



# {CSS}

Nori Wilantika

Politeknik Statistika STIS

# CSS Zen Garden

## The Beauty of CSS Design

A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.

Download the example [html file](#) and [css file](#)

## The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.

We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP, and the major browser creators.

The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

## So What is This About?

There is a continuing need to show the power of CSS. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The HTML remains the same, the only thing that has changed is the external CSS file. Yes, really.

# Zen Garden

The Beauty of CSS Design

*A demonstration of what  
can be accomplished  
through [CSS](#)-based design.  
Select any style sheet from  
the list to load it into this  
page.*

*Download the example  
[html file](#) and [css file](#)*

remains the same, the only thing that has changed is the external [CSS](#) file. Yes, really.

## *The Road to Enlightenment*

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible [DOMs](#), broken [CSS](#) support, and abandoned browsers.

We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the [W3C](#), [WaSP](#), and the major browser creators.

The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

## *So What is This About?*

There is a continuing need to show the power of [CSS](#). The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The [HTML](#)



---

**Mid Century Modern** by  
**Andrew Lohman**

---

**Garments** by **Dan Mall**

---

**Steel** by **Steffen  
Knoeller**

---

**Apothecary** by **Trent  
Walton**

---

**Screen Filler** by **Elliot  
Jay Stocks**

---

**Fountain Kiss** by  
**Jeremy Carlson**

---

**A Robot Named Jimmy**  
by **meltmedia**

---

**Verde Moderna** by  
**Dave Shea**

**First Name**

**Last Name**

**Country**  ▾

**First Name**

**Last Name**

**Country**

▾



# SEJARAH CSS

- HTML dibuat untuk mendeskripsikan konten halaman web, seperti:  
`<h1> Ini adalah sebuah heading </h1>`  
`<p> Ini adalah paragraf. </p>`
- Pada HTML 3.2, tags dan atribut untuk pengaturan tampilan terdapat pada HTML. Pengaturan font dan warna ditambahkan ke setiap halaman, menjadi proses yang panjang dan mahal.
- Untuk mengatasi masalah ini, World Wide Web Consortium (W3C) membuat CSS.
- Pada HTML 5, sebagian besar presentasi atau tampilan dipisahkan dari halaman HTML.
- CSS3 adalah versi terbaru dari CSS





# APA ITU CSS?

- CSS adalah singkatan dari **C**ascading **S**tyle **S**heets
- CSS menjelaskan bagaimana elemen HTML ditampilkan di layar, di kertas, atau di media lain.
- Disebut **cascading** karena dapat didefinisikan pada level yang berbeda:
  1. **Inline styles**, berlaku untuk satu elemen HTML
  2. **Document style sheets** (embedded styles/internal style sheets) berlaku untuk semua elemen dalam elemen body pada suatu dokumen HTML
  3. **External style sheets**, dapat ditautkan dan diterapkan ke berbagai HTML dokumen





# INLINE STYLES

- Inline styles dapat digunakan untuk menerapkan gaya unik untuk satu elemen.
- Untuk menggunakan inline style, tambahkan atribut **style** ke suatu elemen HTML. Dengan **value**, daftar urutan **property:value** yang dipisahkan oleh titik koma.

```
<tag style="property1:value1; property2:value2;"> </tag>

<h1 style="color:blue; text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

- Atribut style dapat berisi properti CSS apa pun.



```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

**This is a heading**

**This is a paragraph.**



## INTERNAL STYLE SHEETS

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h1{
      color:blue;
      text-align:center;
    }

    p {
      color:red;
    }
  </style>
</head>

<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

## EXTERNAL STYLE SHEETS

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="mystyle.css">
</head>
<body>

