

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 IV)



Cascading Style Sheets

Tujuan Pembelajaran

1

CSS Gradient

Membuat warna gradien dengan css..

2

CSS Shadow

Memberi efek shadow.

3

Membuat bentuk gambar

Membuat logo gambar dengan css.

CSS Gradient

Gradien CSS memungkinkan Anda menampilkan transisi yang mulus antara dua atau lebih warna yang ditentukan.

Dua jenis gradien:

- **Gradien Linear** (turun / naik / kiri / kanan / diagonal)
- **Radial Gradients** (ditentukan oleh pusatnya)

CSS Linear Gradient

Untuk membuat gradien linier, Anda harus menentukan setidaknya dua pemberhentian warna. Penghentian warna adalah warna yang ingin Anda transisikan dengan mulus. Anda juga dapat mengatur titik awal dan arah (atau sudut) bersama dengan efek gradien.

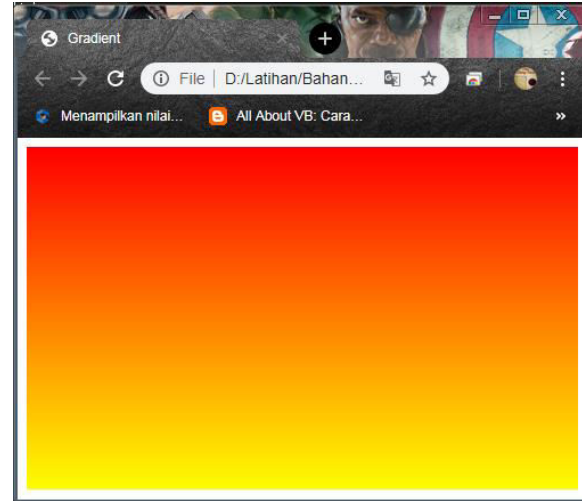
Sintaksis

```
background-image: linear-gradient(arah, warna1, warna2, warna..n);
```

Linear gradien – atas ke bawah

Default arah untuk linear gradient jika tidak diberi arah adalah dari atas ke bawah, seperti pada contoh berikut :

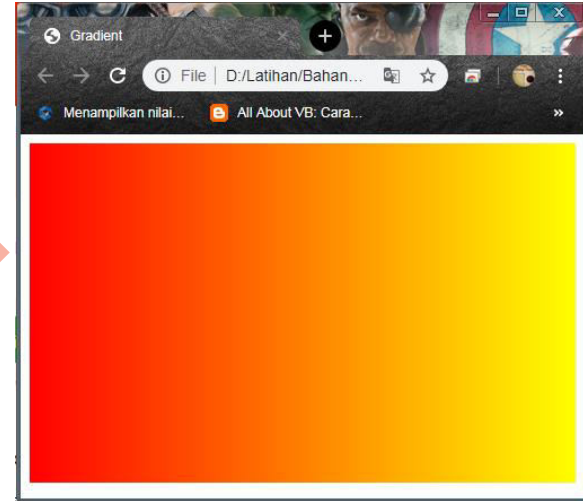
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Gradient</title>
  <style>
    div{
      width: 100%;
      height: 300px;
      background-image: linear-gradient(red,yellow);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



Linear gradien – kiri ke kanan

Untuk memberikan arah linear gradient dari atas ke bawah kita bisa menambahkan nilai arah, seperti pada contoh berikut :

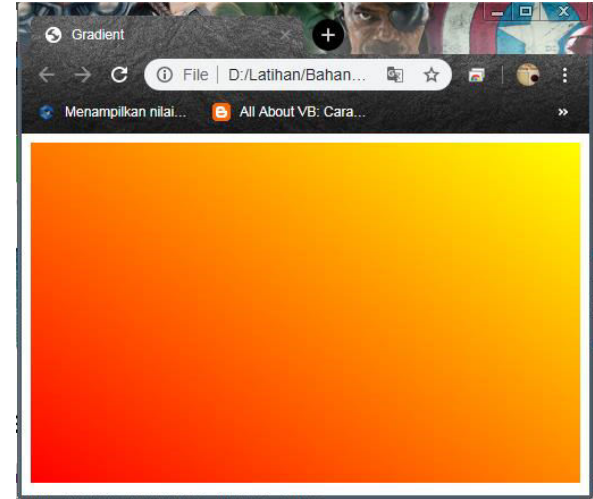
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Gradient</title>
  <style>
    div{
      width: 100%;
      height: 300px;
      background-image: linear-gradient(to right, red,yellow);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



Linear gradien – dengan angles

Jika Anda ingin lebih mengontrol arah gradien, Anda dapat menentukan sudut, alih-alih arah yang telah ditentukan (ke bawah, ke atas, ke kanan, ke kiri, ke kanan bawah, dll.). seperti pada contoh berikut :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Gradient</title>
  <style>
    div{
      width: 100%;
      height: 300px;
      background-image: linear-gradient(35deg, red,yellow);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



Radial gradien

Gradien radial ditentukan oleh pusatnya.

