



# Pemrograman Berbasis Web

Pertemuan 4

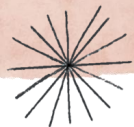
**Nori Wilantika**  
**Lya Hulliyyatus Suadaa**  
**Yeni Rimawati**

Politeknik Statistika STIS  
Prodi DIV Komputasi Statistik

The background is a light cream color with various abstract, hand-drawn style elements. In the top left, there is a horizontal red brushstroke and a cluster of small black dots. On the left side, there is a grey circular watercolor shape with a black starburst line drawing above it. In the bottom left, there is a pinkish-red brushstroke and a small cluster of black dots. On the right side, there is a large yellow circular watercolor shape, a thin black wavy line with two small yellow circular accents, and a grey brushstroke at the bottom right.

# 04

## CSS (2)



GmailGambar

Login

Google

Penelusuran GoogleSaya Lagi Beruntung

Google menawarkan: EnglishBasa Bali

Indonesia

TentangPeriklananBisnisCara kerja Penelusuran

PrivasiPersyaratanSetelan

InspectorConsoleDebuggerNetworkStyle EditorPerformanceMemoryStorageAccessibilityApplication

Search HTML

<!DOCTYPE html>

<html itemscope="" itemtype="http://schema.org/WebPage" lang="id">

overFlow

<link id="dark-mode-custom-link" type="text/css" rel="stylesheet">

<link id="dark-mode-general-link" type="text/css" rel="stylesheet">

<style id="dark-mode-custom-style" type="text/css" lang="en"></style>

<style id="dark-mode-native-style" type="text/css" lang="en"></style>

</head></head>

<body jsmodel="TVhxbe" jsaction="IVKtfe:.CLIENT;hICeId:.CLIENT;KsN8n:sbTXnb:.CLIENT;X;.PY1zjf:.CLIENT;wnJTPd:.CLIENT;JL9Qdc:.CLIENT;koLxhv

Filter Styles

element

inline

\* {

border: 1px solid red;

}

html {

body, html {

height: 100%;

margin: 0;

padding: 0;

}

}

LayoutComputedChangesFontsAnimations

Flexbox

Select a Flex container or item to continue.

Grid

Box Model

margin0

borderpadding

1347x484

html > body > div.l3eUgb > div.o3j99.qarstb



**BADAN PUSAT STATISTIK**

Indonesia | [English](#)

[Cari](#)

[Manual](#) | [Tautan](#) | [Peta Situs](#) | [S&K](#)

[Beranda](#)

[Tentang Kami](#)

[Berita](#)

[Senarai Rencana Terbit](#)

[Publikasi](#)

[Berita Resmi Statistik](#)

[Layanan](#)

[PPID](#)

**Ekonomi dan Perdagangan**

[Ekspor-Impor](#)

[Energi](#)

[Harga Eceran](#)

[Selengkapnya...](#)

**Pertanian dan Pertambangan**

[Hortikultura](#)

[Kehutanan](#)

[Perikanan](#)

[Selengkapnya...](#)



**Galeri Infografis**

Cara Mendapatkan Data BPS  
PNBP • Nol Rupiah • Gratis

## Berita



### BPS: 270,20 juta Penduduk Indonesia Hasil SP2020

Jakarta – Sebagian tahapan Sensus Penduduk 2020 (SP2020) telah selesai dilaksanakan di tahun 2020. SP2020 merupakan Sensus Penduduk pertama yang menggunakan data registrasi kependudukan sebagai basis data. Hal ini dalam rangka mewujudkan Satu Data Kependudukan Indonesia. Hari ini (21/1) berte...