  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>

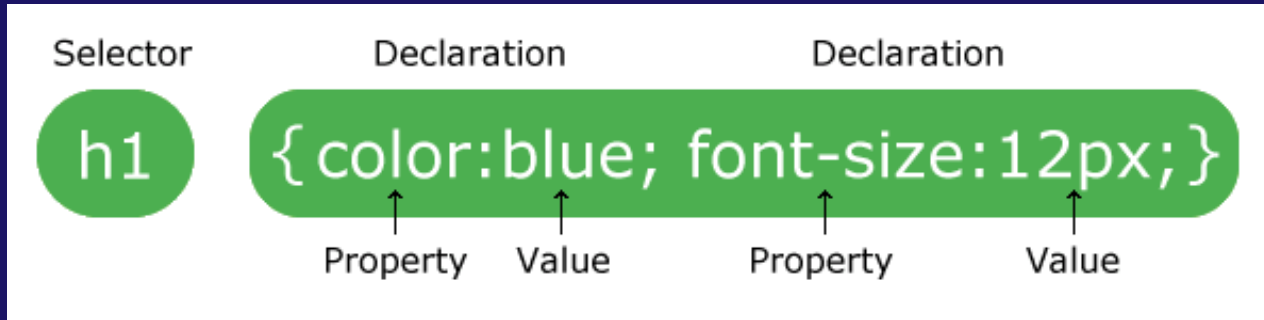
</body>
</html>
```

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```



# CSS SYNTAX



```
/* This is a single-line comment */
```





# CSS PROPERTIES

- Referensi:  
<https://www.w3schools.com/cssref/default.asp>  
<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>
- Mencakup tiga aspek gaya dan tata letak:
  1. **Styling texts**, berkaitan dengan font, warna teks, gaya item pada list, dll
  2. **Styling boxes**, berkaitan dengan pengaturan properti kotak, seperti ukuran, warna, padding, border, margin, dll.
  3. **Aranging boxes**, berhubungan dengan posisi relatif kotak satu sama lain.
- Penjelasan **styling boxes** dan **aranging boxes** pada video CSS Bagian 2.





# FONT PROPERTIES

Property	Possible Values
<code>font-family</code>	Font Families seperti Serif, Sans-Serif, Monospace, Cursive, Fantasy
<code>font-size</code>	length unit, percentage, xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, inherit
<code>font-style</code>	normal, italic, oblique
<code>font-weight</code>	normal, bold
<code>font-variant</code>	normal, small-caps





# TEXT FORMATTING

Property	Possible Values
color	color name, HEX value, RGB
text-align	center, left, right, justify
text-decoration	overline, line-through, underline, underline overline
text-transform	uppercase, lowercase, capitalize
text-indent	<i>length, percentage</i>
letter-spacing	normal, <i>length</i> , initial, inherit
word-spacing	normal, <i>length</i> , initial, inherit





# SIMPLE SELECTORS

- **Element selector**, memilih elemen HTML berdasarkan nama elemen.
- **Id selector**, menggunakan atribut id dari elemen HTML untuk memilih elemen tertentu. Tidak boleh ada lebih dari satu elemen dengan id yang sama. Untuk memilih elemen dengan id tertentu, tulis karakter hash (#), diikuti dengan id elemen.
- **Class selector**, memilih elemen HTML dengan atribut kelas tertentu. Untuk memilih elemen dengan kelas tertentu, tulis karakter titik (.), Diikuti dengan nama kelas.
- **Universal selector**, selector universal (\*) memilih semua elemen HTML di halaman.
- **Grouping selector**, memilih semua elemen HTML dengan definisi gaya yang sama.



# MULTI CLASSES

Perhatikan pengaturan style berikut:

```
p.alert {text - decoration : underline ;      padding - bottom : 10 px ;  
        background - color : orange ;        font - weight :bold ;}  
  
p.blue  {color : blue ;}
```

dengan HTML markup berikut:

```
<p class="alert blue"> What colors do I have?"</p>
```

What colours do I have?

Properti paragraf berasal dari p.alert dan dari p.blue.

# MULTI CLASSES

```
.red1 { color : red ; background - color : yellow ; margin : 5 px ; }  
.blue { color : blue ; margin : 5 px ; }  
.red2 { color : red ; background - color : yellow ; margin : 5 px ; }
```

```
<p class ="red1 blue"> class ="red1 blue", blue after red1 in style sheet</p>  
<p class ="blue red1"> class ="blue red1", blue after red1 in style sheet</p>  
<p class ="red2 blue"> class ="red2 blue", red2 after blue in style sheet</p>  
<p class ="blue red2"> class ="blue red2", red2 after blue in style sheet</p>
```

class="red1 blue", blue after red1 in style sheet

class="blue red1", blue after red1 in style sheet

class="red2 blue", red2 after blue in style sheet

class="blue red2", red2 after blue in style sheet

Pada pengaturan yang "bentrok",  
yang diterapkan adalah yang  
ditulis paling akhir.



# MULTI CLASSES

