

POLITEKNIK NEGERI LHOKSEUMAWE

JURUSAN TEKNOLOGI INFORMASI DAN KOMPUTER
TEKNIK INFORMATIKA




TI-2087 — WORKSHOP WEBSITE DASAR

DOSEN PENGAMPU: Muhammad Davi dan Muhammad Reza Zulman 

Gedung Terpadu Jurusan Teknologi Informasi dan Komputer 

muhammad.davi@pnl.ac.id 

2024-2025 Genap 

Daftar Isi











1	Persiapan Lingkungan Kerja	2
1.1	Kebutuhan Aplikasi dan Tools	2
1.2	Instalasi	2
1.3	Personal Web dengan Jekyll dan GitHub Page	3
2	Hyper Text Markup Language I	2
2.1	Versi HTML	2
2.2	Apa itu HTML?	2
2.3	Apa itu Elemen HTML?	3
2.4	Struktur halaman HTML	4
2.5	HTML Links	4
2.6	HTML Lists	4
2.7	Membuat List dan Link ke Website Teman	6
3	Membuat Layout menggunakan SASS/SCSS	2
3.1	What is SCSS?	2
3.2	What is SASS?	3
3.3	Difference between SCSS and SASS	3
3.4	When to Use SCSS	4
3.5	When to Use SASS	4
3.6	Use SCSS for Styling Layout	4
3.7	Conclusion	6
4	Tugas	7
5	Evaluation Criteria	7

1 — PERSIAPAN LINGKUNGAN KERJA

1 Persiapan Lingkungan Kerja

Persiapan lingkungan kerja yang dimaksud adalah persiapan aplikasi dan tools yang digunakan untuk mengembangkan aplikasi menggunakan Laravel. Pada matakuliah ini kita menggunakan Laradock atau Laragon sebagai lingkungan kerja.

1.1 Kebutuhan Aplikasi dan Tools

-  **Sistem Operasi:**  Windows,  Linux, **atau**  macOS
-  **Integrated Development Environment:** Visual Studio Code
-  **Web Browser:** Google Chrome
-  **GitHub:** Web-based version control repository.
-  **Markdown** untuk menulis laporan.
-  **Git** sebagai version control.
-  **Ruby dan Jekyll** untuk membuat halaman web statis.



Note:

Kebutuhan aplikasi dan tool dapat disesuaikan dengan spesifikasi atau kondisi di laptop masing-masing.

1.2 Instalasi

Beberapa kebutuhan aplikasi dan tools diatas perlu diinstall seperti Ruby, Git, Google Chrome dan Visual Studio Code.

a) Install Ruby dan Jekyll

- | Ruby versi 2.7.0 atau yang lebih tinggi, **ruby -v**.
- | RubyGems, **gem -v**.
- | GCC dan Make, **gcc -v** **g++ -v** dan **make -v**.
- | Download Ruby Installer melalui link berikut [Ruby Installer](#).
- | Ikuti form instalsi Ruby Installer.

b) Install Git

- | Download Git Installer melalui link [Git Installer](#)
- | Pilih sesuai dengan sistem operasi yang digunakan, misla Windows.
- | Pada bagian Standalone Installer pilih yang 64 bit.

c) Install Visual Studio Code

- | Download VSCode Installer melalui link [VSCode Installer](#).
- | Pilih sesuai dengan sistem operasi yang digunakan, misla Windows.
- | Pada bagian System Installer pilih yang 64 bit (x64).

d) Install Google Chrome

- | Download Google Chrome Installer melalui link [Google Chrome Installer](#).
- | Klik tombol Download Chrome.

1.3 Personal Web dengan Jekyll dan GitHub Page

Untuk membuat personal web dengan Jekyll dan publish di GitHub Page ikuti langkah-langkah berikut ini:

- ☐ Buat akun di GitHub dengan username sesuai nama masing-masing.
- ☐ Buat repository baru dengan nama username dan github.io.
Contoh: username github = faiza, maka nama repositorinya faiza.github.io.
- ☐ Clone repository tersebut ke lokal.
- ☐ Masuk ke dalam folder repository tersebut kemudian install Jekyll dengan perintah berikut melalui terminal.

Terminal

```
gem install jekyll bundler
```

- ☐ Jalankan perintah bundle init untuk inisialisasi folder tersebut sebagai proyek jekyll seperti perintah berikut ini. Hasil dari perintah tersebut adalah file baru dengan nama **Gemfile**.