## Kegiatan Statistik

21 Januari 2021 [BPS: 270,20 juta Penduduk Indonesia Hasil SP2020](#)

## Kegiatan Statistik Lainnya

12 Februari 2021 [Desa Cantik, Quick Wins BPS 2021](#)

21 Desember 2020 [Pendaftaran Seleksi JPT Madya dan JPT Pratama BADAN PUSAT STATISTIK](#)

17 Desember 2020 [Kecuk Menyapa: Sinergitas Satu Data Indonesia](#)

14 Desember 2020 [Merajut Data E-Commerce Indonesia: Kolaborasi Public Private Sectors di Era Digital](#)

Survei

[COVID-19](#) BPS

Publikasi

[COVID-19](#) BPS



**DATA PERENCANAAN  
PEMBANGUNAN**

**Statistical  
Geographic  
Information System**



**Satu Data  
Indonesia**



**PELAYANAN STATISTIK TERPADU**  
[pst.bps.go.id](http://pst.bps.go.id)

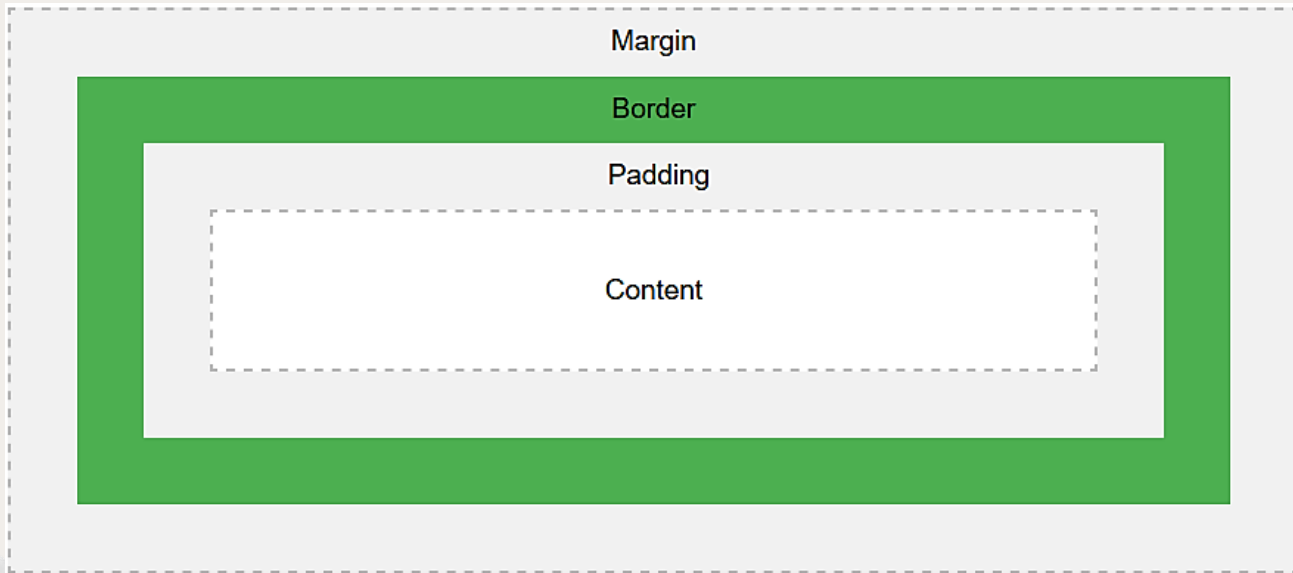
Aplikasi Allstats BPS





# CSS BOX MODEL

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





# CSS BOX MODEL

## Margin

Digunakan untuk membuat ruang di sekitar elemen.

## Padding

Digunakan untuk menghasilkan ruang di sekitar konten elemen, di dalam border yang ditentukan.

## Border

Digunakan untuk menentukan gaya, lebar, dan warna batas elemen.

I have a red bottom border.

I have rounded borders.

I have a blue left border.



# HEIGHT AND WIDTH

Properti height dan width digunakan untuk mengatur tinggi dan lebar elemen.

Properti **height** dan **width** memiliki opsi nilai berikut:

- **auto** - Default. Browser menghitung sendiri ukuran tinggi dan lebar
- **length** - Mendefinisikan tinggi / lebar dalam px, cm dll.
- **%** - Mendefinisikan lebar dalam persen dari blok container
- **initial** - Mengatur tinggi / lebar ke nilai defaultnya
- **inherit** - Tinggi / lebar akan diwarisi dari nilai induknya

Tinggi dan lebar elemen hanya mencakup area content, belum termasuk padding, borders, dan margins.

```
div {  
  height: 100px;  
  width: 500px;  
  background-color: powderblue;  
}
```



# OVERFLOW

Properti **overflow** mengontrol apa yang terjadi pada konten yang terlalu besar untuk dimasukkan ke dalam suatu area.