```
.red1.blue { color : white ; background - color : orange ; }  
.red1 { color : red ; background - color : yellow ; margin : 5 px ; }  
.blue { color : blue ; margin : 5 px ; }  
.red2 { color : red ; background - color : yellow ; margin : 5 px ; }
```

```
<p class ="red1 blue"> class ="red1 blue", blue after red1 in style sheet</p>  
<p class ="blue red1"> class ="blue red1", blue after red1 in style sheet</p>  
<p class ="red2 blue"> class ="red2 blue", red2 after blue in style sheet</p>  
<p class ="blue red2"> class ="blue red2", red2 after blue in style sheet</p>
```

class="red1 blue", blue after red1 in style sheet

class="blue red1", blue after red1 in style sheet

class="red2 blue", red2 after blue in style sheet

class="blue red2", red2 after blue in style sheet

.class1.class2: css memilih semua elemen dengan set nama1 dan nama2 dalam atribut classnya.

≠ .class1. class2



# RELATIVE SELECTORS

Selector	Penulisan	Contoh	Keterangan
Descendant	element element	ol a	Semua elemen a yang di dalam elemen ol
Child	element>element	ol>li	Semua elemen li dimana orang tuanya adalah elemen ol
Adjacent sibling	element+element	p+ol	Semua elemen ol yang ditempatkan segera setelah elemen p
General sibling selector	element~element	p~ol	Semua elemen ol yang diawali dengan elemen p



# RELATIVE SELECTORS

```
ul a {color: red; text-decoration: none;} /* hyperlink in ol */
ol>li {font-style: italic;} /* li children of ol*/
p+ol {font-weight: bold;} /* ol immediately after p */
p~ol {background-color: yellow;} /* ol after p */
```

```
<p> A paragraph with a <a href="/index.php"> hyperlink </a></p>
<ol><li> A list item in an ordered list with
    a <a href="/index.php"> hyperlink </a></li></ol>
<ul><li> A list item in an unsorted list with
    a <a href="/index.php"> hyperlink </a></li></ul>
<ol><li> A list item in an ordered list with
    a <a href="/index.php"> hyperlink </a></li></ol>
```

A paragraph with a [hyperlink](#)

*1. A list item in an ordered list with a [hyperlink](#)*

- A list item in an unsorted list with a [hyperlink](#)

*1. A list item in an ordered list with a [hyperlink](#)*



# PSEUDO-CLASS SELECTORS

Digunakan untuk mendefinisikan style dari suatu elemen pada **keadaan tertentu**, misalnya:

- apakah tautan telah dikunjungi
- apakah checkbox telah dicentang
- posisi penunjuk mouse di atas elemen, dll

```
selector:pseudo-class {property: value;}
```





# PSEUDO-ELEMENTS SELECTORS

Digunakan untuk mendefinisikan style dari suatu elemen pada **bagian tertentu**.

