



# **Pemorograman Web**

## **LAPORAN PRAKTIKUM HTML**

Oleh :

*Muhammad Nabel*

*2411102441211*

Teknik Informatika  
Fakultas Sains & Teknologi  
Universitas Muhammadiyah Kalimantan Timur  
Samarinda, 2025

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

HTML (Hypertext Markup Language) adalah bahasa markup standar yang menjadi fondasi utama dalam pengembangan web sejak pertama kali diperkenalkan oleh Tim BernersLee pada tahun 1990. Di era digital yang berkembang pesat saat ini, website telah menjadi sarana utama untuk menyampaikan informasi, melakukan transaksi bisnis, hingga berinteraksi dengan pengguna internet di seluruh dunia. Sebagai elemen dasar dari setiap halaman web, HTML memungkinkan pengembang menyusun struktur konten digital dan menampilkannya dalam format yang bisa diakses melalui berbagai perangkat serta browser.

Perkembangan teknologi informasi dan internet telah mengubah cara manusia berkomunikasi dan berbagi informasi. Keberadaan HTML sebagai bahasa universal dalam pengembangan web memegang peran penting karena dapat berjalan di berbagai perangkat dan browser tanpa memerlukan plugin tambahan. HTML juga mampu berintegrasi dengan teknologi lain seperti CSS untuk tampilan visual dan JavaScript untuk interaktivitas, sehingga semakin memperkuat perannya dalam ekosistem digital. Proses standarisasi yang dilakukan oleh World Wide Web Consortium (W3C) memastikan HTML terus berkembang mengikuti kebutuhan global, sambil tetap menjaga konsistensi dan kompatibilitas lintas platform.

Dalam dunia pendidikan maupun industri digital modern, HTML memiliki relevansi yang tinggi. Kemampuan memahami dan mengimplementasikan HTML telah menjadi keterampilan dasar yang penting, tidak hanya bagi pengembang web tetapi juga bagi banyak profesional di berbagai bidang. Penguasaan HTML menjadi nilai tambah yang signifikan untuk menghadapi persaingan pasar kerja digital yang semakin ketat. Kini, HTML telah berevolusi hingga versi HTML5 yang menghadirkan berbagai fitur canggih, seperti dukungan multimedia, penyimpanan lokal, hingga kemampuan aplikasi web yang lebih responsif. Hal ini menjadikan HTML semakin relevan dalam pengembangan platform digital masa kini.

#### **1.2 Tujuan**

- a. Memahami konsep dasar serta struktur HTML sebagai bahasa markup standar dalam pengembangan web.
- b. Mempelajari dan mengenali tag-tag penting HTML beserta atributnya yang digunakan dalam pembuatan halaman web.
- c. Mengembangkan kemampuan untuk menyusun konten web secara terstruktur dan sistematis.
- d. Memperoleh keterampilan praktis dalam membuat halaman web sederhana yang fungsional.
- e. Menguasai pembuatan elemen-elemen web seperti formulir, tabel, daftar, serta cara menyisipkan gambar.
- f. Mempelajari cara menghubungkan antarhalaman web menggunakan hyperlink
- g. Mengimplementasikan elemen-elemen semantik HTML5 untuk menciptakan struktur halaman yang lebih baik.
- h. Membangun pemahaman tentang standar web serta praktik terbaik dalam pengembangan HTML.
- i. Mempelajari proses validasi HTML agar kode yang dibuat sesuai dengan standar.

- j. Memahami konsep dasar aksesibilitas web sehingga konten dapat diakses oleh berbagai jenis pengguna.
- k. Mengetahui dasar-dasar responsivitas web untuk mendukung tampilan di berbagai ukuran perangkat.

### 1.3 Tujuan Pustaka

HTML (Hypertext Markup Language) pertama kali dikembangkan oleh Tim Berners-Lee pada tahun 1990 sebagai bagian dari proposal World Wide Web (Castro & Hyslop, 2022). Sejak saat itu, HTML menjadi fondasi utama dalam pembuatan halaman web dengan memanfaatkan elemen dan tag untuk mendefinisikan struktur serta konten (Robbins, 2018). Perjalanan evolusi HTML dimulai dari HTML 2.0 pada tahun 1995 hingga hadirnya HTML5 pada 2014. Setiap versi membawa tambahan fitur dan kemampuan baru yang semakin memperluas fungsinya (Duckett, 2011).

HTML5 memberikan lompatan besar dalam pengembangan web dengan menghadirkan elemen semantik baru seperti `<header>`, `<main>`, dan `<section>`. Selain itu, HTML5 juga menawarkan dukungan multimedia secara langsung tanpa plugin tambahan serta memperkuat kemampuan aplikasi web (Freeman, 2019). Konsep Progressive Enhancement mempertegas peran HTML sebagai lapisan konten dan struktur dasar yang dapat diakses semua pengguna, tanpa bergantung pada kemampuan browser tertentu (Keith, 2020).

Validasi HTML juga memiliki peranan penting dalam pengembangan web modern. Hal ini ditekankan oleh Kyrnin (2021), yang menyatakan bahwa proses validasi memastikan dokumen HTML sesuai standar W3C. Dengan begitu, kompatibilitas lintas browser, aksesibilitas, dan SEO dapat meningkat. Sejalan dengan itu, studi oleh Chen et al. (2017) menunjukkan bahwa penguasaan HTML masih menjadi keterampilan fundamental yang sangat dibutuhkan. Bahkan, 98% lowongan pekerjaan di bidang pengembangan web mensyaratkan kemampuan dalam HTML.

## **BAB II ALAT DAN BAHAN**

### 2.1 Alat

- a. Komputer
- b. Visual Studio Code
- c. Browser

### 2.2 Bahan

- a. Dokumentasi

## BAB III PROSEDUR KERJA

### 1. Latihan 1

Menuliskan struktur dasar HTML dengan judul <h1> dan paragraf<p>

```
1 <!DOCTYPE html>
2 <html lang="id/.">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Struktur HTML Dasar</title>
7 </head>
8 <body>
9   <h1>Selamat Datang di Halaman Saya</h1>
10  <p>ini adalah halaman web pertama saya</p>
11 </body>
12 </html>
```

### 2. Latihan 2

Menuliskan struktur dasar HTML dengan berbagai heading dari <h1> sampai <h6> untuk menunjukkan tingkatan judul dan subjudul

```
1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Belajar Heading</title>
7 </head>
8 <body>
9   <!-- Heading dari h1 sampai h6 -->
10  <h1>ini adalah Heading 1(judul utama)</h1>
11  <h2>ini adalah heading 2(sub judul)</h2>
12  <h3>ini adalah heading 3</h3>
13  <h4>ini adalah heading 4</h4>
14  <h5>ini adalah heading 5</h5>
15  <h6>ini adalah heading 6</h6>
16
17  <!-- contoh penggunaan heading dalam struk dokumen -->
18  <h1>Bab 1: pengenalan web</h1>
19  <h2>1.1 Apa itu HTML?</h2>
20  <h3>Definisi HTML</h3>
21  <h3>sejarah HTML</h3>
22  <h2>1.2 Mengapa Belajar HTML?</h2>
23 </body>
24 </html>
```

### 3. Latihan 3

Menuliskan struktur dasar HTML dengan paragraf `<p>` dan penerapan format teks seperti bold, italic, garis bawah `<u>`, highlight `<mark>`, line break `<br>`, dan garis horizontal `<hr>`.

```

1 <!DOCTYPE html>
2 <html Lang="id">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Belajar Paragraf</title>
7 </head>
8 <body>
9   <h1>Contoh paragraf dan text formatting</h1>
10
11   <!-- paragraf biasa -->
12   <p>
13     ini adalah paragraf pertama. HTML adalah bahasa warkup yang digunakan untuk membuat halaman web.
14   </p>
15
16   <p>
17     ini adalah paragraf kedua. setiap paragraf akan otomatis diberi oleh browser
18   </p>
19
20   <!-- paragraf dengan text formatting -->
21   <p>
22     Dalam paragraf ini kita bisa menggunakan
23     <strong>text tebal(bold)</strong> dan <em>text miring(italic)</em>.
24   </p>
25
26   <p>
27     Kita juga bisa membuat text <u>bergaris bawah</u> dan
28     <mark>highlight</mark>.
29   </p>
30   <!-- Break line -->
31   <p>Baris pertama<br /> Baris kedua dengan line break</p>
32
33   <!-- Horizontal rule -->
34   <hr />
35
36   <p>Gari horizontal di atas dibuat dengan hr.</p>
37 </body>
38 </html>

