

2.1.4 CSS (*Cascading Style Sheets*)

CSS (*Cascading Style Sheets*) adalah bahasa pemrograman yang digunakan dalam pengembangan *web* untuk mengontrol tampilan dan gaya visual dari halaman-halaman *web*. CSS memungkinkan pengembang *web* untuk mengatur elemen-elemen HTML, seperti teks, gambar, dan tata letak, dengan mengatur properti seperti warna, ukuran teks, jarak antar elemen, dan banyak lagi. Dengan CSS, pengembang dapat menciptakan desain yang menarik dan terstruktur, serta mengatur bagaimana halaman *web* merespons berbagai perangkat dan ukuran layar. CSS juga memungkinkan pemisahan antara konten (HTML) dan presentasi (gaya), sehingga perubahan pada tampilan dapat dilakukan tanpa mengubah struktur dasar halaman *web*. Ini membuat CSS menjadi alat yang sangat penting dalam menciptakan pengalaman pengguna yang konsisten dan menarik di seluruh *web*. Selain itu, CSS juga mengadopsi prinsip *cascading*, yang berarti aturan gaya dapat ditentukan dengan tingkat kejelasan tertentu. Ini memungkinkan pengembang untuk menggabungkan aturan gaya dari berbagai sumber dan menentukan prioritasnya.

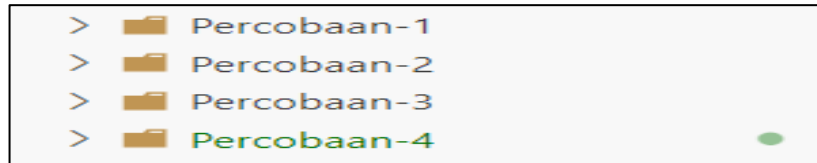
Selain itu, CSS mendukung konsep warisan di mana aturan yang didefinisikan pada tingkat lebih tinggi dapat diterapkan pada elemen-elemen yang lebih dalam dalam struktur dokumen. Dengan kata lain, CSS memberikan fleksibilitas yang luar biasa dalam mengendalikan tampilan halaman *web* dan memastikan bahwa situs *web* dapat tampak baik dan berfungsi dengan baik di semua perangkat dan layar.

CSS juga mendukung fitur *responsive design*, yang memungkinkan pengembang menyesuaikan tata letak halaman agar tampak optimal pada berbagai perangkat, seperti *desktop*, tablet, dan *smartphone*. Dengan menggunakan media *queries*, pengembang dapat menerapkan aturan CSS khusus berdasarkan lebar atau orientasi layar perangkat pengguna. Fitur ini penting dalam pengembangan *web* modern karena membantu menciptakan pengalaman pengguna yang lebih baik di berbagai ukuran layar tanpa perlu membuat versi terpisah dari halaman *web*.

3.1.4 CSS (Cascading Style Sheets)

1. Project

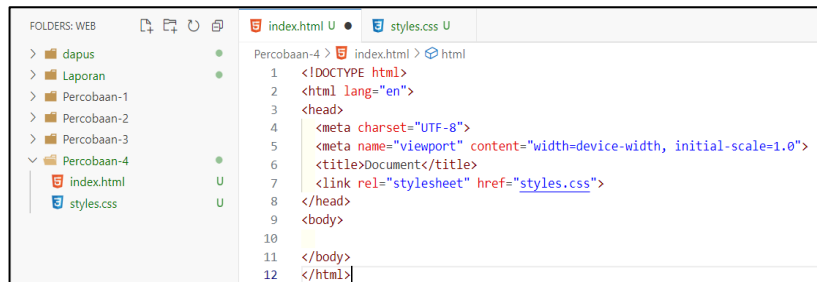
- a. Pada folder utama, buat folder baru dengan nama “Percobaan-4”



- b. Pada folder “Percobaan-4”, buat *file* baru dengan nama “index.html” dan tambahkan struktur dasar HTML 5 kedalamnya.

