**LAPORAN PRAKTIKUM**

**PEMROGRAMAN WEB**



Disusun Oleh :

Martonius Epta Sidabutar

NIM : 202312051

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI TEKNOLOGI BONTANG**

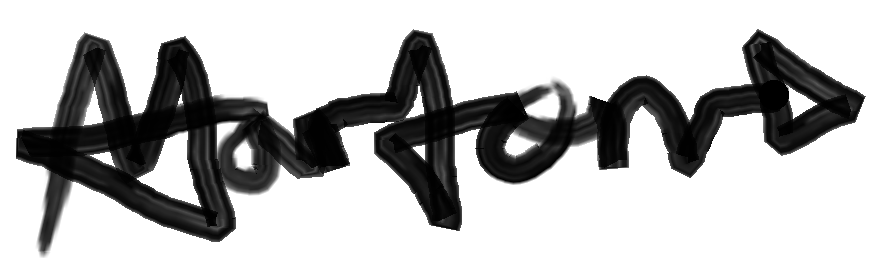
**TAHUN 2025**

# KATA PENGANTAR

Assalamu’alaikum Warahmatullahi Wabarakatuh. Puji dan syukur penulis panjatkan kehadirat Allah SWT, yang telah melimpahkan rahmat dan Hidayah-Nya lah sehingga laporan praktikum sistem basis data ini dapat terselesaikan, tetapi masih sangat jauh dari kesempurnaan.

Penulisan laporan praktikum ini dimaksudkan sebagai tolak ukur Saya dalam mencapai pemahaman beserta pembahasan mengenai Program dari praktikum Pemrograman Dasar kali ini, disamping itu penulis berharap dalam penulisan laporan praktikum ini dapat bermanfaat bagi penulis, pembaca, khususnya mahasiswa Sekolah Tinggi Teknologi Bontang dan masyarakat umumnya.

Bontang, 6 Juni 2025



Martonius Epta Sidabutar

# DAFTAR ISI

Daftar Isi

[KATA PENGANTAR ii](#_Toc203755905)

[DAFTAR ISI iii](#_Toc203755906)

[LEMBAR ASISTENSI vii](#_Toc203755907)

[Modul 1. Pemrograman WEB HTML 1](#_Toc203755908)

[A. Tujuan 1](#_Toc203755909)

[B. Alat dan Bahan 1](#_Toc203755910)

[C. Landasan Teori 1](#_Toc203755911)

[D. Kegiatan Praktikum 3](#_Toc203755912)

[Modul 2. Pengenalan CSS (Cascading Style Sheet) 29](#_Toc203755913)

[A. Tujuan 29](#_Toc203755914)

[B. Teori Dasar CSS 29](#_Toc203755915)

[C. Pratikum 29](#_Toc203755916)

[D. Tugas Praktikum 49](#_Toc203755917)

[Modul 3. Pengenalan CSS (Cascading Style Sheet) 55](#_Toc203755918)

[A. Tujuan 55](#_Toc203755919)

[B. Teori Dasar Bootsrap 55](#_Toc203755920)

[C. Pratikum 55](#_Toc203755921)

[D. Tugas Praktikum 86](#_Toc203755922)

[Modul 4. Javascript 95](#_Toc203755923)

[A. Tujuan 95](#_Toc203755924)

[B. Teori Dasar Bootsrap 95](#_Toc203755925)

[C. Pratikum 95](#_Toc203755926)

[D. Tugas Akhir Praktikum: Mini E-Comerce Product Page 128](#_Toc203755927)

[Modul 5. Dasar-dasar PHP 132](#_Toc203755928)

[A. Tujuan 132](#_Toc203755929)

[B. Teori Dasar PHP 132](#_Toc203755930)

[C. Praktikum 132](#_Toc203755931)

[**Analisis Singkat Kode PHP** 145](#_Toc203755932)

[**Kelebihan Kode:** 145](#_Toc203755933)

[**Kekurangan:** 146](#_Toc203755934)

[**Kesimpulan:** 146](#_Toc203755935)

[D. Tugas Praktikum 160](#_Toc203755936)

[DAFTAR PUSTAKA 166](#_Toc203755937)

# LEMBAR ASISTENSI

**Modul 1. Pemrograman WEB HTML**

## A. Tujuan

Setelah mengikuti praktikum, Mahasiswa diharapkan mampu:

1. Setelah mengikuti praktikum ini mahasiswa mampu memahami struktur dokumen html
2. Mahasiswa mampu membuat web sederhana dengan menggunakan HTML.

## B. Alat dan Bahan

Peralatan dan bahan yang dibutuhkan untuk praktikum ini adalah sebagai berikut:

1. Hardware (Laptop HP)
2. Software:
3. Notepad++
4. Web Brower (Google Chrome)

## C. Landasan Teori

HTML (HyperText Markup Language) adalah bahasa markup standar yang digunakan untuk membuat dan menyusun halaman web di internet. HTML bukan bahasa pemrograman melainkan bahasa markup yang menggunakan tag-tag untuk memberi struktur dan makna pada konten web.

**Mengapa HTML Bukan Bahasa Pemrograman?**

1. HTML tidak memiliki logika pemrograman seperti kondisi (if-else), perulangan (for, while), atau fungsi.
2. HTML hanya digunakan untuk menyusun dan menampilkan konten pada halaman web.

**Apa Peran HTML dalam Pengembangan Web?**

1. HTML mendefinisikan struktur halaman web seperti teks, gambar, tabel, dan formulir.
2. Digunakan bersama dengan CSS (untuk styling) dan JavaScript (untuk interaktivitas).

**Sejarah HTML**

1. Awal Mula HTML (1989–1995)
   1. 1989 – Tim Berners-Lee seorang fisikawan di CERN yang mengusulkan sistem berbasis internet untuk berbagi dokumen.
   2. 1990 – Berners-Lee menciptakan prototipe pertama HTML dan browser WorldWideWeb di komputer NeXT.
   3. 1991 – HTML pertama kali diperkenalkan ke public yang memungkinkan penggunaan hyperlinked text.
   4. 1993 – Mosaic Browser memperkenalkan tag <img>, merevolusi tampilan visual web.
   5. 1994 – W3C (World Wide Web Consortium) didirikan untuk standarisasi teknologi web.
   6. 1995 – HTML 2.0 dirilis sebagai standar resmi pertama.
2. Perkembangan HTML dan Persaingan Browser (1995–2000)
   1. 1995 – Internet Explorer pertama kali diperkenalkan yang memicu persaingan dengan Netscape.
   2. 1997 – HTML 3.2 dirilis dengan menambahkan dukungan untuk tabel, Apple Java, dan kontrol teks.
   3. 1999 – HTML 4.01 menjadi standar utama yang mengintegrasikan CSS, fitur aksesibilitas, dan internasionalisasi.
3. Era XML dan XHTML (2000–2006)
   1. 2000 – XHTML 1.0 diperkenalkan sebagai versi HTML yang lebih ketat dan berbasis XML.
   2. 2004 – Kelompok WHATWG dibentuk untuk mengembangkan HTML5 yang menentang keterbatasan XHTML.
4. Revolusi HTML5 (2008–Sekarang)
   1. 2008 – Draft pertama HTML5 dirilisyang menekankan pada multimedia, semantik, dan API.
   2. 2014 – HTML5 menjadi rekomendasi resmi W3C dengan membawa fitur seperti <video> dan <audio>.
   3. 2016–2017 – HTML5.1 dan HTML5.2 memperkenalkan peningkatan seperti penyimpanan offline dan aksesibilitas.
   4. 2019 – W3C dan WHATWG menjadikan HTML sebagai "living standard" yang terus diperbarui.
5. Fitur Modern HTML
   1. Elemen Semantik seperti <header>, <article>, <nav> untuk SEO dan aksesibilitas.
   2. Dukungan Multimedia, <video> dan <audio> tanpa perlu plugin tambahan.
   3. API Canggih yaitu Geolocation, drag-and-drop, dan Web Workers untuk aplikasi dinamis.
   4. Desain Mobile-First seperti gambar rsponsif dan meta viewport untuk tampilan optimal di perangkat mobile.

**Struktur Dasar HTML**

Dokumen HTML terdiri dari elemen-elemen yang ditandai dengan tag, yang biasanya terdiri dari tag pembuka dan penutup. Struktur dasar dokumen HTML meliputi:

1. <!DOCTYPE html> : Mendefinisikan tipe dokumen sebagai HTML5.
2. <html> : Elemen root yang membungkus seluruh isi halaman.
3. <head> : Bagian yang berisi metadata, judul halaman, dan link ke sumber daya eksternal.
4. <body> : Bagian yang berisi konten utama yang akan ditampilkan di browser.

**Manfaat HTML**

1. Membuat struktur halaman web yang terorganisir.
2. Memungkinkan penyisipan teks, gambar, video, dan elemen multimedia lainnya.
3. Dasar dari semua halaman web yang dapat diakses melalui browser.
4. Memudahkan pengembangan web dengan berbagai teknologi pendukung seperti CSS dan JavaScript.

## D. Kegiatan Praktikum

Kegiatan praktikum Pemrograman Web

1. **Pengenalan HTML dan Struktur dasar Dokumen**

**Penjelasan**

HTML adalah bahasa markup yang menggunakan tag untuk membuat elemen elemen pada halaman web. Setiap elemen memiliki fungsi tertentu seperti heading, paragraf, gambar, dan lain-lain

**Strukur Dasar HTML**

****

**Latihan Pratikum**

* Buatlah halaman HTML sederahana dengan judul “Halaman Pertama Saya”.
* Tambahkan Heading <h1> dan paragraph <p> sesuai contoh diatas.
* Berikan *capture* (*screenshoot/*tangkapan layer) dan analisis dari hasil pratikum anda di laporan.

Jawab

Berikut HTMLnya:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Modul pratikum pertama</title>  </head>  <body>  <h1>Halo, Saya Martonius Epta Sidabutar!</h1>  <h2>Ini Heading h2</h2>  <h3>Ini Heading h3</h3>  <p>Ini adalah paragraf pertama Saya.</p>  </body>  </html> |

Berikut hasil html dibuka dengan web browser

|  |
| --- |
|  |

**Analisis: Halaman ini sederhana dan mudah dipahami, cocok untuk pemula yang baru belajar HTML.**

1. **Elemen HTML Dasar**

**Penjelasan**

Elemen dasar HTML meliputi heading (<h1> sampai <h6>), paragraf (<p>), link (<a>), gambar (<img>), dan daftar (<ul>, <ol>, <li>).

**Contoh Kode**

****

**Latihan Pratikum**

* Buat halaman dengan heading, paragraph yang mengandung link, gambar dengan atribut alt, dan daftar tidak berurutan.
* Berikan *capture* (*screenshoot*/tangkapan layer dan analisis dari hasil pratikum anda di laporan.

Jawab:

Kode HTML:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Halaman Praktikum HTML</title>  </head>  <body>  <h2>Heading Level 2</h2>  <p>Ini adalah paragraf dengan <a href="https://www.stitek.ac.id">link ke website stitek</a></p>  <img src="D:\Marton\Kuliah S1 Teknik Informatika\Logo\LOGO\_STITEK2.PNG" alt="Deskripsi gambar">    <ul>  <li>Item pertama</li>  <li>Item kedua</li>  </ul>  </body>  </html> |

Berikut tampilannya:

|  |
| --- |
|  |

Berikut jika link dibuka:

|  |
| --- |
|  |

**Analisis**: Laporan dapat mencakup hal-hal berikut:

* Heading, paragraf, gambar, dan daftar tampil dengan baik sesuai fungsinya.
* Link dapat diklik dan mengarah ke situs yang benar.
* Gambar ditampilkan dengan deskripsi alternatif yang berguna.

1. **ATRIBUT HTML**

**Penjelasan**

Atribut memberikan informasi tambahan pada elemen HTML, seperti href pada tak <a>, src dan alt pada <img>, serta title untuk tooltip

**Contoh code**

****

**Latihan Pratikum**

* Buat paragraf dengan berbagi format teks menggunakan tak di atas.
* Berikan *capture* (*screenshoot*/tangkapan layar) dan analisis dari hasil praktikum anda di laporan.

Jawab:

Berikut kode HTML:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Halaman Praktikum HTML dengan Atribut Title</title>  </head>  <body>  <h2>Heading Level 2</h2>  <p>Ini adalah paragraf dengan <a href="https://www.stitek.ac.id" title="Pergi ke stitek">link ke website stitek</a></p>  <img src="gambar.jpg" alt="Deskripsi gambar" title="Gambar ini menunjukkan deskripsi gambar">    <ul>  <li>Item pertama</li>  <li>Item kedua</li>  </ul>  </body>  </html> |

Berikut jika kode HTML dijalankan di web browser:

|  |
| --- |
|  |
|  |

**Analisis**: Laporan dapat mencakup hal-hal berikut:

* Atribut title berhasil ditambahkan pada link dan gambar.
* Tooltip yang muncul saat kursor diarahkan ke elemen (link atau gambar) sudah berfungsi dengan baik.

1. **Format Teks**

**Penjelasan**

Tag untuk memformat teks seperti <b> (bold), <i> (italic), <u> (underline), <strong>,

dan <em>.

**Contoh Kode**



**Latihan Praktikum**

* Buat paragraf dengan berbagai format teks menggunakan tak di atas.
* Berikan *capture* (*screenshoot*/tangkapan layar) dan analisis dari hasil praktikum anda di laporan.

Berikut kode HTML:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Halaman Praktikum Format Teks</title>  </head>  <body>  <h2>Format Teks dalam HTML</h2>  <p>Ini adalah teks <b>tebal</b>, <i>miring</i>, dan <u>garis bawah</u>.</p>  <p><strong>Penting!</strong> Ini adalah teks yang ditekankan pentingnya.</p>  <p>Ini adalah teks yang <em>ditonjolkan</em> dengan tag <em>.</p>  </body>  </html> |

Berikut hasil kode HTML dijalankan di web browser:

|  |
| --- |
|  |

**Analisis**: Laporan dapat mencakup hal-hal berikut:

* Semua format teks (tebal, miring, garis bawah, penting, dan tonjolan) diterapkan dengan benar.
* Teks tampil sesuai dengan format yang diinginkan.

1. **Link dan Gambar**

**Penjelsan**

Link dibuat dengan tak <a> dan gambar dengan tak <img>. Atribut penting adalah href untuk link dan src serta alt untuk gambar

**Contoh Kode**



**Latihan Pratikum**

* Buat halaman dengan beberapa link dan gambar yang berfungsi.
* Berikan *capture* (*screenshoot*/tangkapan layar) dan analisis dari hasil praktikum anda di laporan.

**Jawab:**

Berikut code HTML:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Halaman Link dan Gambar</title>  </head>  <body>  <h2>Halaman dengan Link dan Gambar</h2>  <!-- Link ke Kampus STITEK -->  <p>Ini adalah link ke <a href="https://www.stitek.ac.id" target="\_blank">Kampus Stitek</a>.</p>  <!-- Gambar dengan deskripsi -->  <p><img src="D:\Marton\Kuliah S1 Teknik Informatika\Logo\LOGO\_STITEK2.PNG" alt="Foto Stitek" width="300"></p>    <!-- Link ke Gambar -->  <p>Ini adalah link yang akan membuka gambar: <a href="D:\Marton\Kuliah S1 Teknik Informatika\Logo\LOGO\_STITEK2.PNG" target="\_blank">Lihat Foto Stitek</a>.</p>  </body>  </html> |

**Berikut Kode HTML dijalankan di web browser:**

|  |
| --- |
|  |
| **Ketika klik link “Kampus Stitek”** |
| **Ketika klik link “Lihat Foto Stitek:** |

**Analisis**: Laporan dapat mencakup hal-hal berikut:

* Semua *link* berfungsi dengan baik, membuka halaman yang benar atau gambar.
* Gambar ditampilkan dengan benar dengan deskripsi alternatif yang muncul jika gambar tidak dapat dimuat.

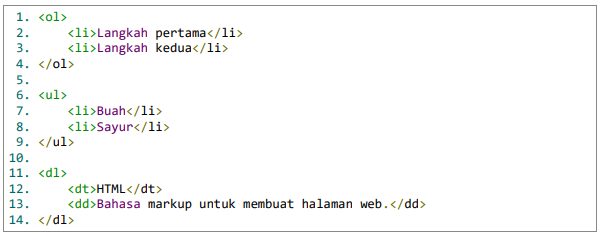
1. **Daftar (List)**

**Penejelasan**

Daftar berurutan menggunakan <ol>, daftar tidak berurutan menggunakan <ul>, dan

daftar definisi menggunakan <dl>, <dt>, dan <dd>.

**Contoh Kode**

****

**Latihan Pratikum**

* Buat halaman dengan ketiga jenis daftar tersebut
* Berikan *capture* (*screenshoot*/tangkapan layar) dan analisis dari hasil praktikum anda di laporan.

**Jawab:**

**Berikut kode HTML:**

|  |
| --- |
| * <!DOCTYPE html> * <html> * <head> * <title>Halaman Daftar Berurutan, Tidak Berurutan, dan Definisi</title> * </head> * <body> * <h2>Daftar Berurutan, Tidak Berurutan, dan Definisi</h2> * <!-- Daftar Berurutan --> * <h3>Daftar Berurutan (Ordered List)</h3> * <ol> * <li>Langkah pertama</li> * <li>Langkah kedua</li> * <li>Langkah ketiga</li> * </ol> * <!-- Daftar Tidak Berurutan --> * <h3>Daftar Tidak Berurutan (Unordered List)</h3> * <ul> * <li>Buah</li> * <li>Sayur</li> * <li>Rempah</li> * </ul> * <!-- Daftar Definisi --> * <h3>Daftar Definisi (Definition List)</h3> * <dl> * <dt>HTML</dt> * <dd>Bahasa markup untuk membuat halaman web.</dd> * <dt>CSS</dt> * <dd>Bahasa untuk menggaya elemen HTML.</dd> * <dt>JavaScript</dt> * <dd>Bahasa pemrograman untuk membuat halaman web interaktif.</dd> * </dl> * </body> * </html> |

**Berikut Kode HTML dibuka di web browser:**

|  |
| --- |
|  |

**Analisis: Laporan dapat mencakup hal-hal berikut:**

* Semua jenis daftar (berurutan, tidak berurutan, dan definisi) diterapkan dengan benar.
* Daftar berurutan menampilkan nomor secara otomatis.
* Daftar tidak berurutan menggunakan tanda bullet.
* Daftar definisi menampilkan istilah dan deskripsi dengan format yang tepat.

1. **Tabel HTML**

**Penejelasan**

Tabel dibuat dengan tag <table>, baris dengan <tr>, header dengan <th>, dan data

dengan <td>.

**Contoh Kode**

**** ****

**Latihan Pratikum**

* Buat tabel data sederhana dengan minimal 3 baris dan 2 kolom.
* Berikan *capture* (*screenshoot*/tangkapan layar) dan analisis dari hasil praktikum anda di laporan.

Jawab:

Berikut Kode HTML:

|  |
| --- |
| * <!DOCTYPE html> * <html> * <head> * <title>Tabel Data Sederhana</title> * </head> * <body> * <h2>Tabel Data Sederhana</h2> * <table border="1"> * <tr> * <th>Nama</th> * <th>Umur</th> * </tr> * <tr> * <td>Abadi</td> * <td>20</td> * </tr> * <tr> * <td>Nugroho</td> * <td>22</td> * </tr> * <tr> * <td>Rina</td> * <td>25</td> * </tr> * </table> * </body> * </html> |

Berikut Kode HTML dijalankan di Web Browser

|  |
| --- |
|  |

**Analisis**: Laporan dapat mencakup hal-hal berikut:

* Tabel berhasil dibuat dengan benar, menampilkan header "Nama" dan "Umur".
* Data ditampilkan dalam tiga baris dengan dua kolom.
* Garis batas terlihat di sekitar tabel dan sel-selnya.

1. **Formulir HTML**

**Penjelasan**

Formulir digunakan untuk mengumpulkan data dari pengguna. Elemen penting : <form>,

<input>, <textarea>, <select>, <button>.

**Contoh Kode**

****

**Latihan Pratikum**

* Buat Formulir pendaftaran dengan input teks, email, dan tombol submit.
* Berikan *capture* (*screenshoot*/tangkapan layar) dan analisis dari hasil praktikum anda di laporan.

**Jawab:**

**Berikut kode HTML:**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Formulir Pendaftaran</title>  </head>  <body>  <h2>Formulir Pendaftaran</h2>  <form action="/submit" method="post">  <label for="nama">Nama:</label>  <input type="text" id="nama" name="nama" required><br><br>  <label for="email">Email:</label>  <input type="email" id="email" name="email" required><br><br>  <label for="alamat">Alamat:</label>  <textarea id="alamat" name="alamat" rows="4" cols="50" required></textarea><br><br>  <input type="submit" value="Kirim">  </form>  </body>  </html> |

**Berikut Kode HTML dijalankan di web Browser:**

|  |
| --- |
|  |

**Analisa:**

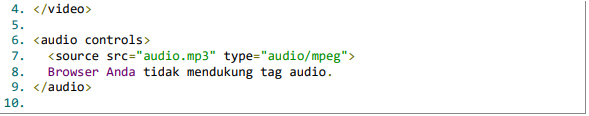
* Formulir berhasil dibuat dengan input teks untuk nama, input email untuk email, dan area teks untuk alamat.
* Formulir dapat disubmit dengan tombol "Kirim" yang mengirimkan data formulir (meskipun pada aplikasi nyata, URL /submit harus dikonfigurasi di server).

1. **Multimedia**

**Penjelasan**

HTML5 mendukung penyisipan video dan audio dengan tag <video> dan <audio>.

**Contoh Kode**

****

**Latihan Pratikum**

* Buat halaman yang menampilkan video dan audio yang dapat diputar.
* Berikan *capture* (*screenshoot*/tangkapan layar) dan analisis dari hasil praktikum anda di laporan.

