

# **PRAKTIKUM PEMROGRAMAN WEB**



## **Informatika Telkom University Surabaya**

**Disusun Oleh:**

**I Gede Arengga Nabakta      NIM. 1203220108**

**PROGRAM STUDI S1 INFORMATIKA  
FAKULTAS INFORMATIKA  
UNIVERSITAS TELKOM SURABAYA  
2024**

**Github :** <https://github.com/rengganabakta/Pemrograman-Web/tree/master/Praktikum%204>

Soal No 1

( Laporan Praktikum )

Code :

HTML : praktikum.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/all.min.css" rel="stylesheet">
  <link rel="stylesheet" href="./assets/style.css">
  <title>Web Design</title>
</head>

<body>
  <!-- Header -->
  <div class="nav-container">
    <header>
      <label for="hamburger">&#9776;</label>
      <input type="checkbox" id="hamburger" />
      <div class="logo">
        
      </div>
      <nav>
        <ul>
          <li><a href="#" class="active">Home</a></li>
          <li><a href="#">HTML</a></li>
          <li><a href="#">CSS</a></li>
          <li><a href="#">Javascript</a></li>
          <li><a href="#">PHP</a></li>
        </ul>
      </nav>
    </header>
  </div>
  <!-- Hero -->
  <div class="hero-container">
    <div class="hero-text">
      <h1>Pemrograman Web</h1>
      <p>Belajar HTML, CSS, Javascript dan PHP</p>
      <a href="#" class="button">Pelajari Lebih Lanjut</a>
    </div>
  </div>
  <!-- Showcase -->
  <div class="showcase-container">
    <section class="showcase">
```

```

        <div class="showcase-image">
            
        </div>
        <div class="showcase-text">
            <h1>Lorem ipsum dolor sit amet consectetur</h1>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
                Consectetur sed consequatur ab architecto aut? Necessitatibus
explicabo similique
                consectetur, corporis deleniti accusantium odit aperiam non
architecto facere
                cupiditate perspiciatis eveniet eaque!</p>
            <p>Molestiae, sequi illo optio vitae error enim earum?</p>
        </div>
    </section>
</div>
<!-- Showcase -->
<div class="showcase-container">
    <section class="showcase">
        <div class="showcase-text">
            <h1>Lorem ipsum dolor sit amet consectetur</h1>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
                Blanditiis necessitatibus consectetur maxime saepe animi
laboriosam eligendi
                aperiam officiis quod, dolore tempore eum natus dolorem
cumque earum minus
                repudiandae excepturi dicta.</p>
            <p>Molestiae, sequi illo optio vitae error enim earum?</p>
        </div>
        <div class="showcase-image">
            
        </div>
    </section>
</div>
<!-- Blog -->
<div class="blog-container">
    <header>
        <h1>Lorem Ipsum</h1>
        <hr>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
Dignissimos,
                vitae!</p>
    </header>
    <div class="blog-index">
        <div class="snippet">
            
            <div class="snippet-txt">
                <h2>Lorem Ipsum</h2>
                <p>Lorem ipsum dolor sit amet consectetur, adipisicing
elit.
                    Ducimus quos esse odio nisi sit quidem fugiat
aspernatur quaerat eligendi ipsum

```

```
        sed placeat in a veniam, exercitationem iusto rem
consequuntur quisquam.</p>
        <p>Dolore provident expedita hic soluta maxime
distinctio, error
        delectus.</p>
        <p><small>Lorem Ipsum</small></p>
    </div>
</div>
<div class="snippet">
    
    <div class="snippet-txt">
        <h2>Lorem Ipsum</h2>
        <p>Lorem ipsum dolor sit amet consectetur, adipisicing
elit.
        Ducimus quos esse odio nisi sit quidem fugiat
aspernatur quaerat eligendi ipsum
        sed placeat in a veniam, exercitationem iusto rem
consequuntur quisquam.</p>
        <p>Dolore provident expedita hic soluta maxime
distinctio, error
        delectus.</p>
        <p><small>Lorem Ipsum</small></p>
    </div>
</div>
<div class="snippet">
    
    <div class="snippet-txt">
        <h2>Lorem Ipsum</h2>
        <p>Lorem ipsum dolor sit amet consectetur, adipisicing
elit.
        Ducimus quos esse odio nisi sit quidem fugiat
aspernatur quaerat eligendi ipsum
        sed placeat in a veniam, exercitationem iusto rem
consequuntur quisquam.</p>
        <p>Dolore provident expedita hic soluta maxime
distinctio, error
        delectus.</p>
        <p><small>Lorem Ipsum</small></p>
    </div>
</div>
</div>
</div>
<!-- Footer -->
<footer class="footer-container">
    <div class="footer-menu">
        <section>
            <div class="logo-container">
                <a href="#">
                    
                </a>
            </div>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Lorem
```

```
        ipsum dolor sit amet consectetur adipisicing elit.</p>
</section>
<section>
    <h3>Belajar Web</h3>
    <ul>
        <li><a href="#">HTML</a></li>
        <li><a href="#">CSS</a></li>
        <li><a href="#">Javascript</a></li>
        <li><a href="#">PHP</a></li>
        <li><a href="#">MySQL</a></li>
    </ul>
</section>
<section>
    <h3>Belajar Coding</h3>
    <ul>
        <li><a href="#">HTML</a></li>
        <li><a href="#">CSS</a></li>
        <li><a href="#">Javascript</a></li>
        <li><a href="#">PHP</a></li>
        <li><a href="#">MySQL</a></li>
    </ul>
</section>
<section>
    <h3>Informasi</h3>
    <ul>
        <li><i class="fas fa-envelope fa-fw"></i>
email@email.com</li>
        <li><i class="fas fa-phone fa-fw"></i> (021) 123456</li>
        <li><i class="fas fa-globe fa-fw"></i> www.url.com</li>
    </ul>
</section>
</div>
<hr>
<div class="footer-copyright">
    <div>
        <small>&copy; Pemrograman Web 2024</small>
    </div>
    <div>
        <a href="#"><i class="fab fa-facebook fa-lg"></i></a>
        <a href="#"><i class="fab fa-twitter fa-lg"></i></a>
        <a href="#"><i class="fab fa-instagram fa-lg"></i></a>
        <a href="#"><i class="fab fa-google-plus fa-lg"></i></a>
        <a href="#"><i class="fab fa-github fa-lg"></i></a>
    </div>
</div>
</footer>
</body>

</html>
```

Penjelasan :

Head :

1. `<!DOCTYPE html>`: Ini adalah deklarasi untuk jenis dokumen yang akan digunakan, yaitu HTML. Ini memberi tahu browser bahwa dokumen ini adalah dokumen HTML.
2. `<html lang="en">`: Ini adalah elemen root dari dokumen HTML. Isinya mengandung seluruh konten dokumen HTML. Atribut `lang="en"` menunjukkan bahwa bahasa dokumen ini adalah bahasa Inggris.
3. `<head>`: Ini adalah elemen kepala dari dokumen HTML. Ini berisi informasi meta dan referensi ke berkas eksternal seperti CSS dan JavaScript yang diperlukan untuk mengatur tata letak dan gaya dokumen. Isinya tidak ditampilkan secara langsung di halaman web.
  - a. `<meta charset="UTF-8">`: Ini menentukan pengaturan karakter yang digunakan dalam dokumen sebagai UTF-8, yang merupakan standar yang paling umum digunakan untuk karakter Unicode.
  - b. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Ini menentukan bagaimana browser harus menyesuaikan tampilan halaman untuk berbagai perangkat dan lebar layar. Dalam hal ini, itu mengatakan kepada browser untuk menggunakan lebar perangkat dan mengatur tingkat zoom awal ke 1.0.
  - c. `<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/all.min.css" rel="stylesheet">`: Ini adalah referensi ke file CSS eksternal yang mengandung gaya font Awesome, yang digunakan untuk ikon yang akan ditampilkan di halaman web.
  - d. `<link rel="stylesheet" href="/assets/style.css">`: Ini adalah referensi ke file CSS lokal yang disebut `style.css` yang berada di folder `assets`. Ini adalah tempat dimana Anda biasanya menyimpan aturan gaya untuk mengatur tata letak dan penampilan halaman web.
4. `<title>Web Design</title>`: Ini adalah judul dari halaman web yang akan ditampilkan di tab browser atau dalam hasil pencarian. Dalam hal ini, judulnya adalah "Web Design".

## Header :

1. `<div class="nav-container">`: Ini adalah sebuah div (bagian dalam dokumen HTML) yang memiliki kelas CSS bernama "nav-container".
2. `<header>`: Ini adalah elemen header HTML. Biasanya, ini digunakan untuk menempatkan bagian kepala dari sebuah halaman web, seperti judul, logo, dan menu navigasi.
3. `<label for="hamburger">&#9776;</label>`: Ini adalah label untuk checkbox dengan id "hamburger". Simbol "≡" adalah ikon garis tiga yang sering digunakan sebagai simbol untuk menu pada perangkat mobile.
4. `<input type="checkbox" id="hamburger" />`: Ini adalah checkbox HTML yang digunakan untuk mengontrol tampilan dan penyembunyian menu navigasi. Saat dicentang (checked), menu akan ditampilkan, dan saat tidak dicentang (unchecked), menu akan disembunyikan.
5. `<div class="logo">`: Ini adalah div yang berisi logo situs web. Logo biasanya merupakan identitas visual dari situs web tersebut.
6. ``: Ini adalah tag gambar HTML yang menampilkan logo situs web. Atribut "src" menunjukkan lokasi file gambar, dalam hal ini "assets/images/logo.png". Atribut "alt" memberikan teks alternatif yang akan muncul jika gambar gagal dimuat atau untuk keperluan aksesibilitas.
7. `<nav>`: Ini adalah elemen nav HTML yang menandakan bagian dari halaman web yang berisi navigasi situs.
8. `<ul>`: Ini adalah elemen daftar tak berurut (unordered list) HTML yang digunakan untuk menyusun daftar menu navigasi.
9. `<li><a href="#" class="active">Home</a></li>`: Ini adalah item daftar dalam menu navigasi. Setiap item terdiri dari sebuah tautan (anchor tag) `<a>` dengan teks yang menunjukkan label menu, dalam hal ini "Home". Atribut "href" menunjukkan URL yang dituju (dalam contoh ini, "#" menunjukkan tautan ke halaman yang sama), dan atribut "class" diberikan nilai "active" untuk menunjukkan bahwa ini adalah halaman aktif atau halaman yang sedang dilihat oleh pengguna.
10. Sampai dengan `<li><a href="#">PHP</a></li>`: Ini adalah item-item lain dalam menu navigasi untuk halaman HTML, CSS, Javascript, dan PHP. Mereka memiliki struktur yang sama dengan item "Home" di atasnya, tetapi

menautkan ke halaman-halaman yang berbeda.

### Hero :

1. `<div class="hero-container">`: Ini adalah sebuah div (bagian dalam dokumen HTML) yang memiliki kelas CSS bernama "hero-container".
2. `<div class="hero-text">`: Ini adalah sebuah div yang berada di dalam hero-container dan memiliki kelas CSS bernama "hero-text".
3. `<h1>Pemrograman Web</h1>`: Ini adalah elemen heading level 1 (judul besar) yang menampilkan teks "Pemrograman Web". Biasanya, judul besar seperti ini digunakan untuk menarik perhatian pengguna dan mengkomunikasikan pesan utama dari halaman web tersebut.
4. `<p>Belajar HTML, CSS, Javascript dan PHP</p>`: Ini adalah elemen paragraf yang berisi teks "Belajar HTML, CSS, Javascript dan PHP". Paragraf ini memberikan penjelasan atau konteks tambahan terkait dengan judul besar di atasnya.
5. `<a href="#" class="button">Pelajari Lebih Lanjut</a>`: Ini adalah tautan (anchor tag) yang mengarah ke "#" (URL yang sama dengan halaman saat ini) dengan teks "Pelajari Lebih Lanjut". Kelas CSS "button". Tautan ini akan mengarah ke halaman lain yang berisi informasi lebih lanjut tentang topik yang disajikan dalam hero.

### Showcase :

1. `<div class="showcase-container">`: Ini adalah div yang membungkus setiap showcase.
2. `<section class="showcase">`: Ini adalah elemen section yang menandakan bagian dari halaman web yang berisi showcase. Setiap showcase memiliki kelas CSS bernama "showcase".
3. `<div class="showcase-image">`: Ini adalah div yang berisi gambar showcase.
4. ``: Ini adalah tag gambar HTML yang menampilkan gambar bunga. Atribut "src" menunjukkan lokasi file gambar, dan atribut "alt" memberikan teks alternatif untuk gambar tersebut.
5. `<div class="showcase-text">`: Ini adalah div yang berisi teks showcase.
6. `<h1>Lorem ipsum dolor sit amet consectetur</h1>`: Ini adalah elemen heading level 1 yang menampilkan judul showcase.



7. `<p>`: Ini adalah elemen paragraf HTML yang berisi teks lorem ipsum sebagai contoh teks penjelasan atau deskripsi dalam showcase. Dalam contoh ini, terdapat dua paragraf.
8. ``: Ini adalah tag gambar HTML yang menampilkan gambar bunga lainnya. Berbeda dengan gambar sebelumnya, gambar ini memiliki atribut "height" yang menentukan tinggi gambar. Atribut "alt" memberikan teks alternatif untuk gambar tersebut.

### Blog :

1. `<div class="blog-container">`: Ini adalah div yang mengelompokkan seluruh konten blog.
2. `<header>`: Ini adalah bagian kepala dari halaman blog. Biasanya, bagian ini berisi judul utama dari blog atau posting tertentu, berisi beberapa teks pendek yang memberikan pengantar atau ringkasan tentang posting tersebut.
3. `<h1>Lorem Ipsum</h1>`: Ini adalah judul utama dari blog atau posting. Ini memberikan informasi tentang topik yang dibahas dalam blog.
4. `<hr>`: Ini adalah elemen horizontal rule yang digunakan untuk membuat garis horizontal untuk memisahkan konten.
5. `<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dignissimos, vitae!</p>`: Ini adalah paragraf yang memberikan beberapa informasi pendek tentang konten blog.
6. `<div class="blog-index">`: Ini adalah div yang berisi daftar posting blog atau indeks blog.
7. `<div class="snippet">`: Ini adalah div yang mewakili satu potongan konten blog atau satu posting blog.
8. ``: Ini adalah gambar ilustrasi atau gambar yang relevan dengan konten blog. Kelas "snippet-img" yang digunakan untuk mengatur gaya visual dari gambar tersebut.
9. `<div class="snippet-txt">`: Ini adalah div yang berisi teks dari posting blog.
10. `<h2>Lorem Ipsum</h2>`: Ini adalah judul dari posting blog. Biasanya, judul ini lebih spesifik dari judul utama di header.

- 11.Paragraf-paragraf berikutnya (<p>...</p>) adalah bagian dari konten posting blog yang berisi penjelasan, deskripsi, atau pemikiran penulis terkait dengan topik yang dibahas.
- 12.<small>Lorem Ipsum</small>: Ini adalah elemen kecil yang berisi informasi tambahan, seperti penulis posting atau tanggal posting.

#### Footer :

1. <footer class="footer-container">: Ini adalah elemen footer yang memiliki kelas CSS bernama "footer-container". Elemen footer digunakan untuk menempatkan informasi yang ingin ditampilkan di bagian bawah halaman web.
2. <div class="footer-menu">: Ini adalah div yang mengelompokkan bagian-bagian utama dari footer, seperti logo, menu, dan informasi kontak.
3. <section>: Ini adalah elemen section yang digunakan untuk mengelompokkan konten-konten terkait dalam footer.
4. <div class="logo-container">: Ini adalah div yang digunakan untuk menempatkan logo situs web di dalamnya.
5. <a href="#">: Ini adalah tautan yang mengarah ke suatu halaman, dalam kasus ini, "#" menunjukkan tautan ke halaman yang sama.
6. : Ini adalah tag gambar HTML yang menampilkan logo situs web.
7. <p>Lorem ipsum dolor sit amet consectetur adipisicing elit...</p>: Ini adalah paragraf yang berisi deskripsi singkat tentang situs web atau perusahaan.
8. <h3>Belajar Web</h3>: Ini adalah elemen heading level 3 yang menampilkan judul untuk bagian menu.
9. <ul>: Ini adalah elemen daftar tak berurut (unordered list) yang berisi daftar tautan menu.
- 10.<li><a href="#">HTML</a></li>: Ini adalah item daftar dalam menu. Setiap item memiliki tautan ke halaman terkait.
- 11.<hr>: Ini adalah elemen horizontal rule yang digunakan untuk memisahkan bagian menu dengan bagian hak cipta.
- 12.<div class="footer-copyright">: Ini adalah div yang mengelompokkan bagian hak cipta dan tautan media sosial.

13.<small>&copy; Pemrograman Web 2024</small>: Ini adalah teks yang menunjukkan hak cipta situs web dan tahun pembuatan.

14. Tautan media sosial seperti Facebook, Twitter, Instagram, dll., ditampilkan menggunakan ikon dari font awesome.

Code :

CSS : stlye.css

