MINGGU 9 (PEMROGRAMAN BERBASIS WEB)

CSS 3

CSS 3 adalah kembangan baru dari CSS. untuk mendukung perkembangan terkait dengan teknologi web yang semakin pesat. pengertian dasar nya masih sama dengan yang dijelaskan pada pertemuan sebelumnya, sebelum css lahir di dunia pemrogaman web, para developer masih menggunakan table untuk membuat layout website. CSS 3 merupakan perkembangan CSS generasi ke 3.
CSS 3 sudah mendukung banyak efek yang bisa langsung di gunakan dengan cepat. misalnya seperti membuat efek tulisan text, gambar atau objek. Dengan css 3 dapat membuat gambar gradient, lebih interaktif di gunakan pada template website, dan lebih hemat bandwidth karena sangat ringan. dapat membuat shadow pada element html, dan masih banyak lagi kelebihan dari.

Ada beberapa perkembangan baru dalam css 3, yaitu transform, transition dan animation. dengan menggunakan transforms css3 memungkinkan untuk membuat translate, rotate(memutar), scale(mengatur skala element html) dan skew, dengan transition css3 memungkinkan untuk mengatur tiap transisi element seperti membuat durasi dan lain nya. dan yang ketiga adalah animation, css3 juga telah menyiapka beberapa perintah yang memungkinkan untuk membuat efek animasi pada element html tanpa harus menggunakan javascript dan flash.

Latihan CSS 3

Sudut Melengkung dengan CSS 3

Dengan CSS3 sudah memungkinkan untuk membuat sudut melengkung pada sebuah element HTML. jadi tampilannya pun sudah lebih interaktif. untuk membuat sudut element menjadi melengkung dengan css3 ini bisa menggunakan perintah **"border-radius"**.
Praktikum 1 : simpan dengan file ***css3-1.html***

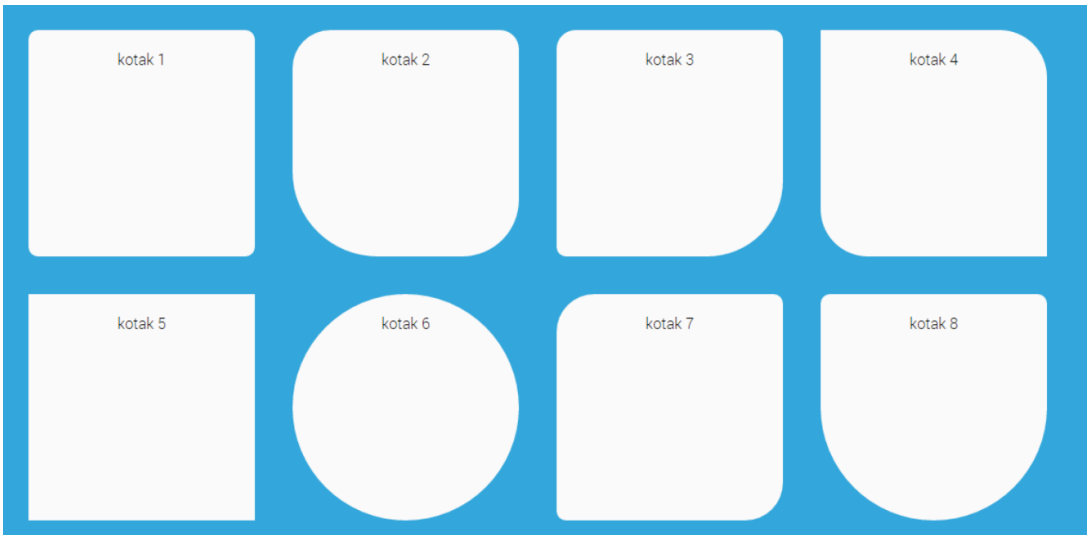
```
<!DOCTYPE html>
<html>
<head>
	<title>Membuat sudut melengkung dengan css3</title>
	<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
	<h1>Membuat sudut melengkung dengan css3 </h1>

	<div class="ketengah">
		<div class="kotak kotak1">kotak 1</div>
		<div class="kotak kotak2">kotak 2</div>
		<div class="kotak kotak3">kotak 3</div>
		<div class="kotak kotak4">kotak 4</div>
		<div class="kotak kotak5">kotak 5</div>
		<div class="kotak kotak6">kotak 6</div>
		<div class="kotak kotak7">kotak 7</div>
		<div class="kotak kotak8">kotak 8</div>
	</div>
</body>
</html>
```