```
selector::pseudo-element {property: value;}
```

Selector	Contoh	Penjelasan
<code>::after</code>	<code>h1::after</code>	Sisipkan sesuatu setelah konten dari setiap elemen <code>&lt;h1&gt;</code>
<code>::before</code>	<code>h1::before</code>	Sisipkan sesuatu sebelum konten dari setiap elemen <code>&lt;h1&gt;</code>
<code>::first-letter</code>	<code>p::first-letter</code>	Memilih huruf pertama dari setiap <code>&lt;p&gt;</code> elemen
<code>::first-line</code>	<code>p::first-line</code>	Memilih baris pertama dari setiap <code>&lt;p&gt;</code> elemen
<code>::marker</code>	<code>::marker</code>	Memilih penanda item <b>list</b>
<code>::selection</code>	<code>p::selection</code>	Memilih bagian elemen yang dipilih oleh pengguna



# This is a heading 😊



## This is a heading

To stop the degradation of the planet's natural environment and to build a future in which humans live in harmony with nature, by; conserving the world's biological diversity, ensuring that the use of renewable natural resources is sustainable, and promoting the reduction of pollution and wasteful consumption.

- Coffee
- Tea
- Milk

1. First
2. Second
3. Third



# ATTRIBUTE SELECTORS

Selector	Contoh	Penjelasan
<code>[attribute]</code>	<code>[target]</code>	Memilih semua elemen dengan atribut <code>target</code>
<code>[attribute=value]</code>	<code>[target=_blank]</code>	Memilih semua elemen dengan <code>target="_blank"</code>
<code>[attribute =value]</code>	<code>[class =top]</code>	Memilih semua elemen dengan nilai atribut <code>class</code> yang dimulai dengan kata <code>"top"</code>
<code>[attribute^=value]</code>	<code>[class^="top"]</code>	Memilih semua elemen dengan nilai atribut <code>class</code> yang dimulai dengan <b>substring</b> <code>"top"</code>
<code>[attribute~=value]</code>	<code>[title~=flower]</code>	Memilih semua elemen dengan atribut judul yang mengandung kata <code>"bunga"</code>
<code>[attribute*=value]</code>	<code>[title*="flow"]</code>	Memilih semua elemen dengan atribut judul yang mengandung <b>substring</b> <code>"bunga"</code>
<code>[attribute\$=value]</code>	<code>a[href\$=".pdf"]</code>	Memilih setiap elemen <code>&lt;a&gt;</code> yang nilai atribut <code>href</code> -nya diakhiri dengan <code>".pdf"</code>



```
<h1 class="top-header">Welcome</h1>
<p class="top-text">Hello world!</p>
<p class="topcontent">Are you learning CSS?</p>
<p class="top.content">Is it fun?</p>
```

```
<style>
[class|=top] {background: yellow;}
</style>
```

**Welcome**

Hello world!

Are you learning CSS?

Is it fun?

```
<style>
[class^="top"] {background: yellow;}
</style>
```

**Welcome**

Hello world!

Are you learning CSS?

Is it fun?



```
  
  

```

```
<style>  
[title~=flower] {border: 5px solid yellow;}  
</style>
```



```
<style>  
[title~=flow] {border: 5px solid yellow;}  
</style>
```





# CASCADING ORDER

Style mana yang akan digunakan jika ada lebih dari satu style yang ditentukan untuk elemen HTML?

Semua gaya dalam halaman akan "bertingkat" menjadi lembar gaya "virtual" baru dengan aturan prioritas berikut:

- Inline style (di dalam elemen HTML)
- External dan internal style sheets
- Default browser

Jadi, inline style memiliki prioritas tertinggi, dan akan menggantikan external dan internal style sheets serta default browser.





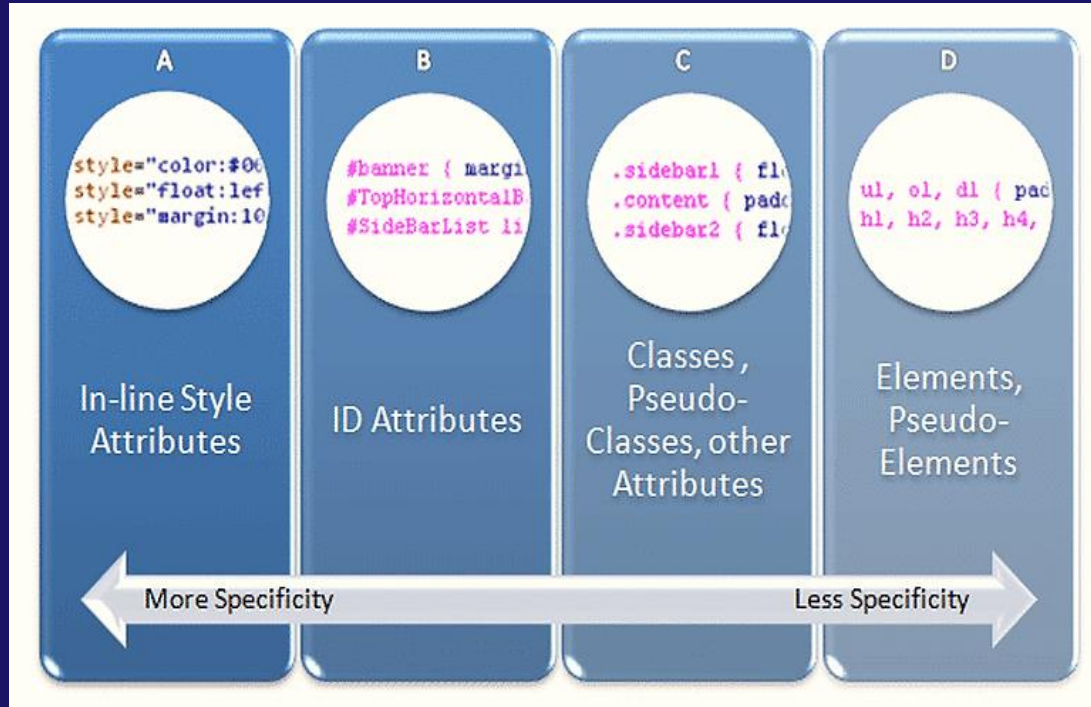
# SPECIFICITY

- Beberapa jenis pengaturan style dapat diterapkan ke elemen HTML yang sama
- Mungkin juga pengaturan style ini memberikan nilai yang berbeda ke properti yang sama
- Jika ada dua atau lebih aturan CSS yang bertentangan yang mengarah ke elemen yang sama, browser mengikuti beberapa aturan untuk menentukan aturan mana yang paling spesifik dan dipilih untuk diterapkan.
- Specificity adalah alasan umum mengapa aturan CSS Anda tidak berlaku untuk beberapa elemen, meskipun menurut Anda seharusnya berlaku.





# SPECIFICITY HIERARCHY



- Bila specificity sama, berlaku aturan “penulisan terakhir”.
- Universal selector (\*) dan nilai yang diwariskan (inherited) memiliki spesifisitas paling rendah



