

LAPORAN PRAKTIKUM

“CASCADING STYLE SHEET”

Dalam rangka memenuhi tugas mata kuliah Praktikum Desain Website

Dosen pengampu: Vearen Dika Sofirudin, S.Pd., M.Ed.



Disusun oleh:

Nama : Eka Putra Noandheli

NIM : K3524049

PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER
FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN
UNIVERSITAS SEBELAS MARET
SURAKARTA
2025

A. Tujuan Praktikum

Tujuan dari praktikum ini adalah untuk:

1. Memahami dan menerapkan struktur dasar dokumen web menggunakan HTML.
2. Mengenali dan menggunakan Internal CSS untuk mengatur gaya serta tampilan halaman.
3. Memahami konsep *margin* dan *padding* untuk mengatur jarak antar elemen.
4. Mampu membuat daftar (*list*) dan navigasi menggunakan tag ``, ``, dan `<a>`.
5. Memahami penggunaan *iframe* untuk menampilkan halaman lain di dalam halaman utama.
6. Menganalisis dan memperbaiki kesalahan sintaksis pada kode HTML.

B. Ringkasan Materi

CSS (*Cascading Style Sheets*) berfungsi untuk memisahkan isi dengan tata letak halaman web. CSS memperkenalkan style atau template untuk mempermudah penulisan halaman web. Dengan CSS, desainer web dapat dengan mudah mengatur tampilan keseluruhan situs hanya dengan mengubah atribut pada satu file CSS tanpa harus mengubah setiap elemen satu per satu.

Cara kerja CSS melibatkan dua elemen dasar:

1. Selector: Merupakan tag HTML atau tag yang diberi nama sendiri oleh desainer.
2. Declarator: Atribut atau properti yang menentukan gaya suatu elemen, seperti `'background'`, `'color'`, dan `'font-family'`.

Struktur dasar penulisan CSS terdiri dari `'selector'` diikuti oleh blok deklarasi yang berisi satu atau lebih `'properti: nilai;'`. Contohnya, `p { background: yellow; color: blue; }` akan membuat semua paragraf memiliki latar belakang kuning dan teks berwarna biru.

C. Langkah Kerja

Praktikum ini terdiri dari beberapa latihan yang saling berkaitan.

1. Latihan 1


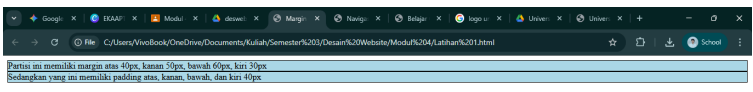
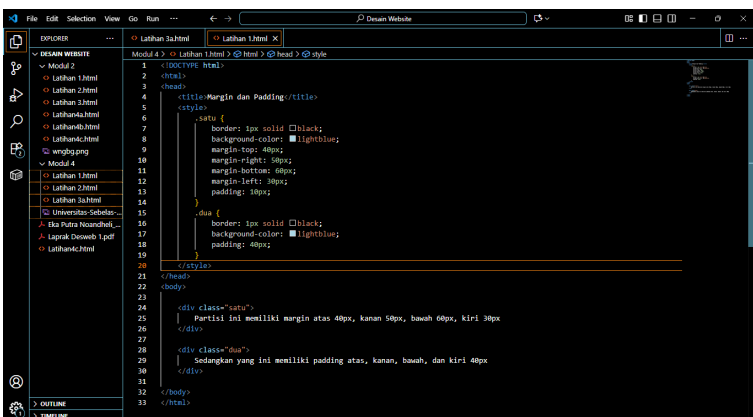
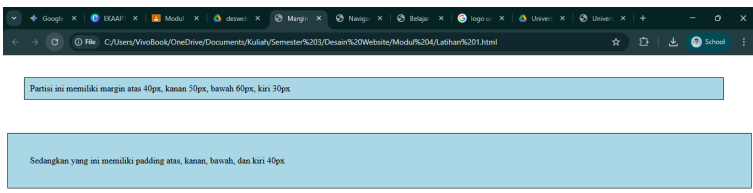
Kode awal sudah memiliki struktur HTML yang lengkap, tetapi ada masalah pada bagian CSS-nya. Kode tersebut hanya memiliki selector (*.satu* dan *.dua*) tanpa properti yang mempengaruhi margin atau padding sesuai dengan teks yang dijelaskan di dalam *div*.

