



#2

CSS Layout

Cascading Style Sheet

ASTROSPHERE | 2023

Tutorial CSS Layout #2:

- Flexbox
- Responsive

Flexbox

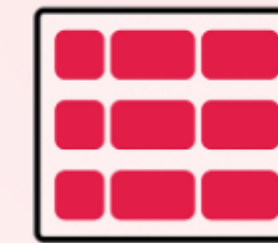
CSS Layout #2

Flexbox

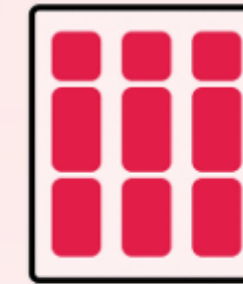
Flexbox adalah modul dalam CSS yang memungkinkan tata letak item dalam suatu kontainer (yang biasa disebut sebagai flex container) menjadi lebih fleksibel dan responsif. Flexbox memungkinkan pengaturan tata letak secara satu dimensi, baik itu baris (horizontal) atau kolom (vertikal), dan memberikan kontrol yang lebih besar terhadap penempatan, penyebaran, dan penyesuaian ukuran item dalam ruang yang tersedia.

CSS Flexbox

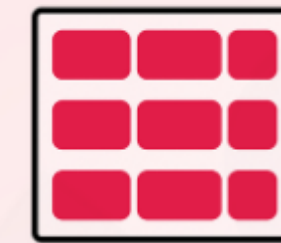
flex-direction



row



column



row-reverse



column-reverse

align-items