Untuk membuat gradien radial Anda juga harus menentukan setidaknya dua pemberhentian warna.

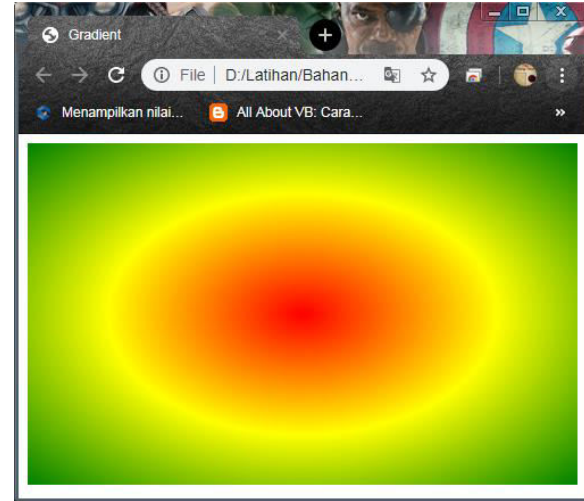
Sintaksis

```
background-image: radial-gradient(shape, warna1, warna2);
```


Radial gradien – default

Secara default, bentuk elips, ukuran sudut terjauh, dan posisi tengah. Contoh berikut menunjukkan gradien radial dengan pemberhentian warna yang berjarak sama:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Gradient</title>
  <style>
    div{
      width: 100%;
      height: 300px;
      background-image: radial-gradient(red,yellow,green);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```

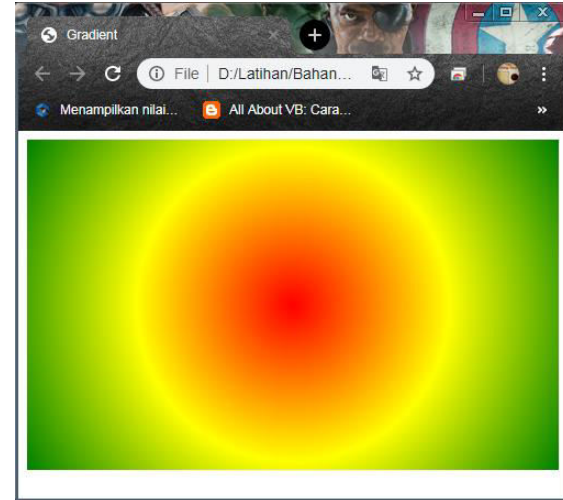


Radial gradien – circle

Parameter bentuk menentukan bentuk. Itu bisa mengambil lingkaran nilai atau elips . Nilai default adalah elips.

Contoh berikut menunjukkan gradien radial dengan bentuk lingkaran:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Gradient</title>
  <style>
    div{
      width: 100%;
      height: 300px;
      background-image: radial-gradient(circle,red,yellow,green);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



CSS Shadow

Dengan CSS Anda dapat menambahkan bayangan ke teks dan elemen. Properti shadow terdapat 2, seperti berikut:

- text-shadow
- box-shadow



Box Shadow

Dengan CSS Anda dapat membuat efek bayangan!

Text shadow

Property text-shadow menambahkan bayangan untuk teks.
Properti ini menerima daftar bayangan untuk diterapkan pada teks.

Sintaks

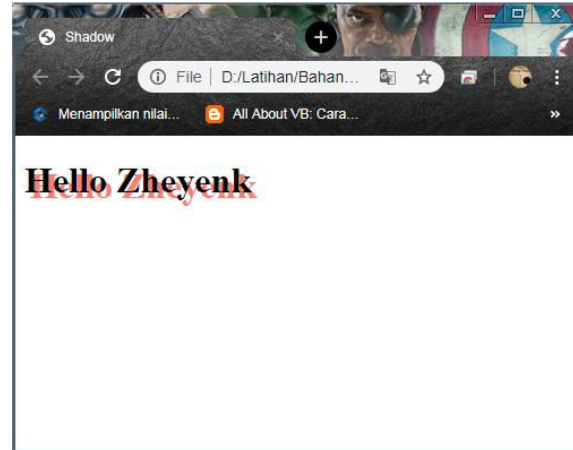
```
text-shadow: x-offset y-offset blur color;
```