File CSS: CSS1.CSS

```
body{
	background: #35A9DB;
	font-family: roboto;
	text-align: center;
}

h1{
	color: #fff;
}

.ketengah{
	margin: 10px auto;
	width: 1150px;
}

.kotak{
	background: #fcfcfc;
	padding: 20px;
	width: 200px;
	float: left;
	margin: 20px;
	height: 200px;
}

.kotak1{
	border-radius: 10px 10px 10px 10px;
}


.kotak2{
	border-radius: 40px 20px 60px 90px;
```

```
}

.kotak3{
      border-radius: 2px 140px 20px 60px;
}

.kotak3{
      border-radius: 20px 10px 80px 10px;
}

.kotak4{
      border-radius: 0px 50px 0px 50px;
}

.kotak5{
      border-radius: 0px 0px 0px 0px;
}

.kotak6{
      border-radius: 100%;
}

.kotak7{
      border-radius: 40px 10px;
}

.kotak8{
      border-radius: 10px 10px 50% 50%;
}
```

Hasil :



Secara umum struktur syntax untuk border radius sebagai berikut :

```
border-radius : nilai_sudut_kiri_atas nilai_sudut_kanan_atas
nilai_sudut_kanan_bawah nilai_sudut_kiri_bawah;
```

Warna Gradient CSS3

Ada 2 tipe warna gradient atau gradasi yang dapat di buat dengan css3. yaitu ***linear gradient*** dan ***radial gradient***. linear gradient adalah gradient yang berbentuk warna yang berdampingan tapi efeknya sangat lembut. jadi kedua warna akan terlihat menyatu. baik itu atas ke bawah atau kiri ke kanan. sedangkan radial gradient adalah warna gradient yang memiliki pusat tengah. jadi seperti membentuk lingkaran dalam pewarnaannya.

Praktikum 2: buat sebuah file bernama ***css3-2.html***

```
<html>
<head>
	<title>Warna gradient dengan css3</title>
	<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
	<h1>Warna gradient dengan css3</h1>

	<div class="ketengah">
		<!-- linear gradient -->
		<div class="kotak gradient1"></div>

		<div class="kotak gradient2"></div>

		<!-- radial gradient -->
		<div class="kotak gradient3"></div>
```

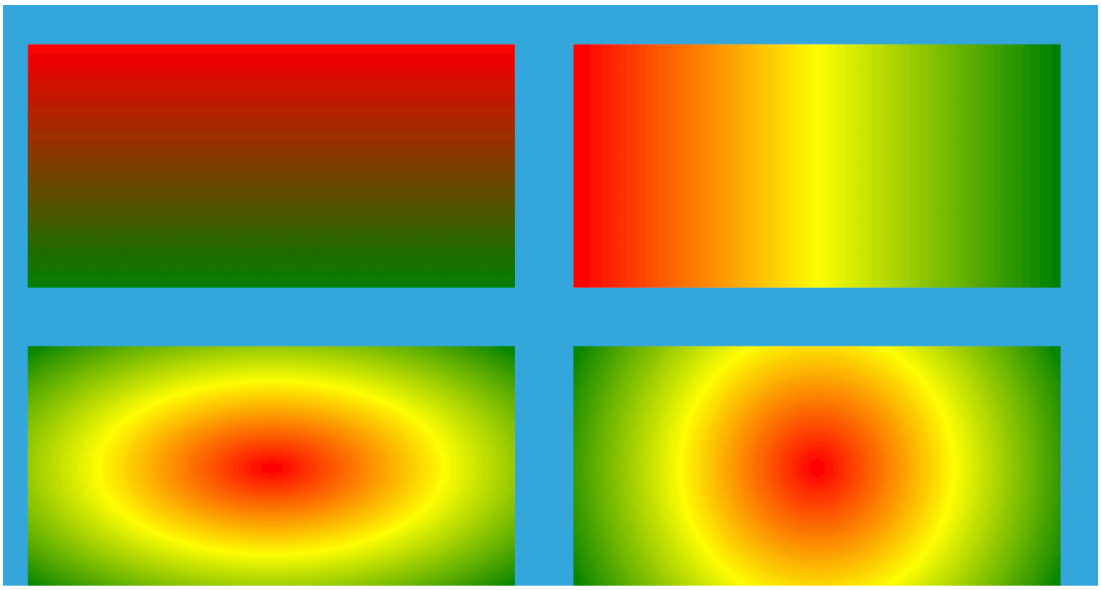
```
            <div class="kotak gradient4"></div>

      </div>
</body>
</html>
```