```

### 4. Latihan 4

Menuliskan struktur dasar HTML dengan penggunaan link untuk menghubungkan halaman eksternal, halaman internal, membuka di tab baru, dan navigasi dalam halaman menggunakan anchor (#).

```

1 <!DOCTYPE html>
2 <html Lang="id">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Belajar link</title>
7 </head>
8 <body>
9   <h1>Contoh Link dalam HTML</h1>
10
11   <!-- Link eksternal -->
12   <h2>Link Eksternal</h2>
13   <p>
14     kunjungan <a href="https://www.google.com">Google</a> untuk mencari informasi
15   </p>
16   <p>
17     Belajar programing di <a href="https://www.w3schools.com">W3Schools</a>
18   </p>
19
20   <!-- Link dengan target blank (buka di tab baru) -->
21   <p>
22     buka <a href="https://github.com" target="_blank">Github</a> di tab baru
23   </p>
24   <!-- Link Internal (ke file lain) -->
25   <h2>Link Internal</h2>
26   <p>kembali ke <a href="#01-struktur-dasar.html">Struktur dasar</a></p>
27   <p>lihat contoh <a href="#02-heading.html">Heading</a></p>
28
29   <!-- Link ke bagian dalam halaman yang sama -->
30   <h2>Link ke Bagian halaman</h2>
31   <p><a href="#bawah">pergi ke bagian bawah</a></p>
32
33   <!-- konten spacer -->
34   <p><lorem ipsum dolor sit amet...</p>
35   <p><lorem ipsum dolor sit amet...</p>
36   <p><lorem ipsum dolor sit amet...</p>
37   <p><lorem ipsum dolor sit amet...</p>
38   <p><lorem ipsum dolor sit amet...</p>
39
40   <!-- Target link internal -->
41   <h2 id="bawah">bagian bawah halaman</h2>
42   <p>ini adalah bagian bawah halaman.</p>
43   <p><a href="#stop">kembali ke atas</a></p>
44 </body>
45 </html>

```

## 5. Latihan 5

Menuliskan strukturr dasar HTML dengan penggunaan tag <img> untuk menampilkan gambar lokal,gambar dari internet,gambar sebagai link,pengaturan ukuran dan menambahkan caption dengan <figure> dan <figcaption>.

```

1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Belajar gambar</title>
7 </head>
8 <body>
9   <h2>contoh gambar dalam html</h2>
10
11   <!-- gambar dengan alt text -->
12   <h3>gambar lokal</h3>
13   
14   <p>gambar diatas adalah contoh gambar lokal dengan lebar 300px</p>
15
16   <!-- gambar dengan ukuran berbeda -->
17   <h3>gambar dengan ukuran berbeda</h3>
18   
19   
20
21   <!-- gambar dari internet -->
22   <h3>gambar dari internet</h3>
23   
28
29   <!-- gambar sebagai link -->
30   <h3>gambar sebagai link</h3>
31   <a href="https://www.google.com">
32     
36   </a>
37
38   <!-- gambar dan figcaption -->
39   <h3>gambar dengan caption</h3>
40   <figure>
41     
46     <figcaption>kucing random</figcaption>
47   </figure>
48
49   <!-- Note: buat folder images jika ingin menggunakan gambar lokal -->
50   <p>
51     <em>
52       Note: untuk gambar lokal buatlah folder "images" di dalam folderr ini dan masukan file gambar {sample.jpg}
53     </em>
54   </p>
55 </body>
56 </html>

```

## 6. Latihan 6

Menuliskan struktur dasar HTML dengan penggunaan berbagai jenis list: unordered list <ul>, ordered list <ol>, nested list, dan description list <dl>.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Belajar list</title>
7 </head>
8 <body>
9   <h1>Contoh list dalam html</h1>
10
11   <!-- unordered list (bullet points) -->
12   <h2>Unordered List (bullet points)</h2>
13   <h3>Bahasa pemrograman populer:</h3>
14   <ul>
15     <li>JavaScript</li>
16     <li>Python</li>
17     <li>Java</li>
18     <li>PHP</li>
19     <li>C++</li>
20   </ul>
21
22   <!-- ordered list (numbering) -->
23   <h2>Ordered List (numbering)</h2>
24   <h3>langkah-langkah membuat website:</h3>
25   <ol>
26     <li>Belajar html</li>
27     <li>belajar css</li>
28     <li>belajar javascript</li>
29     <li>belajar backend (php/node.js)</li>
30     <li>Deploy ke hosting</li>
31   </ol>
32
33   <!-- nested list -->
34   <h2>Nested List(list bersarang)</h2>
35   <h3>teknologi web</h3>
36   <ul>
37     <li>frontend
38       <ul>
39         <li>html</li>
40         <li>css
41           <ol>
42             <li>CSS dasar</li>
43             <li>CSS framework(bootstrap)</li>
44             <li>CSS preprocessor(sass)</li>
45           </ol>
46         <li>javascript</li>
47       </ul>
48     <li>backend
49       <ul>
50         <li>php</li>
51         <li>node.js</li>
52         <li>python</li>
53       </ul>
54     <li>database
55       <ul>
56         <li>mysql</li>
57         <li>postgresql</li>
58         <li>mongodb</li>
59       </ul>
60     <li>
61       <ul>
62         <li>
63           <ol>
64             <li>
65               <ol>
66                 <li>
67                   <ol>
68                     <li>
69                       <ol>
70                         <li>
71                           <ol>
72                             <li>
73                               <ol>
74                                 <li>
75                                   <ol>
76                                     <li>
77                                       <ol>
78                                         <li>
79                                           <ol>
80                                             <li>
81                                               <ol>
82                                                 <li>
83                                                   <ol>
84                                                     <li>
85                                                       <ol>
86                                                         <li>
87                                                           <ol>
88                                                             <li>
89                                                               <ol>
90                                                                 <li>

```

## 7. Latihan 7

Menuliskan struktur dasar HTML dengan penggunaan <div> sebagai container untuk mengelompokkan elemen, membuat layout sederhana, div bersarang, serta penggunaan id dan class.

