MODUL PRAKTIKUM
PABW MINGGU-4

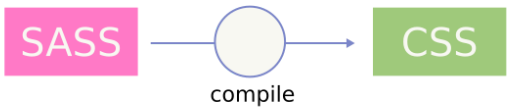
Materi: SASS

SASS (Syntactically Awesome Style Sheets)

I. Penjelasan

SASS merupakan singkatan dari Syntactically Awesome Style Sheets. SASS adalah sebuah bahasa pra-prosesor (preprocessor) untuk CSS. SASS bukanlah bahasa pemrograman, SASS adalah bahasa yang akan membuat CSS. SASS memberikan fitur yang tidak dimiliki CSS seperti variabel, nesting, mixins inheritance, dan lain-lain. Fitur-fitur ini akan membuat kita menulis kode CSS yang lebih rapi dan bersih dan juga menghindari penulisan kode yang berulang.

II. Bagaimana cara kerja SASS ?



SASS seperti bahasa pemrograman, dia harus dikompilasi dulu agar dapat digunakan. Hasil kompilasi SASS akan berupa file CSS. File CSS inilah yang akan digunakan pada web.
Berikut contoh perintah untuk compile SASS menjadi CSS :

```
sass style.scss style.css
```

Kenapa ekstensi nya **.scss** bukan **.sass ?**

Menurut penjelasan thesassway.com, awalnya sintak SASS cukup sulit dipahami karena sangat berbeda dengan CSS.
Pada akhirnya—di versi 3—SASS memperkenalkan struktur **sintak barunya** yaitu SCSS (Sassy CSS) yang lebih mirip dengan CSS dan mudah dipelajari bagi pemula.

III. SASS VS SCSS

Perbandingan sintaks SASS dengan SCSS sebenarnya terletak pada tanda kurung dan tab. SASS menggunakan tab sebagai ganti kurung kurawal dan SASS tidak menggunakan titik koma di akhir properti.

Contoh :

SASS

```
body
      font : 100% $font-stack
      color: $primary-color
```

SCSS

```
body{
      font: 100% $font-stack;
      color: $primary-color;
}
```

IV. Install SASS

Dengan menggunakan Node Package Manager (NPM):
[sudo] npm install -g sass

Macos :
Brew install sass/sass/sass

Untuk windows & linux install nodeJsNPM dulu
https://nodejs.org/en/

Untuk ubuntu jalankan:

```
# Using Ubuntu
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
sudo apt-get install -y nodejs
```

PERCOBAAN

Percobaan 1 Nesting

Buat file berikut ini dengan nama *nesting.scss* :

```
.title {
    border: 4px solid black;
    text-align: center;
    padding: 60px;
    font-family: sans;
    .slogan {
        font-size: 18px;
        color: rgba(0,0,0,0.3);
    }
}
```

Compile code nesting nya, maka struktur folder akan menjadi seperti ini

```
belajar-sass-itera/
 ├──── nesting.css
 ├──── nesting.css.map
 └──── nesting.scss
```

Dan isi dari nesting.css akan menjadi seperti ini :

```
.title {
  border: 4px solid black;
  text-align: center;
  padding: 60px;
  font-family: sans;
}
.title .slogan {
  font-size: 18px;
  color: rgba(0, 0, 0, 0.3);
}
```

Buat file dengan nama *nesting.htnl* :

```
<!DOCTYPE html>
<html>

<head>
    <link rel="stylesheet" href="nesting.css" />
</head>
<body>
    <header>
        <div class="title">
            <h1>Belajar SASS ITERA</h1>
            <span class="slogan">Ez !</span>
        </div>
    </header>

</body>

</html>
```

Hasil nya :


File | /home/mtfiqh/Projects/belajar-sass-itera/nesting.html
Belajar SASS ITERA
Ez !

Percobaan 2 Nested properti :
Nested tidak hanya untuk selektor, bisa juga untuk properti, lakukan percobaan ini agar lebih paham!

nested-property.scss

```
.title {
    border: {
        top: 4px solid black;
        bottom: 4px solid black;
    }
    text-align: center;
    padding: 60px;
    font-family: sans;
    .slogan {
        font-size: 18px;
        color: rgba(0,0,0,0.3);
    }
}
```

Setelah di compile perhatikan file *nested-property.css*, lalu **jelaskan apa yang dimaksud nested property** didalam *nested-property.html*, berikut code untuk *nested-property.html*

```
<!DOCTYPE html>
<html>

<head>
    <link rel="stylesheet" href="nesting.css" />
</head>
<body>
    <header>
        <div class="title">
            <h1>Belajar SASS ITERA</h1>
            <span class="slogan">Ez !</span>
        </div>
    </header>

    <p>
        Penjelasan disini ....
    </p>
</body>

</html>
```

Hasilnya :

Penjelasan disini ....

Percobaan 3 Variabel dalam SASS:

Variable-sass.scss:

```
// Variable
$border-default: 4px solid black;
$backgroun-utama: #008080;
$teks-putih: white;

.title {
    background: $backgroun-utama;
    color: $teks-putih;
    border: {
        top: $border-default;
        bottom: $border-default;
    }
    text-align: center;
    padding: 60px;
    font-family: sans;
    .slogan {
        font-size: 18px;
        color: $teks-putih;
    }
}
```

Setelah di compile, buatlah file dengan nama *variabel-sass.html* dan copy source code dari **percobaan 2,** setelah itu jelaskan apa maksudnya !

