

HYPERTEXT MARKUP LANGUAGE (HTML)

Mata Kuliah: Pemrograman Web

Materi Praktikum Ke: 1

Nama: Rivoldy Adrian Prawira Makatita

NIM: 2411102441207

Tanggal Praktikum: Selasa, 2 September 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 Tinjauan 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>`, `<footer>`, `<article>`, 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

3.1 Latihan 1

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

```
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>Struktur HTML Dasar</title>
7 </head>
8 <body>
9   <h1>Selamat Datang di HTML!</h1>
10  <p>ini adalah halaman web pertama saya</p>
11 </body>
12 </html>
```

3.2 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="en">
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   <h1>ini adalah Heading 1</h1>
10  <h2>ini adalah Heading 2</h2>
11  <h3>ini adalah Heading 3</h3>
12  <h4>ini adalah Heading 4</h4>
13  <h5>ini adalah Heading 5</h5>
14  <h6>ini adalah Heading 6</h6>
15
16  <h1>Bab 1 Pengnalan Web</h1>
17  <h2>1.1 Apa itu HTML</h2>
18  <h3>Definisi HTML</h3>
19  <h3>Sejarah HTML</h3>
20  <h2>1.2 mengapa belajar HTML</h2>
21 </body>
22 </html>
```

3.3 Latihan 3

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

```

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 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 markup yang digunakan untuk membuat halaman web.
14   </p>
15   <p>
16     ini adalah paragraf kedua, setiap paragraf akan otomatis diberi jarak oleh browser.
17   </p>
18   <!-- Paragraf dengan Text Formatting -->
19   <p>
20     Dalam paragraf ini kita bisa menggunakan <strong>text tebal (bold)</strong> dan <em>text miring (italic)</em>.
21   </p>
22   <p>
23     Kita juga bisa membuat text <u>garis bawah (underline)</u> dan <mark>highlight</mark>.
24   </p>
25   <!-- Break Line -->
26   <p>Baris pertama <br />Baris kedua dengan line break</p>
27   <!-- Horizontal Rule -->
28   <hr />
29   <p>Garis horizontal diatas dibuat dengan tag hr</p>
30 </body>
31 </html>

```

3.4 Latihan 4

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

```

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 Link</title>
7 </head>
8 <body>
9   <h1>Contoh Link dalam HTML</h1>
10
11   <h2>Link Eksternal</h2>
12   <p>
13     kunjungi <a href="https://www.google.com">Google</a> untuk mencari informasi.
14   </p>
15   <p>Belajar programming di <a href="https://www.w3schools.com">W3schools</a>.</p>
16   <p>
17     Buka <a href="https://github.com" target="_blank">Github</a> di tab baru.
18   </p>
19
20   <h2>Link Internal</h2>
21   <p>Kembali ke <a href="latihan-1.html">Struktur Dasar</a></p>
22   <p>Lihat contoh <a href="latihan-2.html">Heading</a></p>
23
24   <p>Lorem Ipsum Dolor Sit Amet....</p>
25   <p>Lorem Ipsum Dolor Sit Amet....</p>
26   <p>Lorem Ipsum Dolor Sit Amet....</p>
27   <p>Lorem Ipsum Dolor Sit Amet....</p>
28   <p>Lorem Ipsum Dolor Sit Amet....</p>
29
30   <h2 id="bawah">Bagian Bawah</h2>
31   <p>ini adalah bagian bawah halaman</p>
32   <p><a href="#top">kembali ke Atas</a></p>
33 </body>
34 </html>

```

3.5 Latihan 5

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

```

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 Gambar</title>
7 </head>
8 <body>
9   <h1>Contoh gambar dalam HTML</h1>
10
11   <h2>Gambar Lokal</h2>
12   
13   <p>Gambar diatas adalah contoh gambar lokal dengan lebar 300px.</p>
14
15   <h2>Gambar dengan Ukuran Berbeda</h2>
16   
17   
18
19   <h2>Gambar dari Internet</h2>
20   
25
26   <h2>Gambar sebagai Link</h2>
27   <a href="https://www.google.com">
28     
32   </a>
33
34   <h2>Gambar dengan caption</h2>
35   <figure>
36     
40     <figcaption>Pemandangan matahari terbenam yang indah</figcaption>
41   </figure>
42
43   <p><em>Note: untuk gambar lokal, buatlah folder "Images" didalam folder ini dan masukan file gambar (sample.jpg)</em></p>
44 </body>
45 </html>

```

3.6 Latihan 6

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