```
:root {
  --main-color: #ffd95c;
  --main-color-hover: #e6c34f;
  --dark-color: #2b2b2b;
  --container-width: 1100px;
}

* {
  box-sizing: border-box;
}

html,
body {
  padding: 0;
  margin: 0;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

h1 {
  padding: 0;
  margin: 0;
  font-size: 2rem;
}

.button {
  text-decoration: none;
  display: inline-block;
  margin-top: 1.3em;
  padding: 0.7em 1.2em;
  background-color: var(--main-color);
  color: #464646;
  border-radius: 30px;
  font-size: 1.1em;
}

.button:hover {
  background-color: var(--main-color-hover);
  color: #2e2a2a;
  transition: all 0.3s;
}
```

```
/* HEADER */
.nav-container {
  background-color: var(--dark-color);
  position: sticky;
  top: 0px;
  z-index: 100;
}

.nav-container header {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
  width: 80%;
  height: 60px;
  margin: 0 auto;
}

.nav-container .logo img {
  width: 180px;
  padding-top: 10px;
}

.nav-container ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  display: flex;
}

.nav-container li a {
  display: block;
  color: white;
  padding: 0 1.1em;
  text-decoration: none;
  text-transform: uppercase;
  height: 60px;
  line-height: 60px;
}

.nav-container li a:hover {
  color: var(--main-color);
  transition: all 0.3s;
  border-bottom: 5px solid var(--main-color);
}

.nav-container header label,
#hamburger {
  display: none;
}
```

```
/* END OF HEADER */

/* HERO */
.hero-container {
  background-image: linear-gradient(rgba(0, 0, 0, 0.1),
    rgba(0, 0, 0, 0.3)), url("images/inaxuma.jpg");
  height: 70vh;
  background-position: center;
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
  display: flex;
  align-items: center;
  justify-content: center;
}

.hero-text {
  text-align: center;
  color: white;
  padding: 1rem;
}

.hero-text h1 {
  font-size: 4em;
}

.hero-text p {
  font-size: 2em;
  margin: 0;
  color: #999;
}

/* END OF HERO */

/* SHOWCASE */
.showcase-container:nth-child(even) {
  background-color: #f0f0f0;
}

.showcase {
  display: flex;
  justify-content: space-between;
  max-width: 1100px;
  margin: 0 auto;
  padding: 2em;
}

.showcase-image,
.showcase-text {
  width: 100%;
```

```
    margin: 1em;
}

.showcase-image img {
    width: 100%;
    border: 1px solid silver;
    padding: 0.3rem;
}

/* END OF SHOWCASE */

/* BLOG */
.blog-container {
    max-width: var(--container-width);
    margin: 3rem auto;
}

.blog-container header {
    padding: 0 2em;
    text-align: center;
}

.blog-container header hr {
    width: 25%;
    border: 0;
    border-top: 1px solid #ddd;
}

.blog-index {
    margin-top: 1rem;
    display: flex;
    flex-flow: row wrap;
    justify-content: space-between;
    padding: 0 2%;
}

.blog-index .snippet {
    width: 32%;
    margin-top: 1rem;
    border: 1px solid #ddd;
    border-radius: 0.25rem;
}

.blog-index .snippet-img {
    width: 100%;
}

.blog-index .snippet-txt {
    padding: 0 1em;
}
```

```
.blog-index .snippet-txt small {
  color: #999;
}

/* END OF BLOG */

/* FOOTER */
.footer-container {
  margin-top: 3rem;
  color: white;
  background-color: var(--dark-color);
  padding: 2rem 1rem;
}

.footer-container hr {
  color: #2e2a2a;
}

.footer-container a {
  text-decoration: none;
  color: white;
}

.footer-container a:hover {
  text-decoration: underline;
}

.footer-menu {
  max-width: var(--container-width);
  margin: 0 auto 2rem;
  display: flex;
  flex-flow: row wrap;
  justify-content: space-between;
}

.footer-menu section {
  width: 23%;
}

.footer-menu ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}

.footer-container ul li {
  margin-bottom: 0.5rem;
}

.footer-menu h3 {
  margin-top: 0;
}
```

```
.logo-container img {
  width: 180px;
}

.footer-copyright {
  max-width: var(--container-width);
  margin: 0 auto;
  display: flex;
  flex-flow: row wrap;
  justify-content: space-between;
  padding: 0.5rem;
}

.footer-copyright a {
  margin-left: 0.5rem;
}

.footer-copyright a:hover {
  text-decoration: none;
}

/* END OF FOOTER */

/* BREAKPOINT HANDPHONE */
@media screen and (max-width: 481px) {

  /* header */
  img {
    margin-top: 30px;
  }
  /* header stop */

  /* BLOG */
  .blog-container .snippet {
    width: 98%;
  }

  .blog-index {
    justify-content: center;
  }

  /* END OF BLOG */

  /* FOOTER */
  .footer-menu section {
    width: 100% !important;
    ;
    margin-bottom: 2rem;
  }

  /* END OF FOOTER */
}
```



```

}
/* BREAKPOINT TABLET */
@media screen and (max-width: 768px) {

    /* HEADER */
    .nav-container header label {
        display: inline-block;
        color: white;
        font-style: normal;
        font-size: 2rem;
        padding: 0.5rem;
        padding-left: 1rem;
    }

    .nav-container header {
        width: 100%;
    }

    .nav-container nav ul {
        flex-direction: column;
        width: 100%;
        position: absolute;
        top: 60px;
        left: 0;
    }

    .nav-container li a {
        display: block;
        color: white;
        background-color: var(--dark-color);
        padding: 0 2em;
        text-decoration: none;
        text-transform: uppercase;
        height: auto;
    }

    .nav-container li a:hover {
        border-bottom: none;
    }

    .nav-container .logo {
        position: absolute;
        right: 0px;
        z-index: 1000;
    }

    .nav-container .logo img {
        width: 80px;
    }

    .nav-container nav {
        display: none;
    }

```

```
}

.nav-container header input:checked~nav {
  display: flex;
}

/* END OF HEADER */
/* HERO */
.hero-text h1 {
  font-size: 3em;
}

/* END OF HERO */
/* SHOWCASE */
.showcase {
  flex-direction: column;
  padding: 1em 2em;
}

.showcase-image,
.showcase-text {
  width: auto;
}

/* END OF SHOWCASE */
/* BLOG */
.blog-index .snippet {
  width: 100% !important;
  margin: 1rem 2rem;
}

