



# Mengenal HTML dan CSS

**Silver**- Chapter 2 - Topic 2

---

**Selamat datang di Chapter 2 Topic 2 pada course  
React Native dari Binar Academy!**





# Haalloo Guys! 🙌

Kita lanjut yuk ke **Chapter 2 Topik 2 React Native**. Di Topic 1 kemarin kita udah belajar tentang Front-End, Back-End, dan cara kerja internet, nah kali ini kita akan belajar tentang **HTML** dan **CSS**.

Yuk, langsung aja kita kepoin materi ini.





## Detailnya, kita bakal bahas hal-hal berikut ini:

- Cara Kerja HTML dan CSS
- Pembuatan tampilan layout HTML dan CSS
- Pembuatan tampilan layout dengan FlexBox



**Guys, kamu pernah lihat atau ikut tes simulasi berkendara ketika bikin SIM kan?**

**Di simulasi itu jalannya dibuat semirip mungkin dengan kondisi jalan yang asli.**

**Jadi, kamu akan lebih mudah ketika nanti berkendara langsung di jalan raya.**





Nah sama kayak simulasi berkendara, sebelum kamu bikin layout langsung di react native, sebaiknya kamu pelajari dulu HTML dan CSS. Hmm.. kenapa ya?

Karena **konsep HTML dan CSS sama persis dengan proses bikin layout di react native**, tapi di HTML dan CSS lebih mudah dipahami, guys! 😊





kalau udah jago tentang HTML dan CSS,  
kamu bakal dapet SIM guys! **Surat Izin  
Mengoding~**



Biar bisa ngasih tahu browser untuk menyusun sebuah halaman web dan menampilkannya, kamu harus paham bahasanya.

Nah, **HTML** adalah bahasa yang bisa kamu gunakan 😊



Salah satu website yang pakai HTML adalah Tokopedia. Ups, tapiiii bukan cuma Tokopedia aja yang dibikin pakai HTML lho!

Faktanya, **semua Website yang ada di muka bumi ini dibuat pakai HTML.**





Mantep ya HTML. Tapi..

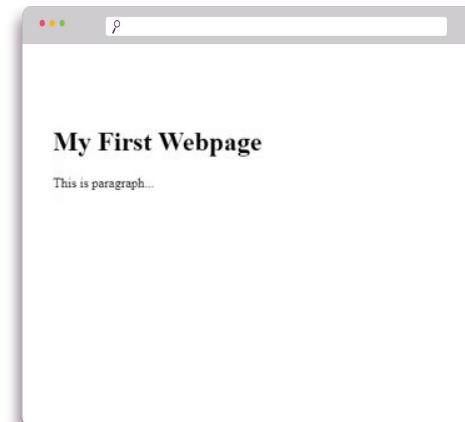
Apa sih sebenarnya HTML ini? Fungsinya buat apa?

**Markup language** adalah sebuah **format dalam menulis kode yang akan diproses menjadi sebuah tampilan**. Karena **HTML adalah bahasa markup**, jadi bisa dipakai buat ngasih tahu browser untuk menyusun sebuah halaman web dan bagaimana browser akan menampilkannya.

```
•••\n\n<!DOCTYPE html>\n<html>\n\n    <head>\n        <title>My First Webpage</title>\n    </head>\n\n    <body>\n        My First Webpage\n        <h1>\n            My First Webpage\n        </h1>\n        <p>This is a paragraph...</p>\n    </body>\n\n</html>
```



Kode dikonversi  
jadi tampilan





## Di dalam HTML ada juga istilah Tag HTML. Kayak gini penjelasannya!

Tag **HTML** adalah semacam **kode yang dibungkus dengan karakter Enclosing Tag (<>)**.

Tag ini biasanya dipakai untuk membungkus teks lain atau bahkan suatu konten. Tag juga dipakai untuk menampilkan sesuatu di dalam browser.





## Contohnya ketika kamu mau nampilin teks “Hello World” di browser

Saat kamu mau menampilkan teks “Hello World” di browser, **kamu perlu pakai tag yang akan membungkus teks “Hello World”** tersebut. Jadi di dalam dokumen HTML akan terlihat kayak gini:

```
<h1>Hello world</h1>
```





**Tag HTML nggak cuma <h1> saja.**

**Ini dia beberapa tag HTML lain dan fungsinya!**

Nama Tag	Fungsinya
<h1>	Menampilkan header text
<h2>	Menampilkan header text yang ukurannya lebih kecil dari h1
<p>	Menampilkan paragraph
<strong>	Membuat text menjadi tebal
<i>	Membuat text menjadi mirin
<img>	Menampilkan sebuah gambar
<u>	Membuat teks garis bawah



Sekarang coba perhatikan, kayak gini nih contoh penggunaan tag-tag HTML lainnya, guys!

Kode HTML

```
● ● ●  
1 <h1>Ini text dengan H1 </h1>  
2 <i>Ini text tag i, cetak teks garis miring</i>  
3 <p>Ini text dengan p (paragraph) </p>  
4 <strong>Ini Text tag strong, cetak tebal</strong>  
5 <h2>Ini text dengan h2</h2>  
6 <u>Ini text tag i, cetak teks garis miring</u>
```

Hasil Tampilan

● ● ●  
**Ini text dengan H1**  
*Ini text tag i, cetak teks garis miring*  
Ini text dengan p (paragraph)  
**Ini Text tag strong, cetak tebal**  
**Ini text dengan h2**  
Ini text tag i, cetak teks garis miring



# HTML nggak cuma bisa nampilin teks seperti “Hello World”, tapi juga bisa nampilin gambar lho!

Hah? Memang bisa nampilin gambar pake HTML? Bisa banget, Sob!

Untuk nampilin gambar, kamu perlu **menggunakan tag <img>**.

Tag <img>ini adalah **tag kosong yang hanya memiliki attribute dan nggak punya closing tag**.



```
1 
```

Attribute



### Lalu gimana caranya nampolin gambar pakai HTML yaa?

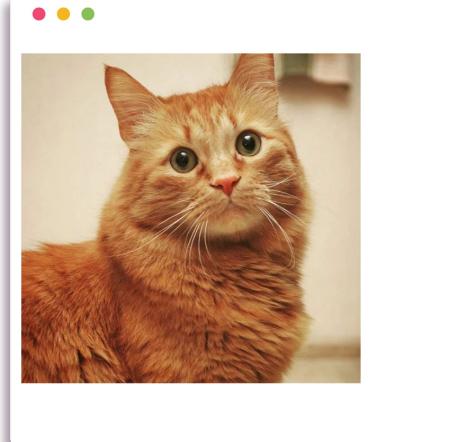
Jadi guys, untuk menampilkan gambar dengan HTML, kamu cuma perlu mengisi **nilai attribute src** dengan **link dari gambar** yang ingin kamu tampilkan. Nanti gambar akan muncul di situ. Gampang deh pokoknya.

Kode HTML



```
1 
```

Hasil Tampilan





# Kamu disuruh bikin form pendaftaran vaksin warga sama Pak RT? Jangan panik! Bikin pakai HTML juga bisa lho.

Form di HTML bisa kamu bikin dengan tag `<form>`

Tag `<form>` akan berfungsi sebagai container tag. Apa itu container tag?

**Container tag** adalah tag html yang penempatannya sebagai “bungkus” tag-tag html lainnya.





# Lalu gimana caranya menampilkan form pakai HTML yaa?

Untuk menampilkan sebuah form, kamu harus menuliskan tag `html <input>` di dalam tag `<form>`.

Tag `<input>` digunakan untuk **menentukan seberapa banyak elemen form** yang ingin ditampilkan.

Kode HTML



```
1 <h2>Form Pendaftaran diri</h2>
2 <form>
3   <input type="text" />
4 </form>
```

Tag `<input>`  
diletakan  
di dalam  
tag `<form>`

Hasil Tampilan



Form Pendaftaran diri

Hasil tampilan  
tag `<input>`



Lihat contoh coding pembuatan form di bawah ini deh.

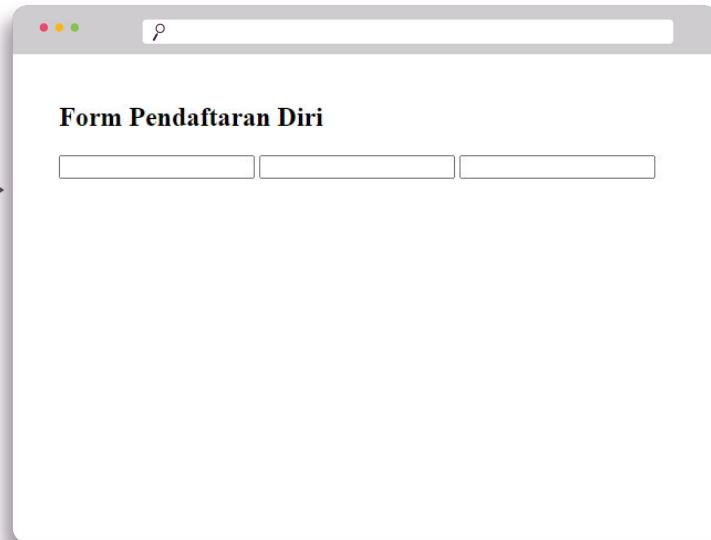
Kalau kamu menuliskan tag **input** sebanyak 3 kali, maka yang akan muncul adalah form input sebanyak 3 kali.

```
macOS window showing HTML code:  


## Form Pendaftaran Diri

```

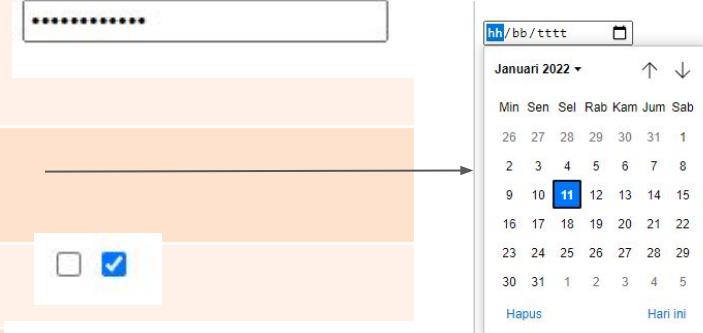




## Tag Input juga punya banyak tipe attributes, guys!

Tabel di bawah ini hanya menampilkan beberapa saja, kalau kamu mau lihat lengkapnya silahkan buka [di sini](#) ya!