```

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 List</title>
7  </head>
8  <body>
9    <h1>Contoh List dalam HTML</h1>
10
11
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
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
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>
47          <li>JavaScript</li>
48        </ul>
49      </li>
50      <li>Backend
51        <ul>
52          <li>PHP</li>
53          <li>Node.js</li>
54          <li>Python</li>
55        </ul>
56      </li>
57      <li>Database
58        <ul>
59          <li>MySQL</li>
60          <li>PostgreSQL</li>
61          <li>MongoDB</li>
62        </ul>
63      </li>
64    </ul>
65
66
67    <h2>Description List</h2>
68    <h3>Istilah dalam Web Development:</h3>
69    <dl>
70      <dt>HTML</dt>
71      <dd>
72        HyperText Markup Language - bahasa markup untuk membuat struktur web
73      </dd>
74
75      <dt>CSS</dt>
76      <dd>Cascading Style Sheets - bahasa untuk styling dan layout web</dd>
77
78      <dt>JavaScript</dt>
79      <dd>Bahasa pemrograman untuk membuat interaktivitas di web</dd>
80
81      <dt>Responsive Design</dt>
82      <dd>
83        Desain web yang dapat menyesuaikan tampilan di berbagai ukuran layar
84      </dd>
85    </dl>
86  </body>
87 </html>

```

3.7 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
12    <h2>Div sebagai Container</h2>
13    <div>
14      <h3>Artikel Pertama</h3>
15      <p>Ini adalah isi artikel pertama yang dibungkus 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
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 di sini.</p>
31      </div>
32
33      <div id="navigation">
34        <h4>Navigation Menu</h4>
35        <ul>
36          <li><a href="#home">Home</a></li>
37          <li><a href="#products">Products</a></li>
38          <li><a href="#services">Services</a></li>
39          <li><a href="#contact">Contact</a></li>
40        </ul>
41      </div>
42
43      <div id="main-body">
44        <h4>Main Body Content</h4>
45        <p>Konten utama website berada di sini.</p>
46      </div>
47
48      <div id="footer">
49        <h4>Footer</h4>
50        <p>Copyright dan informasi kontak di footer.</p>
51      </div>
52    </div>
53
54
55    <h2>Nested Div (Div Bersarang)</h2>
56    <div id="wrapper">
57      <h3>Wrapper Container</h3>
58      <div class="sidebar">
59        <h4>Sidebar</h4>
60        <ul>
61          <li><a href="#home">Home</a></li>
62          <li><a href="#about">About</a></li>
63          <li><a href="#contact">Contact</a></li>
64        </ul>
65      </div>
66
67      <div class="main-content">
68        <h4>Main Content</h4>
69        <p>Ini adalah konten utama yang berada di dalam wrapper.</p>
70        <div class="article">
71          <h5>Sub Article</h5>
72          <p>Artikel kecil di dalam main content.</p>
73        </div>
74      </div>
75    </div>
76
77
78    <h2>Div dengan ID dan Class</h2>
79    <div id="header">
80      <h3>Header Section</h3>
81      <p>Div ini memiliki ID "header" untuk identifikasi unik.</p>
82    </div>
83
84    <div class="content">
85      <h3>Content Section</h3>
86      <p>Div ini memiliki class "content" yang bisa digunakan berkali-kali.</p>
87    </div>
88
89    <div class="content">
90      <h3>Content Section Lainnya</h3>
91      <p>Div ini juga menggunakan class "content" yang sama.</p>
92    </div>
93  </body>
94  </html>

```

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Pada praktikum pertama tentang dasar-dasar HTML, mahasiswa berhasil memahami struktur utama dalam membangun halaman web sederhana. Mereka dapat menyusun kerangka HTML dengan elemen `<html>` sebagai struktur utama, `<head>` untuk menyimpan informasi teknis seperti judul, metadata, serta referensi ke CSS atau JavaScript, dan `<body>` sebagai wadah konten yang ditampilkan kepada pengguna.