flex-start



center



flex-end



stretch

justify-content



flex-start



center



flex-end



space-between

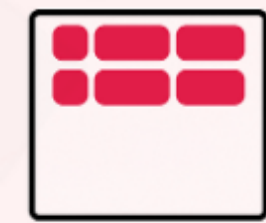


space-around

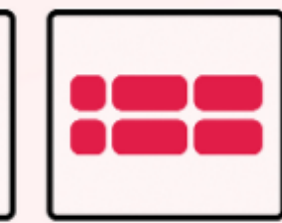


space-evenly

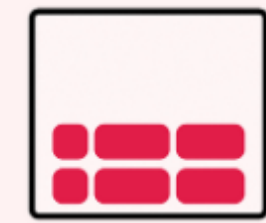
align-content



flex-start



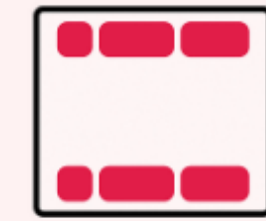
center



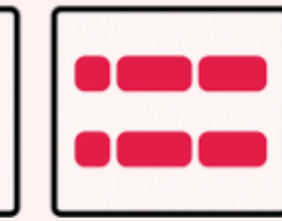
flex-end



stretch



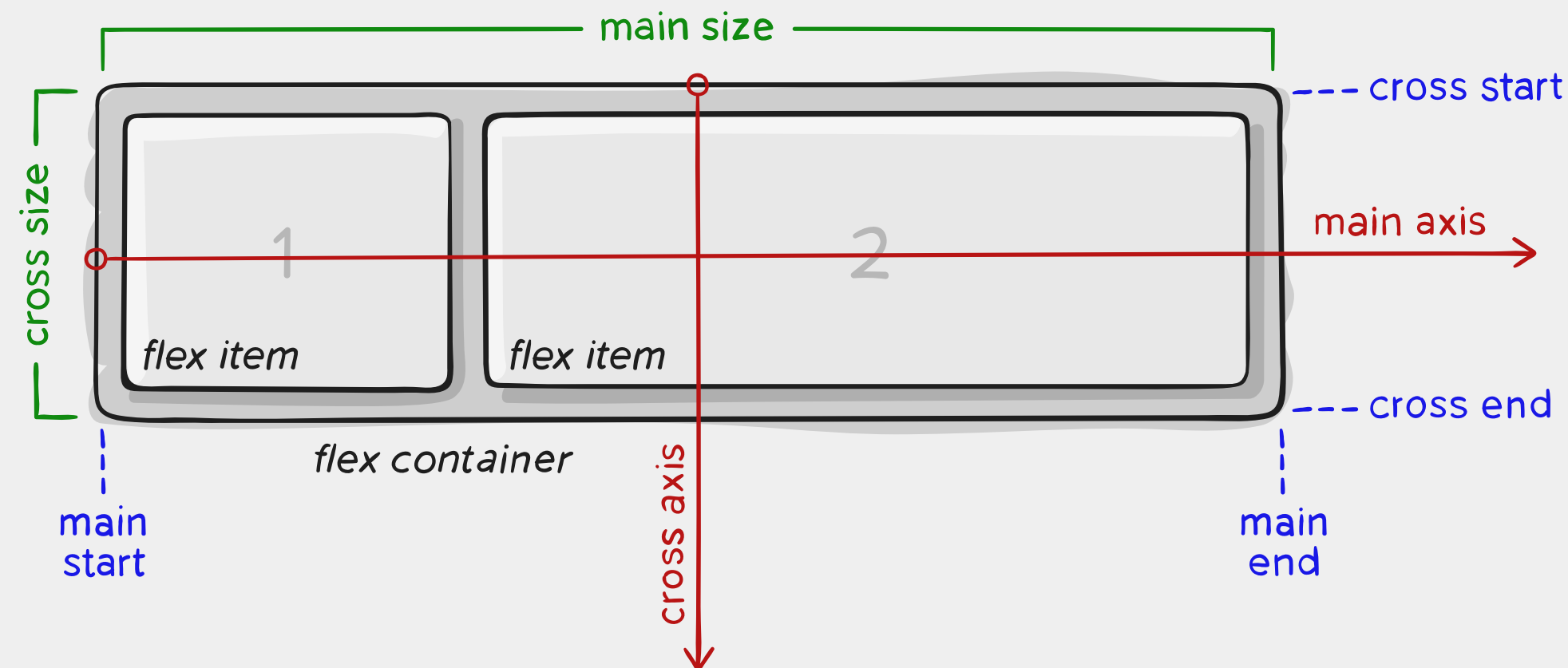
space-between



space-around

Basic Flexbox

Flexbox adalah keseluruhan modul dan bukan properti tunggal, Flexbox melibatkan banyak hal termasuk seluruh rangkaian propertinya. beberapa diantaranya adalah **Flex Container** dan **Flex Item**.

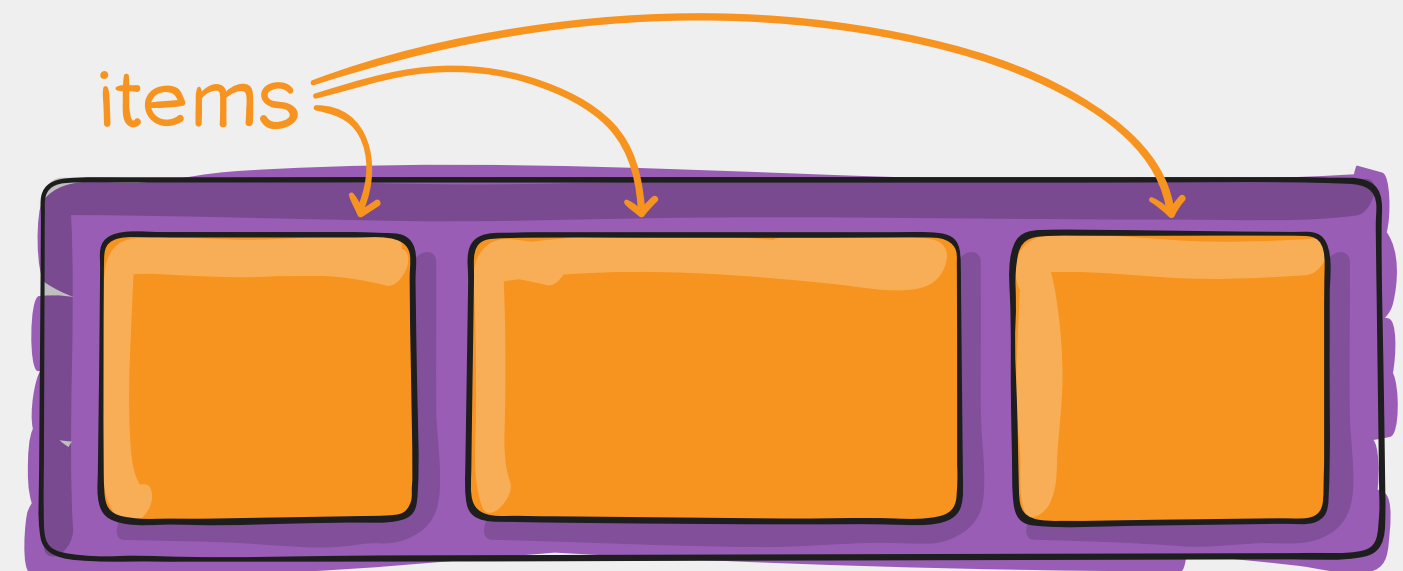
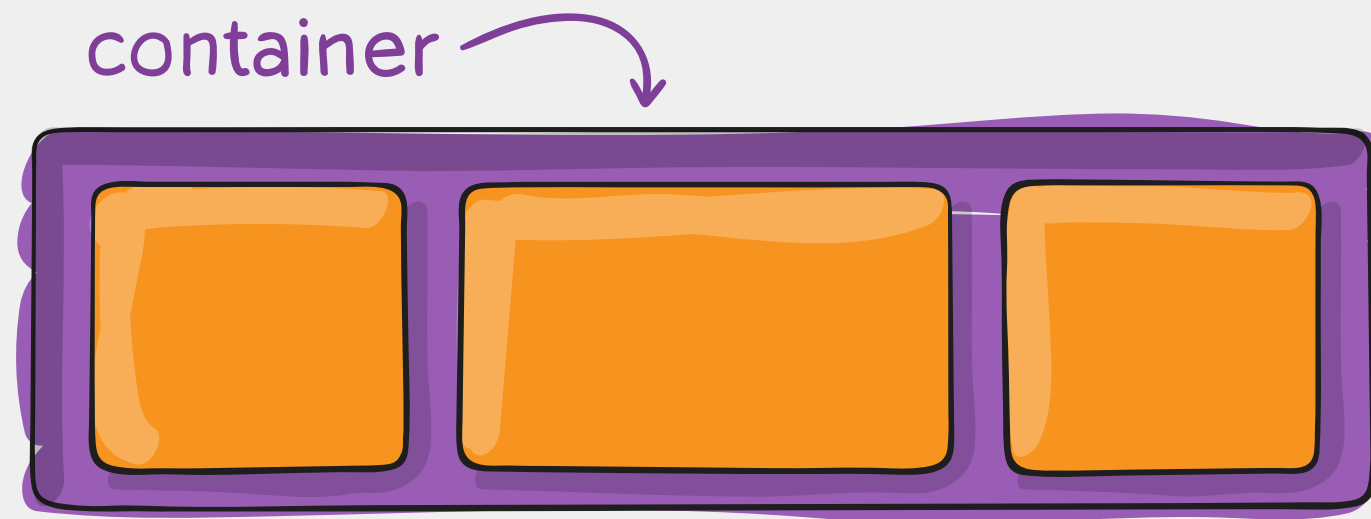