**Jawab:**

**Kode HTML:**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Halaman Video dan Audio</title>  </head>  <body>  <h2>Video dan Audio yang Dapat Diputar</h2>  <!-- Video -->  <h3>Video</h3>  <video width="640" height="360" controls>  <source src="D:\Captures\sukses.mp4" type="video/mp4">  Browser Anda tidak mendukung tag video.  </video>  <br><br>  <!-- Audio -->  <h3>Audio</h3>  <audio controls>  <source src="C:\Users\Asus\Music\Suara 007.m4a"" type="audio/mpeg">  Browser Anda tidak mendukung tag audio.  </audio>  </body>  </html> |

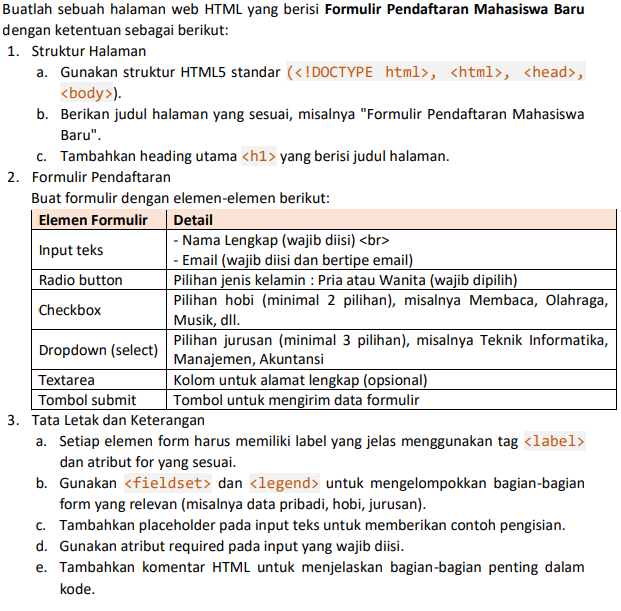
**Kode HTML dibuka di web browser:**

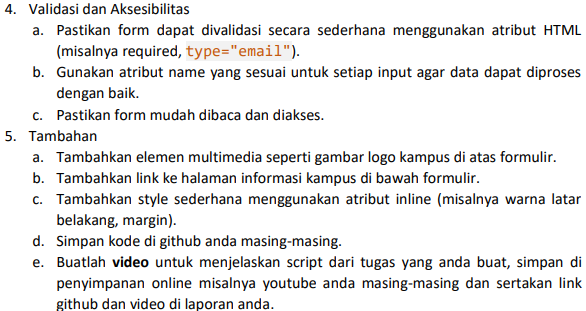
|  |
| --- |
|  |

**Analisa:**

* Video dan audio ditampilkan dengan benar.
* Kontrol video dan audio berfungsi sebagaimana mestinya.

C. **Tugas Pratikum**





Jawab:

Berikut adalah contoh halaman HTML untuk **Formulir Pendaftaran Mahasiswa Baru** dengan memenuhi ketentuan yang diberikan. Saya juga akan memberikan penjelasan untuk setiap bagian penting dalam kode.

|  |
| --- |
| <!DOCTYPE html>  <html lang="id">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Formulir Pendaftaran Mahasiswa Baru</title>  </head>  <body style="font-family: Arial, sans-serif; background-color: #f0f0f0; margin: 0; padding: 20px;">  <!-- Gambar Logo Kampus -->  <img src="logo\_kampus.png" alt="Logo Kampus" style="display: block; margin: 0 auto 20px; width: 200px;">  <h1 style="text-align: center; color: #2c3e50;">Formulir Pendaftaran Mahasiswa Baru</h1>    <!-- Formulir Pendaftaran -->  <form action="/submit" method="post" style="background-color: white; padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);">    <!-- Bagian Data Pribadi -->  <fieldset style="border: 1px solid #ddd; padding: 10px; margin-bottom: 20px;">  <legend style="font-weight: bold;">Data Pribadi</legend>  <!-- Nama Lengkap -->  <label for="nama" style="display: block; margin-bottom: 5px;">Nama Lengkap:</label>  <input type="text" id="nama" name="nama" placeholder="Masukkan Nama Lengkap" required style="width: 100%; padding: 8px; margin-bottom: 10px;">    <!-- Email -->  <label for="email" style="display: block; margin-bottom: 5px;">Email:</label>  <input type="email" id="email" name="email" placeholder="Masukkan Email" required style="width: 100%; padding: 8px; margin-bottom: 10px;">    <!-- Jenis Kelamin -->  <label>Jenis Kelamin:</label><br>  <input type="radio" id="pria" name="jenis\_kelamin" value="Pria" required>  <label for="pria">Pria</label>  <input type="radio" id="wanita" name="jenis\_kelamin" value="Wanita" required>  <label for="wanita">Wanita</label>  </fieldset>  <!-- Bagian Hobi -->  <fieldset style="border: 1px solid #ddd; padding: 10px; margin-bottom: 20px;">  <legend style="font-weight: bold;">Hobi</legend>  <label for="hobi" style="display: block; margin-bottom: 5px;">Pilih Hobi (minimal 2):</label>  <input type="checkbox" id="membaca" name="hobi" value="Membaca">  <label for="membaca">Membaca</label><br>  <input type="checkbox" id="olahraga" name="hobi" value="Olahraga">  <label for="olahraga">Olahraga</label><br>  <input type="checkbox" id="musik" name="hobi" value="Musik">  <label for="musik">Musik</label><br>  <input type="checkbox" id="melukis" name="hobi" value="Melukis">  <label for="melukis">Melukis</label>  </fieldset>  <!-- Bagian Jurusan -->  <fieldset style="border: 1px solid #ddd; padding: 10px; margin-bottom: 20px;">  <legend style="font-weight: bold;">Pilihan Jurusan</legend>  <label for="jurusan" style="display: block; margin-bottom: 5px;">Pilih Jurusan:</label>  <select id="jurusan" name="jurusan" required style="width: 100%; padding: 8px; margin-bottom: 10px;">  <option value="Teknik Informatika">Teknik Informatika</option>  <option value="Manajemen">Manajemen</option>  <option value="Akuntansi">Akuntansi</option>  </select>  </fieldset>  <!-- Bagian Alamat -->  <fieldset style="border: 1px solid #ddd; padding: 10px; margin-bottom: 20px;">  <legend style="font-weight: bold;">Alamat</legend>  <label for="alamat" style="display: block; margin-bottom: 5px;">Alamat Lengkap:</label>  <textarea id="alamat" name="alamat" placeholder="Masukkan alamat lengkap" style="width: 100%; padding: 8px; height: 100px;"></textarea>  </fieldset>  <!-- Tombol Submit -->  <button type="submit" style="background-color: #4CAF50; color: white; padding: 10px 20px; border: none; border-radius: 5px; cursor: pointer;">  Kirim Pendaftaran  </button>  </form>  <!-- Link ke Halaman Informasi Kampus -->  <p style="text-align: center; margin-top: 20px;">  <a href="https://www.kampus.com/informasi" style="color: #3498db;">Klik di sini untuk informasi lebih lanjut tentang kampus</a>  </p>  </body>  </html> |

**Hasil Kode HTML dijalankan di Web Browser**

|  |
| --- |
|  |

**Penjelasan Bagian-bagian Penting:**

1. **Struktur Halaman**:
   * Menggunakan struktur HTML5 standar dengan elemen seperti <html>, <head>, dan <body>.
   * Halaman diberi judul "Formulir Pendaftaran Mahasiswa Baru" dan heading utama <h1> di bagian atas.
2. **Formulir Pendaftaran**:
   * Formulir dibagi menjadi beberapa bagian menggunakan <fieldset> dan <legend> untuk kejelasan.
   * Setiap elemen input seperti teks, radio button, checkbox, dan dropdown disertai label yang sesuai untuk aksesibilitas.
   * Elemen required digunakan untuk memastikan input yang penting wajib diisi.
   * Placeholder ditambahkan pada input teks untuk memberikan contoh pengisian.
3. **Tata Letak dan Gaya**:
   * Menggunakan inline CSS di dalam atribut style untuk memberikan warna latar belakang, margin, dan padding pada elemen.
   * Logo kampus ditambahkan di atas formulir untuk mempercantik halaman.
   * Formulir memiliki latar belakang putih dengan bayangan ringan untuk efek desain yang lebih menarik.
4. **Validasi dan Aksesibilitas**:
   * Menggunakan atribut required untuk validasi input yang wajib diisi.
   * Elemen label digunakan untuk memberikan keterangan yang jelas pada setiap input.
5. **Tambahan**:
   * Tautan ke halaman informasi kampus ditambahkan di bawah formulir.
   * Anda dapat menambahkan gambar logo kampus dengan mengganti URL gambar pada elemen <img>.
   * Simpan file ini ke GitHub dan buat video penjelasan sesuai dengan instruksi.

**Modul 2. Pengenalan CSS (Cascading Style Sheet)**

## A. Tujuan

Setelah mengikuti praktikum, Mahasiswa diharapkan mampu:

1. Mampu memahami dasar-dasar CSS.

2. Mampu mengenal lebih jauh properti CSS.

3. Dapat membuat program sederhana dengan menggunakan CSS

## B. Teori Dasar CSS

CSS (Cascading Style Sheets) adalah bahasa yang digunakan untuk mengatur tampilan dan layout halaman web. CSS memungkinkan kita mengubah warna, ukuran, posisi, dan gaya elemen HTML agar halaman web menjadi menarik dan mudah dibaca. CSS digunakan untuk mengatur tampilan elemen HTML di berbagai media seperti yang tertampil pada layar komputer atau perangkat lainnya. CSS memungkinkan pengembang web untuk mengontrol tata letak banyak halaman sekaligus sehingga menghemat waktu dan usaha dalam desain web.

**Pentingnya menggunakan CSS**

1. **Memisahkan** Konten dan Desain : HTML digunakan untuk menyusun konten sementara CSS menentukan tampilannya.

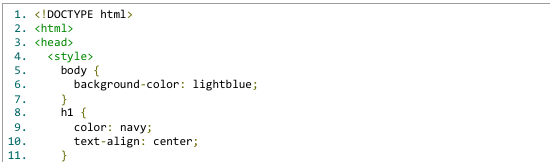
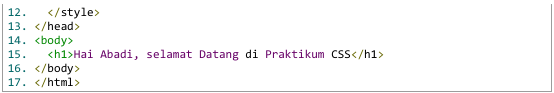
2. **Efisiensi** : Dengan satu file CSS tampilan seluruh situs web dapat diubah tanpa harus mengedit setiap halaman HTML.

3. **Konsistensi** : Memastikan tampilan yang seragam di seluruh halaman web.

4. **Responsivitas** : CSS memungkinkan desain yang dapat menyesuaikan dengan berbagai ukuran layar.

## C. Pratikum

1. Penulisan CSS (Cascading Style Sheets) Penjelasan Dalam penerepannya CSS (Cascading Style Sheets) bisa digunakan dengan bahasa HTML . Contoh Code File index.htm

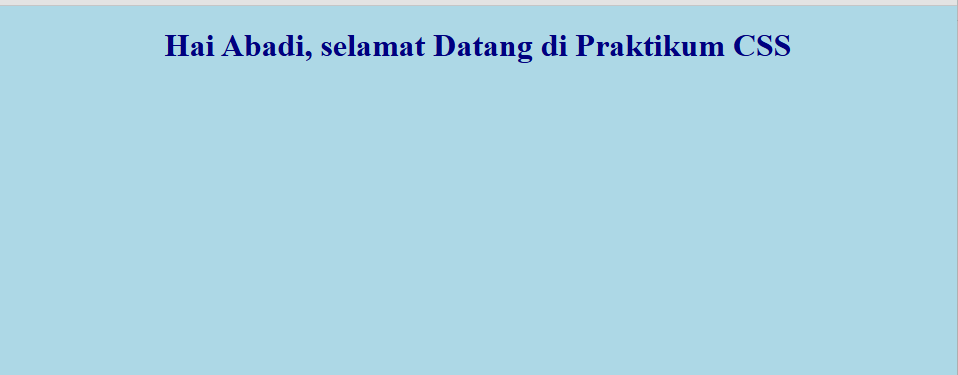
 

**Latihan Pratikum**

* Silakan pratikan contoh kode di atas
* Latihan selanjutnya buat halaman HTML dengan latar belakang berwarna kuning muda. Tambahkan judul dengan warna merah dan teks rata Tengah.
* Berikan *capture*(*printscreen*/tangkapan layer) ada **2 *capture*** yaitu dari contoh code dan Latihan kemudian analisis dari hasil pratikum anda di laporan.

**Jawab:**

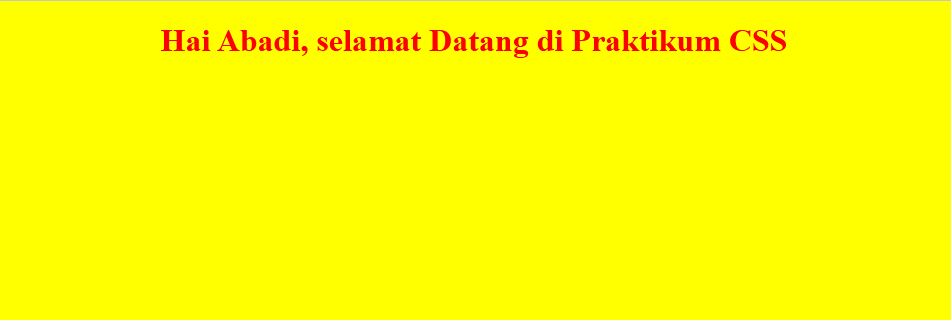
Berikut jika kode atas dipratktikan:



Untuk mengubah latar belakang berwarna kuning dan judul dengan warna merah dan teks rata Tengah, yaitu kode html+css jadi berikut:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  body {  background-color: yellow; /\* latar belakang kuning \*/  text-align: center; /\* teks di tengah \*/  }  h1 {  color: red; /\* judul berwarna merah \*/  }  </style>  </head>  <body>  <h1>Hai Abadi, selamat Datang di Praktikum CSS</h1>  </body>  </html> |

Jika dipraktikan:



Analisis:

* Versi **sebelum** lebih kalem, cocok untuk tampilan formal atau edukatif.
* Versi **sesudah** lebih **menarik dan kuat secara visual**, cocok untuk tampilan yang ingin cepat **menarik perhatian** (misalnya promosi, pengumuman, atau praktikum yang menyenangkan).
* **Struktur CSS** juga jadi lebih rapi dan efisien dengan text-align diatur langsung di body.

**2. Sintaks dan Cara menyisipkan CSS**

**Penjelsaan**

CSS terdiri dari selector dan deklarasi. Selector memilih elemen HTML yang ingin diatur sedangkan deklarasi berisi properti dan nilainya.

Ada tiga cara menyisipkan CSS:

a. Inline CSS (langsung di tag HTML)

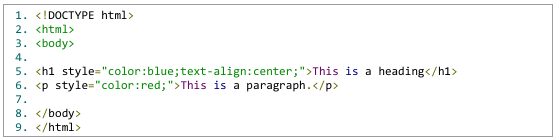
b. Internal CSS (di dalam tag <style> di bagian <head>)

c. External CSS (file terpisah dengan ekstensi .css)

Contoh Kode

Inline CSS:

File inline\_css.htm





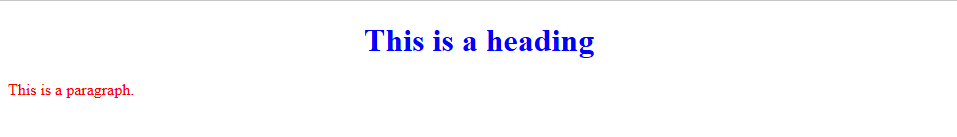


**Latihan pratikum**

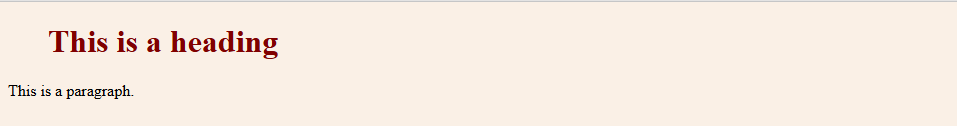
* Silakan pratikkan code diatas
* Buat file HTML dan file CSS eksternal. Hubungkan file CSS ke HTML dan ubah warna teks paragraph menjadi ungu
* Berikan *capture*(*print screen*/tangkapan layar) Ada **2 *capture*** yaitu dari contoh code dan Latihan kemudian analisis dari hasil pratikum anda di laporan

Pratik code di contoh soal:

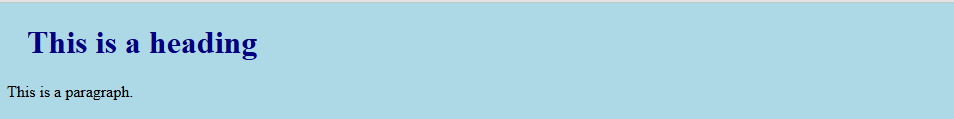
Inline css:

****

Internal css:



Eksternal CSS:

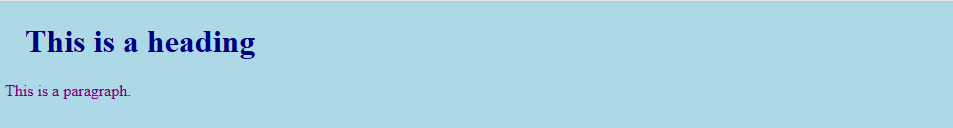


Buat file HTML dan File CSS Eksternal. Hubungkan file css ke HTML dan ubah warna paragraph menjadi ungu:

Dengan mengubah code di mystyle.css dengan tambahan p{color:purple;}

|  |
| --- |
| body {  background-color: lightblue;  }  h1 {  color: navy;  margin-left: 20px;  }  p {  color: purple;  } |

Berikut jika dipratikan:



**Analisis Setelah Perubahan**

Setelah menambahkan selector p dan menetapkan color: purple;:

* Paragraf tampil dengan warna ungu sesuai instruksi.
* Ini menunjukkan bahwa file CSS eksternal berhasil dihubungkan dan diterapkan dengan benar ke file HTML.
* Penggunaan CSS eksternal memungkinkan styling konsisten dan lebih mudah dikelola dibandingkan inline style.

**3. Selector CSS**

**Penjelasan**

Selector digunakan untuk memilih elemen HTML yang akan diberi gaya.

Contoh selector:

a. Selector elemen : p, h1

b. Selector kelas : kelas

c. Selector IDE : #id,

**Contoh Kode**

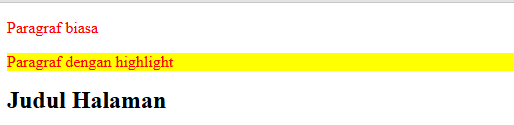


**Latihan Pratikum**

* Silakan praktikkan contoh *code* di atas.
* Buat beberapa paragraf dengan kelas berbeda dan atur warna latar belakangnya. Buat elemen dengan ID dan ubah ukuran *font*-nya.
* Berikan *capture* (*print screen* /tangkapan layar) Ada 2 *capture* yaitu dari contoh *code* dan latihan kemudian analisis dari hasil praktikum Anda di laporan.

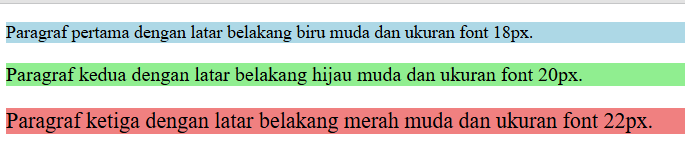
**Jawaban latihan pratikum:**

Jika contoh kode di atas dipraktikan:



Pembuatan paragraf dengan kelas berbeda dan atur warna latar belakang, buat elemen dengan IDE dan ubah ukuran font

Hasilnya:

****

**Dengan kode HTML:**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  .class1 {  background-color: lightblue;  font-size: 18px;}  .class2 {  background-color: lightgreen;  font-size: 20px;}  .class3 {  background-color: lightcoral;  font-size: 22px;}  </style>  </head>  <body>  <p class="class1">Paragraf pertama dengan latar belakang biru muda dan ukuran font 18px.</p>  <p class="class2">Paragraf kedua dengan latar belakang hijau muda dan ukuran font 20px.</p>  <p class="class3">Paragraf ketiga dengan latar belakang merah muda dan ukuran font 22px.</p>  </body>  </html> |

Analisa:

Kesimpulan:

* Kode 1 lebih sederhana, dengan penggunaan kelas dan ID yang terfokus pada elemen penting, dan sangat efektif dalam mengatur elemen spesifik seperti judul dan highlight.
* Kode 2 lebih cocok untuk memberikan tampilan yang bervariasi pada beberapa elemen sejenis (seperti paragraf) dan menawarkan fleksibilitas lebih dalam penataan.

**4. Properti Warna dan Background**

**Penjelasan**

CSS menyediakan properti untuk mengatur warna teks dan latar belakang, seperti:

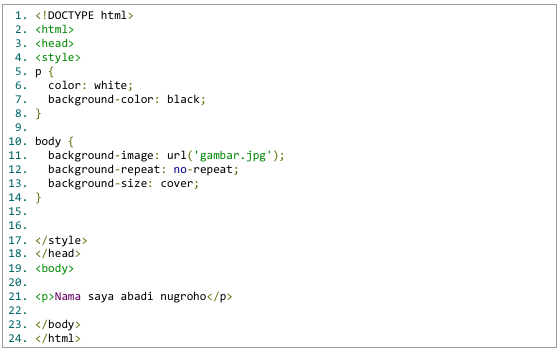
1. color : warna teks

2. background-color : warna latar belakang

3. Background-image : gambar latar belakang

**Contoh kode**

File propertiwarna\_css.html



**Latihan Pratikum**

* Silakan praktikkan contoh kode di atas
* Buat halaman dengan latar belakang gambar dan teks berwarna putih. Ubah warna latar belakang paragraf menjadi abu-abu.
* Berikan *capture* (*print screen*/tangkapan layar) ada **2 *capture*** yaitu dari contoh code dan latihan kemudian analisis dari hasil prktikum Anda di laporan.

Jawab:

Jika kode di atas dipraktikkan maka akan menghasilkan berikut:

Buat halaman dengan latar belakang gambar dan teks berwarna putih. Ubah warna latar belakang paragraf menjadi abu-abu. Berikut:



Dengan kode html:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  body {  background-image: url('file:///C:/Users/Asus/OneDrive/Pictures/Lampiran8.png');  background-repeat: no-repeat;  background-size: cover;  color: white; /\* Text color is white \*/  font-family: Arial, sans-serif;  }  p {  background-color: gray; /\* Gray background for paragraph \*/  padding: 1px;  border-radius: 1px;  }  </style>  </head>  <body>  <h1>Selamat datang di halaman saya</h1>  <p>Ini adalah paragraf dengan latar belakang abu-abu.</p>  </body>  </html> |

Analisa:

Kedua kode di atas memiliki fungsi serupa, yaitu menambahkan gambar latar belakang dan mengatur gaya teks dan paragraf. Perbedaan utama adalah penggunaan path gambar lokal pada kode pertama, yang membatasi akses pada komputer tertentu, sedangkan kode kedua menggunakan gambar dengan path relatif, yang lebih fleksibel untuk digunakan dalam proyek berbasis web.

**6. Box Model (Margin, Border, Padding, dan Content)**

**Penjelasan**

Setiap Elemen HTML dianggap sebagai kotak yang terdiri dari:

a. Content : isi elemen

b. Padding : ruang di dalam kotak yaitu antara isi dengan border

c. Border : Garis tepi kotak

d. Margin : ruang di lluar border yaitu memisahkan elemen dengan elemen lain

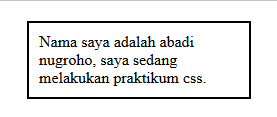
**Contoh kode**

File boxmodel\_css.htm

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  div {  margin: 20px;  padding: 10px;  border: 2px solid black;  width: 200px;  }  </style>  </head>  <body>  <div>Nama saya adalah abadi nugroho, saya sedang melakukan praktikum css.</div>  </body>  </html> |

**Latihan Praktikum**

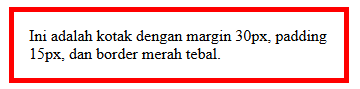
* Silakan Praktikkan contoh code di atas.