Analisis Kode Salah:

- a. Masalah Utama: Kode CSS tidak memiliki properti margin atau padding. Meskipun ada *border* dan *background-color*, kode tersebut tidak menerapkan jarak yang dijelaskan dalam teks di tag *div*.
- b. Tujuan yang Tidak Tercapai: Kode ini tidak bisa menampilkan margin 40px, 50px, 60px, dan 30px, atau *padding* 40px, karena properti tersebut tidak didefinisikan dalam *block <style>*.

Langkah Perbaikan:

- a. Menambahkan Properti CSS: Untuk memenuhi deskripsi yang ada di dalam HTML, Anda harus menambahkan properti CSS yang sesuai ke dalam selector yang relevan.
- b. Perbaikan *.satu*: Tambahkan properti *margin-top*, *margin-right*, *margin-bottom*, dan *margin-left* dengan nilai yang sesuai (40px, 50px, 60px, 30px) ke dalam *selector .satu*. Ini akan menciptakan jarak di luar elemen *div*. Properti *padding* juga ditambahkan ke *.satu* agar elemen terlihat lebih baik secara visual.
- c. Perbaikan *.dua*: Tambahkan properti *padding: 40px;* ke dalam selector *.dua*. Properti *padding* ini akan menciptakan ruang di dalam elemen *div*, antara *border* dan konten teks.
- d. Struktur tetap utuh: Tidak ada perubahan yang diperlukan pada struktur HTML itu sendiri karena struktur aslinya sudah benar. Perbaikan hanya difokuskan pada CSS di dalam tag *<style>*.

Kode Awal	 <pre> 1 <DOCTYPE html> 2 <html> 3 <head> 4 <title>Margin dan Padding</title> 5 <style> 6 7 .satu { 8 border: 1px solid black; 9 background-color: lightblue; 10 } 11 12 .dua { 13 border: 1px solid black; 14 background-color: lightblue; 15 } 16 </style> 17 </head> 18 <body> 19 <div class="satu"> 20 Partisi ini memiliki margin atas 40px, kanan 50px, bawah 60px, kiri 30px 21 </div> 22 23 <div class="dua"> 24 Sedangkan yang ini memiliki padding atas, kanan, bawah, dan kiri 40px 25 </div> 26 27 </body> 28 </html> </pre>
Hasil awal	 <p>Partisi ini memiliki margin atas 40px, kanan 50px, bawah 60px, kiri 30px</p> <p>Sedangkan yang ini memiliki padding atas, kanan, bawah, dan kiri 40px</p>
Kode revisi	 <pre> 1 <DOCTYPE html> 2 <html> 3 <head> 4 <title>Margin dan Padding</title> 5 <style> 6 7 .satu { 8 border: 1px solid black; 9 background-color: lightblue; 10 margin-top: 40px; 11 margin-right: 50px; 12 margin-bottom: 60px; 13 margin-left: 30px; 14 padding: 10px; 15 } 16 17 .dua { 18 border: 1px solid black; 19 background-color: lightblue; 20 padding: 40px; 21 } 22 </style> 23 </head> 24 <body> 25 <div class="satu"> 26 Partisi ini memiliki margin atas 40px, kanan 50px, bawah 60px, kiri 30px 27 </div> 28 29 <div class="dua"> 30 Sedangkan yang ini memiliki padding atas, kanan, bawah, dan kiri 40px 31 </div> 32 33 </body> 34 </html> </pre>
Hasil akhir	 <p>Partisi ini memiliki margin atas 40px, kanan 50px, bawah 60px, kiri 30px</p> <p>Sedangkan yang ini memiliki padding atas, kanan, bawah, dan kiri 40px</p>

2. Latihan 2

Kode pertama adalah fondasi yang baik, tetapi memiliki beberapa kekurangan pada bagian CSS yang membuatnya tidak berfungsi sebagai navigasi yang seharusnya.