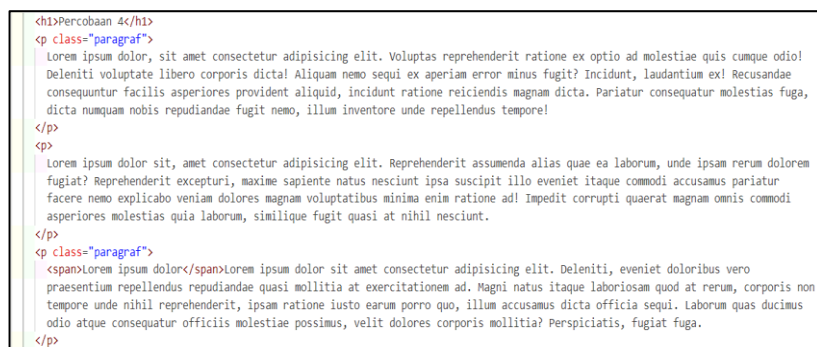


- c. Buat *file* baru dengan nama “style.css” dan tambahkan kode ini pada “index.html” di bagian *head*.



2. CSS Teks

- a. Pada *file* “index.html” tambahkan kode seperti berikut.



- b. Pada file “style.css” tambahkan kode berikut.

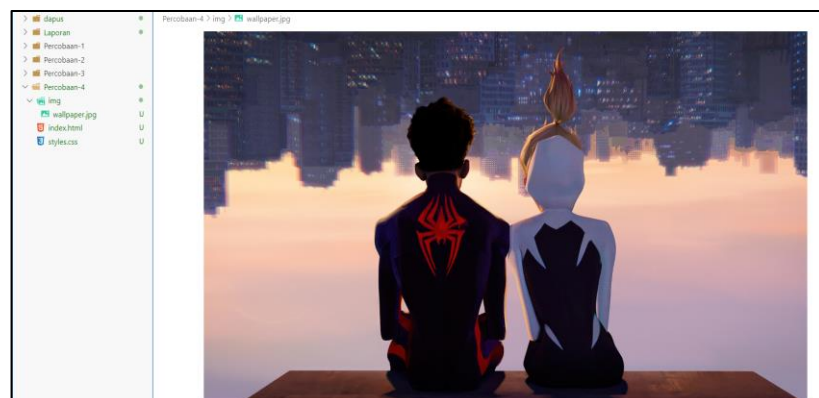
```
Percobaan-4 > styles.css > body
1  * {
2      font-family: 'Courier New', Courier, monospace;
3  }
4
5  .h1 {
6      color: red;
7  }
8
9  p {
10     color: blue;
11     font-family: Arial, Helvetica, sans-serif;
12     font-size: 20px;
13     text-align: center;
14     letter-spacing: 3px;
15 }
16
17 p.paragraf {
18     color: orange;
19     font-weight: bold;
20     font-size: 25px;
21     text-align: left;
22     line-height: 1,5cm;
23 }
24
25 span {
26     text-decoration: underline;
27     text-transform: uppercase;
```

3. Background

- a. Pada file “style.css”, tambahkan kode berikut untuk membuat *background* dengan warna.

```
body {
    background-color: rgb(233, 233, 233);
}
```

- b. Buat folder baru dengan nama “img” dan siapkan gambar untuk di jadikan *background*



- c. Pada file “style.css” tambahkan kode berikut untuk memasukkan *background image*.

```
body {  
    /* background-color: rgb(233, 233, 233); */  
    background-image: url(img/wallpaper.jpg);  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
    background-size: cover;  
}
```

4. Border, Margins, Padding, Height/Width

- a. Pada file “index.html” tambahkan kode seperti berikut.

```
<div class="box-1">  
    Lorem, ipsum dolor sit amet consectetur adipisicing elit. At enim harum  
    officia nostrum beatae assumenda aliquam maxime repellat. Commodi facere  
    nisi est modi sed quos ratione velit excepturi. Similique exercitationem  
    iusto ad explicabo autem nisi unde voluptatibus alias. Nostrum quia  
    aspernatur expedita, voluptatum autem temporibus dicta exercitationem  
    dolorem excepturi cum?  
</div>  
<div class="box-2">  
    Lorem, ipsum dolor sit amet consectetur adipisicing elit. At enim harum  
    officia nostrum beatae assumenda aliquam maxime repellat. Commodi facere  
    nisi est modi sed quos ratione velit excepturi. Similique exercitationem  
    iusto ad explicabo autem nisi unde voluptatibus alias. Nostrum quia  
    aspernatur expedita, voluptatum autem temporibus dicta exercitationem  
    dolorem excepturi cum?  
</div>  
<div class="box-3">  
    Lorem, ipsum dolor sit amet consectetur adipisicing elit. At enim harum  
    officia nostrum beatae assumenda aliquam maxime repellat. Commodi facere  
    nisi est modi sed quos ratione velit excepturi. Similique exercitationem  
    iusto ad explicabo autem nisi unde voluptatibus alias. Nostrum quia  
    aspernatur expedita, voluptatum autem temporibus dicta exercitationem  
    dolorem excepturi cum?  
</div>
```