```
h1      {background-color:yellow; color:blue;}
.orange {background-color:orange; text-decoration:underline;}
#i1     {background-color:red; font-style:italic;}
```

```
<h1 id="i1" class="orange"> What properties does this heading have? </h1>
```

*What properties does this heading have?*

A: h1

B: #content h1

C: `<div id="content"><h1 style="color: #ffffff">Heading</h1></div>`

	Inline	ID	Class	Element
A	0	0	0	1
B	0	1	0	1
C	1	0	0	0

1 < 101 < 1000

# INHERITANCE

Inheritance mengontrol apa yang terjadi ketika tidak ada nilai yang secara eksplisit dikhususkan untuk properti elemen.

```
.alert {text-decoration:underline; padding-bottom:10px;  
        background-color:orange; font-weight:bold;}
```

```
<ol class="alert">  
  <li> Help Me! </li>  
  <li> I'm trapped on a COMP519 slide ! </li>  
</ol>
```

1. Help Me!
2. I'm trapped on a COMP519 slide!

```
<ol>  
  <li class="alert"> Help Me! </li>  
  <li> I'm trapped on a COMP519 slide! </li>  
</ol>
```

1. Help Me!
2. I'm trapped on a COMP519 slide!

```
<body>
```

```
<h1>Hello World</h1>
```

```
<a href="#"> Kembali ke Halaman Utama </a>
```

```
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis ultricies consectetur ex, nec accumsan quam  
tristique id. Donec porttitor maximus molestie. Vivamus et orci ac est rutrum consectetur ut vitae ante. Vivamus  
id nibh dui. Nulla luctus enim elit, in suscipit eros viverra id. Donec pellentesque consectetur metus quis  
volutpat. Maecenas dapibus porta lacus, quis ultrices eros vehicula ut. </p>
```

```
</body>
```

## Hello World

[Kembali ke Halaman Utama](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis ultricies consectetur ex, nec accumsan quam tristique id. Donec porttitor maximus molestie. Vivamus et orci ac est rutrum consectetur ut vitae ante. Vivamus id nibh dui. Nulla luctus enim elit, in suscipit eros viverra id. Donec pellentesque consectetur metus quis volutpat. Maecenas dapibus porta lacus, quis ultrices eros vehicula ut.