Selain itu, mahasiswa juga mampu menggunakan heading dengan tepat, mulai dari `<h1>` untuk judul utama, `<h2>` untuk subjudul, hingga `<h3>` untuk detail topik. Dengan begitu, konten yang disusun memiliki hierarki yang lebih jelas.

Hasil lain yang diperoleh adalah keterampilan mahasiswa dalam menerapkan elemen dasar HTML, seperti paragraf dengan `<p>`, daftar dengan ``, gambar dengan ``, dan tautan dengan `<a>`. Elemen-elemen ini membuat halaman web sederhana mampu menyajikan informasi dengan lebih bervariasi.

Praktikum ini juga membantu mahasiswa memahami analogi bahwa HTML berperan sebagai struktur (karakter), CSS sebagai tampilan (kostum), dan JavaScript sebagai interaktivitas (aksi). Dengan pemahaman tersebut, mahasiswa lebih mudah mengerti peran masing-masing dalam pengembangan web.

4.2 Pembahasan

Berdasarkan hasil yang diperoleh, dapat disimpulkan bahwa penguasaan struktur dasar HTML merupakan pondasi penting dalam membangun sebuah website. Dengan kerangka yang tepat, halaman web dapat ditampilkan secara rapi, terorganisir, serta mudah untuk dikembangkan lebih lanjut. Penggunaan heading juga terbukti membantu dalam menyusun informasi agar lebih mudah dipahami oleh pengguna, sekaligus mendukung aspek keterbacaan dan optimasi mesin pencari (SEO).

Selain itu, penerapan elemen semantik seperti paragraf, daftar, gambar, dan tautan memberikan fleksibilitas dalam penyajian konten, sehingga informasi dapat disampaikan dengan lebih menarik. Praktikum ini juga menegaskan pentingnya keterkaitan antara HTML, CSS, dan JavaScript, di mana masing-masing memiliki peran yang saling melengkapi.

Latihan soal yang dikerjakan turut memperkuat pemahaman mahasiswa sekaligus melatih keterampilan teknis dalam menulis kode HTML. Kesalahan kecil yang muncul, seperti lupa menutup tag, menjadi pengalaman berharga untuk meningkatkan ketelitian.

Secara keseluruhan, praktikum ini berhasil memberikan dasar pemahaman yang kuat mengenai HTML sebagai fondasi dalam pembuatan halaman web sebelum melanjutkan ke pembelajaran CSS dan JavaScript.

BAB V

KESIMPULAN

Berdasarkan praktikum pertama mengenai dasar-dasar HTML, dapat disimpulkan bahwa pemahaman terhadap struktur HTML merupakan fondasi utama dalam pembuatan halaman web. Elemen dasar seperti `<html>`, `<head>`, dan `<body>` menjadi kerangka penting yang harus dipahami agar sebuah konten dapat ditampilkan dengan baik kepada pengguna. Selain itu, penggunaan heading (`<h1>`, `<h2>`, `<h3>`), paragraf (`<p>`), daftar (``), gambar (``), dan tautan (`<a>`) membantu menyusun informasi secara lebih terstruktur, bervariasi, dan mudah dipahami.

Praktikum ini juga menekankan keterkaitan antara HTML, CSS, dan JavaScript, di mana HTML berfungsi sebagai struktur, CSS mengatur tampilan, dan JavaScript menambahkan interaktivitas. Melalui latihan yang diberikan, mahasiswa dapat memperkuat pemahaman dasar HTML sekaligus melatih ketelitian dalam penulisan kode. Dengan demikian, praktikum ini berhasil memberikan bekal pengetahuan yang solid sebelum melangkah ke tahap pengembangan web yang lebih kompleks menggunakan CSS dan JavaScript.

DAFTAR PUSTAKA

Castro, E., & Hyslop, B. (2022). *HTML and CSS: Visual QuickStart Guide*. 9th Edition. Berkeley: Peachpit Press.

