



STT-NF

# Pemrograman Web

Sirojul Munir SSi,MKom | [rojulman@nurulfikri.ac.id](mailto:rojulman@nurulfikri.ac.id)



[dev.xbata.com](http://dev.xbata.com)

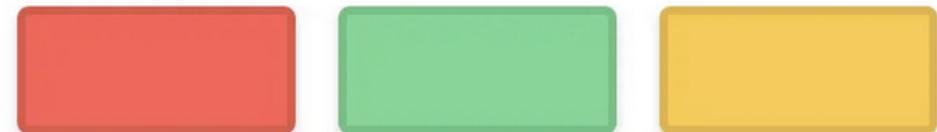
 @rojulman

# 8. CSS Layout Flexbox & Grid

- Layout Flexible Box (Flexbox) dan Layout Grid memungkinkan web desainer / programmer melakukan layout komponen2 halaman web
- Komponen website:
  1. Menu bars
  2. Box Menu
  3. Daftar produk
  4. Gallery foto
  5. Form
  6. Articles

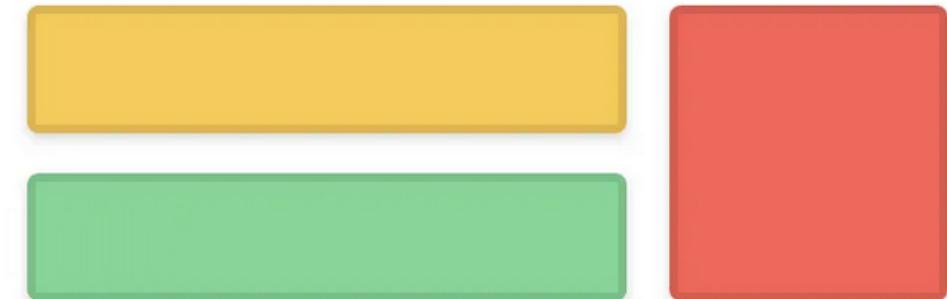
## Flexbox

One-dimensional layout

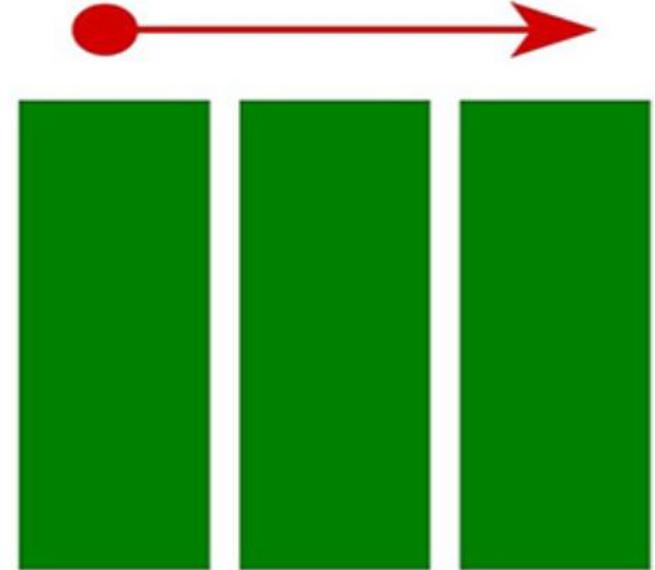


## Grid

Multi-dimensional layout

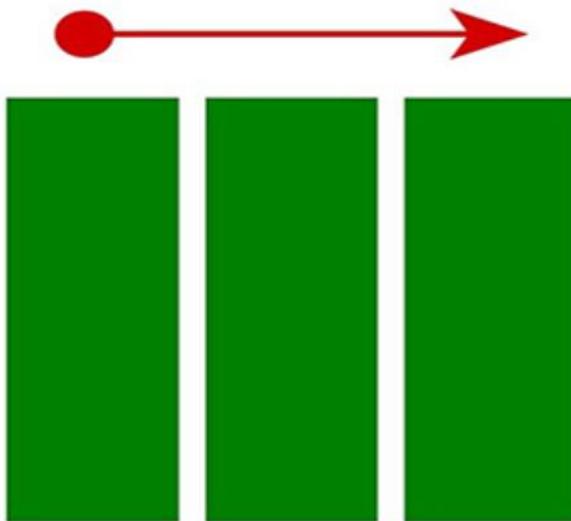


- Layout Flexbox: desain layout komponen berdasarkan satu axis (one dimensions)
- Komponen Flexbox layout disebut: Flex Item
- Layout item flexbox diatur dalam sebuah container pada tag HTML div



```
<div id="container">
  <div class="box box1">1</div>
  <div class="box box2">2</div>
  <div class="box box3">3</div>
  <div class="box box4">4</div>
  <div class="box box5">5</div>
</div>
```

Flexbox  
One Dimensions



## Flexbox One Dimensions

In this book, Philadelphia Ice Creams, comprising the first group, are very palatable, but expensive. In many parts of the country it is quite difficult to get good cream. For that reason, I have given a group of creams, using part milk and part cream, but it must be remembered that it takes smart "juggling" to make ice cream from milk. By far better use condensed milk, with enough water or milk to rinse out the cans.

Ordinary fruit creams may be made with condensed milk at a cost of about fifteen cents a quart, which, of course, is cheaper than ordinary milk and cream.

In places where neither cream nor condensed milk can be purchased, a fair ice cream is made by adding two tablespoonfuls of olive oil to each quart of milk. The cream for Philadelphia Ice

If pure raw cream is stirred rapidly, it swells and becomes frothy, like the beaten whites of eggs, and is "whipped cream." To prevent this in making Philadelphia Ice Cream, one-half the cream is scalded, and when it is very cold, the remaining half of raw cream is added. This gives the smooth, light and rich consistency which makes these creams so different from others.

### Use of Fruits

Use fresh fruits in the summer and the best canned unsweetened fruits in the winter. If sweetened fruits must be used, cut down the given quantity of sugar. Where acid fruits are used, they should be added to the cream after it is partly frozen.

water ices require a longer time than ice creams. It is not well to freeze the mixtures too rapidly; they are apt to be coarse, not smooth, and if they are churned before the mixture is icy cold they will be greasy or "buttery."

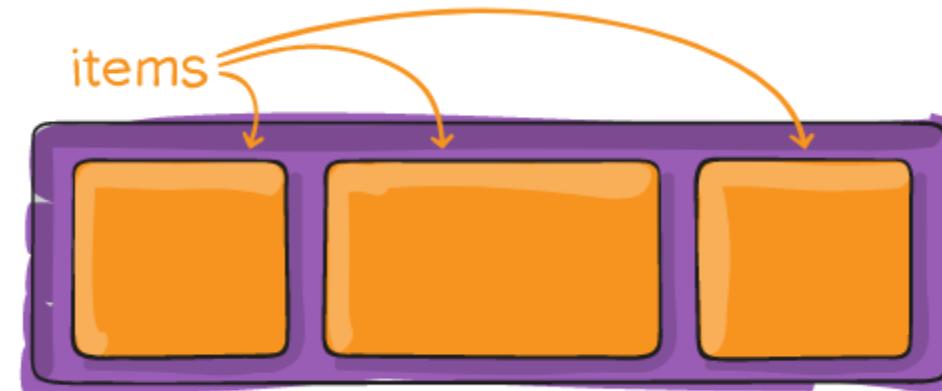
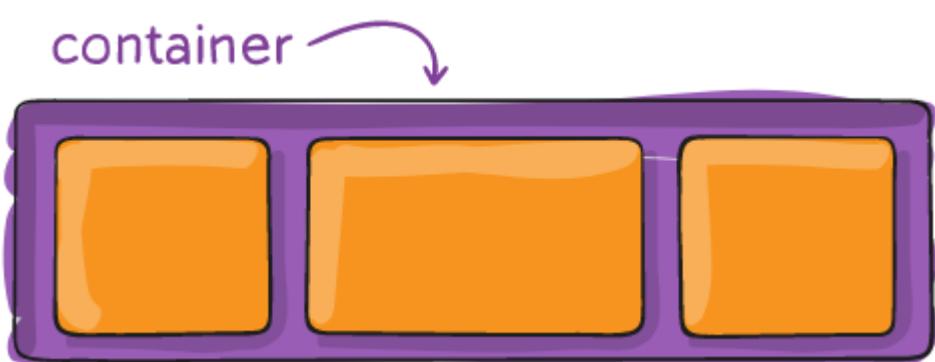
The average time for freezing two quarts of cream should be ten minutes; it takes but a minute or two longer for larger quantities.

Pound the ice in a large bag with a mallet, or use an ordinary ice shaver. The finer the ice, the less time it takes to freeze the cream. A four quart freezer will require ten pounds of ice, and a quart and a pint of coarse rock salt. You may pack the freezer with a layer of ice three inches thick, then a layer of salt one inch thick, or mix the ice and salt in the tub and shovel it around the freezer.