```

1  <!DOCTYPE html>
2  <html lang="id">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Belajar Div Container</title>
7  </head>
8  <body>
9    <h1>contoh div dan container</h1>
10
11    <!-- div sebagai container -->
12    <h2>div sebagai container</h2>
13    <div>
14      <h3>artikel pertama</h3>
15      <p>ini adalah isi artikel pertama yang di bungkus dalam div</p>
16      <p>div membantu mengelompokkan elemen-elemen yang terkait</p>
17    </div>
18
19    <div>
20      <h3>artikel kedua</h3>
21      <p>ini adalah isi artikel kedua dalam div terpisah</p>
22      <p>setiap div dapat berisi beberapa elemen html</p>
23    </div>
24
25    <!-- div untuk layout -->
26    <h2>div untuk layout sederhana</h2>
27    <div id="page">
28      <div id="top header">
29        <h3>website header</h3>
30        <p>logo dan navigasi biasanya disini</p>
31      </div>
32      <div id="navigation">
33        <h4>Navigation menu</h4>
34        <ul>
35          <li><a href="#home">home</a></li>
36          <li><a href="#products">products</a></li>
37          <li><a href="#services">services</a></li>
38          <li><a href="#contact">contact</a></li>
39        </ul>
40      </div>
41
42      <div id="main-body">
43        <h4>main body content</h4>
44        <p>konten utama website berada di sini.</p>
45      </div>
46
47      <div id="footer">
48        <h4>footer</h4>
49        <p>copyright dan informasi kontak footer</p>
50      </div>
51    </div>
52
53    <!-- div dengan id dan class -->
54    <h2>div dengan id dan class</h2>
55    <div id="header">
56      <h3>header section</h3>
57      <p>div ini memiliki id "header" untuk identifikasi unik</p>
58    </div>
59
60    <div class="content">
61      <h3>content section</h3>
62      <p>div ini memiliki class "content" yang bisa di gunakan berkali</p>
63    </div>
64
65    <div class="content">
66      <h3>content selection lainnya</h3>
67      <p>div ini juga menggunakan class "content" yang sama.</p>
68    </div>
69
70    <!-- div nested -->
71    <h3>nested div(div bersarang)</h3>
72    <div id="wrapper">
73      <h3>wrapper container</h3>
74      <div class="sidebar">
75        <h4>sidebar</h4>
76        <ul>
77          <li><a href="#home">home</a></li>
78          <li><a href="#about">about</a></li>
79          <li><a href="#contact">contact</a></li>
80        </ul>
81      </div>
82      <div class="main-content">
83        <h4>maincontent</h4>
84        <p>ini adalah konten utama yang berada di dalam wrapper</p>
85        <div class="article">
86          <h5>sub article</h5>
87          <p>article kecil di dalam main content</p>
88        </div>
89      </div>
90    </div>
91
92  </body>
93  </html>

```



## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **# 4.1 Hasil Praktikum**

Dalam sesi praktikum pertama mengenai dasar HTML, para mahasiswa menunjukkan kemajuan yang signifikan dalam memahami cara membangun halaman web sederhana. Mereka berhasil menguasai struktur dasar HTML yang terdiri dari beberapa komponen utama, yaitu elemen ``<html>`` sebagai fondasi utama, bagian ``<head>`` untuk menampung informasi meta seperti judul halaman dan link ke file CSS atau JavaScript, serta bagian ``<body>`` yang berisi seluruh konten yang akan terlihat oleh pengunjung website.

Kemampuan mahasiswa dalam mengorganisir konten menggunakan sistem heading juga menunjukkan hasil yang memuaskan. Mereka dapat menerapkan tag ``<h1>`` untuk judul utama halaman, ``<h2>`` untuk bagian-bagian penting, dan ``<h3>`` hingga level yang lebih rendah untuk detail pembahasan. Penggunaan hierarki heading ini membuat struktur informasi menjadi lebih sistematis dan mudah diikuti.

Selain struktur utama, mahasiswa juga berhasil menguasai berbagai elemen HTML fundamental. Mereka dapat menggunakan tag ``<p>`` untuk membuat paragraf yang rapi, ``<ul>`` dan ``<li>`` untuk menyusun daftar yang terorganisir, tag ``<img>`` untuk menyisipkan gambar yang relevan, dan ``<a>`` untuk membuat tautan yang menghubungkan berbagai halaman atau sumber informasi. Kombinasi elemen-elemen ini memungkinkan mereka menciptakan halaman web yang lebih dinamis dan informatif.

Yang tidak kalah penting, mahasiswa mulai memahami konsep fundamental bahwa dalam pengembangan web modern, HTML berfungsi sebagai kerangka atau struktur dasar, CSS berperan dalam mengatur tampilan visual, dan JavaScript menambahkan elemen interaktif. Pemahaman konseptual ini sangat penting untuk pembelajaran selanjutnya.

#### **# 4.2 Analisis dan Pembahasan**

Melihat pencapaian yang telah diraih, tampak jelas bahwa penguasaan HTML sebagai bahasa markup dasar merupakan langkah fundamental yang tidak dapat diabaikan dalam journey pengembangan web. Ketika mahasiswa berhasil membangun struktur yang solid dengan HTML, hal ini akan mempermudah proses pengembangan selanjutnya, baik dalam hal styling dengan CSS maupun penambahan fungsionalitas dengan JavaScript.

Penggunaan heading secara hierarkis terbukti memberikan dampak positif yang signifikan. Selain membuat konten lebih mudah dipahami pembaca, struktur heading yang baik juga mendukung praktik SEO yang optimal, membantu mesin pencari memahami topik dan subtopik dalam halaman web. Ini merupakan skill yang akan sangat berguna dalam pengembangan web profesional.

Penerapan berbagai elemen semantik HTML menunjukkan bahwa mahasiswa mulai memahami pentingnya memilih tag yang tepat sesuai dengan tujuan konten. Paragraf untuk teks naratif, list untuk informasi berurutan, gambar untuk dukungan visual, dan link untuk navigasi - semuanya memiliki fungsi spesifik yang ketika digunakan dengan benar akan meningkatkan user experience secara keseluruhan.

Aspek pembelajaran yang menarik adalah ketika mahasiswa mulai memahami bahwa HTML, CSS, dan JavaScript bukanlah entitas yang berdiri sendiri, melainkan saling terintegrasi. Analogi yang digunakan - HTML sebagai struktur, CSS sebagai penampilan, dan JavaScript sebagai behavior - membantu mereka memvisualisasikan bagaimana ketiga teknologi ini bekerja secara sinergis.

Beberapa kesalahan teknis yang muncul selama praktikum, seperti lupa menutup tag atau salah penulisan atribut, justru menjadi momen pembelajaran yang berharga. Kesalahan-kesalahan ini mengajarkan pentingnya ketelitian dalam coding dan membantu mahasiswa mengembangkan skill debugging sejak dini.

Secara keseluruhan, praktikum ini telah mencapai tujuan utamanya yaitu memberikan fondasi yang kokoh dalam HTML sebelum mahasiswa melanjutkan ke materi yang lebih kompleks. Pemahaman yang solid tentang HTML akan sangat membantu ketika mereka mulai belajar CSS untuk styling dan JavaScript untuk interaktivitas dalam praktikum-praktikum selanjutnya.

## **BAB V**

### **KESIMPULAN**

Praktikum HTML dasar telah berhasil memberikan pemahaman fundamental tentang struktur halaman web. Mahasiswa menguasai elemen utama seperti `<html>`, `<head>`, dan `<body>` sebagai kerangka dasar, serta berbagai komponen penting seperti heading, paragraf, list, gambar, dan link untuk menyusun konten yang terstruktur.

Praktikum ini juga memperkenalkan konsep integrasi HTML-CSS-JavaScript, dimana HTML berperan sebagai struktur, CSS mengatur tampilan, dan JavaScript menambah interaktivitas. Melalui latihan praktis, mahasiswa tidak hanya memahami teori tetapi juga melatih ketelitian dalam coding.

Dengan fondasi HTML yang solid ini, mahasiswa siap melanjutkan pembelajaran ke tingkat yang lebih advanced dengan CSS dan JavaScript.

## Daftar Pustaka

Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. Indianapolis: John Wiley & Sons.

Flanagan, D. (2020). *JavaScript: The Definitive Guide* (7th ed.). Sebastopol: O'Reilly Media.

Freeman, E., & Freeman, E. (2012). *Head First HTML and CSS* (2nd ed.). Sebastopol: O'Reilly Media.

Mozilla Developer Network. (2024). *HTML: HyperText Markup Language*. Diakses pada 10 September 2025, dari <https://developer.mozilla.org/en-US/docs/Web/HTML>

Pilgrim, M. (2010). *HTML5: Up and Running*. Sebastopol: O'Reilly Media.

Robbins, J. N. (2018). *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics* (5th ed.). Sebastopol: O'Reilly Media.

W3Schools. (2024). *HTML Tutorial*. Diakses pada 10 September 2025, dari <https://www.w3schools.com/html/>

World Wide Web Consortium (W3C). (2021). *HTML Living Standard*. Diakses pada 10 September 2025, dari <https://html.spec.whatwg.org/>



# Pemrograman Web

## LAPORAN PRAKTIKUM CSS

Oleh :

*Muhammad Nabel*

*2411102441211*

Teknik Informatika  
Fakultas Sains & Teknologi  
Universitas Muhammadiyah Kalimantan Timur  
Samarinda, 2025