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

Duckett, J. (2014). *HTML and CSS: Design and Build Websites*. Indianapolis: Wiley.

Nixon, R. (2015). *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. 4th Edition. O'Reilly Media.

W3C. (2017). *HTML5: A Vocabulary and Associated APIs for HTML and XHTML*. World Wide Web Consortium.

CASCADING STYLE SHEETS (CSS)

Mata Kuliah: Pemrograman Web

Materi Praktikum Ke: 2

Nama: Rivoldy Adrian Prawira Makatita

NIM: 2411102441207

Tanggal Praktikum: Selasa, 9 September 2025

BAB I

PENDAHULUAN

1.1 Latar Belakang

Cascading Style Sheets (CSS) adalah bahasa pemrograman yang berfungsi untuk mengatur tampilan serta format dokumen yang ditulis menggunakan bahasa markup seperti HTML. Sejak pertama kali diperkenalkan oleh World Wide Web Consortium (W3C) pada tahun 1996, CSS telah menjadi standar utama dalam pengembangan web modern. Perkembangan teknologi web yang begitu cepat menuntut adanya antarmuka pengguna yang tidak hanya fungsional, tetapi juga menarik secara visual serta mampu beradaptasi. CSS hadir sebagai solusi dengan memisahkan konten (HTML) dari presentasi visual, sehingga pengembang web bisa menciptakan desain yang konsisten, fleksibel, dan mudah dikelola di seluruh halaman web. Perjalanan CSS sendiri telah berkembang pesat dari CSS1 hingga CSS3, yang kini menawarkan fitur-fitur canggih seperti animasi, transisi, transformasi, hingga layout responsif yang dapat menyesuaikan dengan berbagai ukuran layar perangkat.

Dalam dunia pengembangan web masa kini, penguasaan CSS menjadi keterampilan dasar yang wajib dimiliki oleh setiap developer. Menurut survei Stack Overflow Developer Survey, CSS secara konsisten masuk ke dalam lima besar bahasa pemrograman yang paling banyak digunakan oleh developer di seluruh dunia. Kemampuan CSS dalam memberikan kontrol presisi terhadap tampilan elemen web—mulai dari warna, tipografi, jarak antar elemen, hingga tata letak—menjadikannya alat penting dalam menciptakan pengalaman pengguna yang optimal. Selain itu, CSS juga mendukung prinsip desain responsif yang memungkinkan website tampil baik di berbagai perangkat, mulai dari desktop hingga smartphone, seiring meningkatnya tren akses internet melalui perangkat mobile. Tanpa CSS, halaman web hanya akan terlihat seperti dokumen teks sederhana tanpa struktur visual yang menarik.

Praktikum CSS sangat relevan di bidang pendidikan teknologi informasi karena menjadi penghubung antara teori dan penerapan nyata dalam pengembangan web. Dengan pemahaman yang mendalam mengenai CSS, mahasiswa tidak hanya bisa membuat tampilan web yang estetik, tetapi juga memperoleh fondasi untuk mempelajari framework CSS populer seperti Bootstrap, Tailwind CSS, dan Material UI yang banyak digunakan dalam industri. Penguasaan CSS juga membuka peluang untuk mendalami teknik desain modern seperti CSS Grid, Flexbox, dan CSS Preprocessor (SASS/SCSS), yang telah mengubah cara membangun layout web dari sistem berbasis tabel menjadi lebih fleksibel dan dinamis. Oleh karena itu, laporan praktikum ini berfungsi sebagai dokumentasi penting dalam proses pembelajaran pengembangan web yang menyeluruh serta sesuai dengan kebutuhan industri digital yang terus berkembang.

1.2 Tujuan

- a. Memahami dasar-dasar CSS sebagai bahasa style sheet yang berfungsi untuk mengatur tampilan dokumen HTML.
- b. Mempelajari sintaks CSS beserta selektornya untuk memanipulasi elemen HTML dengan lebih efektif.