X-offset => jarak sumbu x
Y-offset => jarak sumbu y
Blur => Ketebalan blur
Color => warna bayangan

Text shadow – tanpa blur

Nilai blur pada property text-shadow tidak diwajibkan, disini adalah contoh penggunaan text shadow tanpa blur :

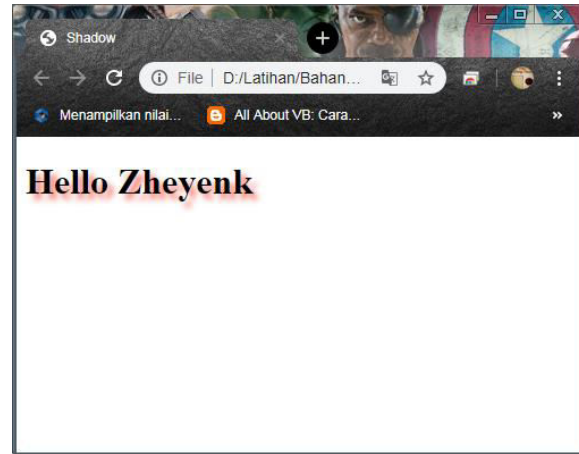
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Shadow</title>
  <style>
    h1{
      text-shadow: 5px 5px salmon;
    }
  </style>
</head>
<body>
  <h1>Hello Zheyenk</h1>
</body>
</html>
```



Text shadow – dengan blur

Nilai blur pada property text-shadow tidak diwajibkan, disini adalah contoh penggunaan text shadow dengan blur :

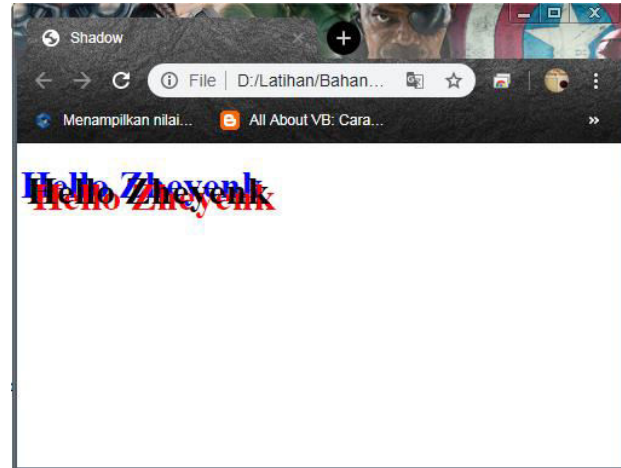
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Shadow</title>
  <style>
    h1{
      text-shadow: 5px 5px 5px salmon;
    }
  </style>
</head>
<body>
  <h1>Hello Zheyenk</h1>
</body>
</html>
```



Text shadow – multiple shadow

Kita dapat membuat lebih dari satu shadow dengan cara memisahkan dengan tanda Koma (,), seperti pada contoh berikut : (Memberikan bayangan dibawah dan atas)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Shadow</title>
  <style>
    h1{
      text-shadow: 5px 5px red,
                  5px -5px blue;
    }
  </style>
</head>
<body>
  <h1>Hello Zheyenk</h1>
</body>
</html>
```



Box shadow

Property box-shadow menambahkan bayangan untuk element.
Properti ini menerima daftar bayangan untuk diterapkan pada element.
Nilai blur spread inset merupakan nilai yang opsional.

Sintaks