Jawab: 

* Buat kotak dengan margin 30px, padding 15px, dan border merah tebal. Ubah ukuran kotak menjadi 300px lebar

Jawab:

Berikut hasil codenya



Berikut codenya:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  div {  margin: 30px;  padding: 15px;  border: 5px solid red; /\* Border merah tebal \*/  width: 300px; /\* Lebar kotak 300px \*/  }  </style>  </head>  <body>  <div>Ini adalah kotak dengan margin 30px, padding 15px, dan border merah tebal.</div>  </body>  </html> |

* Analisa:

**Kode pertama** lebih cocok untuk desain yang lebih sederhana dan kompak, dengan ukuran kotak yang lebih kecil.

**Kode kedua** lebih cocok untuk menonjolkan elemen atau untuk penggunaan yang membutuhkan ruang lebih besar di dalam kotak dan border yang lebih tebal untuk menarik perhatian.

**7. Layout dengan Flexbox**

**Penjelasan**

*Flexbox* adalah metode *layout* modern yang memudahkan pengaturan posisi elemen dalam satu baris atau kolom

**Contoh Kode**

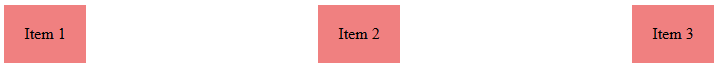
File flexbox\_css.html

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  .container {  display: flex;  justify-content: space-around;  align-items: center;  }  .item {  background-color: lightcoral;  padding: 20px;  margin: 10px;  }  </style>  </head>  <body>  <div class="container">  <div class="item">Item 1</div>  <div class="item">Item 2</div>  <div class="item">Item 3</div>  </div>  </body>  </html> |

**Latihan Pratikum**

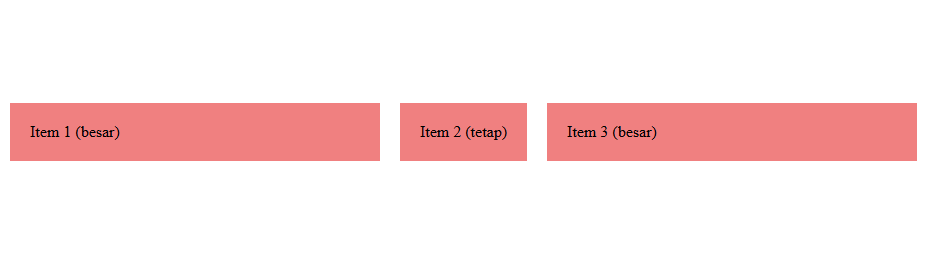
* Silakan praktikkan contoh code di atas.

Jawab:

****

* Buat container flex dengan 3 item. Atur agar item terbesar merata dan sejajar di tengah.

Jawab:

****

Dengan code:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  .container {  display: flex;  justify-content: center; /\* Menyelaraskan semua item di tengah \*/  align-items: center; /\* Menyelaraskan item secara vertikal di tengah \*/  height: 100vh; /\* Memberikan tinggi penuh untuk menampilkan hasil secara jelas \*/  }  .item {  background-color: lightcoral;  padding: 20px;  margin: 10px;  }  .item1 {  flex-grow: 1; /\* Item pertama akan berkembang untuk mengisi ruang yang tersedia \*/  }  .item2 {  flex-grow: 0; /\* Item kedua memiliki ukuran tetap \*/  }  .item3 {  flex-grow: 1; /\* Item ketiga akan berkembang untuk mengisi ruang yang tersedia \*/  }  </style>  </head>  <body>  <div class="container">  <div class="item item1">Item 1 (besar)</div>  <div class="item item2">Item 2 (tetap)</div>  <div class="item item3">Item 3 (besar)</div>  </div>  </body>  </html> |

Analisa dari kedua kode di atas:

* **Kode pertama** lebih cocok untuk desain yang lebih sederhana, di mana item harus diposisikan secara merata tanpa perlu mengubah ukuran.
* **Kode kedua** lebih cocok untuk desain yang membutuhkan pengaturan ukuran fleksibel pada item, seperti menonjolkan item terbesar di tengah atau memberi ukuran tetap pada item lainnya.

**8. CSS Grid Layout**

**Penjelasan**

CSS Grid adalah metode layout dua dimensi yang sangat powerful untuk mengatur elemen-elemen dalam baris dan kolom. Dengan Grid kita bisa membuat layout yang kompleks dengan mudah dan fleksibel.

Konsep Utama CSS Grid:

a. Grid Container: Elemen induk yang diatur sebagai grid dengan properti display: grid;

b. Grid Items: Elemen anak di dalam grid container yang akan diatur posisinya

c. Grid Lines: Garis-garis pembatas baris dan kolom

d. Grid Tracks: Baris dan kolom yang dibentuk oleh grid lines

e. Grid Areas: Area yang dibentuk oleh beberapa grid cells

**Contoh Kode**

File grid\_css.html

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>Grid CSS Example</title>  <style>  .grid-container {  display: grid;  grid-template-columns: 100px 200px 100px;  grid-template-rows: 50px 100px;  gap: 10px;  background-color: #f2f2f2;  padding: 10px;  }  .grid-item {  background-color: #4CAF50;  color: white;  text-align: center;  padding: 20px;  font-size: 18px;  }  </style>  </head>  <body>  <h2>SEKOLAH TINGGI TEKNOLOGI BONTANG</h2>  <p>PROGRAM STUDI TEKNIK INFORMATIKA</p>  <p><strong>Praktikum:</strong> Pemrograman Web</p>  <p><strong>Modul 2:</strong> Pengenalan CSS (Cascading Style Sheet)</p>  <p><strong>Tgl Update:</strong> 02-06-2025</p>  <div class="grid-container">  <div class="grid-item">1</div>  <div class="grid-item">2</div>  <div class="grid-item">3</div>  <div class="grid-item">4</div>  <div class="grid-item">5</div>  <div class="grid-item">6</div>  </div>  </body>  </html> |

**Latihan Praktikum**

* Silakan praktikkan contoh kode diatas. Berikut:  
  
* Buatlah grid container dengan 3 kolom dan 2 baris. Atur jarak antar grid item sebesar 15 px. Buat 6 grid *item* dengan warna latar berbeda. Cobalah mengatur salah satu grid item agar menempati dua kolom sekaligus.

Berikut Code:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>Grid Layout 3x2</title>  <style>  .grid-container {  display: grid;  grid-template-columns: repeat(3, 1fr); /\* 3 kolom sama lebar \*/  grid-template-rows: repeat(2, 100px); /\* 2 baris tinggi 100px \*/  gap: 15px;  padding: 20px;  background-color: #f5f5f5;  }  .grid-item {  display: flex;  align-items: center;  justify-content: center;  color: white;  font-size: 18px;  font-weight: bold;  border-radius: 8px;  }  .item1 { background-color: #e74c3c; }  .item2 { background-color: #3498db; }  .item3 { background-color: #f1c40f; }  .item4 { background-color: #2ecc71; }  .item5 { background-color: #9b59b6; }  .item6 {  background-color: #34495e;  grid-column: span 2; /\* Menempati 2 kolom secara horizontal \*/  }  </style>  </head>  <body>  <div class="grid-container">  <div class="grid-item item1">Item 1</div>  <div class="grid-item item2">Item 2</div>  <div class="grid-item item3">Item 3</div>  <div class="grid-item item4">Item 4</div>  <div class="grid-item item5">Item 5</div>  <div class="grid-item item6">Item 6 (2 kolom)</div>  </div>  </body>  </html> |

Berikut jika code di atas dijalankan di web browser:



* Analisa:

Kesimpulan Perbandingan

| **Aspek** | **Kode 1** | **Kode 2** |
| --- | --- | --- |
| Tipe Grid | Fixed size (px) | Flexible (fr unit) |
| Jumlah Item | 6 | 6 |
| Variasi Warna | Tidak ada (semua hijau) | Ada (setiap item beda warna) |
| Gap antar item | 10px | 15px (lebih lega) |
| Item menempati 2 kolom | ❌ Tidak ada | ✅ Ya, .item6 pakai grid-column: span 2 |
| Estetika visual | Sederhana dan kaku | Modern dan dinamis |
| Cocok untuk | Pembelajaran dasar CSS Grid | Implementasi layout nyata (dashboard, card layout) |

**9. Responsive Desain (Desain Responsif)**

**Penjelasan**

Responsive Design adalah teknik membuat tampilan web yang dapat menyesuaikan diri dengan berbagai ukuran layar perangkat, seperti desktop, tablet, dan smartphone. Tujuannya agar website tetap nyaman dilihat dan digunakan di semua perangkat.

Cara utama membuat desain responsif:

a. Menggunakan media queries untuk mengubah gaya CSS berdasarkan ukuran layar.

b. Menggunakan layout fleksibel seperti Flexbox dan Grid.

c. Menggunakan satuan relatif seperti %, em, dan rem daripada satuan tetap seperti px.

**Media Queries**

Media queries memungkinkan kita menulis aturan CSS yang hanya berlaku pada kondisi tertentu, misalnya ukuran layar tertentu.

**Contoh Kode**

Kode dibawah ini tidak perlu di praktikkan, cukup dipahami.

|  |
| --- |
| /\* Gaya default untuk desktop \*/  body {  background-color: lightblue;  font-size: 18px;  }  /\* Jika layar maksimal 600px (misal smartphone) \*/  @media only screen and (max-width: 600px) {  body {  background-color: lightcoral;  font-size: 14px;  }  } |

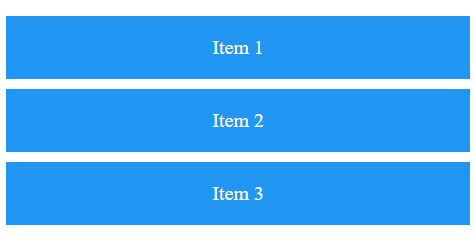
File responsive\_css.html

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>Responsive Grid Example</title>  <style>  .grid-container {  display: grid;  grid-template-columns: repeat(3, 1fr);  gap: 10px;  padding: 10px;  }  .grid-item {  background-color: #2196F3;  color: white;  padding: 20px;  font-size: 20px;  text-align: center;  }  /\* Responsive untuk layar kecil \*/  @media only screen and (max-width: 600px) {  .grid-container {  grid-template-columns: 1fr; /\* 1 kolom \*/  }  }  </style>  </head>  <body>  <div class="grid-container">  <div class="grid-item">Item 1</div>  <div class="grid-item">Item 2</div>  <div class="grid-item">Item 3</div>  </div>  </body>  </html> |

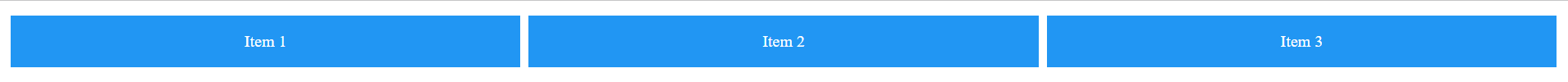
**Laporan Pratikum**

* Mempraktikan contoh kode di atas:

Berikut jika lebar layar terlalu kecil:

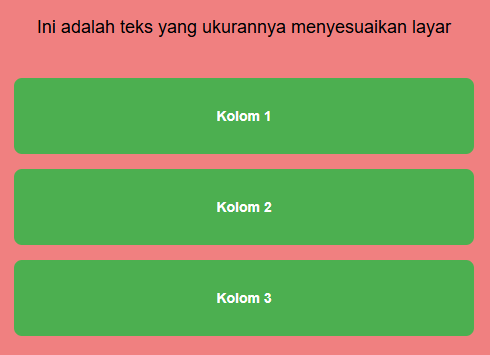
****

Berikut jika lebar layar mencukupi:



* Buat halaman dengan grid 3 kolom untuk layar desktop. Gunakan media query agar pada layar dengan lebar maksimal 600px, grid berubah menjadi 1 kolom. Ubah warna latar belakang halaman sesuai ukuran layar (misal biru untuk desktop, merah untuk smartphone). Coba tambahkan teks yang ukurannya berubah sesuai ukuran layar.

Berikut jawaban jika tampilan smartphone:



Berikut tampilan desktop:



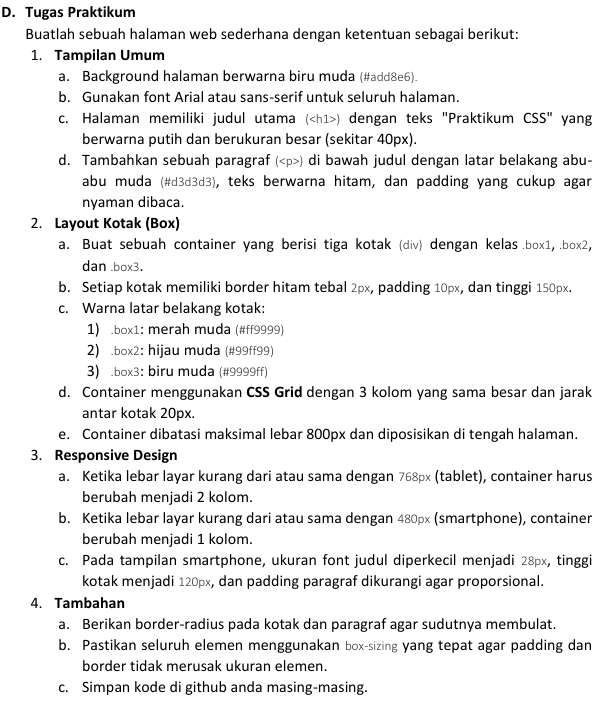
Dengan kode:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>Responsive Grid Layout</title>  <style>  body {  background-color: lightblue;  font-size: 18px;  margin: 0;  font-family: Arial, sans-serif;  }  .container {  display: grid;  grid-template-columns: repeat(3, 1fr);  gap: 15px;  padding: 20px;  }  .grid-item {  background-color: #4CAF50;  color: white;  padding: 30px;  text-align: center;  border-radius: 8px;  font-weight: bold;  }  .responsive-text {  text-align: center;  padding: 20px;  font-size: 24px;  }  @media only screen and (max-width: 600px) {  body {  background-color: lightcoral;  font-size: 14px;  }  .container {  grid-template-columns: 1fr;  }  .responsive-text {  font-size: 18px;  }  }  </style>  </head>  <body>  <div class="responsive-text">  Ini adalah teks yang ukurannya menyesuaikan layar  </div>  <div class="container">  <div class="grid-item">Kolom 1</div>  <div class="grid-item">Kolom 2</div>  <div class="grid-item">Kolom 3</div>  </div>  </body>  </html> |

Analisa:

Kode ini **sangat baik sebagai latihan atau proyek awal** untuk memahami prinsip **CSS Grid dan media queries**. Ia mencerminkan praktik modern dalam membuat halaman yang bisa **beradaptasi otomatis dengan ukuran layar**, memperhatikan **kenyamanan visual**, dan memiliki struktur yang bisa dengan mudah dikembangkan ke versi yang lebih kompleks (misalnya menambahkan gambar, tombol, navigasi, dsb).

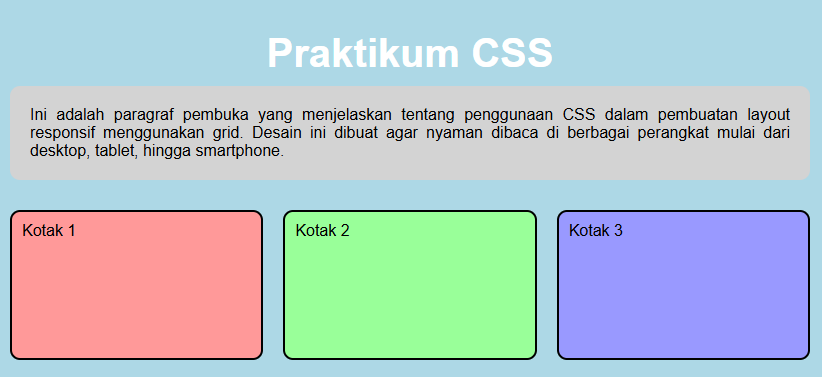
## D. Tugas Praktikum



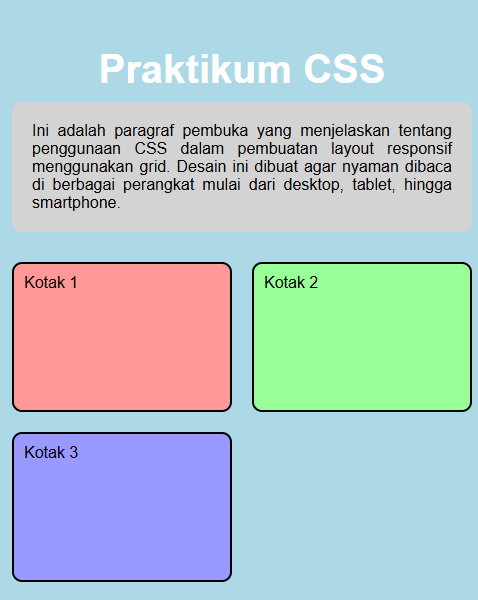


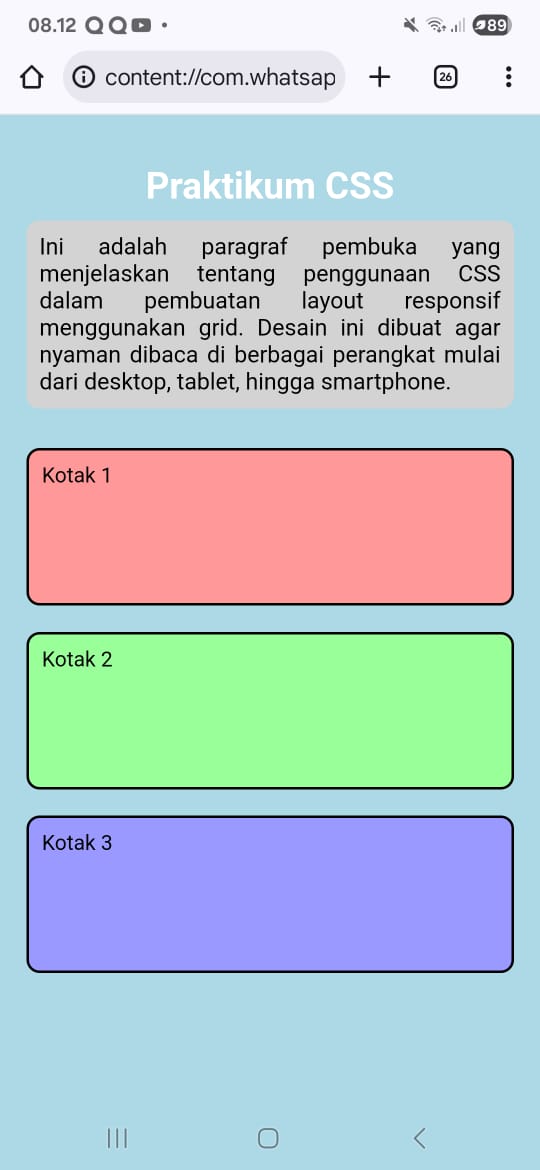
Link vidio dapat didownload pada link berikut: [Link](https://github.com/Martonius3/Laporan-Pratikum-Pemrograman-WEB/raw/refs/heads/Modul-2/202312051%20Martonius%20Pemrograman%20WEB%20Modul%202%20Tugas%20Pratikum.mp4)

Berikut hasilnya:



Berikut jika dijalankan di web browser dnegan lebar kecil (smartphone):





Dengan kode:

|  |
| --- |
| <!DOCTYPE html>  <html lang="id">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Praktikum CSS</title>  <style>  /\* Box sizing untuk semua elemen \*/  \* {  box-sizing: border-box;  }  body {  background-color: #add8e6; /\* biru muda \*/  font-family: Arial, sans-serif;  margin: 0;  padding: 20px;  }  h1 {  color: white;  font-size: 40px;  text-align: center;  margin-bottom: 10px;  }  p {  background-color: #d3d3d3; /\* abu-abu muda \*/  color: black;  padding: 20px;  border-radius: 10px;  max-width: 800px;  margin: 0 auto 30px auto;  text-align: justify;  }  .container {  display: grid;  grid-template-columns: repeat(3, 1fr);  gap: 20px;  max-width: 800px;  margin: 0 auto;  }  .box1, .box2, .box3 {  border: 2px solid black;  padding: 10px;  height: 150px;  border-radius: 10px;  }  .box1 {  background-color: #ff9999;  }  .box2 {  background-color: #99ff99;  }  .box3 {  background-color: #9999ff;  }  /\* Tablet (≤768px): 2 kolom \*/  @media only screen and (max-width: 768px) {  .container {  grid-template-columns: repeat(2, 1fr);  }  }  /\* Smartphone (≤480px): 1 kolom \*/  @media only screen and (max-width: 480px) {  h1 {  font-size: 28px;  }  p {  padding: 10px;  }  .container {  grid-template-columns: 1fr;  }  .box1, .box2, .box3 {  height: 120px;  }  }  </style>  </head>  <body>  <h1>Praktikum CSS</h1>  <p>  Ini adalah paragraf pembuka yang menjelaskan tentang penggunaan CSS dalam pembuatan layout responsif menggunakan grid. Desain ini dibuat agar nyaman dibaca di berbagai perangkat mulai dari desktop, tablet, hingga smartphone.  </p>  <div class="container">  <div class="box1">Kotak 1</div>  <div class="box2">Kotak 2</div>  <div class="box3">Kotak 3</div>  </div>  </body>  </html> |

**Modul 3. Pengenalan CSS (Cascading Style Sheet)**

## A. Tujuan

Setelah mengikuti praktikum, Mahasiswa diharapkan mampu:

1. Mampu memahami bootstrap sebagai framework CSS.

2. Dapat membuat program dengan framework CSS.

## B. Teori Dasar Bootsrap

Boostrap adalah framework CSS *open-source* yang sangat populer untuk membangun website dan aplikasi web yang responsif. Bootstrap 5 adalah versi terbaru yang membawa banyak perbaikan dan fitur baru dibanding versi sebelumnya

**Keunggulan Boostrap 5**

1. Mudah digunakan kelas-kelas CSS siap pakai

2. Sistem grid yang fleksibel untuk layout responsif

3. Banyak komponen UI siap pakai (button, navbar, card, modal, dll)

4. Dukungan utilities untuk styling cepat

5. Tidak lagi bergantung pada jQuery (lebih ringan dan modern).

## C. Pratikum

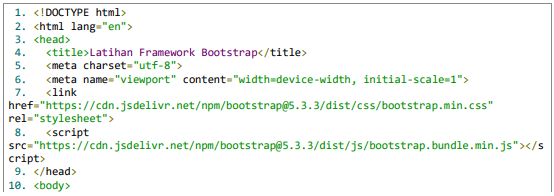
**1. Instalasi dan Setup Bootstrap**

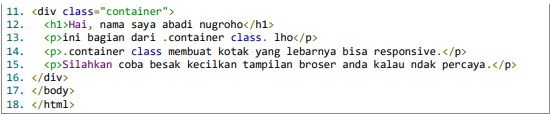
**Menggunakan CDN ( Content Delivery Nerwork)**

Penulisan kode di dalam tag <head> pada file HTML.