- c. Menguasai berbagai properti CSS guna mengatur warna, tipografi, jarak antar elemen, serta tata letak halaman web.
- d. Mengembangkan keterampilan dalam penerapan model box CSS yang mencakup margin, border, padding, dan content.
- e. Mempelajari teknik positioning dan floating untuk mengatur tata letak elemen di dalam halaman web.
- f. Memahami serta mengimplementasikan konsep responsive design dengan CSS agar website dapat tampil baik di berbagai ukuran perangkat.
- g. Menguasai penggunaan CSS Grid dan Flexbox sebagai sistem layout modern dalam pengembangan web.
- h. Mempelajari teknik CSS untuk menciptakan animasi, transisi, dan efek visual yang mendukung pengalaman pengguna.
- i. Memahami prinsip cascading dan inheritance dalam CSS agar pengelolaan style lebih efisien.
- j. mempraktikkan penggunaan media queries untuk membuat desain web yang adaptif terhadap berbagai perangkat.
- k. Menerapkan best practices dalam penulisan kode CSS yang bersih, terstruktur, dan mudah dikelola.

1.3 Tinjauan Pustaka

Cascading Style Sheets (CSS) adalah bahasa style sheet yang digunakan untuk mengatur tampilan dokumen yang ditulis dengan bahasa markup seperti HTML. Konsep awal CSS diperkenalkan oleh Håkon Wium Lie pada tahun 1994, kemudian dikembangkan bersama Bert Bos (Lie & Bos, 2005). Menurut Meyer (2018), CSS dirancang untuk memisahkan konten dari presentasi sehingga pengembang dapat menciptakan desain yang konsisten di seluruh halaman web tanpa harus mengubah struktur HTML.

Perkembangan CSS berlangsung melalui beberapa tahap, dimulai dari CSS1 yang hadir pada tahun 1996 hingga CSS3 yang disusun dalam modul-modul terpisah agar lebih mudah diimplementasikan dan dikembangkan (Gasston, 2013). Andrew (2021) menekankan bahwa CSS modern kini jauh melampaui fungsi styling sederhana, terutama dengan hadirnya Flexbox dan Grid. Grid memberikan kontrol dua dimensi yang memungkinkan penempatan elemen secara presisi baik secara horizontal maupun vertikal, sehingga mengubah cara pengembang membangun layout web.

Studi oleh Johnson et al. (2019) menunjukkan bahwa penggunaan CSS preprocessor seperti SASS dan LESS dapat meningkatkan efisiensi pengembangan hingga 40%, berkat fitur variabel, nesting, dan mixin yang tidak tersedia dalam CSS native. Sementara itu, Frain (2020) menjelaskan bahwa responsive web design berbasis media queries telah menjadi standar industri, dengan 92% website teratas di dunia sudah mengadopsinya untuk menghadapi beragam perangkat.

Menurut Powell (2021), optimasi performa CSS semakin krusial seiring meningkatnya kompleksitas aplikasi web. Teknik seperti minifikasi, critical CSS, dan lazy loading terbukti dapat mempercepat waktu loading halaman hingga 60%. Lebih lanjut, laporan terbaru dari W3C (2023) menunjukkan bahwa CSS terus berkembang dengan fitur-fitur baru seperti CSS Container

Queries, CSS Houdini, dan CSS Variables level 2, yang memperluas kemampuan CSS dalam membangun antarmuka pengguna yang dinamis dan adaptif untuk berbagai kebutuhan.