- Beberapa keuntungan Flexbox Layout:
  - Memastikan semua komponen layout yang diatur memiliki ukuran sama tinggi
  - Kemudahan pengaturan komponen layout secara horizontal dan vertical
  - Dapat dilakukan perubahan urutan (order) dari komponen web yang di layout

# CSS – Flexbox Container

- Sebuah container terdiri atas komponen2 (Flex Item) yang akan dilakukan layout



<https://css-tricks.com/snippets/css/a-guide-to-flexbox/#basics-and-terminology>

# CSS – Flexbox Container

- Sebuah container terdiri atas komponen2 (Flex Item) yang akan dilakukan layout

## THE MARKUP

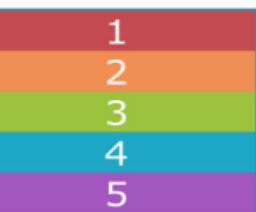
```
<div id="container">  
  <div class="box box1">1</div>  
  <div class="box box2">2</div>  
  <div class="box box3">3</div>  
  <div class="box box4">4</div>  
  <div class="box box5">5</div>  
</div>
```

## THE STYLES

```
#container {  
  display: flex;  
}
```

By default, the divs display as block elements, stacking up vertically. Turning on flexbox mode makes them line up in a row.

display: flex;



block layout mode

flexbox layout mode



# CSS – Flexbox Direction

Digunakan untuk mengatur arah dari komponen layout

## flex-direction

Values: row | column | row-reverse | column-reverse

Default: row

Applies to: flex containers

Inherits: no

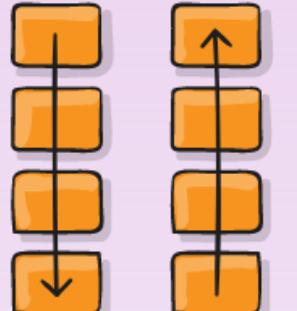
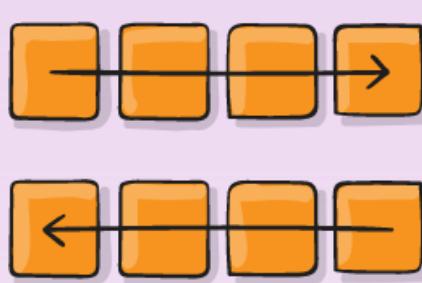
flex-direction: row; (default)



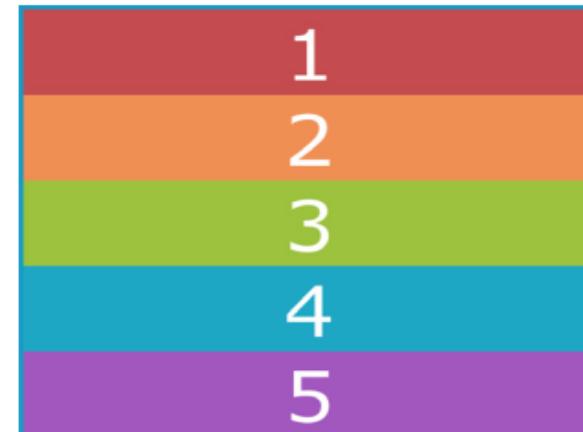
flex-direction: row-reverse;



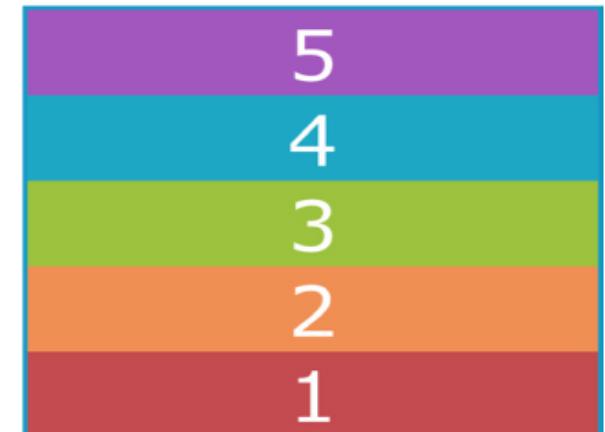
## flex-direction



flex-direction: column;



flex-direction: column-reverse;



<https://css-tricks.com/snippets/css/a-guide-to-flexbox>

# CSS – Flexbox Wrap

Digunakan untuk mengatur arah dari komponen layout yang belum ditentukan jumlah flex itemnya

## THE MARKUP

```
<div id="container">
  <div class="box box1">1</div>
  <!-- more boxes here -->
  <div class="box box10">10</div>
</div>
```

## THE STYLES

```
#container {
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
}

.box {
  width: 25%;
```

1	2	3	4
5	6	7	8
9	10		

9	10		
5	6	7	8
1	2	3	4

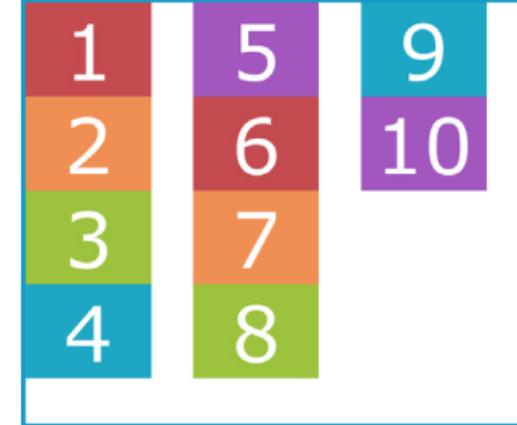
# CSS – Flexbox Wrap

```
#container {  
    display: flex;  
    height: 350px;  
    flex-direction: column;  
    flex-wrap: wrap;  
}  
.box {  
    width: 25%;  
}
```

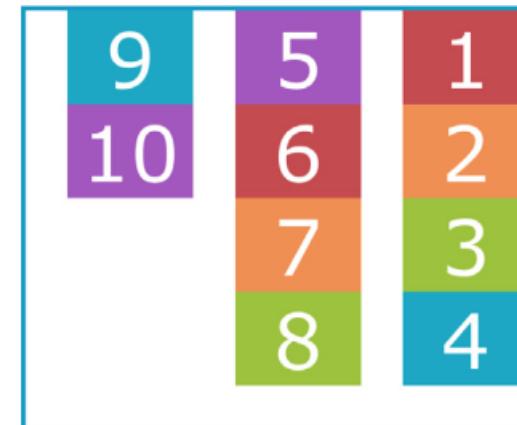
flex-wrap: nowrap; (default)



flex-wrap: wrap;



flex-wrap: wrap-reverse;



Digunakan untuk alignment (rataan) flex item

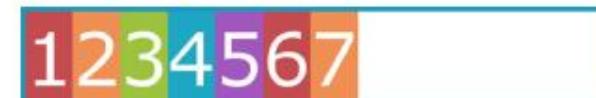
## justify-content

**Values:** flex-start | flex-end | center | space-between | space-around

**Default:** flex-start

justify-content: flex-start; (default)

**Applies to:** flex containers



**Inherits:** no

```
#container {  
  display: flex;  
  justify-content: flex-start;  
}
```

justify-content: flex-end;



justify-content: center;



justify-content: space-between;



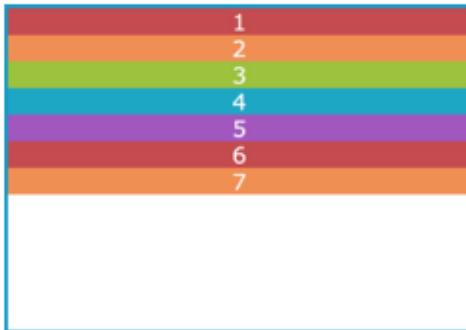
justify-content: space-around;



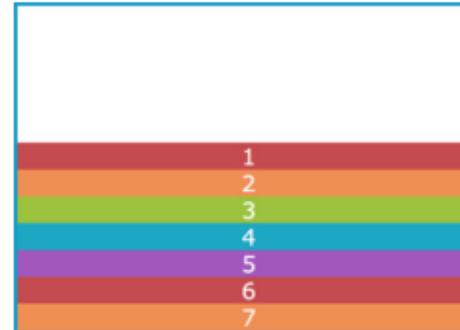
# CSS – Flexbox Alignment Content

Digunakan untuk alignment (rataan) flex item

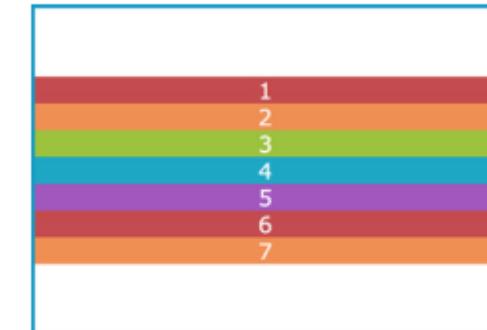
`justify-content: flex-start;` (default)



`justify-content: flex-end;`



`justify-content: center;`



`justify-content: space-between;`



`justify-content: space-around;`



# CSS – Flexbox Alignment Cross Axis

Digunakan untuk alignment flex item dengan pengaturan cross axis (row: up – down, column: left – right)

## align-items

**Values:** flex-start | flex-end | center | baseline | stretch

**Default:** stretch

**Applies to:** flex containers

**Inherits:** no

```
#container {  
    display: flex;  
    flex-direction: row;  
    height: 200px;  
    align-items: flex-start;  
}
```

align-items: flex-start;



1234567

# CSS – Flexbox Alignment Cross Axis

`align-items: flex-start;`



1234567

A horizontal row of seven colored boxes (red, orange, yellow, green, blue, purple, red) containing the numbers 1 through 7 respectively. They are aligned at the top edge of their container.