Item akan ditata mengikuti main axis (dari main start ke main end) atau cross axis (dari cross start hingga cross end).

Flexbox Properties

Flexbox properties dibedakan menjadi 2 jenis yaitu:

- Flex properties for the parent (flex container)
- Flex properties for the children (flex items)



Flex Container

display

properties ini digunakan untuk mendefinisikan flex container, inline atau block tergantung dari nilai yang diberikan



```
.container {  
  display: flex; /* or inline-flex */  
}
```

Flex Container

flex-direction

properties ini digunakan untuk mengatur arah main-axis, sehingga menentukan arah penempatan flex item dalam flex container.



```
.container {  
  flex-direction: row | row-reverse | column | column-reverse;  
}
```


Flex Container

flex-wrap

secara default semua flex item akan mencoba masuk ke dalam satu baris. Kita dapat mengubahnya dan mengizinkan item dibungkus sesuai kebutuhan dengan properti ini.



```
.container {  
  flex-wrap: nowrap | wrap | wrap-reverse;  
}
```

Flex Container

flex-flow

Ini adalah singkatan dari properti flex-direction dan flex-wrap



```
.container {  
  flex-flow: column wrap;  
}
```

Flex Container

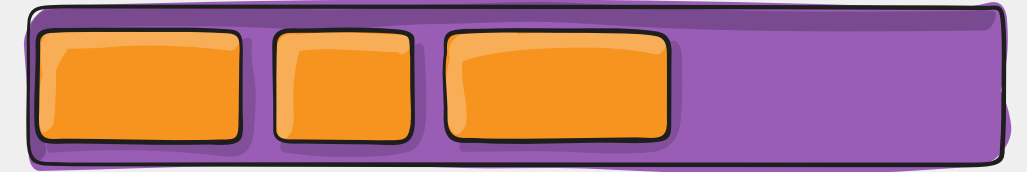
justify-content

Ini mendefinisikan perataan sepanjang main axis. Ini membantu mendistribusikan sisa ruang kosong ekstra ketika semua item fleksibel pada garis tidak fleksibel, atau fleksibel tetapi telah mencapai ukuran maksimumnya.

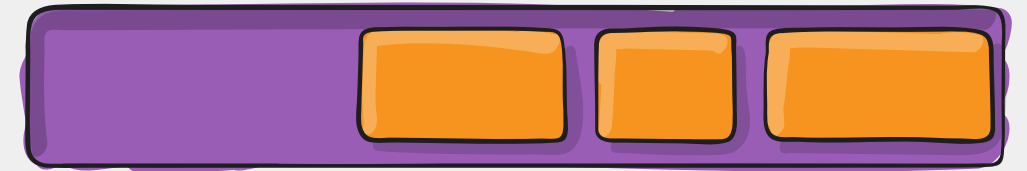


```
.container {  
  justify-content: flex-start | flex-end | center | space-between  
| space-around | space-evenly | start | end | left | right;  
}
```