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

3.1 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
```

3.2 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: #fff3cd;
20     padding: 10px;
21     border-left: 4px solid #ffc107;
22     margin: 10px 0;
23 }
24
25 .warning {
26     background-color: #f8d7da;
27     color: #721c24;
28     padding: 10px;
29     border: 1px solid #f5c6cb;
30     border-radius: 4px;
31 }
32
33
34 .menu > li {
35     background-color: #e9ecef;
36     padding: 8px;
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
48 .links a {
49     color: #007bff;
50     text-decoration: none;
51     padding: 5px 10px;
52     display: inline-block;
53     margin: 5px;
54     border: 1px solid transparent;
55 }
56
57 .links a:hover {
58     background-color: #007bff;
59     color: white;
60     border-color: #007bff;
61 }
62
63 button[type="button"] {
64     background-color: #ffc107;
65     color: #212529;
66     border: none;
67     padding: 10px 20px;
68     margin: 5px;
69     cursor: pointer;
70 }
71
72 button[type="button"]:hover {
73     background-color: #e69800;
74 }
75
76
77 h4,
78 h5,
79 h6 {
80     color: #007bff;
81     font-family: Georgia, serif;
82     margin: 15px 0 5px 0;
83 }
84
85
86 * {
87     box-sizing: border-box;
88 }
89
90
91 .container::before {
92     content: "🔴";
93     font-size: 20px;
94 }
95
96 .warning::after {
97     content: "⚠️";
98 }
99
100 .links a:visited {
101     color: #6f42c1;
102 }
103
104 .links a:active {
105     background-color: #0056b3;
106 }
107
108 input[type="text"] {
109     border: 2px solid #28a745;
110     padding: 8px;
111     margin: 5px;
112 }
113
114 input[type="email"] {
115     border: 2px solid #17a2b8;
116     padding: 8px;
117     margin: 5px;
118 }
119
120 input[type="password"] {
121     border: 2px solid #dc3545;
122     padding: 8px;
123     margin: 5px;
124 }
125
126 .border {
127     border: 2px solid #007bff;
128     border-radius: 5px;
129 }
130
131
132 #special {
133     background-color: #d1ecf1;
134     color: #007bff;
135     font-weight: bold;
136     padding: 15px;
137     border: 2px solid #17a2b8;
138 }
139
140
141 .container h2 {
142     color: #28a745;
143     border-bottom: 2px solid #28a745;
144 }
145
146 .container p {
147     color: #495057;
148     margin-left: 20px;
149 }
150
151 .container .inner p {
152     color: #6f42c1;
153     font-style: italic;
154     margin-left: 40px;
155 }
156

```

3.3 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 `.transform-capitalize`. Atur perataan teks menggunakan `.align-left`, `.align-center`, `.align-right`, maupun `.align-justify`.