Lorem ipsum dolor sit amet  
consectetur adipisicing elit.  
Corporis vitae et quibusdam  
officiis, iste similique iure  
consequuntur dignissimos  
amet eos reprehenderit debitis,  
natus illo, saepe sunt incidunt  
exercitationem repellat ipsum.

**visible (default)**

Lorem ipsum dolor sit amet  
consectetur adipisicing elit.  
Corporis vitae et quibusdam

**hidden**

Lorem ipsum dolor sit amet  
consectetur adipisicing elit.  
Corporis vitae et quibusdam

**auto**

Lorem ipsum dolor sit amet  
consectetur adipisicing elit.  
Corporis vitae et quibusdam  
officiis, iste similique iure  
consequuntur dignissimos

**scroll**





# DISPLAY

- Properti **display** menspesifikasikan bagaimana elemen ditampilkan.
- Setiap elemen HTML memiliki nilai **display** default tergantung pada jenis elemennya. Nilai tampilan default untuk sebagian besar elemen adalah **block** atau **inline**.
- Elemen **block** selalu dimulai pada baris baru dan menempati lebar penuh yang tersedia (membentang ke kiri dan kanan sejauh mungkin). Contoh: **<div>**, **<h1>** - **<h6>**, **<p>**, **<form>**, **<header>**, **<footer>**, **<section>**, **<li>**
- Elemen **inline** tidak dimulai pada baris baru dan hanya membutuhkan lebar yang diperlukan, Contoh: **<span>**, **<a>**, **<img>**
- Nilai **display** default bisa diganti: **block**, **inline**, **none**, **hidden**, **inline-block**

```
<body>
<p>Menu:</p>
```

```
<ul>
  <li><a href="home.html" target="_blank">Home</a></li>
  <li><a href="table.html" target="_blank">Anggota Aktif</a></li>
  <li><a href="form.html" target="_blank">Daftar Sekarang!</a></li>
</ul>
</body>
```

Menu:

- [Home](#)
- [Anggota Aktif](#)
- [Daftar Sekarang!](#)

```
<style>
  li {
    display: inline;
    margin-right: 10px;
  }
</style>
```

Menu:

[Home](#) [Anggota Aktif](#) [Daftar Sekarang!](#)

```
<style>
  li {
    display: inline-block;
    margin-right: 10px;
    width: 150px;
    height: 50px;
    border: 1px solid blue;
  }
</style>
```

Menu:

<a href="#">Home</a>	<a href="#">Anggota Aktif</a>	<a href="#">Daftar Sekarang!</a>



# FLOAT

Properti **float** menentukan bagaimana elemen harus mengapung.

Properti **float** dapat memiliki salah satu dari nilai berikut:

- **left** - Elemen mengapung di sebelah kiri wadahnya
- **right** - Elemen mengapung di sebelah kanan wadahnya
- **none** – Default. Elemen tidak melayang (akan ditampilkan tepat di tempat elemen tersebut muncul).
- **inherit** - Elemen mewarisi nilai float dari induknya

Umumnya digunakan untuk membungkus gambar dengan teks di sekitarnya dan untuk mengatur elemen `<div>` agar posisinya di samping satu sama lain.





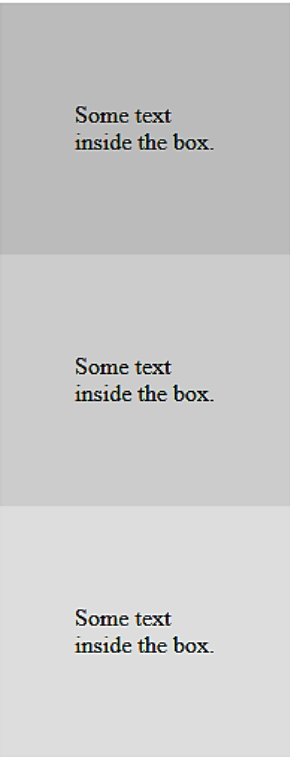
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac.





Some text  
inside the box.