Attributes Value	Fungsi	Tampilan
Text	Untuk bikin form input dengan jenis text	<input type="text" value="Ini text"/>
Password	Untuk bikin form input dengan jenis text password, sehingga apa yang diketik di form nggak akan ditampilkan	<input type="password" value="*****"/>
Date	Untuk bikin form input dengan menampilkan kalender	<input type="date"/>
Checkbox	Untuk bikin form input dengan menampilkan pilihan checkbox	<input type="checkbox"/> <input checked="" type="checkbox"/>
File	Untuk bikin form input yang bertujuan untuk upload file	<input type="file"/> Pilih File Tidak ada file yang dipilih





## Form Pendaftaran Diri

Nama Depan:

Nama Belakang:

Nomor Handphone:

Tanggal Lahir:

 hh/bb/tttt 

Jenis Kelamin:

Laki-Laki  Perempuan

Hobi:

- Makan
- Rebahan
- Nonton

## Okee, saatnya latihan~

Coba kamu buat form pendaftaran diri kayak gambar di samping dengan menggunakan HTML.

Selamat mencoba! 😊



**Guysss, nggak cuma HTML, pemahaman tentang cara kerja **CSS (Cascading Style Sheet)** juga jadi mekanisme penting pada pembuatan website.**

**Kenapa ya? Ini alasannya!**



**HTML** ibarat sebuah rumah yang belum di-finishing, guys, dan **CSS** adalah desain interior rumahnya.

Kamu bisa bayangin dong.. tanpa desain interior yang indah dan nyaman, rumah yang sudah dibikin pasti bakal terlihat membosankan.





## Biar lebih jelas, coba perhatikan perbedaan dua tampilan web di samping deh

Website kedua kelihatan lebih ciampik, kan? Yap~ itu karena sudah diolah pakai CSS. Kalau web pertama, cuma pakai HTML saja aja, guys!

CSS atau **Cascading Style Sheet** merupakan mekanisme yang mengatur gaya atau style pada halaman web agar memiliki visual yang menarik.





## Jadi ingat ya, agar tampilan web ciamik, HTML dan CSS harus collab!

Soalnya, kalau web cuma dibuat menggunakan HTML tanpa dipoles CSS, hasilnya akan seperti gambar di samping. Kelihatan boring dan nggak menarik sama sekali.





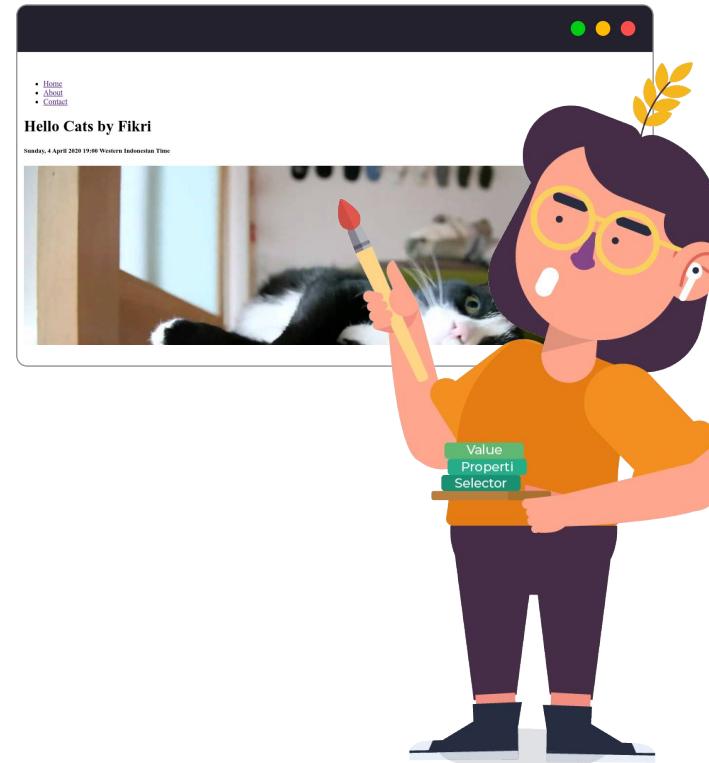
**Terus, gimana sih cara HTML dan CSS bekerjasama?**



## Caranya kayak gini, guys!

Saat dokumen HTML dibuat, aturan CSS perlu diterapkan di dalamnya untuk menentukan bagaimana cara dokumen itu akan ditampilkan.

Nah, aturan CSS itu ada tiga, yaitu **selector**, **properti**, dan **value**. Ketiganya juga disebut dengan **struktur** atau **deklarasi**.





Coba deh kamu perhatikan kolom di bawah ini. Di situ ada struktur CSS yaitu **Selector, Property dan Value**. Masing-masing struktur punya fungsi masing-masing dalam pembuatan program.

**selector { property : value }**

Contoh Kode

**h1 { color : blue }**



```
h1 {  
    font-family : "Arial";  
    text-align : center;  
    font-size : 30px;  
    color : blue;  
}
```



Mari kita bahas satu per satu yuk~

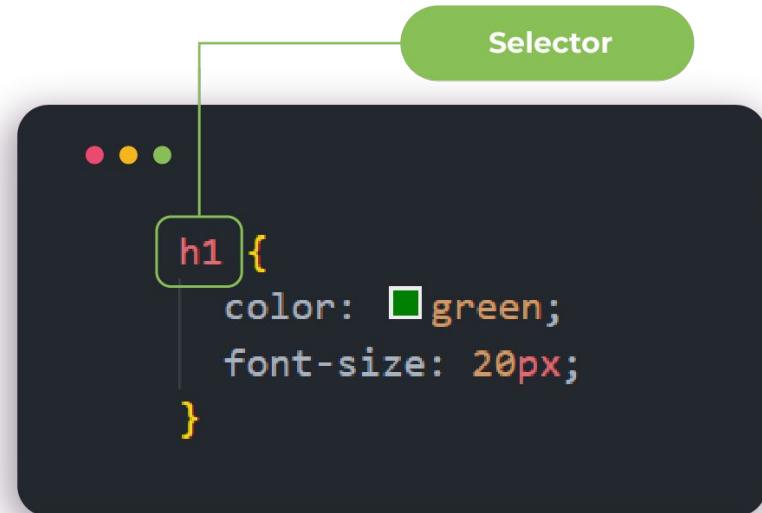




## CSS – Selector

Selector dipakai untuk **memilih dan memanipulasi elemen spesifik pada HTML**. Elemen HTML bisa dipilih berdasarkan tag, id, class bahkan pola / pattern.

Semakin kompleks struktur HTML, maka selector juga bisa semakin kompleks / spesifik.