```
box-shadow: x-offset y-offset blur spread color inset;
```

X-offset => jarak sumbu x

Y-offset => jarak sumbu y

Blur => Ketebalan blur

Spread => ketebalan bayangan

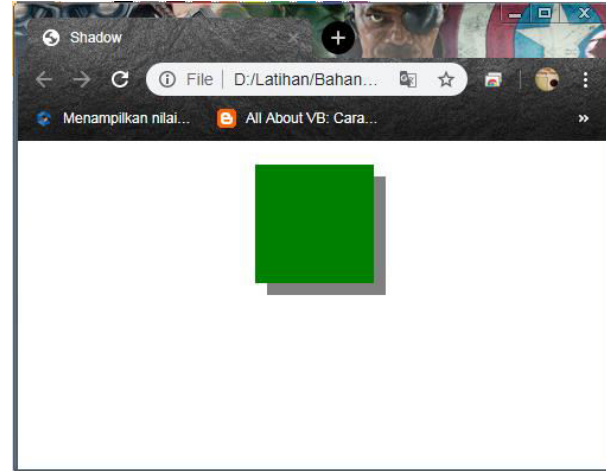
Color => warna bayangan

Inset => merubah arah bayangan ke dalam

Box shadow – tanpa blur

Nilai blur pada property box-shadow tidak diwajibkan, disini adalah contoh penggunaan Box-shadow tanpa blur :

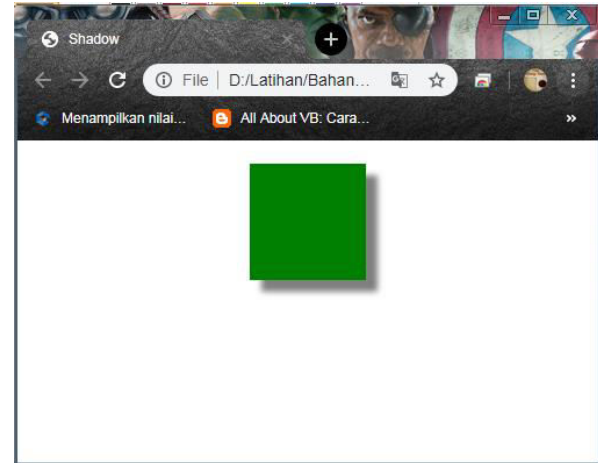
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Shadow</title>
  <style>
    div{
      width: 100px;
      height: 100px;
      background: green;
      margin: 20px auto;
      box-shadow: 10px 10px grey;
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



Box shadow – dengan blur

Nilai blur pada property box-shadow tidak diwajibkan, disini adalah contoh penggunaan Box-shadow dengan blur :

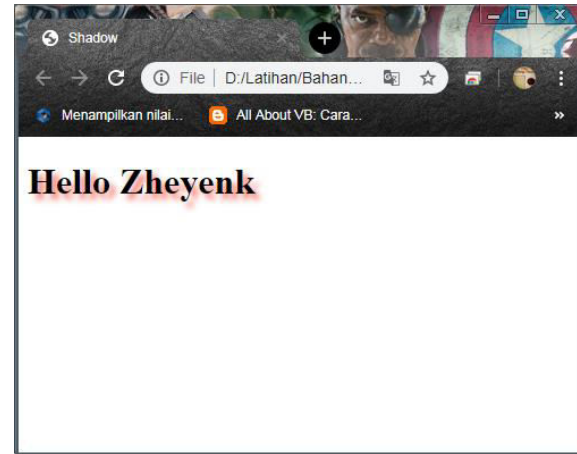
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Shadow</title>
  <style>
    div{
      width: 100px;
      height: 100px;
      background: green;
      margin: 20px auto;
      box-shadow: 10px 10px 5px grey;
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



Text shadow – dengan spread

Nilai blur pada property text-shadow tidak diwajibkan, disini adalah contoh penggunaan text shadow dengan blur :

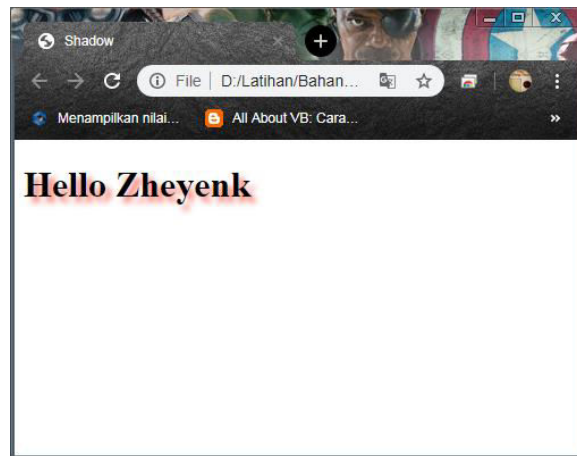
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Shadow</title>
  <style>
    h1{
      text-shadow: 5px 5px 5px salmon;
    }
  </style>
</head>
<body>
  <h1>Hello Zheyenk</h1>
</body>
</html>
```