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam era digital saat ini, Cascading Style Sheets atau yang lebih dikenal dengan CSS telah menjadi tulang punggung dari setiap pengembangan website modern. Sebagai bahasa styling yang khusus dirancang untuk mengontrol presentasi visual dari dokumen HTML, CSS memainkan peran yang sangat vital dalam menciptakan pengalaman pengguna yang memukau dan fungsional.

Perjalanan CSS dimulai sejak tahun 1996 ketika World Wide Web Consortium (W3C) memperkenalkannya sebagai standar resmi. Kehadiran CSS pada masa itu merupakan revolusi dalam dunia web development, karena untuk pertama kalinya pengembang bisa memisahkan antara struktur konten dengan aspek visual secara tegas. Sebelumnya, styling dilakukan langsung dalam HTML yang membuat kode menjadi rumit dan sulit untuk dipelihara.

Evolusi CSS dari generasi pertama hingga CSS3 yang kita kenal sekarang menunjukkan bagaimana teknologi ini terus beradaptasi dengan kebutuhan industri. Versi terbaru menawarkan kapabilitas yang jauh melampaui fungsi dasar, mencakup animasi yang halus, transisi yang elegan, transformasi visual yang kompleks, hingga kemampuan responsive design yang dapat menyesuaikan tampilan secara otomatis berdasarkan karakteristik perangkat yang digunakan.

Data dari survei Stack Overflow Developer Survey secara konsisten menunjukkan bahwa CSS berada di posisi lima besar bahasa pemrograman yang paling populer dan banyak digunakan oleh komunitas developer global. Fenomena ini bukan tanpa alasan, mengingat CSS memberikan fleksibilitas luar biasa dalam mengatur setiap aspek visual - mulai dari pemilihan warna yang tepat, pengaturan tipografi yang readable, pengelolaan spacing yang proporsional, hingga orchestration layout yang kompleks.

Tren mobile-first dalam akses internet juga semakin memperkuat posisi CSS sebagai skill wajib. Kemampuan CSS dalam mendukung responsive design memungkinkan sebuah website dapat memberikan user experience yang optimal, baik ketika diakses melalui desktop dengan layar lebar, tablet dengan orientasi landscape/portrait, maupun smartphone dengan berbagai resolusi.

Konteks pendidikan teknologi informasi menempatkan CSS sebagai jembatan yang menghubungkan pembelajaran teoretis dengan implementasi praktis di dunia nyata. Penguasaan CSS yang solid tidak hanya membekali mahasiswa dengan kemampuan membuat website yang estetis, tetapi juga membuka jalan untuk menguasai framework-framework modern seperti Bootstrap yang mempercepat development, Tailwind CSS dengan pendekatan utility-first, atau Material UI yang mengimplementasikan design system Google.

Selain itu, pemahaman mendalam tentang CSS menjadi prerequisite untuk menguasai teknik-teknik advanced seperti CSS Grid yang revolutioner dalam layouting, Flexbox yang memberikan control alignment yang presisi, dan CSS Preprocessor seperti SASS atau SCSS yang meningkatkan maintainability kode. Transformasi dari era table-based layout menuju

sistem modern ini menandai evolution signifikan dalam cara kita membangun antarmuka web yang lebih dinamis dan scalable.

## 1.2 Tujuan Pembelajaran

Praktikum CSS ini dirancang dengan serangkaian objektif pembelajaran yang komprehensif untuk membekali mahasiswa dengan kompetensi yang relevan dengan kebutuhan industri:

- a. Menguasai fundamental CSS sebagai stylesheet language yang bertanggung jawab terhadap visual presentation dari dokumen HTML
- b. Memahami secara mendalam sintaks CSS dan berbagai jenis selector yang memungkinkan targeting elemen HTML dengan presisi tinggi
- c. Mengeksplorasi dan menerapkan properti-properti CSS essential untuk color management, typography styling, spatial arrangement, dan comprehensive layout design
- d. Membangun expertise dalam CSS Box Model yang mencakup interaksi antara margin, border, padding, dan content area
- e. Mengembangkan skills dalam positioning techniques dan floating mechanisms untuk advanced element placement
- f. Mengimplementasikan responsive design methodology menggunakan CSS agar website dapat beradaptasi sempurna dengan berbagai screen sizes dan device capabilities
- g. Menguasai modern layout systems yaitu CSS Grid untuk two-dimensional layouts dan Flexbox untuk one-dimensional arrangements
- h. Mempelajari teknik advanced untuk menciptakan smooth animations, seamless transitions, dan engaging visual effects yang meningkatkan user engagement
- i. Memahami prinsip cascading dan inheritance mechanisms dalam CSS untuk efficient style management
- j. Mengimplementasikan media queries strategy untuk menciptakan truly adaptive web design yang responsive terhadap device characteristics
- k. Menerapkan industry best practices dalam CSS code organization, maintainability, dan scalability

### 1.3 Tinjau Pustaka

CSS atau Cascading Style Sheets merupakan stylesheet language yang didesain khusus untuk mengatur presentasi visual dari dokumen yang dibuat dengan markup language seperti HTML. Konsep fundamental CSS pertama kali dicetuskan oleh Håkon Wium Lie pada tahun 1994, dan kemudian dikembangkan lebih lanjut bersama dengan Bert Bos melalui kolaborasi yang menghasilkan spesifikasi yang robust (Lie & Bos, 2005).

Filosofi desain CSS berfokus pada separation of concerns, dimana konten dan presentasi dipisahkan secara tegas. Meyer (2018) menekankan bahwa pemisahan ini memungkinkan developer untuk membangun design system yang konsisten across multiple pages tanpa perlu melakukan modifikasi pada struktur HTML yang sudah ada.

Perkembangan CSS mengikuti pola iteratif yang terstruktur, dimulai dari CSS Level 1 yang dirilis pada tahun 1996, berkembang menjadi CSS2 dengan fitur positioning yang lebih advanced, hingga CSS3 yang mengadopsi modular approach untuk memfasilitasi development dan implementation yang lebih efisien (Gasston, 2013).

Revolusi layout modern dimulai dengan introduction of Flexbox dan Grid systems. Andrew (2021) mengidentifikasi bahwa CSS Grid memberikan unprecedented control dalam two-dimensional space, memungkinkan precise element positioning baik secara horizontal maupun vertikal. Hal ini menandai paradigm shift dari table-based layouts menuju sistem yang lebih semantic dan flexible.

Research yang dilakukan oleh Johnson et al. (2019) mengungkapkan bahwa adoption CSS preprocessors seperti SASS dan LESS dapat meningkatkan development efficiency hingga 40%. Peningkatan ini dimungkinkan oleh advanced features seperti variable definitions, nested rules, dan mixin functionality yang tidak tersedia dalam vanilla CSS.

Responsive web design telah menjadi industry standard, dengan research dari Frain (2020) menunjukkan bahwa 92% top-tier websites worldwide telah mengimplementasikan media query-based responsive techniques. Hal ini menunjukkan pentingnya adaptive design dalam mengakomodasi diverse device ecosystem.

Performance optimization menjadi concern yang semakin kritikal seiring dengan meningkatnya complexity aplikasi web modern. Powell (2021) mengidentifikasi bahwa techniques seperti CSS minification, critical CSS extraction, dan lazy loading dapat mempercepat page load time hingga 60%, yang berdampak signifikan terhadap user experience dan SEO performance.

Tren terkini dalam CSS development menunjukkan evolusi yang berkelanjutan. Latest report dari W3C (2023) mengindikasikan bahwa emerging features seperti CSS Container Queries, CSS Houdini APIs, dan enhanced CSS Custom Properties level 2 sedang mengubah landscape web development dengan membuka possibilities untuk dynamic dan adaptive user interfaces yang lebih sophisticated.

## **BAB II**

### **Alat dan Bahan**

#### 2.1 Alat

- a. Komputer
- b. Visual Studio Code
- c. Browser

#### 2.2 Bahan

- a. Dokumentasi

## **BAB III**

### **Prosedur Kerja**

#### 1. Latihan 1

Atur gaya dasar pada body untuk memastikan tipografi, spasi, dan latar halaman terlihat rapi dan nyaman dibaca. Gunakan h1 sebagai judul utama yang tampil menonjol dan terpusat, lalu bedakan tingkat judul selanjutnya dengan h2 dan h3 agar hierarki konten lebih jelas. Untuk isi paragraf (p), fokuskan pada keterbacaan, sedangkan daftar (ul dan li) dibuat lebih teratur agar setiap item mudah dipahami. Tambahkan komponen utilitas seperti .highlight untuk menonjolkan konten penting dalam bentuk kartu yang bersih, serta .info-box untuk menampilkan informasi atau notice dengan nuansa informatif. Terapkan kelas tersebut pada elemen yang sesuai setiap kali ingin menekankan informasi tertentu. Setelah itu, tinjau kembali keterbacaan dan konsistensi visual agar tampilan tetap nyaman digunakan di berbagai ukuran layar.



