

**MODUL PRAKTIKUM
PEMROGRAMAN WEB
PERTEMUAN 2**



ITERA

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO INFORMATIKA DAN SISTEM FISIS
INSTITUT TEKNOLOGI SUMATERA
2020**

Modul 4

CSS Preprocessor

1. Apa Itu CSS Preprocessor

CSS Preprocessor adalah bahasa *script* yang menghasilkan CSS dengan mengkompilasi barisan kode atau sintaks. Ada banyak CSS preprosesor yang dapat dipilih dan digunakan, namun sebagian besar CSS preprosesor akan menambahkan beberapa fitur yang tidak ada di CSS murni, seperti *mixin*, *nesting selector*, *inheritance selector*, dan sebagainya. Fitur-fitur ini membuat struktur CSS lebih mudah dikustomisasi dan dibaca. Untuk menggunakan CSS preprosesor, kita harus menginstal kompiler CSS di web server. Pada praktikum kali ini kita akan menggunakan SASS sebagai CSS Preprosesor.

2. SASS (Syntactically Awesome Style Sheets)

Sass merupakan singkatan dari Syntactically Awesome Style Sheet. Sass adalah salah satu ekstensi CSS Preprosesor yang sering digunakan. Sass sepenuhnya kompatibel dengan semua versi CSS dan efektif untuk mengurangi pengulangan CSS dan karenanya menghemat waktu. Sass sendiri dirancang oleh Hampton Catlin dan dikembangkan oleh Natalie Weizenbaum pada tahun 2006.

2.1. Instalasi SASS

Bagi pengguna Sistem Operasi Windows dan Linux, diperlukan Node Package Manager (NPM) untuk menginstall SASS. Untuk mendapatkan dan menginstal NPM silahkan download melalui :

<https://nodejs.org/en/download/>

Setelah menginstall NPM, silahkan install SASS dengan membuka CLI/terminal dan mengetikkan perintah Gambar 1 berikut lalu menekan Enter

```
npm install -g sass
```

Gambar 1 Instalasi SASS dengan NPM

Instalasi pada MacOS menggunakan perintah Gambar 2 berikut pada terminal dan menekan Enter

```
brew install sass/sass/sass
```

Gambar 2 Instalasi SASS pada MacOS

Untuk memeriksa apakah kita sudah menginstall SASS kita bisa melakukannya dengan mengecek versi SASS yang sudah di install dengan perintah :

```
sass --version
```

Untuk menciptakan sebuah file css dari file sass kita perlu melakukan kompilasi pada sass file dengan cara :

```
sass namafilescss.scss namafilecss.css
```

2.2. Persiapan Praktikum

- Pada direktori repository anda, buatlah folder Modul 4 dan pastikan anda sudah membuat dan menggunakan branch Minggu 2 pada repository local anda.
- Pada folder Modul 4 buatlah file index.html pada Gambar 3 dan Gambar 4 berikut :

```

<!DOCTYPE html>
<html lang="en">
<head>
....<title>Latihan CSS</title>
</head>
<body>
....<div class="navbar">
.....<ul>
.....<li><a href="#">Home</a></li>
.....<li><a href="#">Gallery</a></li>
.....<li><a href="#">Contact</a></li>
.....<li><a href="#">About</a></li>
.....</ul>
....</div>
....<div class="sidebar">
.....<ul>
.....<li><a href="#">HTML</a></li>
.....<li><a href="#">CSS</a></li>
.....<li><a href="#">CSS Preprocessor</a></li>
.....<li><a href="#">PHP</a></li>
.....<li><a href="#">Codeigniter</a></li>
.....<li><a href="#">Laravel</a></li>
.....</ul>
....</div>
....<div class="content">
.....<h1>Praktikum Pemrograman Web</h1>
.....<p>Paragraph Pertama</p>
.....<p>Paragraph Kedua</p>
.....<p>Paragraph Ketiga</p>
....</div>

```

Gambar 3 Baris kode index.html

```

....<div class="footer">
.....<p>
..... 2020&copy;nama lengkap
.....</p>
....</div>
....
</body>
</html>

```

Gambar 4 Lanjutan baris kode pada index.html

2.3.Latihan Variabel pada SASS

Sebuah variabel pada CSS adalah sebuah cara untuk menyimpan informasi yang ingin digunakan kembali pada file CSS. Kita dapat menyimpan hal-hal seperti warna, font, atau nilai CSS apa pun yang mungkin akan digunakan kembali. SASS menggunakan simbol \$ untuk membuat sesuatu menjadi variabel. Salinlah code SASS pada Gambar