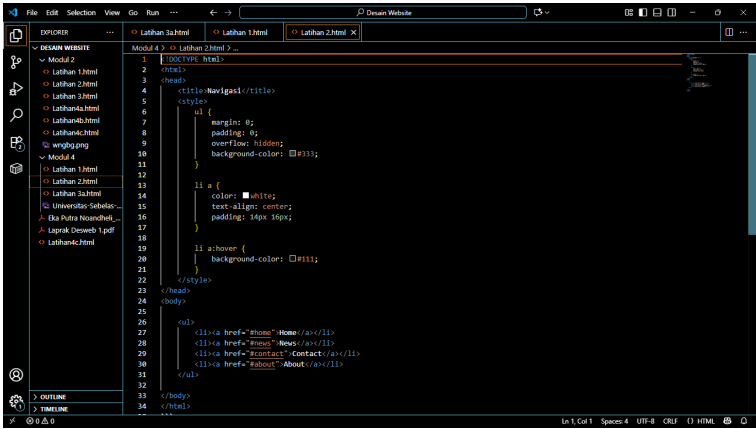
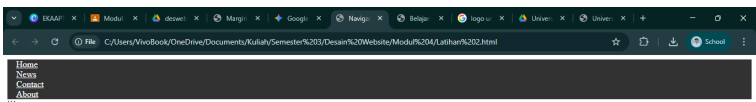
Analisis Kode Salah:

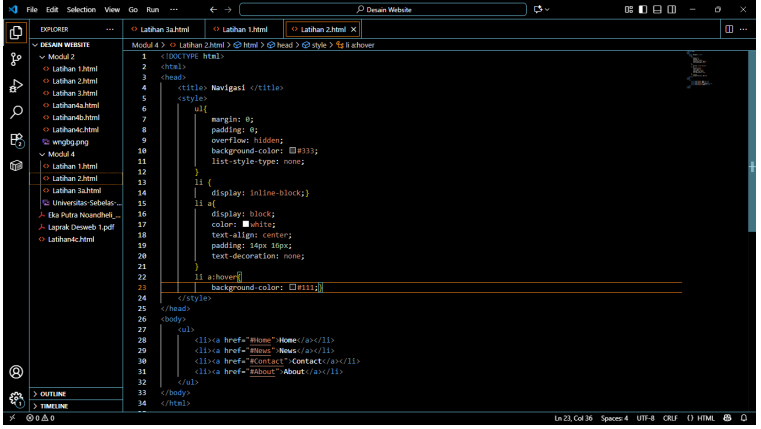
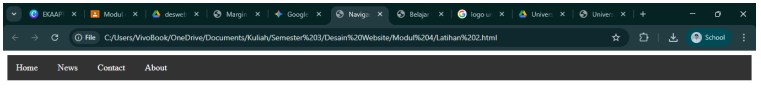
- Tampilan Awal yang Bermasalah: Pada kode ini, elemen `` (*unordered list*) masih akan menampilkan bullet point secara default. Selain itu, elemen `` (*list item*) adalah elemen *block-level*, yang berarti setiap item akan ditampilkan di baris baru dan tersusun ke bawah secara vertikal, bukan horizontal.

- b. Fungsi yang Belum Lengkap: Tautan (<a>) masih akan memiliki garis bawah (*underline*) secara default, yang sering kali ingin dihilangkan pada desain navigasi modern. Kode ini belum menyertakan properti CSS untuk mengatasi masalah ini.

