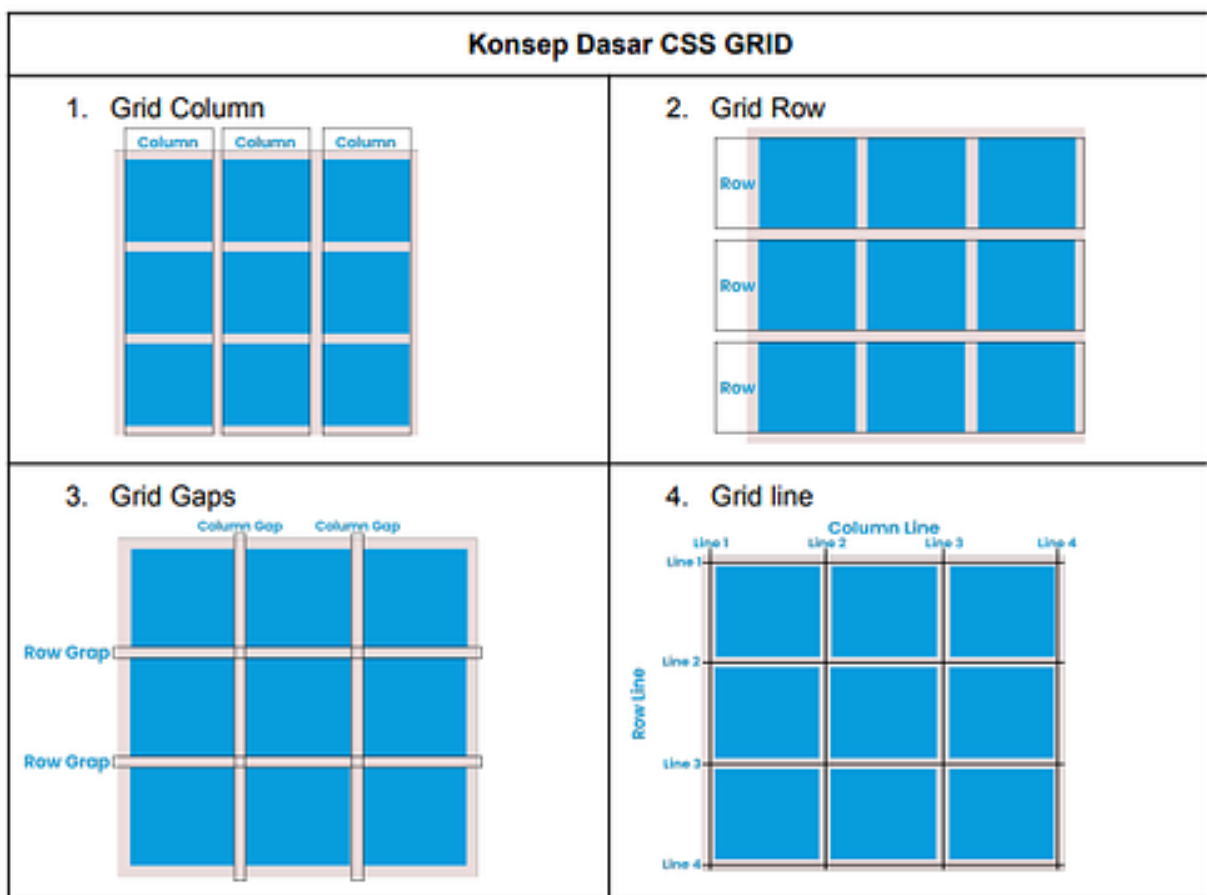
Grid adalah mode layout 2 dimensi yang terdapat di CSS3 yang digunakan mengatur element di suatu halaman web. Grid ini mengatur sistem layout baris dan kolom.

2. Untuk Apa Grid ?

Tujuan grid adalah memberikan kemampuan kepada container untuk mengatur panjang, lebar, ukuran dan posisi items-items secara horizontal dan vertikal, dengan yang diatur baris saja, kolom saja, atau bisa keduanya.

3. Perbedaan CSS GRID dengan FLEXBOX

Secara sederhana CSS Grid adalah CSS yang dapat membagi kolom pada website menjadi beberapa bagian sesuai yang diinginkan, baik secara horizontal maupun vertikal. sedangkan Flexbox hanya bisa diatur dengan 1 dimensi saja. namun kedua CSS tersebut bukanlah kompetitor, keduanya sama-sama dibutuhkan dan dapat saling berkolaborasi untuk menyusun layout yang kompleks.



4. Istilah pada CSS GRID

- Grid Container : element pembungkus items yang memiliki properties grid
- Grid Item : element-element atau items-items yang berada di dalam grid container.
- Grid Line : garis horizontal atau vertikal yang memisahkan grid menjadi beberapa bagian yang ditandai dengan angka
- Grid Cell : potongan yang berada antara baris dan kolom di dalam grid
- Grid area : kumpulan potongan grid cell yang membentuk kotak
- Grid Track : ukuran atau jarak antara 2 grid line, horizontal(baris) atau vertikal(baris)
- Grid gap : jarak antara grid cell

5. Properti pada CSS GRID

1. Properti **grid-template-columns**

Berfungsi untuk menentukan kolom mana untuk menempatkan item. Kita bisa menentukan di mana item akan dimulai dan di mana item akan berakhir.