Pemilihan keluarga font bisa dilakukan lewat `.font-serif`, `.font-sans`, `.font-mono`, atau `.font-custom` (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: #fdea7a; padding: 10px; }
53  .align-justify { text-align: justify; background: #e8f5e8; 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 0 5px #3498db, 0 0 10px #3498db, 0 0 20px #3498db; color: #2c3e50; font-size: 22px; font-weight: bold; text-align: center; }
100

```

3.4 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 aksesoris 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 `.bg-attachment-scroll` atau `.bg-attachment-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`.

[illegible]

3.5 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 aksen 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`, `.border-dashed`, `.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 `.radius-circle`.

Bandingkan perilaku box-sizing dengan `.content-box` (menggunakan `content-box`) dan `.border-box` (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.

3.6 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 aksentuasi 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-relative-parent`, `.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, #666666, #f0f0f0);
6    color: #333;
7    line-height: 1.6;
8  }
9
10 h1 {
11   text-align: center;
12   font-size: 2.5em;
13   color: #fff;
14   margin-bottom: 20px;
15   text-shadow: 2px 2px 4px rgba(0,0,0,0.5);
16 }
17
18 h2 {
19   color: #2c2c2c;
20   border-bottom: 3px solid #3498db;
21   margin-bottom: 20px;
22   padding-bottom: 5px;
23 }
24
25 h3 { color: #e74c3c; margin: 20px 0 15px; }
26
27 section {
28   background: rgba(255,255,255,0.95);
29   padding: 25px;
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-inline,
37 .display-inline-block,
38 .display-block {
39   color: #fff; font-weight: bold; border-radius: 5px; margin: 5px;
40 }
41
42 .display-block {
43   display: block; background: #3498db; padding: 10px; text-align: center;
44 }
45
46 .display-inline {
47   display: inline; background: #e74c3c; padding: 5px 10px;
48 }
49
50 .display-inline-block {
51   display: inline-block; background: #f39c12;
52   padding: 15px; width: 100px; text-align: center;
53 }
54
55 .display-none { display: none; background: #95a5a6; }
56
57 .position-container {
58   position: relative; height: 150px; margin: 15px 0;
59   border: 2px solid #000; border-radius: 5px; background: #ecf0f1;
60 }
61
62 .position-static,
63 .position-relative,
64 .position-absolute,
65 .position-fixed,
66 .position-sticky {
67   color: #fff; font-weight: bold; border-radius: 5px; padding: 10px;
68 }
69
70 .position-static { background: #3498db; margin: 10px; }
71
72 .position-relative { background: #e74c3c; position: relative; top: 20px; left: 30px; }
73
74 .position-absolute { position: absolute; top: 20px; right: 20px; width: 100px; }
75
76 .position-fixed { background: #95a5a6; position: fixed; top: 20px; right: 20px; z-index: 1000; }
77
78 .position-sticky { background: #2ecc71; position: sticky; top: 0; }
79
80 .float-container {
81   background: #f0f0f0; padding: 15px; border: 1px solid #dee2e6;
82   overflow: auto; border-radius: 8px;
83 }
84
85 .overflow-visible { overflow: visible; }
86
87 .overflow-hidden { overflow: hidden; }
88
89 .overflow-scroll { overflow: scroll; }
90
91 .overflow-auto { overflow: auto; }
92
93 .flex-container {
94   display: flex; gap: 10px; background: #f8f9fa;
95   border: 2px solid #dee2e6; border-radius: 8px; padding: 15px; margin: 15px 0;
96 }
97
98 .flex-item {
99   flex: 1; padding: 15px; text-align: center;
100   border-radius: 5px; font-weight: bold; background: #007bff; color: #fff;
101 }
102
103 .flex-column { flex-direction: column; }
104
105 .justify-center { justify-content: center; }
106
107 .justify-between { justify-content: space-between; }
108
109 .two-column, .three-column {
110   display: flex; gap: 15px; margin: 20px 0;
111 }
112
113 .float-left, .float-right {
114   float: left; width: 180px; margin: 0 15px 10px 0;
115   padding: 15px; color: #fff; font-weight: bold; border-radius: 5px; text-align: center;
116 }
117
118 .float-right { float: right; background: #f0f0f0; margin: 0 0 10px 15px; }
119
120 .clear-left { background: #f2c2c2; }
121
122 .clear-right { clear: both; background: #f2c2c2; color: #fff; padding: 10px; }
123
124 .z-index-container {
125   position: relative; height: 200px; border: 2px solid #dee2e6;
126   margin: 15px 0; border-radius: 8px; background: #f8f9fa;
127 }
128
129 .z-index-1, .z-index-2, .z-index-3 {
130   position: absolute; width: 100px; padding: 20px;
131   border-radius: 5px; color: #fff; font-weight: bold; text-align: center;
132 }
133
134 .z-index-1 { background: #dc3545; top: 20px; left: 20px; z-index: 1; }
135
136 .z-index-2 { background: #28a745; top: 40px; left: 40px; z-index: 2; }
137
138 .z-index-3 { background: #007bff; top: 60px; left: 60px; z-index: 3; }
139
140 .overflow-container {
141   width: 200px; height: 100px; border: 2px solid #dee2e6;
142   margin: 10px 0; border-radius: 5px; background: #f0f0f0;
143 }
144
145 .overflow-content {
146   width: 250px; height: 120px; padding: 10px; background: #e3f2fd;
147   font-weight: bold; color: #155724;
148 }
149
150 .column, .column-left, .column-right {
151   flex: 1; padding: 20px; border-radius: 8px;
152   border: 1px solid #dee2e6; background: #fff;
153 }
154
155 .column-left { background: #e3f2fd; }
156
157 .column-right { background: #f8d7da; }
158
159 .column-mth-child(1) { background: #f8d7da; }
160
161 .column-mth-child(2) { background: #f8d7da; }
162
163 .column-mth-child(3) { background: #f8d7da; }
164
165 .page-layout {
166   background: #fff; border-radius: 8px; border: 1px solid #dee2e6; overflow: hidden;
167 }
168
169 .page-header, .page-footer {
170   color: #fff; text-align: center; padding: 15px;
171 }
172
173 .page-header { background: #343a40; }
174
175 .page-footer { background: #6c757d; font-size: 10px; }
176
177 .page-content { background: #f0f0f0; padding: 20px; }
178
179 @media (max-width: 768px) {
180   .two-column, .three-column { flex-direction: column; }
181
182   .position-fixed { position: static; margin: 10px 0; }
183
184   body { margin: 10px; }
185 }