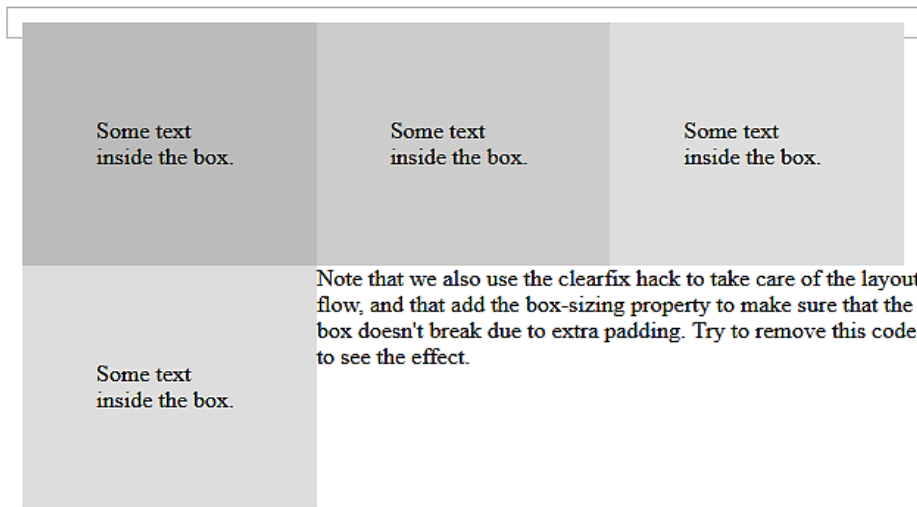
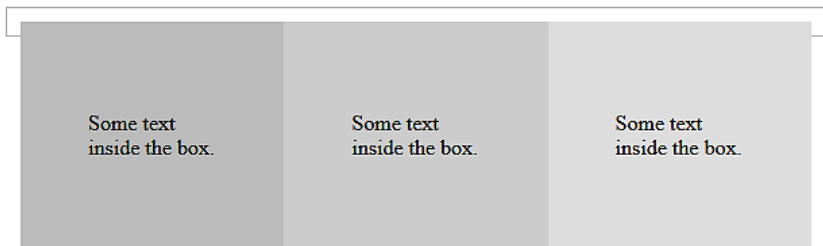
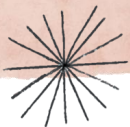
Some text  
inside the box.

Some text  
inside the box.

```
.container{  
  border: 1px solid #bbb;  
  padding: 10px;  
}
```

```
.box {  
  width: 33.33%;  
  padding: 50px;  
  box-sizing: border-box;  
}
```

```
<div class="container">  
  <div class="box" style="background-color:#bbb">  
    <p>Some text inside the box.</p>  
  </div>  
  <div class="box" style="background-color:#ccc">  
    <p>Some text inside the box.</p>  
  </div>  
  <div class="box" style="background-color:#ddd">  
    <p>Some text inside the box.</p>  
  </div>  
</div>
```





# CLEAR

- Cara paling umum untuk menggunakan **clear** properti adalah setelah Anda menggunakan **float** properti pada sebuah elemen.
- Properti **clear** dapat memiliki salah satu dari nilai berikut: **none, left, right, both, inherit**.

## Without clear

div1

div2 - Notice that div2 is after div1 in the HTML code. However, since div1 floats to the left, the text in div2 flows around div1.