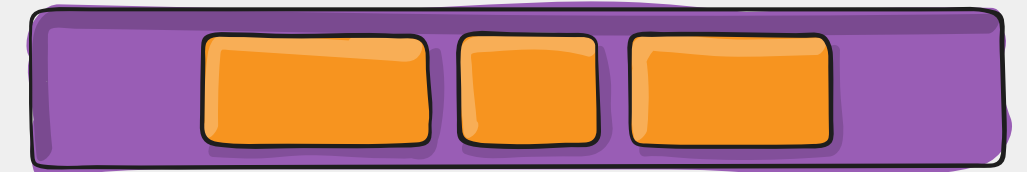
flex-start



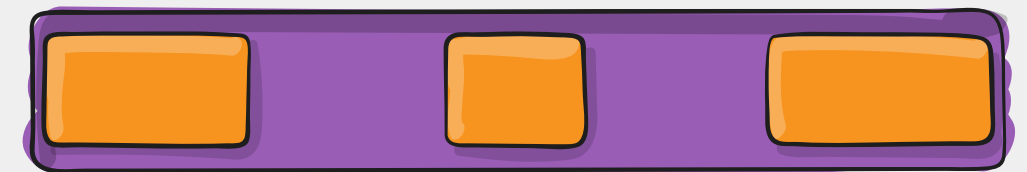
flex-end



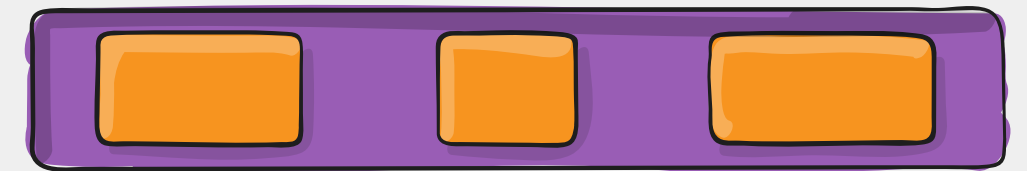
center



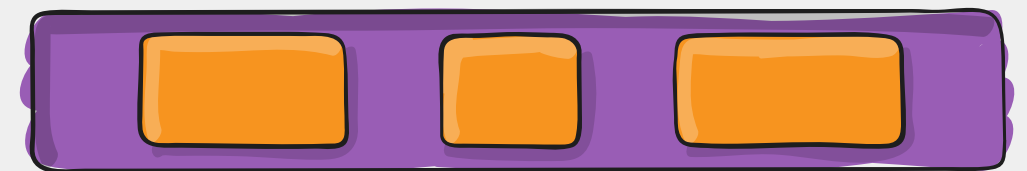
space-between



space-around



space-evenly



Flex Container

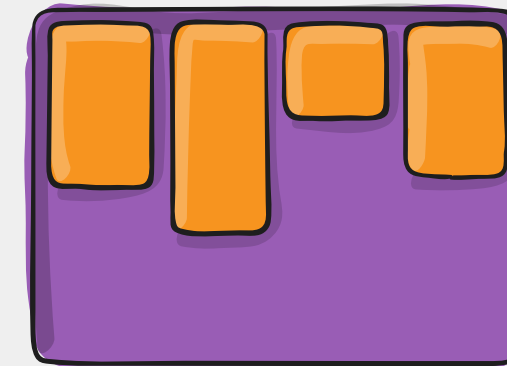
align-items

Ini mendefinisikan perilaku default bagaimana item fleksibel ditata sepanjang cross axis pada baris saat ini. Anggap saja sebagai versi justify-content untuk cross axis (tegak lurus dengan main axis).

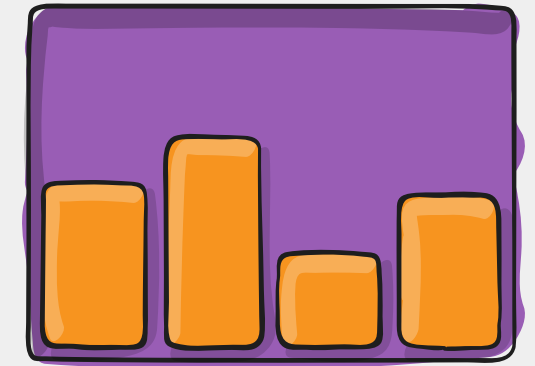


```
.container {  
  align-items: stretch | flex-start | flex-end | center | baseline |  
  first baseline | last baseline | start | end | self-start | self-end ;  
}
```