File css : css2.css

```
body{
      background: #35A9DB;
      font-family: roboto;
      text-align: center;
}

h1{
      color: #fff;
}

.ketengah{
      margin: 10px auto;
      width: 1150px;
}

.kotak{
      background: blue;
      width: 500px;
      float: left;
      height: 250px;
      margin: 30px;
}

.gradient1 {
      background: linear-gradient(red, green); /* Standard syntax */
      background: -webkit-linear-gradient(red, green); /* For Safari 5.1 to
6.0 */
      background: -o-linear-gradient(red, green); /* For Opera 11.1 to 12.0
*/
      background: -moz-linear-gradient(red, green); /* For Firefox 3.6 to 15
*/
}

.gradient2 {
      background: linear-gradient(to right, red , yellow, green); /* Standard
syntax */
      background: -webkit-linear-gradient(left, red , yellow, green); /* For
Safari 5.1 to 6.0 */
      background: -o-linear-gradient(right, red, yellow, green); /* For Opera
11.1 to 12.0 */
      background: -moz-linear-gradient(right, red, yellow, green); /* For
Firefox 3.6 to 15 */
}

.gradient3 {
       background: radial-gradient(red, yellow, green); /* Standard syntax */
```

```
        background: -webkit-radial-gradient(red, yellow, green); /* Safari 5.1
to 6.0 */
        background: -o-radial-gradient(red, yellow, green); /* For Opera 11.6
to 12.0 */
        background: -moz-radial-gradient(red, yellow, green); /* For Firefox
3.6 to 15 */
}

.gradient4 {
        background: radial-gradient(circle, red, yellow, green); /* Standard
syntax */
        background: -webkit-radial-gradient(circle, red, yellow, green); /*
Safari */
        background: -o-radial-gradient(circle, red, yellow, green); /* Opera
11.6 to 12.0 */
        background: -moz-radial-gradient(circle, red, yellow, green); /*
Firefox 3.6 to 15 */
}
```

Hasil :



Efek Bayangan dengan CSS3 (Box Shadow)

untuk membuat efek bayangan pada sebuah element html dapat menggunakan perintah ***box-shadow***.
Syntax sebagai berikut:

```
box-shadow:rata_atas_bawah rata_kiri_kanan jarak_bayangan ketebalan_bayangan
warna;
```

text-shadow : untuk membuat bayangan pada text
box-shadow : untuk membuat bayangan pada element html selain text.

Praktikum 3 : simpan dengan file ***css3-3.html***

```
<!DOCTYPE html>
<html>
<head>
	<title>Membuat Efek Bayangan Dengan CSS3</title>
	<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
	<h1>Membuat Efek Bayangan Dengan CSS3</h1>

	<div class="ketengah">

		<h1 class="tulisan1">Belajar CSS 3</h1>

		<h1 class="tulisan2">Belajar CSS 3</h1>

		<div class="kotak1">kotak 1</div>

		<div class="kotak2">kotak 2</div>

	</div>
</body>
</html>
```

File css : css3.css

```
body{
	background: #35A9DB;
	font-family: sans-serif;
	text-align: center;
}

h1{
	color: #fff;
}

.ketengah{
	margin: 10px auto;
	width: 1150px;
}


/*
nilai 20 adalah nilai tinggi rendah nya bayangan
nilai 10 adalah nilai untuk mengatur rata kiri kanan bayangan
nilai 5 adalah nilai untuk mengatur ketebalan bayangan
dan yang terakhir adalah mengatur warna bayangan
*/
.tulisan1{
	text-shadow: 20px 10px 5px #232323;
}

/*
```

```
nilai 0 pertama adalah nilai tinggi rendah nya bayangan
nilai 0 kedua adalah nilai untuk mengatur rata kiri kanan bayangan
nilai 20 adalah nilai untuk mengatur ketebalan bayangan
dan yang terakhir adalah mengatur warna bayangan
*/
.tulisan2{
      text-shadow: 0px 0px 20px #232323;
}



/*
nilai 20 yang pertama adalah nilai tinggi rendah nya bayangan
nilai 20 kedua adalah nilai untuk rata kiri kanan bayagan
nilai 40 adalah nilai radius banyangan
nilai 10 adalah nilai ketebalan bayangan
dan yang terakhir adalah menentukan warna bayangan
*/
.kotak1{
      background: white;
      width: 300px;
      height: 300px;
      margin: 20px;
      float: left;
      box-shadow: 20px 20px 40px 10px red;
}


/*
nilai 0 yang pertama adalah nilai tinggi rendah nya bayangan
nilai 0 kedua adalah nilai untuk rata kiri kanan bayagan
nilai 10 pertama adalah nilai radius banyangan
nilai 10 kedua adalah nilai ketebalan bayangan
dan yang terakhir adalah menentukan warna bayangan
*/
.kotak2{
      background: white;
      width: 300px;
      margin: 20px;
      float: left;
      height: 300px;
      box-shadow: 0px 0px 10px 10px #333333;
}
```

CSS3 Transform

Css transform biasa digunakan sebagai dasar untuk membuat efek memutar element, membuat element bergerak dan sebagainya. Css transform bisa dipelajari terlebih dahulu sebelum belajar tentang css transition dan css animation.

Metode Transform css :

- **translate()**

translate() adalah fungsi yang bisa di gunakan untuk memindahkan sebuah element, bisa menentukan posisi nya dengan memberikan niai pada parameter x untuk jarak samping dan parameter y untuk jarak tingginya.