`align-items: flex-end;`



1234567

A horizontal row of seven colored boxes (red, orange, yellow, green, blue, purple, red) containing the numbers 1 through 7 respectively. They are aligned at the bottom edge of their container.

`align-items: center;`



1234567

A horizontal row of seven colored boxes (red, orange, yellow, green, blue, purple, red) containing the numbers 1 through 7 respectively. They are centered vertically within their container.

`align-items: stretch;` (default)



1234567

A horizontal row of seven colored boxes (red, orange, yellow, green, blue, purple, red) containing the numbers 1 through 7 respectively. The boxes are stretched vertically to fill the height of their container.

Items are aligned so that the baselines of the first text lines align.

`align-items: baseline;`



123467

A horizontal row of seven colored boxes (red, orange, yellow, green, blue, purple, red) containing the numbers 1 through 7 respectively. The boxes are aligned by their baseline, which is lower than the baseline of the '1' box. A dashed red line indicates the baseline level.

# CSS – Flexbox Alignment Self

Digunakan untuk alignment flex item diberlakukan hanya untuk 1 elemen flex item

## align-self

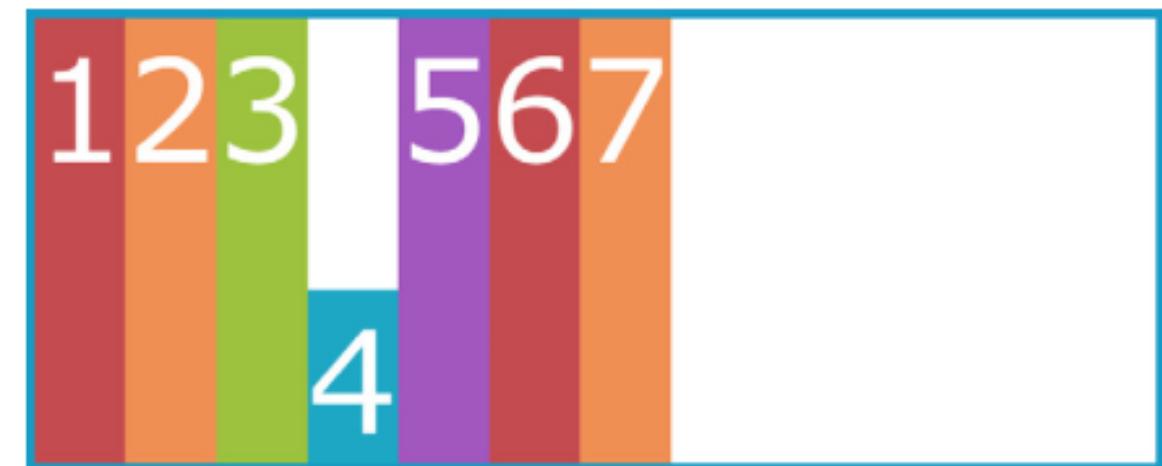
**Values:** flex-start | flex-end | center | baseline | stretch

**Default:** stretch

**Applies to:** flex items

**Inherits:** no

```
.box4 {  
    align-self: flex-end;  
}
```



# CSS – Flexbox Alignment Multipleline

Digunakan untuk multiple line align flex item

## align-content

**Values:** flex-start | flex-end | center | space-around | space-between | stretch

**Default:** stretch

**Applies to:** multi-line flex containers

**Inherits:** no

```
#container {  
    display: flex;  
    flex-direction: row;  
    flex-wrap: wrap;  
    height: 350px;  
    align-items: flex-start;  
}  
  
box {  
    width: 25%;  
}
```

align-content: flex-start;

1	2	3	4
5	6	7	8
9	10		

# CSS – Flexbox Alignment Multiple line

Digunakan untuk multiple line align flex item

`align-content: flex-start;`

1	2	3	4
5	6	7	8
9	10		

`align-content: flex-end;`

1	2	3	4
5	6	7	8
9	10		

`align-content: center;`

1	2	3	4
5	6	7	8
9	10		

`align-content: space-between;`

1	2	3	4
5	6	7	8
9	10		

`align-content: space-around;`

1	2	3	4
5	6	7	8
9	10		

`align-content: stretch; (default)`

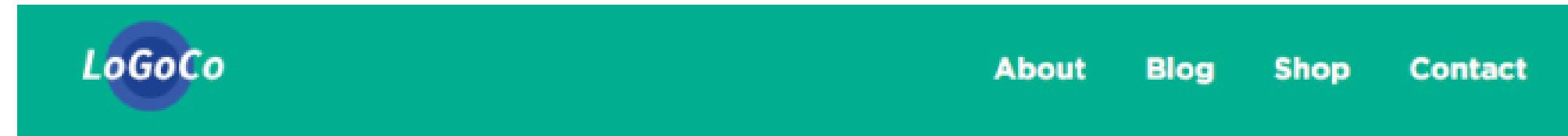
1	2	3	4
5	6	7	8
9	10		

## THE MARKUP

```
<ul>
  <li class="logo"></li>
  <li>About</li>
  <li>Blog</li>
  <li>Shop</li>
  <li>Contact</li>
</ul>
```

## THE STYLES

```
ul {
  display: flex;
  align-items: center;
  background-color: #00af8f;
  list-style: none; /* removes bullets */
  padding: .5em;
  margin: 0;
}
li {
  margin: 0 1em;
}
li.logo {
  margin-right: auto;
}
```



# CSS – Flexbox Grow

Digunakan untuk mengatur flex item pertama dengan ukuran nilai tertentu dan flex item berikutnya bertumbuh

## flex-grow

**Values:** *number*

**Default:** 0

**Applies to:** flex items

**Inherits:** no

## THE MARKUP

```
<div id="container">  
  <div class="box box1">1</div>  
  <div class="box box2">2</div>  
  <div class="box box3">3</div>  
  <div class="box box4">4</div>  
  <div class="box box5">5</div>  
</div>
```

## THE STYLES

```
.box {  
  ...  
  flex: 1 1 auto;  
}
```

`flex: 0 1 auto;` (prevents expansion)



`flex: 1 1 auto;` (allows expansion)



Kebalikan Flex-grow, flex item berikutnya akan mengelil

## flex-shrink

Values: *number*

Default: 1

Applies to: flex items

Inherits: no

`flex: 0 1 100px;`



When the container is wide, the items will not grow wider than their **flex-basis** of 100 pixels because **flex-grow** is set to 0.

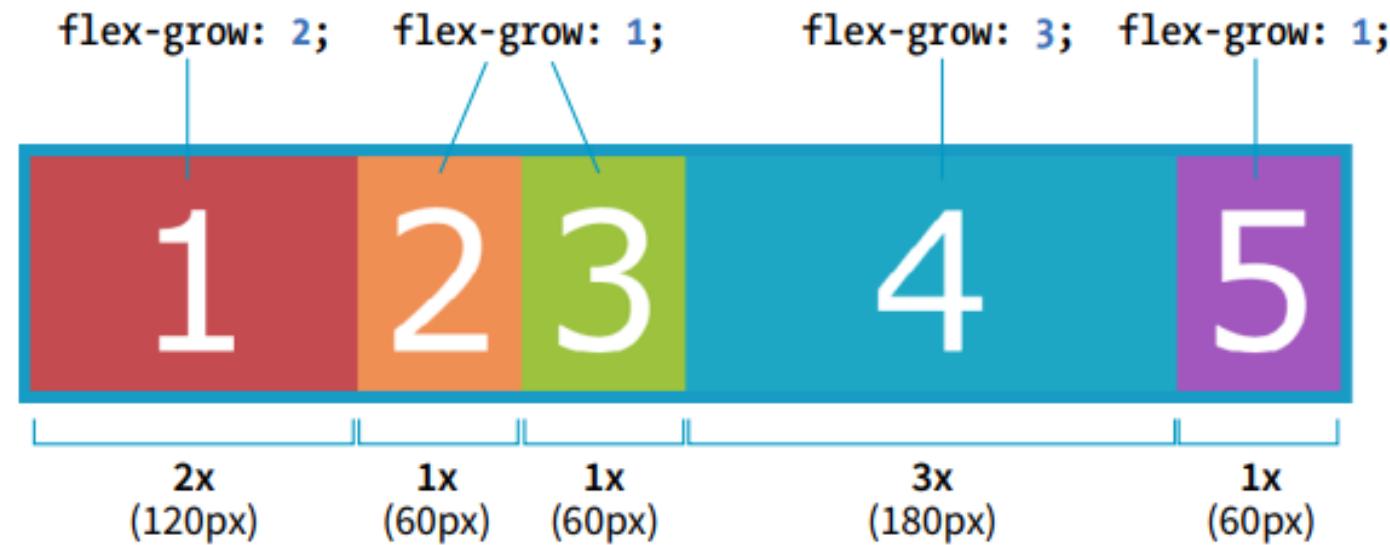


When the container is narrow, the items are allowed to shrink to fit (**flex-shrink: 1**).