/* END OF BLOG */

/* FOOTER */
.footer-container {
  padding: 2rem;
}

.footer-menu section {
  width: 48% !important;
  margin-bottom: 2rem;
}

.footer-copyright div {
  margin-bottom: 2rem;
}

.footer-copyright a:first-child {
  margin-left: 0;
}

/* END OF FOOTER */
```

}

Penjelasan :

1. `:root`: Ini adalah sebuah pseudo-class yang digunakan untuk mengarahkan elemen root dari dokumen HTML, yaitu elemen `<html>`. Dalam konteks ini, itu digunakan untuk mendefinisikan variabel CSS (custom properties).
  - a. `--main-color`: Ini adalah variabel CSS yang didefinisikan di dalam root (`:root`) dan menampung nilai warna utama (main color) yang digunakan di dalam desain. Dalam kasus ini, warna utamanya adalah `#ffd95c` (sejenis kuning).
  - b. `--main-color-hover`: Ini adalah variabel CSS yang juga didefinisikan di dalam root dan menampung nilai warna utama saat tombol dihover (diposisikan pointer di atasnya). Nilainya adalah `#e6c34f` (kuning yang lebih gelap).
  - c. `--dark-color`: Variabel CSS lainnya yang menampung nilai warna gelap yang digunakan di dalam desain. Nilainya adalah `#2b2b2b` (hampir hitam).
2. `--container-width`: Ini adalah variabel CSS yang menentukan lebar kontainer dalam piksel (px). Dalam kasus ini, lebarnya adalah 1100px.
3. `*`: Ini adalah selector universal, yang berarti itu akan mempengaruhi setiap elemen di dalam dokumen HTML. Di sini, digunakan untuk mengatur box-sizing menjadi border-box untuk semua elemen. Ini memastikan bahwa lebar dan tinggi elemen termasuk padding dan border, bukan hanya konten.
4. `html, body`: Ini adalah selector yang memilih elemen `<html>` dan `<body>`. Dalam kasus ini, digunakan untuk mengatur margin dan padding keduanya menjadi 0, serta menetapkan jenis font yang akan digunakan dalam halaman.
5. `h1`: Ini adalah selector untuk elemen `<h1>`, yang biasanya merupakan judul terbesar dalam halaman.
  - a. Di sini, digunakan untuk mengatur margin dan paddingnya menjadi 0,
  - b. serta mengatur ukuran fontnya menjadi 2 rem.
6. `.button`: Ini adalah selector untuk kelas CSS yang disebut `.button`, yang digunakan untuk menargetkan tombol di dalam dokumen. Dalam kasus ini, digunakan untuk menetapkan berbagai properti, termasuk tata letak, warna latar belakang, warna teks, border-radius (untuk sudut tombol yang dibulatkan), dan ukuran font.

7. `.button:hover`: Ini adalah pseudo-class `:hover` yang diterapkan saat pointer berada di atas elemen `.button`. Dalam kasus ini, digunakan untuk mengubah warna latar belakang dan warna teks tombol saat tombol dihover, serta memberikan efek transisi dengan durasi 0.3 detik menggunakan properti `transition`.

#### Header :

1. `.nav-container`: Ini adalah selector untuk kelas CSS yang disebut `.nav-container`, yang digunakan untuk menargetkan kontainer navigasi di dalam dokumen.
  - a. `background-color`: Menentukan warna latar belakang dari kontainer navigasi. Di sini, warnanya diambil dari variabel `--dark-color` yang telah didefinisikan sebelumnya.
  - b. `position: sticky`: Membuat elemen tetap di posisi tertentu saat digulir. Di sini, posisinya akan tetap saat user menggulir ke atas, karena nilai `top` juga didefinisikan.
  - c. `top: 0px`: Menentukan jarak dari bagian atas viewport (area yang terlihat oleh pengguna) tempat kontainer navigasi akan tetap saat digulir. Dalam kasus ini, kontainer navigasi akan tetap menempel di bagian atas.
  - d. `z-index: 100`: Menentukan tingkat kedalaman elemen, yang menentukan bagaimana elemen tumpukannya di atas atau di bawah elemen lain. Dalam hal ini, kontainer navigasi memiliki `z-index 100`, sehingga akan muncul di depan elemen lain.
2. `.nav-container header`: Ini adalah selector untuk elemen `<header>` yang berada di dalam `.nav-container`.
  - a. `display: flex`: Mengatur tata letak elemen header menjadi flexbox, yang memungkinkan penggunaan fleksibilitas dan penataan elemen anak secara lebih mudah.
  - b. `flex-direction: row`: Menentukan arah tata letak flexbox menjadi baris (horizontal).
  - c. `justify-content: space-between`: Menentukan cara flex items (anak-anak header) didistribusikan di sepanjang sumbu utama (horizontal). Dalam hal ini, mereka akan didistribusikan dengan ruang di antara mereka.

- d. width: 80%: Menentukan lebar header menjadi 80% dari lebar parentnya.
  - e. height: 60px: Menentukan tinggi header menjadi 60 piksel.
  - f. margin: 0 auto: Mengatur margin atas dan bawah menjadi 0, dan sisi kiri dan kanan menjadi otomatis (menengahkan elemen header di dalam kontainer).
3. .nav-container .logo img: Ini adalah selector untuk gambar logo yang berada di dalam .nav-container.
- a. width: 180px: Menentukan lebar gambar logo menjadi 180 piksel.
  - b. padding-top: 10px: Menambahkan padding ke bagian atas gambar logo sebesar 10 piksel.
4. .nav-container ul: Ini adalah selector untuk daftar tak berurut (unordered list) di dalam .nav-container.
- a. list-style-type: none: Menghilangkan gaya default (bullet points) dari daftar tak berurut.
  - b. margin: 0: Mengatur margin menjadi 0 untuk menghapus ruang tambahan.
  - c. padding: 0: Mengatur padding menjadi 0 untuk menghapus ruang tambahan.
  - d. display: flex: Mengatur tata letak daftar tak berurut menjadi flexbox.
5. .nav-container li a: Ini adalah selector untuk tautan (link) di dalam item daftar (<li>) di dalam .nav-container.
- a. display: block: Mengubah tautan menjadi blok, sehingga dapat mengisi seluruh lebar item daftar.
  - b. color: white: Menentukan warna teks menjadi putih.
  - c. padding: 0 1.1em: Memberikan padding sebesar 0 di atas dan bawah, serta 1.1em di sisi kiri dan kanan.
  - d. text-decoration: none: Menghilangkan dekorasi tautan bawaan, seperti garis bawah.
  - e. text-transform: uppercase: Mengubah teks menjadi huruf kapital.

- f. `height: 60px`: Menentukan tinggi tautan menjadi 60 piksel.
  - g. `line-height: 60px`: Menyesuaikan tinggi baris agar tautan berada di tengah secara vertikal di dalam kotak tautan.
6. `.nav-container li a:hover`: Ini adalah selector untuk tautan saat dihover di dalam item daftar di dalam `.nav-container`.
- a. `color: var(--main-color)`: Mengubah warna teks menjadi warna yang ditentukan oleh variabel `--main-color`.
  - b. `transition: all 0.3s`: Menambahkan efek transisi ke semua perubahan properti CSS selama 0.3 detik, memberikan efek yang halus saat hover.
  - c. `border-bottom: 5px solid var(--main-color)`: Menambahkan garis bawah berwarna yang ditentukan oleh variabel `--main-color` saat tautan dihover.
7. `.nav-container header label` dan `#hamburger`: Keduanya digunakan untuk menyembunyikan label dan elemen dengan ID hamburger

### Hero :

1. `.hero-container`: Ini adalah selector untuk kontainer hero di dalam dokumen.
- a. `Background-image`: Menentukan gambar latar belakang untuk kontainer hero. Di sini, digunakan linear gradient yang meliputi dua warna rgba (hitam dengan transparansi 0.1 dan hitam dengan transparansi 0.3) sebagai lapisan transparan, diikuti oleh gambar latar belakang dari file "background.jpg" yang berada di dalam folder "images".
  - b. `height`: Menentukan tinggi kontainer hero menjadi 70% dari tinggi viewport (vh).
  - c. `background-position`: Menentukan posisi gambar latar belakang agar berada di tengah kontainer hero.
  - d. `background-repeat`: Mengatur gambar latar belakang agar tidak diulang.
  - e. `background-attachment`: Menetapkan apakah gambar latar belakang akan digulir dengan elemen atau akan tetap dalam posisi yang sama saat halaman digulir. Di sini, diatur sebagai tetap (fixed), sehingga

gambar latar belakang akan tetap di posisinya saat pengguna menggulir halaman.

- f. `background-size`: Menyesuaikan ukuran gambar latar belakang agar mencakup seluruh area kontainer hero.
  - g. `display: flex`: Mengatur tata letak kontainer hero menjadi flexbox.
  - h. `align-items: center`: Menengahkan konten secara vertikal di dalam kontainer hero.
  - i. `justify-content: center`: Menengahkan konten secara horizontal di dalam kontainer hero.
2. `.hero-text`: Ini adalah selector untuk teks hero di dalam kontainer hero.
- a. `text-align`: Mengatur penataan teks menjadi terpusat di dalam kontainer hero.
  - b. `color`: Menentukan warna teks menjadi putih.
  - c. `padding`: Memberikan padding sebesar 1 rem (unit yang setara dengan ukuran font elemen root) di sekitar teks.
  - d. `.hero-text h1`: Ini adalah selector untuk elemen heading (judul) di dalam teks hero.
  - e. `font-size`: Menentukan ukuran font h1 menjadi 4em (empat kali ukuran font standar).
3. `.hero-text p`: Ini adalah selector untuk elemen paragraf di dalam teks hero.
- a. `font-size`: Menentukan ukuran font paragraf menjadi 2em (dua kali ukuran font standar).
  - b. `margin`: Menghapus margin bawaan paragraf.
  - c. `color`: Menentukan warna teks paragraf menjadi abu-abu (#999).

### Showcase :

1. `.showcase-container:nth-child(even)`: Ini adalah selector untuk setiap kontainer showcase yang merupakan elemen ke-genap dalam urutan mereka di dalam dokumen.
- a. `background-color`: Menentukan warna latar belakang untuk kontainer showcase yang merupakan elemen ke-genap. Di sini, warnanya adalah #f0f0f0 (abu-abu muda).

2. `.showcase`: Ini adalah selector untuk konten showcase di dalam dokumen.
  - a. `display: flex`: Mengatur tata letak konten showcase menjadi flexbox.