  
**Contoh Code**

File setup\_bootstrap.html



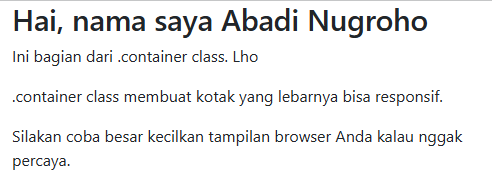


**Latihan Praktikum:**

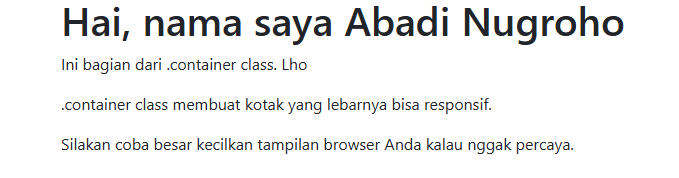
* Silakan praktikan contoh kode di atas.

Jawab:

Saat tampilan browser kecil:



Saat tampilan browser besar:

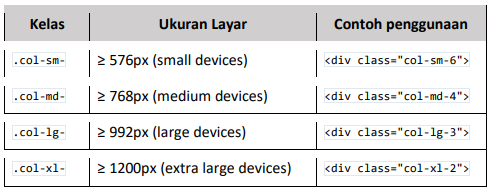


**2. Sistem grid di Bootsrap 5**

**Penjelasan**

Bootsrap menggunakan sistem grid berbasi flexbox yang membagi halaman menjadi 12 kolom. Anda dapat mengatur layout dengan mudah menggunakan kelas grid.

Bootsrap menyediakan kelas grid responsif berdasarkan ukuran layar:

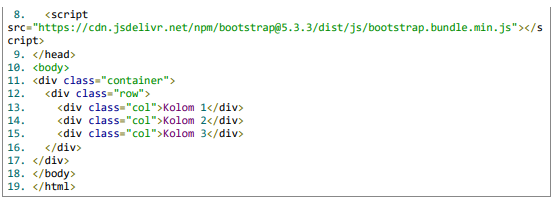


**Contoh Kode**

**Grid 3 Kolom**

File grid\_bootsrap.html





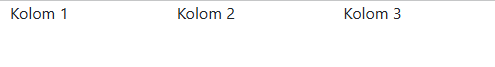
.container : pembungkus utama yang memberikan padding dan mengatur lebar maksimal.

.row : baris yang berisi kolom

.col : kolom yang secara otomatis membagi ruang secara merata.

**Latihan Praktikum**

* Silakan praktikan contoh kode diatas

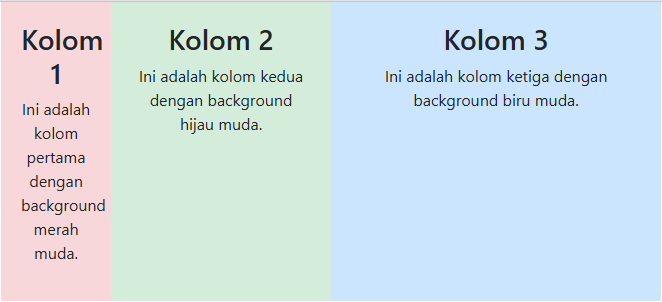
Jawab

* Buatlah halaman web yang memiliki 3 kolom menggunakan grid bootsrap. Setiap kolom berisi teks berbeda dan memiliki background warna berbeda.

Berikut kodenya

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Halaman Web 3 Kolom</title>  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>  <style>  .col-1 {  background-color: #f8d7da; /\* Background merah muda \*/  padding: 20px;  text-align: center;  }  .col-2 {  background-color: #d4edda; /\* Background hijau muda \*/  padding: 20px;  text-align: center;  }  .col-3 {  background-color: #cce5ff; /\* Background biru muda \*/  padding: 20px;  text-align: center;  }  </style>  </head>  <body>  <div class="container">  <div class="row">  <div class="col col-1">  <h3>Kolom 1</h3>  <p>Ini adalah kolom pertama dengan background merah muda.</p>  </div>  <div class="col col-2">  <h3>Kolom 2</h3>  <p>Ini adalah kolom kedua dengan background hijau muda.</p>  </div>  <div class="col col-3">  <h3>Kolom 3</h3>  <p>Ini adalah kolom ketiga dengan background biru muda.</p>  </div>  </div>  </div>  </body>  </html> |

Berikut Hasilnya jika dijalankan di web browser



* **Analisa:**

Secara keseluruhan, kode ini membuat halaman web responsif dengan tiga kolom menggunakan Bootstrap grid system. Desain ini akan berfungsi baik di perangkat dengan ukuran layar besar maupun kecil, dan memberikan pengalaman pengguna yang baik dengan warna latar belakang yang berbeda untuk masing-masing kolom.

**3. Typography dan Utilities**

**Penjelasan**

Bostrap menyediakan kelas untuk mengatur teks dan utilitas styling cepat.

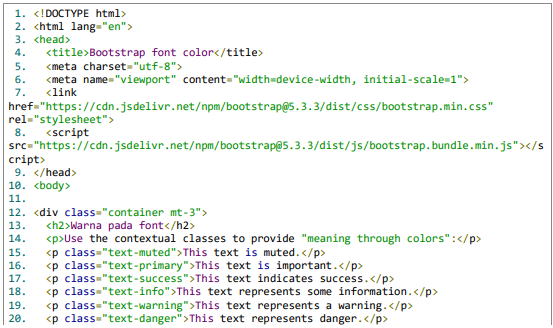
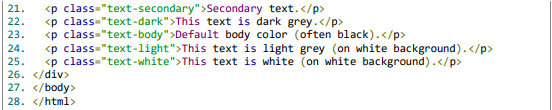
**Contoh Kode**

File typo\_bootsrap.html

**Text colors**

File text\_colors\_bootsrap.html

**Background Colors**

File bg\_colors\_bootsrap.html

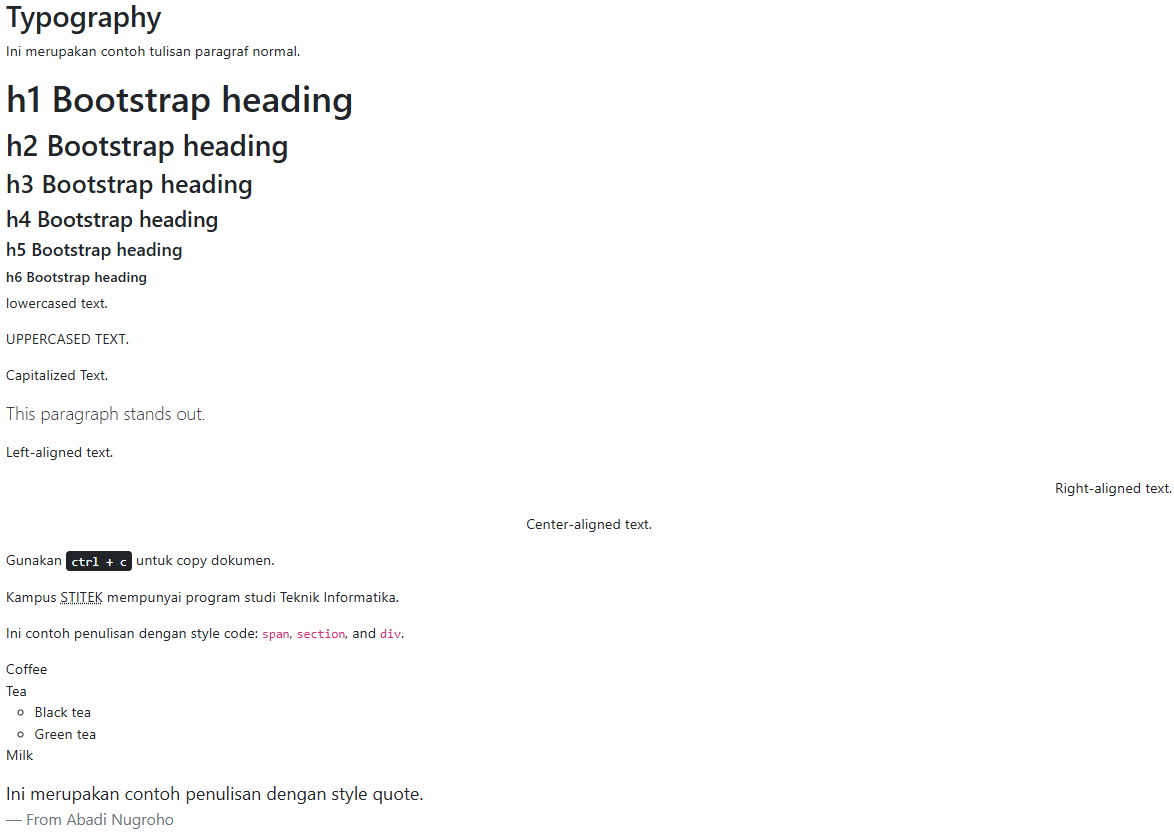


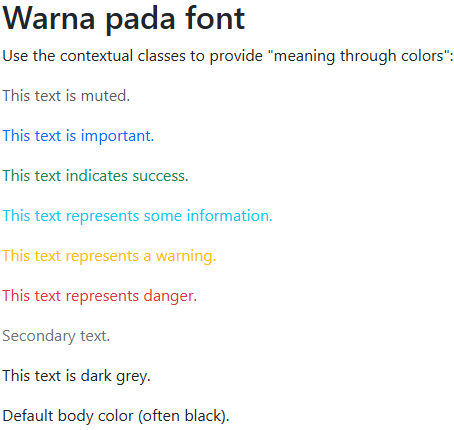
**Latihan Praktikum**

* Silakan praktikkan contoh kode diatas

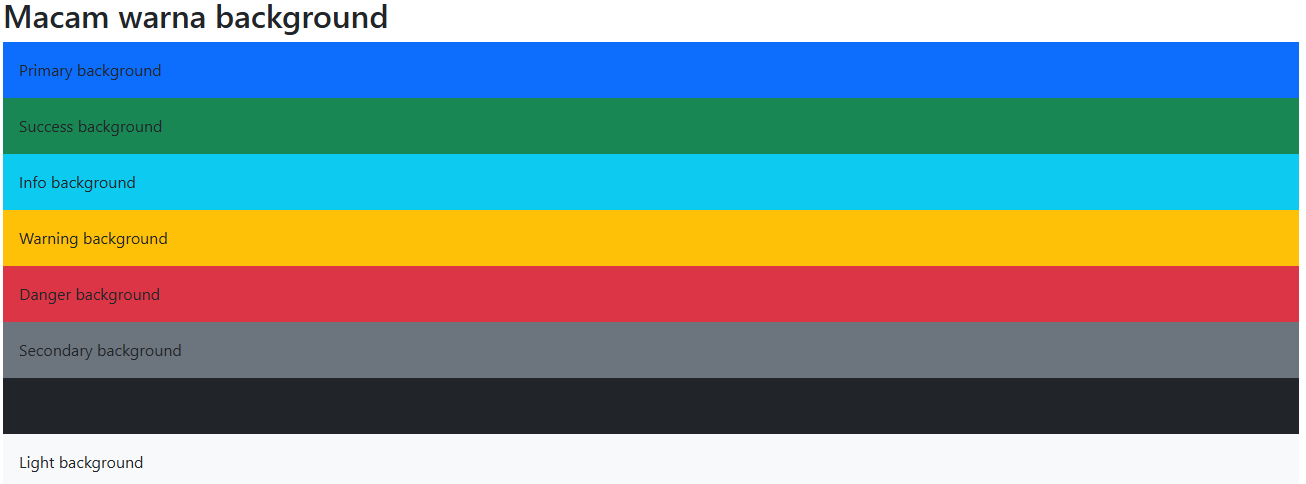
Jawab:

Berikut kode pertama jika dijalankan di web browser:



Berikut Kode kedua jika dijalankan di web browser ;

Berikut kode ketiga dijalankan di web browser:



**4. Tables**

**Penjelasan**

Pada bootsrap juga tersedia berbagai macam desain table. Antara lain sebagai beirkut:

.table

.table-striped

.table-bordered

.table-hover

.table-dark

.table-borderless

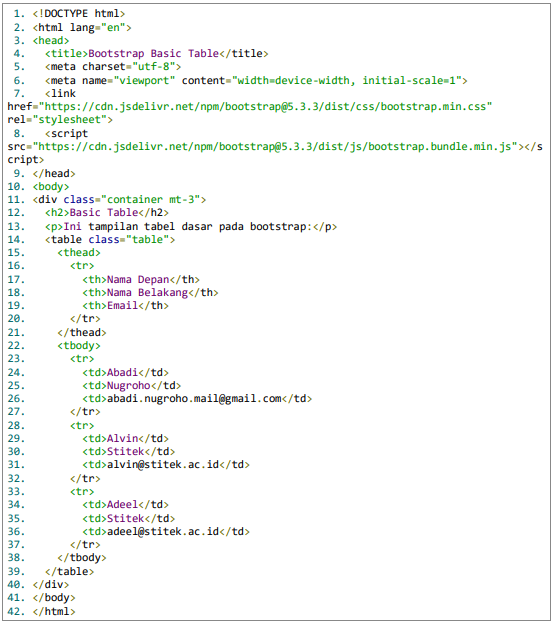
.table-sm

.table-sm

.table-responsive

**Contoh Kode**

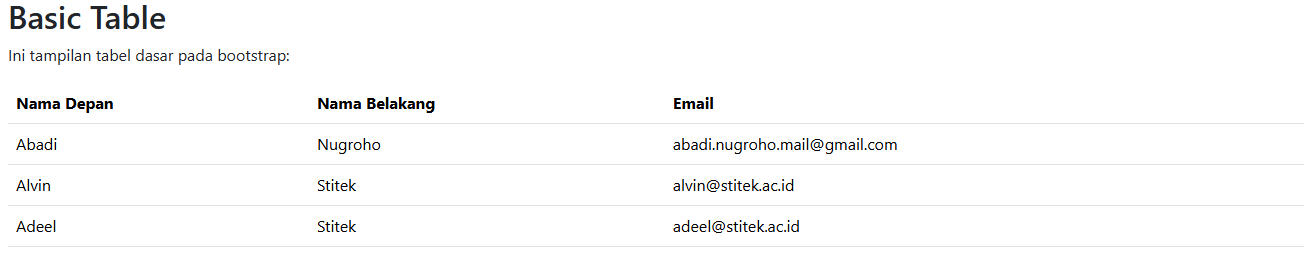
File basic\_table\_bootsrap.html



**Latihan Praktikum**

* Silakan praktikkan contoh kode di atas.

Jawab:

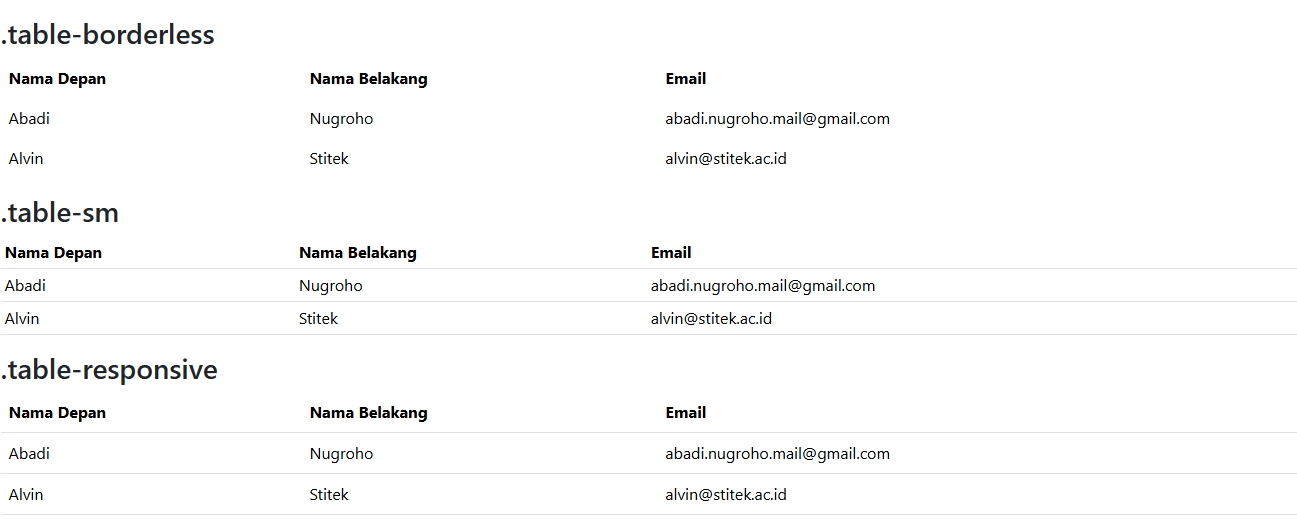
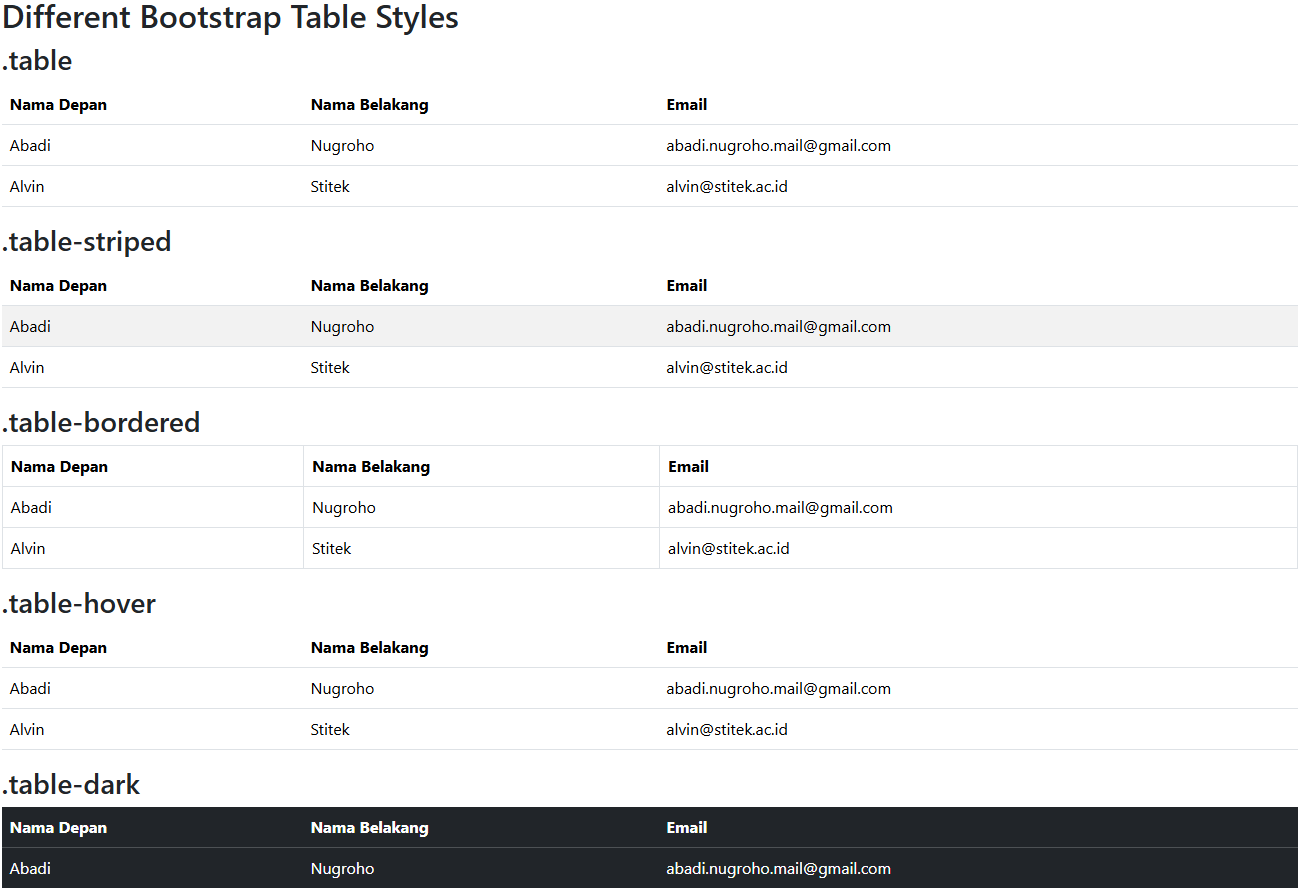


* Silakan coba style yang lain .table, .table-striped, .table-bordered, .table-hover, .table-dark, .table-borderless, . table-sm, .table-responsive.

Berikut kodenya:

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>Bootstrap Table Styles</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>  </head>  <body>  <div class="container mt-3">  <h2>Different Bootstrap Table Styles</h2>    <h3>.table</h3>  <table class="table">  <thead>  <tr>  <th>Nama Depan</th>  <th>Nama Belakang</th>  <th>Email</th>  </tr>  </thead>  <tbody>  <tr>  <td>Abadi</td>  <td>Nugroho</td>  <td>abadi.nugroho.mail@gmail.com</td>  </tr>  <tr>  <td>Alvin</td>  <td>Stitek</td>  <td>alvin@stitek.ac.id</td>  </tr>  </tbody>  </table>  <h3>.table-striped</h3>  <table class="table table-striped">  <thead>  <tr>  <th>Nama Depan</th>  <th>Nama Belakang</th>  <th>Email</th>  </tr>  </thead>  <tbody>  <tr>  <td>Abadi</td>  <td>Nugroho</td>  <td>abadi.nugroho.mail@gmail.com</td>  </tr>  <tr>  <td>Alvin</td>  <td>Stitek</td>  <td>alvin@stitek.ac.id</td>  </tr>  </tbody>  </table>  <h3>.table-bordered</h3>  <table class="table table-bordered">  <thead>  <tr>  <th>Nama Depan</th>  <th>Nama Belakang</th>  <th>Email</th>  </tr>  </thead>  <tbody>  <tr>  <td>Abadi</td>  <td>Nugroho</td>  <td>abadi.nugroho.mail@gmail.com</td>  </tr>  <tr>  <td>Alvin</td>  <td>Stitek</td>  <td>alvin@stitek.ac.id</td>  </tr>  </tbody>  </table>  <h3>.table-hover</h3>  <table class="table table-hover">  <thead>  <tr>  <th>Nama Depan</th>  <th>Nama Belakang</th>  <th>Email</th>  </tr>  </thead>  <tbody>  <tr>  <td>Abadi</td>  <td>Nugroho</td>  <td>abadi.nugroho.mail@gmail.com</td>  </tr>  <tr>  <td>Alvin</td>  <td>Stitek</td>  <td>alvin@stitek.ac.id</td>  </tr>  </tbody>  </table>  <h3>.table-dark</h3>  <table class="table table-dark">  <thead>  <tr>  <th>Nama Depan</th>  <th>Nama Belakang</th>  <th>Email</th>  </tr>  </thead>  <tbody>  <tr>  <td>Abadi</td>  <td>Nugroho</td>  <td>abadi.nugroho.mail@gmail.com</td>  </tr>  <tr>  <td>Alvin</td>  <td>Stitek</td>  <td>alvin@stitek.ac.id</td>  </tr>  </tbody>  </table>  <h3>.table-borderless</h3>  <table class="table table-borderless">  <thead>  <tr>  <th>Nama Depan</th>  <th>Nama Belakang</th>  <th>Email</th>  </tr>  </thead>  <tbody>  <tr>  <td>Abadi</td>  <td>Nugroho</td>  <td>abadi.nugroho.mail@gmail.com</td>  </tr>  <tr>  <td>Alvin</td>  <td>Stitek</td>  <td>alvin@stitek.ac.id</td>  </tr>  </tbody>  </table>  <h3>.table-sm</h3>  <table class="table table-sm">  <thead>  <tr>  <th>Nama Depan</th>  <th>Nama Belakang</th>  <th>Email</th>  </tr>  </thead>  <tbody>  <tr>  <td>Abadi</td>  <td>Nugroho</td>  <td>abadi.nugroho.mail@gmail.com</td>  </tr>  <tr>  <td>Alvin</td>  <td>Stitek</td>  <td>alvin@stitek.ac.id</td>  </tr>  </tbody>  </table>  <h3>.table-responsive</h3>  <div class="table-responsive">  <table class="table">  <thead>  <tr>  <th>Nama Depan</th>  <th>Nama Belakang</th>  <th>Email</th>  </tr>  </thead>  <tbody>  <tr>  <td>Abadi</td>  <td>Nugroho</td>  <td>abadi.nugroho.mail@gmail.com</td>  </tr>  <tr>  <td>Alvin</td>  <td>Stitek</td>  <td>alvin@stitek.ac.id</td>  </tr>  </tbody>  </table>  </div>  </div>  </body>  </html> |

Berikut hasilnya jika dijalankan di web browser:



* Analisa hasil praktikum :

Kode ini menunjukkan berbagai fitur yang disediakan oleh Bootstrap untuk mendesain tabel dengan cara yang fleksibel dan responsif. Dengan menerapkan berbagai kelas Bootstrap seperti .table, .table-striped, .table-bordered, dll., pengguna dapat dengan mudah membuat tabel yang lebih menarik dan fungsional sesuai dengan kebutuhan tampilan atau interaksi yang diinginkan. Penggunaan .table-responsive juga memastikan bahwa tabel tetap responsif di berbagai perangkat, meningkatkan pengalaman pengguna di berbagai ukuran layar.