Text shadow – inset

Nilai blur pada property text-shadow tidak diwajibkan, disini adalah contoh penggunaan text shadow dengan blur :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Shadow</title>
  <style>
    h1{
      text-shadow: 5px 5px 5px salmon;
    }
  </style>
</head>
<body>
  <h1>Hello Zheyenk</h1>
</body>
</html>
```



CSS Transform 2D

Transformasi CSS memungkinkan Anda untuk memindahkan, memutar, skala, dan elemen miring.

Dengan transformproperty CSS Anda dapat menggunakan metode transformasi 2D berikut:

- `translate()`
- `rotate()`
- `scale()`
- `scaleX()`
- `scaleY()`
- `skew()`
- `skewX()`
- `skewY()`
- `matrix()`

Transform – translate()

Metode `translate()` bergerak elemen dari posisi saat ini (sesuai dengan parameter yang diberikan untuk sumbu X dan sumbu Y).

Sintaksis

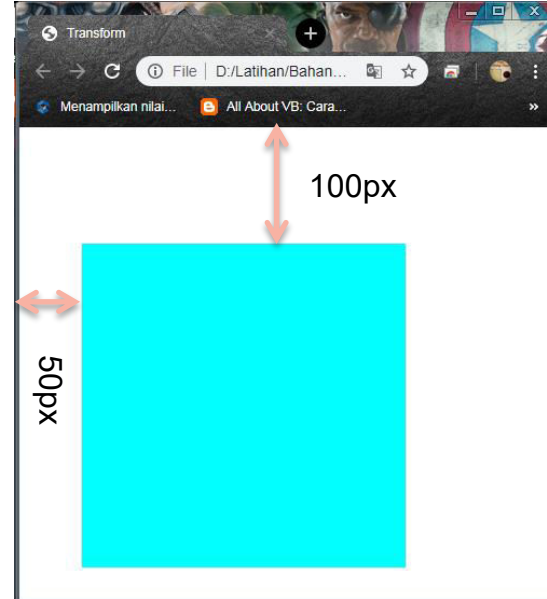
```
transform: translate(50px, 100px);
```

Sumbu x

Sumbu y

Transform – translate()

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Transform</title>
  <style>
    div{
      width: 300px;
      height: 300px;
      background: aqua;
      transform: translate(50px,100px);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



Transform – rotate()

Metode rotate() berputar searah jarum jam merupakan unsur atau berlawanan arah jarum jam sesuai dengan tingkat tertentu.

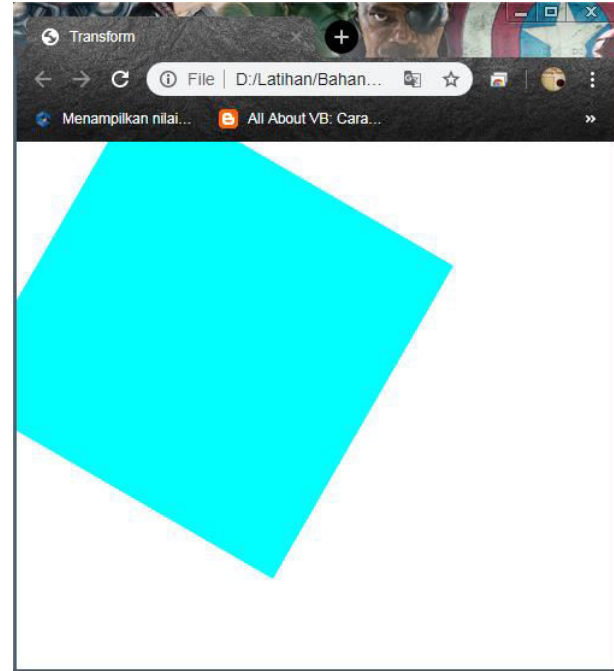
Menggunakan nilai negatif akan memutar elemen berlawanan arah jarum jam.

Sintaksis

```
transform: rotate(30deg);
```


Transform – rotate ()

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Transform</title>
  <style>
    div{
      width: 300px;
      height: 300px;
      background: aqua;
      transform: rotate(30deg);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



Transform – scale()

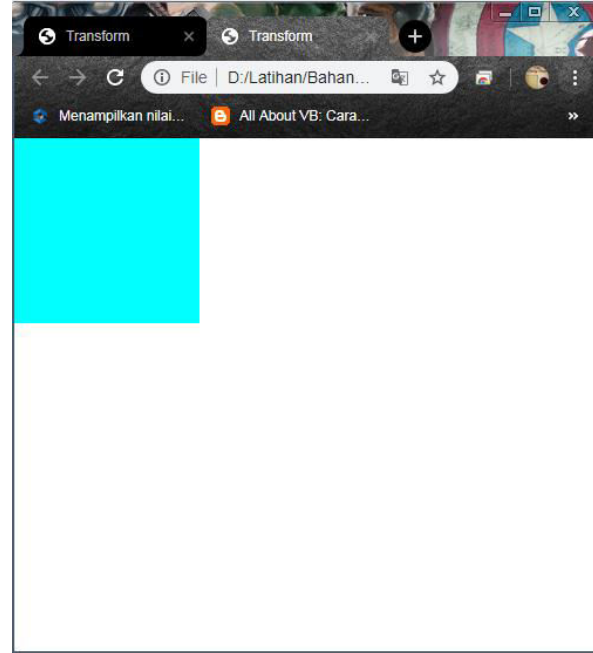
Metode scale() meningkatkan atau mengurangi ukuran elemen (sesuai dengan parameter yang diberikan untuk lebar dan tinggi).

Sintaksis