Terminal

```
bundle init
```

- ☐ Edit file **Gemfile** menggunakan IDE dan tambahkan kode berikut.

IDE

```
gem "jekyll"
```

- ☐ Buat file baru dengan nama **index.html**, kemudian isi dengan kode berikut.

IDE

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Home</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

- ❑ Jalankan jekyll build untuk build web yang telah dibuat sehingga menghasilkan directory _site.

Terminal

```
jekyll build
```

- ❑ Kemudian jalankan perintah jekyll serve untuk menjalankan web yang telah dibuat di web browser dengan alamat <http://localhost:4000>.

Terminal

```
jekyll serve
```

- ❑ Jika web telah berhasil dibuka, edit file **Gemfile.lock** dengan menambahkan platform linux pada bagian platform seperti pada gambar berikut.

IDE

PLATFORMS

x86_64-linux

- ❑ Selanjutnya push repositori ke GitHub dengan perintah-perintah berikut.

Terminal

```
git add .  
git commit -m "pub: fist publish"  
git push
```

- ❑ Buat GitHub Actions untuk CI/CD agar web otomatis publish.



Note:

Ketika menjalankan jekyll dalam tahap pengembangan, gunakan perintah jekyll serve --livereload agar ketika ada perubahan web browser langsung menampilkan hasil perubahannya. Apabila dalam menjalankan jekyll terjadi error karena port konflik jalankan menggunakan --host dan --port.



Bahan Bacaan:

- [Instal Jekyll dan Ruby di MacOS](#)
- [Instal Jekyll dan Ruby di Ubuntu](#)
- [Git Handbook](#)
- [Markdown Guide](#)
- [GitHub](#)

2 — HYPER TEXT MARKUP LANGUAGE I

2 Hyper Text Markup Language I

Sebelum kita mempelajari tentang HyperText Markup Language (HTML) lebih lanjut, mari kita lihat sekilas sejarah HTML.

- * HTML 1.0 dirilis pada tahun 1993 dengan tujuan untuk berbagi informasi yang dapat dibaca dan diakses melalui browser web. Namun tidak banyak developer yang terlibat dalam pembuatan website. Jadi bahasanya juga tidak berkembang.
- * Kemudian muncul HTML 2.0, diterbitkan pada tahun 1995, yang berisi semua fitur HTML 1.0 bersama dengan beberapa fitur tambahan, yang tetap sebagai bahasa markup standar untuk merancang dan membuat situs web hingga Januari 1997 dan menyempurnakan berbagai fitur inti HTML.
- * Kemudian muncul HTML 3.0, di mana Dave Raggett yang memperkenalkan makalah atau draft baru tentang HTML. Ini termasuk peningkatan fitur baru HTML, memberikan karakteristik yang lebih kuat untuk webmaster dalam merancang halaman web. Namun fitur canggih dari HTML baru ini memperlambat browser dalam menerapkan peningkatan lebih lanjut.
- * Kemudian datanglah HTML 4.01, yang banyak digunakan dan merupakan versi sukses dari HTML sebelum HTML 5.0, yang saat ini dirilis dan digunakan di seluruh dunia. HTML 5 dapat dikatakan untuk versi perpanjangan dari HTML 4.01, yang diterbitkan pada tahun 2012.

Seiring berjalannya waktu, HTML juga mengalami tahap perbaikan yang dikembangkan oleh *World Wide Web Consortium (W3C)*. Pada 2014 silam, menjadi waktu di mana HTML telah mengalami upgrade yang cukup signifikan dan terdapat fitur semantic untuk memudahkan pengembang website dalam menyusun kode dan memberitahukan makna dari konten tersebut, sebagai contoh `<article>`, `<footer>`, dan `<header>`.

2.1 Versi HTML

- 1989 – Tim Berners-Lee invented www
- 1991 – Tim Berners-Lee invented HTML
- 1993 – Dave Raggett drafted HTML+
- 1995 – HTML Working Group defined HTML 2.0
- 1997 – W3C Recommendation: HTML 3.2
- 1999 – W3C Recommendation: HTML 4.01
- 2000 – W3C Recommendation: XHTML 1.0
- 2008 – WHATWG HTML5 First Public Draft
- 2012 – [WHATWG HTML5 Living Standard](#)
- 2014 – [W3C Recommendation: HTML5](#)
- 2016 – W3C Candidate Recommendation: HTML 5.1
- 2017 – [W3C Recommendation: HTML5.1 2nd Edition](#)
- 2017 – [W3C Recommendation: HTML5.2](#)

2.2 Apa itu HTML?

- HTML adalah singkatan dari Hyper Text Markup Language.
- HTML adalah standard markup language untuk membuat halaman web.
- HTML menggambarkan struktur halaman web.
- HTML terdiri dari serangkaian elemen.
- HTML terdiri dari elemen-elemen yang memberi tahu browser cara menampilkan konten.