**5. Komponen Bootsrap 5**

**Penjelasan**

Ada berbagai macam komponen yang bisa kita gunakan :

**Contoh Kode**

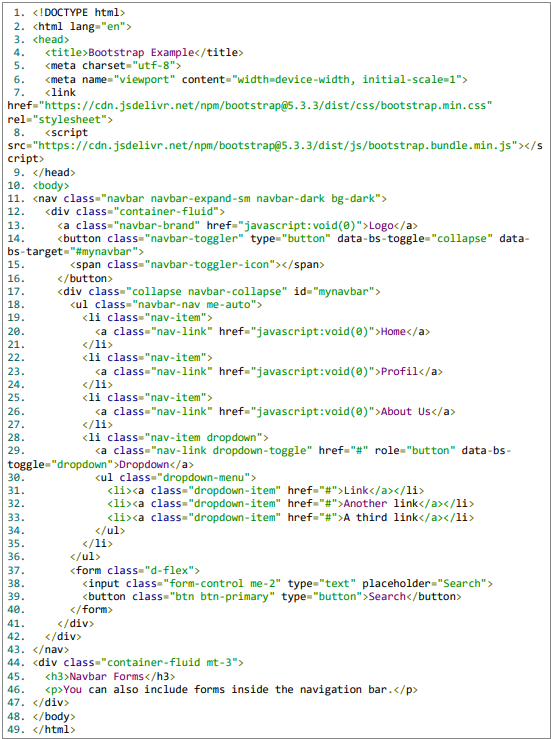
**Button**

File buuton\_bootsrap.html

****

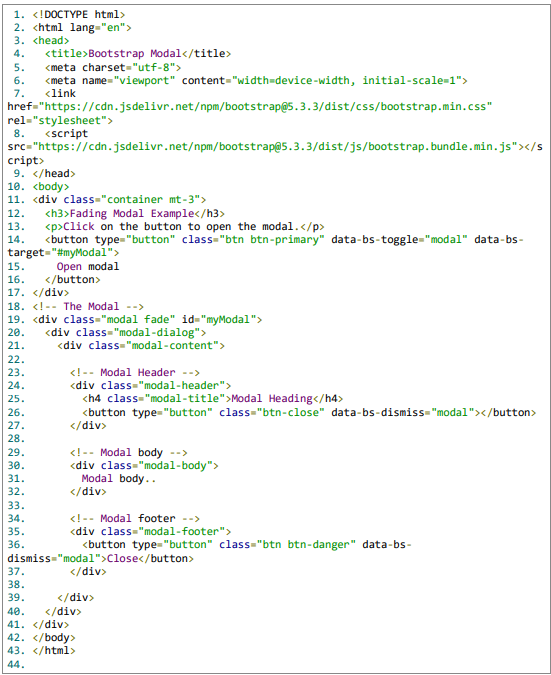
**Navbars**

File navbar\_bootsrap.html



**Modal**

File modal\_bootsrap.html



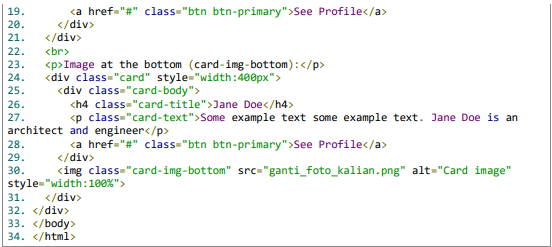
**Pagination**

File pagination\_bootsrap.html



**Card**

File card\_bootsrap.html

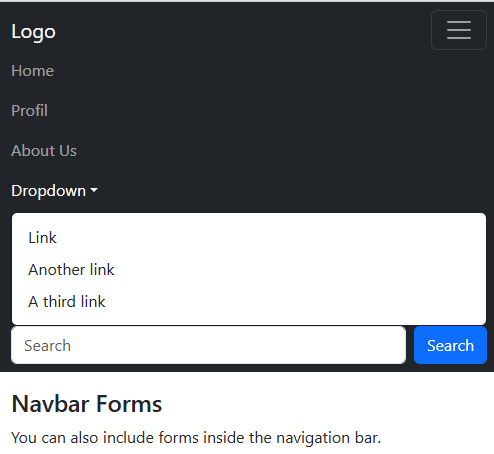
* Silakan praktikkan contoh kode di atas.

Button:



Navbar:

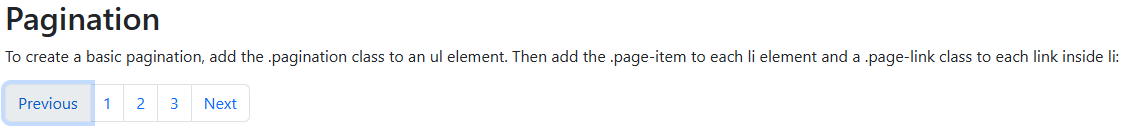




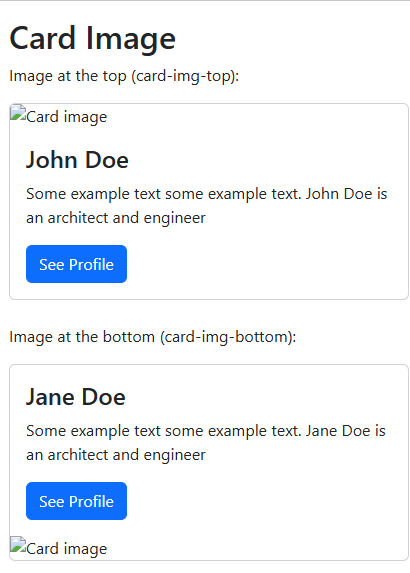
Modal:



Pagination:



Card:



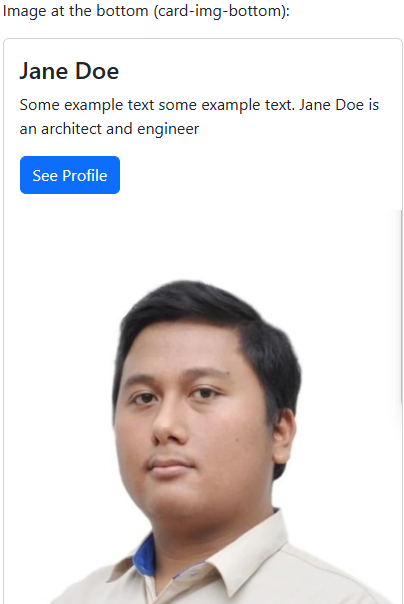
* Pada bagian card silahkan pasang foto kalian masing-masing simpan dalam satu folder yang sama dengan file script card\_bootstrap.htm

Berikut kodenya:

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>Bootstrap Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>  </head>  <body>  <div class="container mt-3">  <h2>Card Image</h2>  <p>Image at the top (card-img-top):</p>  <div class="card" style="width:400px">  <!-- Gantilah 'foto\_saya.png' dengan nama file foto Anda -->  <img class="card-img-top" src="foto\_saya.png" alt="Card image" style="width:100%">  <div class="card-body">  <h4 class="card-title">John Doe</h4>  <p class="card-text">Some example text some example text. John Doe is an architect and engineer</p>  <a href="#" class="btn btn-primary">See Profile</a>  </div>  </div>  <br>  <p>Image at the bottom (card-img-bottom):</p>  <div class="card" style="width:400px">  <div class="card-body">  <h4 class="card-title">Jane Doe</h4>  <p class="card-text">Some example text some example text. Jane Doe is an architect and engineer</p>  <a href="#" class="btn btn-primary">See Profile</a>  </div>  <!-- Gantilah 'foto\_saya.png' dengan nama file foto Anda -->  <img class="card-img-bottom" src="foto\_saya.png" alt="Card image" style="width:100%">  </div>  </div>  </body>  </html> |

Dan hasil screenshot:

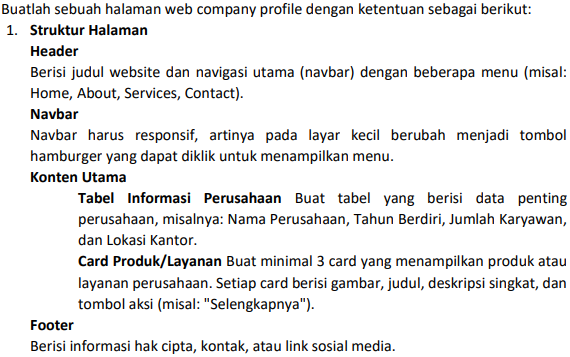


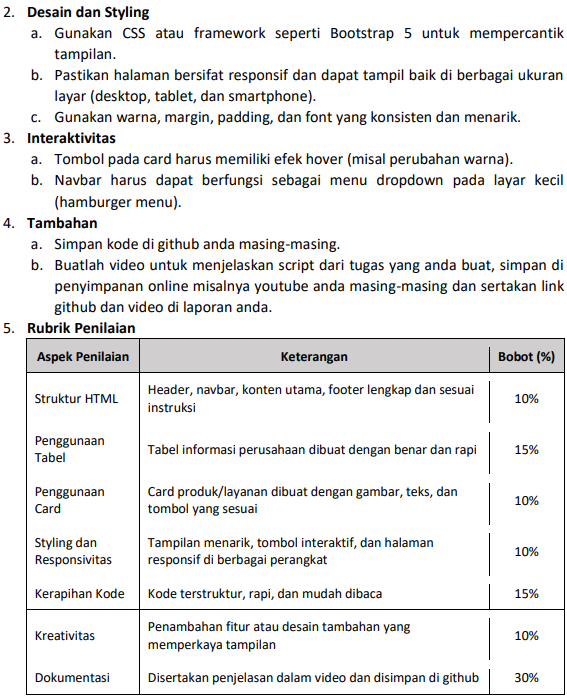


* Analisa:

Bootstrap menyediakan komponen UI siap pakai seperti **button**, **navbar**, **modal**, **pagination**, dan **card** yang memungkinkan pembuatan halaman web responsif dengan desain modern. Setiap komponen menggunakan kelas CSS yang sudah ditentukan untuk memastikan konsistensi visual dan fungsionalitas, dengan kemampuan responsif yang kuat untuk memastikan pengalaman pengguna yang baik di berbagai perangkat.

## D. Tugas Praktikum



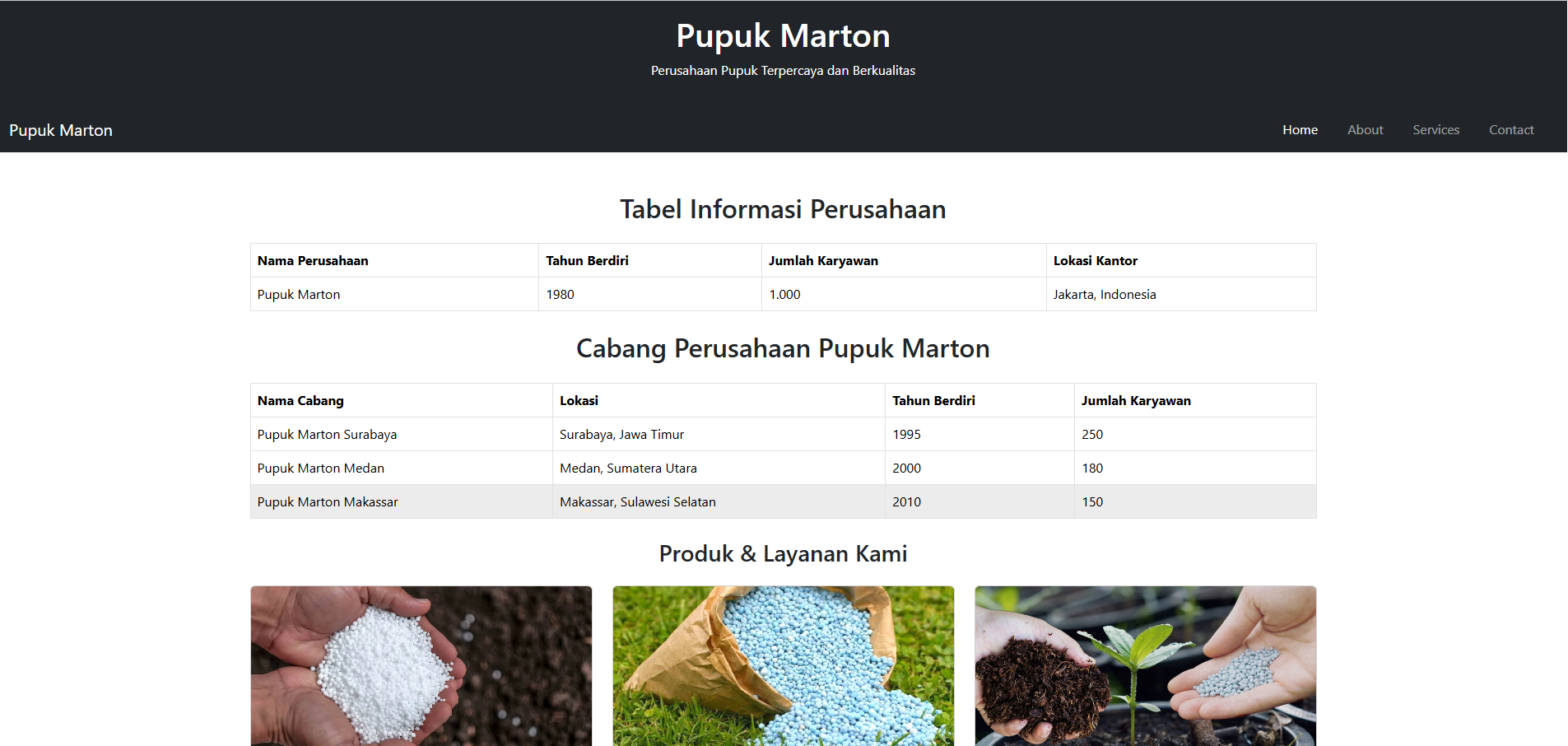
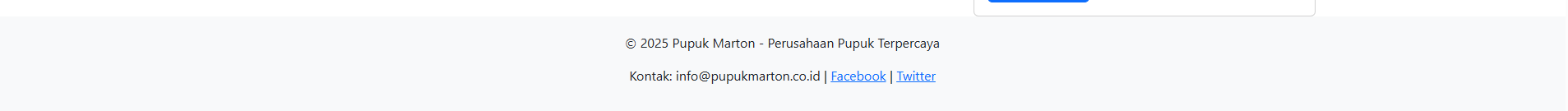


Jawab:

Kode :

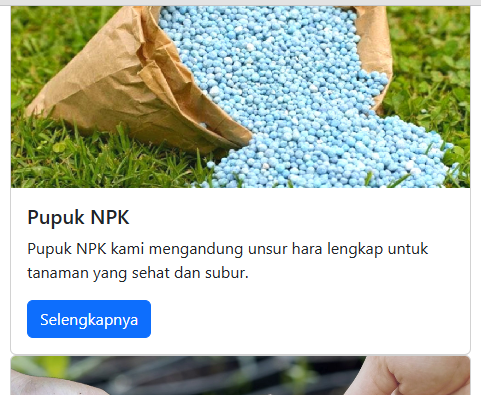
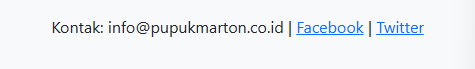
|  |
| --- |
| <!DOCTYPE html>  <html lang="id">  <head>  <title>Company Profile - Pupuk Marton</title>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">  <style>  .card-img-top {  height: 200px;  object-fit: cover;  }  .card:hover {  box-shadow: 0 0 15px rgba(0, 0, 0, 0.1);  transform: translateY(-5px);  transition: transform 0.3s, box-shadow 0.3s;  }  footer {  background-color: #f8f9fa;  padding: 20px;  text-align: center;  }  .navbar-toggler-icon {  background-color: black;  }  .navbar-nav li {  margin-right: 20px;  }  /\* Hover effect for table rows \*/  .table-hover tbody tr:hover {  background-color: #f1f1f1;  }  </style>  </head>  <body>  <!-- Header -->  <header class="bg-dark text-white text-center py-3">  <h1>Pupuk Marton</h1>  <p>Perusahaan Pupuk Terpercaya dan Berkualitas</p>  </header>  <!-- Navbar -->  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">  <div class="container-fluid">  <a class="navbar-brand" href="#">Pupuk Marton</a>  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"  aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">  <span class="navbar-toggler-icon"></span>  </button>  <div class="collapse navbar-collapse" id="navbarNav">  <ul class="navbar-nav ms-auto">  <li class="nav-item">  <a class="nav-link active" href="#">Home</a>  </li>  <li class="nav-item">  <a class="nav-link" href="#">About</a>  </li>  <li class="nav-item">  <a class="nav-link" href="#">Services</a>  </li>  <li class="nav-item">  <a class="nav-link" href="#">Contact</a>  </li>  </ul>  </div>  </div>  </nav>  <!-- Konten Utama -->  <div class="container mt-5">  <h2 class="text-center mb-4">Tabel Informasi Perusahaan</h2>  <!-- Tabel Informasi Perusahaan -->  <table class="table table-bordered">  <thead>  <tr>  <th>Nama Perusahaan</th>  <th>Tahun Berdiri</th>  <th>Jumlah Karyawan</th>  <th>Lokasi Kantor</th>  </tr>  </thead>  <tbody>  <tr>  <td>Pupuk Marton</td>  <td>1980</td>  <td>1.000</td>  <td>Jakarta, Indonesia</td>  </tr>  </tbody>  </table>  <!-- Tabel Cabang Perusahaan (Hoverable) -->  <h2 class="text-center my-4">Cabang Perusahaan Pupuk Marton</h2>  <table class="table table-hover table-bordered">  <thead>  <tr>  <th>Nama Cabang</th>  <th>Lokasi</th>  <th>Tahun Berdiri</th>  <th>Jumlah Karyawan</th>  </tr>  </thead>  <tbody>  <tr>  <td>Pupuk Marton Surabaya</td>  <td>Surabaya, Jawa Timur</td>  <td>1995</td>  <td>250</td>  </tr>  <tr>  <td>Pupuk Marton Medan</td>  <td>Medan, Sumatera Utara</td>  <td>2000</td>  <td>180</td>  </tr>  <tr>  <td>Pupuk Marton Makassar</td>  <td>Makassar, Sulawesi Selatan</td>  <td>2010</td>  <td>150</td>  </tr>  </tbody>  </table>  <!-- Card Produk/Layanan -->  <h3 class="text-center my-4">Produk & Layanan Kami</h3>  <div class="row">  <div class="col-md-4">  <div class="card">  <img src="Urea.png" class="card-img-top" alt="Pupuk Urea">  <div class="card-body">  <h5 class="card-title">Pupuk Urea</h5>  <p class="card-text">Pupuk Urea kami sangat efektif untuk meningkatkan hasil pertanian dan kesuburan tanah.</p>  <a href="#" class="btn btn-primary">Selengkapnya</a>  </div>  </div>  </div>  <div class="col-md-4">  <div class="card">  <img src="NPK.png" class="card-img-top" alt="Pupuk NPK">  <div class="card-body">  <h5 class="card-title">Pupuk NPK</h5>  <p class="card-text">Pupuk NPK kami mengandung unsur hara lengkap untuk tanaman yang sehat dan subur.</p>  <a href="#" class="btn btn-primary">Selengkapnya</a>  </div>  </div>  </div>  <div class="col-md-4">  <div class="card">  <img src="organik.png" class="card-img-top" alt="Pupuk Organik">  <div class="card-body">  <h5 class="card-title">Pupuk Organik</h5>  <p class="card-text">Pupuk Organik kami membantu meningkatkan kualitas tanah secara alami tanpa bahan kimia berbahaya.</p>  <a href="#" class="btn btn-primary">Selengkapnya</a>  </div>  </div>  </div>  </div>  </div>  <!-- Footer -->  <footer>  <p>&copy; 2025 Pupuk Marton - Perusahaan Pupuk Terpercaya</p>  <p>Kontak: info@pupukmarton.co.id | <a href="#">Facebook</a> | <a href="#">Twitter</a></p>  </footer>  <!-- Script Bootstrap -->  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>  </body>  </html> |

Tampilan jika dijalankan di web browser dengan tampilan lebar:

Tampilan di layar tidak lebar:



Berikut link github video penjelasan video: [github](https://github.com/Martonius3/laporan-praktikum-jaringan-komputer/blob/main/202312051%20Martonius%20Video%20Bootsrap.mp4)

**Modul 4. Javascript**

## A. Tujuan

Setelah mengikuti praktikum, Mahasiswa diharapkan mampu:

1. Mampu memahami konsep dasar dan manipulasi DOM dengan JavaScript.

2. Mampu menangani interaksi pengguna menggunaka *event handling.*

3. Dapat menerapkan logika percabangan dan perulangan untuk membuat halaman web dinamis.

4. Mampu membuat validasi form dan kalkulasi sederhana secara *real-time.*

5. Bisa bekerja dengan objek-objek bawaan JavaScript seperti Array, String, Math

dan Date.