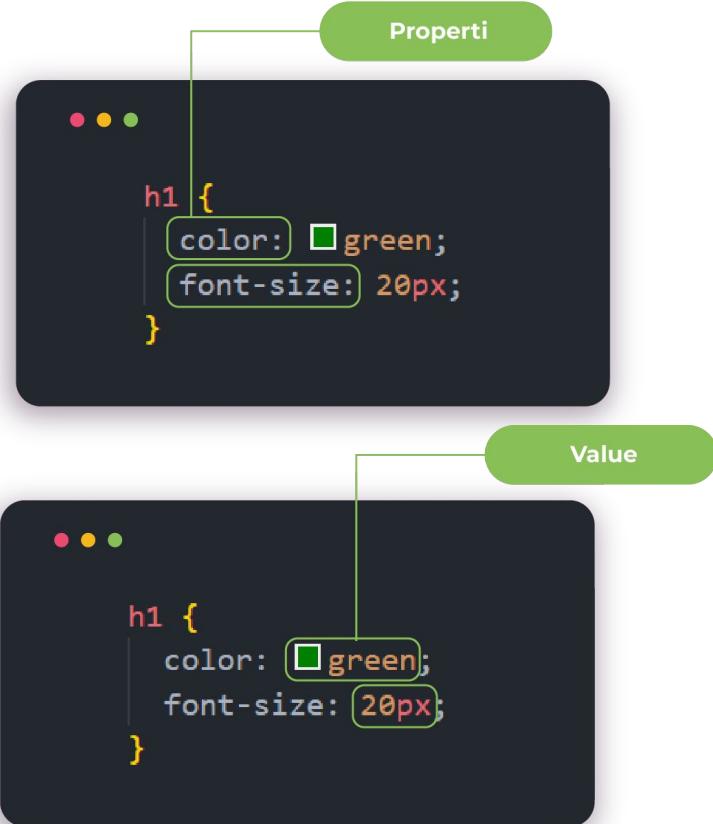




## CSS – Property dan Value

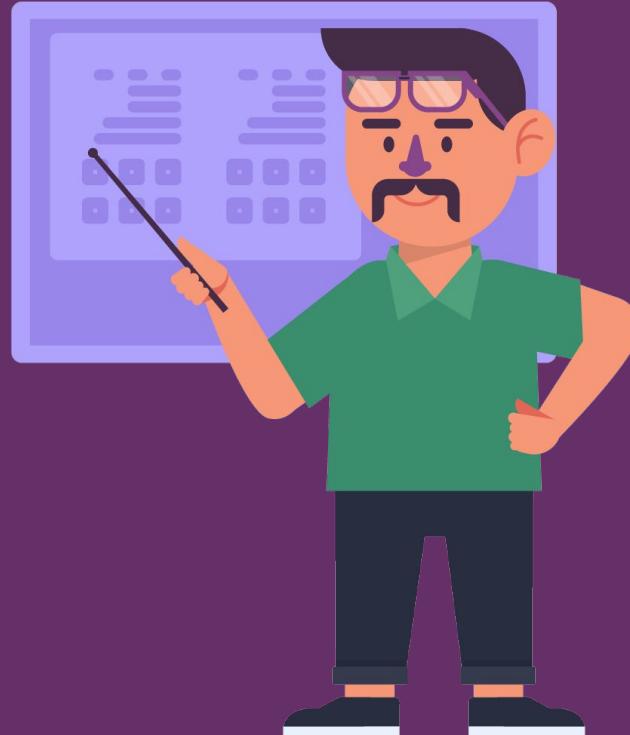
**Property**, untuk menentukan alat atau property apa yang akan dipakai untuk mengubah elemen spesifik pada HTML.

**Value**, untuk memilih nilai apa yang akan digunakan pada property yang dipakai.





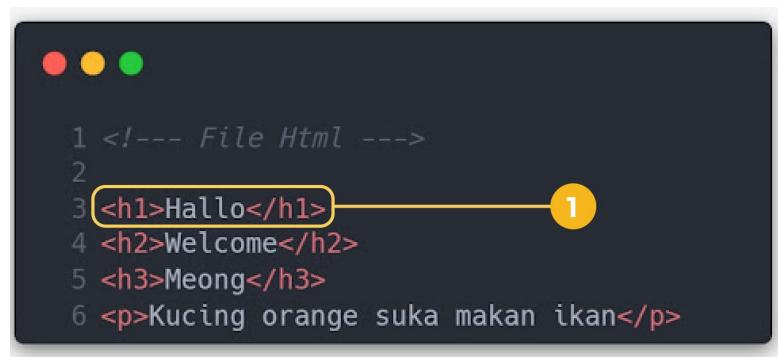
**Supaya lebih gampang dipahami, coba perhatikan simulasi tentang selector, property dan value berikut ini dulu yuk~**





## Kita mulai dari file HTML

Misalnya kamu mau **mengatur warna yang ada pada tag "h1"** di file HTML. Kalau di sini berarti kata "Hallo" ya, guys!



```
1 <!-- File Html --->
2
3 <h1>Hallo</h1>1
4 <h2>Welcome</h2>
5 <h3>Meong</h3>
6 <p>Kucing orange suka makan ikan</p>
```



Maka di File CSS harus nulis kayak gini, Sob:

```
h1 {  
    color: blue;  
}
```

di mana “**h1**” **berfungsi sebagai selector** untuk memilih elemen (kata “Hallo”) yang ingin kamu atur warnanya, “**Color sebagai property** atau alat untuk mengatur warna dari kata “Hallo”, dan “**Blue**” **sebagai value** atau nilai yang digunakan untuk mengatur kata “Hallo” menjadi warna biru.



The screenshot shows a dark-themed code editor window. At the top, there are three circular icons: red, yellow, and green. Below them is a line of code:

```
1 /* File Css */  
2  
3 h1 {  
4     color: blue;  
5 }  
6  
7 h3 {  
8     color: orange;  
9 }  
10  
11 p {  
12     color: lightgreen;  
13 }
```

A yellow oval-shaped annotation highlights the selector "h1 {" in line 3. A yellow circle with the number "2" is positioned to the right of the annotation line, connected by a thin yellow line. The rest of the code is displayed in a standard monospaced font.



Walla, di tampilan web akan jadi kayak gini guys!





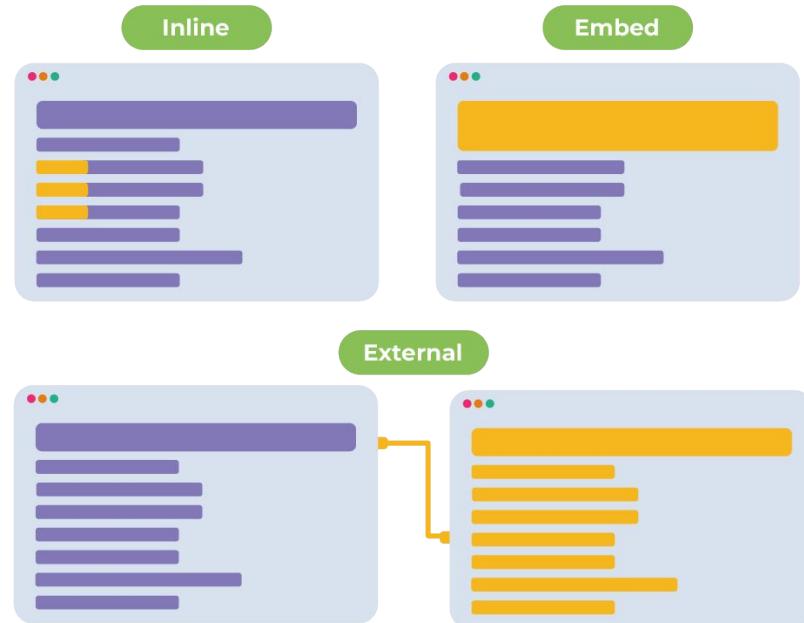
Di atas tadi sudah dikasih tahu kan, HTML dan CSS kan harus collab agar tampilan web lebih bagus, .

Nah, cara mengkolaborasikan mereka adalah dengan menempatkan CSS pada **HTML**, guys!



## Ada tiga cara untuk menempatkan CSS di Kode HTML nih, guys!

1. **Inline**, yaitu ketika kamu menambahkan style dalam tag HTML (berupa atribut), sebaiknya dipakai saat benar-benar diperlukan aja guys!
2. **Embed**, yaitu ketika kamu menambahkan tag `<style>` di dalam tag `<head>` pada HTML.
3. **External**, dilakukan dengan menghubungkan CSS dari file lain.





Sekarang kita bahas satu per satu yuk~



## Penulisan Inline

Penulisan inline CSS digunakan pada tag HTML, guys! Caranya dengan **memasukkan kode CSS ke dalam atribut style**. Jadi, cara ini **hanya akan mempengaruhi satu elemen HTML** yang terkandung dalam atribut style.

Perhatikan contoh gambar di bawah. Pada **style "color:yellow;background-color:orange;" hanya akan mempengaruhi satu elemen, yaitu kalimat "Aku cinta Indonesia!"**, begitu juga dengan style "color:red;background-color:blue;" pada elemen/kalimat "Aku suka pulau lombok!".

- Masukan kode CSS di dalam atribut style
- Penulisan inline style harus menyatu, tanpa spasi.



```
1 <h1 style="color:yellow;background-color:orange;"> Aku cinta Indonesia! </h1>
2 <p style="color:red;background-color:blue;"> Aku suka pulau lombok! </p>
```



Nanti hasilnya kayak gini, guys!



## Meski cara penulisan inline ini kurang direkomendasikan, tapi di beberapa situasi ini bisa membantu, lho!

Cara penulisan inline memang akan nambahin kerjaan kamu, guys! Karena setiap tag HTML harus dikasih style masing-masing. Selain itu, kamu akan kesulitan dalam mengatur layout jika hanya menggunakan inline CSS.

Tapi.. di beberapa situasi justru inline CSS bisa jadi berguna. Contohnya, pada saat kamu nggak punya akses ke file CSS atau harus mengubah style untuk satu elemen saja.





## Penulisan Embed/Internal

Dengan cara penulisan embed/internal, kamu bisa tulis semua kode properti CSS **di dalam tag html khusus, yaitu tag <style>**.

Nah, kalau pakai metode penulisan ini, maka kode CSS yang berada di dalam tag html <style> harus berada dalam satu file dengan **tag-tag html lainnya**.

Ada beberapa kasus di mana penggunaan internal stylesheet justru berguna. Salah satu contohnya adalah untuk mengirimkan template halaman ke seseorang – karena semuanya bisa terlihat dalam satu halaman, maka akan lebih mudah untuk melihat previewnya

```
● ● ●
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My CSS experiment</title>
    <style>
      h1 {
        color: blue;
        background-color: yellow;
        border: 1px solid black;
      }

      p {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>This is my first CSS example</p>
  </body>
</html>
```

## Penulisan Eksternal

Eksternal CSS adalah kode CSS yang ditulis terpisah dengan kode HTML. Jadi, eksternal CSS ini **ditulis di sebuah file khusus** yang **berekstensi .css**.