- b. Pada file “style.css” tambahkan kode seperti berikut.

```
div {  
    border: 2px solid black;  
    width: 450px;  
    padding: 10px;  
    margin-bottom: 5px;  
}  
  
div.box-1 {  
    background-color: #d9534f;  
}  
  
div.box-2 {  
    background-color: black;  
    color: white;  
}  
  
div.box-3 {  
    background-color: #92d050;  
}
```

5. CSS Layout

- a. Pada “index.html” buat kodingan seperti berikut.

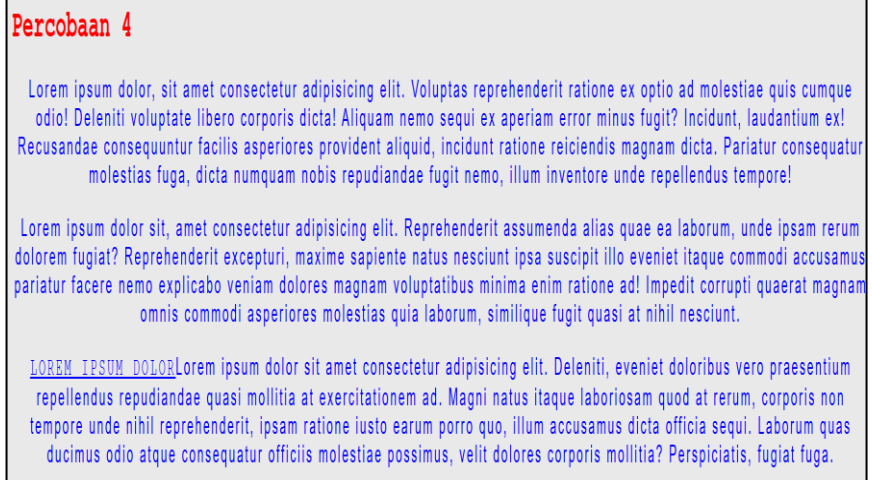
```
<div class="flex-container">
  <div>
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Quibusdam amet cupiditate voluptatibus, sequi reprehenderit officia ducimus obcaecati natus quod, minus quis quisquam delectus, doloremque voluptatem fugiat velit aliquid reiciendis facere. Hic, similique illo. Nam veritatis recusandae atque. Officiis odit vel maiores! Reprehenderit rerum, perferendis perspiciatis suscipit odit possimus accusamus molestiae.
  </div>
  <div>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quisquam ad cumque possimus fugiat tempore delectus expedita quod earum minus accusamus architecto esse odio libero, laborum mollitia at deserunt aliquam id tenetur incidunt cum ratione quibusdam! Debitis totam hic soluta nesciunt mollitia earum libero impedit ex. Praesentium tempore aspernatur modi id.
  </div>
  <div>
    Lorem ipsum dolor, sit amet consectetur adipisicing elit. Asperiores placeat tenetur aperiam consectetur, quod error reprehenderit, esse sed sequi culpa recusandae consequuntur, pariatur iusto iste. Repudiandae culpa aperiam architecto, ex veritatis eveniet ullam dolorum necessitatibus nam expedita quis vel, consequatur labore iusto perferendis facilis nihil modi quas magnam! Sint, recusandae?
  </div>
</div>
```

- b. Pada file “style.css” buat kodingan seperti berikut.

```
.flex-container {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  align-items: center;
  background-color: #dodgerblue;
  padding: 10px;
}
.flex-container > div {
  padding: 10px;
  margin: 5px;
  width: 300px;
  border: 2px solid black;
  background-color: #gray;
}
```