```
transform: scale(lebar,tinggi);
```

Transform – scale()

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Transform</title>
  <style>
    div{
      width: 100px;
      height: 100px;
      background: aqua;
      transform: scale(2,2);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



Transform – skew()

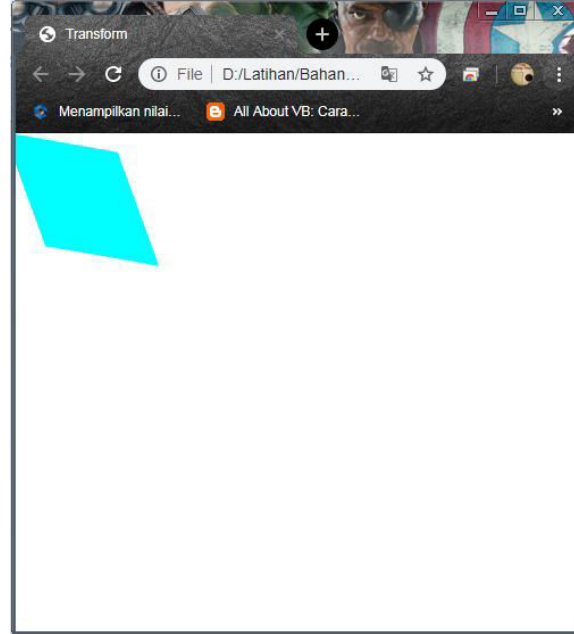
Metode skew() mencondonksn elemen sepanjang X dan Y dengan sudut tertentu.

Sintaksis

```
transform: skew(20deg, 10deg);
```

Transform – skew()

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Transform</title>
  <style>
    div{
      width: 100px;
      height: 100px;
      background: aqua;
      transform: skew(20deg, 10deg);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



Transform – matrix()

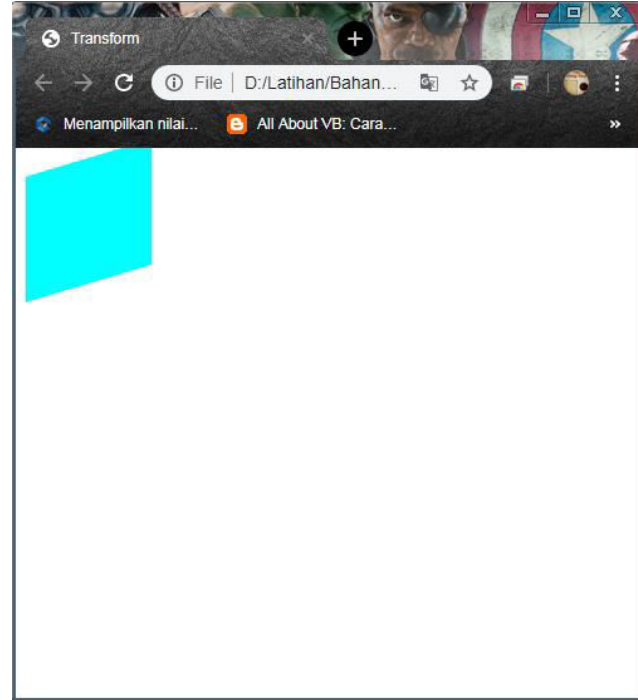
Metode matrix() menggabungkan semua 2D transformasi metode menjadi satu. Metode matrix () mengambil enam parameter, yang berisi fungsi matematika, yang memungkinkan anda untuk memutar, skala, memindahkan (menerjemahkan), dan elemen miring.

Sintaksis