File eksternal CSS biasanya diletakkan setelah bagian <head> pada halaman. Nah, untuk menghubungkan file HTML dengan file CSS kamu bisa menulis kayak gini ya:

```
<link rel="stylesheet"  
      ref="styleGua.css">
```

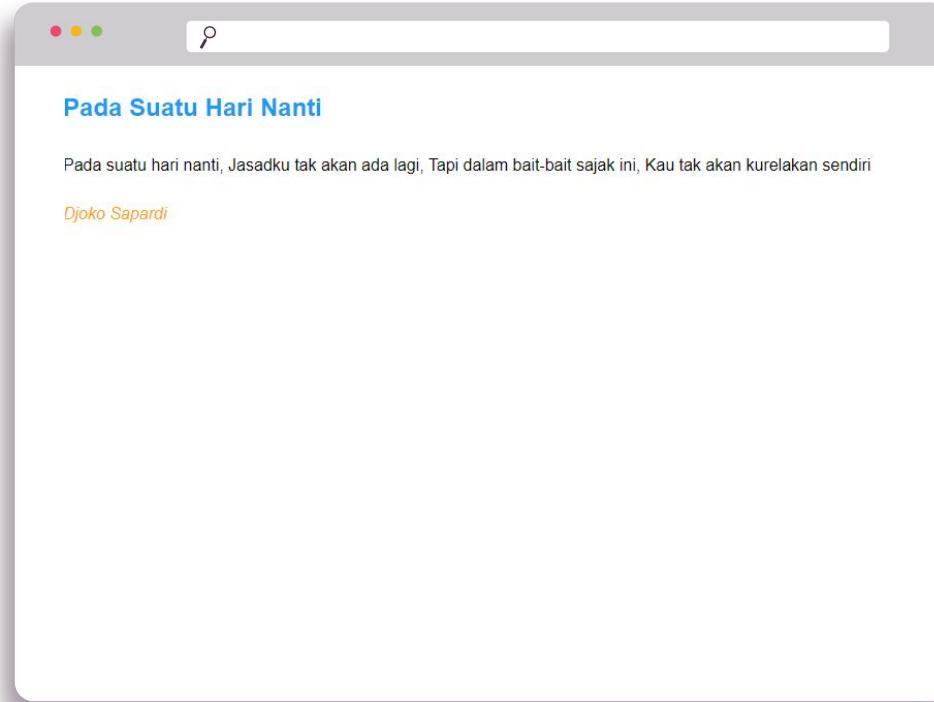
Cara ini lebih sederhana daripada menambahkan kode CSS di setiap elemen HTML yang ingin kamu atur tampilannya.

```
1 /* Isi file StyleGua.css */  
2  
3 p {  
4     font-family: arial;  
5     line-height: 2.75em;  
6 }  
7  
8 i {  
9     font-family: arial;  
10    color: orange;  
11 }  
12  
13 h2 {  
14     font-family: arial;  
15     color: #1B9CFC;  
16 }
```

```
1 <!-- isi file index.html -->  
2  
3 <html>  
4 <head>  
5     <title>Contoh Eksternal CSS</title>  
6     <link rel="stylesheet" type="text/css" href="StyleGua.css">  
7 </head>  
8 <body>  
9     <h2>Pada Suatu Hari Nanti</h2>  
10    <p>Pada suatu hari nanti, Jasadku tak akan ada lagi,  
11        Tapi dalam bait-bait sajak ini, Kau tak akan kurelakan sendiri.  
12    </p>  
13    <i>Djoko Sapardi </i>  
14 </body>  
15 </html>
```



## Nanti hasilnya kayak gini, guys!



**Sob, masih ingat kan kalau CSS ibarat desain interior sebuah rumah.**

**Lalu apa saja sih yang dipercantik sama CSS ini? Yuk langsung kepoin aja!**





## Pertama, CSS akan mempercantik text di HTML, istilahnya adalah Text Styling

Dalam Text Styling, yang bisa kamu lakukan menggunakan CSS adalah kayak gini, guys:

**Color** : digunakan untuk memberi warna pada tulisan.

**Text-align** : digunakan untuk mengatur format paragraf/teks (justify).

**Text-decoration** : digunakan untuk mengatur dekorasi garis pada teks (underline, coret dll).

**Text-indent** : digunakan untuk memberi indentasi pada awal paragraf/teks.



```
h1 {  
    color : blue;           /* nama warna, hexadecimal, rgb */  
    text-align : left;      /* left, right, center, justify */  
    text-decoration : underline; /* none, underline, overline, line-through */  
    text-indent : 30px;      /* px, % */  
}
```



## Text Styling-nya kita lanjut di sini ya guys!

**Text-transform** : digunakan untuk mengubah jenis huruf menjadi huruf besar, kecil/kapital.

**Letter-spacing** : digunakan untuk mengatur spasi/jarak antar huruf.

**Word-spacing** : digunakan untuk mengatur spasi/jarak antar kata.

```
● ● ●  
  
h1 {  
    text-transform: lowercase; /* none, lowercase, uppercase, capitalize */  
    letter-spacing: 10px;      /* normal, px */  
    word-spacing: 15px;       /* normal, px */  
}
```



## Kedua, CSS akan mempercantik latar belakang di HTML, istilahnya adalah “Background Styling”

Dalam Background Styling, yang bisa kita lakukan menggunakan CSS adalah kayak gini, guys!

**Background-color:** digunakan untuk mengatur warna pada background.

**Background-image :** digunakan untuk mengatur gambar yang akan digunakan pada background.

**Background-position:** digunakan untuk mengatur posisi gambar pada background.

**Background-repeat:** digunakan untuk mengatur jenis pengulangan gambar pada background.

```
● ● ●
body {
    background-color: red;           /* nama warna, hexadecimal, rgb */
    background-image: url(photo.png); /* path ke resource */
    background-position: center top; /* x% y%, x-position y-position */
    background-repeat: repeat;       /* x = left/center/right, y = top/center/bottom */
}                                     /* repeat, repeat-x, repeat-y, no-repeat */
```



## Ada Tag di HTML yang nggak punya arti apapun, terus buat apa ya?

Untuk jawab pertanyaan di atas, kita kenalan dulu deh sama Tag Div dan Span.

Tag pada HTML digunakan untuk memberikan ‘maksud’/‘arti’ pada sebuah konten. (Contoh: p untuk paragraph, h1 untuk heading, dan lain-lain), tapi **tag <div> dan <span> nggak memiliki arti apapun.**

Kenapa? Karena keduanya dipakai untuk mengelompokkan tag-tag HTML dan memberikan informasi terhadap tag-tag tersebut.

...

```
<div></div>
```

```
<span></span>
```

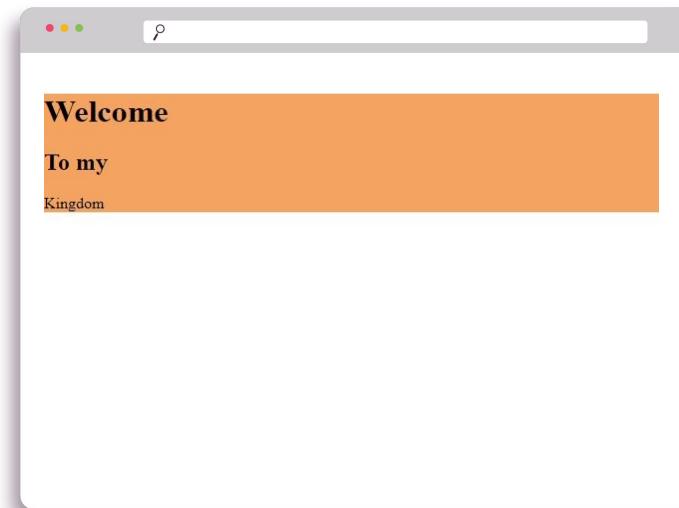


## Bisa dibilang Tag Div dan Span adalah induk dari elemen lain~

Jadi, tag div dan span bisa menampung tag-tag html lain, mereka berperan sebagai induk dari elemen-elemen lain. Efeknya adalah apabila tag div & span dikasih styling seperti background-color, maka semua tag yang ada di dalamnya akan mengikuti. Yuk lihat contohnya pada gambar di bawah ini!

```
1 <div>
2   <h1>Welcome</h1>
3   <h2>To my</h2>
4   <p>Kingdom</p>
5 </div>
```

```
1  div {
2    background-color: orange;
3 }
```





Selain struktur CSS, ada juga box model yang penting dalam layouting.

**Box Model**, istilah untuk apa ini ya? Mari kita bahas dan cari tahu lebih lanjut tentang box model.





### Untuk jelasin kenapa perlu Box Model, kita pakai analogi aja ya biar lebih gampang dimengerti~

Dalam proses pembangunan rumah, tentu kita nggak bisa sembarangan kan? Perlu persiapan dan layout desain rumah yang matang ya, guys!

Kalau rumah kita udah punya layout desain yang matang, nantinya bisa dibangun dengan kokoh, nyaman ditinggali, dan sesuai fungsinya.

Selain itu, kita juga udah tau nih dimana letak kamar, ruang tamu, dapur dan toilet. Betul nggak?





Nah sama saja nih kaya pembuatan website. Ibarat rumah, **pembuatan website juga perlu diawali dengan perencanaan layout** dan perlu ada aturan ukuran, jarak, hingga warna supaya website jadi ‘rumah’ yang nyaman dilihat dan user friendly (gampang diakses).





# Sebelum bahas apa itu CSS Box Model, kamu harus paham beberapa konsep dulu ya~

Ada beberapa konsep yang harus kamu pahami tentang Box Model nih, yaitu:

- Setiap elemen di halaman web ada di dalam sebuah box (kotak).
- Kamu bisa mengatur ukuran dan posisi kotak tersebut.
- Kamu bisa ngasih warna/gambar sebagai background pada kotak tersebut.