```
body{  
  font-family:arial;  
  color:darkgrey;  
}
```

## Hello World

[Kembali ke Halaman Utama](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis ultricies consectetur ex, nec accumsan quam tristique id. Donec porttitor maximus molestie. Vivamus et orci ac est rutrum consectetur ut vitae ante. Vivamus id nibh dui. Nulla luctus enim elit, in suscipit eros viverra id. Donec pellentesque consectetur metus quis volutpat. Maecenas dapibus porta lacus, quis ultrices eros vehicula ut.

```
a{color:inherit;}
```

# BAGIAN 2: STYLING AND ARRANGING BOXES









**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)

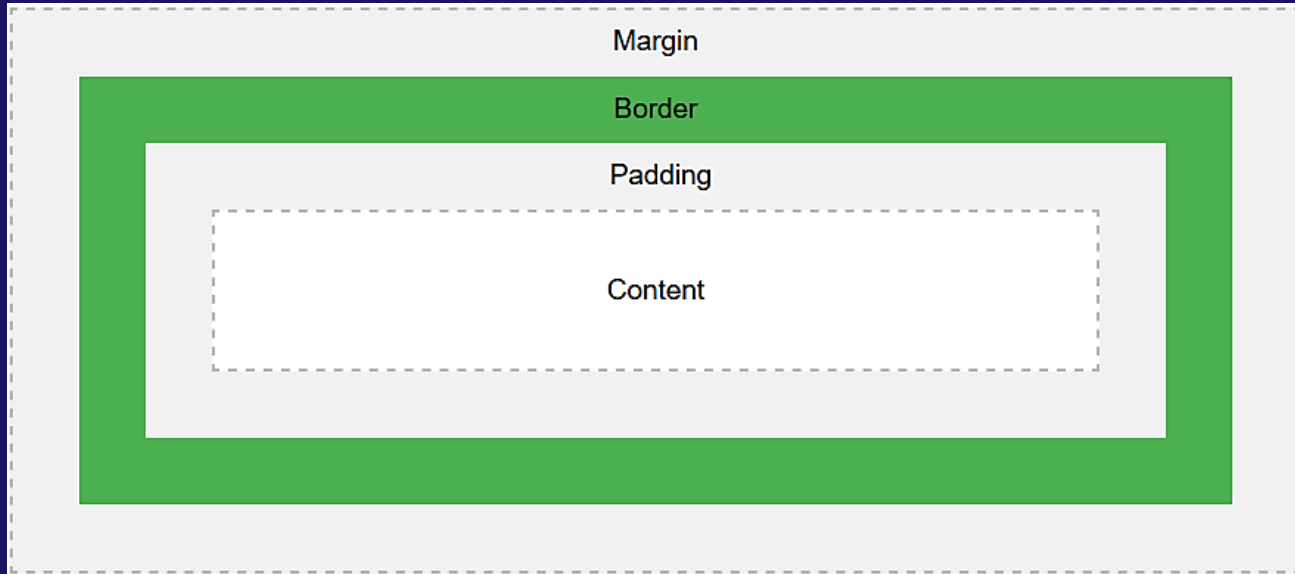
**Aplikasi Allstats BPS**





# CSS BOX MODEL

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