Langkah Perbaikan

- a. Menghilangkan *Bullet Point*: Perbaikan: Properti *list-style-type: none;* ditambahkan ke *selector ul*. Fungsi: Properti ini menghilangkan *bullet point* yang merupakan ciri khas dari elemen , membuat daftar terlihat bersih dan profesional.
- b. Menyusun Item Secara Horizontal: Perbaikan: Properti *display: inline-block;* ditambahkan ke *selector li*. Fungsi: Properti ini mengubah perilaku elemen dari *block* menjadi *inline-block*. Ini memungkinkan setiap item untuk ditempatkan secara berdampingan di baris yang sama, sehingga membentuk bilah navigasi horizontal.
- c. Menghilangkan Garis Bawah Tautan: Perbaikan: Properti *text-decoration: none;* ditambahkan ke *selector li a*. Fungsi: Properti ini menghapus garis bawah dari setiap tautan, memberikan tampilan yang lebih bersih dan sesuai dengan standar desain web modern.

Kode Awal	
Hasil awal	

<p>Kode revisi</p>	
<p>Hasil akhir</p>	

3. Latihan 3a

Kode pertama memiliki satu kesalahan fatal yang bisa merusak struktur halaman.

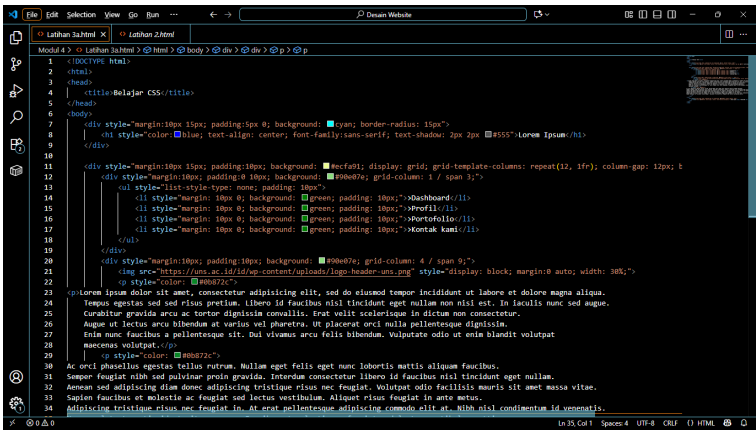
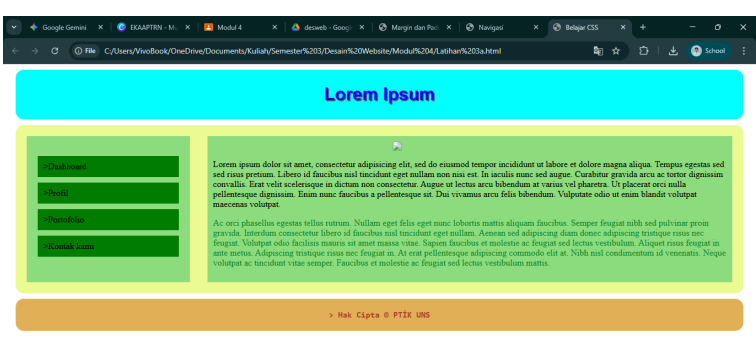

Analisis kode salah:

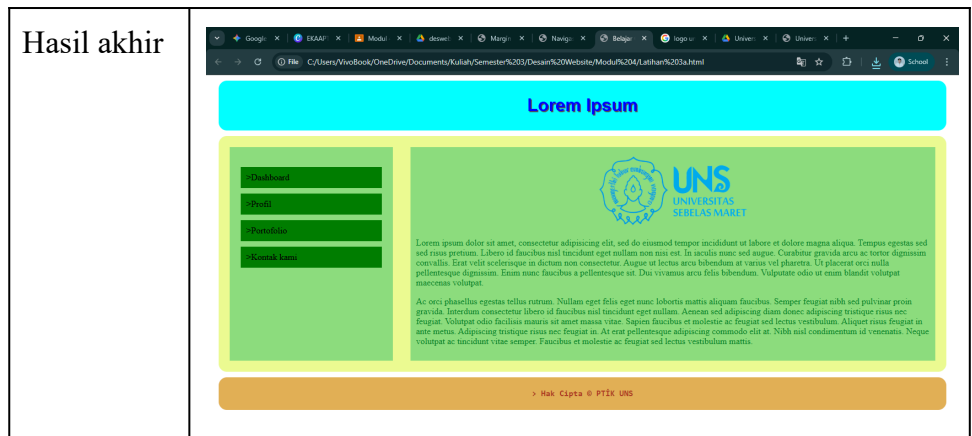
- Masalah Utama: Tag `<p>` Bersarang (*Nested*) Pada kode yang salah, dalam kode disisipkan tag `<p>` di dalam tag `<p>` lainnya, seperti ini: `<p style="..."> <p>Lorem ipsum...</p>`. Ini adalah kesalahan sintaksis yang serius. Dalam HTML, elemen paragraf (`<p>`) tidak boleh mengandung elemen tingkat blok lainnya, termasuk tag `<p>` itu sendiri.
- Hal ini akan membingungkan peramban dan bisa menyebabkan teks tidak diformat dengan benar atau merusak tata letak yang sudah dibuat dengan CSS Grid.

Langkah Perbaikan:

- Memperbaiki Tag Paragraf (`<p>`): Langkah Perbaikan: Tag `<p>` yang bersarang dihapus. Kode ini sekarang memiliki dua tag `<p>` yang terpisah dan berurutan. Hasil: Struktur dokumen menjadi valid dan logis. Paragraf pertama (*Lorem ipsum...*) sekarang berada di dalam tag `<p>`-nya sendiri, diikuti oleh paragraf kedua (*Ac orci phasellus...*) yang juga berada di dalam tag `<p>`-nya sendiri. Hal ini memastikan setiap paragraf ditampilkan dengan benar.

- b. Mengubah Sumber Gambar (*src*): Perubahan Fungsional: Sumber gambar (*src*) pada tag `` diubah dari URL web (`https://.../logo-header-uns.png`) menjadi jalur file lokal (`Universitas-Sebelas-Maret-Logo.png`). Implikasi: Perubahan ini berarti gambar tidak lagi diambil dari internet, melainkan harus disimpan di folder yang sama dengan file HTML. Ini adalah praktik umum untuk memastikan halaman web dapat ditampilkan secara offline dan untuk mengurangi ketergantungan pada koneksi internet.

Kode Awal	
Hasil awal	
Kode revisi	



4. Latihan 3b

a. Membuat File CSS (style.css)

Membuat file dengan nama *style.css*. Kode ini berisi semua gaya yang sebelumnya ditulis langsung di dalam tag HTML pada Latihan 3b.

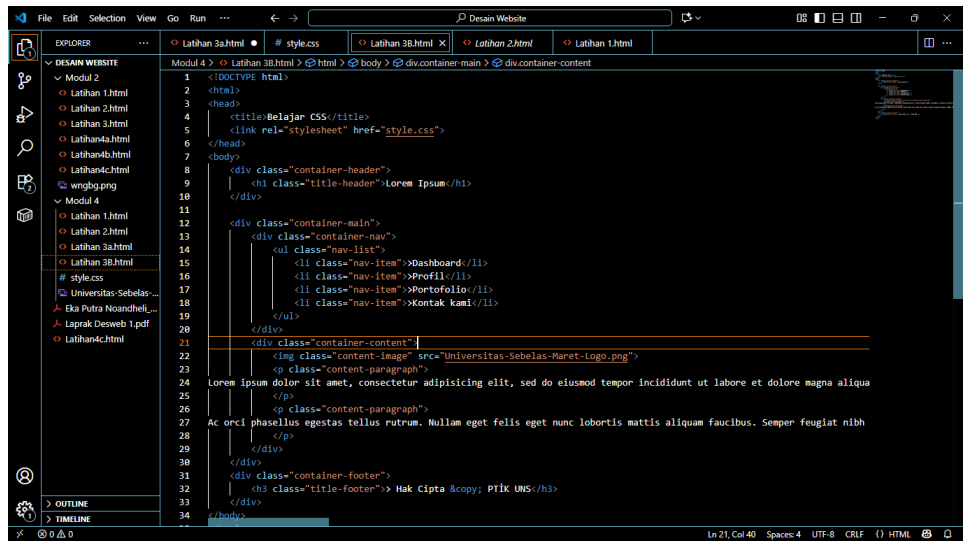
```

1  .container-header {
2      margin: 10px 15px;
3      padding: 5px 0;
4      background-color: cyan;
5      border-radius: 15px;
6  }
7
8  .title-header {
9      color: blue;
10     text-align: center;
11     font-family: sans-serif;
12     text-shadow: 2px 2px #555;
13 }
14
15 .container-main {
16     margin: 10px 15px;
17     padding: 10px;
18     background-color: #e6fa91;
19     display: grid;
20     grid-template-columns: repeat(12, 1fr);
21     column-gap: 12px;
22     border-radius: 15px;
23 }
24
25 .container-nav {
26     margin: 10px;
27     padding: 0 10px;
28     background-color: #90e07e;
29     grid-column: 1 / span 3;
30 }
31
32 .nav-list {
33     list-style-type: none;
34     padding: 10px;