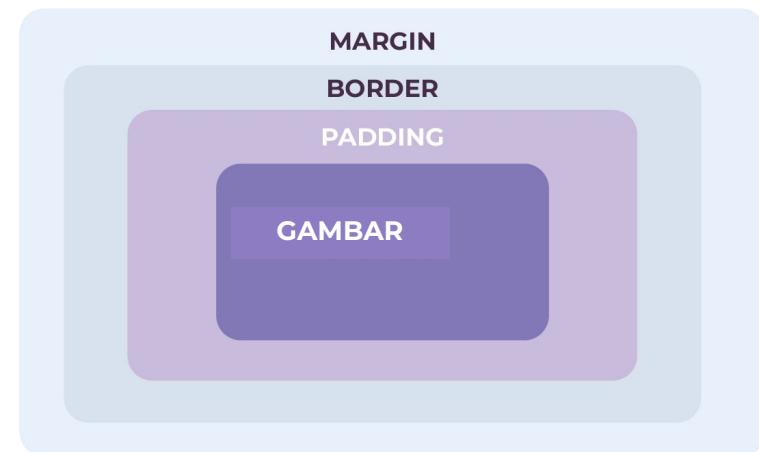


# kalau udah paham dengan konsep box model, kita lanjut ke pengertian dari CSS Box Model!

CSS Box Model **mendefinisikan ‘kotak’ yang dihasilkan oleh sebuah elemen**, lalu menampilkannya sesuai dengan format visualnya.

Misalnya di web browser ada sebuah elemen berupa gambar, nah CSS Box Model melihat elemen gambar ini sebagai sebuah kotak ‘box’.

Jadi, mulai dari paragraf, header, form, gambar, logo hingga video, sebenarnya ditampilkan oleh web browser sebagai ‘box’.





# CSS Box Model juga ada bagian-bagian di dalamnya~

CSS Box Model terdiri dari 4 bagian guys!

- **Margin**, yaitu area transparan yang ada di sekitar kotak (di luar border).
- **Border** : Batas di sekeliling content & padding.
- **Padding**, yaitu area transparan yang ada di dalam kotak (antara konten & border).
- **Content** : Konten sebenarnya ada di dalam box, bisa berupa teks atau gambar.

Yuk kita pelajari detilnya lebih lanjut di slide berikutnya! 😊





# Margin

Margin berfungsi untuk **mengatur jarak elemen HTML dari luar**.

Lihat bentuknya pada gambar di bawah:

Here's a paragraph

Margin

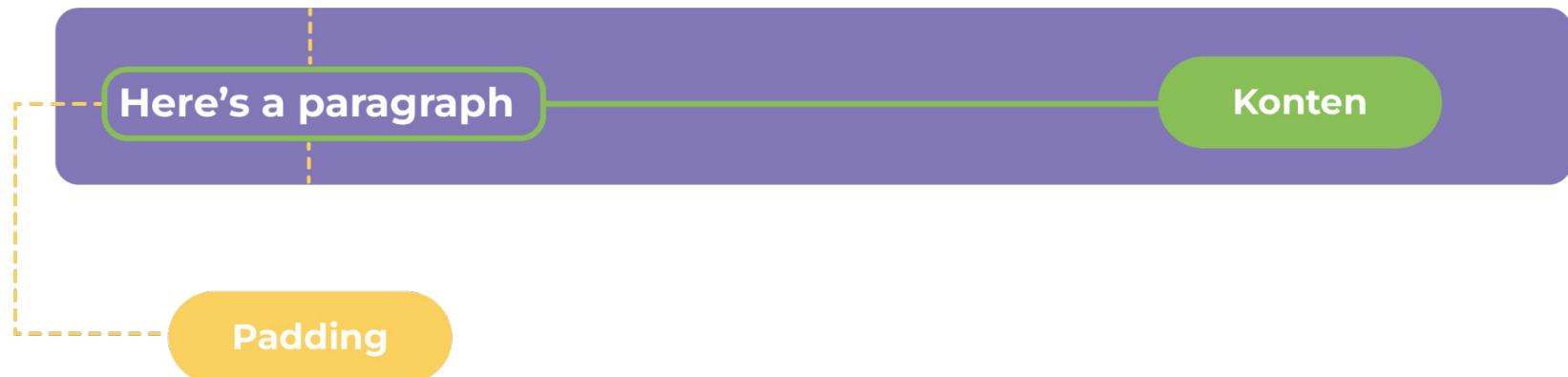
Here's another paragraph



# Padding

Padding adalah **jarak/space dari copy konten ke Box Element**.

Lihat bentuknya pada gambar di bawah:





## Terus, apa bedanya antara Padding dan Margin?

### Yuk lihat visualisasinya di bawah ini~





## Box Model punya alat untuk bantu tata letak sebuah elemen, namanya Box Model Properties!

Nah, di bawah ini adalah beberapa **CSS property yang dipakai untuk bikin margin, padding, dan border:**

Margin-top	padding-top	border-top
Margin-right	padding-right	border-right
Margin-bottom	padding-bottom	border-bottom
Margin-left	padding-left	border-left
Margin	padding	border



Kalau box model mengatur margin, padding, dan border, ada juga yang namanya **Flexbox**, yang nggak kalah penting dalam layouting sebuah aplikasi.

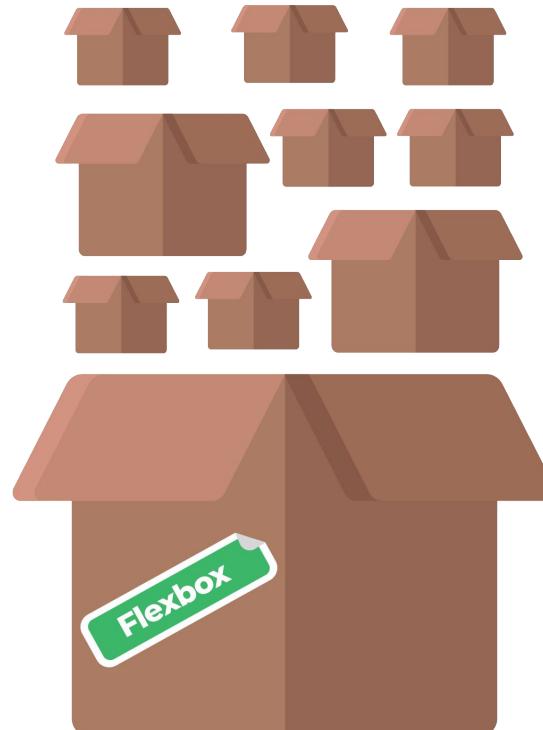
**Apa sih Flexbox itu?**





## Biar gampang, kita analogikan konsep Flexbox ini seperti sebuah kardus

Katakanlah ada sebuah kardus besar yang berisi kardus-kardus kecil di dalamnya. Kamu bisa menyesuaikan tata letak (layout) kardus-kardus kecil tersebut di dalam kardus besar tadi. Nah, **Flexbox ini sama seperti konsep/cara kamu menyesuaikan tata letak kardus-kardus kecil** di dalam kardus besar tersebut, guys!

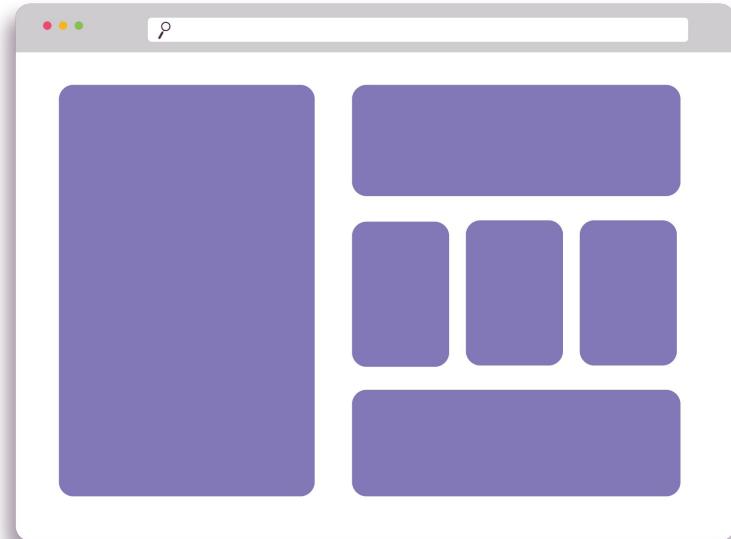




**Jadi, elemen child ibarat kardus kecil dan container ibarat kardus besarnya, guys!**

Flexbox adalah **konsep pengaturan layout** yang mencakup ukuran elemen Child dari suatu Container, untuk beradaptasi dengan Parent/Container-nya.

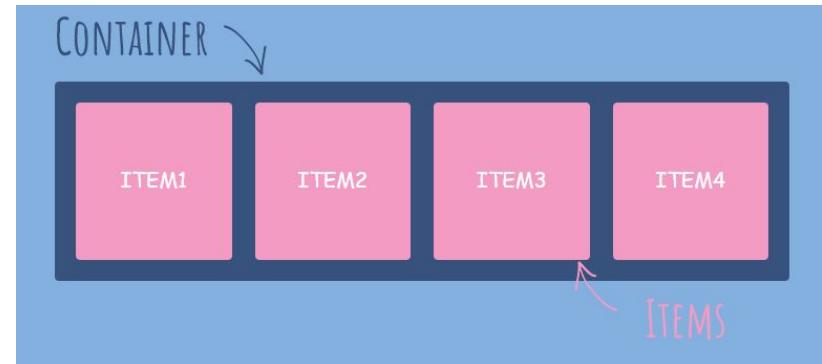
Flexbox umumnya dipakai pada sebuah elemen yang nggak pasti ukurannya atau berubah-ubah (dinamis).