# 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` digunakan untuk penentuan posisi dan format konten, 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.

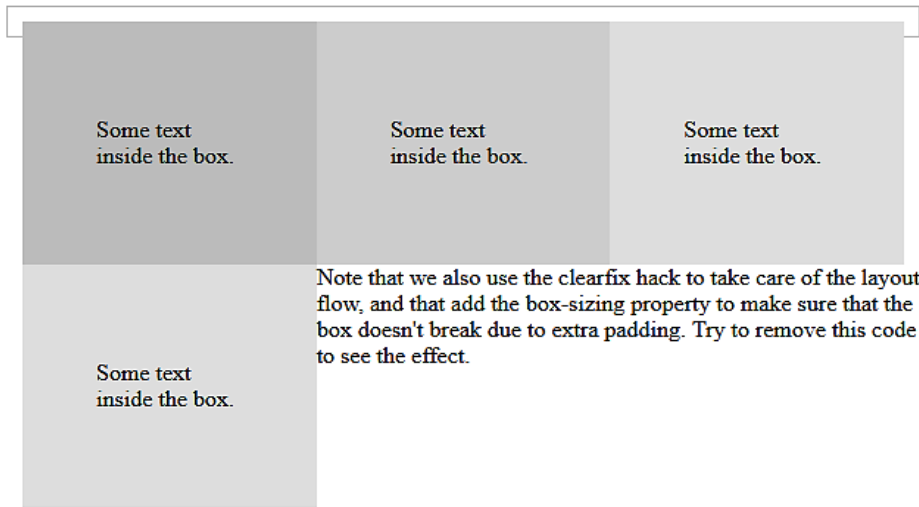
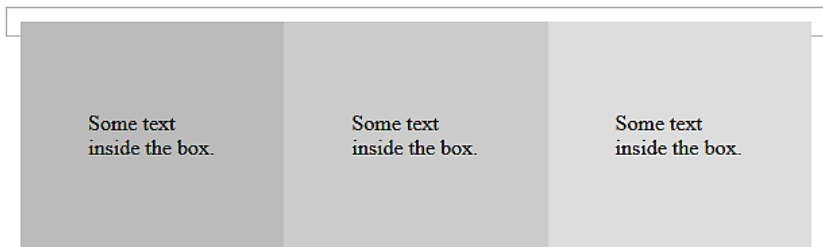
Some text  
inside the box.

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.

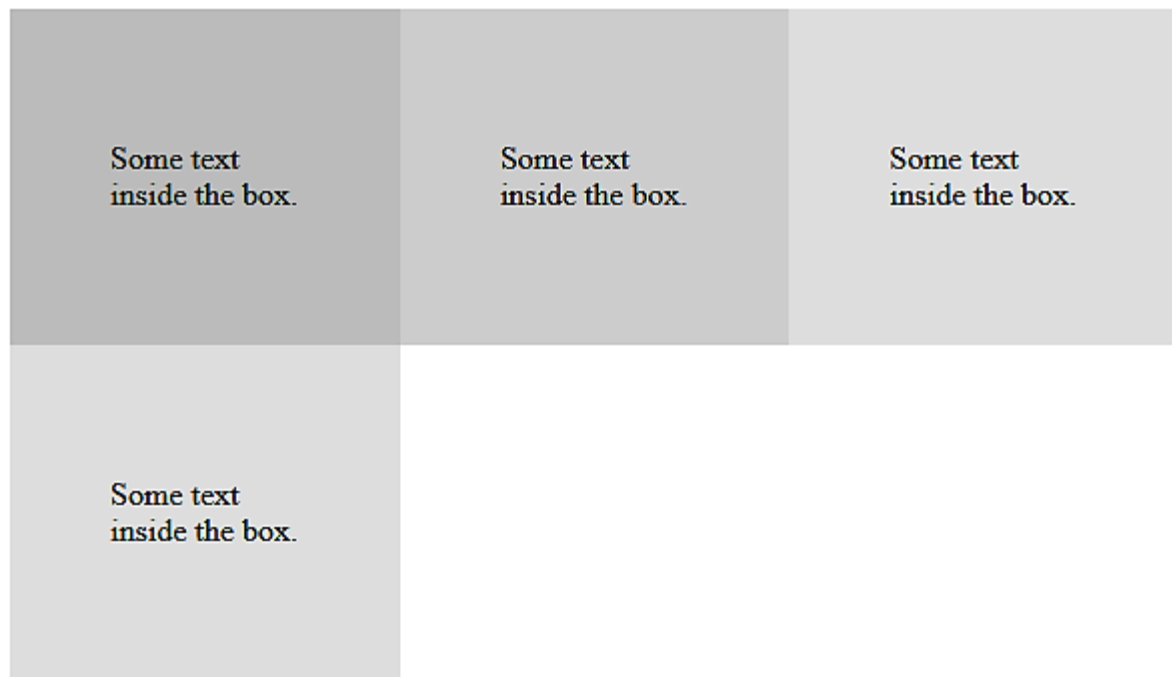
## With clear

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.



# POSITION

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

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





# 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)

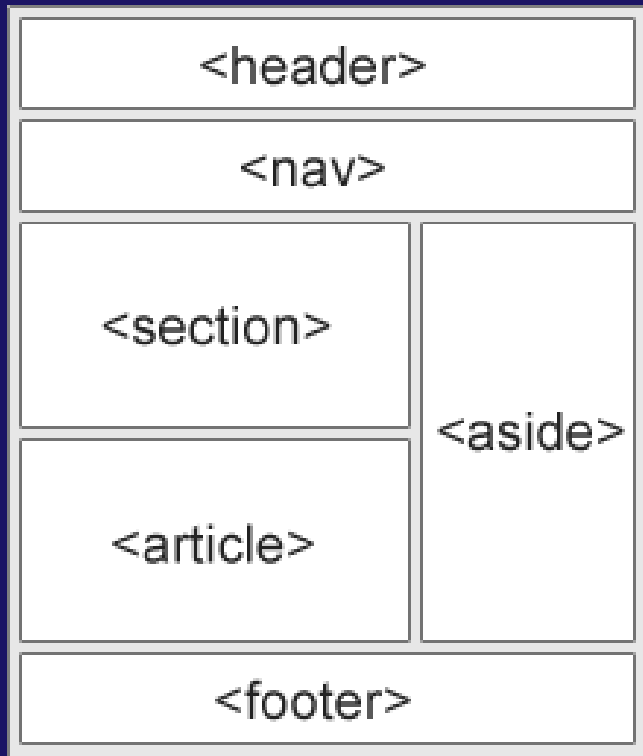
```
<div id="header"> ... </div>
<div id="nav"> ... </div>
<div id="main">
  <div id="content"> ... </div>
  <div id="ads"> ... </div>
</div>
<div id="footer"> ... </div>
```

- 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





# 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>
```