4.1.4 CSS (Cascading Style Sheets)

1. CSS Teks

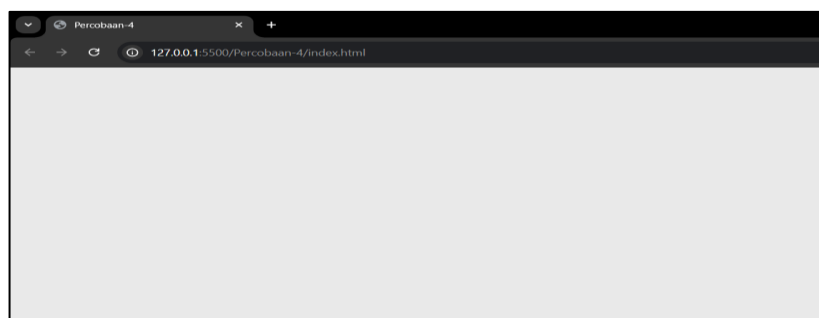


Pada kode HTML di atas, terdapat beberapa *tag* yang difungsikan untuk menampilkan teks dengan berbagai gaya. *Tag* “<h1>” di awal digunakan sebagai judul utama yang akan menampilkan teks “Percobaan 4”. Sebagai *heading* terbesar, *tag* “<h1>” biasanya digunakan untuk memberikan judul yang lebih besar dan mencolok dalam halaman HTML. Selanjutnya, terdapat beberapa *tag* “<p>” yang digunakan untuk menampilkan paragraf teks. *Tag* “<p>” pertama memiliki atribut “*class*=“*paragraf*””, yang artinya *tag* ini akan mengikuti aturan CSS khusus untuk kelas “*paragraf*” yang akan diatur dalam gaya yang berbeda. Di paragraf kedua, *tag* “<p>” tidak memiliki kelas, sehingga akan mengikuti aturan umum yang diterapkan pada semua *tag* “<p>”. Paragraf ketiga kembali menggunakan “*class*=“*paragraf*””, sehingga *tag* ini juga akan memiliki gaya khusus sesuai aturan CSS untuk kelas tersebut. Di dalam paragraf ketiga ini, terdapat *tag* “” yang dapat digunakan untuk menyoroti bagian teks tertentu atau menerapkan gaya tambahan pada teks “*Lorem ipsum dolor*”. Pada bagian CSS, terdapat aturan gaya yang mengatur tampilan *tag* di HTML. Pertama, aturan “*h1* { *color*: red; }” yang mengatur warna teks judul akan menjadi merah sesuai dengan keinginan. Selanjutnya, aturan

CSS untuk tag “<p>” diatur dengan berbagai properti. Aturan ini membuat semua paragraf (tanpa kelas) memiliki warna teks biru “color: blue;”, menggunakan jenis font *Arial* atau yang serupa “font-family: Arial, Helvetica, sans-serif;”, serta ukuran teks 20 piksel “font-size: 20px;”. Selain itu, teks pada paragraf-paragraf ini akan berada di tengah “text-align: center;” dan memiliki jarak antar huruf sebesar 3 piksel “letter-spacing: 3px;”, yang memberikan tampilan teks yang lebih lebar dan seimbang. Terakhir, terdapat aturan CSS untuk tag “<p>” dengan kelas *paragraf*. Aturan ini mengatur teks dalam tag tersebut agar berwarna oranye “color: orange;”, dibuat tebal “font-weight: bold;”, dan ukurannya sedikit lebih besar menjadi 25 piksel “font-size: 25px;”. Teks pada paragraf dengan kelas *paragraf* ini juga diatur rata kiri “text-align: left;” dan memiliki jarak antar baris sebesar 1,5 cm “line-height: 1.5cm”, membuat paragraf tampak lebih renggang dan mudah dibaca. Setiap aturan CSS ini disusun untuk memberikan tampilan yang berbeda pada setiap tag sesuai fungsinya, sehingga tag-tag ini akan tampil dengan gaya yang lebih dinamis dan terstruktur.

2. Background

a. Background Color



Bagian ini dimulai dengan selektor *body*, yang berarti aturan ini akan diterapkan pada seluruh halaman atau elemen *body* dari dokumen HTML. Properti *background-color* digunakan untuk memberi warna latar pada halaman. Warna yang digunakan adalah “rgb(233, 233, 233)”, yang menghasilkan warna abu-abu

muda. Angka dalam *rgb* ini mengatur intensitas merah, hijau, dan biru, dengan nilai yang lebih tinggi mendekati warna putih.