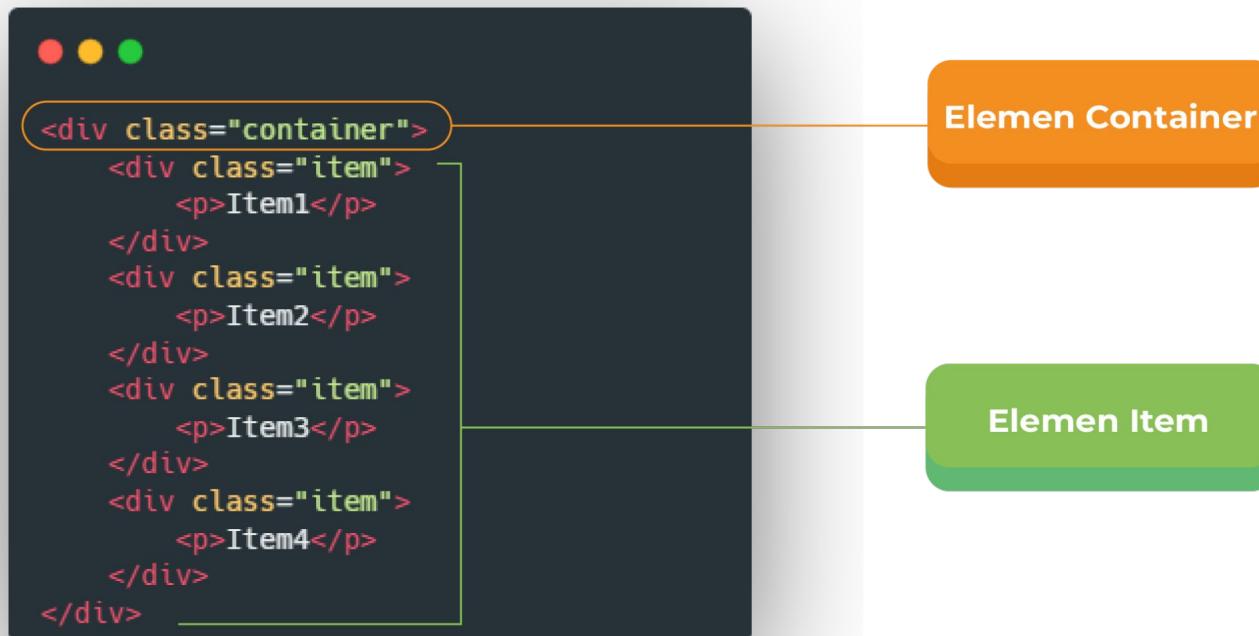
**Yash! Ada dua elemen utama dalam sebuah layout FlexBox yaitu Container dan Item.**

**Element Container** adalah elemen induk yang akan membungkus elemen-elemen lain di dalamnya, sedangkan **element children (item)** adalah elemen-elemen yang ada di dalam elemen induk.





## Ini gambaran Container dan Item

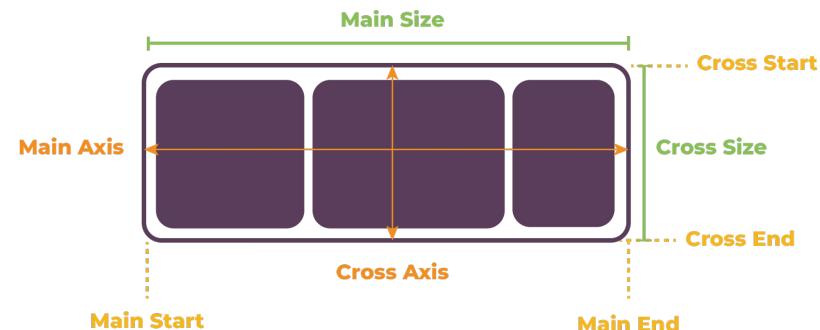


## Selain elemen, ini istilah penting dalam Flexbox yang juga perlu kamu tahu~

**Main Axis:** Sumbu utama dari sebuah container yang menentukan urutan dari penempatan items secara horizontal.

**Main start | Main end:** Mulai dan berakhirnya items yang disimpan dalam container.

**Main size:** Ukuran (width/height) dari container yang akan bikin dimensi dari item-nya relatif terhadap main size. Misalnya main size-nya adalah 300 pixel, sementara kamu mau masukin 3 item, berarti besar setiap item adalah 300 dibagi 3. Yapss, jadi masing-masing item besarnya 100 pixel!





### Properti CSS Element Container

Nah, kamu sudah tahu kan apa itu flexbox? Dalam flexbox, ada yang disebut dengan Element Container. Berikut beberapa property CSS yang dipakai pada Element Container saat kamu mau bikin layout pake Flexbox:

Display

Flex-Direction

Flex-Wrap

Justify-Content

Align-Items

Align-Content



### 1. Property Display

Property ini digunakan untuk **bikin sebuah elemen parent jadi flexbox** dan bikin elemen-elemen di dalamnya bisa berperilaku flex juga. Contoh penulisannya bisa kamu lihat pada gambar, ya!



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



## 2. Property Flex-Direction

Property ini digunakan untuk **mengatur arah/urutan dari items** di dalam container. Flex-direction punya beberapa jenis, yaitu: row, row-reverse, column dan column-reverse.

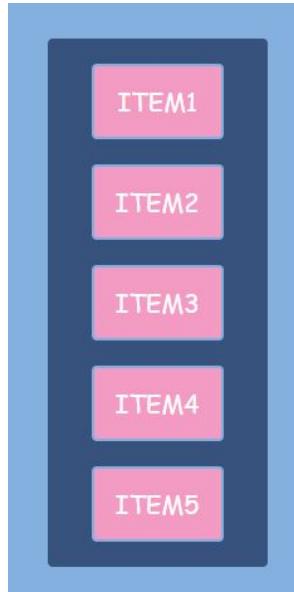
Contoh penulisannya bisa kamu lihat pada gambar, ya!

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

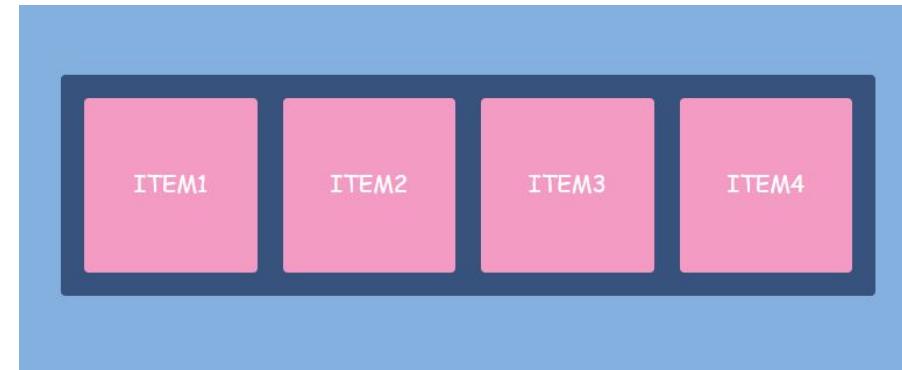


Sekarang kita lihat contoh hasilnya yuk~

Flex-Direction: column



Flex-Direction: Row





### 3. Property Flex-Wrap

Secara default, semua items di dalam container akan ada pada satu baris meskipun ukurannya sudah nggak cukup. **Property Flex-Wrap bisa dipakai untuk mindahin items ke baris di bawahnya.** Ada beberapa jenis Flex-Wrap , yaitu: nowrap, wrap dan wrap-reverse.

Contoh penulisannya bisa kamu lihat pada gambar, ya!

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



### Sekarang kita lihat contoh hasilnya yuk~

Jadi, item yang nggak cukup akan pindah ke baris bawah menggunakan Flex-Wrap, guys!





### 4. Property Justify-Content

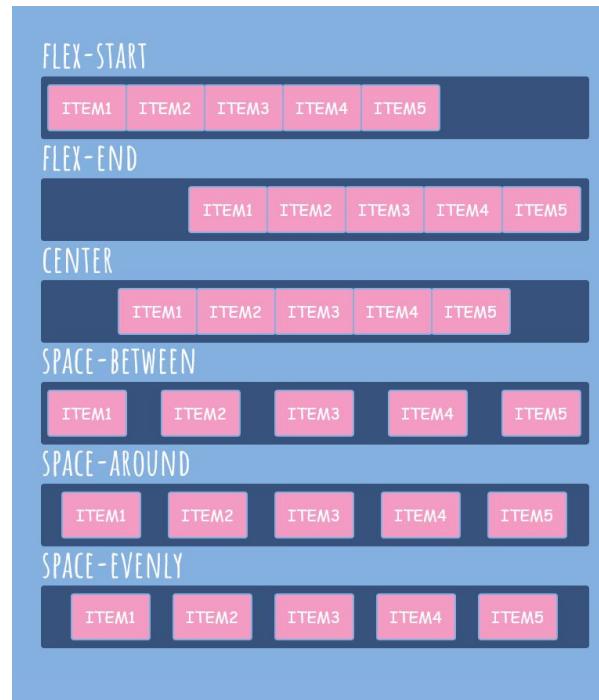
Property ini digunakan untuk **membuat sejajar setiap items** yang ada pada flex container. Ada beberapa jenis Justify-Content, yaitu: flex-start, flex-end, center space-between, space-around dan space evenly.