## With clear

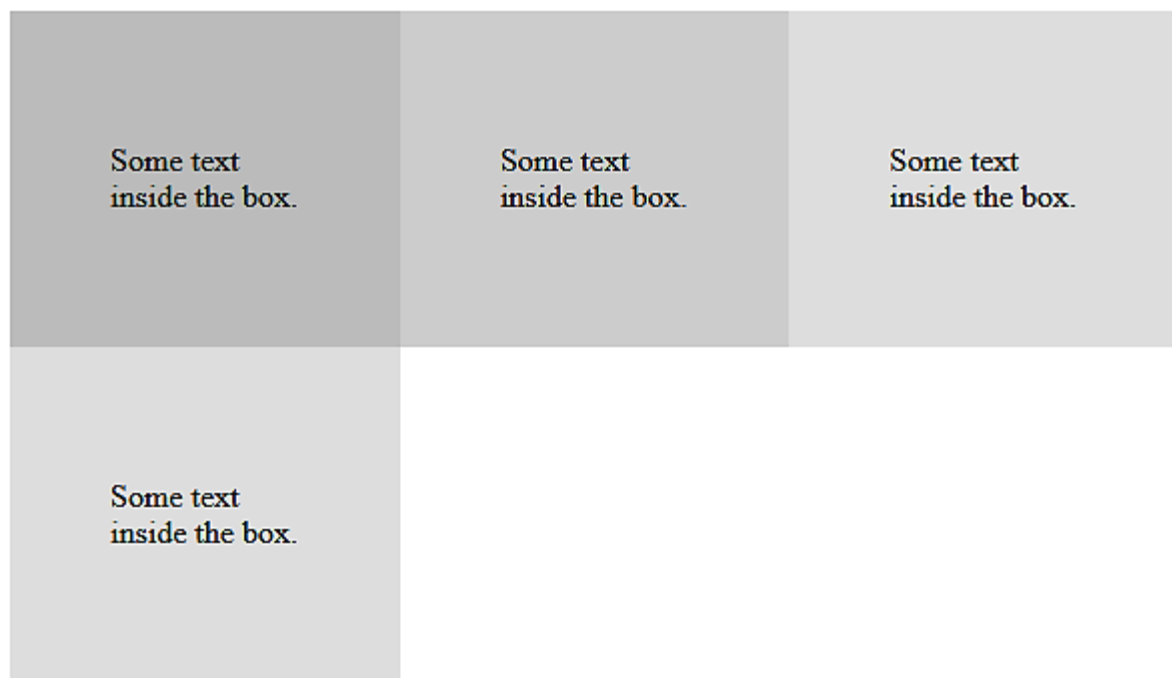
div3

div4 - Here, clear: left; moves div4 down below the floating div3. The value "left" clears elements floated to the left. You can also clear "right" and "both".

```
.div1 {  
  float: left;  
  width: 100px;  
  height: 50px;  
  margin: 10px;  
  border: 3px solid #73AD21;  
}  
  
.div2 {  
  border: 1px solid red;  
}  
  
.div3 {  
  float: left;  
  width: 100px;  
  height: 50px;  
  margin: 10px;  
  border: 3px solid #73AD21;  
}  
  
.div4 {  
  border: 1px solid red;  
  clear: left;  
}
```



```
.container {  
  overflow: auto;  
}  
  
.container::after {  
  content: "";  
  clear: both;  
  display: table;  
}
```



Note that we also use the clearfix hack to take care of the layout flow, and that add the box-sizing property to make sure that the box doesn't break due to extra padding. Try to remove this code to see the effect.

# Cities

[London](#)

[Paris](#)

[Tokyo](#)

## London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.



# LAYOUT VIA DIVISIONS

Untuk waktu yang lama, tata letak halaman web didasarkan pada penggunaan elemen div

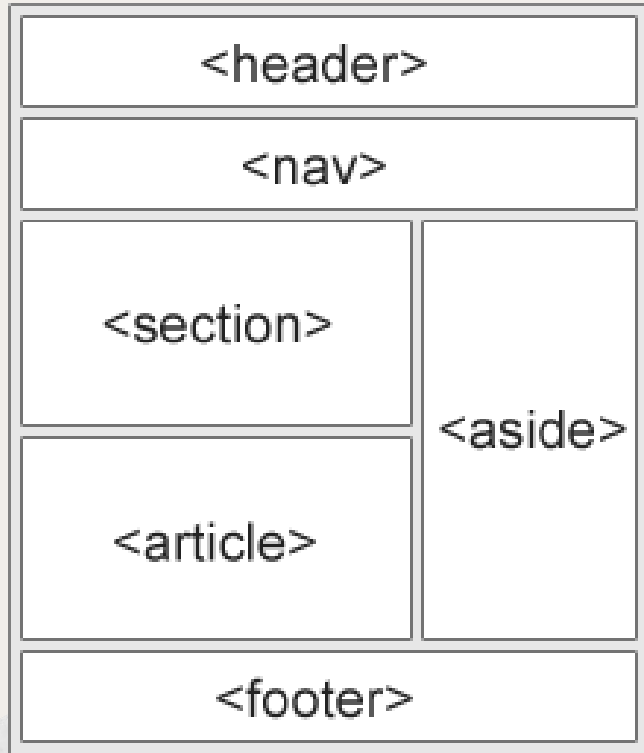
Halaman web biasanya terdiri dari kira-kira beberapa elemen div sebagai berikut (mungkin dengan elemen div tambahan di dalam masing-masing div)

```
<div id="header"> ... </div>
<div id="nav"> ... </div>
<div id="main">
  <div id="content"> ... </div>
  <div id="ads"> ... </div>
</div>
<div id="footer"> ... </div>
```

Layout adalah masalah mengatur tata letak elemen div tersebut.