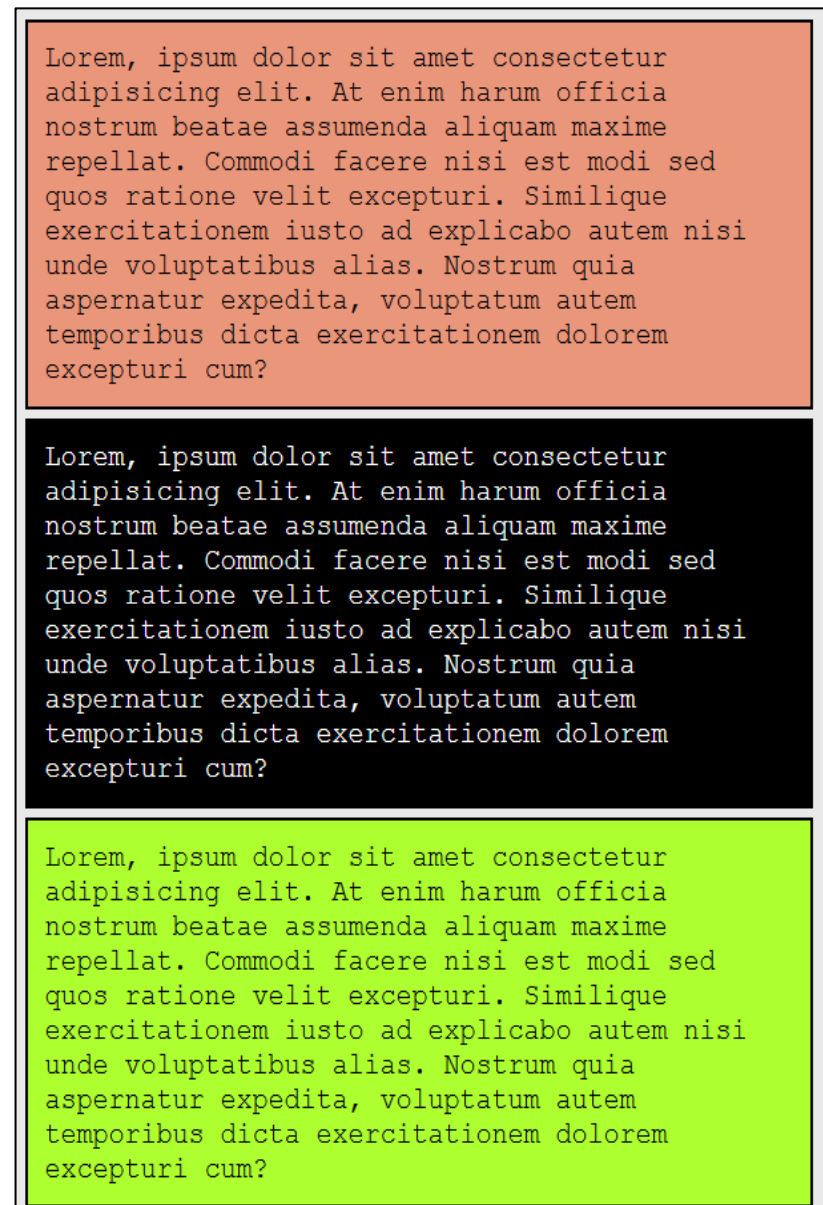
b. *Background Image*



Pada aturan ini, selektor *body* menandakan bahwa gaya ini akan diterapkan pada elemen utama halaman. Properti “*background-image*” digunakan untuk menampilkan gambar sebagai latar dengan alamat “*img/wallpaper.jpg*”. Selanjutnya, “*background-repeat: no-repeat;*” memastikan gambar latar tidak diulang, sehingga hanya satu gambar yang tampil. Properti “*background-attachment: fixed;*” membuat gambar latar tetap di tempat ketika halaman di-*scroll*. Terakhir, “*background-size: cover;*”

mengatur gambar agar memenuhi seluruh layar, meskipun ukuran gambar akan disesuaikan (diperbesar atau diperkecil) untuk mencakup seluruh area latar.