6. Dapat memanipulasi elemen gambar secara dinamis.

## B. Teori Dasar Bootsrap

JavaScript adalah bahasa pemrograman yang membuat halaman web menjadi hidup dan interaktif. Berbeda dengan HTML yang hanya menyusun struktur dan CSS yang mengatur tampilan, JavaScript dapat mengubah konten HTML dan CSS secara langsung di browser tanpa perlu memuat ulang halaman. Kemampuan ini didasarkan pada pilar-pilar utama seperti Manipulasi DOM, Event Handling, dan penerapan Logika Pemrograman. Selain itu, JavaScript menyediakan berbagai Objek Bawaan (Built-in Objects) yang sangat kuat untuk memanipulasi data seperti teks (String), angka (Math), dan waktu (Date).

## C. Pratikum

**1. Manipulasi DOM (Document Object Model)**

a. Penjelasan :

DOM adalah representasi dari dokumen HTML dalam bentuk objek. Dengan Javascrpit, kita bisa berinteraksi dengan objek ini untuk memanipulasi selemen.. Cara paling umum adalah memilih elemen berdasarkan atribut id nya menggunakan document.getElementById(), lalu mengubah konten di dalamnya dengan properti .innerHTML atau mengubah gayanya dengan properti .style.

b. Contoh Kode

File: manipulasi\_dom.html





c. Latihan Pratikum

* + Praktikkan contoh kode di atas.
  + Berikan capture (tangkapan layar) sebelum dan sesudah tombol "Ubah Teks Judul" diklik  
    .
  + Modifikasi fungsi ubahTeks() untuk menambahkan perubahan lain, misalnya mengubah ukuran (elemenJudul.style.fontSize = "36px";). Font



Dengan code:

|  |
| --- |
| function ubahTeks() {  var elemenJudul = document.getElementById("judul");  // Mengubah isi teks  elemenJudul.innerHTML = "Teks Berhasil Diubah!";  // Mengubah warna teks  elemenJudul.style.color = "crimson";  // Mengubah ukuran font  elemenJudul.style.fontSize = "36px";  // Menambahkan huruf tebal  elemenJudul.style.fontWeight = "bold";  // Mengubah jenis font  elemenJudul.style.fontFamily = "Georgia, serif";  } |

* + Analisis dari hasil praktikum anda di laporan.

Pada praktikum kali ini, kami mempelajari cara memanipulasi elemen HTML secara dinamis menggunakan JavaScript melalui DOM (Document Object Model). Fokus utama adalah pada penggunaan getElementById untuk mengakses elemen tertentu, serta metode innerHTML dan style untuk mengubah isi dan tampilannya.

Setelah fungsi ubahTeks() dimodifikasi, terlihat bahwa perubahan tidak hanya terbatas pada isi teks, tetapi juga mencakup gaya visual seperti warna, ukuran, ketebalan huruf, dan jenis font. Hal ini membuktikan bahwa JavaScript sangat fleksibel dalam mengontrol elemen halaman web secara real-time tanpa harus me-*reload* halaman.

Dengan latihan ini, kami memahami bahwa:

* DOM memungkinkan interaksi dinamis dengan elemen HTML.
* JavaScript dapat mengubah konten, atribut, dan gaya CSS secara langsung.
* Perubahan kecil pada DOM dapat memberikan pengaruh besar terhadap tampilan dan pengalaman pengguna.

**Kesimpulan:** Praktikum ini berhasil menunjukkan penerapan dasar manipulasi DOM dan membuka wawasan mengenai pengembangan antarmuka web yang lebih interaktif.

**2. Penanganan Kejadian (*Event Handling*)**

a. Penjelaasn :

Event adalah aksi yang terjadi di halaman web. JavaScript dapat “mendengarkan” event ini dan menjalankan fungsi tertentu ketika terjadi. Event yang umum digunakan adalah onclick, onmouseover, onchange, dan onfocus.

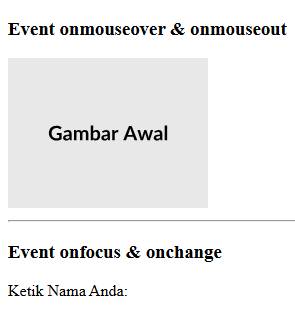
b. Contoh Kode

File: event\_handling.html





c. Latihan Pratikum

* Pratikkan contoh kode di atas.
* Berikan *capture* yang menunjukkan tampilan saat mouse di atas gambar dan setelah Anda mengetik di input  
  berikut: 

Hasil modifikasi:



Dengan code:

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>Latihan Event Handling</title>  <style>  #inputNama {  border: 2px solid #ccc;  padding: 8px;  transition: border-color 0.3s;  }  #inputNama:focus {  border-color: dodgerblue;  }  /\* Gaya dasar gambar \*/  #gambar {  transition: all 0.4s ease;  border-radius: 8px;  }  /\* Gaya saat mouse berada di atas gambar \*/  #gambar.hovered {  transform: scale(1.05);  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.3);  }  </style>  </head>  <body>  <!-- Bagian 1: Event onmouseover dan onmouseout -->  <h3>Event onmouseover & onmouseout</h3>  <img  id="gambar"  src="https://placehold.co/200x150/e8e8e8/000?text=Gambar+Awal"  alt="Gambar"  onmouseover="gantiGambar()"  onmouseout="kembalikanGambar()"  >  <hr>  <!-- Bagian 2: Event onfocus dan onchange -->  <h3>Event onfocus & onchange</h3>  <label for="inputNama">Ketik Nama Anda:</label>  <input type="text" id="inputNama" onchange="tampilkanOutput()">  <p>Output: <span id="hasil"></span></p>  <script>  function gantiGambar() {  const gambar = document.getElementById("gambar");  gambar.src = "https://placehold.co/200x150/3498db/fff?text=Gambar+Kedua";  gambar.classList.add("hovered");  }  function kembalikanGambar() {  const gambar = document.getElementById("gambar");  gambar.src = "https://placehold.co/200x150/e8e8e8/000?text=Gambar+Awal";  gambar.classList.remove("hovered");  }  function tampilkanOutput() {  const nilaiInput = document.getElementById("inputNama").value;  document.getElementById("hasil").innerHTML = "<b>" + nilaiInput + "</b>";  }  </script>  </body>  </html> |

* Analisis dari hasil pratikum Anda di laporan.  
  Latihan ini menunjukkan penerapan **event handling** JavaScript (onmouseover, onmouseout, onchange) untuk memanipulasi elemen HTML secara interaktif. Saat kursor diarahkan ke gambar, tampilannya berubah melalui **pergantian gambar dan efek CSS** seperti bayangan dan perbesaran. Input teks juga merespons perubahan dengan menampilkan hasil secara langsung.

Dengan menggabungkan **JavaScript dan CSS**, pengalaman pengguna menjadi lebih menarik dan dinamis. Praktikum ini berhasil membuktikan pentingnya DOM dan event dalam pengembangan web yang interaktif dan responsif.

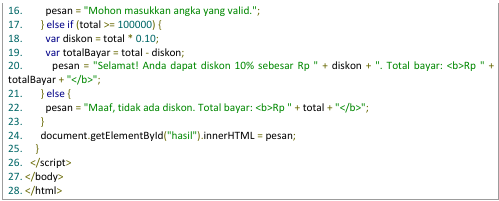
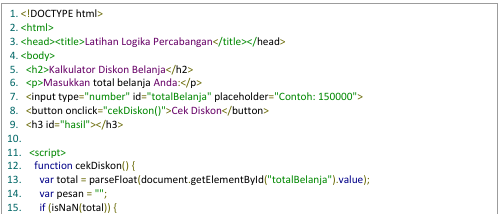
**3. Logika Percabangan (Studi Kasus: Cek Diskon Belanja)**

a. Penjelasan

Struktur kontrol if...else memungkinkan program untuk menjalankan blok kode yang berbeda bedasarkan kondisi tertentu. Ini adalah dasar dari pengambilan keputusan dalam pemrograman

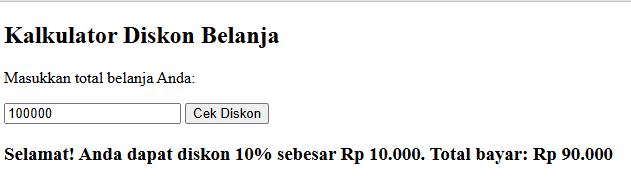
b. Contoh Kode

File: logika percabangan.html

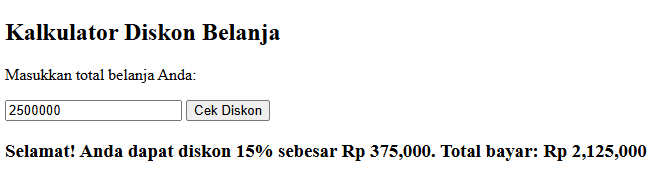


c. Latihan Pratikum

* Pratikkan contoh kode di atas.
* Lakukan dua pengujian belanja di atas dan di bawah 100000. Berikan *capture*-nya



* Modifikasi kode untuk menambahkan diskon 15% jika belanja di atas 250000



* Analisis dari hasil pratikum Anda di laporan

Analisa : Logika percabangan dalam JavaScript sangat efektif untuk membuat keputusan dinamis berdasarkan kondisi input. Latihan ini berhasil menunjukkan bagaimana struktur kontrol dapat digunakan untuk menyederhanakan proses perhitungan dan meningkatkan pengalaman pengguna dalam aplikasi web sederhana.

**4. Perulangan untuk membuat daftar**

a. Penjelasan

Perulangan (loop) digunakan untuk menjalankan kode yang sama berulang kali. Perulangan for sangat efektif untuk mengelolah data yang ada di dalam array dan menampilkannya sebagai daftar (list) di HTML.

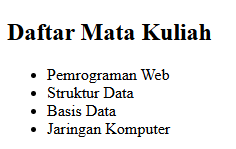
b. Contoh Kode

File: perulangan\_daftar.html



c. Latihan Pratikum

* Pratikkan contoh di atas. Jawab:



* Ganti isi dari array matkul dengan daftar nama teman sekelas Anda (minimal 5 nama). Amati perubahannya dan beirkan *capture*.



* Analisis dari hasil pratikum anda di laporan.

Berikut analisanya: Tujuan

Memahami bagaimana JavaScript menggunakan perulangan (for) untuk mengakses dan menampilkan isi array ke dalam elemen HTML.

⚙️ Proses

Data array berisi daftar teman sekelas.

Melalui for loop, setiap elemen array ditambahkan sebagai item <li> dalam daftar <ul>.

Tampilan halaman berubah secara otomatis sesuai isi array.

📈 Hasil

Halaman berhasil menampilkan seluruh nama yang ada di dalam array.

Perubahan isi array secara langsung memengaruhi konten yang ditampilkan tanpa perlu mengubah struktur HTML.

✅ Kesimpulan

Latihan ini menunjukkan bahwa JavaScript dapat secara dinamis menampilkan data dalam jumlah berapa pun menggunakan perulangan. Praktikum ini juga memperkuat pemahaman bahwa DOM dapat dimodifikasi dengan script, menjadikannya dasar penting untuk membangun aplikasi web interaktif.

**5. Validasi Form Sederhana**

a. Penjelasan

Validasi memastikan pengguna memasukan data yang benar sebelum form diproses. Kita bisa menggunakan if untuk memeriksa apakah sebuah input kosong (“”) dan memberikan peringatan kepada pengguna.

b. Contoh Kode

File: validasi\_form.html

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>Latihan Validasi Form</title>  <style>  .error {  color: red;  font-size: 0.9em;  }  </style>  </head>  <body>  <h2>Form Registrasi</h2>  <form onsubmit="return validasi()">  <label for="nama">Nama Lengkap:</label><br>  <input type="text" id="nama" name="nama"><br>  <small id="nama-error" class="error"></small><br><br>  <input type="submit" value="Daftar">  </form>  <script>  function validasi() {  var inputNama = document.getElementById("nama").value;  var errorElement = document.getElementById("nama-error");  if (inputNama == "") {  errorElement.innerHTML = "Nama tidak boleh kosong!";  return false; // Mencegah form dikirim  }  errorElement.innerHTML = ""; // Menghapus pesan error jika valid  alert("Registrasi berhasil!");  return true;  }  </script>  </body>  </html> |

C. Latihan Pratikum

* Pratikkan contoh di atas. Coba klik tombol “Daftar” saat input masih kosong dan berikan *capture*. Berikut hasilnya:



* Tambahkan satu input lagi untuk “Email” dan buat validasi untuk memastikan email tidak kosong. Berikut kodenya:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>Latihan Validasi Form</title>  <style>  .error {  color: red;  font-size: 0.9em;  }  </style>  </head>  <body>  <h2>Form Registrasi</h2>  <form onsubmit="return validasi()">  <!-- Input Nama -->  <label for="nama">Nama Lengkap:</label><br>  <input type="text" id="nama" name="nama"><br>  <small id="nama-error" class="error"></small><br><br>  <!-- Input Email -->  <label for="email">Email:</label><br>  <input type="text" id="email" name="email"><br>  <small id="email-error" class="error"></small><br><br>  <!-- Tombol Submit -->  <input type="submit" value="Daftar">  </form>  <script>  function validasi() {  var inputNama = document.getElementById("nama").value;  var inputEmail = document.getElementById("email").value;  var errorNama = document.getElementById("nama-error");  var errorEmail = document.getElementById("email-error");  var valid = true;  // Validasi Nama  if (inputNama.trim() === "") {  errorNama.innerHTML = "Nama tidak boleh kosong!";  valid = false;  } else {  errorNama.innerHTML = "";  }  // Validasi Email  if (inputEmail.trim() === "") {  errorEmail.innerHTML = "Email tidak boleh kosong!";  valid = false;  } else {  errorEmail.innerHTML = "";  }  // Jika valid, munculkan alert  if (valid) {  alert("Registrasi berhasil!");  }  return valid; // Return true jika semua input valid  }  </script>  </body>  </html> |



* Analisis dari hasil pratikum anda di laporan. Analisa:

**🎯 Tujuan**

Praktikum ini bertujuan untuk mengimplementasikan validasi form menggunakan JavaScript agar pengguna tidak dapat mengirimkan form tanpa mengisi **nama** dan **email**.

**⚙️ Hasil Implementasi**

Setelah input email ditambahkan dan divalidasi, sistem form kini:

* Menolak pengiriman jika **nama atau email kosong**.
* Memberi **pesan kesalahan langsung di bawah input** (dengan teks merah).
* Menampilkan **alert “Registrasi berhasil!”** hanya jika semua input telah terisi.

**📈 Hasil Pengujian**

| **Input Nama** | **Input Email** | **Output** |
| --- | --- | --- |
| Kosong | Kosong | Dua pesan error tampil, form tidak terkirim ❌ |
| Diisi | Kosong | Pesan error hanya untuk email ❌ |
| Kosong | Diisi | Pesan error hanya untuk nama ❌ |
| Diisi | Diisi | Alert muncul, form terkirim ✅ |

**6. Bekerja dengan Array (Aplikasi To-Do List)**

a. Penjelasan

Array adalah variabel yang bisa menampung banyak nilai. Ini sangat berguna untuk mengelola daftar data. Dalam latihan ini kita akan membuat aplikasi “To-Do List” di mana pengguna bisa menambahkan item baru ke dalam daftar.

b. Contoh Kode

File: todo\_list.html



C. Latihan Pratikum

* Pratikkan kode di atas. Coba tambahkan 3-4 tugas dan berikan *capture* hasilnya. Berikut: 
* Tantangan: Tambahkan tombol “Hapus” di sebelah setiap item daftar yang bisa menghapus item tersebut dari arry dan tampilan. (Petunjuk:gunakan array.splice(indeks,1))

Berikut screenshot: 

Dengan code:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>Aplikasi To-Do List</title>  <style>  li {  margin-bottom: 5px;  }  button.hapus-btn {  margin-left: 10px;  color: white;  background-color: red;  border: none;  padding: 3px 8px;  cursor: pointer;  }  </style>  </head>  <body>  <h2>To-Do List Sederhana</h2>  <input type="text" id="input-todo" placeholder="Ketik tugas baru...">  <button onclick="tambahTugas()">Tambah</button>  <ul id="daftar-tugas"></ul>  <script>  var tugas = [];  function tambahTugas() {  var inputElement = document.getElementById("input-todo");  var isiTugas = inputElement.value.trim();  if (isiTugas !== "") {  tugas.push(isiTugas);  inputElement.value = "";  tampilkanDaftar();  } else {  alert("Tugas tidak boleh kosong!");  }  }  function hapusTugas(index) {  tugas.splice(index, 1); // Menghapus item dari array  tampilkanDaftar(); // Menampilkan ulang daftar yang sudah diperbarui  }  function tampilkanDaftar() {  var daftarElement = document.getElementById("daftar-tugas");  daftarElement.innerHTML = "";  for (let i = 0; i < tugas.length; i++) {  daftarElement.innerHTML +=  "<li>" + tugas[i] +  " <button class='hapus-btn' onclick='hapusTugas(" + i + ")'>Hapus</button></li>";  }  }  </script>  </body>  </html> |

Analisa:

**📊 Analisis Hasil Praktikum – Versi dengan Tombol Hapus**

**🎯 Tujuan**

Menambahkan fitur interaktif untuk menghapus item tertentu dari daftar tugas yang ditampilkan, serta dari array tugas.

**⚙️ Cara Kerja**

* Saat tombol **Hapus** diklik, fungsi hapusTugas(index) dijalankan.
* Fungsi tersebut akan menghapus elemen array pada posisi index menggunakan array.splice(index, 1).
* Daftar diperbarui secara dinamis menggunakan tampilkanDaftar().

**✅ Hasil Pengujian**

| **Aksi** | **Hasil** |
| --- | --- |
| Menambah tugas | Tugas ditampilkan di daftar ✅ |
| Klik tombol **Hapus** | Tugas dihapus dari array & tampilan ✅ |
| Klik "Tambah" dengan kosong | Alert: “Tugas tidak boleh kosong!” ❌ |

**📌 Kesimpulan**

* Praktikum ini menunjukkan bahwa JavaScript dapat memanipulasi **data dan tampilan secara dinamis** menggunakan array, perulangan, dan event handling.
* Dengan menggunakan splice() dan perulangan, aplikasi menjadi lebih **interaktif dan modular**.
* Ini menjadi fondasi penting dalam pembuatan aplikasi **To-Do List modern**, seperti yang ada pada framework seperti Vue atau React.

**✅ Kesimpulan**

* Validasi menggunakan JavaScript efektif untuk **mencegah input kosong** dan memastikan kelengkapan data sebelum dikirim ke server.
* Memberi umpan balik langsung kepada pengguna meningkatkan **user experience**.
* Praktikum ini memperkuat pemahaman tentang **event handling**, **DOM manipulation**, dan **struktur logika validasi** di sisi klien (*client-side validation*).

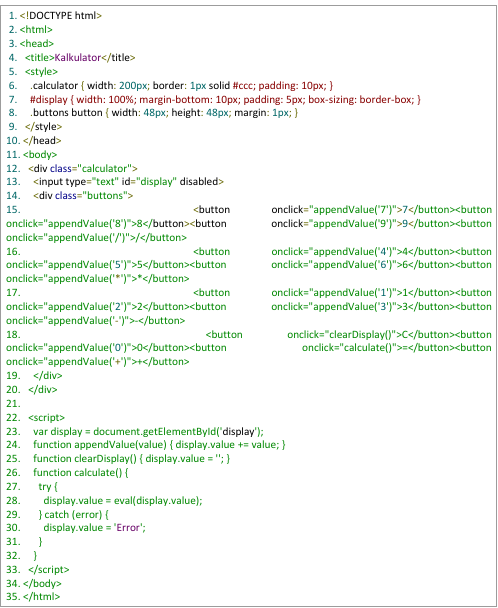
**7. Kalkulator Sederhana**

a. Penjelasan

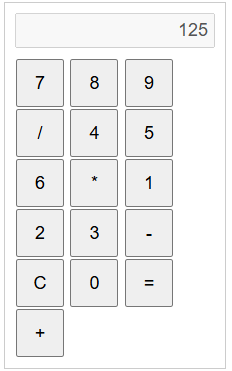
Ini adalah contoh yang menggabungkan sebua konsep: Manipulasi DOM, event handling pada banyak tombol, dan logika untuk melakukan perhittungan

b. Contoh Kode

File: Kalkulaotr.html



* Pratikan kode di atas: 5\*25 = 125



* **🎯 Tujuan Praktikum**
* Membangun aplikasi kalkulator sederhana berbasis web untuk memahami bagaimana input angka dan operator matematika dapat diproses menggunakan **JavaScript** dan fungsi **eval()**.
* **⚙️ Cara Kerja Kalkulator**
* Pengguna mengetik angka dan operator menggunakan tombol.
* Setiap tombol menambahkan karakter ke dalam elemen input dengan appendValue().
* Saat tombol = ditekan, fungsi calculate() dijalankan.
* Di dalam calculate(), ekspresi matematika dalam bentuk string (misalnya "2+3\*4") dievaluasi menggunakan fungsi eval() dan hasilnya ditampilkan ke layar.
* Jika terjadi kesalahan (input tidak valid), akan ditampilkan pesan Error menggunakan try-catch.
* **🔍 Penjelasan Fungsi eval()**
* Fungsi **eval()** dalam JavaScript digunakan untuk **mengeksekusi string sebagai kode JavaScript**. Dalam konteks kalkulator ini:
* javascript
* CopyEdit
* eval("2 + 3 \* 4") // → 14
* Fungsi eval():
* Membaca string "2 + 3 \* 4".
* Menjalankan ekspresi sebagai **perintah JavaScript**.
* Mengembalikan hasil evaluasinya ke variabel display.value.
* Keunggulan:
* Mudah digunakan untuk kalkulasi ekspresi matematika secara dinamis.
* Kekurangan:
* Kurang aman jika input berasal dari pengguna secara langsung, karena eval() dapat mengeksekusi kode JavaScript berbahaya (misalnya: eval("alert('hacked')")).
* Tidak dianjurkan untuk digunakan dalam aplikasi produksi tanpa kontrol validasi ketat.
* **✅ Hasil Praktikum**
* Kalkulator berfungsi dengan baik dalam menangani input valid dan memberikan hasil yang akurat.
* Saat pengguna melakukan kesalahan input (misalnya 5++3), fungsi try-catch menangkap error dan menampilkan "Error" di layar.
* Fungsi eval() memudahkan evaluasi ekspresi matematika sederhana secara langsung.
* **📌 Kesimpulan**
* Praktikum ini menunjukkan bagaimana eval() dapat dimanfaatkan untuk menyelesaikan kalkulasi secara instan dari string ekspresi matematika. Namun, **keamanan menjadi pertimbangan penting** saat menggunakan eval(), terutama jika aplikasi dikembangkan lebih lanjut untuk pengguna umum. Disarankan menggunakan parser matematika khusus seperti **math.js** untuk implementasi yang lebih aman dan fleksibel.