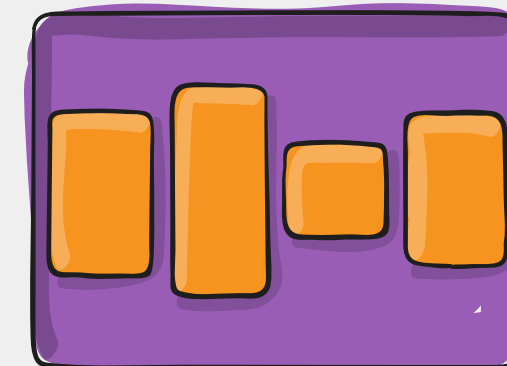
flex-start



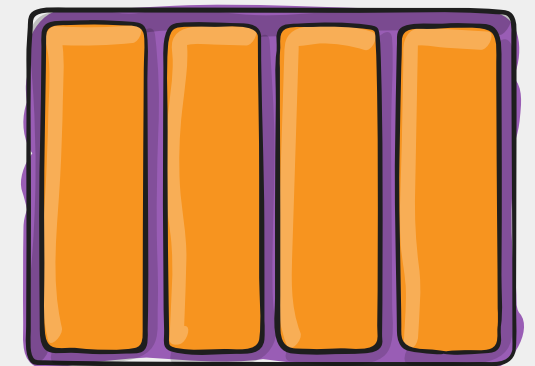
flex-end



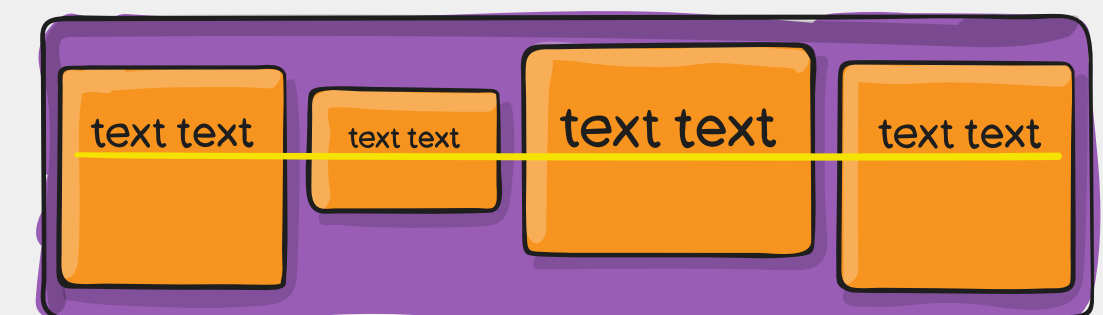
center



stretch



baseline



Flex Container

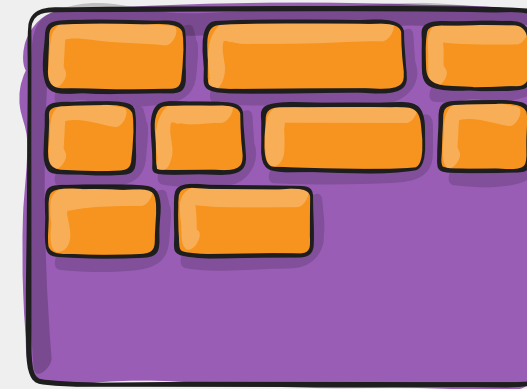
align-content

Ini menyelaraskan garis fleksibel container ketika ada ruang ekstra di cross axis, mirip dengan bagaimana justify-content menyelaraskan masing-masing item dalam main axis.

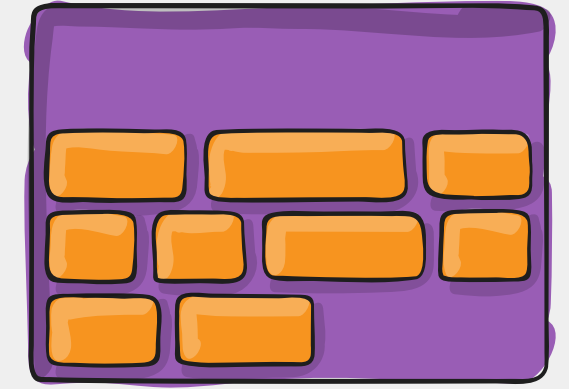


```
.container {  
  align-content: flex-start | flex-end | center | space-between |  
  space-around | space-evenly | stretch | start | end | baseline |  
  first baseline | last baseline ;  
}
```