1. Contoh Dokumen HTML

IDE

```
<!DOCTYPE html>
<html>
<head>
<title>Judul Halaman Web</title>
</head>
<body>

<h1>Heading Pertama</h1>
<p>Paragraf Pertama.</p>

</body>
</html>
```

2. Penjelasan dari Contoh Dokumen HTML

- <!DOCTYPE html> dideklarasikan sebagai dokumen HTML5.
- Elemen <html> adalah elemen root dari halaman HTML.
- Elemen <head> terdiri dari meta informasi tentang halaman HTML.
- Elemen <title> spesifik untuk judul halaman HTML yang akan ditampilkan sebagai judul pada browser atau halaman tab.
- Elemen <body> didefinisikan sebagai body dokumen yang akan ditampilkan pada halaman web seperti heading, paragraf, gambar, link, tabel, list dan lain sebagainya.
- Elemen <h1> sebagai heading yang paling besar.
- Elemen <p> untuk mendefinisikan bagian paragraf.

2.3 Apa itu Elemen HTML?

Sebuah elemen HTML memiliki tag pembuka, konten dan tag penutup: <tag pembuka> konten </tag penutup>. Contoh: <h1>Heading Pertama</h1> atau <p>Paragraf Pertama</p>, seperti yang ditampilkan pada tabel berikut.

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph	</p>
 	none	none

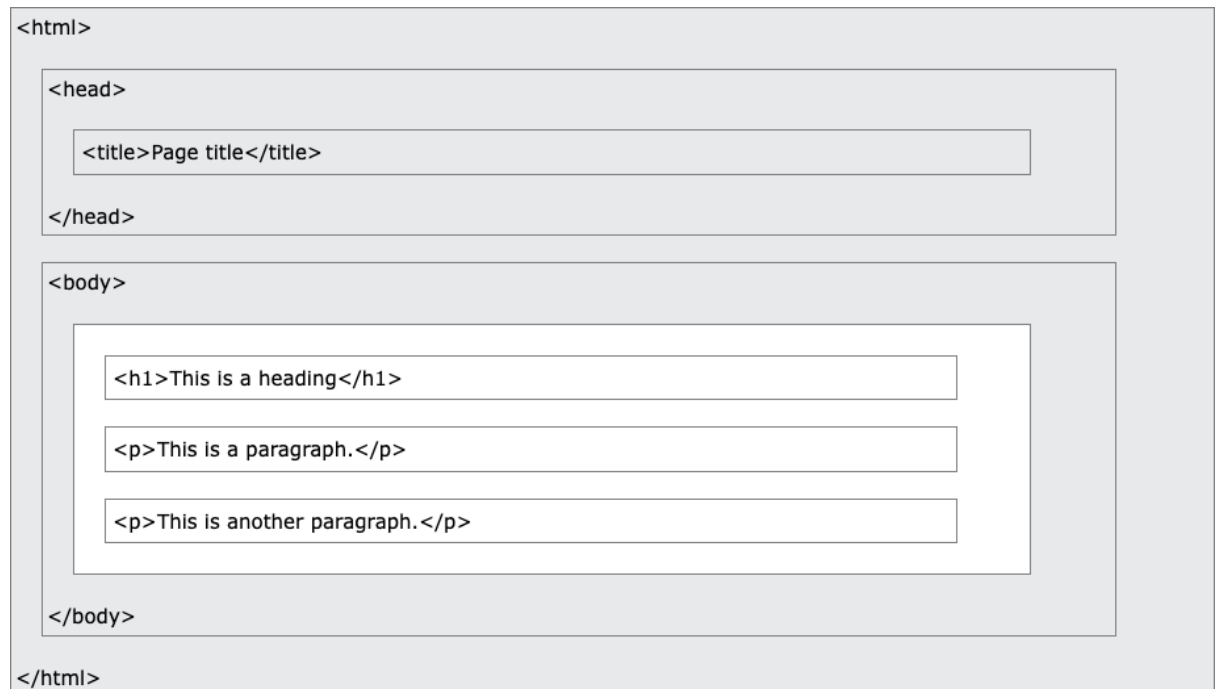


Note:

Beberapa elemen HTML tidak memiliki konten dan tag penutup, seperti
. Itu disebut elemen kosong (*empty elements*).