Contoh penulisannya bisa kamu lihat pada gambar, ya!

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



Sekarang kita lihat contoh hasilnya yuk~





## 5. Property Align-Items

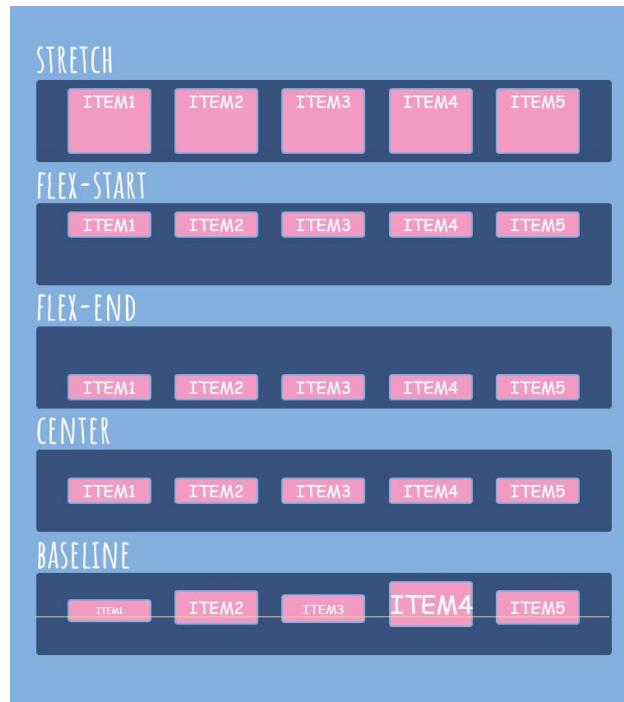
Property ini digunakan untuk **mengatur perilaku penajaran items terhadap cross axis**. Ada beberapa jenis Align-Items, yaitu: flex-start, flex-end, center stretch dan baseline.

Contoh penulisannya bisa kamu lihat pada gambar, ya!

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



Sekarang kita lihat contoh hasilnya yuk~





## 6. Property Align-Content

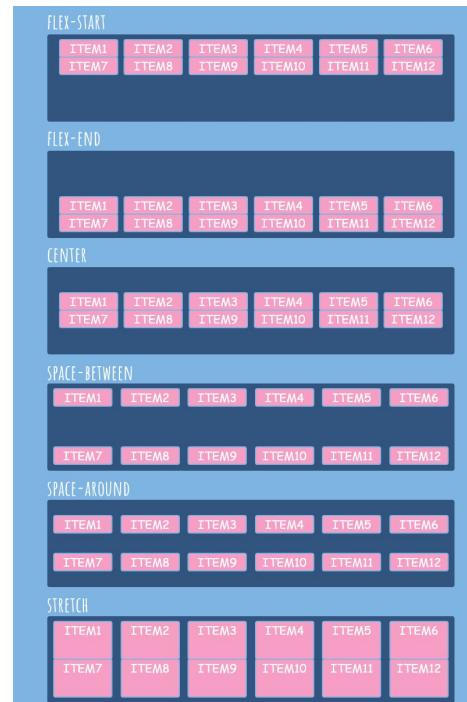
Property ini digunakan untuk **mengatur jarak antar items terhadap cross axis** (hanya berfungsi kalau items lebih dari 1 baris). Ada beberapa jenis Align-Content, yaitu: flex-start, flex-end, center, space-between, space-around.

Contoh penulisannya bisa kamu lihat pada gambar, ya!

```
.container {  
    align-content: flex-start | flex-end | center |  
                 space-between | space-around;  
}
```



Sekarang kita lihat contoh hasilnya yuk~





### Properti CSS Element Items

Nah kalau yang di bawah ini adalah beberapa property CSS yang dipakai pada elemen items dalam layouting dengan flexbox, yaitu:

**Order**

**Flex-Grow**

**Align-Self**



## 1. Properti Order

Properti ini berfungsi untuk **mengatur urutan** setiap items.

Contoh penulisannya bisa kamu lihat pada gambar, ya!

```
<div class="container">
  <div class="satu">1</div>
  <div class="dua">2</div>
  <div class="tiga">3</div>
  <div class="empat">4</div>
</div>
```

```
.satu { order: 2; }
.dua { order: 3; }
.tiga { order: 4; }
.empat { order: 1; }
```



Sekarang kita lihat contoh hasilnya yuk~

No.	Coding	Tampilan Hasil Coding
1.	<pre>...  &lt;div class="container"&gt;   &lt;div class="satu"&gt;1&lt;/div&gt;   &lt;div class="dua"&gt;2&lt;/div&gt;   &lt;div class="tiga"&gt;3&lt;/div&gt;   &lt;div class="empat"&gt;4&lt;/div&gt; &lt;/div&gt;</pre>	
2.	<pre>...  .satu { order: 2; } .dua { order: 3; } .tiga { order: 4; } .empat { order: 1; }</pre>	



### 2. Properti Flex-Grow

Ini adalah property yang diterapkan pada flex item, berfungsi untuk **mengatur ukuran berdasarkan seberapa besar space kosong** yang akan diambil dari flex-container

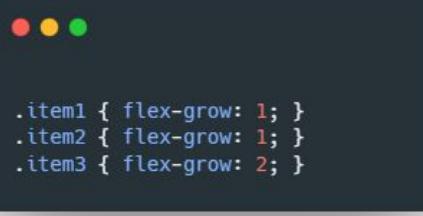
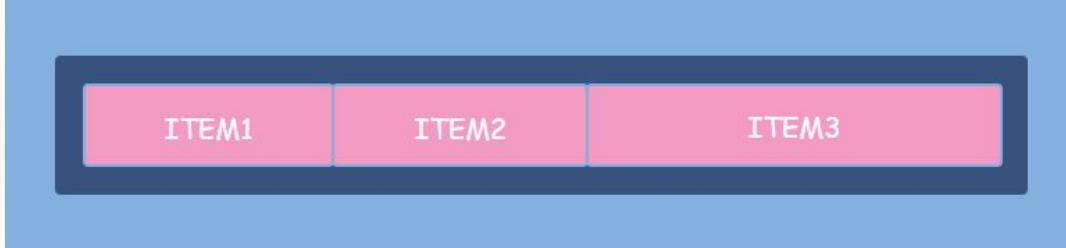
Contoh penulisannya bisa kamu lihat pada gambar, ya!



```
.item1 { flex-grow: 1; }  
.item2 { flex-grow: 1; }  
.item3 { flex-grow: 2; }
```



Sekarang kita lihat contoh hasilnya yuk~

Coding	Tampilan Hasil Coding
 <pre>.item1 { flex-grow: 1; } .item2 { flex-grow: 1; } .item3 { flex-grow: 2; }</pre>	



### 3. Properti Align-Self

Ini adalah property yang diterapkan pada flex item, berfungsi **mengatur tata letak penajaran** dari setiap elemen item.

Contoh penulisannya bisa kamu lihat pada gambar, ya!