  - b. `justify-content: space-between`: Menentukan cara flex items (anak-anak showcase) didistribusikan di sepanjang sumbu utama (horizontal). Dalam hal ini, mereka akan didistribusikan dengan ruang di antara mereka.
  - c. `max-width: 1100px`: Menetapkan lebar maksimum konten showcase menjadi 1100 piksel.
  - d. `margin: 0 auto`: Mengatur margin atas dan bawah menjadi 0, dan sisi kiri dan kanan menjadi otomatis (menengahkan konten showcase di dalam dokumen).
  - e. `padding: 2em`: Memberikan padding sebesar 2 em (unit yang setara dengan ukuran font elemen root) di sekitar konten showcase.
3. `.showcase-image`, `.showcase-text`: Ini adalah selector untuk elemen gambar dan teks di dalam konten showcase.
  - a. `width: 100%`: Menetapkan lebar elemen gambar dan teks menjadi 100% dari lebar konten showcase.
  - b. `margin: 1em`: Memberikan margin sebesar 1 em di sekitar elemen gambar dan teks.
4. `.showcase-image img`: Ini adalah selector untuk gambar di dalam elemen gambar di dalam konten showcase.
  - a. `width: 100%`: Menetapkan lebar gambar menjadi 100% dari lebar elemen gambar.
  - b. `border: 1px solid silver`: Menetapkan border dengan ketebalan 1 piksel dan warna silver (perak) untuk gambar.
  - c. `padding: 0.3rem`: Memberikan padding sebesar 0.3 rem di sekitar gambar.

### Blog :

1. `.blog-container`: Ini adalah selector untuk kontainer blog di dalam dokumen.
  - a. `max-width: var(--container-width)`: Menetapkan lebar maksimum kontainer blog sesuai dengan nilai variabel `--container-width` yang



telah didefinisikan sebelumnya.

- b. `margin: 3rem auto`: Memberikan margin sebesar 3 rem di atas dan bawah serta otomatis di sisi kiri dan kanan untuk menengahkan kontainer blog di dalam dokumen.
2. `.blog-container header`: Ini adalah selector untuk elemen header di dalam kontainer blog.
  - a. `padding: 0 2em`: Memberikan padding sebesar 0 di atas dan bawah serta 2 em di sisi kiri dan kanan di dalam elemen header.
  - b. `text-align: center`: Mengatur penataan teks di dalam header menjadi terpusat.
3. `.blog-container header hr`: Ini adalah selector untuk elemen horizontal rule (garis horizontal) di dalam header kontainer blog.
  - a. `width: 25%`: Menetapkan lebar garis horizontal menjadi 25% dari lebar parentnya.
  - b. `border: 0`: Menghilangkan border default dari elemen `hr`.
  - c. `border-top: 1px solid #ddd`: Menetapkan border pada sisi atas elemen `hr` dengan ketebalan 1 piksel dan warna #ddd (abu-abu muda).
4. `.blog-index`: Ini adalah selector untuk indeks blog di dalam kontainer blog.
  - a. `margin-top: 1rem`: Memberikan margin sebesar 1 rem di bagian atas indeks blog.
  - b. `display: flex`: Mengatur tata letak indeks blog menjadi flexbox.
  - c. `flex-flow: row wrap`: Mengatur arah tata letak flexbox menjadi baris (horizontal) dengan wrap (pembungkus) item jika tidak cukup tempat.
  - d. `justify-content: space-between`: Menentukan cara flex items (anak-anak indeks blog) didistribusikan di sepanjang sumbu utama (horizontal) dengan ruang di antara mereka.
  - e. `padding: 0 2%`: Memberikan padding sebesar 0 di atas dan bawah serta 2% di sisi kiri dan kanan di dalam indeks blog.
5. `.blog-index .snippet`: Ini adalah selector untuk setiap potongan blog (snippet) di dalam indeks blog.
  - a. `width: 32%`: Menetapkan lebar potongan blog menjadi 32% dari lebar

parentnya.

- b. `margin-top: 1rem`: Memberikan margin sebesar 1 rem di bagian atas potongan blog.
  - c. `border: 1px solid #ddd`: Menetapkan border pada potongan blog dengan ketebalan 1 piksel dan warna #ddd (abu-abu muda).
  - d. `border-radius: 0.25rem`: Memberikan sudut yang sedikit bulat pada potongan blog.
6. `.blog-index .snippet-img`: Ini adalah selector untuk gambar di dalam potongan blog.
- a. `width: 100%`: Menetapkan lebar gambar menjadi 100% dari lebar elemen gambar di dalam potongan blog.
7. `.blog-index .snippet-txt`: Ini adalah selector untuk teks di dalam potongan blog.
- a. `padding: 0 1em`: Memberikan padding sebesar 0 di atas dan bawah serta 1 em di sisi kiri dan kanan di dalam teks potongan blog.
8. `.blog-index .snippet-txt small`: Ini adalah selector untuk teks kecil di dalam potongan blog.
- a. `color: #999`: Menetapkan warna teks kecil menjadi #999 (abu-abu tua).

#### Footer :

1. `.footer-container`: Ini adalah selector untuk kontainer footer di dalam dokumen.
- a. `margin-top: 3rem`: Memberikan margin sebesar 3 rem di bagian atas kontainer footer.
  - b. `color: white`: Menetapkan warna teks menjadi putih.
  - c. `background-color: var(--dark-color)`: Menetapkan warna latar belakang kontainer footer sesuai dengan variabel `--dark-color` yang telah didefinisikan sebelumnya.
  - d. `padding: 2rem 1rem`: Memberikan padding sebesar 2 rem di atas dan bawah serta 1 rem di sisi kiri dan kanan di dalam kontainer footer.
2. `footer-container hr`: Ini adalah selector untuk elemen horizontal rule (garis horizontal) di dalam kontainer footer.
- a. `color: #2e2a2a`: Menetapkan warna garis horizontal menjadi #2e2a2a

(hampir hitam).

3. `.footer-container a`: Ini adalah selector untuk tautan (link) di dalam kontainer footer.
  - a. `text-decoration: none`: Menghilangkan dekorasi tautan bawaan, seperti garis bawah.
  - b. `color: white`: Menetapkan warna teks tautan menjadi putih.
4. `.footer-container a:hover`: Ini adalah selector untuk tautan saat dihover di dalam kontainer footer.
  - a. `text-decoration: underline`: Menggarisbawahi teks tautan saat dihover.
5. `.footer-menu`: Ini adalah selector untuk menu footer di dalam kontainer footer.
  - a. `max-width: var(--container-width)`: Menetapkan lebar maksimum menu footer sesuai dengan variabel `--container-width` yang telah didefinisikan sebelumnya.
  - b. `margin: 0 auto 2rem`: Memberikan margin 0 di atas dan bawah serta otomatis di sisi kiri dan kanan untuk menengahkan menu footer di dalam kontainer footer.
  - c. `display: flex`: Mengatur tata letak menu footer menjadi flexbox.
  - d. `flex-flow: row wrap`: Mengatur arah tata letak flexbox menjadi baris (horizontal) dengan wrap (pembungkus) item jika tidak cukup tempat.
  - e. `justify-content: space-between`: Menentukan cara flex items (anak-anak menu footer) didistribusikan di sepanjang sumbu utama (horizontal) dengan ruang di antara mereka.
6. `.footer-menu section`: Ini adalah selector untuk setiap bagian menu di dalam menu footer.
  - a. `width: 23%`: Menetapkan lebar setiap bagian menu menjadi 23% dari lebar parentnya.
7. `.footer-menu ul`: Ini adalah selector untuk daftar tak berurut (unordered list) di dalam menu footer.
  - a. `list-style-type: none`: Menghilangkan gaya default (bullet points) dari daftar tak berurut.
  - b. `margin: 0`: Mengatur margin menjadi 0 untuk menghapus ruang

tambahan.

- c. padding: 0: Mengatur padding menjadi 0 untuk menghapus ruang tambahan.

8. .footer-container ul li: Ini adalah selector untuk setiap elemen daftar (list item) di dalam kontainer footer.

- a. margin-bottom: 0.5rem: Memberikan margin sebesar 0.5 rem di bagian bawah setiap elemen daftar.

9. .footer-menu h3: Ini adalah selector untuk elemen heading (judul) di dalam menu footer.

- a. margin-top: 0: Menghilangkan margin atas dari elemen heading.

10..logo-container img: Ini adalah selector untuk gambar logo di dalam kontainer logo.

- a. width: 180px: Menetapkan lebar gambar logo menjadi 180 piksel.

11..footer-copyright: Ini adalah selector untuk hak cipta di dalam footer.

- a. max-width: var(--container-width): Menetapkan lebar maksimum hak cipta sesuai dengan variabel --container-width yang telah didefinisikan sebelumnya.
- b. margin: 0 auto: Memberikan margin 0 di atas dan bawah serta otomatis di sisi kiri dan kanan untuk menengahkan hak cipta di dalam kontainer footer.
- c. display: flex: Mengatur tata letak hak cipta menjadi flexbox.
- d. flex-flow: row wrap: Mengatur arah tata letak flexbox menjadi baris (horizontal) dengan wrap (pembungkus) item jika tidak cukup tempat.
- e. justify-content: space-between: Menentukan cara flex items (anak-anak hak cipta) didistribusikan di sepanjang sumbu utama (horizontal) dengan ruang di antara mereka.
- f. padding: 0.5rem: Memberikan padding sebesar 0.5 rem di sekitar hak cipta.

12..footer-copyright a: Ini adalah selector untuk tautan (link) di dalam hak cipta.

- a. margin-left: 0.5rem: Memberikan margin kiri sebesar 0.5 rem di sebelah kiri setiap tautan di dalam hak cipta.

13..footer-copyright a:hover: Ini adalah selector untuk tautan saat dihover di dalam hak cipta.

- a. text-decoration: none: Menghilangkan dekorasi tautan bawaan, seperti garis bawah, saat tautan dihover.

### Handphone (481 px ) :

@media screen and (max-width: 481px) : adalah sebuah aturan CSS yang diterapkan pada layar dengan lebar maksimum 481 piksel. Ini merupakan bagian dari CSS Media Queries, yang memungkinkan pengembang web untuk menyesuaikan tata letak dan gaya halaman web sesuai dengan karakteristik layar yang berbeda, seperti lebar, tinggi, orientasi, dan lainnya.

#### 1. HEADER:

- a. img: Ini adalah selector untuk semua elemen gambar di dalam dokumen.
  - i. margin-top: 30px;; Menetapkan margin atas sebesar 30 piksel untuk semua gambar di dalam dokumen saat lebar layar kurang dari atau sama dengan 481 piksel.

#### 2. BLOG:

- a. .blog-container .snippet: Ini adalah selector untuk potongan-potongan blog di dalam kontainer blog.
  - i. width: 98%;: Menetapkan lebar potongan blog menjadi 98% dari lebar kontainer blog, sehingga potongan blog akan hampir mengisi seluruh lebar kontainer saat tampil di layar handphone.