```
1  body {
2    font-family: Arial, sans-serif;
3    line-height: 1.6;
4    margin: 20px;
5    background-color: #f4f4f4;
6  }
7
8
9  h1 {
10   color: #333;
11   text-align: center;
12   background-color: #4CAF50;
13   color: white;
14   padding: 20px;
15   margin: 0 0 20px 0;
16 }
17
18 h2 {
19   color: #2196F3;
20   border-bottom: 2px solid #2196F3;
21   padding-bottom: 5px;
22 }
23
24 h3 {
25   color: #FF9800;
26 }
27
28
29 p {
30   color: #666;
31   font-size: 16px;
32 }
33
34
35 ul {
36   background-color: white;
37   padding: 15px;
38   border-left: 4px solid #4CAF50;
39 }
40
41 li {
42   margin-bottom: 5px;
43 }
44
45
46 .highlight {
47   background-color: white;
48   padding: 20px;
49   margin: 20px 0;
50   border-radius: 5px;
51   box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
52 }
53
54 .info-box {
55   background-color: #e7f3ff;
56   border: 1px solid #2196F3;
57   padding: 15px;
58   margin: 20px 0;
59   border-radius: 5px;
60 }
61
```

## 2. Latihan 2

Mulailah dengan mengatur gaya dasar melalui body, lalu tempatkan judul utama menggunakan h1 di bagian tengah. Untuk paragraf (p), pastikan tampilannya nyaman dibaca. Sediakan blok khusus seperti .highlight untuk menekankan konten penting, serta .warning sebagai elemen peringatan yang otomatis menampilkan ikon dengan bantuan ::after. Tambahkan pula ikon pada kontainer dengan memanfaatkan .container::before.

Untuk navigasi, tata menu bertingkat menggunakan .menu > li dan .menu li li agar struktur lebih jelas. Kelola perilaku tautan pada .links a dengan state :hover, :visited, dan :active, serta siapkan gaya khusus untuk tombol melalui button[type="button"] berikut efek :hover. Gunakan elemen h4, h5, dan h6 sebagai subjudul, dan aktifkan aturan \* { box-sizing: border-box } untuk mempermudah pengaturan layout.

Berikan gaya tersendiri pada input sesuai tipenya, misalnya input[type="text"], input[type="email"], dan input[type="password"]. Sediakan utilitas .border untuk elemen dengan batasan visual, serta #special untuk komponen khusus. Terakhir, kelola hierarki dan keterbacaan konten di dalam .container, dengan mengatur tampilan pada .container h2, .container p, serta paragraf dalam .container .inner p agar konsistensi dan kenyamanan visual tetap terjaga.