5 dan Gambar 6 berikut dengan nama variabel.scss dan compile sehingga menjadi file css dan lihat perubahannya.

```
$mycolour: - ■ rgb(95, 223, 255);
$bgcolour: - □ #333;
$number1: 20px;
$align1: center;

.navbar{
  ....
  ....ul{
  ....    list-style-type: none;
  ....    margin: 0;
  ....    padding: 0;
  ....    overflow: hidden;
  ....    background-color: $bgcolour;
  ....  }
  ....li{
  ....    float: right;
  ....    margin-right: $number1;
  ....  }
}
```

Gambar 5 Baris kode variabel.scss

```
....li a{
....    display: block;
....    color: - ■ white;
....    text-align: $align1;
....    padding: 14px $number1;
....    text-decoration: none;
....  }
....
....li a: hover{
....    background-color: $mycolor;
....  }
}
```

Gambar 6 Lanjutan baris kode pada variabel.scss

Setelah menjadi file css, terapkan file variabel.css pada index.html.

2.4. Latihan Nesting Selector pada SASS

Ketika menulis HTML, kita mungkin pernah menggunakan hierarki bersarang. Namun disisi lain, CSS tidak menerapkan sistem nested, baik pada selector maupun property.

SASS memberikan fitur nested selector pada PHP. Salinlah code SASS pada Gambar 7 dan Gambar 8 berikut dengan nama nested.scss dan compile sehingga menjadi file css dan lihat perubahannya pada file CSS.

```
.sidebar{
  float: left;
  height: auto;
  width: 300px;
  background-color: #f1f1f1;

  ul {
    list-style-type: none;
    padding: 0;
    width: 200px;
    text-align: center;
    background-color: #f1f1f1;
  }
}
```

Gambar 7 Baris kode pada nesting.scss

```
li{
  a {
    display: block;
    color: #000;
    padding: 10px 16px;
    width: 250px;
    text-decoration: none;
  }
}

li{
  a: hover {
    background-color: #555;
    color: white;
  }
}
```

Gambar 8 Lanjutan baris kode pada nesting.scss

Setelah menjadi file css, terapkan file variabel.css pada index.html.

2.5. Latihan Nesting Property pada SASS

Banyak property pada CSS yang memiliki prefix yang sama, seperti font-family, font-size, font-weight atau text-align, text-transform dan text-overflow. Selain pada

selector, nested juga mampu diterapkan pada property SASS. Pada file nested.scss, tambahkan kode pada Gambar 9 berikut hanya pada class sidebar

```
padding: {  
  top: 10px;  
  bottom: 300px;  
}
```

Gambar 9 Tambahan kode pada selector class .sidebar

Sehingga sidebar menjadi seperti Gambar 10 ini,

```
.sidebar {  
  float: left;  
  height: auto;  
  width: 300px;  
  background-color: #f1f1f1;  
  padding: {  
    top: 10px;  
    bottom: 300px;  
  }  
}
```

Gambar 10 Baris kode pada selector .sidebar setelah ditambah beberapa baris kode

Pada selector ul, tambahkan kode pada Gambar 11 berikut :

```
margin: {  
  top: 20px;  
  bottom: 20px;  
  left: 0;  
  right: 0;  
}
```

Gambar 11 Tambahan baris kode pada selector ul

Sehingga pada selector ul pada file nested.scss menjadi seperti Gambar 12 ini :

```
ul {  
  list-style-type: none;  
  padding: 0;  
  width: 200px;  
  text-align: center;  
  background-color: #f1f1f1;  
  margin: {  
    top: 20px;  
    bottom: 20px;  
    left: 0;  
    right: 0;  
  }  
}
```

Gambar 12 Baris kode pada selector ul setelah ditambah beberapa baris kode

Simpan dan amati perubahannya pada file CSS anda.