# CSS – Flexbox Absolute vs Relative

Digunakan untuk mengatur rasio flex item secara absolute atau relative

```
.box {  
  /* applied to all boxes */  
  flex: 1 0 0%;  
}  
.box1 {  
  flex: 2; /* shortcut value for flex: 2 1 0px */  
}  
.box4 {  
  flex: 3; /* shortcut value for flex: 3 1 0px */  
}
```



Digunakan untuk mengatur urutan item flex secara numerik  
**order**

**Values:** *integer*

**Default:** 0

**Applies to:** flex items and absolutely positioned children of flex containers

**Inherits:** no

```
.box3 {  
    order: 1;  
}
```



order: 0 (default)    order: 0 (default)    order: 0 (default)    order: 0 (default)    order: 1

# CSS – Flexbox Order

Digunakan untuk mengatur urutan item flex secara ordinal

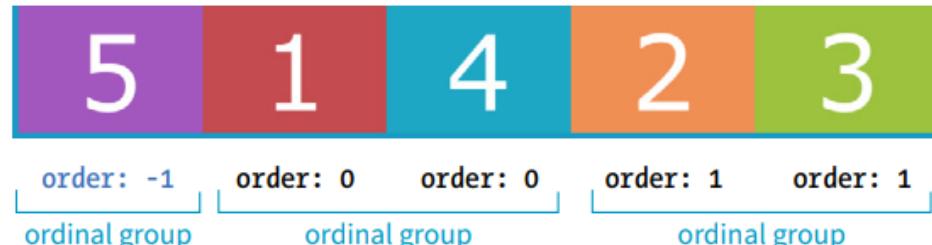
```
.box3 {  
  order: 1;  
}
```



```
.box2, .box3 {  
  order: 1  
}
```



```
.box5 {  
  order: -1  
}
```

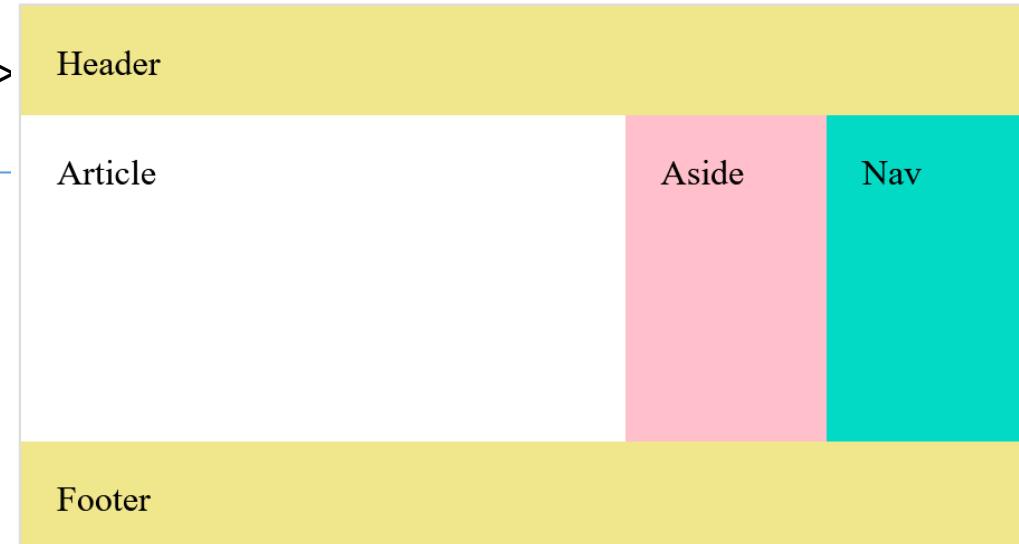


# CSS – Flexbox Web Layout : Praktikum 8.2

```
<style>
* {
    box-sizing: border-box;
}
body {
    display: flex;
    min-height: 100vh;
    flex-direction: column;
    margin: 0;
}
#main {
    display: flex;
    flex: 1;
}
#main > article {
    flex: 1;
    order: 1;
}
#main > nav,
#main > aside {
    flex: 0 0 20vw;
}

#main > nav {
    background: #03DAC5;;
    order: 3;
}
#main > aside {
    background: pink;
    order: 2;
}
header, footer {
    background: khaki;
    height: 20vh;
}
header, footer, article, nav, aside {
    padding: 1em;
}
</style>
```

```
<body>
<header>Header</header>
<div id="main">
    <article>Article</article>
    <nav>Nav</nav>
    <aside>Aside</aside>
</div>
<footer>Footer</footer>
</body>
```





# Referensi Layout Flexbox

---

1. <https://tobiasahlin.com/blog/common-flexbox-patterns/>
2. <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
3. [https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)

# Praktikum 8.3

---

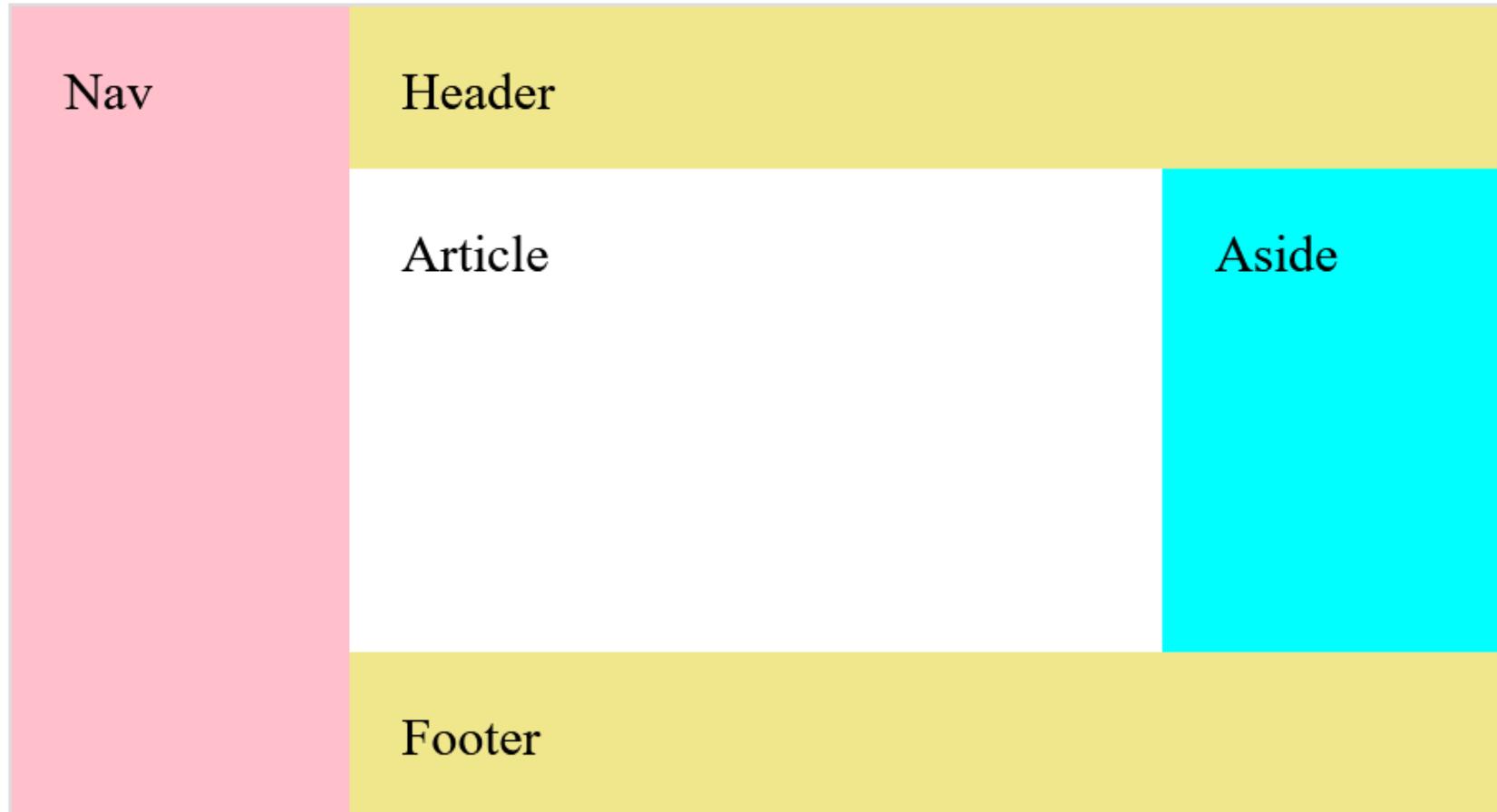
Gunakan Flexbox untuk menampilkan foto dengan desain seperti Gambar 8.3 (ganti kotak dengan foto/gambar)