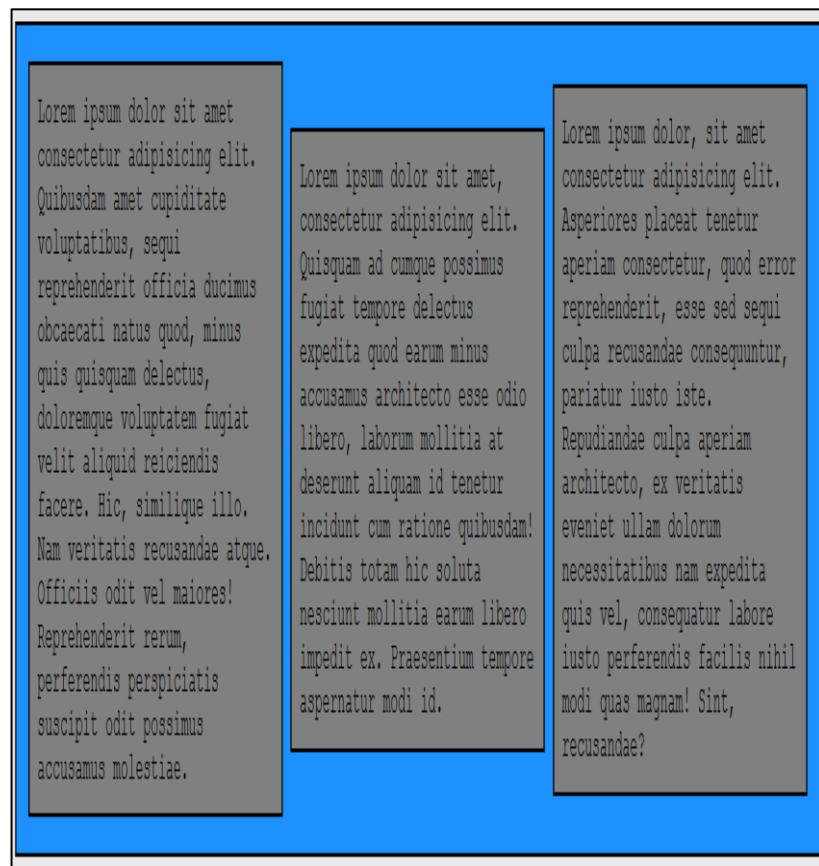
c. *Border, Margins, Padding, Height/Width*



Pada bagian ini setiap tag “<div>” diberikan kelas khusus, yaitu “class=“box-1””, “class=“box-2””, dan “class=“box-3””, yang memungkinkan setiap bagian memiliki gaya unik. Tag “<div>” pertama memiliki kelas “box-1” yang nantinya akan memberikan latar belakang warna *darksalmon* sesuai dengan

aturan CSS. Tag “<div>” kedua dengan kelas “box-2” diberi latar belakang hitam dengan teks berwarna putih agar mudah terbaca. Tag “<div>” ketiga, dengan kelas “box-3”, memiliki latar belakang berwarna *greenyellow*. Pada bagian CSS, aturan untuk selektor *div* mengatur gaya umum untuk semua tag “<div>” di halaman ini. Properti “*border: 2px black solid;*” memberikan garis tepi berwarna hitam dengan ketebalan 2 piksel. Properti “*width: 450px;*” mengatur lebar setiap *div* menjadi 450 piksel, sedangkan “*padding: 10px;*” menambahkan ruang di dalam *div* agar teks tidak terlalu menempel pada tepi. Properti “*margin-bottom: 5px;*” memberi jarak vertikal antara setiap *div*, sehingga terlihat lebih rapi dan terpisah.

d. CSS Layout



Pada bagian ini terdapat tag “<div class=“flex-container”>” yang berfungsi sebagai *container* utama untuk menampung