```

b. Ubah File HTML (Latihan3b.html)

Membuat file dengan nama Latihan3b.html. Kode ini berisi hanya kode html dan sekarang menautkan ke file *style.css* dan menggunakan class untuk menerapkan gaya.



The screenshot shows the VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'DESAIN WEBSITE' with files for 'Modul 2', 'Modul 3', 'Modul 4', and 'style.css'. The code editor shows the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Belajar CSS</title>
5   <link rel="stylesheet" href="style.css">
6 </head>
7 <body>
8   <div class="container-header">
9     <h1 class="title-header">Lorem Ipsum</h1>
10  </div>
11
12  <div class="container-main">
13    <div class="container-nav">
14      <ul class="nav-list">
15        <li class="nav-item"><a href="#">Dashboard</a></li>
16        <li class="nav-item"><a href="#">Profil</a></li>
17        <li class="nav-item"><a href="#">Portofolio</a></li>
18        <li class="nav-item"><a href="#">Kontak kami</a></li>
19      </ul>
20    </div>
21    <div class="container-content">
22      
23      <p class="content-paragraph">
24        Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
25      </p>
26      <p class="content-paragraph">
27        Ac orci phasellus egestas tellus rutrum. Nullam eget felis eget nunc lobortis mattis aliquam faucibus. Semper feugiat nibh
28      </p>
29    </div>
30  </div>
31  <div class="container-footer">
32    <h3 class="title-footer"><a href="#"> Hak Cipta &copy; PTik UNS</a>
33  </div>
34 </body>
```

c. Hasil



D. Kesimpulan

Dari serangkaian praktikum ini, dapat disimpulkan bahwa sintaksis yang benar adalah hal yang paling krusial dalam pemrograman web. Sebuah kesalahan kecil, seperti tag yang tidak ditutup atau karakter yang salah, dapat membuat seluruh kode tidak berfungsi. Selain itu, praktikum ini memperlihatkan bahwa:

1. HTML dan CSS bekerja secara sinergis, di mana HTML menyediakan kerangka dasar, sementara CSS memberikan tampilan yang menarik dan terorganisir.
2. Konsep dasar CSS seperti margin dan padding sangat penting untuk mengontrol jarak dan tata letak elemen.
3. Penggunaan eksternal CSS (seperti yang dilakukan pada Latihan 3b) adalah praktik terbaik yang membuat kode lebih rapi, terpusat, dan mudah dikelola.
4. Kemampuan untuk menganalisis dan memperbaiki kesalahan adalah keterampilan fundamental yang harus dikuasai untuk menciptakan halaman web yang valid dan fungsional.

Secara keseluruhan, praktikum ini memberikan pemahaman dasar yang kokoh tentang bagaimana elemen-elemen web saling berinteraksi untuk membentuk halaman yang terstruktur dan interaktif.