HTML :

```
<div class="grid-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
</div>
```

CSS :

```
.grid-container {
  display: grid;
  grid-template-columns: 100px 100px 100px 100px;
  gap: 10px;
  background-color: #f1f1f1;
  padding: 10px;
}
.grid-container > div {
  background-color: DodgerBlue;
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
```

OUTPUT :



2. Properti **grid-template-rows**

menentukan baris mana untuk menempatkan item. Kita akan menentukan di mana item akan dimulai, dan di mana item akan berakhir.

HTML :

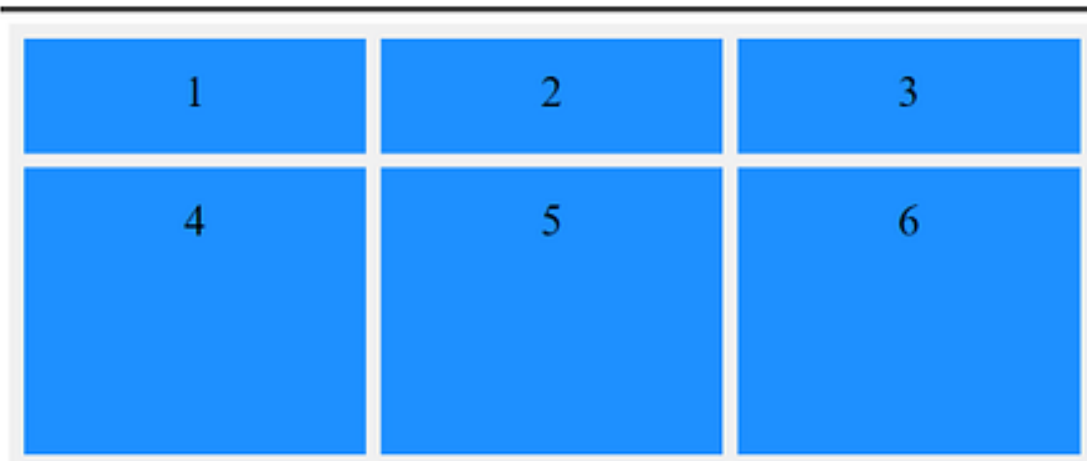
```
<div class="grid-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
```

```

        <div>5</div>
        <div>6</div>
    </div>
    CSS :
    .grid-container {
        display: grid;
        grid-template-columns: auto auto auto;
        grid-template-rows: 80px 200px;
        gap: 10px;
        background-color: #f1f1f1;
        padding: 10px;
    }
    .grid-container > div {
        background-color: DodgerBlue;
        text-align: center;
        padding: 20px 0;
        font-size: 30px;
    }

```

OUTPUT :



3. Property justify-content

Properti ini untuk menyelaraskan seluruh item di dalam container pada sumbu horizontal.

Value dari Justify-Content :

- space-evenly jarak kanan kiri dan tengah dari item nya akan sama rata.
- space-around akan memberi jarak pada tiap tiap item nya. jaraknya yg paling kanan dan kiri yaitu setengah dari jarak tengah item.
- space-between pada paling kiri dan kanan akan menempel dengan sisi container jadi tidak ada jarak nya.
- center posisi item akan berada ditengah container nya.
- start maka seluruh grid nya akan ada di kiri pada container nya. akan terjadi jika mempunyai sisa di dalam container nya.
- end posisi akan berada di kanan / di akhir dari containernya.

1. Implementasi Pada Property Flex-direction :





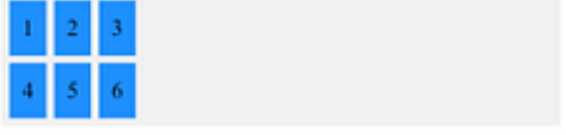
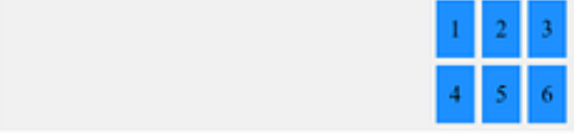
HTML :

```
<div class="grid-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
</div>
```

CSS :

```
.grid-container {
  display: grid;
  grid-template-columns: 50px 50px 50px; /*Make the grid smaller than
  the container*/
  gap: 10px;
  background-color: #f1f1f1;
  padding: 10px;
  justify-content:center;
  /* Value : space-evenly; / space-around; / space-between; / center; /
  start; / end; */
}
.grid-container > div {
  background-color: DodgerBlue;
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
```

OUTPUT :