- **scale()**
  fungsi method scale() ini di gunakan untuk melebarkan atau menjauhkan element dengan menambah atau mengurangi ukuran element. seperi effect zoom lah kira-kiranya.
- **skewX()**
  Skew() di gunakan untuk memiringkan element dengan mengatur tiap nilai pada sumbu x nya.
- **skewY()**
  Skew() di gunakan untuk memiringkan element dengan mengatur tiap nilai pada sumbu y nya.
- **rotate()**
  rotate() adalah fungsi yang bisa digunakan untuk memutar sebuah element. untuk memutar elemenya bisa menentukan nilainya di dalam parameter fungsi rotate ini. untuk nilainya sendiri bisa ditentukan dengan cara memberikan nilai derajat putar nya.
- **matrix()**
  fungsi matrix berguna untuk menggabungkan beberapa transform lainnya seperti menggabungkan rotate(), skew() scale() dan lainnya menjadi satu. sehingga transform tersebut akan di lakukan secara sekaligus.

Translate()

Syntax:

```
translate(nilai_jarak_samping, nilai_jarak_atas)
```

Praktikum 4 : simpan dengan file ***css3-translate.html***

```
<!DOCTYPE html>
<html>
<head>
      <title>Tutorial CSS Transform</title>
      <link rel="stylesheet" type="text/css" href="style.css">
</head>
```

```
<body>
      <h1>Tutorial CSS Transform</h1>
      <h2>translate()</h2>
      <div class="kotak1">
            Ini adalah kotak 1. kotak ini tidak di berikan method translate
      </div>
      <div class="kotak2">
            Ini adalah kotak 2. ini contoh fungsi translate() CSS Transform.
      </div>
</body>
</html>
```

File css : css4.css

```
body{
      background: #35A9DB;
      font-family: roboto;
      text-align: center;
}

h1,h2{
      color: #fff;
}

.ketengah{
      margin: 10px auto;
      width: 1150px;
}

.kotak1{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 20px;
      width: 150px;
      height: 150px;
}

.kotak2{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 20px;
      width: 150px;
      height: 150px;
      -ms-transform: translate(50px,100px); /* Support IE 9 */
    -webkit-transform: translate(50px,100px); /* support Safari */
    transform: translate(50px,100px); /* Standard syntax */
}
```

scale()

Praktikum 5 : simpan dengan file ***css3-scale.html***

```
<!DOCTYPE html>
<html>
<head>
      <title>Tutorial CSS Transform</title>
      <link rel="stylesheet" type="text/css" href="scale.css">
</head>
<body>
      <h1>Tutorial CSS Transform</h1>
      <h2>scale()</h2>
      <div class="kotak1">
```

```
            Ini adalah kotak 1. kotak ini tidak di berikan method scale
      </div>
      <div class="kotak2">
            Ini adalah kotak 2. ini contoh fungsi scale() CSS Transform.
      </div>
</body>
</html>
```

File css : scale.css

```
body{
      background: #35A9DB;
      font-family: roboto;
      text-align: center;
}

h1,h2{
      color: #fff;
}

.ketengah{
      margin: 10px auto;
      width: 1150px;
}

.kotak1{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 50px;
      width: 150px;
      height: 150px;
}

.kotak2{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 50px;
      width: 150px;
      height: 150px;
      -ms-transform: scale(2,3); /* Support IE 9 */
    -webkit-transform: scale(2,3); /* support Safari */
    transform: scale(2,2); /* Standard syntax */
}
```

skew()

Praktikum 6 : simpan dengan file ***css3-skew.html***

```
<!DOCTYPE html>
<html>
<head>
	<title>Tutorial CSS Transform</title>
	<link rel="stylesheet" type="text/css" href="skew.css">
</head>
<body>
	<h1>Tutorial CSS Transform</h1>
	<h2>skew()</h2>
	<div class="kotak1">
		Ini adalah kotak 1. kotak ini tidak di berikan method skew
	</div>
	<div class="kotak2">
		Ini adalah kotak 2. ini contoh fungsi skew() CSS Transform.
skew().
	</div>
	<div class="kotak3">
		Ini adalah kotak 2. ini contoh fungsi skew() CSS Transform.
skew() 2 parameter.
	</div>
	<div class="kotak4">
		Ini adalah kotak 2. ini contoh fungsi skew() CSS Transform.
skewY().
	</div>
	<div class="kotak5">
		Ini adalah kotak 2. ini contoh fungsi skew() CSS Transform.
skewX().
	</div>

</body>
</html>
```