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

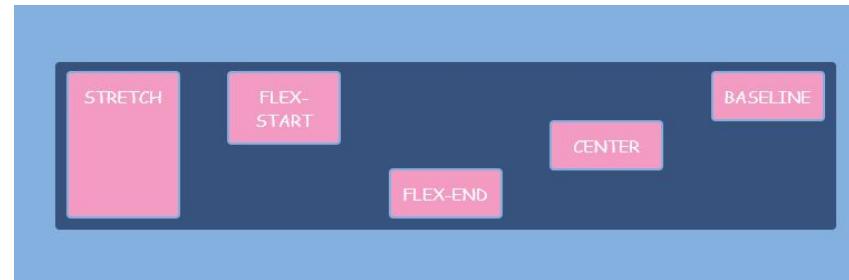


Sekarang kita lihat contoh hasilnya yuk~

### Coding

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

### Tampilan Hasil Coding

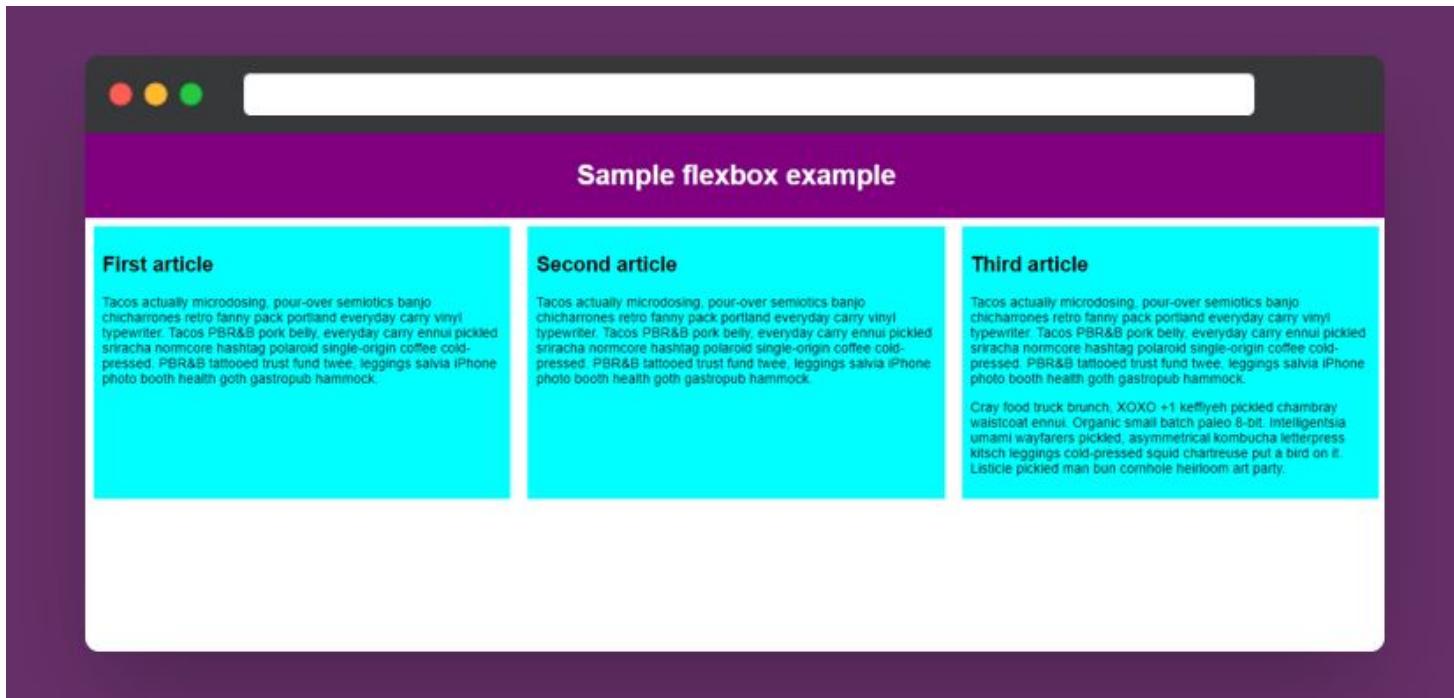


**Sekarang waktunya kamu latihan bikin  
flexbox!**

Nggak susah kok latihannya...



Buatlah sebuah layout flexbox memakai HTML dan CSS. Layout yang kamu bikin harus sama persis dengan tampilan di bawah ini ya.



Saatnya kita  
**Latihan!**



Biar kamu makin paham dengan materi di sesi-sesi sebelumnya dan biar kamu makin semangat untuk mengeksplor lebih banyak hal terkait materi ini, kita latihan HTML dan CSS yuk, guys!

#### A. Instruksi

- Buat sebuah folder khusus dengan nama **html-css-layout**
- Kemudian tulis semua kode HTML, CSS, beserta file-file gambar yang kamu perlukan di dalam folder tersebut
- Initialisasi git pada folder **html-css-layout** tersebut, kemudian push semua file jawaban yang telah kamu buat di dalam folder tersebut ke remote-repository (bebas, silahkan pilih mau pakai Github, Gitlab, atau BitBucket)
- Facilitator akan menunjuk kamu untuk melakukan presentasi/demo singkat terkait project yang kamu buat kali ini.

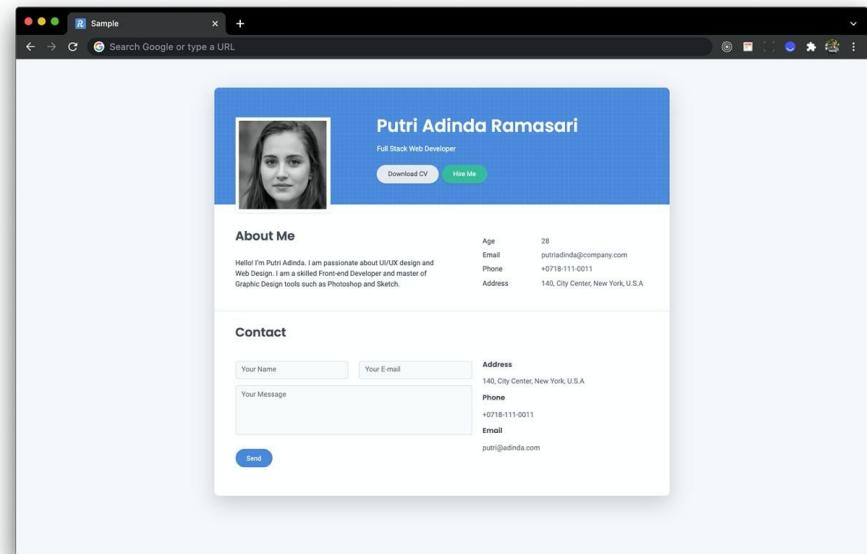
## B. Tugas

Buatlah tampilan sebuah website dengan menggunakan HTML, CSS, dan Flexbox.

Tampilan website yang kamu buat **harus memiliki tema atau 70% sama** dengan tampilan website pada gambar disamping ini.

**Sesuaikan nama, informasi, dan konten pada website tersebut dengan biodata kamu.**

Kamu sangat dianjurkan untuk menambah kreasi sendiri pada project website kamu ini, seperti mengganti warna background dengan warna pilihan kamu, menambahkan informasi hobi kamu, atau mungkin menambahkan foto kucing kesayangan kamu.



## Explore more!

Namanya latihan perlu alat-alat pendukung ya! ini ada beberapa resource materi tambahan yang bisa bantu kamu mengerjakan project ini, guys! ini juga bisa bantu meningkatkan pemahaman kamu lho!

1. <https://www.youtube.com/watch?v=-J372iDFU8>
2. <https://www.youtube.com/watch?v=u7w0B2f91rk>

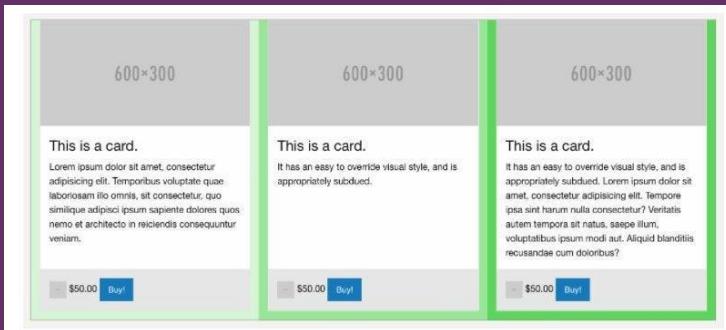
kalau kamu mengalami kesulitan, atau ada konsep/teori yang kamu masih belum pahami dengan baik, silahkan tanyakan ke Facilitator. Facilitator akan dengan senang hati bantu kamu agar bisa dapat pengetahuan maksimal, guys!

# Saatnya kita Quiz!



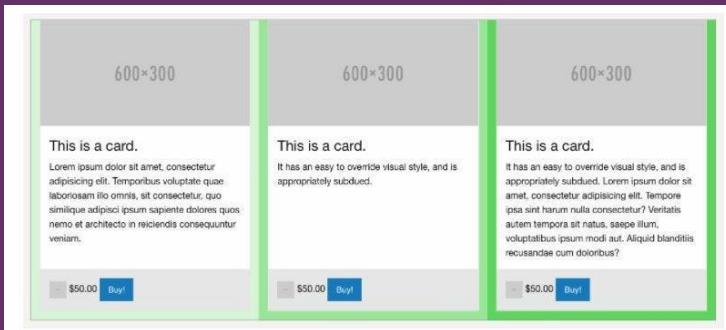


# 1. CSS property apa yang digunakan untuk membuat cards pada tampilan menjadi sebuah row?



- A. flex-direction: 'row'
- B. position: 'row'
- C. margin: 'row'

# 1. CSS property apa yang digunakan untuk membuat cards pada tampilan menjadi sebuah row?



- A. flex-direction: 'row'
- B. position: 'row'
- C. margin: 'row'

Menggunakan flex-direction untuk mengatur arah dari sebuah layout



## 2. Manakah cara yang benar untuk menambahkan background image pada sebuah website?

- A. Menggunakan <img> tag
- B. Menggunakan CSS dengan properti background-image
- C. Menggunakan tag <link> dan tag <img>



## 2. Manakah cara yang benar untuk menambahkan background image pada sebuah website?

- A. Menggunakan <img> tag
- B. Menggunakan CSS dengan properti background-image
- C. Menggunakan tag <link> dan tag <img>

background-image adalah sebuah properti yang berfungsi membuat background sebuah element menjadi sebuah gambar



### 3. Tag yang berfungsi membuat bold pada sebuah text?

- A. <pre>
- B. <b>
- C. <br>



### 3. Tag yang berfungsi membuat bold pada sebuah text?

- A. <pre>
- B. <b>
- C. <br>

<b> adalah properti untuk membuat text menjadi bold, <br> untuk membuat baris baru, <pre> untuk membuat blok teks preformat



**4. Dibawah ini yang mana yang merupakan attribute pada sebuah tag img html?**

- A. source
- B. src
- C. Tidak keduanya



#### 4. Dibawah ini yang mana yang merupakan attribute pada sebuah tag img html?

- A. source
- B. src
- C. Tidak keduanya

src adalah attribut yang terdapat pada tag html untuk menampilkan image, tag img



## 5. Yang bukan merupakan property CSS Flexbox buat Element Container adalah...

- A. align-shelf
- B. align-content
- C. align-items



## 5. Yang bukan merupakan property CSS Flexbox buat Element Container adalah...

- A. align-shelf
- B. align-content
- C. align-items

align-shelf adalah properti yang digunakan pada element items pada sebuah flexbox layout.

# Terima Kasih!



Next Topic

loading...