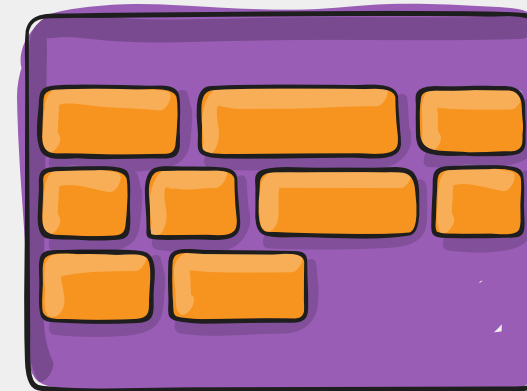
flex-start



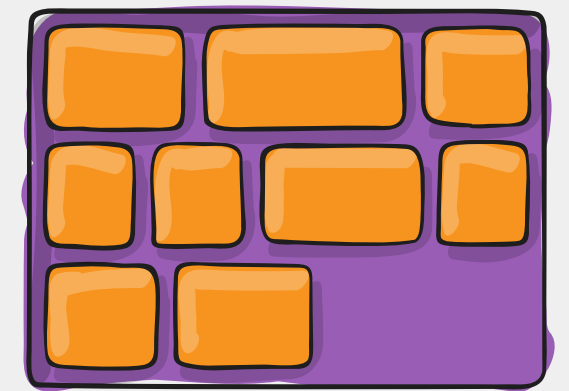
flex-end



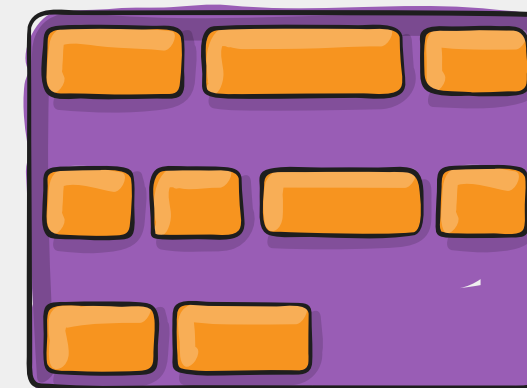
center



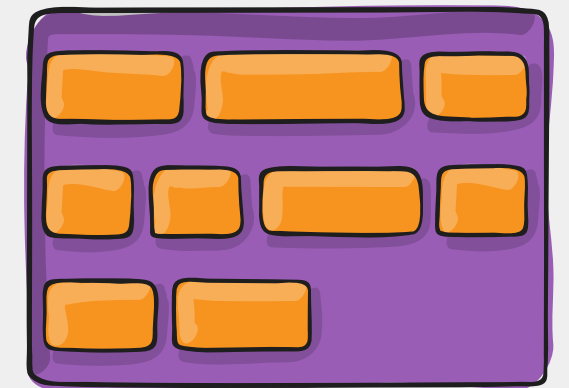
stretch



space-between



space-around



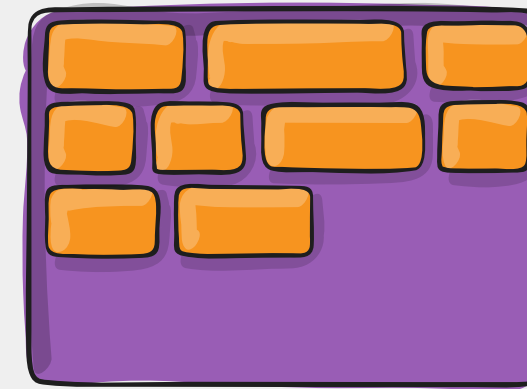
Flex Container

gap, row-gap, column-gap

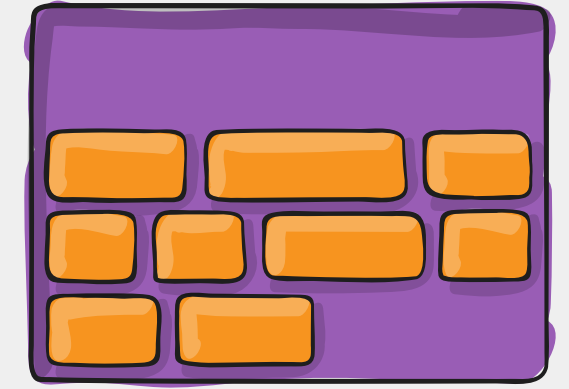
Properti gap secara eksplisit mengontrol jarak antar flex item. Yang berlaku adalah jarak antar item saja bukan pada tepi luarnya.

```
.container {  
  display: flex;  
  ...  
  gap: 10px;  
  gap: 10px 20px; /* row-gap column gap */  
  row-gap: 10px;  
  column-gap: 20px;  
}
```

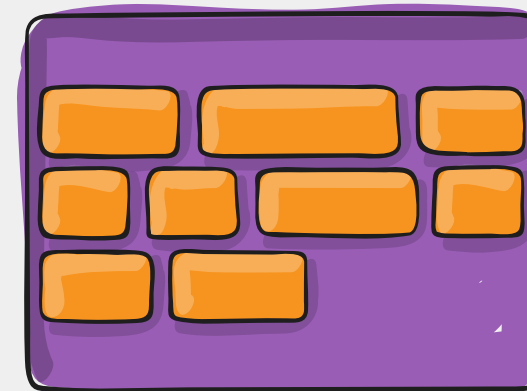
flex-start



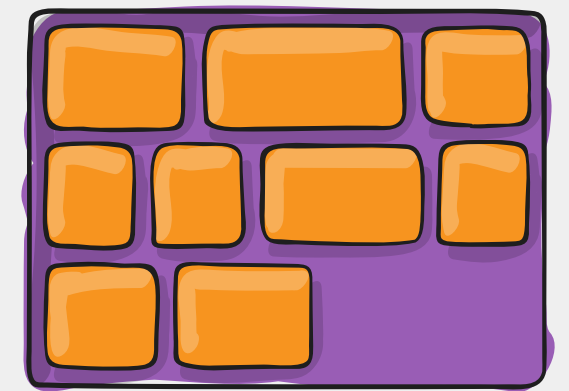
flex-end



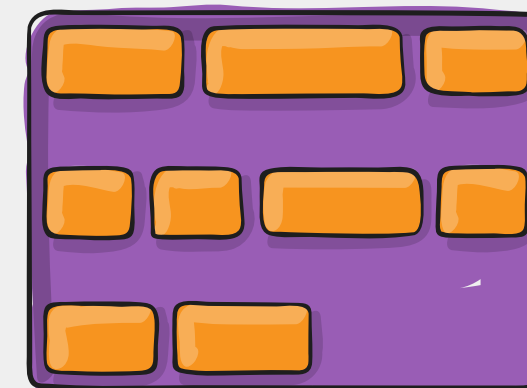
center



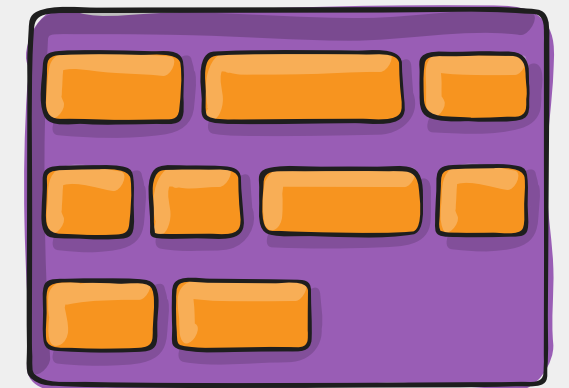
stretch



space-between



space-around



Flex Items

order

Secara default, flex item disusun dalam urutan sumber. Namun, properti order mengontrol urutan kemunculannya dalam wadah fleksibel.



```
.item {  
  order: 5; /* default is 0 */  
}
```