File css : skew.css

```
body{
	background: #35A9DB;
	font-family: roboto;
	text-align: center;
}

h1,h2{
	color: #fff;
}

.ketengah{
	margin: 10px auto;
	width: 1150px;
}
```

```
.kotak1{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 50px;
      width: 150px;
      height: 150px;
}

.kotak2{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 50px;
      width: 150px;
      height: 150px;
      -ms-transform: skew(20deg); /* Support IE 9 */
    -webkit-transform: skew(20deg); /* support Safari */
    transform: skew(20deg); /* Standard syntax */
}

.kotak3{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 50px;
      width: 150px;
      height: 150px;
      -ms-transform: skew(30deg,20deg); /* Support IE 9 */
    -webkit-transform: skew(30deg,20deg); /* support Safari */
    transform: skew(30deg,20deg); /* Standard syntax */
}

.kotak4{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 50px;
      width: 150px;
      height: 150px;
      -ms-transform: skewY(20deg); /* Support IE 9 */
    -webkit-transform: skewY(20deg); /* support Safari */
    transform: skewY(20deg); /* Standard syntax */
}

.kotak5{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 50px;
      width: 150px;
      height: 150px;
      -ms-transform: skewX(20deg); /* Support IE 9 */
    -webkit-transform: skewX(20deg); /* support Safari */
    transform: skewX(20deg); /* Standard syntax */
}
```

Praktikum 7 : simpan dengan file ***css3-rotate.html***

```
<!DOCTYPE html>
<html>
<head>
	<title>Tutorial CSS Transform</title>
	<link rel="stylesheet" type="text/css" href="rotate.css">
</head>
<body>
	<h1>Tutorial CSS Transform</h1>
	<h2>rotate()</h2>
	<div class="kotak1">
		Ini adalah kotak 1. kotak ini tidak di berikan method rotate
	</div>
	<div class="kotak2">
		Ini adalah kotak 2. ini contoh fungsi rotate() CSS Transform.
rotate(30deg).
	</div>
	<div class="kotak3">
		Ini adalah kotak 2. ini contoh fungsi rotate() CSS Transform.
rotate(90deg).
	</div>
	<div class="kotak4">
		Ini adalah kotak 2. ini contoh fungsi rotate() CSS Transform.
rotate(-90deg).
	</div>

</body>
</html>
```

File css : rotate.css

```
body{
	background: #35A9DB;
	font-family: roboto;
	text-align: center;
}

h1,h2{
	color: #fff;
}

.ketengah{
	margin: 10px auto;
	width: 1150px;
}

.kotak1{
	background: #fcfcfc;
	padding: 20px;
	float: left;
	margin: 50px;
	width: 150px;
```

```
      height: 150px;
}

.kotak2{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 50px;
      width: 150px;
      height: 150px;
      -ms-transform: rotate(30deg); /* Support IE 9 */
    -webkit-transform: rotate(30deg); /* support Safari */
    transform: rotate(30deg); /* Standard syntax */
}

.kotak3{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 50px;
      width: 150px;
      height: 150px;
      -ms-transform: rotate(90deg); /* Support IE 9 */
    -webkit-transform: rotate(90deg); /* support Safari */
    transform: rotate(90deg); /* Standard syntax */
}

.kotak4{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 50px;
      width: 150px;
      height: 150px;
      -ms-transform: rotate(-60deg); /* Support IE 9 */
    -webkit-transform: rotate(-60deg); /* support Safari */
    transform: rotate(-60deg); /* Standard syntax */
}
```

matrix()

Praktikum 8 : simpan dengan file ***css3-matrix.html***

```
<!DOCTYPE html>
<html>
<head>
      <title>Tutorial CSS Transform </title>
      <link rel="stylesheet" type="text/css" href="matrix.css">
</head>
<body>
      <h1>Tutorial CSS Transform</h1>
      <h2>matrix()</h2>
      <div class="kotak1">
            Ini adalah kotak 1. kotak ini tidak di berikan method matrix
```

```
      </div>
      <div class="kotak2">
            Ini adalah kotak 2. ini contoh fungsi matrix() CSS Transform.
      </div>

</body>
</html>
```

File css : matrix.css

```
body{
      background: #35A9DB;
      font-family: roboto;
      text-align: center;
}

h1,h2{
      color: #fff;
}

.ketengah{
      margin: 10px auto;
      width: 1150px;
}

.kotak1{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 50px;
      width: 150px;
      height: 150px;
}

.kotak2{
      background: #fcfcfc;
      padding: 20px;
      float: left;
      margin: 50px;
      width: 150px;
      height: 150px;
      matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())
      -ms-transform: matrix(1,2,-1,1,5,4); /* Support IE 9 */
    -webkit-transform: matrix(1,2,-1,1,5,4); /* support Safari */
    transform: matrix(1,2,-1,1,5,4); /* Standard syntax */
}
```