```

1 body {
2   font-family: Arial, sans-serif;
3   margin: 20px;
4   background-color: #f9f9f9;
5 }
6
7 h1 {
8   color: #333;
9   text-align: center;
10 }
11
12 p {
13   color: #666;
14   line-height: 1.6;
15 }
16
17
18 .highlight {
19   background-color: #ffffcc;
20   padding: 10px;
21   border-left: 4px solid #ffc107;
22   margin: 10px 0;
23 }
24
25 .warning {
26   background-color: #f8d7da;
27   color: #212121;
28   padding: 10px;
29   border: 1px solid #f5c6cb;
30   border-radius: 4px;
31 }
32
33
34 .menu > li {
35   background-color: #e0e0e0;
36   padding: 5px;
37   margin: 5px 0;
38   list-style-type: none;
39 }
40
41 .menu li li {
42   background-color: #f8f9fa;
43   margin-left: 20px;
44   font-size: 14px;
45 }
46
47 .links a {
48   color: #007bff;
49   text-decoration: none;
50   padding: 5px 10px;
51   display: inline-block;
52   margin: 5px;
53   border: 1px solid transparent;
54 }
55
56 .links a:hover {
57   background-color: #007bff;
58   color: white;
59   border-color: #007bff;
60 }
61
62 button[type="button"] {
63   background-color: #ffc107;
64   color: #212121;
65   border: none;
66   padding: 10px 20px;
67   margin: 5px;
68   cursor: pointer;
69 }
70
71 button[type="button"]:hover {
72   background-color: #eda800;
73 }
74
75 h4,
76 h5,
77 h6 {
78   color: #f0f0f0;
79   font-family: Georgia, serif;
80   margin: 15px 0 5px 0;
81 }
82
83
84 * {
85   box-sizing: border-box;
86 }
87
88
89 .container::before {
90   content: "● ";
91   font-size: 20px;
92 }
93
94
95 .warning::after {
96   content: "▲";
97 }
98
99 .links a:visited {
100  color: #f42c2c;
101 }
102
103 .links a:active {
104  background-color: #0056b3;
105 }
106
107
108 input[type="text"] {
109   border: 2px solid #28a745;
110   padding: 8px;
111   margin: 5px;
112 }
113
114
115 input[type="email"] {
116   border: 2px solid #17a2b8;
117   padding: 8px;
118   margin: 5px;
119 }
120
121
122 input[type="password"] {
123   border: 2px solid #dc3545;
124   padding: 8px;
125   margin: 5px;
126 }
127
128
129 .border {
130   border: 2px solid #007bff;
131   border-radius: 5px;
132 }
133
134
135 #speciel {
136   background-color: #d1ecf1;
137   color: #0056b3;
138   font-weight: bold;
139   padding: 10px;
140   border: 2px solid #17a2b8;
141 }
142
143
144 .container h2 {
145   color: #28a745;
146   border-bottom: 2px solid #28a745;
147 }
148
149
150 .container p {
151   color: #495057;
152   margin-left: 20px;
153 }
154
155
156 .container .inner p {
157   color: #f42c2c;
158   font-style: italic;
159   margin-left: 40px;
160 }

```

### 3. Latihan 3

Mulailah dengan mengatur tampilan dasar pada body, termasuk font, margin, serta warna teks dan latar agar halaman terlihat bersih. Gunakan h1 di posisi tengah dengan efek gradasi untuk judul utama, lalu tampilkan h2 dengan garis bawah agar subjudul lebih menonjol. Bungkus setiap blok konten di dalam .section supaya tampak rapi seperti sebuah kartu.

Untuk dekorasi teks, siapkan kelas utilitas seperti .decoration-none, .decoration-underline, .decoration-overline, .decoration-through, dan .decoration-multiple. Untuk pengaturan huruf, gunakan .transform-none, .transform-uppercase, .transform-lowercase, atau .transformcapitalize. Atur perataan teks menggunakan .align-left, .align-center, .align-right, maupun .alignjustify.

Pemilihan keluarga font bisa dilakukan lewat .font-serif, .font-sans, .font-mono, atau .fontcustom (dengan memastikan @import Roboto aktif). Ukuran teks diatur melalui .size-small, .size-normal, .size-large, .size-xlarge, serta opsi berbasis em atau rem. Bobot huruf tersedia dalam .weight-thin, .weight-normal, .weight-bold, .weight-bolder, hingga .weight-900. Untuk gaya teks, gunakan .style-normal, .style-italic, atau .style-oblique.

Kendalikan jarak baris melalui .line-small, .line-normal, dan .line-large. Atur jarak huruf dengan .ls-tight, .ls-normal, atau .ls-wide, sementara jarak antar kata dapat diubah dengan .ws-normal maupun .ws-wide. Tambahkan pula penekanan visual menggunakan bayangan teks dengan kelas .shadow-simple, .shadow-colored, .shadow-multiple, atau .shadow-glow sesuai kebutuhan desain.