**Gambar 8.3** Representative Flex Layout Example

# Praktikum 8.4

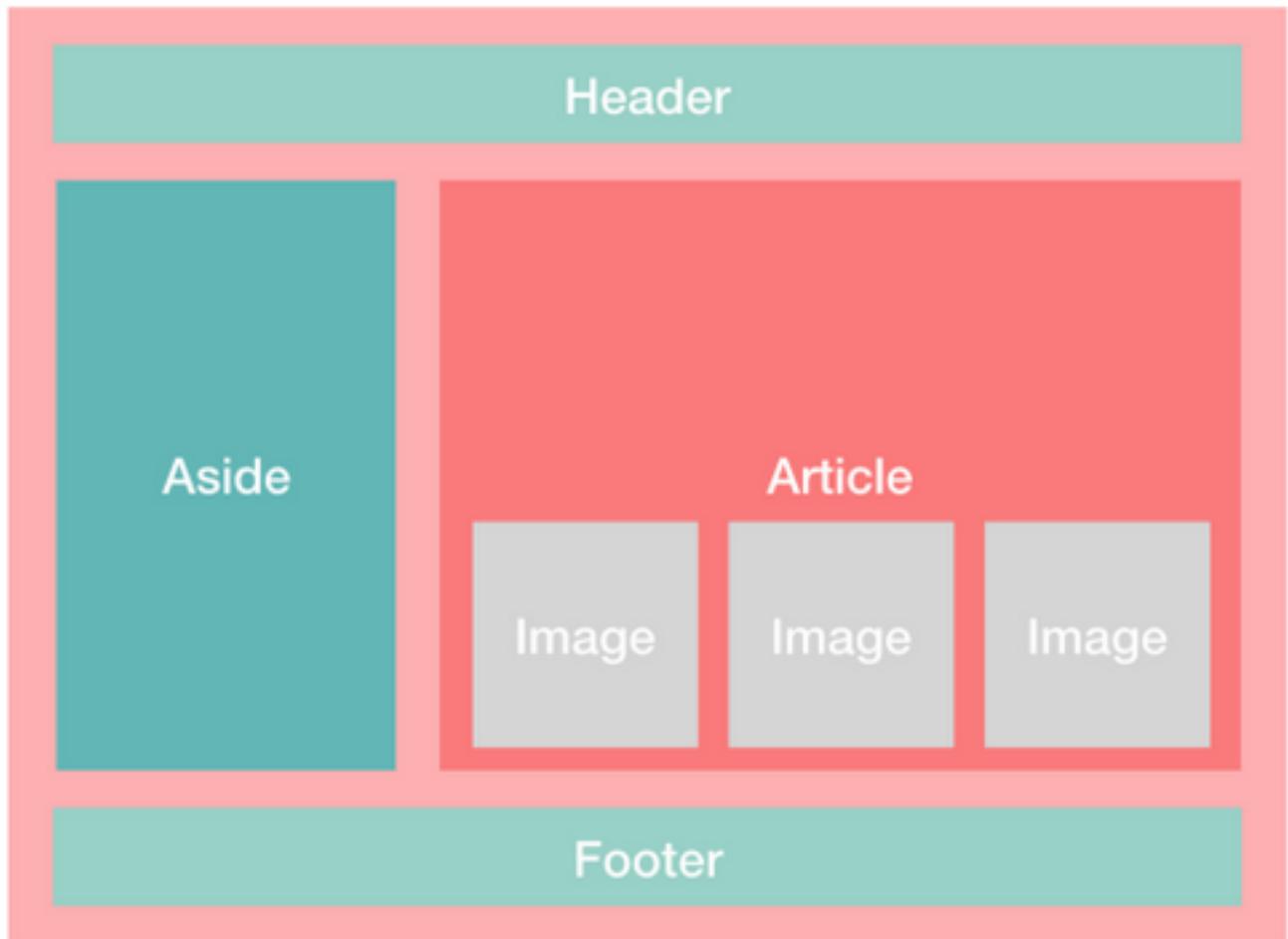
- Gunakan Flexbox untuk desain seperti Gambar 8.4



Gambar 8.4

# Praktikum 8.5

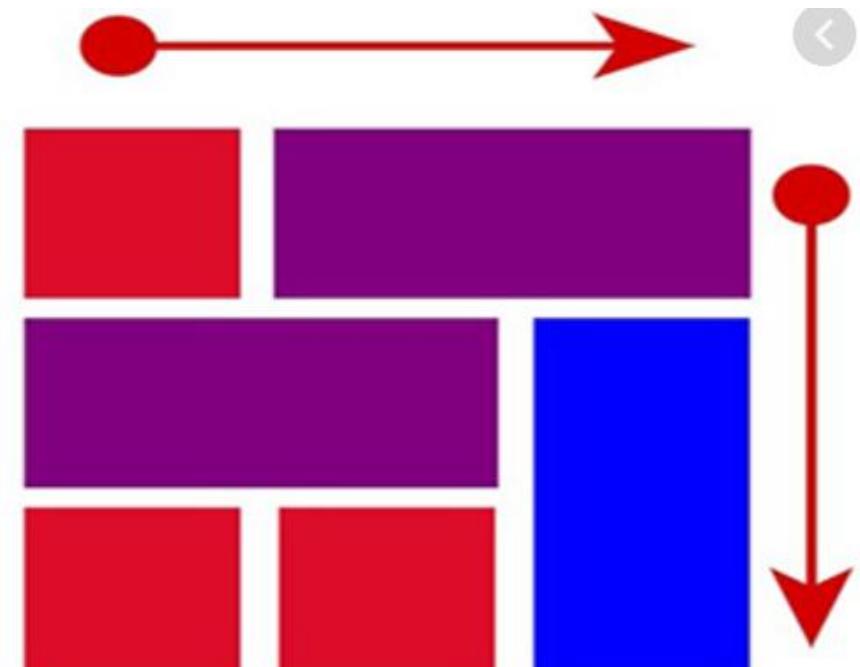
- Gunakan Flexbox untuk desain seperti Gambar 8.5



Gambar 8.5

# CSS – Layout Grid

- Layout Grid memungkinkan web desainer / programmer melakukan layout komponen2 halaman web dalam bentuk rows dan column
- Layout Grid membuat beberapa macam ukuran komponen elemen web menjadi flexible untuk berbagai ukuran
- Dua dimensi: rows & columns
- Elemen Grid disebut : Grid Item



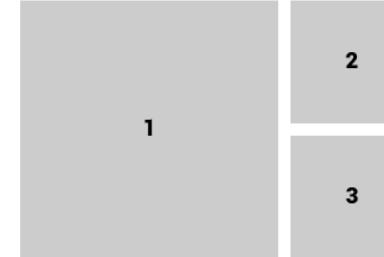
**CSS Grids**  
Two Dimensions

# CSS – Cara Kerja Layout Grid

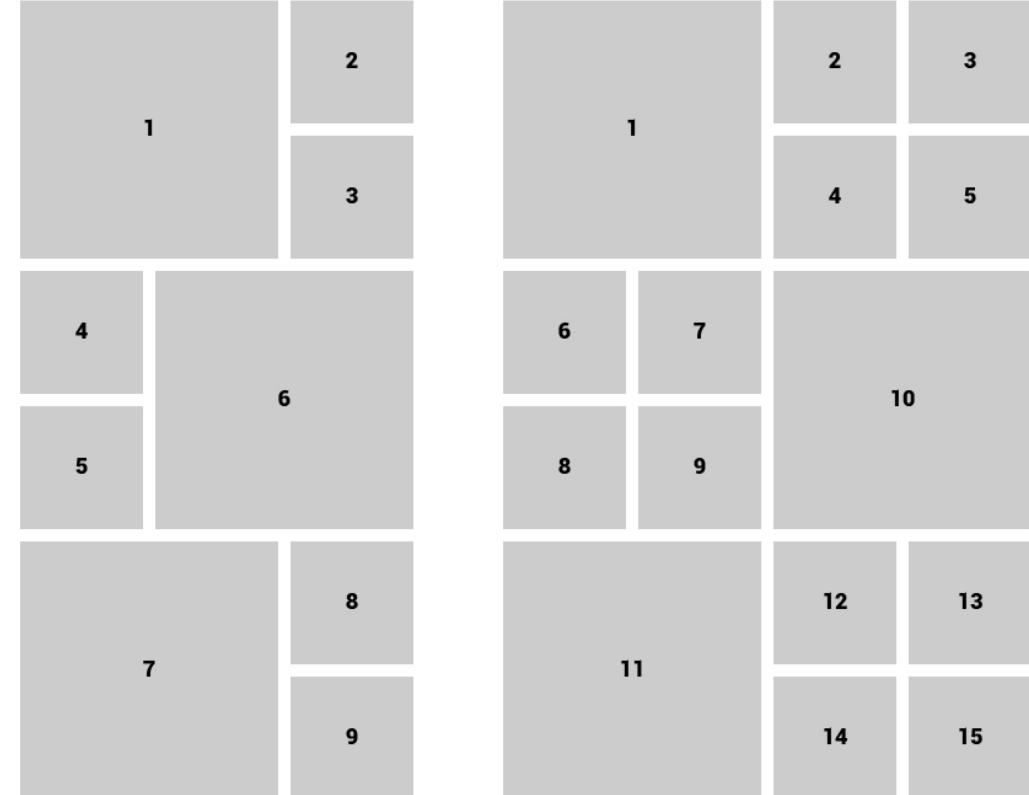
- Gunakan property display untuk membagi elemen grid pada container
- Lakukan setting rows dan column dari Grid
- Set area setiap grid item ke dalam area pada Grid

```
.grid-container {  
    display: grid;  
}
```

