

LAPORAN PRAKTIKUM

PEMROGRAMAN WEB D



SOFTWARE ENGINEERING
LABORATORY

NAMA : R Sy Calvin Putra Bhayangkara

NIM : 202431121

KELAS : D

DOSEN : Muhammad Fadli Prathama, S.SI., M.MSI

NO.PC : 27

ASISTEN : 1. Sakura Amastasya Salsabila S

2. Fhazel Kesra Arivi

3. Muhammad Farhan Fahrezy

4. Kashrina Masyid Azka

INSTITUT TEKNOLOGI PLN

TEKNIK INFORMATIKA

2025

I. Website Informasi Mengenai Pencemaran Lingkungan

Website Informasi Mengenai Pencemaran Lingkungan adalah sebuah media berbasis web yang dirancang untuk menyampaikan berbagai jenis informasi mengenai permasalahan pencemaran yang terjadi di lingkungan sekitar. Website ini berfungsi sebagai sarana edukasi bagi masyarakat agar dapat memahami jenis-jenis pencemaran, penyebab terjadinya pencemaran, dampak yang ditimbulkan, serta upaya yang dapat dilakukan untuk mengurangi atau mencegah pencemaran tersebut.

Website ini dikembangkan menggunakan teknologi dasar web, yaitu HTML sebagai struktur halaman, CSS sebagai pengaturan tampilan visual, dan JavaScript untuk menambahkan interaktivitas. Melalui halaman utama, pengguna dapat melihat penjelasan umum mengenai pencemaran lingkungan serta daftar berbagai jenis pencemaran seperti pencemaran udara, air, tanah, suara, cahaya, sampah plastik, dan radioaktif. Setiap topik dapat dibuka dan ditutup secara dinamis sehingga memudahkan pengguna dalam membaca informasi secara lebih nyaman. Berikut dibawah adalah penjelasan dari setiap langkah pembuatannya:

A. Halaman Pertama

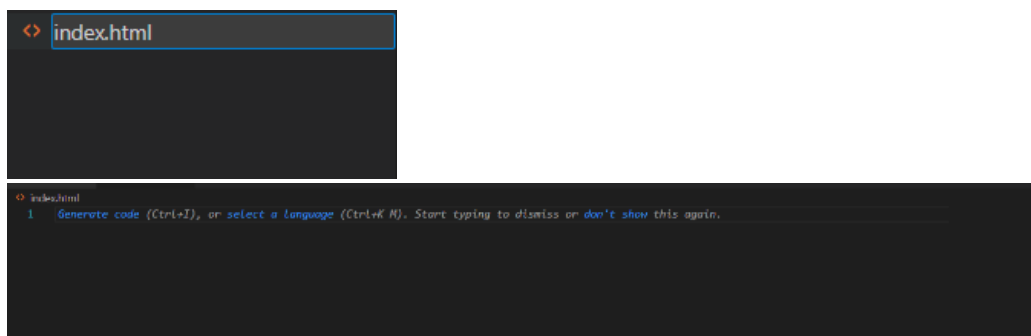
Halaman pertama membuat deskripsi penjelasan setiap 7 informasi mengenai pencemaran lingkungan. Dengan menggunakan interaksi pada setiap judul. Ketika judul diklik, muncul pertanyaan kepada pengguna: “Apakah Anda ingin menyembunyikan penjelasan ini?”. Jika pengguna menyetujuinya, maka paragraf penjelasan untuk judul tersebut disembunyikan. Kemudian buat interaksi untuk kembali menampilkan penjelasannya dengan pertanyaan “Apakah Anda ingin menampilkan penjelasan ini?”. Dan membuat sebuah tombol di bawah penjelasan untuk mengubah mode tampilan halaman. Tersedia dua mode tampilan:

- Mode tampilan rapi dengan perpaduan warna dan kontras yang baik.
- Mode tampilan berantakan, dengan perpaduan warna dan kontras yang buruk.

Saat tombol ini ditekan, halaman harus berpindah antara mode rapi dan mode berantakan.

Dan juga menggunakan tema berwarna coklat yang sesuai laki laki dengan NIM ganjil, serta tidak boleh memakai warna lain, seperti hitam untuk textnya, intinya dominan warna sesuai tema. Kemudian background halaman dilarang berwarna PUTIH/HITAM. Jika tema warna tidak sesuai dengan ketentuan tersebut. Lalu di bawah tombol pengubah mode tampilan, tambahkan tombol “Tampilkan Gambar”. Jika tombol ini diklik, maka pengguna diarahkan ke halaman baru yang berisi gambar atau foto dari 7 contoh pencemaran lingkungan yang sama seperti yang telah dijelaskan di halaman sebelumnya. Berikut penjelasan halaman yang saya buat:

1. Membuat File HTML



Langkah ini membuat file html yang bernama index.html yang dimana untuk menyimpan source kode yang kita buat dan selanjutnya muncul file kosong dimana untuk membuat source kode pada halaman pertama yang di buat.

2. Menuliskan Struktur Dasar HTML

```
1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6   <title>Informasi Pencemaran Lingkungan</title>
7   <link rel="stylesheet" href="style.css" />
8   <script defer src="script.js"></script>
9 </head>
10 <body class="mode-rapi">
```

Langkah ini menjelaskan struktur dasar halaman web yang menggunakan HTML5 untuk menampilkan informasi mengenai pencemaran lingkungan. Pada bagian pertama terdapat deklarasi `<!DOCTYPE html>` yang menunjukkan bahwa dokumen ditulis dengan standar HTML modern. Tag `<html lang="id">` digunakan untuk menandakan bahwa bahasa utama halaman adalah bahasa Indonesia. Di dalam bagian `<head>`, terdapat pengaturan penting seperti `<meta charset="UTF-8">` agar karakter dapat terbaca dengan benar serta `<meta name="viewport" content="width=device-width, initial-scale=1.0">` agar tampilan web bersifat responsif di berbagai perangkat. Halaman ini juga diberi judul melalui tag `<title>`. Selanjutnya langkah ini juga memuat file CSS eksternal `style.css` untuk mengatur tampilan visual dan file JavaScript `script.js` yang dipanggil menggunakan atribut `defer`, sehingga skrip akan berjalan setelah struktur HTML selesai dimuat. Pada bagian `<body>` diberikan kelas `mode-rapi`, yang kemungkinan digunakan untuk menentukan mode tampilan rapi pada halaman. Secara keseluruhan, langkah ini membentuk pondasi utama halaman web sebelum konten tambahan dimasukkan.

3. Membuat Bagian Header

```
11 <header class="site-header">
12   <div class="identity">
13     <h1 id="Nama">R Sy Calvin Putra Bhayangkara</h1>
14     <p id="NIM">202431121</p>
15   </div>
16   <div class="mode-indicator" id="modeIndicator">Mode: Rapi</div>
17 </header>
```

Langkah ini membuat bagian header pada halaman web yang berfungsi untuk menampilkan identitas pengguna serta indikator mode tampilan. Elemen `<header class="site-header">` digunakan sebagai wadah utama untuk informasi pada bagian atas halaman. Di dalamnya terdapat `<div class="identity">` yang berisi elemen `<h1 id="Nama">` untuk menampilkan nama lengkap “R Sy Calvin Putra Bhayangkara” dan elemen `<p id="NIM">` yang menampilkan nomor induk mahasiswa “202431121.” Bagian identitas ini memberikan informasi penting tentang pembuat atau pemilik halaman. Langkah ini juga menambahkan elemen `<div class="mode-indicator" id="modeIndicator">Mode: Rapi</div>` yang berfungsi menampilkan status mode tampilan halaman, yaitu “Mode: Rapi.” Elemen ini biasanya digunakan bersama JavaScript untuk mengubah tampilan antara mode rapi dan mode acak. Secara keseluruhan, langkah ini menyusun struktur header agar halaman terlihat lebih informatif dan terorganisir.

4. Membuat Kontainer Isi Halaman

```
20 <main class="container">
```

Langkah ini menambahkan elemen `<main class="container">` yang berfungsi sebagai area utama dari isi halaman web. Tag `<main>` digunakan untuk menandai bagian konten paling penting, yaitu bagian yang berisi informasi utama yang ingin disampaikan kepada pengunjung. Dengan memberikan kelas `container`, elemen ini dapat diatur melalui CSS agar memiliki tata letak yang rapi, terpusat, dan responsif. Bagian `<main>` ini akan menjadi tempat diletakkannya seluruh elemen konten seperti kartu informasi, teks, gambar, atau fitur lain yang menjadi inti dari website.

5. Membuat Bagian Pengertian Umum

```

21 <section class="intro">
22   <h2>Pengertian Umum Pencemaran Lingkungan</h2>
23   <p>
24     Pencemaran lingkungan adalah masuknya atau dimasukkannya makhluk hidup, zat, energi, atau komponen lain ke lingkungan
25     sehingga kualitasnya menurun sampai ke tingkat yang mengganggu kehidupan manusia dan makhluk hidup lain. Pencemaran
26     dapat terjadi pada air, udara, tanah, dan lingkungan laut; penyebabnya meliputi aktivitas industri, sampah domestik,
27     limbah pertanian, dan lain-lain.
28   </p>
29 </section>

```

Langkah ini membuat sebuah bagian pengantar menggunakan elemen `<section class="intro">` yang berfungsi untuk menyajikan informasi awal mengenai topik pencemaran lingkungan. Di dalamnya terdapat elemen `<h2>` yang menampilkan judul “Pengertian Umum Pencemaran Lingkungan” sebagai heading utama untuk bagian ini. Setelah itu, terdapat elemen `<p>` yang berisi penjelasan lengkap tentang apa yang dimaksud dengan pencemaran lingkungan, yaitu masuknya makhluk hidup, zat, atau energi ke dalam lingkungan hingga menurunkan kualitasnya dan mengganggu kehidupan manusia maupun makhluk hidup lainnya. Paragraf tersebut juga menjelaskan bahwa pencemaran dapat terjadi pada berbagai komponen seperti air, udara, tanah, dan laut, serta menyebutkan beberapa penyebab seperti aktivitas industri, sampah rumah tangga, dan limbah pertanian. Bagian ini berfungsi sebagai pembuka yang memperkenalkan konsep dasar sebelum pengguna membaca informasi lebih lanjut pada bagian berikutnya.

6. Membuat Daftar Masalah Pencemaran

```

31 <section class="problems">
32   <h2>Masalah-Masalah Pencemaran Lingkungan</h2>
33   <article class="problem" data-id="1">
34     <h3 class="title">1. Pencemaran Udara</h3>
35     <p class="desc">Pencemaran udara disebabkan oleh emisi kendaraan bermotor, pembakaran bahan bakar fosil,
36     aktivitas industri, dan pembakaran sampah. Polutan seperti partikulat, SO2, NOx, dan CO mengganggu kesehatan
37     pernapasan serta menyebabkan hujan asam dan perubahan iklim.</p>
38   </article>
39   <article class="problem" data-id="2">
40     <h3 class="title">2. Pencemaran Air</h3>
41     <p class="desc">Limbah industri, limbah rumah tangga, dan limbah pertanian (pupuk serta pestisida) mencemari sungai,
42     danau, dan laut. Hal ini menurunkan kualitas air, mematikan organisme akuatik, dan mengancam pasokan air bersih.</p>
43   </article>
44   <article class="problem" data-id="3">
45     <h3 class="title">3. Pencemaran Tanah</h3>
46     <p class="desc">Pembuangan limbah padat langsung ke tanah, penggunaan pestisida berlebih, serta tumpahan bahan
47     kimia dapat merusak struktur dan kesuburan tanah, mengkontaminasi tanaman, dan masuk ke rantai makanan.</p>
48   </article>
49   <article class="problem" data-id="4">
50     <h3 class="title">4. Pencemaran Suara</h3>
51     <p class="desc">Kebisingan dari lalu lintas, kegiatan industri, konstruksi, dan tempat hiburan dapat menyebabkan
52     gangguan tidur, stres, dan gangguan pendengaran pada manusia serta mengganggu hewan lokal.</p>
53   </article>
54   <article class="problem" data-id="5">
55     <h3 class="title">5. Pencemaran Cahaya</h3>
56     <p class="desc">Pencahayaian berlebih di daerah perkotaan (light pollution) mengganggu ritme biologis manusia dan hewan,
57     menyamarkan bintang di langit malam, serta memengaruhi ekosistem yang sensitif terhadap kegelapan.</p>
58   </article>
59   <article class="problem" data-id="6">
60     <h3 class="title">6. Pencemaran Plastik / Sampah</h3>
61     <p class="desc">Sampah plastik yang tidak terkelola menjadi ancaman besar bagi lingkungan darat dan laut. Plastik
62     sulit terurai, merusak habitat, dan sering masuk ke rantai makanan melalui organisme kecil yang memakannya.</p>
63   </article>
64   <article class="problem" data-id="7">
65     <h3 class="title">7. Pencemaran Radioaktif</h3>
66     <p class="desc">Kebocoran bahan radioaktif dari fasilitas nuklir, limbah radioaktif yang tidak ditangani, atau
67     kecelakaan saat pengangkutan dapat menimbulkan bahaya jangka panjang bagi kesehatan manusia dan lingkungan.</p>
68   </article>
69 </section>

```

Langkah ini membuat sebuah bagian khusus yang menampilkan berbagai jenis masalah pencemaran lingkungan menggunakan elemen `<section class="problems">`. Di bagian atas terdapat tag `<h2>` yang memberikan judul “Masalah-Masalah Pencemaran Lingkungan,” sehingga pembaca mengetahui bahwa isi selanjutnya adalah daftar permasalahan yang terjadi di lingkungan. Di dalam section ini terdapat beberapa elemen `<article>` yang masing-masing mewakili satu jenis pencemaran. Setiap artikel diberi kelas `problem` dan atribut `data-id` untuk memudahkan pengolahan melalui JavaScript nantinya. Pada setiap artikel terdapat elemen `<h3 class="title">` yang menampilkan nama jenis pencemaran, diikuti oleh elemen `<p class="desc">` yang menjelaskan penyebab, dampak, dan karakteristik dari pencemaran tersebut. Langkah ini juga memuat tujuh jenis pencemaran, dimulai dari pencemaran udara, air, tanah, suara, cahaya, sampah plastik, hingga pencemaran radioaktif. Penjelasan pada masing-masing paragraf menggambarkan bagaimana setiap jenis pencemaran terjadi, apa sumbernya, dan apa bahaya yang ditimbulkan bagi manusia, hewan, maupun lingkungan secara keseluruhan. Struktur artikel yang terpisah-pisah ini mempermudah pengguna dalam memahami tiap permasalahan sekaligus memungkinkan fitur interaktif seperti show/hide deskripsi atau mode tampilan rapi dan acak bekerja dengan lebih optimal. Dengan demikian, langkah ini menata informasi secara sistematis agar mudah dibaca dan dipelajari oleh pengunjung halaman.

7. Membuat Area Tombol

```
78 <div class="btn-area">
79   <button class="mode-toggle-btn" id="toggleModeBtn">Ubah Mode Tampilan</button>
80 </div>
81
82 <div class="btn-area">
83   <a href="halaman_2.html" class="btn-gambar">Tampilkan Gambar</a>
84 </div>
```

Langkah ini menambahkan dua area tombol yang berfungsi untuk memberikan interaksi kepada pengguna. Elemen pertama adalah `<div class="btn-area">` yang di dalamnya terdapat sebuah tombol `<button class="mode-toggle-btn" id="toggleModeBtn">Ubah Mode Tampilan</button>`. Tombol ini berfungsi untuk mengubah tampilan halaman antara mode rapi dan mode acak. Biasanya tombol ini dihubungkan dengan JavaScript sehingga ketika diklik, kelas pada elemen `<body>` dapat berubah dan mengubah tata letak atau urutan tampilan informasi sesuai mode yang dipilih, ini juga membuat sebuah tombol kedua dalam `<div class="btn-area">`, tetapi berupa link `Tampilkan Gambar`. Tombol ini berfungsi mengarahkan pengguna ke halaman kedua yang berisi gambar-gambar terkait pencemaran lingkungan. Dengan menempatkan tombol dalam div terpisah, penataan tombol menjadi lebih mudah dilakukan melalui CSS agar tampil rapi dan teratur. Secara keseluruhan, bagian ini memberikan navigasi dan kontrol yang penting bagi pengunjung halaman.

8. Membuat Footer

```
87 <footer class="site-footer">
88   <p>Informasi Pencemaran Lingkungan - R Sy Calvin Putra Bhayangkara</p>
89 </footer>
```

Langkah ini menambahkan bagian footer pada halaman web menggunakan elemen `<footer class="site-footer">`. Footer ini berfungsi sebagai penutup halaman sekaligus memberikan informasi tambahan yang biasanya berada di bagian paling bawah. Di dalam footer terdapat elemen `<p>` yang berisi teks “Informasi Pencemaran Lingkungan - R Sy Calvin Putra Bhayangkara,” yang menunjukkan identitas pembuat atau pemilik situs. Penempatan informasi

ini pada footer membuat halaman terlihat lebih profesional dan terstruktur karena pengguna dapat melihat dengan jelas siapa yang membuat konten tersebut.

9. Penutup Tag HTML

```
90 </body>
91 </html>
```

Langkah ini menutup seluruh struktur dokumen HTML dengan benar menggunakan tag `</body>` dan `</html>`. Tag `</body>` menandai bagian akhir dari seluruh konten yang tampil di halaman, termasuk teks, tombol, gambar, dan elemen-elemen lain yang sudah dibuat sebelumnya. Setelah itu, tag `</html>` digunakan untuk menutup dokumen HTML secara keseluruhan, memastikan bahwa seluruh bagian halaman telah dibungkus dengan struktur yang lengkap dan valid. Penutupan ini sangat penting agar browser dapat membaca dan memproses file HTML tanpa error.

B. Halaman Kedua

Pada halaman kedua membuat halaman gambar, tampilkan setiap gambar dengan ukuran yang pas, dan Halaman kedua harus responsif menggunakan aturan:

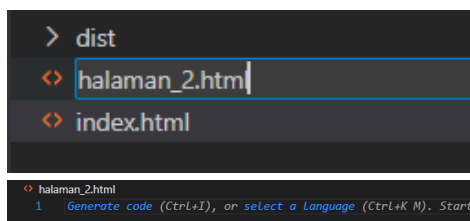
- Mobile ($\leq 600\text{px}$): card ditampilkan dalam 1 kolom.
- Tablet ($600\text{--}992\text{px}$): card ditampilkan dalam 2 kolom.
- Desktop ($\geq 992\text{px}$): card ditampilkan dalam 3 kolom.

Tema warna halaman kedua harus sama dengan tema warna halaman pertama dan mengikuti mode tampilan rapi, selanjutnya juga membuat tombol di bawahnya untuk Kembali ke halaman sebelumnya. Dan membuat sebuah tombol di bawah penjelasan untuk mengubah mode tampilan halaman. Tersedia dua mode tampilan:

- Mode tampilan rapi dengan perpaduan warna dan kontras yang baik.
- Mode tampilan berantakan, dengan perpaduan warna dan kontras yang buruk.

Saat tombol ini ditekan, halaman harus berpindah antara mode rapi dan mode berantakan. Berikut penjelasan dan tahapan membuat halaman gambar:

1. Membuat File Baru



Langkah ini membuat file html yang bernama halaman_2 yang dimana untuk menyimpan source kode yang kita buat dan selanjutnya muncul file kosong dimana untuk membuat source kode pada halaman gambar yang di buat.

2. Menambahkan Struktur Dasar HTML

```
1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4   <meta charset="utf-8" />
5   <meta name="viewport" content="width=device-width,initial-scale=1" />
6   <title>Gambar: Contoh Pencemaran Lingkungan</title>
7   <link rel="stylesheet" href="style.css" />
8   <script defer src="script.js"></script>
9 </head>
10 <body class="mode-rapi">
```

Langkah ini membuat struktur awal untuk halaman kedua yang menampilkan gambar-gambar pencemaran lingkungan. Bagian pertama diawali dengan deklarasi `<!DOCTYPE html>` yang menandakan bahwa dokumen menggunakan standar HTML5. Tag `<html lang="id">` menunjukkan bahwa bahasa utama halaman adalah bahasa Indonesia. Pada bagian `<head>`, kode ini menetapkan pengaturan penting seperti `<meta charset="utf-8">` agar karakter dapat ditampilkan dengan benar, serta `<meta name="viewport" content="width=device-width,initial-scale=1">` untuk memastikan tampilan halaman tetap responsif di perangkat mobile maupun desktop. Halaman ini diberi judul “Gambar: Contoh Pencemaran Lingkungan” melalui elemen `<title>`. Langkah ini juga memuat file CSS eksternal `style.css` untuk mengatur tampilan halaman, serta file JavaScript `script.js` yang dimuat menggunakan atribut `defer`, sehingga skrip hanya akan berjalan setelah struktur HTML selesai diproses. Pada bagian `<body>`, halaman diberi kelas `mode-rapi` yang berfungsi untuk menandai bahwa tampilan halaman sedang berada pada mode rapi. Kelas ini biasanya digunakan oleh JavaScript untuk mengontrol perubahan tampilan seperti mode acak dan mode rapi. Bagian ini merupakan kerangka awal dari halaman sebelum konten gambar ditambahkan.

3. Menambahkan Indikator Mode

```
<div class="mode-indicator" id="modeIndicatorGallery">Mode: Rapi</div>
```

Langkah ini menambahkan elemen `<div>` dengan kelas `mode-indicator` dan ID `modeIndicatorGallery` yang berfungsi sebagai indikator mode tampilan pada halaman galeri gambar. Teks di dalam elemen tersebut, yaitu “Mode: Rapi,” menunjukkan bahwa halaman sedang berada dalam mode tampilan rapi sama seperti halaman utama. Elemen ini biasanya terhubung dengan JavaScript untuk mengubah teks dan tampilan ketika pengguna menekan tombol pengubah mode, misalnya beralih ke mode acak. Dengan adanya indikator ini, pengguna dapat mengetahui dengan jelas mode tampilan apa yang sedang digunakan pada halaman galeri.

4. Membuat Header Halaman Gambar

```
14 <header class="images-header">
15   <div class="identity">
16     <h1>Contoh Pencemaran Lingkungan</h1>
17   </div>
18 </header>
```

Langkah ini membuat bagian header khusus untuk halaman yang menampilkan gambar-gambar pencemaran lingkungan. Header ini dibuat menggunakan elemen `<header class="images-header">`, yang berfungsi sebagai area paling atas halaman untuk memberikan judul atau informasi pengantar. Di dalam header terdapat `<div class="identity">` yang menjadi wadah untuk elemen judul. Elemen `<h1>` di dalamnya menampilkan teks “Contoh Pencemaran Lingkungan,” sehingga pengguna langsung mengetahui bahwa halaman ini berisi gambar-gambar yang berkaitan dengan topik pencemaran. Langkah ini membuat tampilan halaman menjadi lebih terstruktur dan profesional karena bagian header memberikan identitas jelas terhadap isi halaman. Selain itu, penggunaan kelas seperti `images-header` dan `identity` memungkinkan pengaturan gaya tampilan melalui CSS agar sesuai dengan tema halaman utama.

5. Membuat Galeri Gambar

```
<main class="gallery">
  <article class="card">
    
    <h3>Pencemaran Udara</h3>
  </article>

  <article class="card">
    
    <h3>Pencemaran Air</h3>
  </article>

  <article class="card">
    
    <h3>Pencemaran Tanah</h3>
  </article>

  <article class="card">
    
    <h3>Pencemaran Suara</h3>
  </article>

  <article class="card">
    
    <h3>Pencemaran Cahaya</h3>
  </article>

  <article class="card">
    
    <h3>Pencemaran Plastik</h3>
  </article>

  <article class="card">
    
    <h3>Pencemaran Radioaktif</h3>
  </article>
</main>
```

Langkah ini membuat bagian utama halaman galeri menggunakan elemen `<main class="gallery">` yang berfungsi sebagai tempat menampilkan kumpulan gambar mengenai berbagai jenis pencemaran lingkungan. Di dalamnya terdapat beberapa elemen `<article class="card">`, di mana setiap card berisi satu gambar dan judul yang menggambarkan jenis pencemaran tertentu. Setiap card terdiri dari elemen `` untuk menampilkan gambar yang diambil dari folder image, serta elemen `<h3>` yang memberikan nama jenis pencemaran. Struktur ini membuat tampilan galeri menjadi terorganisir dengan baik, karena setiap jenis pencemaran memiliki card sendiri-sendiri yang mudah dikenali oleh pengguna. Langkah ini menampilkan tujuh jenis pencemaran, yaitu pencemaran udara, air, tanah, suara, cahaya, plastik, dan radioaktif. Dengan menggunakan class gallery dan card, CSS dapat mengatur tampilan agar responsif sesuai instruksi sebelumnya—misalnya 1 kolom pada mobile, 2 kolom pada tablet, dan 3 kolom pada desktop. Selain itu, struktur card seperti ini juga mendukung fitur mode tampilan rapi dan acak ketika terhubung dengan JavaScript. Secara keseluruhan, langkah ini membentuk tampilan galeri gambar yang jelas, rapi, dan mudah dipahami.

6. Membuat Area Tombol

```
57 <div class="btn-area">
58 |   <button class="mode-toggle-btn" id="toggleModeBtn">Ubah Mode Tampilan</button>
59 | </div>
60
61 <div class="back-area">
62 |   <a href="index.html" class="btn-back">Kembali</a>
63 | </div>
```

Langkah ini menambahkan area tombol interaktif pada halaman galeri. Bagian pertama menggunakan elemen `<div class="btn-area">` yang berisi tombol `<button class="mode-toggle-btn" id="toggleModeBtn">Ubah Mode Tampilan</button>`. Tombol ini berfungsi untuk mengubah mode tampilan galeri antara mode rapi dan mode acak, sama seperti pada halaman utama. JavaScript biasanya menggunakan ID `toggleModeBtn` untuk mendeteksi klik dan kemudian mengubah tata letak kartu gambar sesuai mode yang dipilih pengguna. Langkah ini juga menyertakan bagian navigasi kembali melalui `<div class="back-area">`. Di dalamnya

terdapat elemen `Kembali` yang berfungsi sebagai tombol untuk kembali ke halaman utama. Penggunaan class `btn-back` memungkinkan tombol ini diberi gaya khusus melalui CSS agar tampil konsisten dengan tema halaman. Dengan adanya kedua tombol ini, pengguna memiliki kontrol lebih mudah, baik untuk mengubah mode tampilan maupun kembali ke halaman sebelumnya, sehingga pengalaman penggunaan halaman menjadi lebih nyaman dan intuitif.

7. Membuat Footer

```
65 <footer class="site-footer">
66   <p>Informasi Pencemaran Lingkungan - R Sy Celvin Putra Bhayangkara</p>
67 </footer>
```

Langkah ini menambahkan bagian footer pada halaman galeri menggunakan elemen `<footer class="site-footer">`. Bagian footer ini berfungsi sebagai penutup halaman dan biasanya berada pada bagian paling bawah konten. Di dalamnya terdapat elemen `<p>` yang menampilkan teks “Informasi Pencemaran Lingkungan - R Sy Celvin Putra Bhayangkara,” yang menunjukkan identitas pembuat atau pemilik halaman. Dengan menempatkan informasi ini pada footer, halaman menjadi lebih lengkap secara struktur dan tampak lebih profesional. Footer ini juga memberikan kesan konsistensi karena sama seperti footer pada halaman utama, sehingga seluruh tampilan website memiliki identitas yang seragam.

8. Menutup Dokumen HTML

```
69 </body>
70 </html>
```

Langkah ini menutup seluruh struktur halaman galeri dengan menggunakan tag `</body>` dan `</html>`. Tag `</body>` berfungsi menandai akhir dari seluruh konten yang ditampilkan di halaman, termasuk header, galeri gambar, tombol interaksi, serta footer. Sementara itu, tag `</html>` digunakan untuk menutup dokumen HTML secara keseluruhan. Penutupan ini sangat penting agar browser dapat memproses dan membaca struktur halaman dengan benar tanpa error. Dengan ditutupnya kedua tag ini, halaman galeri selesai disusun dan siap untuk ditampilkan sepenuhnya kepada pengguna.

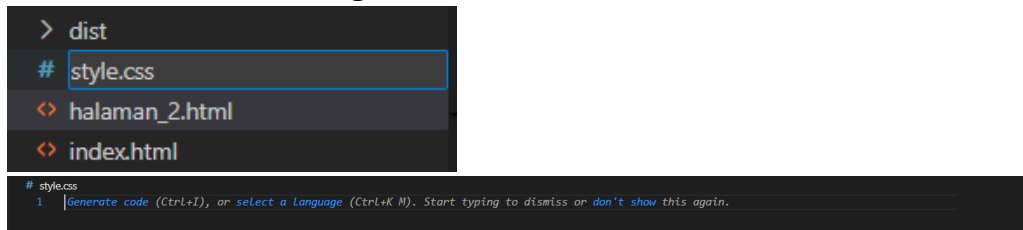
C. Membuat Style Website

Pada tahap ini dilakukan proses pembuatan style atau tampilan visual website menggunakan file `css`. Style dibuat untuk mengatur warna, tata letak, ukuran teks, tampilan tombol, responsivitas, dan keseluruhan estetika halaman. Pembuatan style dimulai dari pengaturan dasar seperti reset margin dan padding agar seluruh elemen memiliki tampilan yang konsisten di berbagai browser. Selanjutnya ditentukan tema warna coklat, sesuai permintaan, yang digunakan pada background, header, footer, dan elemen lain agar website memiliki identitas visual yang seragam.

Bagian header dan footer diberi style agar tampil rapi dengan jarak yang proporsional, sementara teks identitas dan judul dibuat lebih besar agar mudah terlihat. Pada halaman utama, setiap artikel masalah pencemaran diberi card style dengan batas, padding, serta jarak antar elemen agar informasi mudah dibaca. Pada halaman kedua, style difokuskan pada tampilan galeri dengan pengaturan grid sehingga gambar tersusun rapi. Website juga dibuat responsif, artinya tata letak berubah otomatis sesuai ukuran layar: satu kolom pada mobile, dua kolom pada tablet, dan tiga kolom pada desktop.

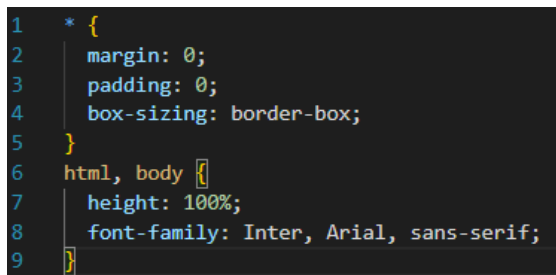
Selain itu, tombol pengubah mode tampilan dibuat dengan style khusus sehingga mudah dikenali dan digunakan oleh pengunjung website. Secara keseluruhan, proses pembuatan style bertujuan memastikan website terlihat menarik, mudah digunakan, dan nyaman dilihat pada berbagai perangkat. Berikut ini langkah – langkah membuat style dalam file css:

1. Memulai dari File Kosong



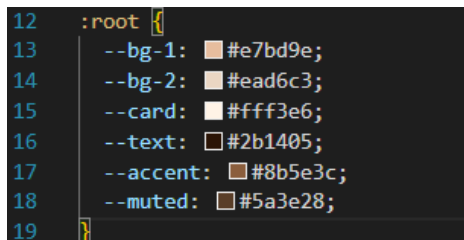
Langkah ini membuat file style website dengan menggunakan file style.css untuk mendesain halaman web yang telah di bikin setelah membuat file html yang sudah jadi.

2. Menambahkan Reset Dasar



Langkah ini melakukan pengaturan dasar pada CSS agar tampilan website menjadi rapi dan konsisten di semua browser. Bagian `* { margin: 0; padding: 0; box-sizing: border-box; }` digunakan sebagai *CSS reset* untuk menghilangkan jarak bawaan pada setiap elemen, sehingga seluruh elemen dapat diatur dari awal tanpa gangguan bawaan browser. Penggunaan `box-sizing: border-box` membantu agar ukuran elemen lebih mudah dikontrol karena padding dan border dihitung ke dalam total ukuran elemen. Langkah ini juga mengatur tampilan dasar untuk elemen `html` dan `body` dengan menetapkan `height: 100%` sehingga halaman dapat memanfaatkan tinggi layar secara penuh. Selain itu, font standar website diatur menggunakan `font-family: Inter, Arial, sans-serif` agar seluruh teks terlihat modern, bersih, dan mudah dibaca.

3. Mendefinisikan Tema Warna Coklat



Langkah ini mendefinisikan variabel warna menggunakan selector `:root` pada CSS. Pendekatan ini digunakan agar seluruh warna yang dipakai pada website lebih mudah dikelola, diubah, dan diterapkan secara konsisten. Variabel seperti `--bg-1` dan `--bg-2` berfungsi sebagai warna dasar (background) dengan nuansa coklat muda sesuai tema website. Variabel `--card` digunakan untuk warna latar card yang menampilkan informasi atau gambar agar tampak lebih lembut dan nyaman dilihat. Selain itu, variabel `--text` ditetapkan sebagai warna utama teks yang menggunakan coklat gelap agar kontras dan mudah dibaca. Variabel `--accent` menjadi warna

pendukung untuk elemen penting seperti tombol dan judul, sehingga terlihat menonjol. Terakhir, `--muted` digunakan untuk warna teks atau elemen sekunder, misalnya label kecil atau penjelasan tambahan. Dengan menyimpan semua warna dalam variabel, CSS menjadi lebih terstruktur, serta memudahkan perubahan tema tanpa harus mengedit warna satu per satu.

4. Mengatur Body Umum

```
22 body {  
23   background: var(--bg-1);  
24   color: var(--text);  
25   line-height: 1.6;  
26 }
```

Style pada bagian **body** digunakan untuk mengatur tampilan dasar seluruh halaman web. Background diatur menggunakan variabel tema `--bg-1`, yaitu warna coklat muda yang memberikan kesan hangat dan konsisten dengan tema yang digunakan. Warna teks menggunakan `--text`, yaitu coklat gelap, agar tulisan memiliki kontras yang jelas dan mudah dibaca. Selain itu, properti `line-height: 1.6` diterapkan untuk memberikan jarak antar baris yang nyaman sehingga paragraf terlihat lebih rapi dan tidak terlalu rapat. Pengaturan ini menjadi fondasi visual utama sebelum elemen-elemen lain, seperti card, tombol, dan layout, diberi style tambahan.

5. Mode Indicator

```
29 .mode-indicator {  
30   position: fixed;  
31   top: 20px;  
32   left: 20px;  
33   padding: 10px 18px;  
34   background: #4b3629;  
35   color: white;  
36   border-radius: 12px;  
37   font-weight: 700;  
38   box-shadow: 0 6px 15px rgba(0,0,0,0.2);  
39   z-index: 999;  
40 }
```

Style `.mode-indicator` digunakan untuk menampilkan kotak kecil penanda mode tampilan (Rapi atau Acak) di sudut kiri atas layar. Elemen ini dibuat `fixed`, sehingga tetap terlihat meskipun halaman digulir. Posisi ditempatkan pada `top: 20px` dan `left: 20px` agar mudah ditemukan pengguna. Kotaknya diberi warna coklat gelap (`#4b3629`) dengan teks berwarna putih untuk kontras yang kuat. Padding dibuat cukup lebar agar indikator nyaman dibaca, ditambah `border-radius: 12px` untuk tampilan yang lebih halus. Efek `box-shadow` digunakan untuk memberikan kedalaman sehingga elemen terlihat seperti “mengambang” di atas halaman. Z-index tinggi memastikan indikator tidak tertutupi elemen lain.

6. Header Halaman

```
43 .site-header {  
44   text-align: center;  
45   padding: 22px 20px;  
46   border-bottom: 2px solid rgba(0,0,0,0.06);  
47   background: rgba(255,255,255,0.06);  
48 }  
49  
50 .identity h1 {  
51   font-size: 1.3rem;  
52   font-weight: 700;  
53 }  
54 .identity p {  
55   font-size: 1rem;  
56   color: var(--muted);  
57   margin-top: 4px;  
58 }
```

Bagian `.site-header` berfungsi sebagai header utama halaman yang menampilkan identitas pengguna atau judul website. Teks di bagian ini disejajarkan ke tengah (`text-align: center`) agar tampil rapi dan fokus. Padding yang cukup lebar diberikan supaya area header terasa lega dan tidak terlalu rapat terhadap tepi halaman. Di bagian bawah header ditambahkan garis tipis (`border-bottom`) untuk memisahkan header dari konten utama, sementara background transparan lembut (`rgba(255,255,255,0.06)`) memberikan efek lapisan halus agar sesuai dengan tema warna coklat. Untuk bagian `.identity h1`, ukuran teks dibuat menonjol dengan font-size lebih besar serta font-weight tebal untuk memperjelas judul atau nama. Sedangkan `.identity p` diberi ukuran font sedikit lebih kecil dan warna menggunakan `var(--muted)` sehingga tampak lebih lembut dan tidak mendominasi. Margin kecil di bagian atas paragraf menambah jarak dari judul agar tampil lebih terstruktur dan mudah dibaca.

7. Container Pengatur Lebar Konten

```
62 .container {  
63   max-width: 1100px;  
64   margin: 26px auto;  
65   padding: 0 16px 40px;  
66   width: 100%;  
67 }
```

Bagian `.container` digunakan sebagai pembungkus utama agar seluruh isi halaman tetap berada dalam lebar yang rapi dan tidak melebar terlalu jauh di layar besar. Properti `max-width: 1100px` membatasi lebar maksimal konten sehingga tampil proporsional, sementara `width: 100%` memastikan container tetap menyesuaikan pada layar kecil. Pengaturan `margin: 26px auto` membuat container berada tepat di tengah halaman secara horizontal. Bagian padding diberikan di kiri, kanan, dan bawah untuk memberi ruang agar konten tidak terlalu menempel pada tepi layar, sehingga keseluruhan tampilan lebih nyaman dilihat dan teratur.

8. Bagian Intro

```
70 .intro {  
71   background: white;  
72   padding: 16px;  
73   border-radius: 12px;  
74   box-shadow: 0 6px 18px rgba(0,0,0,0.06);  
75 }  
76 .intro h2 { margin-bottom: 8px; color: var(--text); }  
77 .intro p { color: #3b2b20; }
```

Langkah ini digunakan sebagai kotak penjelasan utama pada halaman. Latar belakang putih membuat isi teks lebih mudah dibaca dan kontras dengan tema coklat halaman. Padding memberi ruang dalam agar teks tidak terlalu menempel pada tepi kotak, sedangkan `border-radius: 12px` membuat sudut tampak halus dan modern. Efek bayangan (`box-shadow`) menambah kedalaman sehingga kotak terlihat terangkat dari latar. Pada bagian teks, judul `<h2>` diberi jarak bawah agar lebih rapi, sementara paragraf `<p>` memakai warna coklat gelap yang tetap nyaman dibaca dan sesuai dengan tema warna keseluruhan.

9. Daftar Masalah Pencemaran

```

80 .problems { margin-top: 18px; }
81 .problems h2 { margin-bottom: 12px; color: #3b2b20; }
82
83 .problem {
84   margin-bottom: 12px;
85   background: linear-gradient(180deg, #fff, #fffaf6);
86   padding: 12px 14px;
87   border-radius: 10px;
88   border-left: 6px solid var(--accent);
89   transition: transform .15s ease, box-shadow .15s ease;
90 }
91 .problem:hover {
92   transform: translateY(-4px);
93   box-shadow: 0 8px 22px rgba(0,0,0,0.08);
94 }
95 .problem .title {
96   font-size: 1.05rem;
97   margin-bottom: 6px;
98   color: var(--text);
99 }
100 .problem .desc { color: #4b3629; }

```

Langkah ini digunakan sebagai wadah untuk seluruh daftar masalah pencemaran. Properti `margin-top: 18px` memberi jarak dari elemen sebelumnya agar tata letaknya lebih rapi. Pada judulnya (`.problems h2`), jarak bawah ditambahkan supaya teks tidak menempel dengan daftar isi, serta warna coklat gelap digunakan agar tetap sesuai dengan tema halaman. Setiap item masalah berada dalam kelas `.problem`, yang tampil seperti kartu kecil dengan latar gradasi lembut antara putih dan krem. Padding diberikan agar isi teks memiliki ruang yang nyaman, sementara `border-radius: 10px` membuat sudut lebih halus. Garis tepi kiri berwarna coklat aksen (`var(--accent)`) digunakan sebagai penanda visual agar setiap item terlihat lebih tegas. Efek transisi ditambahkan sehingga ketika di-hover, kartu sedikit terangkat dengan `transform: translateY(-4px)` dan bayangan lebih besar muncul, memberikan efek interaktif yang responsif. Bagian judul dalam kartu (`.problem .title`) dibuat lebih besar dan diberi jarak bawah agar mudah dibaca. Sedangkan deskripsi (`.problem .desc`) menggunakan warna coklat yang lembut sehingga nyaman untuk dilihat.

10. Desain Tombol

```

103 .btn-area {
104   margin-top: 24px;
105   display: flex;
106   justify-content: center;
107 }
108
109 .mode-toggle-btn,
110 .btn-gambar,
111 .btn-back {
112   padding: 12px 22px;
113   background: var(--accent);
114   color: white !important;
115   border-radius: 12px;
116   font-weight: 700;
117   text-decoration: none;
118   border: none;
119   cursor: pointer;
120   transition: 0.2s;
121   display: inline-block;
122 }
123 .mode-toggle-btn:hover,
124 .btn-gambar:hover,
125 .btn-back:hover {
126   opacity: 0.9;
127 }

```

Langkah ini berfungsi sebagai wadah untuk menempatkan tombol di tengah halaman menggunakan `display: flex` dan `justify-content: center`. Ruang atas (`margin-top: 24px`) ditambahkan agar tombol tidak terlalu dekat dengan elemen sebelumnya. Untuk tampilan tombol, kelas `.mode-toggle-btn`, `.btn-gambar`, dan `.btn-back` dibuat seragam menggunakan padding yang cukup lebar sehingga tombol terasa nyaman ditekan. Latar belakang coklat aksen (`var(--accent)`) digunakan agar tombol konsisten dengan tema warna situs, sedangkan teks

dibuat putih agar kontras dan mudah dibaca. Sudut yang membulat dengan border-radius: 12px memberikan kesan modern, dan efek transisi membuat tombol tampak halus saat berubah. Ketika kursor diarahkan ke tombol, efek opacity: 0.9 membuat tampilan sedikit menggelap sebagai umpan balik visual kepada pengguna.

11. Halaman Gambar

```
130 .images-header {
131   text-align: center;
132   padding: 22px 20px;
133   border-bottom: 2px solid #00000006;
134   background: #ffffff04;
135 }
136 .images-header h1 {
137   font-size: 1.2rem;
138   color: var(--text);
139 }
140 .images-header p {
141   color: #6b4f3a;
142   margin-top: 6px;
143 }
```

Bagian .images-header digunakan sebagai header pada halaman galeri gambar. Elemen ini diberi perataan teks di tengah agar judul tampil simetris dan lebih menarik. Padding yang cukup lebar membuat ruang di dalam header terasa lega, sementara garis tipis pada bagian bawah (border-bottom) membantu memisahkan header dari konten utama. Latar belakang dengan warna putih transparan memberikan efek lembut tanpa mengganggu tema coklat keseluruhan. Untuk judul, ukuran font 1.2rem dipilih agar terlihat jelas, sedangkan teks paragraf diberi warna coklat lebih lembut untuk menciptakan hierarki visual dan kenyamanan saat dibaca.

12. Galeri

```
145 .gallery {
146   display: grid;
147   gap: 20px;
148   padding: 20px;
149 }
150 .card {
151   background: var(--card);
152   border-radius: 12px;
153   overflow: hidden;
154   box-shadow: 0 4px 18px #00000015;
155 }
156 .card img {
157   width: 100%;
158   height: 210px;
159   object-fit: cover;
160 }
161 .card h3 {
162   text-align: center;
163   padding: 12px;
164   color: var(--text);
165 }
```

Langkah ini berfungsi mengatur tata letak gambar agar tersusun rapi menggunakan sistem grid. Jarak antar-elemen diatur dengan gap: 20px sehingga setiap card memiliki ruang yang proporsional dan tidak saling berhimpitan. Padding sebesar 20px juga diberikan untuk memberi jarak antara isi galeri dan tepi halaman. Setiap item galeri dibungkus dalam elemen .card, yang memiliki latar berwarna coklat muda dari variabel --card, sehingga tampil senada dengan tema keseluruhan website. Efek border-radius 12px membuat sudut card lebih halus dan modern, sementara bayangan (box-shadow) memberikan kesan mengangkat sehingga card tampak lebih menonjol. Gambar pada card diatur agar memenuhi seluruh lebar card dengan tinggi tetap 210px melalui properti object-fit: cover, sehingga tampil rapi tanpa distorsi. Judul pada setiap card dibuat berada di tengah dengan padding yang cukup agar nyaman dipandang, serta diberi warna sesuai variabel --text untuk menjaga konsistensi desain.

13. Responsif

```

176 @media (max-width: 600px) {
177   .gallery { grid-template-columns: 1fr; }
178 }
179 @media (min-width: 601px) and (max-width: 992px) {
180   .gallery { grid-template-columns: repeat(2, 1fr); }
181 }
182 @media (min-width: 992px) {
183   .gallery { grid-template-columns: repeat(3, 1fr); }
184 }

```

Tampilan ini digunakan untuk membuat tampilan galeri menjadi responsif agar susunan card dapat menyesuaikan ukuran layar perangkat. Pada layar kecil hingga lebar maksimum 600px, galeri ditampilkan dalam satu kolom sehingga setiap gambar terlihat besar dan mudah dilihat pada perangkat mobile. Ketika layar berada pada rentang lebar 601px hingga 992px, tata letaknya berubah menjadi dua kolom, yang cocok untuk tampilan tablet. Sementara itu, untuk layar berukuran 992px ke atas, galeri disusun dalam tiga kolom menggunakan `grid-template-columns: repeat(3, 1fr)`, sehingga tampilan menjadi lebih lebar, rapi, dan efisien pada perangkat laptop atau desktop. Dengan pengaturan ini, galeri dapat menyesuaikan dirinya secara otomatis untuk memberikan pengalaman pengguna terbaik di berbagai ukuran layar.

14. Footer

```

168 .site-footer {
169   margin-top: auto;
170   padding: 18px 20px;
171   text-align: center;
172   color: var(--muted);
173 }

```

Bagian `.site-footer` digunakan sebagai footer halaman yang menampilkan informasi penutup atau identitas pembuat website. Properti `margin-top: auto` memastikan footer selalu berada di bagian bawah halaman, bahkan jika konten utama tidak memenuhi seluruh tinggi layar. Padding ditambahkan agar teks tidak terlalu menempel pada tepi dan terlihat seimbang secara visual. Teks disejajarkan di tengah (`text-align: center`) agar simetris dan mudah dibaca, sementara warna menggunakan `var(--muted)` yang memberikan efek lembut namun tetap konsisten dengan tema coklat keseluruhan. Dengan pengaturan ini, footer tampil profesional, rapi, dan tidak mengganggu konten utama.

15. Mode Berantakan

```

194 body.mode-berantakan {
195   background: #ffddc7;
196   color: #333;
197 }
198 body.mode-berantakan .mode-indicator {
199   background: yellow;
200   color: purple;
201 }
202 body.mode-berantakan .problem {
203   background: #ffe9a8;
204   border-left-color: cyan;
205 }
206 body.mode-berantakan .btn-gambar,
207 body.mode-berantakan .mode-toggle-btn,
208 body.mode-berantakan .btn-back {
209   background: #00ff66 !important;
210   color: #111 !important;
211 }
212 body.mode-berantakan .card {
213   border-left-color: magenta;
214 }

```

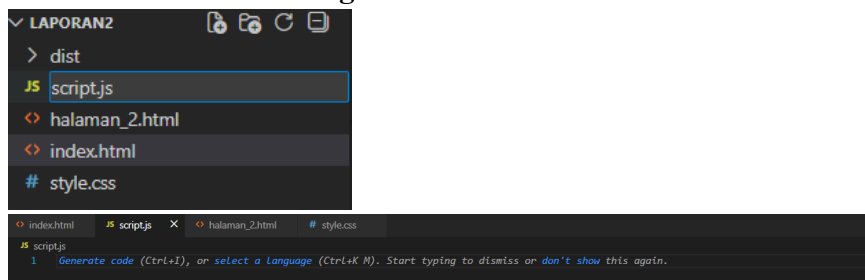
Style ini digunakan untuk membuat tampilan mode berantakan sebagai alternatif dari mode rapi. Ketika kelas `mode-berantakan` diterapkan pada `<body>`, background halaman berubah

menjadi oranye pucat (#ffddc7) dengan teks gelap (#333) agar tetap terbaca. Indikator mode (.mode-indicator) juga berubah warna menjadi kuning dengan teks ungu, sehingga langsung terlihat berbeda dari mode rapi. Setiap kartu masalah (.problem) menggunakan background kuning muda (#ffe9a8) dengan garis tepi kiri berwarna cyan untuk menonjolkan efek acak. Tombol-tombol interaktif seperti .btn-gambar, .mode-toggle-btn, dan .btn-back diberi latar hijau terang (#00ff66) dan teks gelap, menegaskan perbedaan visual yang mencolok. Selain itu, card di halaman galeri juga menampilkan garis kiri berwarna magenta. Secara keseluruhan, mode ini menciptakan tampilan yang penuh warna dan berbeda drastis dari mode rapi, memberikan pengalaman interaktif yang menyenangkan dan jelas terlihat perbedaannya.

D. Script Pada Javascript

Kode ini berfungsi untuk membuat tampilan mode tampilan rapi dengan perpaduan warna dan kontras yang baik dan mode tampilan berantakan, dengan perpaduan warna dan kontras yang buruk. Serta untuk menampilkan interaksi pada setiap judul. Ketika judul diklik, muncul pertanyaan kepada pengguna: “Apakah Anda ingin menyembunyikan penjelasan ini?”. Jika pengguna menyetujuinya, maka paragraf penjelasan untuk judul tersebut disembunyikan. Kemudian buat interaksi untuk kembali menampilkan penjelasannya dengan pertanyaan “Apakah Anda ingin menampilkan penjelasan ini?”. Berikut penjelasan setiap langkah yang dibuat untuk membuat script:

1. Membuat File JS Kosong



Setelah membuat html dan css selanjutnya membuat untuk script agar penggunaan metode intraksi shoe / hide dan mode tampilan agar berjalan. Langkah pertama membuat file script js untuk menyimpan source code yang di buat untuk menyimpan script untuk penggunaan intraksi dan mode tampilan.

2. Interaksi Show / Hide Penjelasan

```

2 document.querySelectorAll(".problem").forEach(problem => {
3
4     const title = problem.querySelector(".title");
5     const desc = problem.querySelector(".desc");
6
7     desc.style.display = "block";
8     problem.dataset.hidden = "false";
9
10    function toggleDesc() {
11        let hidden = problem.dataset.hidden === "true";
12
13        if (!hidden) {
14            if (confirm("Apakah Anda ingin menyembunyikan penjelasan ini?")) {
15                desc.style.display = "none";
16                problem.dataset.hidden = "true";
17            }
18        } else {
19            if (confirm("Apakah Anda ingin menampilkan penjelasan ini?")) {
20                desc.style.display = "block";
21                problem.dataset.hidden = "false";
22            }
23        }
24    }
25 }

```

Script ini digunakan untuk menambahkan **fungsi show/hide** pada setiap item masalah (.problem) di halaman utama. Pertama, script memilih semua elemen dengan kelas .problem dan mengambil elemen judul (.title) serta deskripsi (.desc) di dalamnya. Secara default, deskripsi ditampilkan (desc.style.display = "block") dan status disimpan pada atribut data-hidden sebagai "false". Fungsi toggleDesc() dibuat untuk menangani interaksi pengguna: ketika elemen diklik, script memeriksa apakah deskripsi sedang terlihat atau tersembunyi. Jika terlihat, muncul konfirmasi untuk menyembunyikan deskripsi, dan jika disetujui, deskripsi disembunyikan dan data-hidden diubah menjadi "true". Sebaliknya, jika deskripsi tersembunyi, konfirmasi muncul untuk menampilkannya kembali. Dengan cara ini, pengguna dapat mengontrol informasi mana yang ingin mereka lihat atau sembunyikan secara interaktif.

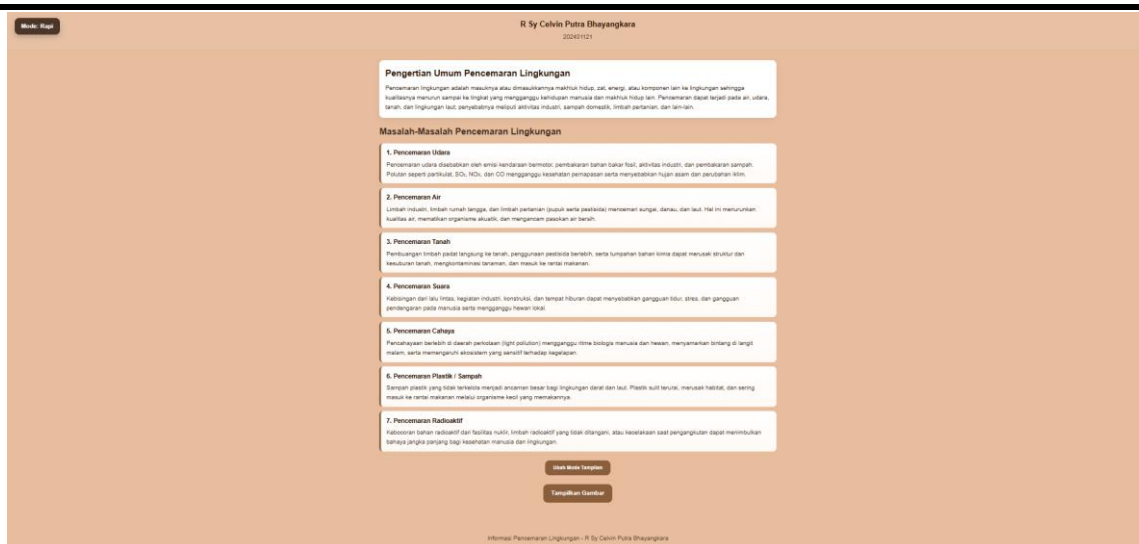
3. Mode Tampilan (Rapi / Berantakan)

```
29 function updateModeIndicator() {
30   const berantakan = document.body.classList.contains("mode-berantakan");
31   const textMode = berantakan ? "Berantakan" : "Rapi";
32
33   document.querySelectorAll(".mode-indicator").forEach(el => {
34     el.textContent = "Mode: " + textMode;
35   });
36 }
37
38 document.addEventListener("DOMContentLoaded", () => {
39
40   const toggleBtn = document.getElementById("toggleModeBtn");
41
42   if (toggleBtn) {
43     toggleBtn.addEventListener("click", () => {
44       document.body.classList.toggle("mode-berantakan");
45       document.body.classList.toggle("mode-rapi");
46       updateModeIndicator();
47     });
48   }
49
50   updateModeIndicator();
51 });
```

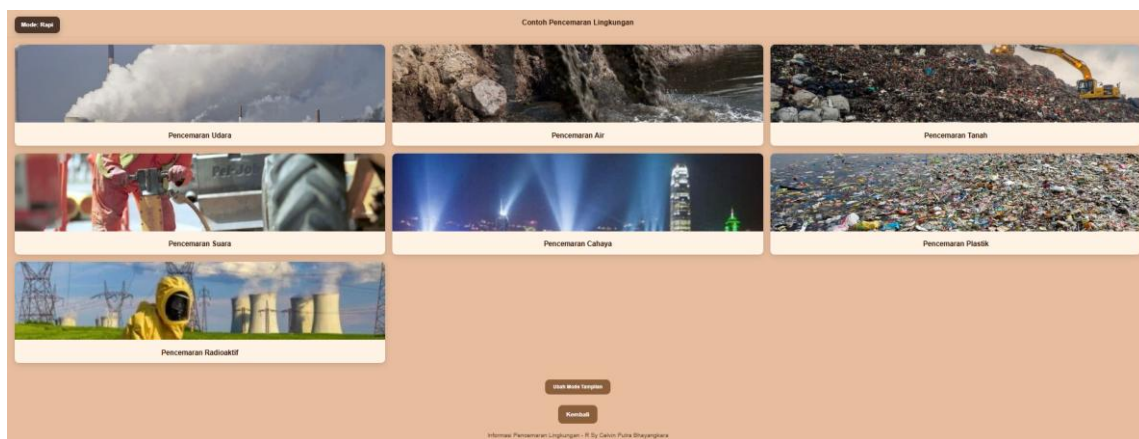
Langkah ini berfungsi untuk mengatur dan menampilkan mode tampilan pada halaman, yaitu mode “Rapi” dan mode “Berantakan”. Fungsi updateModeIndicator() memeriksa apakah <body> memiliki kelas mode-berantakan. Jika iya, teks indikator diubah menjadi “Mode: Berantakan”; jika tidak, menjadi “Mode: Rapi”. Seluruh elemen dengan kelas .mode-indicator kemudian diperbarui sesuai kondisi ini. Setelah halaman selesai dimuat (DOMContentLoaded), script mencari tombol dengan ID toggleModeBtn dan menambahkan event listener untuk klik. Ketika tombol diklik, kelas mode-berantakan dan mode-rapi pada <body> saling ditukar menggunakan classList.toggle(), sehingga tampilan halaman berubah secara visual. Setelah itu, updateModeIndicator() dipanggil untuk memperbarui teks indikator sesuai mode yang aktif. Dengan cara ini, pengguna dapat mengubah mode tampilan halaman secara interaktif dan indikator selalu menampilkan status terkini.

E. Hasil Output Membuat Website

- **Halaman Pertama:**



• Halaman Kedua:



II. Kesimpulan

Website Informasi Pencemaran Lingkungan ini dibangun dengan struktur HTML yang rapi, dibagi menjadi header, konten utama, tombol interaktif, galeri gambar, dan footer. Header menampilkan identitas pembuat dan indikator mode tampilan, sedangkan konten utama terdiri dari pengertian pencemaran dan daftar masalah pencemaran yang ditampilkan dalam kartu interaktif. Setiap kartu memiliki judul dan deskripsi yang bisa ditampilkan atau disembunyikan melalui konfirmasi interaktif menggunakan JavaScript. Tombol-tombol seperti “Ubah Mode Tampilan” dan navigasi ke halaman galeri dibuat agar mudah digunakan dan konsisten dengan tema visual.

Tampilan website diatur menggunakan CSS dengan tema warna coklat untuk menciptakan estetika yang hangat dan konsisten. Layout dibuat responsif dengan sistem grid agar galeri gambar tersusun rapi pada berbagai ukuran layar: satu kolom pada mobile, dua kolom pada tablet, dan tiga kolom pada desktop. Elemen seperti kartu, tombol, header, dan footer diberi padding, border-radius, dan efek bayangan untuk meningkatkan kenyamanan visual dan interaktivitas. Selain mode rapi, website juga menyediakan mode berantakan sebagai mode alternatif yang berbeda secara visual, diatur menggunakan kelas khusus dan perubahan warna yang mencolok.

JavaScript digunakan untuk menambahkan interaktivitas, termasuk toggle mode tampilan dan show/hide deskripsi setiap masalah pencemaran. Fungsi-fungsi ini memastikan pengguna dapat

mengontrol informasi yang ingin mereka lihat serta mengetahui mode tampilan aktif melalui indikator. Secara keseluruhan, website ini dirancang agar informasi tentang pencemaran lingkungan tersampaikan dengan jelas, tampilan menarik, interaktif, dan dapat diakses dengan baik di berbagai perangkat.