  - ii. .blog-index: Ini adalah selector untuk indeks blog di dalam kontainer blog.
  - iii. justify-content: center;; Mengatur tata letak indeks blog agar potongan-potongan blog menjadi terpusat secara horizontal.

#### 3. FOOTER:

- a. .footer-menu section: Ini adalah selector untuk setiap bagian dari menu footer.
  - i. width: 100% !important;; Menetapkan lebar setiap bagian menu footer menjadi 100% dari lebar parentnya, sehingga setiap bagian menu akan menempati seluruh lebar layar.

- ii. `margin-bottom: 2rem;`: Memberikan margin bawah sebesar 2 rem di antara setiap bagian menu footer untuk memberikan ruang yang cukup di antara mereka.

### Tablet ( 768px ):

@media screen and (max-width: 768px) : adalah sebuah aturan CSS yang diterapkan pada layar dengan lebar maksimum 481 piksel. Ini merupakan bagian dari CSS Media Queries, yang memungkinkan pengembang web untuk menyesuaikan tata letak dan gaya halaman web sesuai dengan karakteristik layar yang berbeda, seperti lebar, tinggi, orientasi, dan lainnya.

#### 1. HEADER:

- a. `.nav-container header label`: Mengatur tampilan label di dalam header.
  - i. `display: inline-block;`: Menjadikan label tampil sebagai blok yang dapat disusun dalam satu baris dengan konten lainnya.
  - ii. `color: white;`: Menetapkan warna teks label menjadi putih.
  - iii. `font-style: normal;`: Menetapkan gaya font normal untuk label.
  - iv. `font-size: 2rem;`: Menetapkan ukuran font label menjadi 2 rem.
  - v. `padding: 0.5rem;`: Memberikan padding sebesar 0.5 rem di sekitar label.
  - vi. `padding-left: 1rem;`: Memberikan padding sebesar 1 rem di sisi kiri label.
- b. `.nav-container header`: Mengatur tata letak header.
  - i. `width: 100%;`: Menjadikan header memiliki lebar maksimum 100% dari parentnya.
- c. `.nav-container nav ul`: Mengatur daftar tautan navigasi.
  - i. `flex-direction: column;`: Mengatur tautan navigasi menjadi berorientasi kolom (vertikal).
  - ii. `width: 100%;`: Memberikan lebar maksimum 100% pada daftar tautan navigasi.
  - iii. `position: absolute;`: Menjadikan posisi daftar tautan navigasi menjadi absolut.

- iv. `top: 60px;`: Memberikan jarak 60 piksel dari atas untuk daftar tautan navigasi.
- v. `left: 0;`: Menjadikan posisi daftar tautan navigasi berada di sebelah kiri.
- d. `.nav-container li a`: Mengatur tampilan tautan di dalam header.
  - i. `display: block;`: Menjadikan tautan tampil sebagai blok.
  - ii. `color: white;`: Menetapkan warna teks tautan menjadi putih.
  - iii. `background-color: var(--dark-color);`: Menetapkan warna latar belakang tautan sesuai dengan variabel `--dark-color`.
  - iv. `padding: 0 2em;`: Memberikan padding sebesar 0 di atas dan bawah serta 2 em di sisi kiri dan kanan di dalam tautan.
  - v. `text-decoration: none;`: Menghilangkan dekorasi tautan bawaan.
  - vi. `text-transform: uppercase;`: Mengubah teks tautan menjadi huruf besar.
  - vii. `height: auto;`: Menyesuaikan tinggi tautan secara otomatis.
- e. `.nav-container li a:hover`: Mengatur tampilan tautan saat dihover di dalam header.
  - i. `border-bottom: none;`: Menghilangkan garis bawah saat tautan dihover.

## 2. HERO:

- a. `.hero-text h1`: Mengatur tampilan judul h1 di dalam hero.
  - i. `font-size: 3em;`: Menetapkan ukuran font h1 menjadi 3 em.

## 3. SHOWCASE:

- a. `.showcase`: Mengatur tampilan showcase.
  - i. `flex-direction: column;`: Mengatur showcase menjadi berorientasi kolom (vertikal).
  - ii. `padding: 1em 2em;`: Memberikan padding sebesar 1 em di atas dan bawah serta 2 em di sisi kiri dan kanan di dalam showcase.
- b. `.showcase-image, .showcase-text`: Mengatur tampilan gambar dan teks di

dalam showcase.

- i. `width: auto;` Mengatur lebar gambar dan teks secara otomatis.

#### 4. BLOG:

- a. `.blog-index .snippet`: Mengatur tampilan potongan blog di dalam indeks blog.
  - i. `width: 100% !important;` Menetapkan lebar potongan blog menjadi 100% dari lebar parentnya dengan `!important` untuk menindih aturan lain yang sudah ada sebelumnya.
  - ii. `margin: 1rem 2rem;` Memberikan margin sebesar 1 rem di atas dan bawah serta 2 rem di sisi kiri dan kanan di dalam potongan blog.

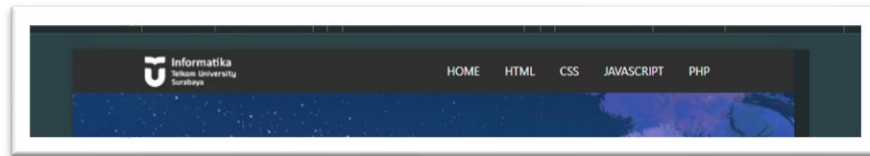
#### 5. FOOTER:

- a. `.footer-container`: Mengatur tampilan kontainer footer.
  - i. `padding: 2rem;` Memberikan padding sebesar 2 rem di sekitar kontainer footer.
- b. `.footer-menu section`: Mengatur tampilan setiap bagian dari menu footer.
  - i. `width: 48% !important;` Menetapkan lebar setiap bagian menu footer menjadi 48% dari lebar parentnya dengan `!important` untuk menindih aturan lain yang sudah ada sebelumnya.
  - ii. `margin-bottom: 2rem;` Memberikan margin bawah sebesar 2 rem di antara setiap bagian menu footer.
- c. `.footer-copyright div`: Mengatur tampilan div di dalam hak cipta di footer.
  - i. `margin-bottom: 2rem;` Memberikan margin bawah sebesar 2 rem di antara setiap div di dalam hak cipta.
- d. `.footer-copyright a:first-child`: Mengatur tampilan tautan pertama di dalam hak cipta.
  - i. `margin-left: 0;` Menghilangkan margin kiri pada tautan pertama di dalam hak cipta.

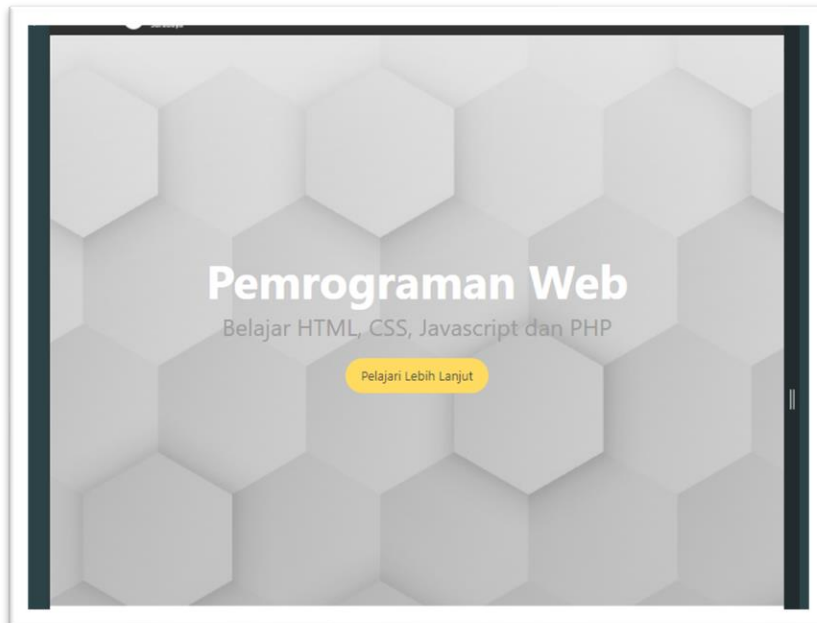
Output :



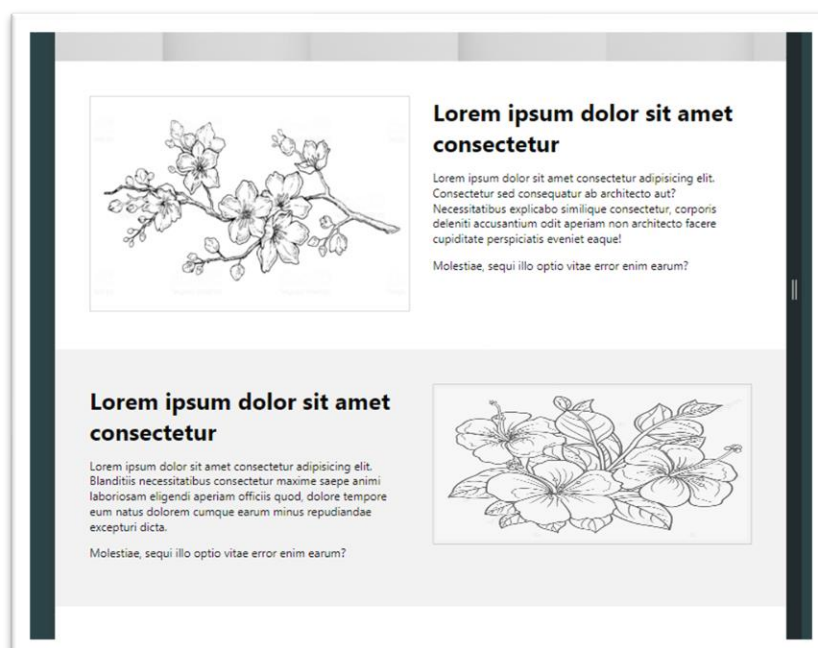
## Header Desktop:



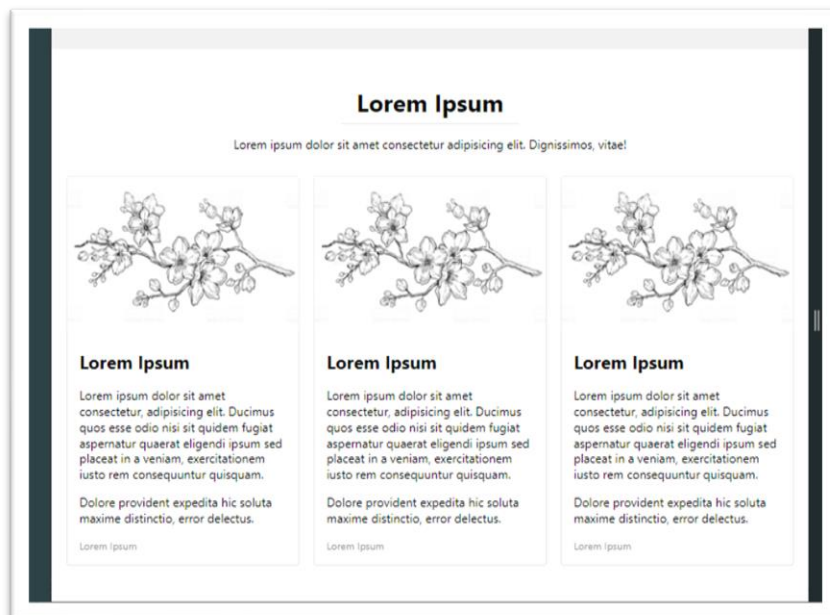
## Hero Desktop:



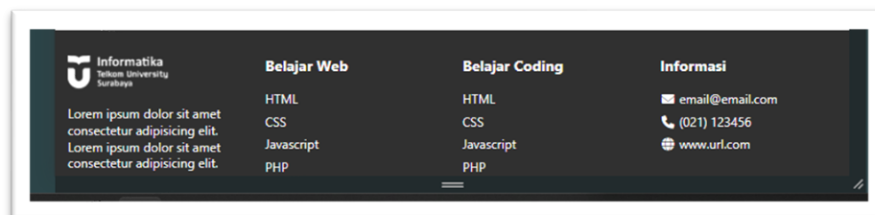
## Showcase Desktop:



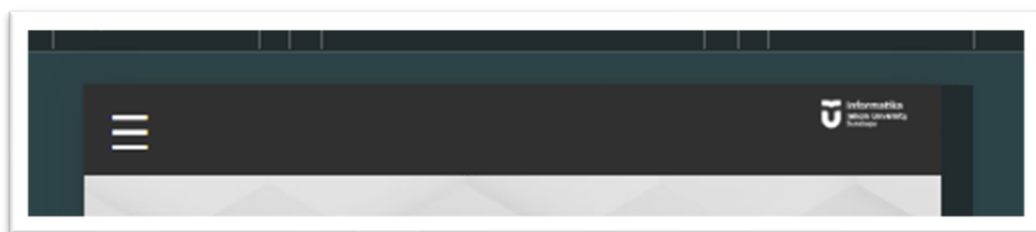
## Blog Desktop:



## Footer Desktop:



## Header Tablet :



Hero :



### Lorem ipsum dolor sit amet consectetur

Lorem ipsum dolor sit amet consectetur adipisicing elit. Consectetur sed consequatur ab architecto aut? Necessitatibus explicabo similique consectetur, corporis deleniti accusantium odit aperiam non architecto facere cupiditate perspiciatis eveniet eaque!

Molestiae, sequi illo optio vitae error enim earum?

### Lorem ipsum dolor sit amet consectetur

Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis necessitatibus consectetur maxime saepe animi laboriosam eligendi aperiam officiis quod, dolore tempore eum natus dolorem cumque earum minus repudiandae excepturi dicta.

Molestiae, sequi illo optio vitae error enim earum?



### Lorem Ipsum

Lorem ipsum dolor sit amet consectetur adipisicing elit. Dignissimos, vitae!




### Lorem Ipsum

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ducimus quos esse odio nisi sit quidem fugiat aspernatur quaeat eligendi ipsum sed placeat in a veniam, exercitationem iusto rem consequuntur quisquam.

## Showcase :

### Lorem Ipsum

Lorem ipsum dolor sit amet consectetur adipisicing elit. Dignissimos, vitae!





### Lorem Ipsum

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ducimus quos esse odio nisi sit quidem fugiat aspernatur quærat eligendi ipsum sed placeat in a veniam, exercitationem iusto rem consequuntur quisquam.

Dolore provident expedita hic soluta maxime distinctio, error delectus.

Lorem Ipsum






### Lorem Ipsum

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ducimus quos esse odio nisi sit quidem fugiat aspernatur quærat eligendi ipsum sed placeat in a veniam, exercitationem iusto rem consequuntur quisquam.

Dolore provident expedita hic soluta maxime distinctio, error delectus.

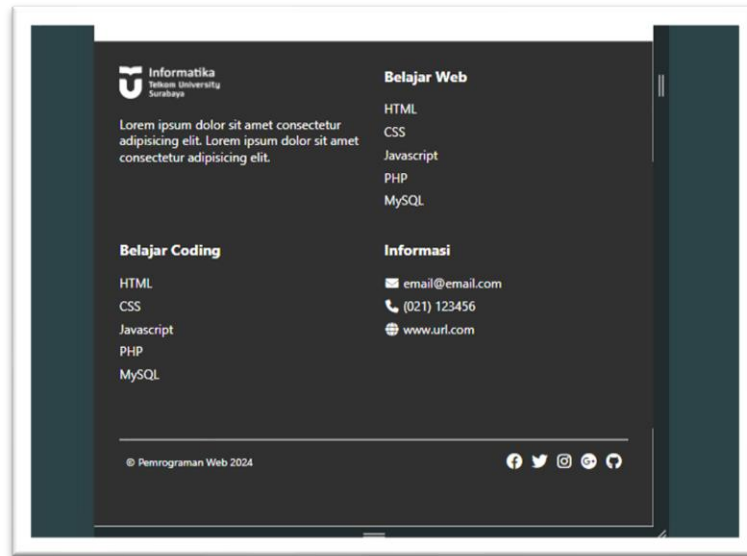
Lorem Ipsum



### Lorem Ipsum

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ducimus quos esse odio

### Footer Tablet :



### Tugas :

1. Buatlah File tugas.html seperti dibawah ini dan Buatlah kode CSS nya sehingga menghasilkan output seperti dibawah ini
2. Footer media sosial terlihat tampilan seperti diatas, perbaiki menu sosial media tersebut agar sesuai ukuran dan tulisannya
3. Buatlah menu jika dimaksimalkan halaman menu ada dibagian kiri seperti pada gambar poin
  - a. jika dikecilkan seperti layar tablet maka menu ada diatas seperti gambar dibawah ini
4. Dokumentasikan hasil praktikum tersebut **SEMUA (screenshot kode program, output padabrowser, penjelasan kode program yang ditulis)** dalam bentuk Laporan Praktikum.

### Code :

#### HTML : index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link rel="stylesheet" href="style_tugas.css" />
  <title>Web | CSS Tingkat Lanjut</title>
</head>

<body>
  <div class="container">
    <!-- NAV BAR START -->
```

```

<div class="sidebar">
  <nav>
    <ul>
      <li><a href="">About</a></li>
      <li><a href="">Profil</a></li>
      <li><a href="">HTML</a></li>
      <li><a href="">CSS</a></li>
    </ul>
  </nav>
</div>
<!-- NAV BAR ENDS -->

<!-- HERO START -->
<div class="containerHero">

  <main class="content">
    <section class="hero">
      <div class="hero-content">
        <div class="imgWh1">
          
          <h1>Pratikum Pemrograman WEB </h1>
        </div>
        <br />
        <div class="imgWh2">
          <h2>Universitas Telkom Surabaya</h2>
          
        </div>
        <p>
          Lorem ipsum dolor sit amet consectetur adipisicing
elit. Dignissimos, aperiam dolore
          assumenda velit repellendus recusandae magni
consectetur mollitia facere incidunt
          inventore perspiciatis debitis doloribus ullam minima
culpa voluptatem. Repellendus,
          option.
        </p>
        <a href="" class="action-btn">Profile Saya</a>
      </div>
    </section>
  </main>
<!-- HERO ENDS -->

<!-- FOOTER START -->
<div class="footer">
  <footer>
    <ul>
      <li>
        <a>
        <p>Instagram</p>