Screen width: 640~1023



Screen width: 1024+



# CSS – Grid Terminologi

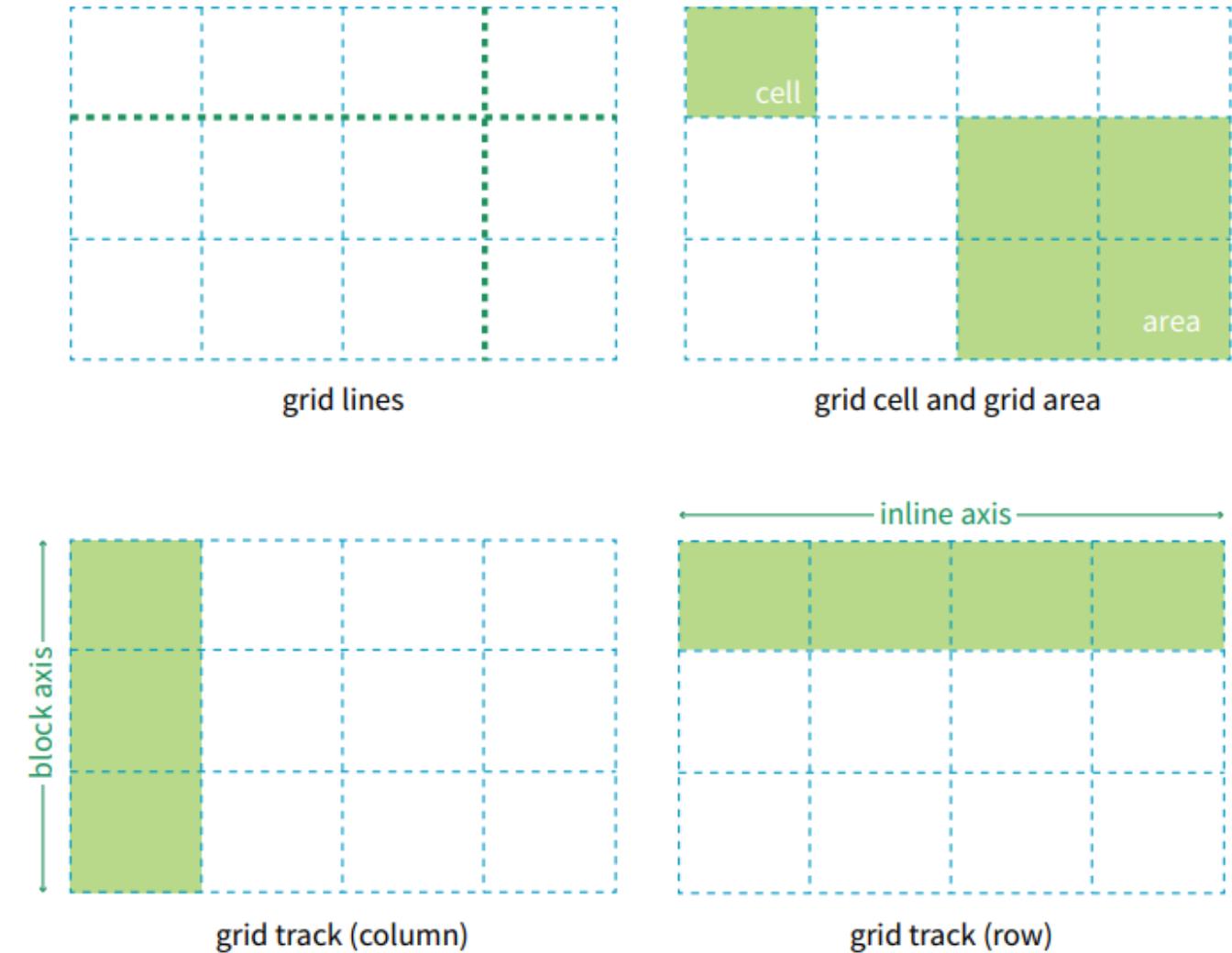
---

- Beberapa hal yang perlu dipahami untuk konsep GRID Layout:
  1. **Grid Line**  
Garis horizontal atau vertical membagi GRID
  2. **Grid Cell**  
Unit terkecil dari sebuah GRID yang memiliki border, dan tidak ada garis didalamnya
  3. **Grid Area**  
Area persegi Panjang yang terdiri dari satu atau lebih sel grid yang berdekatan
  4. **Grid Track**  
Ruang antara dua garis grid yang saling berdekatan

# CSS – Grid Terminologi

Beberapa hal yang perlu dipahami untuk konsep GRID Layout:

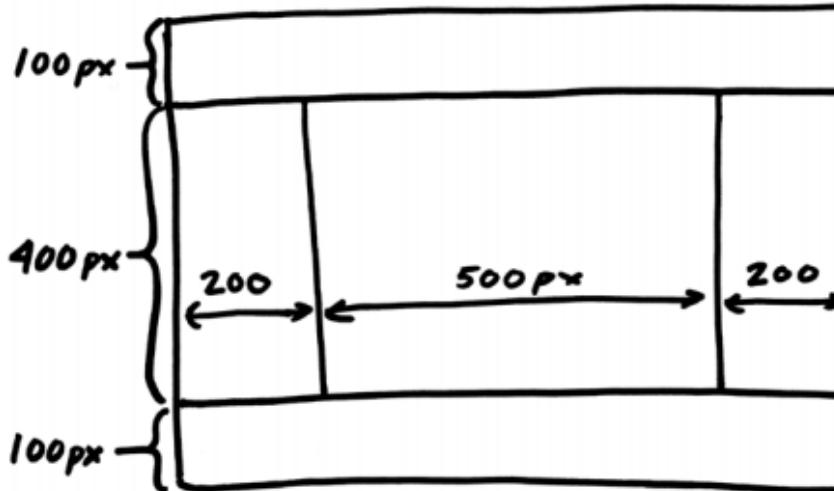
1. Grid Line
2. Grid Cell
3. Grid Area
4. Grid Track



# CSS – Deklarasi Grid

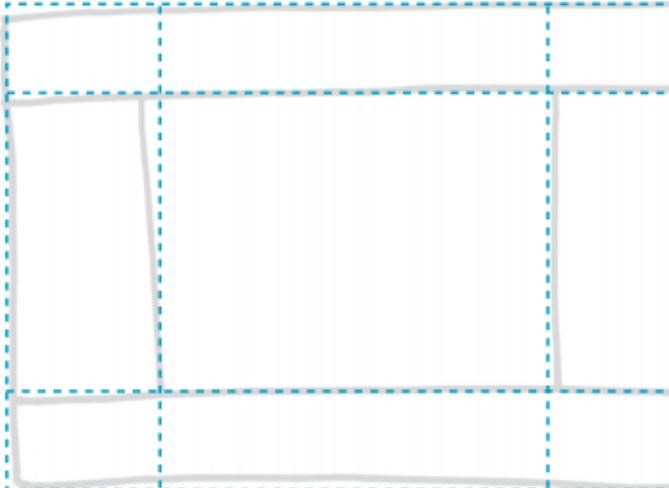
## THE MARKUP

```
<div id="layout">  
  <div id="one">One</div>  
  <div id="two">Two</div>  
  <div id="three">Three</div>  
  <div id="four">Four</div>  
  <div id="five">Five</div>  
</div>
```