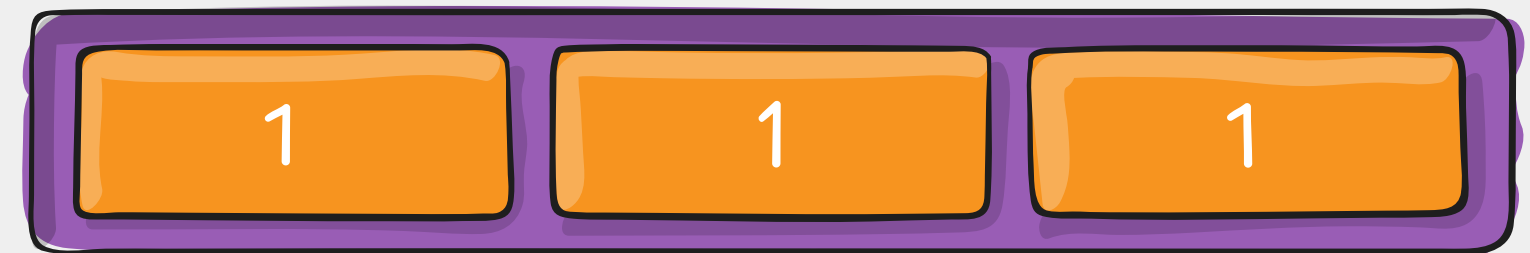
Flex Items

flex-grow

Ini mendefinisikan kemampuan flex item untuk berkembang jika diperlukan. Ia menerima nilai tanpa kesatuan yang berfungsi sebagai proporsi. Ini menentukan jumlah ruang yang tersedia di dalam wadah fleksibel yang harus digunakan oleh item tersebut.



```
.item {  
  flex-grow: 4; /* default 0 */  
}
```



Flex Items

flex-shrink

Ini mendefinisikan kemampuan flex item untuk menyusut jika diperlukan.



```
.item {  
  flex-shrink: 3; /* default 1 */  
}
```

Flex Items

flex-basis

Ini mendefinisikan ukuran default suatu elemen sebelum ruang yang tersisa didistribusikan.



```
.item {  
  flex-basis: | auto; /* default auto */  
}
```

Flex Items

flex

properties ini adalah properties singkat dari flex-grow, flex-shrink, dan flex-basis

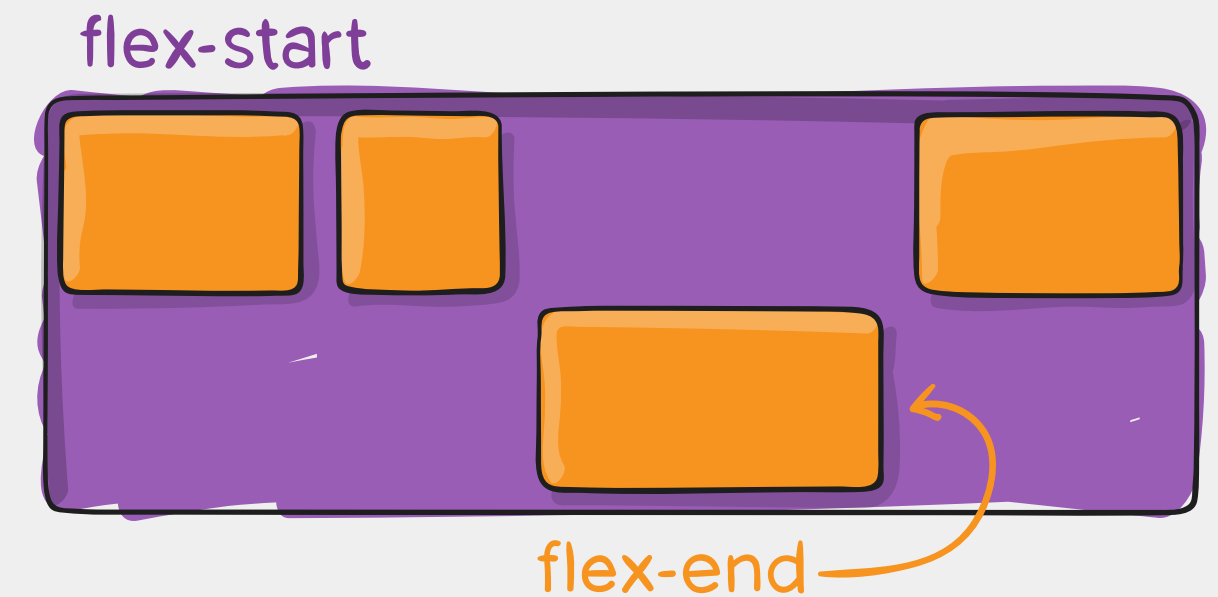


```
.item {  
  flex: 1 1 200px;  
}
```

Flex Items

align-self

Hal ini memungkinkan perataan default (atau yang ditentukan oleh align-items) untuk diganti untuk item fleksibel individual.



```
.item {  
  align-self: auto | flex-start | flex-end | center | baseline | stretch;  
}
```

Referensi

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

Responsive

CSS Layout #2

Responsive



Responsive Web Design, merupakan konsep pengaturan layout di mana ukuran dari tiap HTML element akan menyesuaikan dengan viewport yang digunakan. Konsep ini hanya menggunakan HTML and CSS dan tidak menggunakan Javascript. Responsive web design akan membuat halaman web rapih di semua ukuran layar device seperti gambar diatas.

Responsive

Viewport

Viewport adalah daerah pada layar yang menampilkan suatu konten. Dalam konteks kali ini, viewport adalah daerah yang menampilkan halaman web yang sedang kita akses. Ukuran viewport tidak selalu sama dengan resolusi layar device.