```

1  @import url('https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;700&display=swap');
2
3
4  body {
5      font-family: Arial, sans-serif;
6      margin: 20px;
7      background: #f5f5f5;
8      color: #333;
9  }
10
11  h1 {
12      text-align: center;
13      font-size: 2.5em;
14      margin-bottom: 30px;
15      background: linear-gradient(45deg, #3498db, #2c3e50);
16      background-clip: text;
17      -webkit-background-clip: text;
18      -webkit-text-fill-color: transparent;
19  }
20
21  h2 {
22      color: #34495e;
23      border-bottom: 2px solid #3498db;
24      margin-bottom: 15px;
25      padding-bottom: 5px;
26  }
27
28  .section {
29      background: #fff;
30      padding: 20px;
31      margin: 20px 0;
32      border-radius: 8px;
33      box-shadow: 0 2px 4px rgba(0,0,0,0.1);
34  }
35
36
37  .decoration-none { text-decoration: none; }
38  .decoration-underline { text-decoration: underline; text-decoration-color: #3498db; }
39  .decoration-overline { text-decoration: overline; text-decoration-color: #e74c3c; }
40  .decoration-through { text-decoration: line-through; text-decoration-color: #95a5a6; }
41  .decoration-multiple { text-decoration: underline overline; text-decoration-color: #9b59b6; }
42
43
44  .transform-none { text-transform: none; }
45  .transform-uppercase { text-transform: uppercase; color: #e74c3c; font-weight: bold; }
46  .transform-lowercase { text-transform: lowercase; color: #3498db; }
47  .transform-capitalize { text-transform: capitalize; color: #27ae60; font-weight: 600; }
48
49
50  .align-left { text-align: left; }
51  .align-center { text-align: center; background: #ecf0f1; padding: 10px; }
52  .align-right { text-align: right; background: #fde7a5; padding: 10px; }
53  .align-justify { text-align: justify; background: #e8f5e9; padding: 15px; border-left: 4px solid #27ae60; }
54
55
56  .font-serif { font-family: "Times New Roman", serif; }
57  .font-sans { font-family: Arial, sans-serif; }
58  .font-mono { font-family: "Courier New", monospace; background: #f8f9fa; padding: 5px; }
59  .font-custom { font-family: "Roboto", sans-serif; color: #2473cc; }
60
61
62  .size-small { font-size: 12px; }
63  .size-normal { font-size: 16px; }
64  .size-large { font-size: 24px; }
65  .size-xlarge { font-size: 32px; }
66  .size-em { font-size: 1.5em; }
67  .size-rem { font-size: 1.2rem; }
68
69
70  .weight-thin { font-weight: 100; }
71  .weight-normal { font-weight: 400; }
72  .weight-bold { font-weight: 700; }
73  .weight-bolder { font-weight: bolder; }
74  .weight-900 { font-weight: 900; }
75
76
77  .style-normal { font-style: normal; }
78  .style-italic { font-style: italic; }
79  .style-oblique { font-style: oblique; }
80
81
82  .line-small { line-height: 1.2; background: #ffebee; padding: 10px; }
83  .line-normal { line-height: 1.6; background: #e3f2fd; padding: 10px; }
84  .line-large { line-height: 2; background: #f3e5f5; padding: 10px; }
85
86
87  .ls-tight { letter-spacing: -1px; font-weight: bold; }
88  .ls-normal { letter-spacing: normal; }
89  .ls-wide { letter-spacing: 3px; text-transform: uppercase; font-weight: bold; color: #8e44ad; }
90
91
92  .ws-normal { word-spacing: normal; }
93  .ws-wide { word-spacing: 10px; background: #fff3e0; padding: 10px; }
94
95
96  .shadow-simple { text-shadow: 2px 2px 4px rgba(0,0,0,.3); font-size: 20px; font-weight: bold; }
97  .shadow-colored { text-shadow: 3px 3px 0 #3498db, 6px 6px 0 #2c3e50; font-size: 18px; font-weight: bold; }
98  .shadow-multiple { text-shadow: 1px 1px 0 #e74c3c, 2px 2px 0 #f39c12, 3px 3px 0 #27ae60, 4px 4px 0 #2980b9; font-size: 24px; font-weight: bold; }
99  .shadow-glow { text-shadow: 0 5px #3498db, 0 0 10px #3498db, 0 20px #3498db; color: #2c3e50; font-size: 22px; font-weight: bold; text-align: center; }

```

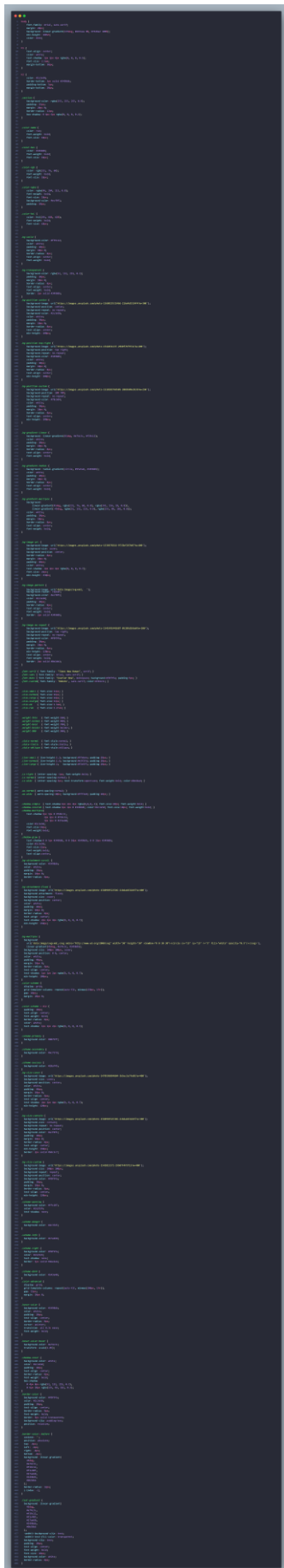
#### 4. Latihan 4

Mulailah dengan mengatur tampilan dasar lewat body menggunakan latar gradasi, tinggi minimum 100vh, margin yang rapi, dan warna teks yang serasi. Judul utama (h1) ditempatkan di tengah dengan teks putih serta efek bayangan, sedangkan subjudul (h2) diberi aksentasi garis bawah. Bungkus setiap konten dalam `.section` agar tampilannya menyerupai kartu

Untuk pengaturan warna teks, gunakan kelas `.color-name`, `.color-hex`, `.color-rgb`, `.color-rgba`, atau `.color-hsl`. Latar belakang dapat dikelola dengan `.bg-solid` dan `.bg-transparent`, lalu atur posisi gambar melalui `.bg-position-center`, `.bg-position-top-right`, atau `.bg-position-custom`. Efek gradasi tersedia lewat `.bg-gradient-linear`, `.bg-gradient-radial`, dan `.bg-gradient-multiple`. Variasi gambar bisa ditampilkan dengan `.bg-image-url`, pola berulang menggunakan `.bg-image-pattern`, atau tanpa pengulangan dengan `.bg-image-no-repeat`. Untuk perilaku latar, gunakan `.bgattachment-scroll` atau `.bgattachment-fixed`, tampilkan beberapa lapisan dengan `.bg-multiple`, dan kendalikan skalanya lewat `.bg-size-cover`, `.bg-size-contain`, atau `.bg-size-custom`.

Tipografi dapat diatur dengan `.font-serif`, `.font-sans`, `.font-mono`, atau `.font-custom`. Ukuran teks tersedia lewat `.size-small`, `.size-normal`, `.size-large`, `.size-xlarge`, serta opsi berbasis em dan rem. Ketebalan huruf bisa dipilih dari `.weight-thin`, `.weight-normal`, `.weight-bold`, `.weight-bolder`, hingga `.weight-900`. Untuk gaya teks, gunakan `.style-normal`, `.style-italic`, atau `.style-oblique`. Spasi baris diatur lewat `.line-small`, `.line-normal`, dan `.line-large`. Jarak huruf bisa diubah dengan `.ls-tight`, `.ls-normal`, atau `.ls-wide`, sedangkan jarak kata dikontrol lewat `.ws-normal` dan `.ws-wide`. Tambahkan penekanan visual melalui bayangan teks dengan kelas `.shadow-simple`, `.shadow-colored`, `.shadow-multiple`, atau `.shadow-glow`

Sebagai palet siap pakai, gunakan grid `.color-scheme` dengan varian `.scheme-primary`, `.scheme-secondary`, `.scheme-success`, `.scheme-warning`, `.scheme-danger`, `.scheme-info`, `.scheme-light`, dan `.scheme-dark`. Untuk tata letak lanjutan, manfaatkan `.color-advanced`. Tambahkan interaksi melalui `.hover-color` yang membuat teks berubah warna dan sedikit membesar saat hover. Efek visual lainnya mencakup bayangan berwarna lewat `.shadow-color`, bingkai gradasi menggunakan `.border-color (::before)`, serta teks gradasi melalui `.text-gradient`.



## 5. Latihan 5

Mulailah dengan mengatur tampilan dasar pada body, meliputi tipografi, margin, latar bernuansa netral, dan line-height agar konten nyaman dibaca. Judul utama (h1) ditampilkan dengan efek teks gradasi yang terklip, subjudul (h2) menggunakan garis bawah sebagai penekanan, sementara h3 berfungsi sebagai aksent visual. Untuk membungkus konten, gunakan `.section` agar tampil rapi menyerupai kartu.

Untuk mendemonstrasikan box model, manfaatkan `.box-model-demo` dengan elemen `.margin-demo`, `.border-demo`, `.padding-demo`, dan `.content-demo`, serta sertakan label pendukung melalui `.box-labels` seperti `.label-margin`, `.label-border`, `.label-padding`, dan `.label-content`. Tambahkan utilitas jarak menggunakan `.margin-all` atau `.margin-auto`, pengaturan ruang dalam melalui `.padding-all` maupun `.padding-individual`, serta variasi border dengan `.border-solid`, `.borderdashed`, `.border-dotted`, dan `.border-double`.

Gunakan komponen `.card` yang terdiri dari `.card-header` dan `.card-body` untuk konten terstruktur.

Untuk tombol, letakkan di dalam `.button-demo`, lalu terapkan gaya varian `.btn-primary`, `.btn-outline`, dan `.btn-rounded`, masing-masing dilengkapi efek saat `:hover`. Demonstrasikan bentuk sudut dengan grid `.border-radius-demo`, yang memuat `.radius-small`, `.radius-large`, dan `.radiuscircle`.

Bandingkan perilaku `box-sizing` dengan `.content-box` (menggunakan `content-box`) dan `.borderbox` (menggunakan `border-box`). Susun tata letak dengan `.layout-demo` yang berisi `.sidebar` (lengkap dengan `h4`, `ul`, dan `li`) serta `.main-content`. Terakhir, tampilkan area inspeksi menggunakan `.inspector-demo`, dengan elemen interaktif `.inspect-me` yang berubah saat disentuh `hover` untuk memberikan umpan balik visual.





## 6. Latihan 6

Mulailah dengan mengatur tampilan dasar pada body, meliputi font, margin, tinggi minimum layar penuh, latar belakang gradasi, serta line-height agar halaman terlihat rapi dan mudah dibaca. Untuk judul, gunakan h1 dengan teks putih dan bayangan, h2 diberi garis bawah berwarna biru, dan h3 dijadikan aksent dengan warna merah. Bungkus setiap blok konten di dalam section supaya tampil menyerupai kartu.

Untuk mendemonstrasikan perilaku display, gunakan kelas `.display-block`, `.display-inline`, `.display-inline-block`, dan `.display-none`. Ilustrasikan penempatan elemen dengan `.position-container` serta varian: `.position-static`, `.position-relative` (dilengkapi offset), `.position-relativeparent`, `.position-absolute` (menempel di pojok kanan atas), `.position-fixed` (tetap di viewport), serta lingkungan scroll `.position-container-2` untuk elemen `.position-sticky` yang menempel di bagian atas.

Pengaturan float dan overflow ditampilkan lewat `.float-container` bersama kelas `.overflow-visible`, `.overflow-hidden`, `.overflow-scroll`, dan `.overflow-auto`. Elemen mengapung dapat menggunakan `.float-left` atau `.float-right`, sementara penetralnya dengan `.clear-float`. Untuk tata letak fleksibel, gunakan `.flex-container` dengan item `.flex-item`, ditambah variasi `.flex-column`, `.justify-center`, dan `.justify-between`.

Grid sederhana bisa dibuat dengan `.two-column` dan `.three-column`, berisi blok `.column`, `.column-left`, dan `.column-right`, lengkap dengan pewarnaan otomatis melalui selektor `:nth-child`. Tumpukan lapisan ditunjukkan lewat `.z-index-container` dengan elemen `.z-index-1`, `.z-index-2`, dan `.z-index-3`. Uji pemotongan konten menggunakan `.overflow-container` berisi `.overflow-content`

Siapkan juga kerangka halaman melalui `.page-layout` yang mencakup `.page-header`, `.page-content`, dan `.page-footer`. Responsivitas dijaga dengan media query `@media (max-width: 768px)`, yang mengubah grid `.two-column` dan `.three-column` menjadi susunan vertikal, mengatur `.position-fixed` agar kembali statis, serta menyesuaikan margin pada body.

Siapkan juga kerangka halaman melalui `.page-layout` yang mencakup `.page-header`, `.page-content`, dan `.page-footer`. Responsivitas dijaga dengan media query `@media (max-width: 768px)`, yang mengubah grid `.two-column` dan `.three-column` menjadi susunan vertikal, mengatur `.position-fixed` agar kembali statis, serta menyesuaikan margin pada body