```

```

        </a>
    </li>
    <li>
        <a>
        <p> Facebook </p>
    </a>
    </li>
    <li>
        <a>
        <p> Twitter </p>
    </a>
    </li>
    <li>
        <a>
        <p> Telegram </p>
    </a>
    </li>
</ul>
</footer>
</div>
</div>
<!-- FOOTERS ENDS -->
</div>
</body>
</html>

```

#### Perubahan Code :

1. Penambahan class baru seperti imgWh1, imgWh2, topImg, dan containerHero yang Dimana :
2. imgWh1 : digunakan untuk meletakkan img dan h1 pada div yang sama agar saat mode tablet bisa didapatkan img di atas h1
3. imgWh2 : digunakan untuk meletakkan imh dan h2 pada div yang sama agar bisa di letakan samping sampingan secara horizontal.
4. topImg : digunakan untuk menyembunyikan imh saat mode desktop dan memunculkanya saat mode tablet.
5. containerHero : digunakan untuk meletakkan hero dengan footer pada div yang sama agar container dan containerHero bisa dibuat flex kesamping.

#### Penjelasan Code :

HTML : index.html

#### Head :

1. <!DOCTYPE html>: Ini adalah deklarasi untuk menentukan jenis dokumen HTML yang digunakan, dalam hal ini, HTML5.
2. <html lang="en">: Tag pembuka untuk elemen HTML. Atribut lang="en" menandakan bahwa bahasa dokumen adalah bahasa Inggris.

3. `<head>`: Bagian ini berisi meta-informasi tentang dokumen HTML seperti karakter set, viewport, dan tautan ke CSS.
4. `<meta charset="UTF-8">`: Mendefinisikan pengkodean karakter dokumen sebagai UTF-8, yang merupakan standar untuk encoding Unicode.
5. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Mendefinisikan viewport yang memungkinkan halaman untuk meresponsifkan ukuran perangkat yang berbeda. Dalam hal ini, lebarnya akan sama dengan lebar perangkat, dan skala awalnya adalah 1.0.
6. `<link rel="stylesheet" href="style_tugas.css" />`: Ini adalah tag untuk menghubungkan dokumen dengan file CSS eksternal bernama "style\_tugas.css". Ini memungkinkan pengaturan gaya dan tata letak untuk halaman tersebut.
7. `<title>Web | CSS Tingkat Lanjut</title>`: Ini adalah judul dari halaman web yang akan ditampilkan di tab browser.

### Body :

1. `<div class="sidebar">`: Ini adalah sebuah div dengan kelas "sidebar". Pada dasarnya, ini adalah sebuah wadah untuk elemen-elemen navigasi yang akan ditampilkan di sisi halaman web. Kelas "sidebar" digunakan untuk menerapkan gaya khusus atau tata letak tertentu pada elemen ini.
2. `<nav>`: Ini adalah elemen HTML yang menandakan bagian navigasi. Ini adalah bagian yang menaungi daftar tautan yang mengarah ke berbagai halaman atau bagian halaman web.
3. `<ul>`: Ini adalah elemen HTML yang menandakan daftar tak terurut (unordered list). Di dalamnya, kita dapat menempatkan elemen-elemen `<li>` yang mewakili item-item dalam daftar.
4. `<li><a href="">About</a></li>`: Ini adalah satu item dalam daftar navigasi. Setiap item terdiri dari elemen `<li>` yang menandakan item daftar, dan di dalamnya terdapat tautan (`<a>`) yang mengarahkan pengguna ke halaman atau bagian halaman yang sesuai. Dalam contoh ini, "About" adalah teks yang ditampilkan untuk tautan tersebut.

### Hero :

1. `<div class="containerHero">`: Ini adalah sebuah div dengan kelas "containerHero". Biasanya, kelas "container" digunakan untuk mengelompokkan dan mengatur konten di dalamnya, dan dalam hal ini, "containerHero" menandakan bahwa ini adalah bagian utama dari halaman web.
2. `<main class="content">`: Ini adalah elemen `<main>` yang menandakan bagian utama konten halaman. Kelas "content" yang digunakan untuk



memberikan gaya atau tata letak tertentu pada bagian ini.

3. `<section class="hero">`: Ini adalah sebuah section dengan kelas "hero". Bagian ini digunakan untuk menampilkan konten utama dari halaman web, seperti judul besar, gambar, dan teks singkat yang menjelaskan tujuan atau fokus halaman.
4. `<div class="hero-content">`: Ini adalah sebuah div dengan kelas "hero-content" yang digunakan untuk mengelompokkan konten dalam bagian hero tersebut.
5. `<div class="imgWh1">`: Ini adalah sebuah div dengan kelas "imgWh1". Kelas ini digunakan untuk menerapkan gaya atau tata letak tertentu pada elemen-elemen di dalamnya.
6. ``: Ini adalah elemen gambar yang menampilkan logo atau gambar yang relevan. Atribut class="topImg" yang digunakan untuk memberikan gaya khusus pada gambar ini.
7. `<h1>Pratikum Pemrograman WEB </h1>`: Ini adalah judul utama atau heading level 1 (H1) yang menandakan topik atau tujuan utama dari halaman web.
8. `<div class="imgWh2">`: Ini adalah sebuah div dengan kelas "imgWh2" yang digunakan untuk mengelompokkan konten lainnya dalam bagian hero.
9. `<h2>Universitas Telkom Surabaya</h2>`: Ini adalah heading level 2 (H2) yang menandakan subjudul atau informasi tambahan yang relevan dengan halaman web.
10. `<p>`: Ini adalah elemen paragraf yang berisi teks deskripsi atau konten tambahan yang mendetailkan informasi yang diberikan sebelumnya.
11. `<a href="" class="action-btn">Profile Saya</a>`: Ini adalah tautan atau link yang mengarahkan pengguna ke halaman profil atau aksi tertentu. Kelas "action-btn" digunakan untuk memberikan gaya khusus pada tautan ini.

#### Footer :

1. `<div class="footer">`: Ini adalah sebuah div dengan kelas "footer", yang biasanya digunakan untuk mengelompokkan dan mengatur konten dalam bagian footer halaman web.
2. `<footer>`: Ini adalah elemen HTML yang menandakan bagian footer dari halaman web. Biasanya, ini digunakan untuk menempatkan informasi tambahan seperti tautan, kontak, hak cipta, atau kredit.
3. `<ul>`: Ini adalah elemen HTML yang menandakan daftar tak terurut (unordered list), yang digunakan di dalam footer untuk menampilkan daftar tautan atau informasi lainnya.
4. `<li>`: Ini adalah elemen HTML yang menandakan item dalam daftar footer.

5. ``: Ini adalah elemen gambar yang menampilkan ikon Instagram. Atribut `src` menentukan lokasi gambar, sementara atribut `alt` digunakan untuk memberikan teks alternatif untuk gambar (biasanya digunakan untuk aksesibilitas).
6. `<a>`: Ini adalah elemen tautan (link) yang mengelilingi teks "Instagram". Tautan ini mengarahkan pengguna ke halaman atau profil Instagram yang sesuai.
7. `<p>Instagram</p>`: Ini adalah elemen paragraf yang berisi teks "Instagram". digunakan sebagai label untuk tautan ke Instagram.

Code :  
CSS : style\_tugas.css

```
html {
  height: 100vh;
}

body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  margin: 0;
  padding: 0;
}

.container {
  display: flex;
  min-height: 100vh;
  max-height: fit-content;
  background-image: linear-gradient(to bottom, #924dbf, #0f0529);
}

/* Sidebar */
.sidebar {
  width: 13%;
  height: 100%;
  display: flex;
  flex-direction: column;
  justify-content: flex-start;
  align-items: center;
  padding-top: 20px;
}

.sidebar nav ul {
  margin-left: -5%;
  padding: 0;
  margin-top: 150px;
}

.sidebar nav ul li {
  align-items: center;
```

```

width: 55%;
list-style: none;
text-decoration: none;
padding: 0;
margin: 7vh 5vh;
border-radius: 5px;
}

.sidebar nav ul li:hover{
background-color: #924dbf;
animation-delay: 0.3ms;
}

.sidebar nav ul li a {
font-weight: 650;
display: flex;
justify-content: center;
text-decoration: none;
color: aliceblue;
font-size: 3vh;
margin: 10%;
}

/* Hero */
.containerHero {
min-height: 100vh;
max-height: fit-content;
display: flex;
flex-direction: column;
justify-content: flex-start;
align-items: center;
padding-top: 20px;
background-image: linear-gradient(to bottom, #e5e5e5, #9a9a9a);
}

.content {
width: 100vw;
}

.hero-content {
padding: 20px 0;
/* Sesuaikan dengan nilai padding yang diinginkan */
}

.content h1 {
font-size: 5vh;
font-weight: 350;
margin: 0% 0% 0% 0%;
display: flex;
justify-content: flex-start;
align-items: center;