## THE STYLES

```
#layout {  
  display: grid;  
}
```



# CSS – Grid Track

**grid-template-rows**

**grid-template-columns**

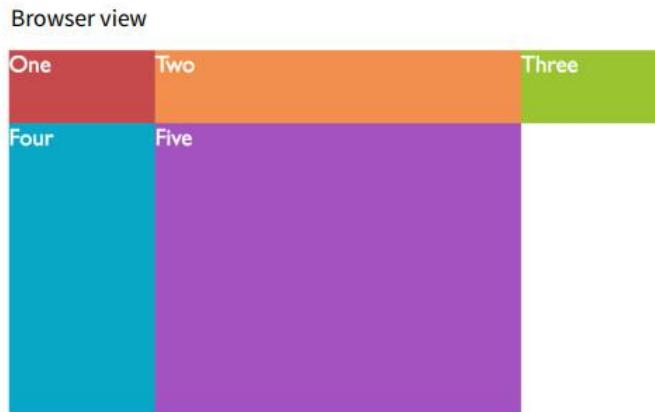
**Values:** none | *list of track sizes and optional line names*

**Default:** none

**Applies to:** grid containers

**Inherits:** no

```
#layout {  
  display: grid;  
  grid-template-rows: 100px 400px 100px;  
  grid-template-columns: 200px 500px 200px;  
}
```



```
<div id="layout">  
  <div id="one">One</div>  
  <div id="two">Two</div>  
  <div id="three">Three</div>  
  <div id="four">Four</div>  
  <div id="five">Five</div>  
</div>
```



# CSS – Grid :: Fractional Unit = fr

Unit satuan yang digunakan untuk mengatur jarak width sebuah GRID

```
#layout {  
    display: grid;  
    grid-template-rows: 100px 400px 100px;  
    grid-template-columns: 200px 1fr 200px;  
}
```



# CSS – Grid Template Area

Digunakan untuk mengatur area GRID

## grid-template-areas

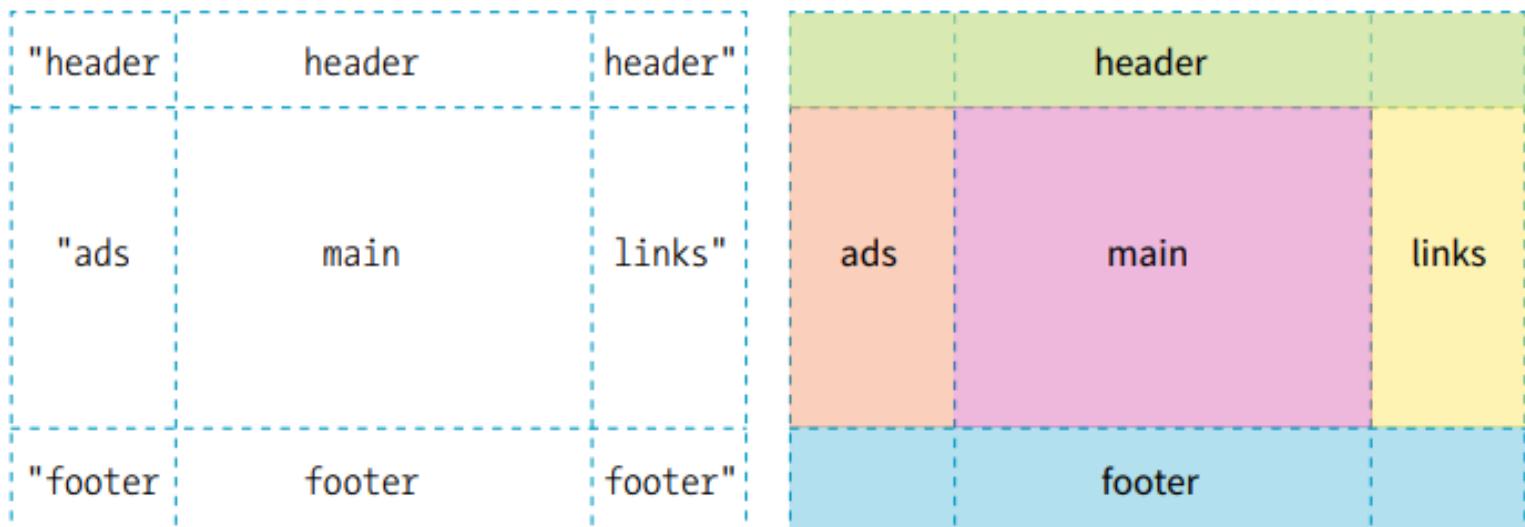
**Values:** none | series of area names

**Default:** none

**Applies to:** grid containers

**Inherits:** no

```
#layout {  
  display: grid;  
  grid-template-rows: [header-start] 100px [content-start] 400px  
  [footer-start] 100px;  
  grid-template-columns: [ads] 200px [main] 1fr [links] 200px;  
  grid-template-areas:  
    "header header header"  
    "ads     main     links"  
    "footer  footer  footer";  
}
```



# CSS – Grid Positioning

Digunakan menggambarkan lokasi item GRID dengan menentukan empat garis yang membatasi area grid target.

**grid-row-start**

**Values:** auto | *grid line* | span *number* | span '*line name*' | *number* '*line name*'

**grid-row-end**

**Default:** auto

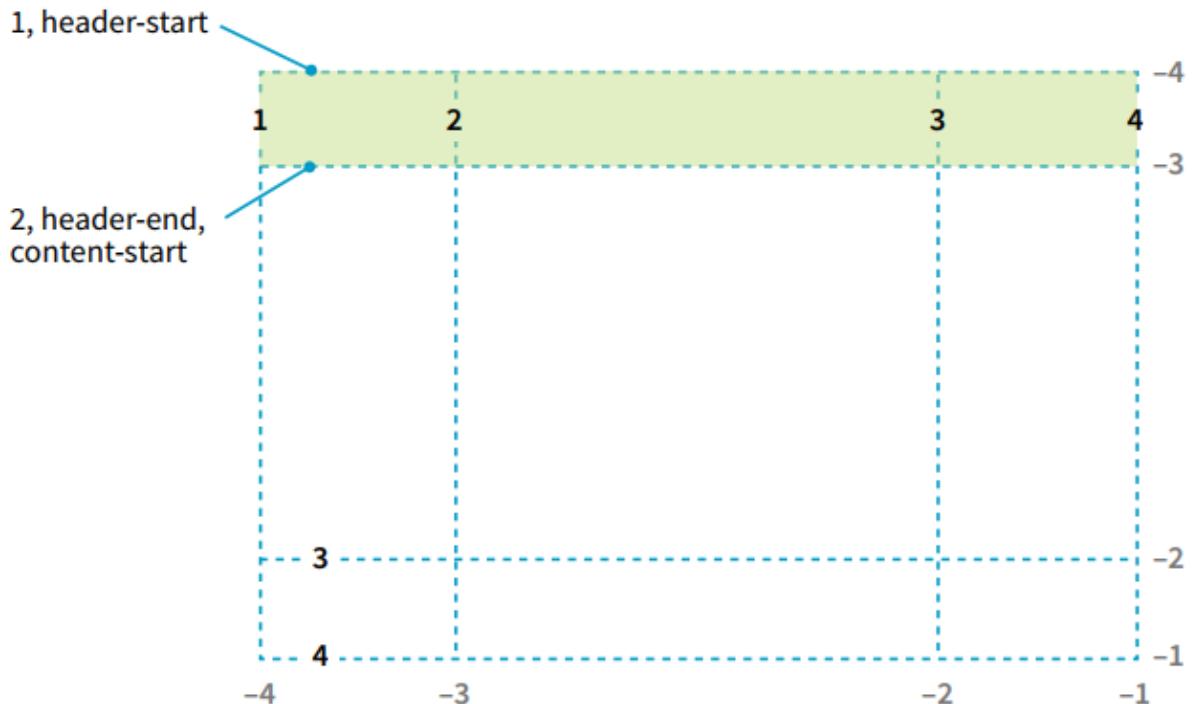
**grid-column-start**

**Applies to:** grid items

**grid-column-end**

**Inherits:** no

```
#one {  
    grid-row-start: 1;  
    grid-row-end: 2;  
    grid-column-start: 1;  
    grid-column-end: 4;  
}
```