elemen-elemen di dalamnya. *Container* ini memiliki atribut *class* yang bernama "*flex-container*". Di dalam *container* ini, ada tiga elemen "*<div>*" lain, masing-masing berisi teks paragraf. Setiap *div* di dalam *container* utama ini akan mengikuti gaya yang diberikan pada *class* "*flex-container*" dan elemen-elemen *div* di dalamnya. Kemudian pada bagian CSS, untuk elemen dengan *class* "*flex-container*" diatur menggunakan *selector* "*.flex-container*". Pertama, "*width: fit-content;*" memastikan bahwa lebar *container* akan menyesuaikan dengan isi kontennya, tidak melebar lebih dari yang dibutuhkan. Kemudian, "*display: flex;*" menjadikan *container* ini sebagai *flexbox*, yang memudahkan pengaturan elemen-elemen di dalamnya agar tersusun rapi dalam satu baris. Dengan "*flex-wrap: wrap;*", elemen-elemen di dalam *container* dapat berpindah ke baris berikutnya apabila lebar layar tidak cukup. Selain itu, "*justify-content: center;*" membuat elemen-elemen anak diatur agar berada di tengah-tengah secara horizontal, sementara "*align-items: center;*" memusatkan elemen-elemen tersebut secara vertikal. *Container* ini juga diberi warna latar belakang biru dengan "*background-color: dodgerblue;*", serta jarak dari tepi sebesar 10 piksel dengan "*padding: 10px;*". Untuk setiap *div* di dalam *container* "*flex-container*", CSS mengatur gaya tambahan menggunakan *selector* "*.flex-container > div*". Setiap *div* diatur memiliki "*padding: 10px;*", yang menambah jarak di dalam *div* dari semua sisi. Selain itu, "*margin: 5px;*" menambahkan jarak antar *div*, sehingga mereka tidak saling menempel. Setiap *div* juga diatur memiliki lebar tetap 300 piksel melalui "*width: 300px;*", dan ditambahkan batas hitam setebal 2 piksel di sekelilingnya dengan "*border: 2px black solid;*". Warna latar belakang abu-abu pada setiap *div* diberikan melalui "*background-color: gray;*".

BAB V

PENUTUP

5.1 Kesimpulan

5.1.1 *Basic Tag HTML Part 1*

Dari hasil percobaan sebelumnya, menunjukkan bahwa struktur dasar HTML terdiri dari berbagai elemen seperti *tag* HTML, *head*, dan *body* yang menentukan tampilan halaman *web*. Penggunaan *tag heading*, paragraf, dan *formatting* memberikan variasi dan penekanan pada teks, sedangkan *tag komentar* membantu pengembang untuk mencatat informasi penting dalam kode tanpa memengaruhi tampilan. *Hyperlink* memungkinkan navigasi antar halaman dan ke URL *eksternal*, sedangkan *tag gambar* menyediakan cara untuk menampilkan media visual dengan atribut yang mendukung. Semua elemen ini berkontribusi pada pembentukan halaman *web* yang interaktif dan informatif.

5.1.2 *Basic Tag HTML Part 2*

Dari percobaan sebelumnya, dapat disimpulkan bahwa HTML memiliki elemen-elemen dasar seperti *list*, tabel, dan *form input*. *List* terbagi menjadi *unordered list* dan *ordered list*, yang masing-masing menggunakan *tag* “” dan “” serta diisi dengan elemen “”. Tabel dibangun dengan baris dan kolom menggunakan *tag* “<tr>”, “<td>”, dan “<th>”, yang dapat diatur ukurannya. *Form input* menyediakan berbagai tipe *input* untuk data pengguna, seperti *text*, *email*, *number* dan lainnya, serta atribut *id* dan *name* untuk identifikasi dan pengumpulan data. Dengan elemen-elemen ini, dapat memungkinkan pembuatan struktur halaman yang interaktif dan fungsional.

5.1.3 *HTML Layout*

Dari percobaan ini, dapat disimpulkan bahwa materi HTML Layout ini adalah penggunaan *semantic tag* seperti “<nav>”, “<section>”, “<article>”, dan “<footer>” membantu memberi makna yang jelas pada struktur halaman. Setiap elemen dibungkus dalam *div* dengan atribut *id* untuk identifikasi yang spesifik, yang memudahkan penerapan gaya atau

interaksi dengan CSS atau JavaScript. Elemen-elemen seperti “<h1>”, “<h2>”, “”, dan “<p>” membantu membentuk konten dan navigasi dengan lebih terstruktur. Struktur ini membuat halaman lebih rapi, mudah dipahami, dan meningkatkan aksesibilitas bagi pengguna maupun mesin pencari.

5.1.4 CSS (*Cascading Style Sheets*)

Modul ini menjelaskan penggunaan HTML untuk menyusun elemen-elemen teks dan CSS untuk mengatur tampilannya. Melalui *class* dan *selector*, setiap elemen dapat diberi gaya khusus seperti warna latar, ukuran teks, dan tata letak fleksibel menggunakan *flexbox*. Properti seperti *padding*, *margin*, dan *border* menambah struktur visual yang rapi, sementara *background-image* digunakan untuk memperindah tampilan dengan gambar latar.