Dengan menggunakan transition atau transisi di css3 ini dapat dimungkinkan untuk memanipulasi property atau value sebuah element dengan sangat lembut. memberikan durasi dan banyak lagi kelebihan dan kemudahan transition css3, Ada beberapa perintah css3 transition, di antaranya adalah :

- **Transition**
  Perintah standar transisi css3, biasanya digunakan untuk memanipulasi property atau nilai value css dengan sangat baik.
- **Transition-delay**
  digunakan untuk memberikan waktu tunda pada saat peng-eksekusian sebuah efek css3
  Transition-duration
  digunakan untuuk memberikan waktu durasi terhadap sebuah efek yang dibuat dengan css3.
- **Transition-property**
  untuk menentukan property css yang akan di gunakan,manipulasi atau lainnya(di pilih).
- **Transition-timing-function**
  untuk menentukan kecepatan sebuah efek yang dibuat dengan css3.

Praktikum 9 : simpan dengan file ***css3-transition.html***

```
<!DOCTYPE html>
<html>
<head>
      <title>Tutorial CSS3 Transition </title>
      <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
      <h1>Tutorial CSS3 Transition</h1>
      <div class="kotak">ini kotak</div>
      <div class="kotak2">ini kotak 2</div>

</body>
</html>
```

File css : transition.css

```
body{
      background: #35A9DB;
      font-family: roboto;
      text-align: center;
}

h1{
      color: #fff;
}

div{
      margin: 10px;
}

.kotak{
      background: #fff;
      width: 100px;
      height: 100px;
      transition: width 1s;
      -webkit-transition: width 1s;
}

.kotak:hover{
      width: 300px;
}

.kotak2{
      background: #fff;
      width: 100px;
      height: 100px;
      transition: width 1s, height 2s;
      -webkit-transition: width 1s, height 2s;
}

.kotak2:hover{
      width: 300px;
      height: 300px;
}
```

CSS3 menyediakan fitur baru, yaitu animasi. Dengan animasi css3 bisa membuat berbagai animasi. tanpa harus menggunakan javascript dan lainnya. sebelum membuat animasi dengan css.

@keyframes adalah fungsi baru dari css3 untuk membuat animasi. ada banyak kelebihan menggunakan css untuk membuat animasi. karena animasi yang di buat tidak berat dan tidak akan memakan bandwidth yang besar.

Praktikum 10 : simpan dengan file ***css3-animation.html***

```
<!DOCTYPE html>
<html>
<head>
	<title>Membuat efek animasi dengan css3</title>
	<link rel="stylesheet" type="text/css" href="contoh.css">
</head>
<body>
	<h1>Membuat efek animasi dengan css3</h1>
	<h2>Contoh penggunaan @keyframe</h2>

	<div class="kotak"></div>
</body>
</html>
```

File css : animation.css

```
body{
	background: #35A9DB;
	font-family: roboto;
	color: #fff;
}

h1{
	width: 550px;
}

.kotak{
	background: blue;
	width: 200px;
	height: 200px;
	animation-name: animasi_kotak;
	animation-duration: 2s;
}

@keyframes animasi_kotak{
	from{ background: blue;}
	to{background: red;}
}
```

Mengenal CSS Pre-Processor: SASS, LESS, dan Stylus

Preprocessor (pra pengolahan) yang merupakan cara lain dalam menuliskan kosakata CSS, penulisan menggunakan bahasa preproccesed (pra olah/mentah) dan kemudian dikonversi langsung kedalam bahasa CSS murni maupun tanpa dikonversi dengan menambahkan mesin side server agar browser mampu mengkonversinya.

Seorang web developer, tidak peduli seberapa baru, dapat dipastikan akan bertemu dan menulis banyak kode CSS (Cascading StyleSheet). Pada implementasinya, penggunaaan preprocessor memiliki beberapa fitur. Dengan memperluas pola CSS dalam bentuk variabel, operator, interpolasi, fungsi, mixin dan masih banyak lagi. Dalam hal ini SASS, LESS dan Stylus adalah yang paling terkenal.

Mengapa CSS Preprocessor?

Sebenarnya CSS sudah sangat baik, tetapi CSS tidak memberikan kebebasan berekspresi dalam penulisannya. Salah satu tujuan dari CSS adalah untuk memberikan kemudahan dan kesederhanaan sehingga siapapun dapat dengan cepat menangkap bahasa yang digunakan dan mempelajarinya. Namun fokus tersebut justru membatasi apa yang lebih dapat dilakukan pengembang CSS yang pada umumnya tidak menyukai keterbatasan. Developer membutuhkan variabel yang lebih luas, dan jika CSS tidak mampu memberikan variabel tersebut, maka developer akan mencari cara untuk membuat varibel yang lebih dekat dengan kebutuhan.

Preprosesor, dengan fitur canggihnya membantu menuliskan kode yang dapat digunakan kembali, dapat dikembangkan dan dapat diperluas lagi pada CSS. Selain itu, juga bermanfaat dalam meningkatkan produktivitas, dan mengurangi jumlah kode yang ditulis dalam sebuah proyek.