**8. Bekerja dengan Objek String**

a. Penjelasan

Onjek string menyediakan banyak metode (method) untuk memanipulasi teks. Contohnya .toUpperCase() untuk mengubah menjadi huruf kapital, .length untuk menghitung jumlah karakter, dan .substring() untuk mengambila sebagian teks.

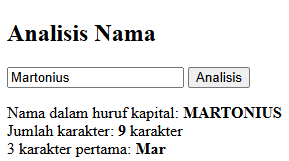
b. Contoh Kode

File: objek\_string.html



C. Latihan Pratikum

* Pratikkan kode di atas, masukan nama lengkap Anda dan klik tombol “Analisis”
* Berikan *capture* hasilnya.



* Analisis dari hasil pratikkum Anda di laporan.

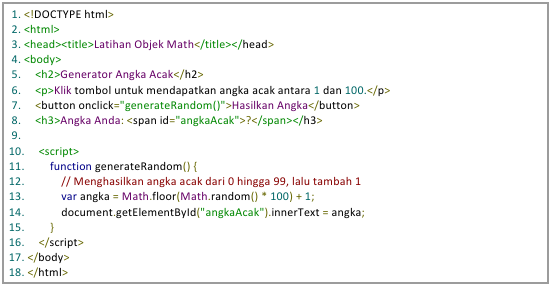
Praktikum berhasil menunjukkan bahwa metode toUpperCase(), length, dan substring() pada objek String dapat digunakan untuk menganalisis data teks secara efisien. Input "Martonius" dapat diubah ke huruf kapital, dihitung panjangnya, dan diekstrak 3 huruf awalnya. Ini membuktikan bahwa manipulasi string di JavaScript sangat berguna untuk validasi dan analisis data pengguna.

**9. Menggunakan Objek Math**

a. Penjelasan

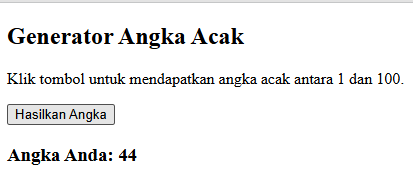
Objek Math berisi properti dan metode untuk konstanta dan fungsi matematika. Math.random() menghasilkan angka acak antara 0 dan 1, sementara Math.floor() membulatkan angka ke bawah ke bilangan bulat terdekat. Kombinasi keduanya sangat berguna untuk menghasilkan angka acak dalam rentang tertentu.

b. Contoh Kode

File: objek\_math.html 

c. Latihan Pratikum

* Pratikkan kode di atas. Klik tombol beberapa kali untuk melihat angka yang berbeda
* Berikan *capture* yang menunjukkan salah satu angka acak yang dihasilkan.
* Analisis dari hasil pratikkum Anda di laporan



Kode ini menggunakan **JavaScript** dan **Math API** untuk menghasilkan angka acak antara 1 hingga 100.

* **Math.random()** menghasilkan angka acak antara 0 (inklusif) dan 1 (eksklusif).
* **Math.floor()** digunakan untuk mengubah angka desimal menjadi bilangan bulat.
* Dengan perkalian Math.random() \* 100, angka acak dihasilkan antara 0 hingga 99, kemudian ditambah 1 untuk memastikan angka acak berada di rentang 1 hingga 100.

**Fungsi generateRandom()** dijalankan saat tombol diklik, dan angka acak yang dihasilkan ditampilkan dalam elemen dengan ID angkaAcak.

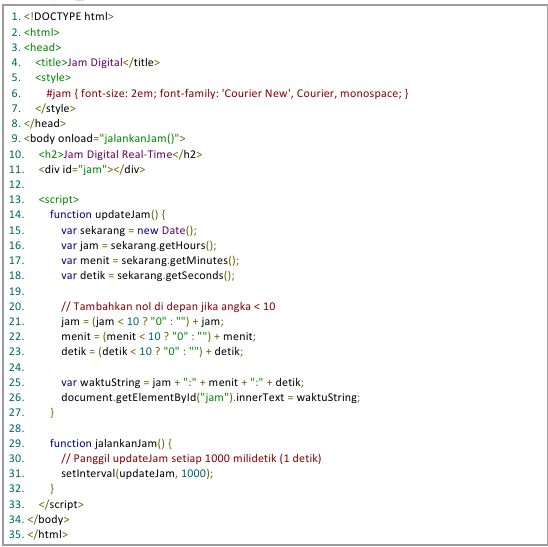
**10. Menampilkan Tanggal dan Jam Dinamis**

a. Penjelasan

Objek Date digunakan untuk bekerja dengan tanggal dan waktu. Dengan membuat instance baru (new Date()), kita bisa mendapatkan waktu saat ini dan menampilkannnya. Untuk membuat jam yang berjalan, kita perlu memanggil fungsi pembaruan waktu secara berulang menggunakan setInterval()

b. Contoh Kode

File: objek\_date.html



C. latihan pratikum

* Pratikkan kode di atas dan amati jam yang berjalan.
* Berikan *capture* halaman tersebut

Berikut: 

* Analisis dari hasil pratikum Anda di laporan, jelaskan fungsi dari setInterval()

**Analisis Kode – Jam Digital Real-Time**

* Kode ini membuat aplikasi jam digital yang memperbarui waktu setiap detik.
* Fungsi **updateJam()** mengambil waktu saat ini menggunakan objek **Date()**, kemudian memformat jam, menit, dan detik untuk selalu menampilkan dua digit (misalnya, 09:03:05).
* Fungsi **setInterval()** digunakan untuk memanggil **updateJam()** setiap 1000 milidetik (1 detik), sehingga tampilan waktu selalu diperbarui secara real-time.

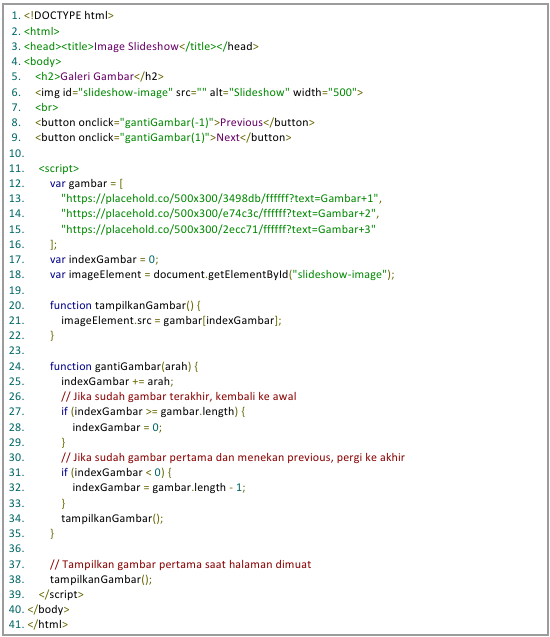
Praktikum ini menunjukkan cara sederhana untuk menggunakan **JavaScript** dalam membuat aplikasi berbasis waktu.

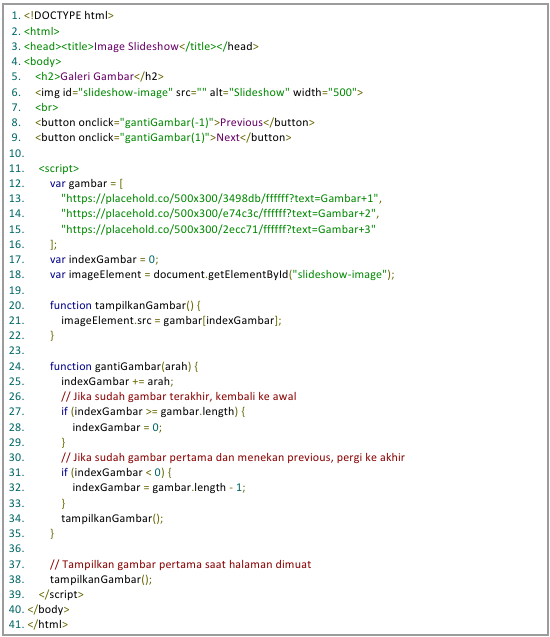
**11. Membuat Image Slideshow Sederhana**

a. Penjelasan

Manipulasi gambar adalah salah satu aplikasi DOM yang paling visual. Dengan mengubah properti .src dari sebuah elemen **Kesalahan! Nama file tidak ditentukan.**, kita dapat membuat slideshow. Kita akan menyimpan URL gambar dalam sebuah array dan menggunakan sebuah counter untuk beralih antar gambar.

b. Contoh Kode

File: Slideshow.html



C. Latihan Pratikum

* Pratikkan kode di atas. Coba Klik tombol “Next” dan “Previous”.



* Tambahkan dua URL gambar baru ke dalam *array* gambar.

var gambar = [

"https://placehold.co/500x300/3498db/ffffff?text=Gambar+1",

"https://placehold.co/500x300/e74c3c/ffffff?text=Gambar+2",

"https://placehold.co/500x300/2ecc71/ffffff?text=Gambar+3",

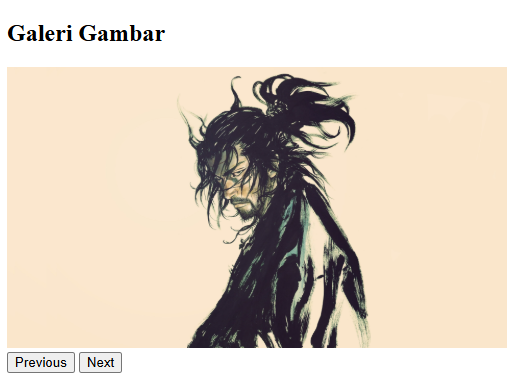
"1.jpg",

"2.jpg",

"3.png"

];

* Berikan *capture* salah satu gambar yang tampil dalam slideshow



* Analisis dari hasil pratikum Anda di laporan

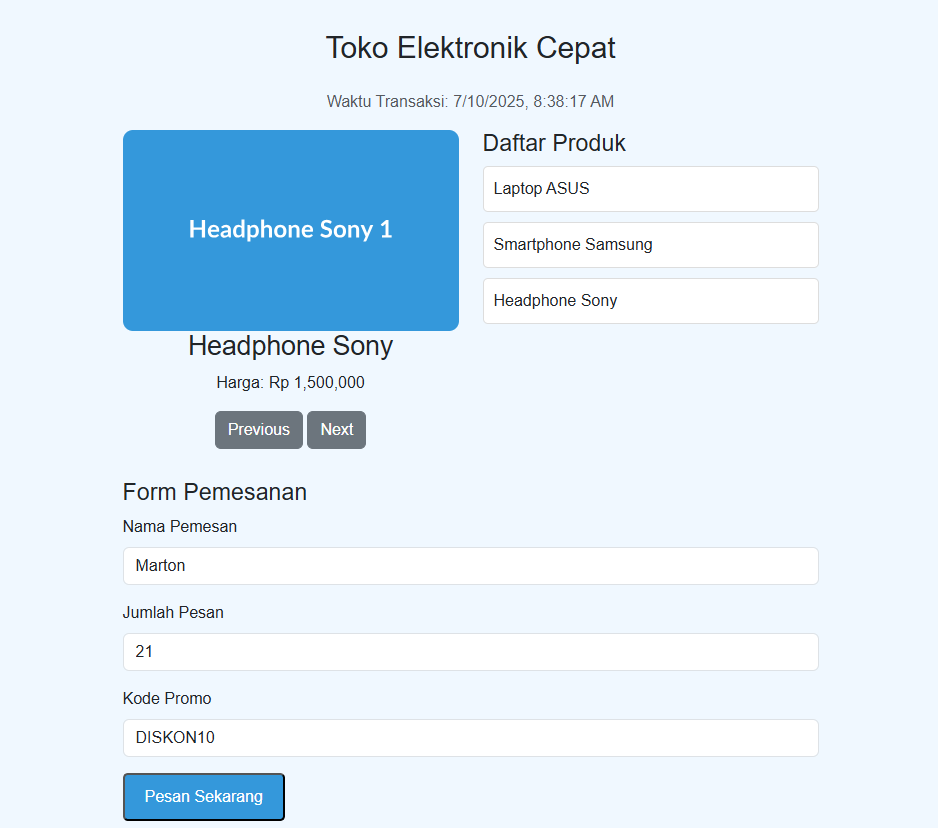
Kode ini berfungsi untuk menampilkan **image slideshow** yang memungkinkan pengguna untuk mengganti gambar menggunakan tombol "Next" dan "Previous".

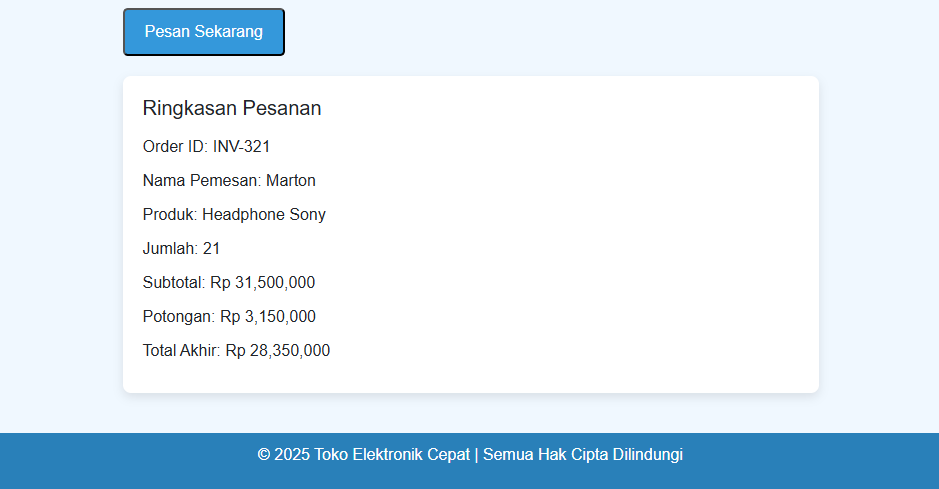
**Fungsi utama:**

1. **Gambar lokal** digunakan dengan mengubah URL gambar dalam array gambar ke jalur file lokal (misalnya, images/gambar1.jpg).
2. **gantiGambar(arah)** mengubah gambar berdasarkan arah yang diberikan (tombol "Next" atau "Previous").
3. **tampilkanGambar()** memperbarui elemen <img> dengan gambar yang sesuai berdasarkan **index** dalam array.

Dengan **jalur file lokal**, gambar ditampilkan langsung dari folder lokal, memungkinkan pengelolaan lebih mudah pada aplikasi berbasis file lokal tanpa bergantung pada URL eksternal.

## D. Tugas Akhir Praktikum: Mini E-Comerce Product Page





Vidio penjelasan web mini e-comerce : [link](file:///D:\Marton\Kuliah%20S1%20Teknik%20Informatika\Semester%204\PEMROGRAMAN%20WEB\Modul%204\video1777814522.mp4)

Dengan Code:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Latihan Logika Percabangan</title>  </head>  <body>  <h2>Kalkulator Diskon Belanja</h2>  <p>Masukkan total belanja Anda:</p>  <input type="number" id="totalBelanja" placeholder="Contoh: 150000">  <button onclick="cekDiskon()">Cek Diskon</button>  <h3 id="hasil"></h3>  <script>  function cekDiskon() {  var total = parseFloat(document.getElementById("totalBelanja").value);  var pesan = "";  if (isNaN(total)) {  pesan = "Mohon masukkan angka yang valid.";  } else if (total >= 100000) {  var diskon = total \* 0.15; // Ubah dari 0.10 menjadi 0.15 (diskon 15%)  var totalBayar = total - diskon;  pesan = "Selamat! Anda dapat diskon 15% sebesar Rp " + diskon.toLocaleString() +  ". Total bayar: <b>Rp " + totalBayar.toLocaleString() + "</b>";  } else {  pesan = "Maaf, tidak ada diskon. Total bayar: <b>Rp " + total.toLocaleString() + "</b>";  }  document.getElementById("hasil").innerHTML = pesan;  }  </script>  </body>  </html> |

**Modul 5. Dasar-dasar PHP**

## A. Tujuan

Setelah mengikuti praktikum, Mahasiswa diharapkan mampu:

1. Mampu memahami konsep dasar dan sintaks bahasa pemrograman PHP.

2. Dapat membuat program dengan PHP untuk menghasilkan halaman web yang dinamis

3. Mampu mengolah data dari input form dan variabel superglobals.

## B. Teori Dasar PHP

File PHP adalah dokumen teks dengan ekstensi .php yang dapat berisi campuran kode HTML, CSS, JavaScript, dan skrip PHP. Skrip PHP dieksekusi di server, dan hasilnya dikirim ke browser sebagai HTML biasa.

PHP dapat digunakan untuk membuat halaman web yang dinamis, mengelola file di server, mengumpulkan data dari form, mengatur cookies dan sesi, serta berinteraksi dengan berbagai jenis database.

PHP populer karena bersifat open-source (gratis), berjalan di berbagai platform (Windows, Linux, macOS), kompatibel dengan hampir semua server web, dan didukung oleh komunitas yang besar serta dokumentasi yang melimpah.

Untuk pengembangan PHP di komputer lokal, Anda memerlukan lingkungan server yang terdiri dari Web Server (seperti Apache), PHP, dan Database (seperti MySQL/MariaDB).

Cara termudah adalah dengan menginstal paket perangkat lunak terintegrasi seperti XAMPP (disarankan), Laragon atau WAMP. Paket ini sudah mencakup semua komponen yang dibutuhkan dalam satu kali instalasi.

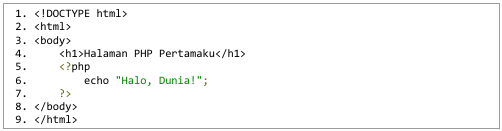
## C. Praktikum

**1. Basic syntax**

a. Penjelasan

Skrip PHP ditulis di antara <?php dan ?>. Perintah echo digunakan untuk menampilkan output. Setiap pernyataan diakhiri dengan titik koma (;).

b. Contoh kode File: syntax.php



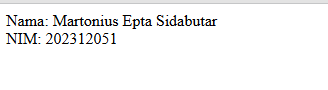
c. latihan pratikum

* + Pratikkan contoh kode di atas.

Berikut hasilnya



* + Buat file baru, lalu gunakan echo untuk menampilkan Nama dan NIM Anda pada baris yang berbeda



Dengan code .php

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <?php  // Menampilkan Nama  echo "Nama: Martonius Epta Sidabutar<br>";    // Menampilkan NIM  echo "NIM: 202312051";  ?>  </body>  </html> |

Analisis:

**Perbandingan dan Kesimpulan:**

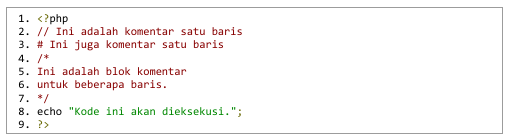
* **Integrasi PHP dengan HTML**:
  + Kedua kode ini menggunakan integrasi PHP di dalam HTML. Perbedaannya terletak pada tujuan dan isi yang ditampilkan.
  + Kode pertama menampilkan pesan statis sederhana "Halo, Dunia!" di dalam HTML, sementara kode kedua lebih berfokus pada menampilkan dua informasi terpisah (Nama dan NIM) dengan penataan sedikit lebih kompleks (menggunakan <br> untuk pemisahan baris).
* **Fungsionalitas echo**:
  + Pada kedua kode, perintah echo digunakan untuk menampilkan teks. echo adalah metode yang paling umum digunakan di PHP untuk menampilkan informasi atau output.
  + Di kode pertama, hanya satu output yang dihasilkan, sedangkan di kode kedua, ada dua output terpisah, dan satu di antaranya diberi pemisahan baris dengan tag <br>.
* **Output**:
  + Output dari kedua kode sangat berbeda dalam konteks tampilan. Kode pertama hanya menampilkan satu kalimat di halaman web, sedangkan kode kedua memisahkan dua informasi berbeda (Nama dan NIM) dengan cara yang lebih terstruktur menggunakan baris baru.
* **Praktik dan Penggunaan**:
  + Kode pertama cocok untuk aplikasi sederhana atau halaman yang hanya perlu menampilkan teks langsung dari PHP.
  + Kode kedua lebih cocok untuk aplikasi yang memerlukan tampilan terstruktur, seperti menampilkan data pengguna atau informasi lainnya yang dibutuhkan dalam baris yang berbeda.

Secara keseluruhan, kedua kode ini menunjukkan penggunaan dasar PHP untuk menampilkan teks, namun dengan pendekatan yang berbeda untuk tujuan masing-masing.

**2. PHP Coments**

a. Penjelasan

Komentar adalah catatan dalam kode yang tidak dieksekusi oleh PHP

b. Contoh kode File: comments.php ****

c. Latihan Pratikum

* + Pratikkan contoh kode:



* + Tambahkan ketiga jenis komentar pada skrip dari latihan sebelumnya untuk menjelaskan fungsinya.



Dengan code

|  |
| --- |
| <?php  // Ini adalah komentar satu baris: Komentar ini hanya mencakup satu baris  // Digunakan untuk memberikan penjelasan singkat atau catatan pada kode  echo "Halo, Dunia!"; // Menampilkan teks "Halo, Dunia!" pada halaman  # Ini juga komentar satu baris: Komentar ini juga digunakan untuk penjelasan singkat  # Biasanya digunakan di baris yang lebih sederhana  echo "Nama: Martonius<br>"; // Menampilkan "Nama: James" di halaman, kemudian pindah ke baris baru  /\*  Ini adalah blok komentar  untuk beberapa baris.  Komentar ini dapat digunakan untuk menjelaskan kode yang lebih panjang atau memberikan  penjelasan detail tentang fungsi atau blok kode tertentu.  \*/  echo "NIM: 202312051"; // Menampilkan "NIM: 1234567890" di halaman  ?> |

* + Analisis:

**Analisis Kode PHP dengan Komentar:**

1. **Komentar dalam PHP**:
   * Komentar sangat penting dalam kode untuk memberikan penjelasan atau deskripsi mengenai fungsi atau tujuan bagian kode tertentu.
   * PHP mendukung tiga jenis komentar: komentar satu baris (// atau #), dan komentar blok yang mencakup beberapa baris (/\* ... \*/).
2. **Jenis Komentar**:
   * **Komentar Satu Baris (// dan #)**:
     + Digunakan untuk memberikan penjelasan singkat tentang kode di satu baris.
     + Misalnya, // Menampilkan teks "Halo, Dunia!" memberikan penjelasan tentang baris yang berisi perintah echo.
     + # adalah bentuk lain dari komentar satu baris yang memiliki fungsi yang sama dengan //.
   * **Komentar Blok (/\* ... \*/)**:
     + Digunakan untuk memberikan penjelasan lebih panjang yang mencakup beberapa baris kode.
     + Dalam kode ini, komentar blok digunakan untuk menjelaskan fungsi dari seluruh bagian kode atau untuk memberikan dokumentasi tambahan.
     + Menjadi lebih berguna ketika kode yang perlu dijelaskan lebih kompleks atau ketika ada beberapa baris yang perlu didokumentasikan bersama-sama.
3. **Manfaat Komentar**:
   * **Pemeliharaan Kode**: Membantu developer lain atau diri kita sendiri untuk memahami maksud dan fungsi kode, terutama ketika proyek menjadi lebih besar dan kompleks.
   * **Debugging**: Komentar juga digunakan untuk menonaktifkan sementara bagian dari kode tanpa menghapusnya.
   * **Dokumentasi**: Memberikan konteks lebih jelas tentang apa yang dilakukan kode dan mengapa.
4. **Penggunaan Komentar pada Kode**:
   * Kode yang diberikan menggunakan semua jenis komentar untuk menjelaskan bagian-bagian kode dengan cara yang berbeda. Ini memberikan dokumentasi yang memadai, memudahkan pemahaman, dan membantu dalam kolaborasi tim.