```

1 body {
2   font-family: Arial, sans-serif;
3   margin: 20px;
4   min-height: 100vh;
5   background: linear-gradient(135deg, #0070c0, #0070c0);
6   color: #fff;
7   line-height: 1.6;
8 }
9
10 h1 {
11   text-align: center;
12   font-size: 2.5em;
13   color: #fff;
14   margin-bottom: 30px;
15   text-shadow: 2px 2px 4px rgba(0,0,0,0.5);
16 }
17
18 h2 {
19   color: #2c3e50;
20   border-bottom: 3px solid #3498db;
21   margin-bottom: 20px;
22   padding-bottom: 5px;
23 }
24
25 h3 { color: #0070c0; margin: 20px 0 10px; }
26
27 section {
28   background: rgba(25,25,25,0.95);
29   padding: 20px;
30   margin: 20px 0;
31   border-radius: 12px;
32   box-shadow: 0 4px 8px rgba(0,0,0,0.1);
33 }
34
35 .display-block {
36   display: block;
37   display: inline-block;
38   display: inline-block;
39   display: none {
40     color: #fff; font-weight: bold; border-radius: 6px; margin: 5px;
41   }
42 }
43 .display-block {
44   display: block; background: #3498db; padding: 10px; text-align: center;
45 }
46 .display-inline {
47   display: inline; background: #0070c0; padding: 5px 10px;
48 }
49 .display-inline-block {
50   display: inline-block; background: #f39c12;
51   padding: 15px; width: 100px; text-align: center;
52 }
53 .display-none { display: none; background: #95a5a6; }
54
55 .position-container {
56   position: relative; height: 150px; margin: 15px 0;
57   border: 2px solid #0070c0; border-radius: 8px; background: #ecf0f1;
58 }
59 .position-static {
60   position: static;
61 }
62 .position-relative {
63   position: relative;
64 }
65 .position-absolute {
66   position: absolute;
67 }
68 .position-fixed {
69   position: fixed;
70   color: #fff; font-weight: bold; border-radius: 5px; padding: 10px;
71 }
72 .position-sticky {
73   position: sticky; background: #3498db; margin: 10px;
74 }
75 .position-relative-parent { position: relative; }
76 .position-absolute { background: #f39c12; position: absolute; top: 20px; right: 20px; width: 100px; }
77 .position-fixed { background: #0070c0; position: fixed; top: 20px; right: 20px; z-index: 1000; }
78
79 .position-container-2 {
80   height: 200px; overflow: scroll; border: 2px solid #0070c0;
81   border-radius: 8px; background: #ecf0f1;
82 }
83 .position-sticky { background: #2ecc71; position: sticky; top: 0; }
84
85 .float-container {
86   background: #f8d7da; padding: 15px; border: 1px solid #0070c0;
87   overflow: auto; border-radius: 8px;
88 }
89 .overflow-visible { overflow: visible; }
90 .overflow-hidden { overflow: hidden; }
91 .overflow-scroll { overflow: scroll; }
92 .overflow-auto { overflow: auto; }
93
94 .flex-container {
95   display: flex; gap: 10px; background: #f8d7da;
96   border: 2px solid #0070c0; border-radius: 8px; padding: 10px; margin: 10px 0;
97 }
98 .flex-item {
99   flex: 1; padding: 10px; text-align: center;
100   border-radius: 5px; font-weight: bold; background: #0070c0; color: #fff;
101 }
102
103 .flex-column { flex-direction: column; }
104 .justify-center { justify-content: center; }
105 .justify-between { justify-content: space-between; }
106
107 .two-column, .three-column {
108   display: flex; gap: 10px; margin: 20px 0;
109 }
110
111 .float-left, .float-right {
112   float: left; width: 100px; margin: 0 10px 10px 0;
113   padding: 10px; color: #fff; font-weight: bold; border-radius: 8px; text-align: center;
114 }
115 .float-right { float: right; background: #0070c0; margin: 0 0 10px 10px; }
116 .float-left { background: #0070c0; }
117 .clear-float { clear: both; background: #0070c0; color: #fff; padding: 10px; }
118
119 .z-index-container {
120   position: relative; height: 200px; border: 2px solid #0070c0;
121   margin: 15px 0; border-radius: 8px; background: #f8d7da;
122 }
123 .z-index-1, .z-index-2, .z-index-3 {
124   position: absolute; width: 100px; padding: 20px;
125   border-radius: 8px; color: #fff; font-weight: bold; text-align: center;
126 }
127 .z-index-1 { background: #0070c0; top: 20px; left: 20px; z-index: 1; }
128 .z-index-2 { background: #0070c0; top: 40px; left: 40px; z-index: 2; }
129 .z-index-3 { background: #0070c0; top: 60px; left: 60px; z-index: 3; }
130
131 .overflow-container {
132   width: 200px; height: 100px; border: 2px solid #0070c0;
133   margin: 10px 0; border-radius: 8px; background: #f8d7da;
134 }
135 .overflow-content {
136   width: 150px; height: 120px; padding: 10px; background: #f39c12;
137   font-weight: bold; color: #0070c0;
138 }
139 .column, .column-left, .column-right {
140   flex: 1; padding: 20px; border-radius: 8px;
141   border: 1px solid #0070c0; background: #fff;
142 }
143 .column-left { background: #0070c0; }
144 .column-right { background: #0070c0; }
145 .column-child1 { background: #f8d7da; }
146 .column-child2 { background: #f8d7da; }
147 .column-child3 { background: #f8d7da; }
148
149 .page-layout {
150   background: #fff; border-radius: 8px; border: 1px solid #0070c0; overflow: hidden;
151 }
152 .page-header, .page-footer {
153   color: #fff; text-align: center; padding: 10px;
154 }
155 .page-header { background: #0070c0; }
156 .page-footer { background: #0070c0; font-size: 14px; }
157 .page-content { background: #f8d7da; padding: 10px; }
158
159 @media (max-width: 768px) {
160   .two-column, .three-column { flex-direction: column; }
161   .position-fixed { position: static; margin: 10px 0; }
162   body { margin: 10px; }
163 }

```

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### Hasil Praktikum

Melalui serangkaian aktivitas praktikum yang telah dilakukan, berhasil diciptakan sebuah halaman web profile yang mendemonstrasikan implementasi CSS secara komprehensif. Website yang dihasilkan menunjukkan transformasi visual yang signifikan dibandingkan dengan struktur HTML sederhana, dimana CSS berperan sebagai engine utama dalam mengatur aesthetic dan functional aspects.

Implementasi CSS pada project ini mencakup berbagai aspek fundamental styling. Manipulasi color scheme dilakukan untuk menciptakan visual harmony yang menarik, dimana pemilihan warna background, text, dan accent colors disesuaikan dengan prinsip design yang modern. Typography treatment juga mendapat perhatian khusus dengan pengaturan font-family, font-size, font-weight, dan line-height untuk meningkatkan readability dan visual hierarchy.

Layout arrangement menjadi salah satu achievement utama dalam praktikum ini. Melalui strategic spacing management menggunakan margin dan padding properties, setiap elemen mendapat breathing space yang proporsional. Hal ini menghasilkan clean layout yang tidak cramped serta memudahkan user dalam consuming information yang disajikan.

Eksplorasi selector mechanisms menunjukkan hasil yang memuaskan dalam targeting specific elements. Element selectors digunakan untuk establishing baseline styles, sementara class selectors memberikan flexibility dalam applying reusable styles across multiple elements. ID selectors terbukti efektif untuk styling unique components yang memerlukan special treatment.

Konsep box model visualization menjadi highlight tersendiri dalam practical exercise ini. Melalui careful manipulation border styles, strategic padding implementation, dan thoughtful margin adjustments, setiap content block memiliki defined boundaries yang jelas. Visual demonstration ini membantu dalam understanding spatial relationships antar elemen dan bagaimana mereka saling berinteraksi dalam layout structure.

#### Analisis dan Pembahasan

Berdasarkan observasi terhadap outcomes yang telah dicapai, terbukti bahwa CSS memiliki impact yang fundamental dalam menentukan user experience melalui visual presentation. Salah satu discovery menarik adalah manifestation dari CSS cascading mechanism, yang menunjukkan hierarchy priorities ketika multiple styling rules diterapkan pada target element yang sama.

Cascade priority system mendemonstrasikan logical order dimana inline styles memperoleh highest specificity weight, diikuti oleh ID selectors, class selectors, dan terakhir element selectors. Understanding terhadap specificity calculation ini crucial untuk avoiding styling conflicts dan ensuring predictable rendering results. Dalam practical implementation, fenomena ini terlihat jelas ketika attempting override default styles dengan custom rules.

Selector strategy yang diterapkan dalam praktikum membuktikan effectiveness dalam achieving targeted styling objectives. Universal element selectors terbukti ideal untuk establishing global baseline styles seperti typography defaults dan color schemes yang consistent across entire page. Sebaliknya, class-based selectors memberikan reusability advantage, memungkinkan application consistent styling patterns pada multiple elements tanpa code duplication.

ID selectors menunjukkan utility dalam handling unique page components yang memerlukan distinctive visual treatment. Pendekatan ini particularly valuable untuk landmark elements seperti navigation bars, hero sections, atau footer areas yang memiliki specific design requirements.

Box model implementation menghasilkan insights mendalam tentang spatial design principles. Margin properties terbukti effective dalam controlling external spacing dan creating visual separation antar elements, while padding memberikan internal breathing room yang meningkatkan content legibility. Border styling bukan hanya berfungsi sebagai visual delimiter, tetapi juga sebagai design element yang dapat enhance aesthetic appeal.

Content area optimization melalui width dan height adjustments memungkinkan precise control terhadap element dimensions, resulting dalam proportional layout yang balanced. Integration keempat box model components (margin, border, padding, content) menghasilkan cohesive design system yang scalable dan maintainable.

Secara holistic, praktikum ini menegaskan fundamental principle bahwa CSS serves sebagai critical layer dalam modern web development architecture. Separation of concerns antara content structure (HTML) dan visual presentation (CSS) tidak hanya meningkatkan code organization, tetapi juga memfasilitasi easier maintenance dan consistent design implementation across complex web projects.

Achievement ini memberikan strong foundation untuk advancing ke more sophisticated CSS techniques seperti responsive design, grid systems, dan modern layout methodologies yang essential dalam contemporary web development landscape.

## **BAB V KESIMPULAN**

Praktikum CSS berhasil membuktikan peran vital styling dalam pengembangan web modern. Mahasiswa menguasai tiga konsep fundamental: cascade untuk menentukan prioritas aturan, selector untuk targeting elemen spesifik, dan box model untuk mengatur spatial layout.

CSS terbukti lebih dari sekadar tool estetika, melainkan foundation untuk menciptakan web structure yang organized, consistent, dan user-friendly. Pemahaman ini menjadi bekal solid untuk eksplorasi teknik CSS advanced selanjutnya.

## **DAFTAR PUSTAKA**

Powell, T. A. (2021). *Web Performance Optimization: CSS and JavaScript Best Practices*. San Diego: Academic Press.

W3C CSS Working Group. (2023). *CSS Snapshot 2023*. Diakses pada 10 September 2025, dari <https://www.w3.org/TR/CSS/>

W3Schools. (2024). *CSS Tutorial*. Diakses pada 10 September 2025, dari <https://www.w3schools.com/css/>

World Wide Web Consortium (W3C). (2024). *Cascading Style Sheets Level 3*. Diakses pada 10 September 2025, dari <https://www.w3.org/Style/CSS/>

Github: <https://github.com/mhmmdnabiel24/Pemrograman-web.git>