Fitur CSS Preprocessor

Seperti bahasa pemrograman pada umumnya, pre-processor memiliki sintaks yang berbeda, tapi mendukung pengkodean CSS murni dan sintaksnya juga seperti CSS murni. SASS dan Stylus memiliki gaya tambahan. Di SASS, bisa

menghilangkan kurung kurawal dan titik koma, sedangkan di Stylus, juga bisa menghilangkan titik dua. Ini adalah opsional di SASS dan Stylus.

Pada contoh di bawah ini, dapat ditemukan versi SASS, LESS dan Stylus dan output CSS.

Variable

Variabel adalah fitur yang paling dinginkan untuk sebuah CSS. Setiap pengembang, ingin mendefinisikan warna dasar dan menggunakannya di seluruh file CSS, sebagai pengganti menulis hex atau diberi nama warna di properti setiap saat. Selain halnya "warna", variabel juga dapat digunakan untuk "width", "font-size", "font-family", "border", dll.

Variabel dalam SASS dimulai dengan $ sign, di LESS @ sign dan no prefix di Stylus. Keduanya dalam SASS dan LESS, nilai diberikan dengan titik dua (:), dan tanda sama dengan (=) pada Stylus.

```
// LESS
@font-size: 16px;


div {
    font-size: @font-size;
}
```

```
// STYLUS
font-size = 16px


div
    font-size font-size
```

```
// CSS
div {
    font-size: 16px;
}
```

Nesting

CSS tidak memiliki hirarki visual saat bekerja dengan *child selector*. Dengan demikian harus menuliskan *selector* dan kombinasinya dalam baris yang

terpisah. Nesting memberikan hirarki visual seperti pada HTML guna meningkatkan keterbacaan. Dalam beberapa kasus, nesting menyebabkan oversize *selector* dan sesuatu seperti "selector train", jadi gunakanlah dengan bijak.

```
// SASS
$link-color: #999;
$link-hover: #229ed3;

ul {
    margin: 0;

    li {
        float: left;
    }

    a {
        color: $link-color;

        &:hover {
            color: $link-hover;
        }
    }
}
```

```
// LESS
@link-color: #999;
@link-hover: #229ed3;

ul {
    margin: 0;

    li {
        float: left;
    }

    a {
        color: @link-color;

        &:hover {
            color: @link-hover;
        }
    }
}
```

```
// STYLUS
link-color = #999
link-hover = #229ed3

ul
    margin 0
    li
        float left
    a
        color link-color
        &:hover
            color link-hover
```

```
// CSS
ul { margin: 0; }
ul li { float: left; }
ul a { color: #999; }
ul a:hover { color: #229ed3; }
```

Mixins

Mixins adalah kumpulan definisi yang dikompilasi sesuai dengan beberapa parameter atau aturan statis. Sehingga dapat dengan mudah menulis gradien latar belakang lintas browser atau CSS *arrows* dll.

```
// SASS
@mixin bordered($width) {
    border: $width solid #ddd;

    &:hover {
        border-color: #999;
    }
}

h1 {
    @include bordered(5px);
}
```

```
// LESS
.bordered (@width) {
    border: @width solid #ddd;

    &:hover {
        border-color: #999;
    }
}

h1 {
    .bordered(5px);
}
```

```
// STYLUS
bordered(w)
    border: n solid #ddd
    &:hover
        border-color: #999

h1
    bordered(5px)
```

```
// CSS
h1 { border: 5px solid #ddd; }
h1:hover { border-color: #999; }
```

Extends

Extends berguna untuk membagi definisi generik dengan *selector* daripada menyalinnya. Semua *selector* yang diperluas dikelompokkan dalam CSS yang dikompilasi. SASS memperluas setiap hal dari *extends selector* yang mencakup *child selectors* dan *inherited properties*. Namun, dalam LESS dapat memilih setiap *extended selector* dengan menambahkan atribut "all" untuk memperluas metode atau hanya dapat memilih *main instance*.

```
// SASS
.block { margin: 10px 5px; }

p {
  @extend .block;
  border: 1px solid #eee;
}

ul, ol {
  @extend .block;
  color: #333;
  text-transform: uppercase;
}
```

```
// LESS
.block { margin: 10px 5px; }

p {
  &:extend(.block);
  border: 1px solid #eee;
}

ul, ol {
  &:extend(.block);
  color: #333;
  text-transform: uppercase;
}
```

```
// STYLUS
.block
    margin 10px 5px


p
    @extend .block
    border 1px solid #eee


ul
ol
    @extend .block
    color #333
    text-transform uppercase
```

```
// CSS
.block, p, ul, ol { margin: 10px 5px; }


p { border: 1px solid #eee; }
ul, ol { color: #333; text-transform: uppercase; }
```