```

```
padding-left: 2%;
}

.content h2 {
  font-size: 5vh;
  font-weight: 800;
  margin: 0% 3% 0% 3%;
}

.content p {
  font-size: 2vh;
  font-weight: 350;
  padding-left: 2%;
  margin-bottom: 50px;
  max-width: 50vw;
}

.imgWh1 {
  margin: 0;
  padding: 0;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}

.imgWh1 h1 {
  align-self: flex-start;
  margin-top: 100px;
}

.imgWh1 img {
  display: none;
  width: 30%;
  height: auto;
}

.imgWh2 {
  display: flex;
  justify-content: flex-start;
  align-items: center;
}

.imgWh2 img {
  min-width: 100px;
  width: 250px;
  height: auto;
  padding: 0px 50px 0px 50px;
}
```

```
.imgWh2 h2 {  
  
    padding-left: 2%;  
    margin: 0;  
}  
  
.action-btn {  
    border-radius: 28px;  
    color: #ffffff;  
    font-size: 15px;  
    background: #494848;  
    padding: 10px 30px 10px 30px;  
    margin-left: 2%;  
    text-decoration: none;  
}  
  
/* FOOTER */  
.footer {  
    width: 100%;  
    height: 10%;  
    padding: 20px 0;  
    margin-top: auto;  
    background-color: #494848;  
    color: white;  
    display: flex;  
    justify-content: center;  
    text-align: center;  
    position: relative;  
    bottom: 0;  
}  
  
.footer ul {  
    margin: 0;  
    padding: 0;  
    display: flex;  
    list-style: none;  
}  
  
.footer li {  
    display: flex;  
    flex-direction: column;  
    justify-content: center;  
    align-items: center;  
    margin-right: 10px;  
    width: 100px;  
    /* Adjust margin if needed */  
}  
  
.footer img {  
    width: 30px;  
    height: auto;  
    border-radius: 8px;
```

```
display: flex;
justify-content: center;
align-items: center;
}

.footer p {
font-size: 2vh;
}

@media screen and (max-width: 768px) {
.container{
display: block;
background-image: linear-gradient(to right, #fbf5a3d5, #924dbf);

}

.sidebar {
width: 100vw;
height: 100%;
display: flex;
flex-direction: row;
justify-content: center;
align-items: center;
padding-top: 0;
}

.sidebar nav ul {
padding: 0;
margin: 0;
display: flex;
justify-content: center;
align-items: center;
}

.sidebar nav ul li {
list-style: none;
text-decoration: none;
display: flex;
justify-content: center;
align-items: center;
padding: 1.5%;
margin: 3vh 3vh;
background-color: transparent;
}

.sidebar nav ul li a {
margin: 0;
font-weight: 650;
padding: 10%;
}

.sidebar nav ul li:hover{
background-color: transparent;
```

```
        animation-delay: 0.3ms;
    }
    .sidebar nav ul li a:hover{
        background-image: linear-gradient(to left, #f4efc4d5, #e5c7f9);
        animation-delay: 0.3ms;
        color: #924dbf;
        border-radius: 5px;
    }

    .containerHero{
        padding: 0;
    }

    .hero-content{
        display: flex;
        flex-direction: column;
        justify-content: center;
        align-items: center;
        margin-left: -2%;
    }
    .content h1 {
        width: 100vw;
        font-size: 4vh;
        font-weight: 350;
        display: flex;
        justify-content: center;
        align-items: center;
        padding-left: -2%;
    }
    .content h2 {
        width: 100vw;
        font-size: 3vh;
        font-weight: 800;
        margin: 0% 3% 0% 3%;
        display: flex;
        justify-content: center;
    }
    .imgWh2 img {
        display: none;
    }

    .imgWh1 img {
        margin-left: 15%;
        display: block;
        width: 40%;
        height: auto;
    }

    .imgWh1 h1 {
        margin-top: 30px;
    }
}
```

### Penjelasan :

1. `html`: Ini adalah selector untuk elemen HTML, yang berarti aturan gaya ini akan diterapkan pada seluruh elemen HTML dalam dokumen.
  - a. `height: 100vh`;: Ini mengatur tinggi (height) elemen HTML menjadi 100 viewport height (vh). Ini akan membuat elemen HTML mencapai tinggi sebesar viewport, sehingga elemen HTML akan mengisi seluruh tinggi layar pengguna.
2. `body`: Ini adalah selector untuk elemen body dalam dokumen HTML.
  - a. `font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif`;: Ini mengatur jenis font yang akan digunakan untuk teks dalam elemen body. Browser akan mencoba menggunakan font 'Segoe UI', jika tidak tersedia, akan mencoba font alternatif yang lain, seperti Tahoma, Geneva, Verdana, dan font serif pada umumnya.
  - b. `margin: 0`;: Ini menghilangkan margin pada elemen body. Margin adalah ruang di sekitar elemen, dan mengatur nilainya menjadi 0 akan menghapus ruang di sekitar body.
  - c. `padding: 0`;: Ini menghilangkan padding pada elemen body. Padding adalah ruang di dalam elemen, dan mengatur nilainya menjadi 0 akan menghapus ruang di dalam body.
3. `.container`: Ini adalah selector untuk elemen dengan kelas "container" dalam dokumen HTML.
  - a. `display: flex`;: Ini mengatur tata letak (layout) elemen menjadi tata letak flexbox. Dengan menggunakan flexbox, elemen "container" akan memiliki kemampuan untuk mengatur posisi dan ukuran anak-anaknya dengan lebih fleksibel.
  - b. `min-height: 100vh`;: Ini mengatur tinggi minimum elemen "container" menjadi 100 viewport height (vh). Ini memastikan bahwa "container" memiliki setidaknya tinggi yang sama dengan tinggi layar pengguna.
  - c. `max-height: fit-content`;: Ini mengatur tinggi maksimum elemen "container" agar sesuai dengan kontennya. Dengan demikian, jika konten dalam "container" melebihi tinggi layar pengguna, "container" akan tumbuh sesuai dengan kontennya.
  - d. `background-image: linear-gradient(to bottom, #924dbf, #0f0529)`;: Ini menetapkan gambar latar belakang berupa gradien linear dari atas ke bawah dengan warna mulai dari ungu (#924dbf) ke hitam (#0f0529). Gradien ini akan menjadi latar belakang "container".

### Sidebar :

1. `.sidebar`: Ini adalah selector untuk elemen dengan kelas "sidebar".
  - a. `width: 13%`;: Mengatur lebar elemen "sidebar" menjadi 13% dari lebar elemen induknya.



- b. `height: 100%;`: Mengatur tinggi elemen "sidebar" menjadi 100% dari tinggi elemen induknya.
  - c. `display: flex;`: Mengatur tata letak elemen "sidebar" menggunakan model flexbox.
  - d. `flex-direction: column;`: Mengatur arah tata letak menjadi kolom, sehingga elemen anak akan ditata secara vertikal.
  - e. `justify-content: flex-start;`: Mengatur konten di dalam "sidebar" untuk diletakkan di bagian atas.
  - f. `align-items: center;`: Mengatur konten di dalam "sidebar" untuk diberi posisi tengah secara horizontal.
  - g. `padding-top: 20px;`: Memberikan padding di bagian atas elemen "sidebar" sebesar 20 piksel.
2. `.sidebar nav ul`: Ini adalah selector untuk elemen `ul` (unordered list) yang berada di dalam elemen `nav` yang berada di dalam elemen dengan kelas "sidebar".
- a. `margin-left: -5%;`: Mengatur margin kiri negatif sebesar 5%, menggeser elemen `ul` ke kiri.
  - b. `padding: 0;`: Menghilangkan padding dari elemen `ul`.
  - c. `margin-top: 150%;`: Mengatur margin atas sebesar 150% dari tinggi elemen `ul`, menggeser `ul` ke bawah.
3. `.sidebar nav ul li`: Ini adalah selector untuk elemen-elemen `li` (list item) yang berada di dalam elemen `ul` yang berada di dalam elemen `nav` yang berada di dalam elemen dengan kelas "sidebar".
- a. `align-items: center;`: Mengatur konten di dalam setiap item daftar untuk diberi posisi tengah secara vertikal.
  - b. `width: 55%;`: Mengatur lebar setiap item daftar menjadi 55% dari lebar elemen induknya.
  - c. `list-style: none;`: Menghilangkan tanda penomoran dari daftar.
  - d. `text-decoration: none;`: Menghilangkan dekorasi (seperti garis bawah) dari tautan.
  - e. `padding: 0;`: Menghilangkan padding dari setiap item daftar.
  - f. `margin: 7vh 5vh;`: Memberikan margin sebesar 7 viewport height (vh) untuk bagian atas dan bawah, dan 5 viewport height (vh) untuk bagian kiri dan kanan.
  - g. `border-radius: 5px;`: Memberikan sudut bulat sebesar 5 piksel pada setiap item daftar.
4. `.sidebar nav ul li:hover`: Ini adalah selector untuk setiap item daftar saat dihover.
- a. `background-color: #924dbf;`: Mengubah warna latar belakang saat item daftar dihover menjadi ungu tua (#924dbf).

- b. `animation-delay: 0.3ms`:: Mengatur penundaan animasi selama 0,3 milidetik.
- 5. `.sidebar nav ul li a`: Ini adalah selector untuk setiap tautan di dalam item daftar.
  - a. `font-weight: 650`:: Mengatur ketebalan font menjadi 650.
  - b. `display: flex`:: Mengatur tata letak tautan menjadi flexbox.
  - c. `justify-content: center`:: Mengatur tautan untuk diletakkan di tengah secara horizontal.
  - d. `text-decoration: none`:: Menghilangkan dekorasi (seperti garis bawah) dari tautan.
  - e. `color: aliceblue`:: Mengatur warna teks menjadi biru muda (aliceblue).
  - f. `font-size: 3vh`:: Mengatur ukuran font tautan menjadi 3 viewport height (vh).
  - g. `margin: 10%`:: Memberikan margin sebesar 10% dari lebar dan tinggi tautan.

### Hero :

1. `.containerHero`: Ini adalah selector untuk elemen dengan kelas "containerHero".
  - a. `min-height: 100vh`:: Mengatur tinggi minimum elemen "containerHero" menjadi 100 viewport height (vh), sehingga memastikan elemen ini memiliki setidaknya tinggi sebesar tinggi layar pengguna.
  - b. `max-height: fit-content`:: Mengatur tinggi maksimum elemen "containerHero" agar sesuai dengan kontennya. Ini akan membuat elemen ini hanya memiliki tinggi yang cukup untuk menampung konten di dalamnya.
  - c. `display: flex`:: Mengatur tata letak elemen "containerHero" menggunakan model flexbox.
  - d. `flex-direction: column`:: Mengatur arah tata letak menjadi kolom, sehingga elemen anak akan ditata secara vertikal.
  - e. `justify-content: flex-start`:: Mengatur konten di dalam "containerHero" untuk diletakkan di bagian atas.
  - f. `align-items: center`:: Mengatur konten di dalam "containerHero" untuk diberi posisi tengah secara horizontal.
  - g. `padding-top: 20px`:: Memberikan padding di bagian atas elemen "containerHero" sebesar 20 piksel.
  - h. `background-image: linear-gradient(to bottom, #e5e5e5, #9a9a9a)`:: Menetapkan gradien linear sebagai latar belakang "containerHero", dimulai dari warna abu-abu (#e5e5e5) di bagian atas hingga warna abu-abu tua (#9a9a9a) di bagian bawah.

2. `.content`: Ini adalah selector untuk elemen dengan kelas "content".
  - a. `width: 100vw`:: Mengatur lebar elemen "content" menjadi 100 viewport width (vw), sehingga elemen ini akan memiliki lebar yang sama dengan lebar layar pengguna.
3. `.hero-content`: Ini adalah selector untuk elemen dengan kelas "hero-content".
  - a. `padding: 20px 0`:: Memberikan padding sebesar 20 piksel di bagian atas dan bawah elemen "hero-content".
4. `.content h1`: Ini adalah selector untuk heading level 1 (h1) di dalam elemen dengan kelas "content".
  - a. `font-size: 5vh`:: Mengatur ukuran font menjadi 5 viewport height (vh).
  - b. `font-weight: 350`:: Mengatur ketebalan font menjadi 350.
  - c. `margin: 0% 0% 0% 0%`:: Menghilangkan margin di semua sisi.
  - d. `display: flex`:: Mengatur tata letak menjadi flexbox.
  - e. `justify-content: flex-start`:: Mengatur konten untuk diletakkan di bagian kiri secara horizontal.
  - f. `align-items: center`:: Mengatur konten untuk diberi posisi tengah secara vertikal.
  - g. `padding-left: 2%`:: Memberikan padding sebesar 2% di bagian kiri.
5. `.content h2`: Ini adalah selector untuk heading level 2 (h2) di dalam elemen dengan kelas "content".
  - a. `font-size: 5vh`:: Mengatur ukuran font menjadi 5 viewport height (vh).
  - b. `font-weight: 800`:: Mengatur ketebalan font menjadi 800.
  - c. `margin: 0% 3% 0% 3%`:: Mengatur margin di atas dan bawah menjadi 0%, dan di kiri dan kanan menjadi 3%.
6. `.content p`: Ini adalah selector untuk elemen paragraf (p) di dalam elemen dengan kelas "content".
  - a. `font-size: 2vh`:: Mengatur ukuran font menjadi 2 viewport height (vh).
  - b. `font-weight: 350`:: Mengatur ketebalan font menjadi 350.
  - c. `padding-left: 2%`:: Memberikan padding sebesar 2% di bagian kiri.
  - d. `margin-bottom: 50px`:: Memberikan margin sebesar 50 piksel di bagian bawah.
  - e. `max-width: 50vw`:: Mengatur lebar maksimum paragraf menjadi 50 viewport width (vw).
7. `.imgWh1`: Ini adalah selector untuk elemen dengan kelas "imgWh1".
  - a. `margin: 0; padding: 0`:: Menghilangkan margin dan padding dari elemen.
  - b. `display: flex`:: Mengatur tata letak menjadi flexbox.
  - c. `flex-direction: column`:: Mengatur arah tata letak menjadi kolom.
  - d. `justify-content: center`:: Mengatur konten untuk diletakkan di tengah secara vertikal.