OUTPUT Pada Value Justify-Content :	
<p>- justify-content: space-evenly;</p> 	<p>- justify-content: space-around;</p> 
<p>- space-between;</p> 	<p>- justify-content: center;</p> 
<p>- justify-content: start;</p> 	<p>- justify-content: end;</p> 

4. Property align-content'

hampir sama dengan justify konten yang membedakan yaitu untuk sumbu vertical. Value dari Align-Content :

- center; maka dia akan menemmel ke tengah container nya secara vertical.
- space-evenly; jarak kanan kiri dan tengah dari item nya akan sama rata secara vertical.
- space-around; akan memberi jarak pada tiap tiap item nya. jaraknya yg paling kanan dan kiri yaitu setengah dari jarak tengah item secara vertical.
- space-between; pada paling kiri dan kanan akan menempel dengan sisi container jadi tidak ada jarak nya secara vertical.
- start; maka item akan menempel diatas container nya secara vertical.
- end; maka item akan menempel ke bawah container nya secara vertical.

1. Implementasi pada Property Align-Content :

HTML :

```
<div class="grid-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
</div>
```

CSS :

```
.grid-container {
  display: grid;
  height: 400px;
  grid-template-columns: auto auto auto;
  gap: 10px;
  background-color: #f1f1f1;
  padding: 10px;
  align-content: space-evenly;
  /* Value : center; / space-evenly / space-around; / space-between; /
  start; / end; */
}

.grid-container > div {
  background-color: DodgerBlue;
  text-align: center;
  padding: 20px 0;
}
```

OUTPUT Pada Value Align-Content :

- **align-content: center;**

1	2	3
4	5	6

- **align-content: space-evenly;**

1	2	3
4	5	6

- **align-content: space-around;**

1	2	3
4	5	6

- **align-content: space-between;**

1	2	3
4	5	6

- **align-content: start;**

1	2	3
4	5	6

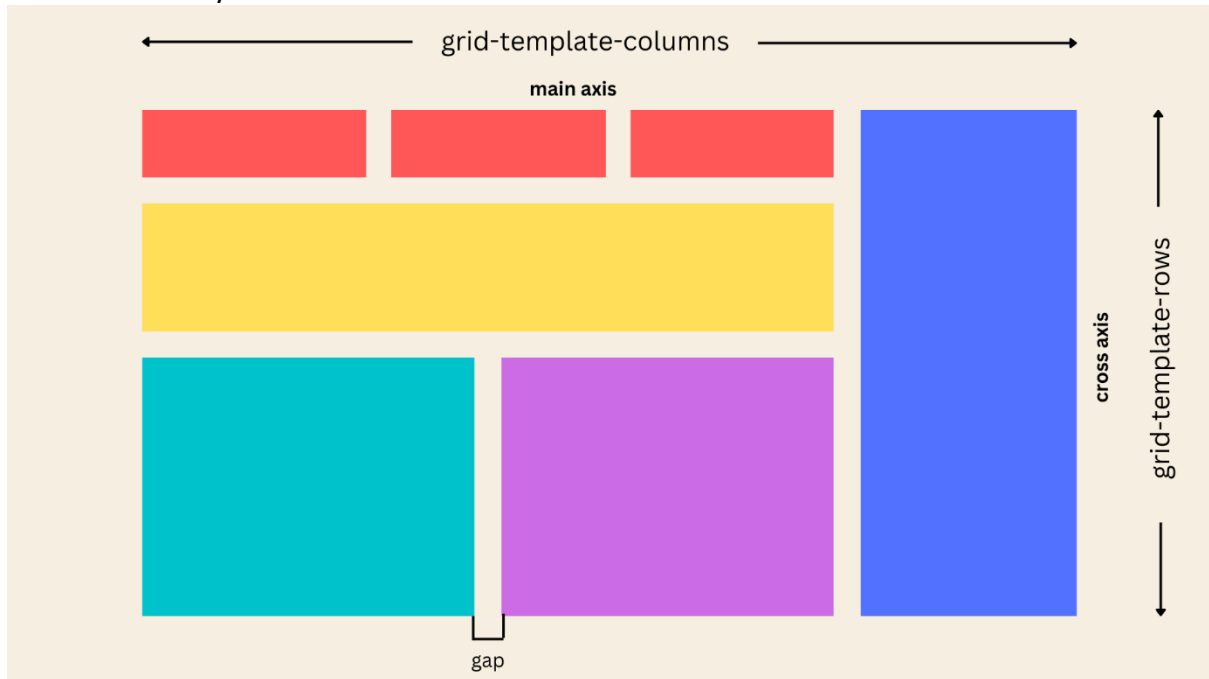
- **align-content: end;**

1	2	3
4	5	6

TUGAS PROJECT :

Buatlah tampilan halaman web seperti pada mockup dibawah dengan menggunakan konsep box flex atau grid (optional) sesuai kemampuan dan pemahaman kamu, sesuai komposisi warna yang telah ditetapkan dengan menggunakan Bahasa HTML dan CSS :

Web1.html & Style1.css



Web2.html & Style2.css