Color Operations

Ketiga pre-processor ini memiliki fungsi untuk dimainkan dengan warna. Developer bisa meringankan warna dasar atau menjenuhkannya, bahkan bisa mencampur dua atau lebih warna yang berbeda. Fitur ini tidak terlalu penting tapi bagus untuk dimiliki.

```
// SASS
saturate($color, $amount)
desaturate($color, $amount)
lighten($color, $amount)
darken($color, $amount)
adjust-hue($color, $amount)
opacify($color, $amount)
transparentize($color, $amount)
mix($color1, $color2[, $amount])
grayscale($color)
complement($color)
```

```
// LESS
saturate(@color, @amount)
desaturate(@color, @amount)
lighten(@color, @amount)
darken(@color, @amount)
fadein(@color, @amount)
fadeout(@color, @amount)
fade(@color, @amount)
spin(@color, @amount)
mix(@color1, @color2, @weight)
grayscale(@color)
contrast(@color)
```

```
// STYLUS
red(color)
green(color)
blue(color)
alpha(color)
dark(color)
light(color)
hue(color)
saturation(color)
lightness(color)
```

If/Else Statements

Dengan ekspresi dan instruksi pengendalian dapat membantu membangun definisi gaya yang serupa sesuai kondisinya atau variabel yang sesuai. SASS dan Stylus mendukung kondisi if/else. Tapi dalam LESS, bisa dengan CSS Guards.

```
// SASS
@if lightness($color) > 30% {
    background-color: black;
}

@else {
    background-color: white;
}
```

```
// LESS
.mixin (@color) when (lightness(@color) > 30%) {
    background-color: black;
}
.mixin (@color) when (lightness(@color) =<; 30%) {
    background-color: white;
}
```

```
// STYLUS
if lightness(color) > 30%
    background-color black
else
    background-color white
```

Loops

Loop berguna saat iterasi melalui array atau menciptakan serangkaian gaya seperti pada lebar grid. Seperti dalam kasus if / else, LESS menggunakan CSS Guards dan rekursif mixin untuk perulangan.

```
// SASS
@for $i from 1px to 3px {
    .border-#{i} {
        border: $i solid blue;
    }
}
```

```
// LESS
.loop(@counter) when (@counter > 0){
    .loop((@counter - 1));

    .border-@{counter} {
        border: 1px * @counter solid blue;
    }
}
```

```
// STYLUS
for num in (1..3)
    .border-{num}
        border 1px * num solid blue
```

Math

Operasi matematika dapat digunakan untuk konversi aritmatika atau unit konversi. SASS dan Stylus mendukung aritmatika antar unit yang berbeda. Selain

matematika sederhana, pre-processor juga memiliki dukungan matematika yang kompleks seperti batas tertinggi, pembulatan, mendapatkan nilai min atau max dalam daftar dll.

```
// SASS

1cm * 1em => 1 cm * em

2in * 3in => 6 in * in

(1cm / 1em) * 4em => 4cm

2in + 3cm + 2pc => 3.514in

3in / 2in => 1.5
```

```
// LESS

1cm * 1em => 1cm * 1em

2in * 3in => 6in

(1cm / 1em) * 4em => 4cm

2in + 3cm + 2pc => 3.514in

3in / 2in => 1.5in
```

```
// STYLUS

1cm * 1em => 1 cm * em

2in * 3in => 6in

(1cm / 1em) * 4em => 4cm

2in + 3cm + 2pc => 5.181in

3in / 2in => 1.5in
```

Imports

Memisahkan kode dalam potongan kecil sangat membantu untuk mengekspresikan deklarasi dan meningkatkan kemampuan atas kontrol basis kode. Anda dapat mengelompokkan potongan kode yang serupa di folder yang serupa dan mengimpornya ke file css utama. Juga dengan pernyataan import.

```
// SASS
@import "library";
@import "mixins/mixin.scss";
@import "reset.css";
```

```
// LESS
@import "library"
@import "mixins/mixin.less"
@import "reset.css"
```

```
// STYLUS
@import "library"
@import "mixins/mixin.styl"
@import "reset.css"
```

Kesimpulan

Ketiga pre-processor CSS yang dijelaskan di sini kurang lebih memiliki fitur serupa. Jadi pilihlah sesuai dengan selera dan mulailah mencoba menggunakannya di proyek berikutnya.

Repositori Assignment #6 Pertemuan 9

Push hasil latihan ke Github (repository webdas) dan kirim urlnya melalui kulino pada blok (Repositori Assignment #6 Pertemuan 9). Untuk susunan folder dan file sebagai berikut:

- Repominggu9 (folder utama)
  - latihanCSS3
    - css3-1.html
    - dan seterusnya sesuai susunan folder Latihan diatas.