# CSS – Grid Positioning by area

Digunakan menggambarkan lokasi item GRID dengan menentukan property area GRID

## grid-area

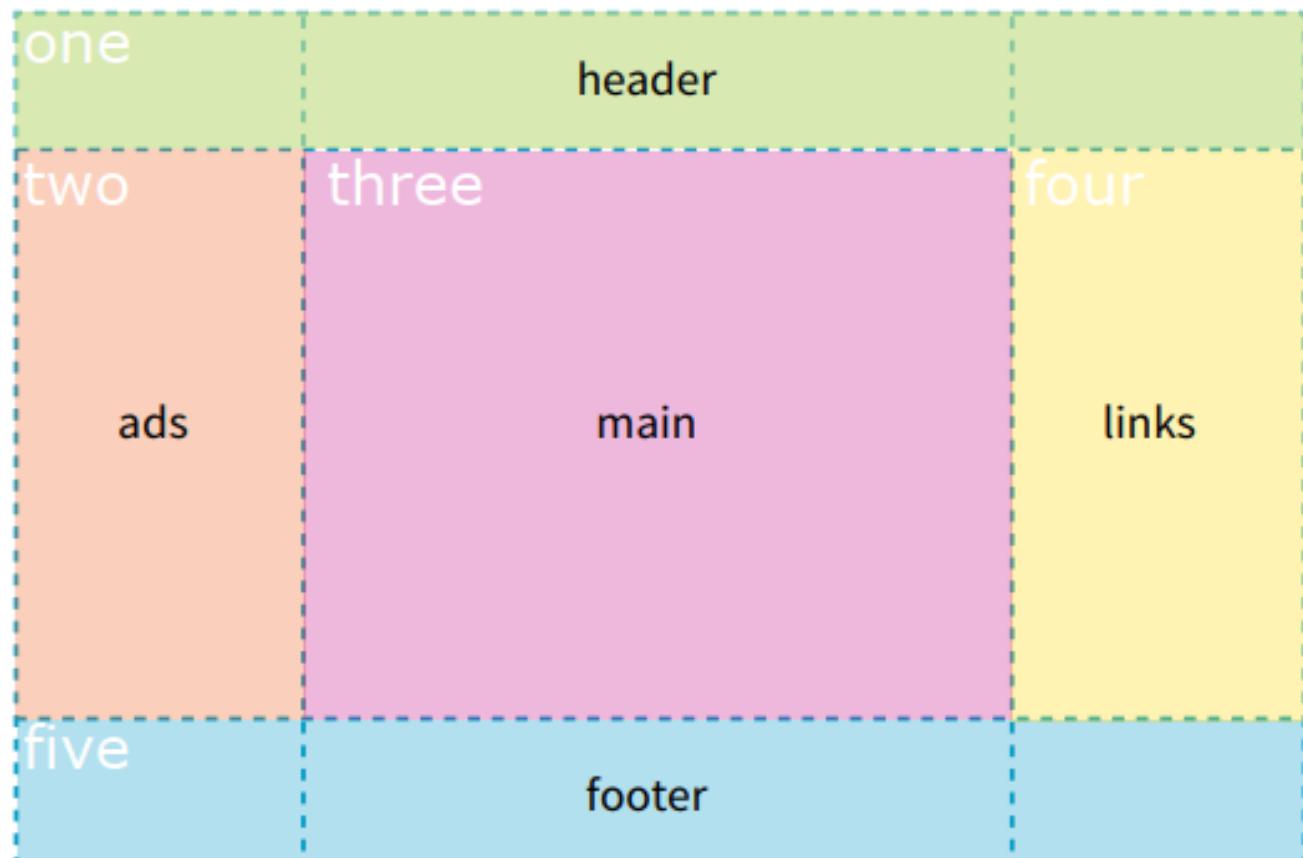
**Values:** *area name | 1 to 4 line identifiers*

**Default:** see individual properties

**Applies to:** grid items

**Inherits:** no

```
#one { grid-area: header; }
#two { grid-area: ads; }
#three { grid-area: main; }
#four { grid-area: links; }
#five { grid-area: footer; }
```



# CSS – Grid Auto Property

Digunakan membuat Grid Track column dan Rows

**grid-auto-rows**

**grid-auto-columns**

**Values:** *list of track sizes*

**Default:** auto

**Applies to:** grid containers

**Inherits:** no

## THE MARKUP

```
<div id="littlegrid">
  <div id="A">A</div>
  <div id="B">B</div>
</div>
```

## THE STYLES

```
#littlegrid {
  display: grid;
  grid-template-columns: 200px 200px;
  grid-template-rows: 200px 200px;
  grid-auto-columns: 100px;
  grid-auto-rows: 100px;
}

#A {
  grid-row: 1 / 2;
  grid-column: 2 / 3;
}

#B {
  grid-row: 3 / 4;
  grid-column: 5 / 6;
}
```



Digunakan menentukan flow item Grid berdasarkan column dan Rows

## grid-auto-flow

**Values:** row or column | dense *(optional)*

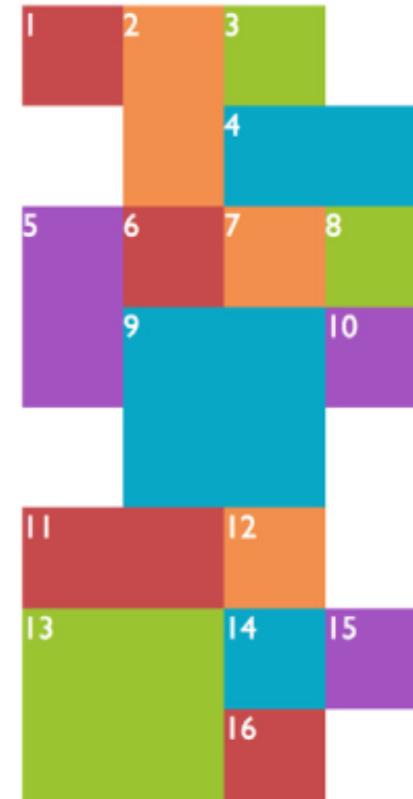
**Default:** row

**Applies to:** grid containers

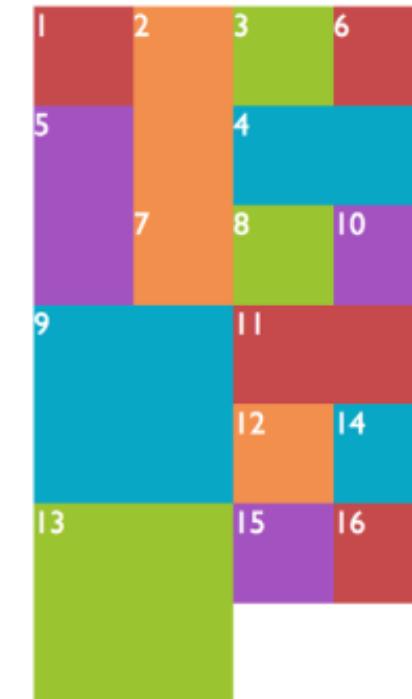
**Inherits:** no

```
#listings {  
  display: grid;  
  grid-auto-flow: column;  
}  
  
#listings {  
  display: grid;  
  grid-auto-flow: dense rows;  
}
```

Default flow pattern



Dense flow pattern



Digunakan menentukan jarak spasi antara item GRID

**grid-row-gap**

**grid-column-gap**

**Values:** *length (must not be negative)*

**Default:** 0

**Applies to:** grid containers

**Inherits:** no

**grid-gap**

**Values:** *grid-row-gap grid-column-gap*

**Default:** 0 0

**Applies to:** grid containers

**Inherits:** no

```
div#container {  
    border: 2px solid gray;  
    display: grid;  
    grid: repeat(4, 150px) / repeat(4, 1fr);  
    grid-gap: 20px 50px;  
}
```



*grid-row-gap: 20px;  
grid-column-gap: 50px;*

# CSS – Grid Alignment

Digunakan menentukan alignment antar item GRID

## justify-self

**Values:** start | end | center | left | right | self-start | self-end | stretch | normal | auto

**Default:** auto (looks at the value for justify-items, which defaults to normal)

**Applies to:** grid items

**Inherits:** no

## justify-self



# CSS – Grid Alignment

Digunakan menentukan alignment antar item GRID

## align-self

**Values:** start | end | center | left | right | self-start | self-end | stretch | normal | auto

**Default:** auto (looks at the value for align-items)

**Applies to:** grid items

**Inherits:** no

## align-self



# CSS – Grid Alignment

Digunakan menentukan alignment antar item GRID

## justify-content

**Values:** start | end | left | right | center | stretch | space-around | space-between | space-evenly

**Default:** start

**Applies to:** grid containers

**Inherits:** no

## justify-content:

start

One	Two	Three	Four	
Five	Six	Seven	Eight	
Nine	Ten	Eleven	Twelve	

end

One	Two	Three	Four	
Five	Six	Seven	Eight	
Nine	Ten	Eleven	Twelve	

center

One	Two	Three	Four	
Five	Six	Seven	Eight	
Nine	Ten	Eleven	Twelve	

space-around

One	Two	Three	Four	
Five	Six	Seven	Eight	
Nine	Ten	Eleven	Twelve	

space-between

One	Two	Three	Four	
Five	Six	Seven	Eight	
Nine	Ten	Eleven	Twelve	

space-evenly

One	Two	Three	Four	
Five	Six	Seven	Eight	
Nine	Ten	Eleven	Twelve	

# CSS – Grid Alignment

Digunakan menentukan alignment antar item GRID

## align-content

**Values:** start | end | left | right | center | stretch | space-around |  
space-between | space-evenly

**Default:** start

**Applies to:** grid containers

**Inherits:** no

### align-content:

start

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

end

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

center

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

space-around

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

space-between

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

space-evenly

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve