

The slide features abstract geometric designs in the corners. The top-left corner is filled with overlapping triangles in shades of blue, green, and red. The bottom-right corner contains a cluster of overlapping triangles in various shades of gray.

# CSS (lanjut V)



Cascading Style Sheets

# Tujuan Pembelajaran

1

## **CSS Animation**

Membuat animasi dengan css.

2

## **Box sizing**

Memberikan untuk web browser.

3

## **CSS Responsive**

Membuat layout website yang responsive.

# CSS Animation

---

- Animasi memungkinkan elemen berangsur-angsur berubah dari satu gaya ke gaya lainnya.
- Anda dapat mengubah properti CSS sebanyak yang Anda inginkan, sebanyak yang anda inginkan.
- Untuk menggunakan animasi CSS, Anda harus terlebih dahulu menentukan beberapa kerangka kunci untuk animasi.
- `@keyframes` menampung gaya apa yang akan dimiliki elemen pada waktu tertentu.

## Aturan `@keyframes`

Saat Anda menentukan gaya CSS di dalam `@keyframes` aturan, animasi secara bertahap akan berubah dari gaya saat ini ke gaya baru di waktu tertentu.

Agar animasi berfungsi, Anda harus mengikat animasi ke suatu elemen.

# CSS Animation

---

Property animation :

- @keyframes
- animation-name
- animation-duration
- animation-delay
- animation-iteration-count
- animation-direction
- animation-timing-function
- animation-fill-mode
- animation

# CSS Animation Property

---

Tag	Deskripsi
@keyframes	Penampung gaya pada animasi.
Animation-name	Untuk memanggil nama animasi yang telah dibuat.
Animation-duration	Memberikan durasi pada animasi.
Animation-delay	Memberikan waktu tunggu/delay pada animasi
Animation-iteration-count	Menentukan berapa kali animasi harus dijalankan.

# CSS Animation Property

---

Tag	Deskripsi
animation-direction	Menentukan apakah animasi harus dimainkan depan, belakang atau dalam siklus alternatif.
animation-timing-function	Menentukan kurva kecepatan animasi.
animation-fill-mode	Menentukan gaya untuk elemen target ketika animasi tidak bermain.
animation	Penulisan singkat untuk property animasi

# CSS Animation (from-to)

---

Buat satu buah div dengan gaya css seperti berikut :

```
/* Membuat Animasi */
@keyframes contoh {
  from {background-color: red;}
  to {background-color: yellow;}
}

/* Menggunakan animasi */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: contoh;
  animation-duration: 4s;
}
```



# CSS Animation (Persen)

---

Membuat animasi dengan nilai persen membuat kita menjadi dapat memberikan lebih banyak gaya.

```
/* Membuat Animasi */
@keyframes contoh {
  0% {background-color:red; left:0px; top:0px;}
  25% {background-color:yellow; left:200px; top:0px;}
  50% {background-color:blue; left:200px; top:200px;}
  75% {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}

/* Menggunakan Animasi */
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: contoh;
  animation-duration: 4s;
}
```





# CSS Animation (animation-delay)

---

Properti animation-delay menetapkan penundaan untuk memulai animasi.

Nilai yang diberikan berupa angka dengan satuan second(detik) sebagai waktu Tunggu.

## Sintaksis

```
animation-delay: 2s;
```

# CSS Animation

## (Animation-iteration-count)

---

Properti animation-iteration-count menentukan berapa kali animasi harus dijalankan.

### Sintaksis

```
animation-iteration-count: 3;
```

Dengan kode diatas maka animasi akan dijalankan 3 kali selain itu untuk membuatnya berjalan secara terus menerus dapat memberikan nilai **infinite** .

# CSS Animation

## (Animation-direction)

---

Properti animation-direction menentukan apakah animasi harus dimainkan depan, belakang atau dalam siklus alternatif.

Properti arah-animasi dapat memiliki nilai berikut:

- normal- Animasi dimainkan seperti biasa (ke depan). Ini standar
- reverse - Animasi diputar dalam arah terbalik (mundur)
- alternate - Animasi diputar terlebih dahulu, lalu mundur
- alternate-reverse - Animasi diputar mundur pertama, lalu maju

# CSS Animation

## (Animation-direction)

---

Sintaksis

```
animation-direction: reverse;
```

# CSS Animation

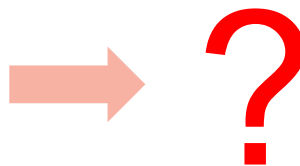
## (Animation-direction)

---

Membuat animasi dengan nilai persen membuat kita menjadi dapat memberikan lebih banyak gaya.

```
/* Membuat Animasi */
@keyframes contoh {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}

div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: contoh;
  animation-duration: 4s;
  animation-direction: reverse;
}
```



# CSS Animation

## (Animation-timing-function)

---

Properti animation-timing-function menentukan kurva kecepatan animasi.

Properti fungsi-waktu-animasi dapat memiliki nilai-nilai berikut:

- ease - Menentukan animasi dengan awal yang lambat, lalu cepat, lalu akhiri perlahan
- linear - Menentukan animasi dengan kecepatan yang sama dari awal hingga akhir
- ease-in - Menentukan animasi dengan awal yang lambat
- ease-out - Menentukan animasi dengan akhir yang lambat
- ease-in-out - Menentukan animasi dengan awal dan akhir yang lambat

# CSS Animation

## (Animation-timing-function)

---

Sintaksis

```
animation-timing-function: ease-in-out;
```

# CSS Animation

## (Animation-timing-function)

---

```
<div id="div1">linear</div>
<div id="div2">ease</div>
<div id="div3">ease-in</div>
<div id="div4">ease-out</div>
<div id="div5">ease-in-out</div>
```



```
div {
  width: 100px;
  height: 50px;
  background-color: red;
  font-weight: bold;
  position: relative;
  animation-name: geser;
  animation-duration: 5s;
  animation-iteration-count: infinite;
}

#div1 {animation-timing-function: linear;}
#div2 {animation-timing-function: ease;}
#div3 {animation-timing-function: ease-in;}
#div4 {animation-timing-function: ease-out;}
#div5 {animation-timing-function: ease-in-out;}

@keyframes geser {
  from {left: 0px;}
  to {left: 300px;}
}
```



# CSS Animation

## (Animation-fill-mode)

---

Properti animation-fill-mode menentukan gaya untuk elemen target ketika animasi tidak bermain (sebelum dimulai, setelah itu berakhir, atau keduanya).

Properti mode isi animasi dapat memiliki nilai-nilai berikut:

- none- Nilai standar. Animasi tidak akan menerapkan gaya apa pun pada elemen sebelum atau setelah dijalankan
- forwards - Elemen akan mempertahankan nilai style yang diatur oleh keyframe terakhir (tergantung pada animasi-arrah dan animasi-iterasi-hitung)
- backwards - Elemen akan mendapatkan nilai style yang diatur oleh keyframe pertama (tergantung pada arah animasi), dan mempertahankan ini selama periode penundaan animasi

# CSS Animation (Animation-fill-mode)

---

Sintaksis

```
animation-fill-mode: forwards;
```

# CSS Animation

## (Animation-fill-mode)

---

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  position: relative;  
  animation-name: contoh;  
  animation-duration: 3s;  
  animation-fill-mode: forwards;  
}  
  
@keyframes contoh {  
  from {top: 0px;}  
  to {top: 200px; background-color: blue;}  
}
```



# CSS Animation (Animation)

---

Properti singkat untuk animasi. Membuat penulisan menjadi singkat dengan satu property yang digabungkan :

```
div {  
  animation-name: contoh;  
  animation-duration: 5s;  
  animation-timing-function: linear;  
  animation-delay: 2s;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}
```



```
div {  
  animation: contoh 5s linear 2s infinite alternate;  
}
```

# Box-sizing

---

Properti CSS box-sizing memungkinkan kita untuk memasukkan padding dan border dalam lebar dan tinggi total elemen.

Secara default, lebar dan tinggi elemen dihitung seperti ini:

$\text{width} + \text{padding} + \text{border} = \text{lebar aktual suatu elemen}$

$\text{height} + \text{padding} + \text{border} = \text{tinggi aktual suatu elemen}$

Ini berarti: Ketika Anda mengatur lebar / tinggi elemen, elemen tersebut sering tampak lebih besar dari yang Anda atur (karena batas dan padding elemen ditambahkan ke lebar / tinggi elemen yang ditentukan).

# Box-sizing

```
<div class="div1">Div ini lebih kecil (lebar juga 300px dan tinggi 100px).</div>  
<br>  
<div class="div2">Div ini lebih besar (lebar juga 300px dan tinggi 100px).</div>
```

```
.div1 {  
  width: 300px;  
  height: 100px;  
  border: 1px solid blue;  
}
```

Div ini lebih kecil (lebar juga 300px dan tinggi 100px).

```
.div2 {  
  width: 300px;  
  height: 100px;  
  border: 1px solid red;  
  padding: 50px;  
}
```

Div ini lebih besar (lebar juga 300px dan tinggi 100px).

# Box-sizing

```
<div class="div1">Div ini lebih kecil (lebar juga 300px dan tinggi 100px).</div>  
<br>  
<div class="div2">Div ini lebih besar (lebar juga 300px dan tinggi 100px).</div>
```

```
.div1 {  
  width: 300px;  
  height: 100px;  
  border: 1px solid blue;  
  box-sizing: border-box;  
}
```

Div ini lebih kecil (lebar juga 300px dan tinggi 100px).

```
.div2 {  
  width: 300px;  
  height: 100px;  
  border: 1px solid red;  
  padding: 50px;  
  box-sizing: border-box;  
}
```

Div ini lebih besar (lebar juga 300px dan tinggi 100px).

# Web Responsif

---

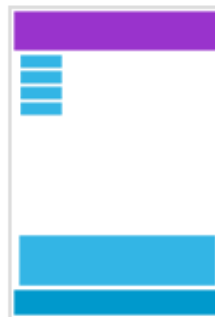
Desain web yang responsif membuat halaman web Anda terlihat bagus di semua perangkat.

Desain web responsif hanya menggunakan HTML dan CSS.

Desain web responsif bukan program atau JavaScript.



**Desktop**



**Tablet**



**Telepon**

Ini disebut desain web responsif ketika Anda menggunakan CSS dan HTML untuk mengubah ukuran, menyembunyikan, mengecilkan, memperbesar, atau memindahkan konten agar terlihat bagus di layar mana pun.



# Web Responsif - viewport

---

## Apa itu Viewport?

viewport adalah area yang terlihat pengguna dari halaman web.

viewport bervariasi dengan perangkat, dan akan lebih kecil di ponsel daripada di layar komputer.

## Mengatur Viewport

HTML5 memperkenalkan metode untuk memungkinkan perancang web mengendalikan viewport, melalui `<meta>` tag.

Anda harus memasukkan `<meta>` elemen viewport berikut di semua halaman web

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

Sebuah `<meta>` elemen viewport memberikan petunjuk pada browser bagaimana mengontrol dimensi halaman dan scaling.

Bagian ini `width=device-width` mengatur lebar halaman untuk mengikuti lebar layar perangkat (yang akan bervariasi tergantung pada perangkat).

Bagian ini `initial-scale=1.0` mengatur tingkat pembesaran awal saat halaman pertama kali dimuat oleh browser.

# Web Responsif - viewport



Tanpa tag meta viewport



Dengan tag meta viewport

# Web Responsif – media query

---

## Apa itu Media query?

**Media query** adalah teknik CSS yang diperkenalkan dalam CSS3.

Menggunakan `@media` untuk menyertakan blok properti CSS hanya jika kondisi tertentu benar.

## Contoh :

Jika jendela peramban 600px atau lebih kecil, warna latar belakang akan menjadi biru muda.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Responsif</title>
  <style>
    @media screen and (max-width: 600px) {
      body {
        background-color: lightblue;
      }
    }
  </style>
</head>
<body>

</body>
</html>
```

# Studi kasus 1



---

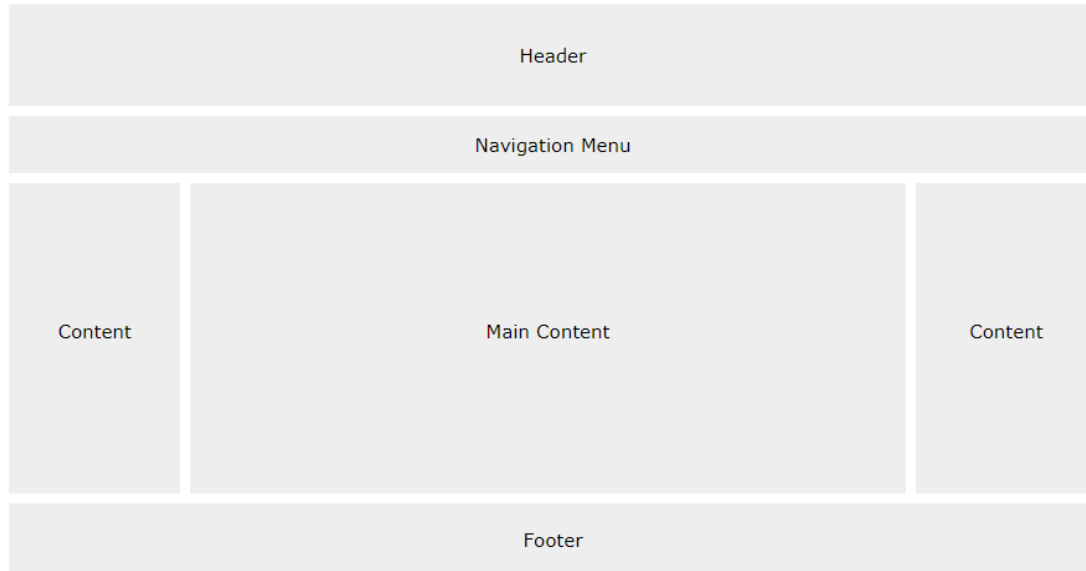
Folder script/study-kasus1

# Studi kasus 2



---

Membuat layout yang responsive :



# Referensi

- <https://www.w3schools.com/>



**Terima  
kasih**