Percobaan 4 Partials dan Import pada SASS :

Buatlah folder baru dengan nama ***src/sass*** lalu buat file di dalam folder tersebut dengan nama *_reset.scss* dengan source code berikut ini :

```
html,
body,
ul,
ol {
  margin:  0;
  padding: 0;
}
```

Lalu buat file dengan nama *main.scss* di dalam folder yang sama :

```
@import 'reset';
body {
  font: 100% Helvetica, sans-serif;
  background-color: #efefef;
}
```

Tanda garis bawah (underscore) pada file reset, merupakan tanda bahwa file berikut adalah file *partials*, yang bisa kita import pada file scss lain, pada hal ini kita import pada main.scss dengan syntax **@import**

Pada folder awal (root) buat folder **css**
Lalu compile dengan menjalankan :

```
sass src/sass/main.scss css/main.css
```

Perhatikan folder nya, dan perhatikan file **css** nya

Percobaan 5 Fitur Mixins :

Mixin mirip seperti prosedur pada bahasa pemrograman. Mixin memungkinkan kita menggunakan ulang beberapa grup deklarasi CSS. Pembuatan mixins dapat dilakukan dengan direktif *@mixin* lalu untuk menggunakannya kita pakai direktif *@include*.

Buatlah file *mixins.scss* di dalam folder *src/sass* :

```
// deklarasi mixins
@mixin transform($property) {
  -webkit-transform: $property;
  -ms-transform: $property;
  transform: $property;
}

// menggunakan mixins
.box {
    @include transform(rotate(30deg));
}
```

Lalu, compile pada folder **css** dengan nama ***mixins.css*** !

Maka, hasil css nya setelah mixins.scss di compile adalah :

```
.box {
  -webkit-transform: rotate(30deg);
  -ms-transform: rotate(30deg);
  transform: rotate(30deg);
}

/*# sourceMappingURL=mixins.css.map */
```

Percobaan 6 Extend/Inheritance:
Fitur ini memungkinkan kita untuk berbagi bebereapa properti dengan selektor yang lain.
Kita harus menggunakan direktif @extend untuk mengambil properti dari selektor yang membagikannya.
Oh ya, semua selektor dalam SASS dapat membagikan propertinya secara default.

Buat file di dalam folder *src/sass* dengan nama *inheritance.scss* dengan isinya:

```
%message-shared {
    border: 1px solid #ccc;
    padding: 10px;
    margin: 1rem 0;
    color: #333;
}
// yang ini tidak akan ditulis di dalam CSS karena tidak pernah digunakan dalam @extends
%equal-heights {
    display: flex;
    flex-wrap: wrap;
}

.message {
    @extend %message-shared;
}

.success {
    @extend %message-shared;
    border-color: green;
}

.error {
    @extend %message-shared;
    border-color: red;
}

.warning {
    @extend %message-shared;
    border-color: orange;
}
```

Lalu compile ke dalam folder css lagi, buat file dengan nama *inheritance.html*

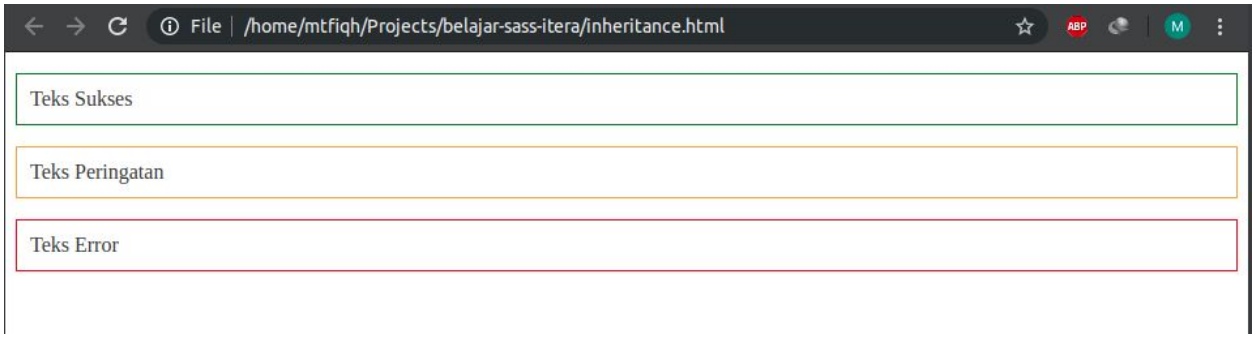
```
<!DOCTYPE html>
<html>

<head>
    <link rel="stylesheet" href="css/inheritance.css" />
</head>
<body>
    <div class="message success">Teks Sukses</div>
    <div class="message warning">Teks Peringatan</div>
    <div class="message error">Teks Error</div>

</body>

</html>
```

Hasil :



Percobaan 7 Operator pada SASS:

Buat file dengan nama *operator.scss* di dalam *src/sass*

```
.container {
  width: 100%;
}

article{
  float: left;
  width: 600px / 960px * 100%;
}

aside{
  float: right;
  width: 300px / 960px * 100%;
}
```

Compile lagi ke dalam file css dengan nama yang sama seperti scss nya !
Lalu amati hasilnya !

Percobaan 8 List / Array :

Buat file di dalam folder yang sama *src/sass* dengan nama list.scss

```
$colors: red yellow blue teal orange;

@each $color in $colors {
  .bg-#{$color} {
    background: $color;
  }
}
```

$colors: red yellow blue teal orange; merupakan array / list yang di deklarasikan dalam scss.
@each merupakan direktif untuk melakukan pengulangan

Compile kedalam folder css lagi dengan nama file yang sama ! amati lagi hasilnya !