- e. align-items: center;; Mengatur konten untuk diberi posisi tengah secara horizontal.
- 8. .imgWh1 h1: Ini adalah selector untuk heading level 1 (h1) di dalam elemen dengan kelas "imgWh1".
  - a. align-self: flex-start;; Mengatur posisi h1 ke atas (pada bagian kiri) dari konten induknya.
  - b. margin-top: 100px;; Memberikan margin atas sebesar 100 piksel.
- 9. .imgWh1 img: Ini adalah selector untuk elemen gambar (img) di dalam elemen dengan kelas "imgWh1".
  - a. display: none;; Mengatur elemen gambar untuk disembunyikan.
- 10..imgWh2: Ini adalah selector untuk elemen dengan kelas "imgWh2".
  - a. display: flex;; Mengatur tata letak menjadi flexbox.
  - b. justify-content: flex-start;; Mengatur konten untuk diletakkan di bagian kiri secara horizontal.
  - c. align-items: center;; Mengatur konten untuk diberi posisi tengah secara vertikal.
- 11..imgWh2 img: Ini adalah selector untuk elemen gambar (img) di dalam elemen dengan kelas "imgWh2".
  - a. min-width: 100px;; Menetapkan lebar minimum gambar menjadi 100 piksel.
    - i. width: 250px;; Menetapkan lebar gambar menjadi 250 piksel.
  - b. height: auto;; Menjaga rasio aspek gambar tetap dengan mengatur tinggi ke otomatis.
  - c. padding: 0px 50px 0px 50px;; Memberikan padding di kiri dan kanan gambar sebesar 50 piksel.
- 12..imgWh2 h2: Ini adalah selector untuk heading level 2 (h2) di dalam elemen dengan kelas "imgWh2".
  - a. padding-left: 2%;; Memberikan padding di kiri sebesar 2%.
- 13..action-btn: Ini adalah selector untuk tautan dengan kelas "action-btn".
  - a. border-radius: 28px;; Memberikan sudut bulat pada tautan sebesar 28 piksel.
  - b. color: #ffffff;; Mengatur warna teks menjadi putih.
  - c. font-size: 15px;; Mengatur ukuran font menjadi 15 piksel.
  - d. background: #494848;; Memberikan latar belakang dengan warna abu-abu (#494848).
  - e. padding: 10px 30px 10px 30px;; Memberikan padding sebesar 10 piksel di atas dan bawah, serta 30 piksel di kiri dan kanan.
  - f. margin-left: 2%;; Memberikan margin kiri sebesar 2%.

**Footer :**

- 1. .footer: Ini adalah selector untuk elemen dengan kelas "footer".

- a. `width: 100%;`: Mengatur lebar elemen footer menjadi 100% dari lebar elemen induknya.
  - b. `height: 10%;`: Mengatur tinggi elemen footer menjadi 10% dari tinggi elemen induknya.
  - c. `padding: 20px 0;`: Memberikan padding sebesar 20 piksel di bagian atas dan bawah elemen footer.
  - d. `margin-top: auto;`: Mengatur margin atas elemen footer menjadi otomatis, sehingga elemen footer akan diletakkan di bagian bawah halaman.
  - e. `background-color: #494848;`: Mengatur warna latar belakang elemen footer menjadi abu-abu tua (#494848).
  - f. `color: white;`: Mengatur warna teks pada footer menjadi putih.
  - g. `display: flex;`: Mengatur tata letak elemen footer menggunakan model flexbox.
  - h. `justify-content: center;`: Mengatur konten di dalam footer untuk diletakkan di tengah secara horizontal.
  - i. `text-align: center;`: Mengatur teks di dalam footer untuk diletakkan di tengah secara horizontal.
  - j. `position: relative;`: Mengatur posisi elemen footer menjadi relatif terhadap posisi normalnya.
  - k. `bottom: 0;`: Mengatur elemen footer untuk diletakkan di bagian bawah halaman.
2. `.footer ul`: Ini adalah selector untuk elemen `ul` (unordered list) di dalam elemen dengan kelas "footer".
  - a. `margin: 0; padding: 0;`: Menghilangkan margin dan padding dari elemen `ul`.
  - b. `display: flex;`: Mengatur tata letak elemen `ul` menggunakan model flexbox.
  - c. `list-style: none;`: Menghilangkan tanda penomoran dari daftar.
3. `.footer li`: Ini adalah selector untuk elemen-elemen `li` (list item) di dalam elemen `ul` yang berada di dalam elemen dengan kelas "footer".
  - a. `display: flex;`: Mengatur tata letak elemen-elemen `li` menggunakan model flexbox.
  - b. `flex-direction: column;`: Mengatur arah tata letak menjadi kolom.
  - c. `justify-content: center;`: Mengatur konten di dalam setiap item daftar untuk diletakkan di tengah secara vertikal.
  - d. `align-items: center;`: Mengatur konten di dalam setiap item daftar untuk diberi posisi tengah secara horizontal.
  - e. `margin-right: 10px;`: Memberikan margin sebesar 10 piksel di sebelah kanan setiap item daftar.

- f. `width: 100px;`: Mengatur lebar setiap item daftar menjadi 100 piksel.
- 4. `.footer img`: Ini adalah selector untuk elemen gambar (`img`) di dalam elemen-elemen `li` yang berada di dalam elemen `ul` yang berada di dalam elemen dengan kelas `"footer"`.
  - a. `width: 30px;`: Mengatur lebar gambar menjadi 30 piksel.
  - b. `height: auto;`: Menjaga rasio aspek gambar tetap dengan mengatur tinggi ke otomatis.
  - c. `border-radius: 8px;`: Memberikan sudut bulat pada gambar sebesar 8 piksel.
- 5. `.footer p`: Ini adalah selector untuk elemen paragraf (`p`) di dalam elemen dengan kelas `"footer"`.
  - a. `font-size: 2vh;`: Mengatur ukuran font menjadi 2 viewport height (`vh`).

### Responsive :

1. `@media screen and (max-width: 768px) { ... }`: Ini adalah aturan media query yang akan diterapkan hanya ketika lebar layar perangkat maksimum adalah 768 piksel atau kurang.
2. `.container`: Di dalam aturan ini, properti dan nilai-nilai CSS untuk elemen dengan kelas `"container"` diubah.
  - a. `display: block;`: Mengubah tata letak elemen menjadi blok, yang berarti elemen tersebut akan menempati seluruh lebar yang tersedia.
  - b. `background-image: linear-gradient(to right, #fbf5a3d5, #924dbf);`: Mengubah gambar latar belakang menjadi gradien linear dari kanan ke kiri, mulai dari warna kuning muda (`#fbf5a3d5`) hingga ungu tua (`#924dbf`).
3. `.sidebar`: Di sini, tata letak dan penataan elemen dengan kelas `"sidebar"` diubah.
  - a. `width: 100vw;`: Mengatur lebar elemen menjadi 100 viewport width (`vw`), sehingga elemen akan mengisi seluruh lebar layar.
  - b. `flex-direction: row;`: Mengatur arah tata letak menjadi baris, sehingga elemen anak akan ditata secara horizontal.
  - c. `padding-top: 0;`: Menghilangkan padding atas.
4. `.sidebar nav ul`: Tata letak daftar navigasi di dalam sidebar diubah.
  - a. `justify-content: center;`: Mengatur konten di dalam daftar untuk diletakkan di tengah secara horizontal.
  - b. `align-items: center;`: Mengatur konten di dalam daftar untuk diberi posisi tengah secara vertikal.
5. `.sidebar nav ul li`: Penataan elemen-elemen daftar navigasi diubah.
  - a. `margin: 3vh 3vh;`: Memberikan margin sebesar 3 viewport height di atas dan bawah, dan 3 viewport height di kiri dan kanan.
  - b. `background-color: transparent;`: Mengubah warna latar belakang

menjadi transparan.

6. `.sidebar nav ul li a`: Ini adalah tautan di dalam daftar navigasi.
  - a. `padding: 10%;`: Memberikan padding sebesar 10% di dalam tautan.
7. `.sidebar nav ul li a: hover`: Ini adalah gaya yang akan diterapkan ketika tautan dihover oleh pengguna.
  - a. `background-image: linear-gradient(to left, #f4efc4d5, #e5c7f9);`: Mengubah gambar latar belakang menjadi gradien linear dari kiri ke kanan, mulai dari warna kuning muda (`#f4efc4d5`) hingga ungu muda (`#e5c7f9`).
  - b. `color: #924dbf;`: Mengubah warna teks menjadi ungu tua.
  - c. `border-radius: 5px;`: Memberikan sudut bulat pada tautan sebesar 5 piksel.
8. `.containerHero`: Aturan ini menyesuaikan tata letak dan penampilan konten hero di dalam kontainer hero.
  - a. `padding: 0;`: Menghilangkan padding dari kontainer hero.
9. `.hero-content`: Mengatur tata letak dan penampilan konten hero di dalamnya.
  - a. `display: flex;`: Mengatur tata letak menjadi flexbox.
  - b. `flex-direction: column;`: Mengatur arah tata letak menjadi kolom.
  - c. `justify-content: center;`: Mengatur konten untuk diletakkan di tengah secara vertikal.
  - d. `align-items: center;`: Mengatur konten untuk diberi posisi tengah secara horizontal.
  - e. `margin-left: -2%;`: Memberikan margin kiri sebesar -2% untuk menyesuaikan posisi konten.
10. `.content h1`: Ini adalah selector untuk heading level 1 (`h1`) di dalam konten.
  - a. `width: 100vw;`: Mengatur lebar heading `h1` menjadi 100 viewport width (`vw`).
  - b. `font-size: 4vh;`: Mengatur ukuran font menjadi 4 viewport height (`vh`).
  - c. `padding-left: -2%;`: Memberikan padding kiri sebesar -2%.
11. `.content h2`: Ini adalah selector untuk heading level 2 (`h2`) di dalam konten.
  - a. `width: 100vw;`: Mengatur lebar heading `h2` menjadi 100 viewport width (`vw`).
  - b. `font-size: 3vh;`: Mengatur ukuran font menjadi 3 viewport height (`vh`).
12. `.imgWh2 img`: Selector ini mengatur gambar di dalam elemen dengan kelas `"imgWh2"`.
  - a. `display: none;`: Mengatur gambar untuk disembunyikan.
13. `.imgWh1 img`: Ini adalah selector untuk gambar di dalam elemen dengan kelas `"imgWh1"`.
  - a. `margin-left: 15%;`: Memberikan margin kiri sebesar 15%.
  - b. `display: block;`: Mengatur elemen gambar menjadi tampil sebagai

blok.

- c. `width: 40%;`: Mengatur lebar gambar menjadi 40% dari lebar elemen induknya.
- d. `height: auto;`: Menjaga rasio aspek gambar tetap dengan mengatur tinggi ke otomatis.

14..`imgWh1 h1`: Ini adalah selector untuk heading level 1 (`h1`) di dalam elemen dengan kelas "`imgWh1`".

- a. `margin-top: 30px;`: Memberikan margin atas sebesar 30 piksel.

Ouput :





Tablet :