# LAYOUT VIA HTML5 ELEMENTS



Semantic Elements (mendefinisikan isinya dengan jelas):

Elemen **<header>** berisi teks pengantar untuk dokumen, artikel, atau section

Sebuah elemen **<nav>** berisi satu set navigasi hyperlink

**<section>** berisi teks dengan tema yang sama, **<artikel>** berisi teks yang independen/konten mandiri (bisa *nested* satu sama lain)

Elemen **<aside>** mendefinisikan konten selain dari **<section>/ <artikel>** (seperti sidebar)

Sebuah elemen **<footer>** biasanya berisi informasi penulis dokumen, informasi hak cipta, tautan ke informasi kontak, dll



# POSITION

Properti **position** menentukan jenis posisi yang digunakan untuk elemen. Elemen kemudian diposisikan menggunakan properti **top**, **bottom**, **left**, dan **right**.

Jenis	Keterangan
static	Default. Elemen diposisikan sesuai dengan alur normal halaman.
fixed	Elemen selalu berada di tempat yang sama meskipun halaman di-scroll.
absolute	Elemen diposisikan relatif terhadap ancestor non-static (bila tidak ada, terhadap <body>). Bergerak bersama dengan pengguliran halaman.
relative	Diposisikan relatif terhadap posisi normalnya. Bergerak bersama dengan pengguliran halaman. Meninggalkan celah di posisi semula.
sticky	Diposisikan berdasarkan posisi scroll pengguna.



# CSS FLEXBOX LAYOUT

CSS Flexbox Layout Module mempermudah desain struktur halaman web secara fleksibel dan responsif tanpa menggunakan **float** dan **positioning**.

Untuk mulai menggunakan model Flexbox, kita harus menyiapkan elemen kontainer terlebih dahulu, dengan menyetel properti **display** menjadi **flex**.

Elemen turunan langsung dari kontainer fleksibel, secara otomatis menjadi item fleksibel.

```
<style>
.flex-container {
  display: flex;
  background-color: #f1f1f1;
}
</style>
```

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



# CONTOH FLEXBOX LAYOUT

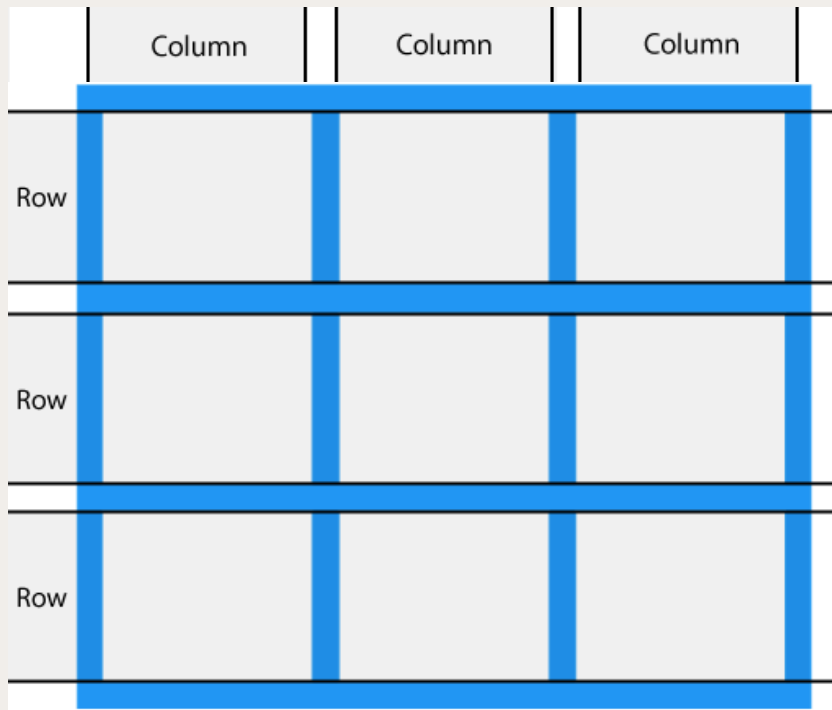
- Flex direction: row/column
  - Layout image 1 (row)
  - Layout image 2 (column)
- Website Layout:
  - Layout page 1
  - Layout page 2
- Form Layout:
  - Layout form 1
  - Layout form 2