2.4 Struktur halaman HTML

Berikut adalah struktur halaman HTML:



Note:

Konten yang diisi didalam section `<body>` akan ditampilkan oleh browser sebagai halaman web. Sedangkan yang diisi didalam section `<title>` akan ditampilkan sebagai judul pada browser bar atau tab halaman web.

2.5 HTML Links

1. HTML Link - Sintak

Tag HTML untuk mendefinisikan hyperlink adalah `<a>` seperti pada kode berikut ini.

```
1 <a href="url">link text</a>
```

2. HTML Link - Atribut Target Atribut target spesifik untuk menentukan link dokumen ketika dibuka atau diklik. Beberapa atribut target yang dapat digunakan adalah sebagai berikut.

- `_self` - Default, membuka dokumen pada tab atau jendela yang sama ketika diklik.
- `_blank` - Membuka dokumen pada tab atau jendela baru.
- `_parent` - Membuka dokumen pada frame induknya.
- `_top` - Membuka dokumen pada window dengan full body.

2.6 HTML Lists

1. Daftar Berurutan (*Ordered List*)

Daftar berurutan dimulai dengan tag `` dan diakhiri dengan tag ``. Setiap daftar item dimulai dengan tag `` dan diakhir dengan tag ``. Secara default daftar setiap item ditampilkan dalam bentuk angka. Contoh penggunaan daftar berurutan adalah sebagai berikut.

```

1 <ol>
2   <li>Kopi</li>
3   <li>Susu</li>
4   <li>Teh</li>
5 </ol>

```

2. Daftar Tak Berurutan (*Unordered List*)

Daftar tak berurutan dimulai dengan tag `` dan diakhiri dengan tag ``. Setiap daftar item dimulai dengan tag `` dan diakhiri dengan tag `` sama dengan daftar berurutan. Secara default daftar tak berurutan ditampilkan dalam bentuk bullet (peluru). Contoh penggunaan daftar tak berurutan adalah sebagai berikut.

```

1 <ul>
2   <li>Kopi</li>
3   <li>Susu</li>
4   <li>Teh</li>
5 </ul>

```

Daftar tak berurutan memiliki beberapa item yang bisa dipilih, diantaranya:

Nilai	Deskripsi
disc	Item daftar yang berbentuk bullet (default)
circle	Item daftar yang berbentuk lingkaran
square	Item daftar yang berbentuk persegi
none	Item daftar yang tidak berbentuk (kosong)

Berikut beberapa contoh penggunaan daftar tak berurutan.

• Disc

```

1 <ul style="list-style-type:disc;">
2   <li>Kopi</li>
3   <li>Susu</li>
4   <li>Teh</li>
5 </ul>

```

• circle

```

1 <ul style="list-style-type:circle;">
2   <li>Kopi</li>
3   <li>Susu</li>
4   <li>Teh</li>
5 </ul>

```

• square

```

1 <ul style="list-style-type:square;">
2   <li>Kopi</li>
3   <li>Susu</li>
4   <li>Teh</li>
5 </ul>

```

• none

```

1 <ul style="list-style-type:none;">
2   <li>Kopi</li>
3   <li>Susu</li>
4   <li>Teh</li>
5 </ul>

```

3. Daftar Deskripsi (*Description Lists*) Element `<dl>` digunakan untuk mendefinisikan daftar deskripsi. Element `<dt>` digunakan untuk mendefinisikan ketentuan (*term*) dan elemen `<dd>` untuk mendeskripsikan ketentuan dari daftar deskripsi. Berikut contoh menggunakan daftar deskripsi.

```
1 <dl>
2   <dt>Kopi</dt>
3   <dd>- Kopi hitam manis.</dd>
4   <dt>Susu</dt>
5   <dd>- Susi coklat dingin.</dd>
6 </dl>
```

2.7 Membuat List dan Link ke Website Teman

Pada tahap ini kita akan membuat link dengan HTML dan Markdown untuk terhubung dengan website-website teman sekelas.

- ❑ Pastikan commit yang dilokal dan di GitHub sudah sama, jika belum lakukan git pull.
- ❑ Ubah file index.html menjadi seperti berikut ini.

```
IDE
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Home</title>
</head>
<body>
  <h1>Nama: Muhammad Davi</h1>
  <p>Deskripsi Diri: Saya berasal dari Aceh Timur, sekarang bekerja di Politeknik Negeri Lhokseumawe sebagai Tenaga Pendidik. Jelaskan selengkap mungkin....</p>
  <h2>Berikut Teman-teman saya di kelas:</h2>
  <ul>
    <li><a href="https://mohdrzu.github.io/" target="_blank">Muhammad Reza Zulman</a></li>
  </ul>
</body>
</html>
```

- ❑ Pada elemen `<h1>`, `<p>`, dan `` sesuaikan dengan data pribadi dan kelas masing-masing.
- ❑ Buat file baru di direktori root dengan nama friends.md dan isi dengan kode berikut.

IDE

```
---  
layout: default  
title: Friends  
---  
# Friends page  
  
Berikut teman-teman saya di kelas:  
- \[Muhammad Reza Zulman\]\(https://mohdrzu.github.io/\)  
- .....
```

- ☐ Pada bagian link sesuaikan dengan link yang telah dibuat pada index.html.
- ☐ Jika semua sudah benar dan selesai publish ke github masing-masing.

Bahan Bacaan:

- [W3Schools Online Web Tutorials](#)

3 — MEMBUAT LAYOUT MENGGUNAKAN SASS/SCSS

3 Membuat Layout menggunakan SASS/SCSS

SASS and SCSS are both syntaxes of the SASS (Syntactically Awesome Stylesheets) preprocessor, each offering enhanced features that extend the capabilities of traditional CSS. While both share the same functionality, their syntaxes differ, catering to different preferences among developers.

3.1 What is SCSS?

SCSS (Sassy CSS) is a syntax of SASS that adopts a CSS-like structure. This makes it easier for developers who are already familiar with CSS to transition into using advanced features offered by SASS, like variables, mixins, and nesting. SCSS files have the .scss extension and are fully compatible with standard CSS, meaning you can use any valid CSS in an SCSS file.

Here are the main features of SCSS:

- Variables: Store reusable values like colors and fonts for consistent styling.
- Nesting: Nest CSS selectors in a hierarchical manner for better readability.
- Mixins: Create reusable chunks of styles, avoiding repetitive code.
- Inheritance: Use @extend to share styles between selectors, simplifying the code-base.
- Partial and Importing: Modularize CSS using @import, keeping styles organized and maintainable.

IDE

```
/* .scss file */
$bgcolor: blue;
$textcolor: red;
$fontsize: 25px;

/* Use the variables */
body {
  background-color: $bgcolor;
  color: $textcolor;
  font-size: $fontsize;
}
```

Output CSS:

```
1 body {
2   background-color: blue;
3   color: red;
4   font-size: 25px;
5 }
6 /* now this can apply resulting html file */
```


3.2 What is SASS?

SASS (Syntactically Awesome Stylesheets) is the core preprocessor that extends CSS. It provides powerful tools like variables, nested rules, mixins, and functions, allowing for more efficient and organized stylesheets. SASS uses indentation-based syntax, meaning there are no curly braces or semicolons ;, unlike regular CSS. SASS files use the .sass extension.

Here are the main features of SASS:

- Variables: Define reusable values like colors and fonts.
- Nesting: Organize your CSS in a hierarchical, visual structure.
- Partials: Break down large stylesheets into smaller files with @import.
- Mixins: Reuse sets of CSS rules across different selectors.
- Inheritance: Use @extend to share styles between selectors.

IDE

```
/* .sass file */
$bgcolor: blue
$textcolor: red
$fontsize: 25px

/* Use the variables */
body
  background-color: $bgcolor
  color: $textcolor
  font-size: $fontsize
```

Output CSS:

```
1 body {
2   background-color: blue;
3   color: red;
4   font-size: 25px;
5 }
6 /* now this can apply resulting html file */
```

3.3 Difference between SCSS and SASS

Here are the key differences between SCSS and SASS:

Feature	SCSS	SASS
Syntax	CSS-like syntax with braces and semicolons	Indentation-based syntax without braces or semicolons
File Extension	.scss	.sass
Compatibility	Fully compatible with all versions of CSS	Requires a different syntax, not directly compatible with standard CSS
Flexibility	Familiar to those who know CSS	More concise and cleaner for some developers
Usage	Ideal for developers transitioning from CSS	Preferred by those who favor a streamlined syntax

3.4 When to Use SCSS

SCSS is a good choice for developers who are familiar with CSS. It uses the same structure, making it easier to learn and work with, especially in larger projects.

- Easy for CSS developers to pick up
- Ideal for larger projects
- Works well with standard CSS

3.5 When to Use SASS

SASS is great for developers who prefer a simpler and cleaner syntax. It's perfect for smaller projects where readability and brevity are important.

- Best for simple, clean code
- Great for smaller projects
- No need for CSS compatibility

3.6 Use SCSS for Styling Layout

Untuk praktikum kali ini kita akan mengubah layout website agar lebih rapi dan menarik. Ikuti langkah-langkah berikut ini.

1. Update file navigation.html yang ada didalam folder _includes dengan kode berikut ini.

```

IDE
<ul>
  {% for item in site.data.navigation %}
    <li>
      <a href="{ {{ item.link }}" {% if page.url == item.link %} class="active"{% endif %}>
        {{ item.name }}
      </a>
    </li>
  {% endfor %}
</ul>

```

2. Update kode SCSS pada file main.scss seperti kode berikut ini.

```
$primary-color: #3b7b3a;
$secondary-color: #fbbc05;
$font-stack: Helvetica, sans-serif;

body,
h1,
h2,
p {
  margin: 0; padding: 0; box-sizing: border-box;
}

body {
  font-family: $font-stack; line-height: 1.6;
  background-color: #f4f4f4;
}

header {
  text-align: center; padding: 24px; color: #fff;
  background-color: $primary-color;
}

main {
  padding: 24px; margin-top: 24px;
  box-shadow: 0 0 8px rgba(0, 0, 0, 0.1);
  background-color: #fff;
}

footer {
  text-align: center; padding: 8px; margin-top: 24px; color: #fff;
  background-color: $primary-color;
}

h2 {
  color: #333; margin-bottom: 8px;
}

nav {
  margin: 24px 0;
  ul {
    list-style-type: none; margin: 0; padding: 0; overflow: hidden;
    background-color: $primary-color;
  }
  li {
    float: left;
    a {
      display: block; color: white; text-align: center; padding: 8px 16px;
      text-decoration: none;
      &:hover:not(.active) {
        background-color: $secondary-color;
      }
    }
  }
}

.container {
  width: 80%; margin: auto; overflow: hidden;
}

.profile-picture {
  width: 150px; height: 150px; border-radius: 50%; margin: 24px auto;
  background-color: $secondary-color;
}

.active {
  background-color: $secondary-color;
}
```

3. Update file default.html yang ada didalam folder _layouts dengan kode berikut ini.

```
IDE

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>{{ page.title }}</title>

  <!-- Stylesheet -->
  <link rel="stylesheet" href="{{ site.baseurl }}/assets/css/styles.css">

  <!-- Favicon -->
  <link rel="icon" href="{{ site.baseurl }}/assets/images/favicon.ico">

  <!-- JavaScript -->
  <script src="{{ site.baseurl }}/assets/js/scripts.js"></script>

  <!-- plugins -->
  {% feed_meta %}
  {% seo %}
</head>
<body>
  <div class="container">
    <header>
      
      <h1>{{ site.title }}</h1>
      <p>{{ site.description }}</p>
    </header>

    <nav>
      {% include navigation.html %}
    </nav>

    <main>
      {{ content }}
    </main>

    <footer>
      <p>&copy; {{ site.time | date: '%Y' }} {{ site.title }}. All rights reserved.</p>
    </footer>
  </div>
</body>
</html>
```

3.7 Conclusion

In summary, SCSS and SASS provide the same functionality, but the choice between them depends on your personal preference for syntax. SCSS is more suited for developers familiar with traditional CSS, while SASS's minimalistic approach appeals to those who prioritize cleaner code. Both offer features like variables, mixins, and nesting to enhance CSS development.

Bahan Bacaan:

- [GeeksforGeeks](#)

4 Tugas

Tugas yang perlu dipersiapkan antara lain:

- ✍ Website yang dikerjakan secara individu dan tidak terdeteksi plagiat.
- ✍ Postingan diwebsite setiap pertemuan.

5 Evaluation Criteria

Segiap tugas akan dinilai berdasarkan kriteria penilaian seperti berikut.

Criteria	Points
Responsi	15%
Laporan	15%
Hasil Kerja	30%
Seminar	40%
Total	100%