```
transform: matrix(scaleX (), skewY (), skewX (), scaleY (), translateX (), translateY ());
```

Transform – matrix()

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Transform</title>
  <style>
    div{
      width: 100px;
      height: 100px;
      background: aqua;
      transform: matrix(1, -0.3, 0, 1, 0, 0);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```



CSS Transform 3D

Transformasi CSS memungkinkan anda untuk memindahkan, memutar, skala, dan elemen Miring secara 3D.

Dengan transform properti CSS Anda dapat menggunakan metode transformasi 3D berikut:

- rotateX()
- rotateY()
- rotateZ()

Transform – rotateX()

Metode rotateX() memutar elemen di sekitar sumbu X-nya di tingkat tertentu:

Sintaksis

```
transform: rotateX(180deg);
```

Transform – rotateX()

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Transform</title>
  <style>
    div{
      width: 100px;
      height: 100px;
      background: aqua;
      transition: 1s;
    }
    div:hover{
      transform: rotateX(180deg);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```

Script diatas akan memutar element dengan sumbu X sebanyak 180deg saat mouse berada diatas element (hover).

Transform – rotateY()

Metode rotateY() memutar elemen di sekitar sumbu Y-nya di tingkat tertentu:

Sintaksis

```
transform: rotateY(180deg);
```

Transform – rotateY()

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Transform</title>
  <style>
    div{
      width: 100px;
      height: 100px;
      background: aqua;
      transition: 1s;
    }
    div:hover{
      transform: rotateY(180deg);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```

Script diatas akan memutar element dengan sumbu Y sebanyak 180deg saat mouse berada diatas element (hover).

Transform – rotateZ()

Metode rotateZ() memutar elemen di sekitar sumbu Z-nya di tingkat tertentu:

Sintaksis

```
transform: rotateZ(180deg);
```

Transform – rotateZ()

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Transform</title>
  <style>
    div{
      width: 100px;
      height: 100px;
      background: aqua;
      transition: 1s;
    }
    div:hover{
      transform: rotateZ(180deg);
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```

Script diatas akan memutar element dengan sumbu Z sebanyak 180deg saat mouse berada diatas element (hover).

Prespektif

Properti perspective digunakan untuk memberikan elemen 3D strategis beberapa perspektif. Properti perspective mendefinisikan seberapa jauh benda tersebut menjauh dari pengguna. Jadi, nilai yang lebih rendah akan menghasilkan efek 3D yang lebih intensif daripada nilai yang lebih tinggi.

Saat mendefinisikan perspective properti untuk suatu elemen, elemen CHILD-lah yang mendapatkan tampilan perspektif, BUKAN elemen itu sendiri.

Sintaks

```
perspective:none | length;
```

Prespektif

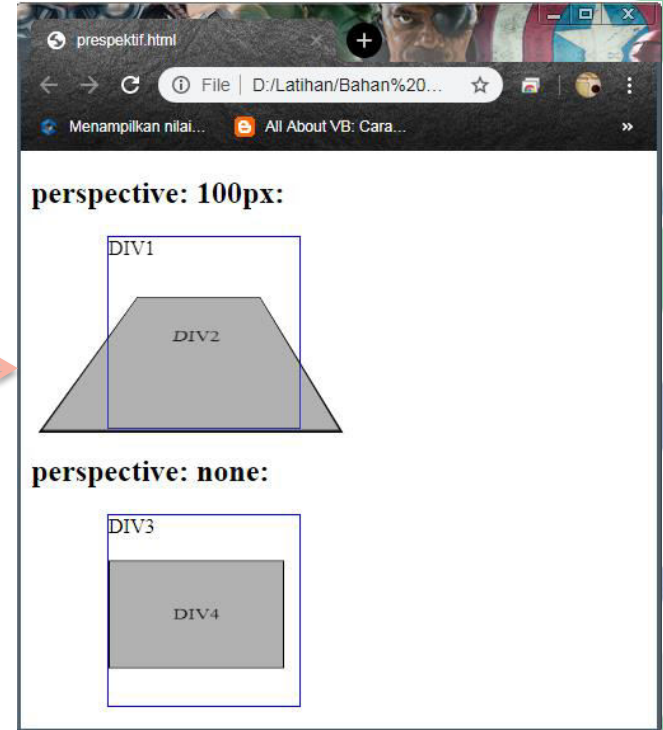
```
<h2>perspective: 100px:</h2>
<div id="div1">DIV1
  <div id="div2">DIV2</div>
</div>

<h2>perspective: none:</h2>
<div id="div3">DIV3
  <div id="div4">DIV4</div>
</div>
```

```
#div1 {
  position: relative;
  height: 150px;
  width: 150px;
  margin-left: 60px;
  border: 1px solid blue;
  perspective: 100px;
  perspective: none | length;
}