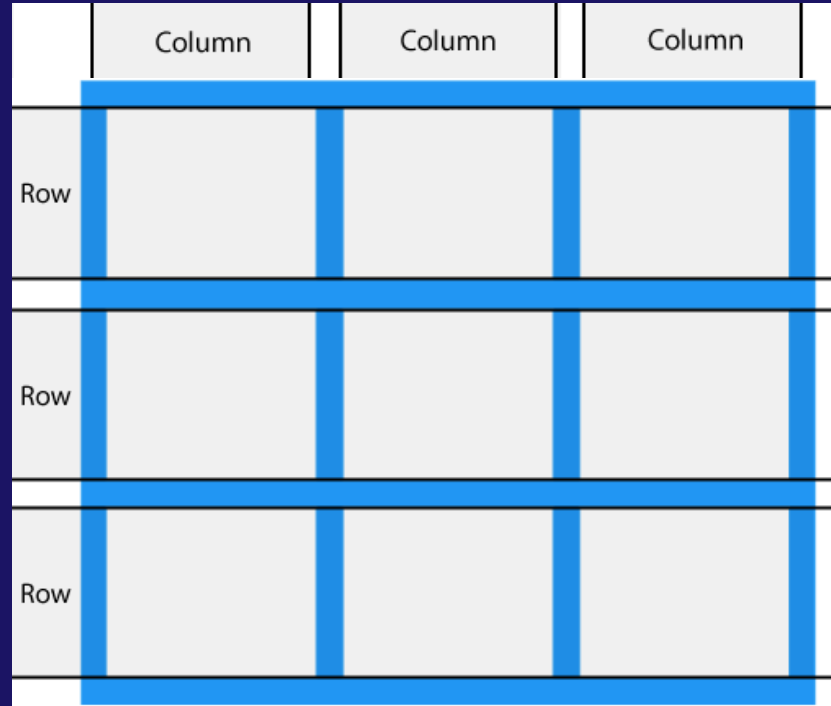




# CSS GRID LAYOUT

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`.





# 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.



# CREDITS

- Template presentasi dibuat oleh Slidesgo dengan icon dari Flaticon dan gambar dari Freepik.
- Materi disadur dari:
  1. Jackson, J. C. (2007). Web Technologies: A Computer Science Perspective. Pearson/Prentice Hall.
  2. COMP519 Web Programming (2020-21). Diakses pada Januari 2021, dari <https://cgi.csc.liv.ac.uk/~ullrich/COMP519/>
  3. W3Schools Online Web Tutorials. Diakses pada Januari 2021, dari <https://www.w3schools.com/>
  4. Web Programming UNPAS. Diakses pada Februari 2021, dari <https://www.youtube.com/channel/UCKXmLjEr95LVtGuIm3l2dPg>



SEKIAN.