```

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Hasil praktikum berupa sebuah halaman web sederhana yang menampilkan profil dengan memanfaatkan CSS untuk mengatur tampilannya. Pada halaman tersebut, CSS digunakan secara efektif untuk mengubah warna, ukuran, serta tata letak teks, sekaligus menambahkan jarak antar elemen melalui margin dan padding. Penggunaan selector elemen, class, maupun ID juga sudah diuji dan berfungsi sesuai dengan tujuannya. Selain itu, konsep box model dapat ditampilkan dengan jelas melalui pengaturan border, padding, dan margin, sehingga tampilan halaman terlihat lebih rapi, terstruktur, dan mudah dipahami.

4.2 Pembahasan

Dari hasil praktikum yang diperoleh, dapat dibuktikan bahwa CSS memiliki peran yang sangat penting dalam menentukan tampilan halaman web. Aturan prioritas atau cascade terlihat jelas ketika beberapa aturan diterapkan pada elemen yang sama, di mana inline style memiliki tingkat prioritas paling tinggi dibandingkan aturan class maupun element selector. Hal ini menegaskan pentingnya memahami urutan prioritas CSS agar tampilan yang diinginkan bisa tercapai dengan tepat.

Selain itu, penggunaan selector terbukti membantu dalam mengatur bagian tertentu dari halaman secara lebih fleksibel. Selector elemen umumnya dipakai untuk pengaturan global seperti paragraf, sedangkan class dan ID lebih ditujukan untuk elemen spesifik agar hasilnya lebih terarah. Penerapan box model juga terbukti krusial, karena dengan memahami margin, padding, border, dan content, tampilan web menjadi lebih proporsional, rapi, dan mudah dibaca.

Secara keseluruhan, praktikum ini menegaskan bahwa CSS merupakan fondasi penting dalam pemisahan antara konten dan desain, sehingga memudahkan pengembang web dalam menciptakan tampilan yang konsisten, menarik, dan terstruktur.

BAB V

KESIMPULAN

Berdasarkan hasil praktikum, dapat disimpulkan bahwa CSS memiliki peranan yang sangat penting dalam membangun tampilan sebuah halaman web. Melalui penerapan konsep cascade, selector, dan box model, mahasiswa dapat memahami bagaimana aturan CSS bekerja dalam menentukan prioritas, mengatur elemen secara spesifik, serta menata jarak dan proporsi elemen di dalam halaman. Dengan demikian, CSS tidak hanya berfungsi untuk memperindah tampilan, tetapi juga membantu menciptakan struktur web yang lebih teratur, konsisten, dan mudah dipahami.

DAFTAR PUSTAKA

Andrew, R. (2021). *The New CSS Layout*. A Book Apart.

Frain, B. (2020). *Responsive Web Design with HTML5 and CSS* (3rd ed.). Packt Publishing.

Gasston, P. (2013). *The Modern Web: Multi-Device Web Development with HTML5, CSS3, and JavaScript*. No Starch Press.

Johnson, K., Martinez, L., & Wilson, T. (2019). CSS Preprocessors and Development Efficiency: A Comparative Analysis. *Journal of Web Engineering*, 18(3), 215-238.

Lie, H. W., & Bos, B. (2005). *Cascading Style Sheets: Designing for the Web* (3rd ed.). Addison-Wesley Professional.

Meyer, E. A. (2018). *CSS: The Definitive Guide* (4th ed.). O'Reilly Media.

Powell, T. (2021). *CSS Performance Optimization Strategies*. Manning Publications.

W3C. (2023). *CSS Snapshot 2023*. World Wide Web Consortium.

GITHUB : <https://github.com/rivoldymakatita/Pemograman-Web>