#div2, #div4 {
  padding: 50px;
  position: absolute;
  border: 1px solid black;
  background-color: red;
  background: rgba(100,100,100,0.5);
  transform-style: preserve-3d;
  transform: rotateX(45deg);
}

#div3 {
  position: relative;
  height: 150px;
  width: 150px;
  margin-left: 60px;
  border: 1px solid blue;
  perspective: none;
}
```



Backface-visibility

Properti backface-visibility mendefinisikan apakah bagian belakang elemen harus terlihat atau tidak.

Bagian belakang elemen adalah gambar cermin dari bagian depan yang ditampilkan. Properti ini berguna ketika suatu elemen diputar. Ini memungkinkan Anda memilih apakah pengguna harus melihat bagian belakang atau tidak.

Sintaks

```
backface-visibility: hidden | visible;
```

Backface-visibility

Properti backface-visibility mendefinisikan apakah bagian belakang elemen harus terlihat atau tidak.

Bagian belakang elemen adalah gambar cermin dari bagian depan yang ditampilkan. Properti ini berguna ketika suatu elemen diputar. Ini memungkinkan Anda memilih apakah pengguna harus melihat bagian belakang atau tidak.

Sintaks

```
backface-visibility: hidden | visible;
```

Backface-visibility

Apa perbedaan
antara div2
dengan div1?

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  position: relative;
  height: 60px;
  width: 60px;
  background-color: red;
  transition: 2s;
}
#div1:hover {
  backface-visibility: hidden;
  transform: rotateY(180deg);
}
#div2:hover {
  backface-visibility: visible;
  transform: rotateY(180deg);
  backface-visibility: hidden | visible;
}
</style>
</head>
<body>
  <p>"backface-visibility: hidden"</p>
  <div id="div1">DIV 1</div>

  <p>"backface-visibility: visible"</p>
  <div id="div2">DIV 2</div>
</body>
</html>
```

Transition

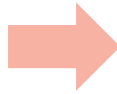
Transisi CSS memungkinkan Anda mengubah nilai properti dengan lancar, selama durasi yang diberikan.

Properti pada transition:

- transition
- transition-delay
- transition-duration
- transition-property
- transition-timing-function

Transition-duration

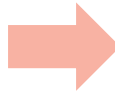
```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  transition-duration: 2s;
}
div:hover {
  width: 300px;
}
</style>
</head>
<body>
  <div></div>
</body>
</html>
```



Lebar div akan mencapai 300px
dalam waktu 3 detik
Saat dihover.

Transition-delay

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  transition-delay: 2s;
}
div:hover {
  width: 300px;
}
</style>
</head>
<body>
  <div></div>
</body>
</html>
```



Lebar div akan mencapai 300px
Setelah menunggu selama 3 detik
Saat dihover.

Transition-timing-function

Properti transition-timing-function menentukan kurva kecepatan efek transisi.

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

- **ease**

=> menentukan efek transisi dengan awal yang lambat, lalu cepat, lalu akhiri perlahan (ini default)

- **linear**

=> Menentukan efek transisi dengan kecepatan yang sama dari awal hingga akhir

- **ease-in**

=> Menentukan efek transisi dengan awal yang lambat

- **ease-out**

=> Menentukan efek transisi dengan ujung yang lambat

- **ease-in-out**

=> Menentukan efek transisi dengan awal dan akhir yang lambat

- **cubic-bezier(n,n,n,n)**

=> memungkinkan Anda menentukan nilai Anda sendiri dalam fungsi cubic-bezier

Transition-timing-function

```
<!DOCTYPE html>
<html>
<head>
<style>
  div {
    width: 100px;
    height: 100px;
    background: red;
    transition-duration: 2s;
  }
  div:hover {
    width: 300px;
  }
  #div1 {transition-timing-function: linear;}
  #div2 {transition-timing-function: ease;}
  #div3 {transition-timing-function: ease-in;}
  #div4 {transition-timing-function: ease-out;}
  #div5 {transition-timing-function: ease-in-out;}
</style>
</head>
<body>
  <h1>Property transition-timing-function</h1>
  <div id="div1">linear</div><br>
  <div id="div2">ease</div><br>
  <div id="div3">ease-in</div><br>
  <div id="div4">ease-out</div><br>
  <div id="div5">ease-in-out</div><br>
</body>
</html>
```


Studi kasus 1



Folder Img / Pertemuan8 (Video#7)

Studi kasus 2



Folder Img / Pertemuan8 (Video#8)

Referensi

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



**Terima
kasih**