*Agar halaman web dapat menjadi responsif, dapat ditambahkan meta data berikut di dalam element **<head>** di file HTML*

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

- ***width=device-width*** memberikan instruksi kepada browser untuk mengikuti lebar layar dari device.
- ***initial-scale=1.0*** memberikan instruksi kepada browser untuk melakukan set nilai zoom level halaman tersebut menjadi 1.

Responsive

Media Queries

Media query adalah sebuah syntax yang diperkenalkan semenjak **CSS3**, yaitu dengan menggunakan **@media** yang akan mengindikasikan bahwa **CSS** tersebut hanya akan berlaku jika kondisi yang di set untuk **@media** bernilai **true**.

Sebagai contoh nih, kalau ukuran layar lebih kecil dari **600px** maka **background-color** akan bernilai **lightblue**:

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

Dengan menggunakan **media query**, kita bisa mengatur lebar suatu element dan/atau memberikan style lain yang berbeda-beda sesuai dengan ukuran dari browser. Atau biasanya konsep ini disebut **breakpoint**.

Responsive

Let's try to code!!! 🥰👉

Responsive

HTML

1

```
<div class="header">
  <h1>Chania</h1>
</div>

<div class="row">
  <div class="col-3 col-s-3 menu">
    <ul>
      <li>The Flight</li>
      <li>The City</li>
      <li>The Island</li>
      <li>The Food</li>
    </ul>
  </div>

  <div class="col-6 col-s-9">
    <h1>The City</h1>
    <p>Chania is the capital of the Chania
region on the island of Crete. The city can be
divided in two parts, the old town and the
modern city.</p>
  </div>
```

note:

angka didalam bulatan itu hanya untuk urutan code nya yak. code ini berada dalam 1 file html kokk

2

```
<div class="col-3 col-s-12">
  <div class="aside">
    <h2>What?</h2>
    <p>Chania is a city on the island of
Crete.</p>
    <h2>Where?</h2>
    <p>Crete is a Greek island in the
Mediterranean Sea.</p>
    <h2>How?</h2>
    <p>You can reach Chania airport from all
over Europe.</p>
  </div>
</div>
```

3

```
<div class="footer">
  <p>Resize the browser window to see how the
content respond to the resizing.</p>
</div>
```

Responsive

CSS

1

```
* {
  box-sizing: border-box;
}

.row::after {
  content: "";
  clear: both;
  display: table;
}

[class*="col-"] {
  float: left;
  padding: 15px;
}

html {
  font-family: "Lucida Sans", sans-serif;
}

.header {
  background-color: #9933cc;
  color: #ffffff;
  padding: 15px;
}
```

2

```
.menu ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}

.menu li {
  padding: 8px;
  margin-bottom: 7px;
  background-color: #33b5e5;
  color: #ffffff;
  box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0
1px 2px rgba(0,0,0,0.24);
}

.menu li:hover {
  background-color: #0099cc;
}

.aside {
  background-color: #33b5e5;
  padding: 15px;
  color: #ffffff;
  text-align: center;
  font-size: 14px;
  box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0
1px 2px rgba(0,0,0,0.24);
}
```

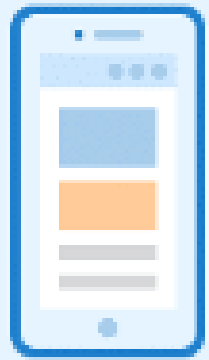
*lanjut next slide yakk, code
disebelah more important supaya
result nya jadi responsive!!!*

3

```
.footer {
  background-color: #0099cc;
  color: #ffffff;
  text-align: center;
  font-size: 12px;
  padding: 15px;
}
```

Responsive CSS

Smartphone



1

```
/* For mobile phones: */  
[class*="col-"] {  
  width: 100%;  
}
```

```
@media only screen and (min-width: 600px) {  
  /* For tablets: */  
  .col-s-1 {width: 8.33%;}  
  .col-s-2 {width: 16.66%;}  
  .col-s-3 {width: 25%;}  
  .col-s-4 {width: 33.33%;}  
  .col-s-5 {width: 41.66%;}  
  .col-s-6 {width: 50%;}  
  .col-s-7 {width: 58.33%;}  
  .col-s-8 {width: 66.66%;}  
  .col-s-9 {width: 75%;}  
  .col-s-10 {width: 83.33%;}  
  .col-s-11 {width: 91.66%;}  
  .col-s-12 {width: 100%;}  
}
```

2

Tablet



```
@media only screen and (min-width: 768px) {  
  /* For desktop: */  
  .col-1 {width: 8.33%;}  
  .col-2 {width: 16.66%;}  
  .col-3 {width: 25%;}  
  .col-4 {width: 33.33%;}  
  .col-5 {width: 41.66%;}  
  .col-6 {width: 50%;}  
  .col-7 {width: 58.33%;}  
  .col-8 {width: 66.66%;}  
  .col-9 {width: 75%;}  
  .col-10 {width: 83.33%;}  
  .col-11 {width: 91.66%;}  
  .col-12 {width: 100%;}  
}
```

3

Desktop





**THANKS FOR
JOINING THE CLASS!!**

@programming.tadulako | protad123@gmail.com

#GrowTogetherWithUs