2.6. Latihan Import dan Partial SASS File

Salah satu cara memodularisasi CSS dan membantu menjaga agar lebih mudah dikelola adalah dengan menggunakan fitur parsial pada scss. File parsial SASS berisi sedikit potongan CSS yang dapat disertakan pada file SASS lainnya. File parsial SASS diberi nama dengan garis bawah di depan. Misalnya `_base.scss`. Garis bawah memberi tahu SASS bahwa file tersebut hanya sebagian file dan tidak boleh dibuat menjadi file CSS. File parsial digunakan dengan sintaks `@import`. Buatlah file `_base.scss` dan salin kode pada Gambar 13 dibawah.

```
//_base.scss
$font-stack: Helvetica, sans-serif;
$primary-color: rgb(248, 65, 65);

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

Gambar 13 Baris kode pada file `_base.scss`

Kemudian buatlah file `modul.scss` dan salin kode pada Gambar 14 dibawah

```
//modul.scss
@import "base";

.content {
  background-color: cornflowerblue;
  color: white;
}
```

Gambar 14 Baris kode pada file `modul.scss`

Compile dan analisa perbedaannya antara `modul.scss` dengan `modul.css`

2.7. Latihan Mixin pada SASS

Mixin pada SASS mirip dengan fungsi pada bahasa pemrograman lain. Mixin memungkinkan kita membuat grup deklarasi CSS yang dapat digunakan kembali di file SASS tersebut. Untuk membuat mixin, digunakan sintaks `@mixin` dilanjutkan dengan nama mixinnya. Dapat pula digunakan variabel `$property` didalam tanda kurung sebagai parameter mixin. Kemudian untuk menggunakannya digunakan sintaks `@include` diikuti dengan nama mixin dan properti (bila ada). Berikut Gambar 15 merupakan contoh penerapan penggunaan mixin. Dan Gambar 16 hasil yang akan terjadi bila Gambar 15 dikompilasi.

```
@mixin transform($property) {  
  padding: $property;  
  background-color: deeppink;  
  color: aliceblue;  
  margin-left: 330px;  
}  
  
.empat {  
  @include transform(20px);  
}
```

Gambar 15 Contoh implementasi fitur `@mixin` pada SASS

```
.empat {  
  padding: 20px;  
  background-color: deeppink;  
  color: aliceblue;  
  margin-left: 330px;  
}  
/*# sourceMappingURL=mixin.css.map */
```

Gambar 16 Hasil kompilasi fitur `@mixin`

2.8. Latihan Extend/Inheritance pada SASS

Berbeda dengan Mixin, Inheritance pada SASS mirip penggunaannya dengan Object pada bahasa pemrograman lain. Extend/Inheritance memungkinkan kita berbagi sekumpulan properti dari sebuah selector dengan selector lain. Inheritance membantu

kita mengurangi penggunaan properti yang sama pada selector yang berbeda. Simpan file berikut pada Gambar 17 dengan nama `inheritance.scss` dan compile lalu terapkan inheritance pada `index.html`. Perhatikan CSS yang sudah di kompilasi.

```
%shared-prop {  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
  
%inheritance-eg {  
  font-size: medium;  
  font-family: Verdana, Geneva, Tahoma, sans-serif;  
  padding: 5px;  
  color: white;  
  border: 10px solid black;  
}  
  
.satu {  
  @extend %shared-prop;  
  margin-left: 330px;  
  background-color: chartreuse;  
}  
  
.dua {  
  @extend %inheritance-eg;  
  margin-left: 330px;  
  background-color: gold;  
}  
  
.tiga {  
  @extend %shared-prop;  
  margin-left: 330px;  
  background-color: red;  
}
```

Gambar 17 Bari kode `inheritance.scss`

2.9. Latihan Operator pada SASS

Pada SCSS kita juga mampu untuk melakukan operasi matematika seperti +, -, *, / dan %. Berikut Gambar 18 merupakan penerapan operator pada SASS.

```
$var: 100;