# CSS GRID LAYOUT

CSS Grid Layout Module menawarkan sistem tata letak berbasis grid, dengan baris dan kolom, sehingga lebih mudah mendesain halaman web tanpa harus menggunakan **float** atau **positioning**.

CSS Grid terdiri dari elemen parent (*grid container*), dengan satu atau beberapa elemen child (*grid item*).







# CSS GRID LAYOUT

Untuk membuat elemen HTML menjadi *grid container*, kita harus menyetel properti **display** menjadi **grid**.

Semua elemen turunan langsung dari *grid container* secara otomatis menjadi *grid item*.

Beberapa property lainnya:

- **grid-template-columns** - mendefinisikan jumlah kolom dalam tata letak grid, dan dapat menentukan lebar setiap kolom.
- **grid-template-rows** - mendefinisikan ketinggian setiap baris.
- **grid-area** - dapat digunakan untuk menetapkan nama untuk *grid item*. *Grid item* yang telah diberi nama dapat dirujuk oleh **grid-template-areas**.



# CONTOH GRID LAYOUT

- Image Layout
  - Layout image 1 (auto-fill)
  - Layout image 2 (flex factor/fr)
- Website Layout:
  - Layout page 1
  - Layout page 2
- Form Layout:
  - Layout form 1
  - Layout form 2



# STYLE GUIDE

- HTML dan CSS menyediakan banyak fitur, tetapi ini harus digunakan dengan bijaksana.
- Gunakan warna dan font (non-default) dengan hati-hati. terutama menggunakan font gelap dengan latar belakang terang.
- 8-10% orang memiliki beberapa tipe kebutaan warna. Hindari kombinasi warna merah / hijau.
- Beberapa orang menggunakan *screen reader* untuk membaca konten halaman web. Selalu sertakan properti alt untuk gambar.
- Gunakan gambar dengan tepat. Hindari gambar latar belakang yang cerah yang membuat teks di depannya sulit dibaca. Hindari gambar yang dapat diklik alih-alih tombol standar untuk tautan karena dapat memperlambat pengunduhan laman.



# STYLE GUIDE

- Bagi halaman web yang besar menjadi beberapa halaman yang lebih kecil atau sediakan menu untuk navigasi
- Pastikan konsistensi di seluruh halaman web. Tetap berpegang pada fitur standar dan uji di beberapa browser.
- Gunakan relative units untuk mengatur font size, bukan fixed pixel size.
- Jangan bergantung pada ukuran jendela tertentu atau ukuran font khusus untuk tata letak karena ukuran tersebut mungkin saja berubah. gunakan desain yang adaptif atau responsif.



# RESPONSIVE DESIGN

Halaman web dapat dilihat menggunakan berbagai perangkat: desktop, tablet, dan telepon. Halaman web tidak boleh meninggalkan informasi apapun dan menyesuaikan kontennya dengan perangkat apa pun.

Desain web responsif (RWD) adalah menggunakan CSS dan HTML untuk mengubah ukuran, menyembunyikan, mengecilkan, memperbesar, atau memindahkan konten agar terlihat bagus di layar mana pun.

Desain responsif menggunakan satu halaman web dan gaya. Penyesuaian dengan jenis perangkat pengguna dilakukan melalui penggunaan **media query, flexible grid, relative unit**, dan **responsive image**.

Berbeda dengan *adaptive design*, yang menggunakan halaman web / atau *style* yang berbeda tergantung pada perangkat pengguna.

# Thanks!

Do you have any questions?



wilantika@stis.ac.id

lya@stis.ac.id

yeni.rima@stis.ac.id

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik