

# **LAPORAN PRAKTIKUM**

## **PEMROGRAMAN WEB D**



**SOFTWARE ENGINEERING  
LABORATORY**

NAMA : Egi vioneuva pradana

NIM 202431060

KELAS : D

DOSEN : Muhammad Fadli Prathama

NO.PC : 10

ASISTEN : 1. Sakura Amastasya Salsabila Setiyanto

2. Phazel Kesra Arivi

3. Muhammad Farhan Fahrezy

4. Kashrina Masyid Azka

**INSTITUT TEKNOLOGI PLN**  
**TEKNIK INFORMATIKA**

**2025**

## PENCEMARAN LINGKUNGAN

### 1.HTML

```
Laporan2 > index.html > html > body.mode-rapi > main > div.control-panel > button#tampilkan
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>Laporan Praktikum: Pencemaran Lingkungan</title>
7      <link rel="stylesheet" href="style.css">
8  </head>
```

#### <!DOCTYPE html>

- Artinya: Saya mendeklarasikan bahwa ini adalah dokumen web terbaru (HTML5). Ini seperti label di gerbang utama.

#### <html lang="id">

- Artinya: Ini adalah pembungkus semua konten. lang="id" memberi tahu mesin pencari dan perangkat lunak bahwa bahasa utama halaman ini adalah Bahasa Indonesia.

#### <head>

- Artinya: Ini adalah "kepala" atau metadata dokumen. Isinya adalah informasi untuk browser, bukan untuk dilihat pengguna.

#### <meta charset="UTF-8">

- Artinya: Saya mengatur agar browser bisa membaca semua karakter, termasuk yang ada di Bahasa Indonesia, tanpa ada yang rusak (encoding standar internasional).

#### <meta name="viewport" content="width=device-width, initial-scale=1.0">

- Artinya: Pengaturan ini penting sekali! Ini memastikan halaman web saya bisa tampil responsif dan menyesuaikan lebarnya dengan layar perangkat apa pun (HP, tablet, atau PC).

#### <title>Laporan Praktikum: Pencemaran Lingkungan</title>

- Artinya: Teks ini akan muncul sebagai judul di tab browser Anda.

#### <link rel="stylesheet" href="style.css">

- Artinya: Saya memanggil file desain (CSS) bernama style.css. Semua warna, ukuran font, dan tata letak halaman ini diatur di file tersebut.

#### </head>

- Artinya: Selesai dengan informasi rahasia.

```

9   <body class="mode-rapi">
10
11   <header>
12     <h1>Laporan Praktikum Pemrograman Web D</h1>
13     <p>Nama: Egi Vioneuva Pradana | NIM: 202431060</p>
14   </header>
15
16   <main>
17     <h2>Informasi Pencemaran Lingkungan</h2>
18     <p class="penjelasan-umum">Pencemaran lingkungan adalah masuknya zat, energi, atau komponen lain ke dalam lingkungan oleh kegiatan manusia sehingga kualitasnya turun sampai ke tingkat tertentu yang menyebabkan lingkungan tidak dapat berfungsi sesuai peruntukannya. Ada berbagai jenis pencemaran yang mengancam keselarasan ekosistem.</p>
19
20   <section id="pencemaran-list">
21   </section>
22
23   <hr>
24
25   <div class="control-panel">
26     <button id="toggleModeBtn">Ubah Mode Tampilan (Saat Ini: Rapi)</button>
27   </div>
28
29   <div class="control-panel">
30     <button id="tampilkanGambarBtn">Tampilkan Gambar 7 Pencemaran</button>
31   </div>
32
33   <hr>
34
35

```

## <body class="mode-rapi">

- Artinya: Ini adalah "tubuh" halaman. Kelas mode-rapi adalah penanda yang kemungkinan digunakan untuk menentukan tema tampilan (misalnya, tema standar/rapi) yang bisa diubah-ubah.

## <header>

- Artinya: Ini adalah bagian pembuka atau kop halaman.
  - <h1>...</h1> adalah judul utama, yang paling besar.
  - <p>...</p> adalah paragraf untuk informasi detail (Nama dan NIM Anda).

## <main>

- Artinya: Saya memulai bagian konten utama yang paling penting di halaman ini.

## <h2>Informasi Pencemaran Lingkungan</h2>

- Artinya: Ini adalah subjudul untuk bagian informasi.

## <p class="penjelasan-umum">...</p>

- Artinya: Ini adalah paragraf yang berisi definisi pencemaran. Kelas penjelasan-umum bisa jadi digunakan untuk memberi gaya khusus pada teks ini.

## <section id="pencemaran-list">

- Artinya: Saya menyiapkan sebuah wadah (section) ber-ID unik. Wadah ini disiapkan untuk menampung daftar-daftar jenis pencemaran (saat ini masih kosong).

## <hr>

- Artinya: Membuat garis horizontal sebagai pemisah visual antar bagian.

## <div class="control-panel">

- Artinya: Saya membuat wadah yang mengelompokkan tombol-tombol kontrol.

## <button id="toggleModeBtn">Ubah Mode Tampilan...</button>

- Artinya: Ini adalah tombol pertama. ID unik (toggleModeBtn) berfungsi agar kode JavaScript tahu tombol mana yang harus dioperasikan untuk mengubah tema.

**<button id="tampilkanGambarBtn">Tampilkan Gambar 7 Pencemaran</button>**

- Artinya: Ini adalah tombol kedua. ID unik (tampilkanGambarBtn) akan dipanggil oleh JavaScript untuk menampilkan galeri gambar saat diklik.

```
35      <section id="halamanGambar" class="tersembunyi">
36          <h3>7 Contoh Visual Pencemaran Lingkungan</h3>
37          <div id="gambarContainer" class="card-layout">
38              </div>
39      </section>
40
41  </main>
42
43  <footer>
44      <button onclick="history.back()">Kembali ke Halaman Sebelumnya</button>
45  </footer>
46
47  <script src="script.js"></script>
48
49 </body>
50 </html>
```

- **<section id="halamanGambar" class="tersembunyi">**
  - Artinya: Saya membuat wadah galeri gambar. Kelas tersembunyi di sini sangat penting—ini yang membuat galeri ini tidak terlihat saat halaman pertama kali dibuka, sampai tombol di Baris 30 diklik.
- **<div id="gambarContainer" class="card-layout"></div>**
  - Artinya: Ini adalah tempat kosong (kontainer) di mana gambar-gambar akan dimasukkan (mungkin menggunakan JavaScript). Kelas card-layout mengatur susunan gambar, mungkin dalam format kartu.
- **<footer>**
  - Artinya: Ini adalah bagian kaki halaman, biasanya untuk informasi hak cipta atau navigasi.
- **<button onclick="history.back()">...</button>**
  - Artinya: Ini adalah tombol navigasi. Perintah onclick="history.back()" adalah kode JavaScript sederhana yang secara otomatis mengembalikan pengguna ke halaman web sebelumnya di riwayat browser mereka.
- **<script src="script\_js.js"></script>**
  - Artinya: Ini adalah bagian yang menghubungkan otak fungsional halaman. Semua kode JavaScript yang membuat tombol berfungsi, mengganti mode tampilan, dan memuat gambar, semuanya ada di file script\_js.js ini.

- </body>
  - Artinya: Akhir dari konten yang terlihat.
- </html>
  - Artinya: Akhir dari seluruh dokumen web.

## 2.STYLE.CSS

```
Laporan2 > # style.css > ↵ body.mode-berantakan
1  /* Tema Warna Halaman Default: HIJAU (Mode Rapi) */
2  :root {
3    --color-primary: #388e3c;
4    --color-secondary: #c8e6c9;
5    --color-text: #212121;
6    --color-background: #c8e6c9;
7    --color-card-bg: #dcedc8;
8  }
9
10 body.mode-berantakan {
11   /* Mode Berantakan: Warna dan Kontras Buruk */
12   --color-primary: #ffeb3b;
13   --color-secondary: #00bcd4;
14   --color-text: #f44336;
15   --color-background: #795548;
16   --color-card-bg: #e0f7fa;
17 }
18
19 }
```

Bagian ini sangat penting karena menggunakan Variabel CSS (dimulai dengan --color-). Ini memungkinkan Anda mengubah seluruh skema warna halaman hanya dengan memodifikasi beberapa baris saja.

### A. Tema Default: Mode Rapi

**Kode mendefinisikan variabel-variabel warna di dalam :root (yang berarti variabel ini berlaku untuk seluruh dokumen).**

- --color-primary (Hijau 388e3c): Warna utama, digunakan untuk judul, border, dan tombol.
- --color-secondary (Coklat 8e6c9c): Warna sekunder, digunakan untuk border dan latar belakang footer.
- --color-text, --color-background, --color-card-bg: Mendefinisikan warna teks (hitam), latar belakang halaman, dan latar belakang elemen kartu. Ini adalah Tema Hijau Default.

### B. Tema Alternatif: Mode Berantakan

- Ini adalah tema khusus yang hanya aktif ketika elemen <body> memiliki kelas mode-berantakan.
- Semua variabel warna (primary, secondary, text, background) ditimpak dengan kombinasi warna yang kontras buruk (kuning, biru, merah terang). Tujuannya
- adalah untuk menunjukkan mode tampilan yang benar-benar berbeda (dan sengaja dibuat kurang nyaman dilihat).

```

19
20 body {
21   font-family: Arial, sans-serif;
22   line-height: 1.6;
23   margin: 0;
24   padding: 0;
25   transition: background-color 0.5s, color 0.5s;
26   background-color: var(--color-background);
27   color: var(--color-text);
28 }
29
30 header {
31   background-color: var(--color-primary);
32   color: white;
33   padding: 20px;
34   text-align: center;
35 }
36
37 main {
38   padding: 20px;
39 }
40

```

Ini mengatur tampilan standar teks, margin, dan tata letak dasar.

- **body : Mengatur seluruh tubuh halaman.**
  - Menggunakan font Arial, sans-serif.
  - line-height: 1.6 membuat jarak antar baris teks lebih lega dan nyaman dibaca.
  - Menghilangkan margin default (margin: 0;).
  - Mengatur warna latar belakang dan warna teks menggunakan variabel yang sudah didefinisikan di atas (misalnya, var(--color-background)).
  - transition: ... menambahkan efek animasi halus (0.5 detik) ketika warna latar belakang atau teks diubah (ini penting saat berganti tema).
- **header : Mengatur bagian kop (judul laporan).**
  - Latar belakangnya menggunakan warna utama (--color-primary).
  - Teksnnya berwarna putih (white).
  - Teks diatur rata tengah (text-align: center) dengan padding 20px di sekelilingnya.
- **main : Konten utama diberi jarak (padding: 20px) dari tepi layar.**

```

41 h2 {
42   border-bottom: 2px solid var(--color-primary);
43   padding-bottom: 10px;
44   color: var(--color-primary);
45 }
46
47 .judul-pencemaran {
48   cursor: pointer;
49   background-color: var(--color-card-bg);
50   padding: 15px;
51   margin-bottom: 10px;
52   border-radius: 5px;
53   border-left: 5px solid var(--color-primary);
54 }
55
56 .penjelasan-paragraf {
57   margin: 10px 0 20px 20px;
58   padding: 10px;
59   border: 1px dashed var(--color-secondary);
60 }
61
62 /* Kelas untuk menyembunyikan elemen */
63 .tersembunyi {
64   display: none;
65 }
66

```

Bagian ini mengatur detail visual dari elemen-elemen spesifik.

- **h2** : Subjedul level 2 diberi border bawah tebal (2px solid) dan warnanya menggunakan warna utama (--color-primary).
- **.judul-pencemaran** : Ini adalah gaya khusus untuk elemen yang mungkin berupa tautan atau daftar.
  - **cursor: pointer** mengubah kursor menjadi bentuk tangan saat diarahkan (memberi kesan bisa diklik).
  - **Memiliki latar belakang warna kartu (--color-card-bg)** dan diberi border tebal di sebelah kiri dengan warna utama.
- **.penjelasan-paragraf** : Paragraf diberi jarak (margin dan padding) yang lega, dan ditambahkan border putus-putus (dashed) dengan warna sekunder untuk membedakannya.
- **.tersembunyi** : Ini adalah kelas ajaib untuk menyembunyikan elemen. display: none; akan memastikan elemen yang memiliki kelas ini (seperti galeri gambar) tidak ditampilkan sama sekali di halaman.

```

67  /* Responsive Web Design (RWD) untuk Card Gambar */
68  .card-layout {
69    display: grid;
70    gap: 20px;
71    padding: 10px 0;
72  }
73
74  /* Mobile (<=600px): 1 kolom */
75  @media (max-width: 600px) {
76    .card-layout {
77      grid-template-columns: 1fr; /* 1 kolom */
78    }
79  }
80
81  /* Tablet (601px - 992px): 2 kolom */
82  @media (min-width: 601px) and (max-width: 992px) {
83    .card-layout {
84      grid-template-columns: 1fr 1fr; /* 2 kolom */
85    }
86  }
87
88  /* Desktop (>=993px): 3 kolom */
89  @media (min-width: 993px) {
90    .card-layout {
91      grid-template-columns: 1fr 1fr 1fr; /* 3 kolom */
92    }
93  }
94

```

Bagian ini memastikan galeri gambar Anda terlihat rapi di semua perangkat. Tata letak diatur menggunakan CSS Grid.

- **.card-layout :**
  - Secara default, diatur sebagai display: grid dengan jarak antar kartu 20px.
- **Media Query untuk Ponsel :**
  - Ketika lebar layar maksimal 600px (Ponsel), grid diatur hanya memiliki 1 kolom (grid-template-columns: 1fr;).
- **Media Query untuk Tablet :**
  - Ketika lebar layar antara 601px hingga 992px (Tablet), grid diatur menjadi 2 kolom yang sama lebarnya.

- **Media Query untuk Desktop :**

- Ketika lebar layar lebih dari 993px (Desktop), grid diatur menjadi 3 kolom yang sama lebarnya.

```

95  /* Styling Card Gambar */
96  .pencemaran-card img {
97    width: 100%;
98    height: 180px;
99    display: block;
100   object-fit: cover;
101 }
103
104 .pencemaran-card {
105   padding-bottom: 0;
106 }
107
108 .pencemaran-card p {
109   padding: 10px;
110   margin: 0;
111   font-size: 0.9em;
112   text-align: center;
113   background-color: var(--color-card-bg);
114 }
115
116 /* Styling Tombol */
117 button {
118   padding: 10px 15px;
119   border: none;
120   border-radius: 5px;
121   cursor: pointer;
122   background-color: var(--color-primary);
123   color: white;
124   transition: background-color 0.3s;
125 }
126
127 button:hover {
128   background-color: #2e6e32;
129 }
130
131 .control-panel {
132   text-align: center;
133   margin: 20px 0;
134 }
135
136 footer {
137   text-align: center;
138   padding: 20px;
139   margin-top: 20px;
140   background-color: var(--color-secondary);
141 }

```

**.pencemaran-card img :** Mengatur tampilan gambar di dalam kartu. Gambar diatur agar tingginya sama (180px) dan menggunakan object-fit: cover untuk memastikan gambar memenuhi wadahnya tanpa terdistorsi (memotong bagian yang tidak perlu).

#### Tombol (button) :

- Tombol diberi padding yang nyaman, tanpa border, dan sudutnya membulat (border-radius: 5px).
- Latar belakang tombol menggunakan warna utama (--color-primary) dan teksnya putih.
- transition lagi-lagi menambahkan efek halus saat warna tombol berubah.

**button:hover :** Mengubah warna latar belakang tombol menjadi sedikit lebih gelap ketika kursor diarahkan ke atasnya, memberikan umpan balik visual yang baik.

**.control-panel :** Mengatur agar tombol-tombol di panel kontrol berada di tengah.

#### footer

- Teks diatur rata tengah.
- Diberi jarak atas (margin-top) dan latar belakang menggunakan warna sekunder (--color-secondary).

### 3.Script.js

```

import { document } from "https://www.w3schools.com/js/DocumentObjectModel/API.js";
document.addEventListener("DOMContentLoaded", () => {
  const pencemaranData = [
    {
      judul: "Pencemaran Air oleh Sampah Plastik",
      deskripsi: "Sampah plastik di sungai atau laut merusak ekosistem akuatis. Sampah ini tidak mudah terurai dan dapat membahayakan kehidupan laut yang menganggapnya sebagai makanan.",
      gambarSrc: "assets/pencemaran_plastik.jpeg"
    },
    {
      judul: "Pencemaran Udara dari Asap Pabrik",
      deskripsi: "Asap tebal yang keluar dari cerobong pabrik melepaskan zat berbahaya (seperti sulfur dioksida) ke atmosfer, menyebabkan kabut asap (smog) dan hujan asam.",
      gambarSrc: "assets/udara_asap_pabrik.jpeg"
    },
    {
      judul: "Pencemaran Tanah dari Limbah Industri",
      deskripsi: "Pembuangan limbah beracun dan bahan kimia berbahaya (misalnya logam berat) secara sembarangan ke tanah dapat menjadikannya tidak subur dan mengontaminasi sumber air tanah.",
      gambarSrc: "assets/tanah_limbah_industri.jpeg"
    },
    {
      judul: "Pencemaran Suara di Perkotaan",
      deskripsi: "Tingkat kebisingan yang tinggi dari lalu lintas, pesawat, atau lokasi konstruksi dapat mengganggu kesehatan manusia dan hewan, menyebabkan stres, dan masalah pendengaran.",
      gambarSrc: "assets/suara_di_perkotaan.jpeg"
    },
    {
      judul: "Pencemaran Air oleh Tumpahan Minyak",
      deskripsi: "Kebocoran atau tumpahan minyak di perairan membentuk lapisan di permukaan air, menghalangi oksigen dan cahaya, serta merusak bulu burung laut dan insang ikan.",
      gambarSrc: "assets/air_tumpahan_minyak.jpeg"
    },
    {
      judul: "Pencemaran Udara dari Pembakaran Sampah Terbuka",
      deskripsi: "Praktik membakar sampah di tempat terbuka menghasilkan dioksin dan furan, yang sangat beracun dan berkontribusi terhadap pencemaran udara lokal.",
      gambarSrc: "assets/udara_sampah.jpeg"
    },
    {
      judul: "Pencemaran Tanah dari Penggunaan Pestisida Berlebihan",
      deskripsi: "Penggunaan pestisida dan pupuk kimia yang berlebihan dalam pertanian dapat meresap ke dalam tanah, merusak mikroorganisme yang bermanfaat, dan mencemari hasil panen.",
      gambarSrc: "assets/tanah_pestisida.jpeg"
    }
  ];
});

```

Bagian ini memastikan semua elemen HTML yang dibutuhkan sudah dimuat dan siap diprogram.

Semua kode ini berada di dalam event listener **DOMContentLoaded** yang berarti kode baru akan berjalan setelah seluruh dokumen HTML selesai dimuat.

Ini adalah database sederhana Anda yang disimpan dalam bentuk Array JavaScript bernama pencemaranData.

- Array ini berisi 7 objek yang mewakili setiap jenis pencemaran.
- Setiap objek memiliki tiga properti utama:
  - judul: Judul jenis pencemaran (misalnya, "Pencemaran Plastik").
  - deskripsi: Penjelasan detail mengenai jenis pencemaran tersebut.
  - gambarSrc: Jalur (path) ke file gambar/video yang berkaitan, digunakan untuk menampilkan visual.

**Tujuan:** Menyimpan semua informasi konten secara terstruktur di satu tempat, terpisah dari HTML.

```

40   const listContainer = document.getElementById('pencemaran-list');
41   const toggleModeBtn = document.getElementById('toggleModeBtn');
42   const tampilanGambarBtn = document.getElementById('tampilanGambarBtn');
43   const halamanGambar = document.getElementById('halamanGambar');
44   const gambarContainer = document.getElementById('gambarContainer');
45

```

**Variabel-variabel konstan (const) dibuat untuk menyimpan referensi ke elemen HTML yang memiliki ID unik (seperti yang Anda buat di HTML):**

- listContainer: Wadah untuk daftar pencemaran (pencemaran-list).
- toggleModeBtn: Tombol untuk mengganti tema.
- tampilanGambarBtn: Tombol untuk menampilkan galeri gambar.
- halamanGambar: Wadah galeri gambar itu sendiri.
- gambarContainer: Wadah tempat kartu gambar akan dimuat

```

48 pencemaranData.forEach((data, index) => {
49   // Buat elemen Judul
50   const judulDiv = document.createElement('div');
51   judulDiv.className = 'judul-pencemaran';
52   judulDiv.innerHTML = `<h3>${index + 1}. ${data.judul}</h3>`;
53   judulDiv.dataset.index = index; // Menyimpan index untuk referensi
54
55   // Buat elemen Penjelasan (Awalnya tersembunyi)
56   const penjelasanParagraf = document.createElement('p');
57   penjelasanParagraf.className = 'penjelasan-paragraf tersembunyi';
58   penjelasanParagraf.textContent = data.deskripsi;
59
60   // Tambahkan ke container
61   listContainer.appendChild(judulDiv);
62   listContainer.appendChild(penjelasanParagraf);
63 });
64

```

Kode ini secara otomatis mengisi bagian daftar (pencemaran-list) berdasarkan data dari pencemaranData:

1. Melakukan looping (forEach) pada setiap data pencemaran.

2. Untuk setiap data, dibuat dua elemen:

- Judul (judulDiv): Sebuah <div> dengan kelas judul-pencemaran dan diberi ID unik yang berisi judul pencemaran.
- Penjelasan (penjelasanParagraf): Sebuah <p> yang berisi deskripsi. Penting, paragraf ini diberi kelas tersembunyi (sesuai CSS Anda), sehingga tidak terlihat secara default.

3. Kedua elemen ini kemudian dimasukkan ke dalam listContainer di HTML.

**Tujuan: Membangun daftar konten dan menyembunyikan penjelasannya.**

```

65 // Menambahkan Event Listener untuk Interaksi (Hiding/Showing)
66 listContainer.addEventListener('click', (event) => {
67   let target = event.target;
68   // Pastikan kita mengklik div judul-pencemaran (atau elemen di dalamnya)
69   while (target && !target.classList.contains('judul-pencemaran')) {
70     target = target.parentElement;
71   }
72
73   if (target && target.classList.contains('judul-pencemaran')) {
74     const index = target.dataset.index;
75     const penjelasan = listContainer.children[parseInt(index) * 2 + 1];
76
77     // Cek apakah paragraf sedang ditampilkan
78     if (penjelasan.classList.contains('tersembunyi')) {
79       // Tampilkan pertanyaan konfirmasi sebelum menampilkan
80       const tampilan = confirm(`Apakah Anda ingin menampilkan penjelasan "${target.querySelector('h3').textContent}" ini?`);
81       if (tampilan) {
82         penjelasan.classList.remove('tersembunyi');
83       }
84     } else {
85       // Tampilkan pertanyaan konfirmasi sebelum menyembunyikan
86       const sembunyikan = confirm(`Apakah Anda ingin menyembunyikan penjelasan "${target.querySelector('h3').textContent}" ini?`);
87       if (semبunyikan) {
88         penjelasan.classList.add('tersembunyi');
89       }
90     }
91   }
92 });
93

```

Ini adalah event listener yang bekerja pada seluruh listContainer (daftar judul).

1. Ketika pengguna mengklik di area daftar: Kode akan mencari elemen terdekat dengan kelas judul-pencemaran.
2. Jika elemen yang diklik adalah judul:

- Kode menentukan elemen penjelasan mana yang terkait dengan judul tersebut.

- Cek Kondisi: Jika penjelasan sedang tersembunyi (contains('tersembunyi')), sistem akan meminta konfirmasi (confirm) apakah pengguna ingin menampilkan penjelasan tersebut. Jika pengguna setuju, kelas tersembunyi dihapus (remove), sehingga penjelasan muncul.
- Jika penjelasan sedang ditampilkan, sistem akan meminta konfirmasi apakah pengguna ingin menyembunyikannya. Jika setuju, kelas tersembunyi ditambahkan (add), sehingga penjelasan kembali hilang.

**Tujuan:** Membuat mekanisme interaktif (toggle) untuk membaca deskripsi pencemaran satu per satu.

```

94 // --- Mengubah Mode Tampilan ---
95 toggleModeBtn.addEventListener('click', () => {
96   const body = document.body;
97   const isRapi = body.classList.contains('mode-rapi');
98
99   if (isRapi) {
00     body.classList.remove('mode-rapi');
01     body.classList.add('mode-berantakan');
02     toggleModeBtn.textContent = 'Ubah Mode Tampilan (Saat Ini: Berantakan)';
03   } else {
04     body.classList.remove('mode-berantakan');
05     body.classList.add('mode-rapi');
06     toggleModeBtn.textContent = 'Ubah Mode Tampilan (Saat Ini: Rapi)';
07   }
08 });

```

Ini adalah *event listener* untuk tombol `toggleModeBtn`:

1. Ketika tombol diklik: Kode memeriksa apakah `<body>` saat ini memiliki kelas `mode-rapi`.
2. Jika Mode Rapi Aktif (`if (isRapi)`):
  - Hapus kelas `mode-rapi`.
  - Tambahkan kelas `mode-berantakan`.
  - Ubah teks tombol menjadi "Ubah Mode Tampilan (Saat Ini: Berantakan)". (Ini akan memicu perubahan warna yang diatur di CSS).
3. Jika Mode Berantakan Aktif (`else`):
  - Hapus kelas `mode-berantakan`.
  - Tambahkan kembali kelas `mode-rapi`.
  - Ubah teks tombol menjadi "Ubah Mode Tampilan (Saat Ini: Rapi)".

**Tujuan:** Mengganti tema visual halaman secara instan.

```

.09
.10 // --- Menampilkan Gambar ---
.11 tampilkanGambarBtn.addEventListener('click', () => {
.12
.13   if (halamanGambar.classList.contains('tersembunyi')) {
.14     halamanGambar.classList.remove('tersembunyi');
.15     tampilkanGambarBtn.textContent = 'Sembunyikan Gambar';
.16
.17
.18   if (gambarContainer.children.length === 0) {
.19     pencemaranData.forEach(data => {
.20       const card = document.createElement('div');
.21       card.className = 'pencemaran-card';
.22
.23       card.innerHTML =
.24         `
.25         <p>${data.judul}</p>
.26       `;
.27       gambarContainer.appendChild(card);
.28     });
.29   }
.30 } else {
.31   halamanGambar.classList.add('tersembunyi');
.32   tampilkanGambarBtn.textContent = 'Tampilkan Gambar 7 Pencemaran';
.33
.34 }
.35 });

```

Ini adalah event listener untuk tombol **tampilkanGambarBtn**:

1. **Ketika tombol diklik:** Kode memeriksa apakah wadah galeri (halamanGambar) sedang tersembunyi.
2. **Jika Tersembunyi:**
  - Hapus kelas tersembunyi dari halamanGambar (membuat galeri muncul).
  - Ubah teks tombol menjadi "Sembunyikan Gambar".
  - Penting: Cek apakah galeri sudah berisi kartu (gambarContainer.children.length === 0).
    - Jika belum terisi, lakukan *looping* pada pencemaranData, buat elemen <div> baru (card) untuk setiap data, isi dengan tag <img> dan <p> (menggunakan data.gambarSrc dan data.judul), lalu masukkan (appendChild) kartu tersebut ke dalam galeri.
3. **Jika Sedang Ditampilkan (else):**
  - Tambahkan kembali kelas tersembunyi ke halamanGambar (membuat galeri hilang).
  - Ubah teks tombol kembali menjadi "Tampilkan Gambar 7 Pencemaran".

**Tujuan: Membuat tombol yang berfungsi sebagai toggle untuk menampilkan/menyembunyikan galeri, dan memastikan galeri hanya dibangun sekali (tidak dibuat ulang setiap kali tombol diklik).**