.content{
  ...margin-left: $var/ 3 * 10px;
}

.footer{
  ...margin-top: $var * 2px;
  ...margin-left: (1px+2px) * $var;
  ...position: relative;
  ...height: auto;
  ...padding: 1px 0;
  ...background-color: salmon;
  ...text-align: center;
  ...p{
    ...text-align: center;
  }
}
```

Gambar 18 Contoh penerapan fitur operator pada SASS

Tugas

Buatlah personal website kalian masing-masing dengan minimal terdapat 4 navbar yaitu Home, Contact, Gallery dan About. Foto pada navbar Gallery dapat diisi gambar random. Personal website tersebut haruslah menerapkan SASS Style.

Modul 5

PHP Dasar

1. Sejarah PHP

PHP adalah singkatan dari “PHP: Hypertext Preprocessor”, yaitu bahasa pemrograman disisi server yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. Sejarah PHP bermula pada tahun 1994 ketika programmer kelahiran Denmark yang sekarang berdomisili di Canada, **Rasmus Lerdorf** membuat sebuah script (kode program) dengan bahasa Perl untuk web pribadinya.

Dengan alasan untuk meningkatkan performa, Rasmus Lerdorf kemudian membuat ulang kode program tersebut dalam bahasa C. Ia juga mengembangkannya lebih lanjut sehingga memiliki script tersebut memiliki kemampuan untuk memproses form HTML dan berkomunikasi dengan database. Lerdorf kemudian merilis kode tersebut ke publik dengan sebutan **Personal Home Page Tools (PHP Tools) version 1.0**.

Seiring dengan pengembangan dan penambahan fitur web pada saat itu, pada April 1996, Rasmus Lerdorf mengumumkan PHP/FI versi 2.0. PHP versi 2 ini dirancang Lerdorf pada saat mengerjakan sebuah proyek di University of Toronto yang membutuhkan pengolahan data dan tampilan web yang rumit. Hingga modul ini dibuat, PHP sudah berkembang hingga PHP versi 7.x.x

2. Variabel

Variabel adalah sebuah wadah untuk menyimpan sebuah informasi(nilai). Variabel pada PHP diawali dengan tanda dollar \$ diikuti dengan nama variabelnya. Contoh penggunaan variabel dapat dilihat pada Gambar 19 .Berbeda dengan bahasa pemrograman lain, pada PHP tidak ada perintah yang mengharuskan pendeklarasian sebuah variabel. Variabel tersebut tercipta ketika langsung di isi dengan nilai. Beberapa aturan pada pemberian nama variabel adalah :

- Nama variabel harus diawali dengan huruf atau underscore (_)
- Nama variabel tidak boleh diawali dengan angka

- Nama variabel hanya boleh mengandung karakter Alpha Numeric
- Berlaku case-sensitive pada variabel PHP.

```

3 //penulisan yang benar
4 $a = 'hello';
5 $number = 100;
6 $nama_saya;
7
8 //penulisan yang salah
9 $nama saya
10

```

Gambar 19 Contoh penerapan variabel pada PHP

3. Operator

Operator merupakan suatu tool atau alat yang disediakan untuk keperluan manipulasi data. Operator dapat dibedakan menjadi :

Operator Aritmatika

Tabel 1 dibawah ini merupakan daftar operator aritmatika pada PHP

Tabel 1 Operator Aritmatika pada PHP

Nama Operasi	Contoh	Keterangan
Penjumlahan	\$a + \$b	Penjumlahan \$a dan \$b
Pengurangan	\$a - \$b	Selisih \$a dan \$b
Perkalian	\$a * \$b	Perkalian \$a dan \$b
Pembagian	\$a / \$b	Pembagian \$a dibagi \$b
Modulo/Modulus	\$a % \$b	Sisa hasil \$a dibagi \$b

Operator Logika

Tabel 2 dibawah ini merupakan daftar operator logika pada PHP

Tabel 2 Operator Logika pada PHP

Nama Operasi	Contoh	Keterangan
AND	\$a and \$b	Bernilai benar jika \$a dan \$b benar

OR	\$a or \$b	Bernilai benar jika \$a atau \$b benar
XOR	\$a xor \$b	Benar jika hanya salah satu \$a atau \$b benar, tidak keduanya
Not True	!\$a	Inverse dari \$a
&&	\$a && \$b	Bernilai benar jika \$a dan \$b benar
	\$a \$b	Bernilai benar jika \$a atau \$b benar

Operator Pembandingan

Tabel 3 dibawah ini merupakan daftar operator pembanding pada PHP

Tabel 3 Operator Pembandingan pada PHP

Nama Operasi	Contoh	Keterangan
Sama Dengan	\$a == \$b	Bernilai benar jika \$a sama dengan \$b
Identik	\$a === \$b	Bernilai benar jika \$a sama dengan \$b, dan memiliki tipe data yang sama (PHP4 keatas)
Tidak Sama Dengan	\$a != \$b	Benar jika \$a tidak sama dengan \$b
Tidak Identik	\$a !== \$b	Benar jika \$a tidak sama dengan \$b atau keduanya tidak memiliki tipe data yang sama
Lebih Kecil	\$a < \$b	Benar jika \$a lebih kecil dari \$b
Lebih Kecil atau Sama	\$a <= \$b	Benar jika \$a lebih kecil dari \$b atau sama dengan \$b

Lebih Besar	$\$a > \b	Benar jika \$a lebih besar dari \$b
Lebih Besar atau Sama	$\$a \geq \b	Benar jika \$a lebih besar dari \$b atau sama dengan \$b

4. Percabangan

- **If**

Contoh kode If dapat diperhatikan pada Gambar 20

```

1  <?php
2
3  if (kondisi) {
4      ....//kode yang akan dijalankan
5      ....//ketika kondisi benar
6  }
7
8  ?>

```

Gambar 20 Sintaks IF pada PHP

- **If Else**

Contoh kode If Else dapat diperhatikan pada Gambar 21

```

1  <?php
2
3  if (kondisi) {
4      ....//kode yang akan dijalankan
5      ....//ketika kondisi true
6  } else {
7      ....//kode yang akan dijalankan
8      ....//ketika kondisi false
9  }
10
11 ?>

```

Gambar 21 Sintaks IF ELSE pada PHP

- **If Else If Else**

Contoh kode If ElseIf Else dapat diperhatikan pada Gambar 22

```
1  <?php
2
3  if (kondisi_1) {
4      ....//kode yang akan dijalankan
5      ....//ketika kondisi 1 benar
6  }elseif (kondisi_2) {
7      ....//kode yang akan dijalankan
8      ....//ketika kondisi 1 salah dan
9      ....//kondisi 2 benar
10 }elseif (kondisi_3) {
11     ....//kode yang akan dijalankan
12     ....//ketika kondisi 2 salah dan
13     ....//kondisi 3 benar
14 }else {
15     ....//kode yang akan dijalankan
16     ....//ketika ketiga kondisi salah
17 }
18
19 ?>
```

Gambar 22 Sintaks IF, ELSEIF, ELSE pada PHP

- **Switch case**

Contoh kode Do While dapat diperhatikan pada Gambar 23

```
1  <?php
2
3  switch ($variable) {
4      ....case $nilai1:
5          ....//kode yang akan dijalankan
6          ....//jika $variabel sama dengan $nilai1
7          ....break;
8      ....case $nilai2:
9          ....//kode yang akan dijalankan
10         ....//jika $variabel sama dengan $nilai2
11         ....break;
12         ....
13     ....default:
14         ....//kode yang akan dijalankan
15         ....//jika tidak ada $nilai yang sama dengan $variabel
16         ....break;
17 }
18
19 ?>
```

Gambar 23 Sintaks Switch Case pada PHP

5. Perulangan

- **For**

Contoh kode For dapat diperhatikan pada Gambar 24

```
1  <?php
2
3  √ for (nilai_awal; kondisi_loop_berhenti; increment/decrement) {
4      ....//Kode yang akan dijalankan ketika looping berjalan
5  }
6
7  ?>
```

Gambar 24 Sintaks FOR pada PHP

- **While**

Contoh kode While dapat diperhatikan pada Gambar 25

```
1  <?php
2
3  nilai_awal;
4  √ while (kondisi_loop_berhenti) {
5      ....//Kode yang akan dijalankan ketika looping berjalan
6
7      ....increment/decrement;
8  }
9
10 ?>
```

Gambar 25 Sintaks WHILE pada PHP

- **Do While**

Contoh kode Do While dapat diperhatikan pada Gambar 26

```

1  <?php
2
3  nilai_awal;
4  do {
5      ....//Kode yang akan dijalankan ketika looping berjalan
6      ....
7      ....increment/decrement;
8  } while (kondisi_loop_berhenti);
9
10 ?>

```

Gambar 26 Sintaks DO WHILE pada PHP

6. Latihan

1. Buatlah proses seperti kalkulator yang bisa melakukan perhitungan :

Penjumlahan	(+)
Pengurangan	(-)
Pembagian	(/)
Perkalian	(*)
Modulus	(%)

Terdapat tiga buah variabel utama yaitu, dua variabel untuk input bilangan dan satu variabel untuk operator

Hasil yang diharapkan bisa diperhatikan pada Gambar 27 :

```
bilangan 1 = 20  
bilangan 2 = 5
```

Berikut merupakan hasil dari setiap operasi

```
PENJUMLAHAN  
Operator : +  
Hasil : 25
```

```
PENGURANGAN  
Operator : -  
Hasil : 15
```

```
PERKALIAN  
Operator : *  
Hasil : 100
```

```
PEMBAGIAN  
Operator : /  
Hasil : 4
```

```
MODULUS  
Operator : %  
Hasil : 0|
```

Gambar 27 Output dari Latihan 1

2. Buatlah program untuk mengurutkan data dari 10 buah data yang acak!
("lanirne", "aduh", "qifuat", "toda", "anevi", "samid", "kifuat");
3. Buatlah program untuk mencari bilangan prima dari 1 – 50 tanpa menggunakan fungsi bawaan!