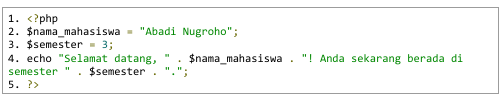
Secara keseluruhan, penggunaan komentar yang baik sangat penting dalam pengembangan perangkat lunak untuk menjaga kode tetap terorganisir, mudah dipahami, dan mudah dipelihara.

**3. PHP Variables**

a. Penjelasan

Variabel digunakan untuk menyimpan data dan diawali dengan tanda dolar ($)

b. Contoh Kode File:variables.php



* Jika kode di atas di pratikan



* Membuat dua variabel untuk menyimpan panjang dan lebar, lalu hitung dan tampilkan luasnya

Dengan skrip

|  |
| --- |
| <?php  // Mendeklarasikan variabel panjang dan lebar  $panjang = 5; // panjang dalam satuan meter  $lebar = 3; // lebar dalam satuan meter  // Menghitung luas  $luas = $panjang \* $lebar;  // Menampilkan hasil luas  echo "Luas persegi panjang dengan panjang $panjang meter dan lebar $lebar meter adalah $luas meter persegi.";  ?> |

Dan jika di buka di browser:



* Analisis:

**Analisis Kode PHP untuk Menghitung Luas Persegi Panjang**

1. **Deklarasi Variabel**:
   * **$panjang = 5;** dan **$lebar = 3;**: Kode ini mendeklarasikan dua variabel yang menyimpan nilai panjang dan lebar dari persegi panjang. Nilai-nilai ini bisa disesuaikan sesuai kebutuhan.
   * Variabel ini menggunakan tipe data numerik (integer), yang cocok untuk perhitungan matematis.
2. **Perhitungan Luas**:
   * **$luas = $panjang \* $lebar;**: Dengan rumus dasar geometri untuk menghitung luas persegi panjang (Luas = Panjang x Lebar), kode ini melakukan perhitungan dan menyimpan hasilnya dalam variabel $luas.
3. **Menampilkan Hasil**:
   * **echo** digunakan untuk menampilkan hasil perhitungan kepada pengguna. Pesan yang ditampilkan menggabungkan nilai dari variabel menggunakan string concatenation (.).
   * Output dihasilkan dalam format yang mudah dimengerti, dengan menyebutkan nilai panjang, lebar, dan hasil luas.
4. **Keuntungan Penggunaan Variabel**:
   * **Pemeliharaan**: Dengan menyimpan panjang dan lebar dalam variabel, kode menjadi lebih fleksibel dan mudah dipelihara. Anda hanya perlu mengubah nilai variabel untuk menghitung luas persegi panjang yang berbeda.
   * **Pembacaan Kode yang Lebih Jelas**: Menggunakan variabel dengan nama yang deskriptif ($panjang, $lebar, $luas) meningkatkan keterbacaan kode dan memudahkan orang lain memahami apa yang dihitung.

**Kesimpulan:**

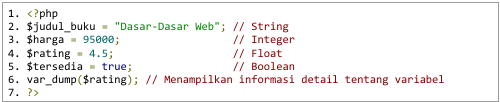
Kode ini memberikan contoh dasar bagaimana PHP dapat digunakan untuk melakukan perhitungan matematika sederhana dan menampilkan hasilnya dalam format yang ramah pengguna. Ini juga menunjukkan penggunaan variabel dalam PHP untuk menyimpan dan memanipulasi data.

**4. PHP Data Types**

a. Penjelasan

PHP Mendukung tipe data seperti String, Integer, Float, Boolean, dan Array

b. Contoh Kode File: datatypes.php



c. Latihan Pratikum

* Contoh kode di atas jika dipratikan akan mengha



* Buat variabel untuk setiap tipe data dan gunakan var\_dump() untuk memeriksa tipe dan nilainya.



Dengan code:

|  |
| --- |
| <?php  // String  $nama = "John Doe";  var\_dump($nama); // Menampilkan tipe dan nilai variabel $nama  // Integer  $umur = 25;  var\_dump($umur); // Menampilkan tipe dan nilai variabel $umur  // Float  $tinggi = 175.5;  var\_dump($tinggi); // Menampilkan tipe dan nilai variabel $tinggi  // Boolean  $aktif = true;  var\_dump($aktif); // Menampilkan tipe dan nilai variabel $aktif  // Array  $warna\_favorit = array("Merah", "Biru", "Hijau");  var\_dump($warna\_favorit); // Menampilkan tipe dan nilai variabel $warna\_favorit  // Object (contoh menggunakan kelas sederhana)  class Mobil {  public $merek = "Toyota";  public $model = "Corolla";  }  $mobil = new Mobil();  var\_dump($mobil); // Menampilkan tipe dan nilai objek $mobil  // Null  $nilai = null;  var\_dump($nilai); // Menampilkan tipe dan nilai variabel $nilai  ?> |

* Analisis:

**Analisis Pratikum: Pemeriksaan Tipe Data dengan var\_dump() di PHP**

Pratikum ini bertujuan untuk memperkenalkan berbagai tipe data dasar dalam PHP dan cara memeriksa tipe data serta nilai dari variabel menggunakan fungsi var\_dump(). Fungsi ini sangat berguna untuk debugging dan memahami struktur data dalam sebuah program. Berikut adalah analisis dari pratikum di atas:

**1. Pengenalan Tipe Data dalam PHP**

PHP mendukung beberapa tipe data dasar yang digunakan untuk menyimpan berbagai jenis nilai. Dalam pratikum ini, kita melihat berbagai tipe data yang dapat digunakan di PHP:

* **String**: Digunakan untuk menyimpan data teks. Dalam contoh ini, variabel $nama berisi nama seseorang.
* **Integer**: Digunakan untuk menyimpan angka bulat. Variabel $umur menyimpan usia dalam bentuk integer.
* **Float**: Digunakan untuk menyimpan angka desimal atau pecahan. Variabel $tinggi menyimpan tinggi badan dengan angka desimal.
* **Boolean**: Menyimpan dua nilai, yaitu true atau false. Variabel $aktif menunjukkan apakah seseorang aktif atau tidak.
* **Array**: Digunakan untuk menyimpan beberapa nilai dalam satu variabel. Variabel $warna\_favorit adalah array yang menyimpan beberapa warna favorit.
* **Object**: Digunakan untuk menyimpan data dalam bentuk objek, yang merupakan instance dari sebuah kelas. Variabel $mobil adalah objek dari kelas Mobil.
* **Null**: Tipe data ini digunakan untuk menyatakan bahwa variabel tidak memiliki nilai atau tidak terdefinisi.

**2. Penggunaan var\_dump()**

* Fungsi **var\_dump()** digunakan untuk menampilkan tipe data dan nilai variabel. Ini memberikan informasi yang lebih detail daripada hanya menggunakan echo. var\_dump() juga menunjukkan panjang string untuk tipe data string dan jumlah elemen dalam array.

**3. Keuntungan Menggunakan var\_dump()**

* **Mempermudah Debugging**: Fungsi var\_dump() sangat berguna ketika Anda bekerja dengan data yang lebih kompleks atau saat mencoba menemukan masalah dengan tipe data. Ini memungkinkan Anda untuk melihat tidak hanya nilai variabel tetapi juga tipe data yang sedang digunakan.
* **Memahami Struktur Data**: Dalam contoh array dan objek, var\_dump() memberikan gambaran jelas tentang struktur data yang lebih kompleks, misalnya jumlah elemen dalam array atau properti dalam objek.
* **Verifikasi Data**: Dengan menggunakan var\_dump(), Anda bisa memastikan bahwa tipe data yang Anda harapkan benar-benar sesuai dengan yang ada di dalam variabel, misalnya memverifikasi bahwa sebuah variabel yang seharusnya menyimpan angka tidak mengandung string.

**4. Praktik dan Aplikasi dalam Pengembangan**

* **Pengembangan Aplikasi Dinamis**: Dalam pengembangan aplikasi web menggunakan PHP, seringkali kita bekerja dengan data yang masuk dari formulir atau API. Dengan var\_dump(), Anda bisa melihat struktur data yang diterima dan memverifikasi bahwa data tersebut sesuai dengan yang diharapkan.
* **Pengelolaan Tipe Data**: Mengelola tipe data yang tepat dalam PHP sangat penting untuk menghindari kesalahan dalam perhitungan atau manipulasi data, terutama ketika data tersebut dikirim ke database atau diproses lebih lanjut.
* **Objek dan Array**: Dengan objek dan array yang lebih kompleks, var\_dump() memberi Anda gambaran yang sangat berguna untuk memeriksa nilai-nilai dalam koleksi data tersebut, memungkinkan Anda untuk memastikan bahwa data sudah terstruktur dengan benar.

**5. Kesimpulan**

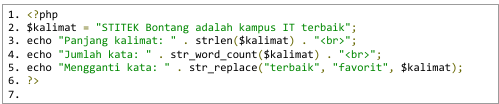
Pratikum ini menunjukkan cara menggunakan var\_dump() untuk memeriksa tipe data dan nilai variabel dalam PHP. Ini adalah keterampilan penting untuk debugging, pemahaman data, dan pemrograman yang efisien. Dengan memahami berbagai tipe data yang tersedia di PHP dan menggunakan alat seperti var\_dump(), Anda dapat menulis kode yang lebih robust dan bebas dari kesalahan terkait tipe data.

**5. PHP Strings**

a. Penjelasan

PHP Memiliki banyak fungsi bawaan untuk memanipulasi teks (string).

b. Contoh Kode File: strings.php



c. Latihan Pratikum

* + Kode di atas dipraktikkan:

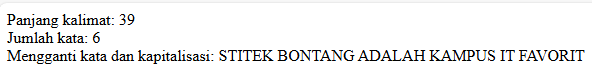


* + Gunakan fungsi strtoupper() untuk membuat semua teks menjadi huruf kapital

Berikut codenya:

|  |
| --- |
| <?php  $kalimat = "STITEK Bontang adalah kampus IT terbaik";  echo "Panjang kalimat: " . strlen($kalimat) . "<br>";  echo "Jumlah kata: " . str\_word\_count($kalimat) . "<br>";  // Mengganti kata 'terbaik' menjadi 'favorit' lalu mengubah seluruh kalimat jadi huruf kapital  $kalimatBaru = str\_replace("terbaik", "favorit", $kalimat);  echo "Mengganti kata dan kapitalisasi: " . strtoupper($kalimatBaru);  ?> |

Dan jika dipraktikkan :



* Analisis:

Kode PHP di atas melakukan tiga operasi dasar pada sebuah string:

1. **strlen($kalimat)**  
   Fungsi ini menghitung total jumlah karakter dalam string, termasuk spasi.  
   ➤ Berguna untuk validasi panjang teks atau data input.
2. **str\_word\_count($kalimat)**  
   Menghitung jumlah kata dalam string berdasarkan pemisah spasi.  
   ➤ Bermanfaat untuk analisis teks dasar seperti memeriksa kelengkapan input kalimat.
3. **str\_replace("terbaik", "favorit", $kalimat) + strtoupper()**  
   Pertama-tama mengganti kata "terbaik" dengan "favorit", lalu mengubah seluruh string menjadi huruf kapital.  
   ➤ Berguna dalam normalisasi data teks, seperti membuat heading atau data konsisten untuk keperluan pencarian/penyimpanan.

**Kelebihan:**

* Kode ringkas dan mudah dibaca.
* Menggabungkan beberapa fungsi string dasar untuk manipulasi teks yang umum dilakukan di aplikasi web.

**Kekurangan:**

* Tidak memperhitungkan sensitivitas huruf besar-kecil dalam str\_replace(), jadi hanya cocok jika input selalu dalam format yang sama.
* Tidak menangani karakter khusus atau multibahasa (non-UTF-8) jika dibutuhkan.

Secara keseluruhan, kode ini sangat cocok untuk pembelajaran dasar manipulasi string di PHP.

**6. PHP Math**

a. Penjelasan

PHP dapat melakukan operasi matematika dasar dan kompleks.

b. Contoh Kode File: math.php



c. Latihan Praktikum:

* Jika kode di atas di pratikan akan menghasilkan:



* Hitung nilai dari (10+2)/3\*5 dan tampilkan hasilnya

Berikut hasilnya:



Dengan kode:

|  |
| --- |
| <?php  echo "Hasil dari 15 % 4 adalah " . (15 % 4) . " (sisa bagi)<br>";  echo "Angka acak antara 10 dan 20: " . rand(10, 20) . "<br>";  // Menghitung nilai dari (10 + 2) / 3 \* 5  $hasil = (10 + 2) / 3 \* 5;  echo "Hasil dari (10 + 2) / 3 \* 5 adalah " . $hasil;  ?> |

* Analisis:

**Analisis Singkat Kode PHP**

Kode ini mendemonstrasikan penggunaan operator aritmatika dan fungsi built-in untuk manipulasi angka dalam PHP. Berikut rincian analisisnya:

**1. 15 % 4**

* Operator modulus (%) digunakan untuk menghitung sisa bagi dari pembagian 15 dengan 4.
* **Hasilnya adalah 3**, karena 15 dibagi 4 hasilnya 3 dengan sisa 3.
* ✅ Cocok untuk operasi logika seperti menentukan bilangan ganjil/genap, pengulangan, atau siklus rotasi.

**2. rand(10, 20)**

* Fungsi rand() menghasilkan angka acak antara dua nilai yang ditentukan (inklusif).
* **Contoh output: 13, 17, 20, dll.**
* ✅ Digunakan dalam simulasi, permainan, atau keamanan dasar (misal kode OTP).

**3. (10 + 2) / 3 \* 5**

* Evaluasi ekspresi aritmatika mengikuti urutan operator (aturan matematika):
  + **10 + 2 = 12**
  + **12 / 3 = 4**
  + **4 \* 5 = 20**
* PHP mengikuti **urutan operasi matematika (PEMDAS)**.
* ✅ Berguna untuk menghitung rumus matematis yang lebih kompleks.

**Kelebihan Kode:**

* Menunjukkan pemahaman dasar tentang operator aritmatika dan fungsi random dalam PHP.
* Dapat dijadikan dasar untuk membuat kalkulator, simulasi, atau sistem berbasis angka acak.

**Kekurangan:**

* Tidak ada validasi input karena semua nilai sudah ditentukan (hardcoded).
* rand() kurang aman untuk aplikasi kriptografi, lebih baik gunakan random\_int() untuk keamanan.

**Kesimpulan:**

Kode ini cocok untuk pemula yang ingin belajar tentang operasi matematika dan penggunaan fungsi random di PHP. Ini merupakan latihan yang baik untuk memahami dasar-dasar logika pemrograman.

**7. PHP Constants**

a. Penjelasan

Konstanta adalah variabel yang nilainya tidak dapat diubah. Didefinisikan menggunakan define().

b. Contoh Kode File: constants.php



* jika dipraktikkan:



* pembuatan konstanta untuk menyimpan nama mata kuliah dan tampilkan.



Dengan kode:

|  |
| --- |
| <?php  define("NAMA\_KAMPUS", "STITEK Bontang");  define("MATA\_KULIAH", "Pemrograman Web");  echo "Saya kuliah di " . NAMA\_KAMPUS . "<br>";  echo "Mata kuliah yang sedang saya pelajari adalah " . MATA\_KULIAH;  ?> |

* Analisis:

Kode ini menggunakan fungsi define() untuk mendeklarasikan **konstanta**, yaitu nilai tetap yang tidak dapat diubah selama eksekusi program.

**🔹 Elemen Kode:**

1. **define("NAMA\_KAMPUS", "STITEK Bontang")**  
   → Menyimpan nama kampus sebagai konstanta.
2. **define("MATA\_KULIAH", "Pemrograman Web")**  
   → Menyimpan nama mata kuliah sebagai konstanta tambahan.
3. **Output**  
   → Menampilkan kedua nilai konstanta dengan echo.

**🔹 Kelebihan:**

* ✅ **Aman dari perubahan nilai**: Konstanta tidak bisa ditimpa secara tidak sengaja.
* ✅ **Rapi dan terstruktur**: Memudahkan pemeliharaan program, terutama untuk nilai tetap seperti nama instansi, mata kuliah, dsb.
* ✅ **Meningkatkan keterbacaan kode**: Nama konstanta mewakili isi nilainya secara jelas.

**🔹 Kekurangan:**

* ❌ Tidak bisa menyimpan nilai yang akan berubah atau dinamis (misalnya nama pengguna).
* ❌ Tidak bisa menggunakan konstanta di dalam class secara fleksibel tanpa const (perlu pendekatan berbeda).

**🔹 Kesimpulan:**

Kode ini sangat cocok sebagai dasar untuk memahami penggunaan **konstanta global** dalam PHP, ideal untuk nilai tetap yang sering digunakan dalam aplikasi.

**8. PHP Operators**

a. Penjelasan

Operator digunakan untuk melakukan operasi, seperti perbandingan (==,>) dan logika (&&,||).

b. Contoh Kode File: operators.php

|  |
| --- |
| <?php  $umur = 18;  $sudah\_punya\_sim = false;  if ($umur >= 17 && $sudah\_punya\_sim == true) {  echo "Anda boleh mengemudi.";  } else {  echo "Anda tidak boleh mengemudi.";  }  ?> |

* Kode di atas di praktikkan:



* Ubah nilai variabel agar kondisi terpenuhi dan pesan “Anda boleh mengemudi” di tampilkan.

Dengan kode:

|  |
| --- |
| <?php  $umur = 18;  $sudah\_punya\_sim = true; // diubah dari false ke true  if ($umur >= 17 && $sudah\_punya\_sim == true) {  echo "Anda boleh mengemudi.";  } else {  echo "Anda tidak boleh mengemudi.";  }  ?> |

Dan jika dipraktikkan:



* Analisis:

**✅ Tujuan Kode:**

Untuk menentukan apakah seseorang **boleh mengemudi** berdasarkan **usia** dan **kepemilikan SIM**.

**🔹 Struktur Logika:**

php

CopyEdit

if ($umur >= 17 && $sudah\_punya\_sim == true)

* **$umur >= 17** → syarat minimal usia legal untuk mengemudi.
* **$sudah\_punya\_sim == true** → memastikan sudah memiliki Surat Izin Mengemudi (SIM).
* Kedua kondisi **harus bernilai benar (true)** agar pernyataan dalam blok if dijalankan.

**🔹 Perubahan:**

* Nilai $sudah\_punya\_sim diubah dari false ke true agar kondisi terpenuhi.

**🔹 Hasil:**

* Menampilkan: **"Anda boleh mengemudi."**

**Kesimpulan:**

Kode ini merupakan contoh sederhana penggunaan **percabangan if-else** dan operator logika **AND (&&)** dalam PHP.  
Cocok untuk skenario keputusan berbasis banyak syarat. Sangat bermanfaat dalam validasi data pada aplikasi berbasis form atau sistem izin akses.

**9. PHP if Statements**

a. Penjelasan

Struktur if...elseif...else digunakan untuk membuat keputusan bedasarkan berbagai kondisi.

b. Contoh Kode FIle: if\_statements.php



c. Latihan Praktikum

* Kode di atas jika dipraktikkan:



* Buat if statement untuk memeriksa apakah sebuah angka positif, negatif, atau nol.

Kode:

|  |
| --- |
| <?php  $angka = -7;  if ($angka > 0) {  echo "Angka positif";  } elseif ($angka < 0) {  echo "Angka negatif";  } else {  echo "Angka nol";  }  ?> |

Dan jika dipraktikkan:



* Analisis:

**🎯 Tujuan:**

Menentukan **kategori nilai sebuah angka**: apakah **positif**, **negatif**, atau **nol**.

**🔍 Logika Kondisional:**

* if ($angka > 0) → Mengecek apakah angka lebih besar dari nol.
* elseif ($angka < 0) → Mengecek apakah angka lebih kecil dari nol.
* else → Menangani kemungkinan terakhir, yaitu angka **nol**.

**✅ Kelebihan:**

* Struktur kode jelas dan mudah dibaca.
* Menangani semua kemungkinan input angka (positif, negatif, nol).
* Berguna untuk validasi numerik dasar, seperti dalam kalkulator, input user, atau sistem monitoring.

**📌 Kesimpulan:**

Kode ini merupakan contoh praktis penggunaan **struktur percabangan** dalam PHP. Cocok untuk pemula yang ingin memahami dasar-dasar pengambilan keputusan berdasarkan kondisi numerik.

**10. PHP switch Statement**

a. Penjelasan

Switch adalah alternatif if-elseif-else yang lebih efisien untuk satu variabel dengan banyak kemungkinan nilai.

b. Contoh Kode File: switch.php



c. Latihan praktikum

* Kode di atas jika dipraktikkan:



* Buat switch statement untuk menampilkan pesan berbeda berdasarkan $ukuran\_baju (“S”,”M”,”L”,”XL”).

Berikut kodenya:

|  |
| --- |
| <?php  $ukuran\_baju = "M";  switch ($ukuran\_baju) {  case "S":  echo "Ukuran Small - cocok untuk anak-anak atau remaja.";  break;  case "M":  echo "Ukuran Medium - pas untuk orang dewasa berpostur sedang.";  break;  case "L":  echo "Ukuran Large - cocok untuk postur tubuh besar.";  break;  case "XL":  echo "Ukuran Extra Large - untuk tubuh sangat besar.";  break;  default:  echo "Ukuran tidak tersedia.";  }  ?> |

Jika dipraktikkan:



* Analisis:

**🎯 Tujuan:**

Kode ini menggunakan struktur switch untuk menentukan pesan yang sesuai berdasarkan nilai dari variabel $ukuran\_baju, seperti **"S"**, **"M"**, **"L"**, atau **"XL"**.

**🔍 Logika Program:**

* switch ($ukuran\_baju) akan mencocokkan nilai dengan setiap case.
* Jika cocok, maka pesan yang relevan akan ditampilkan menggunakan echo.
* break mencegah eksekusi lanjut ke case lain.
* default menangani nilai yang tidak sesuai dengan case yang tersedia.

**✅ Kelebihan:**

* ✅ Lebih efisien dan rapi dibandingkan if-elseif saat menangani banyak nilai tetap.
* ✅ Mudah dibaca dan dimodifikasi jika perlu menambahkan ukuran baru (misalnya XXL).
* ✅ Cocok digunakan dalam **form input pengguna**, **stok produk**, atau **pengaturan preferensi ukuran**.

**📌 Kesimpulan:**

Struktur switch dalam contoh ini sangat tepat digunakan untuk seleksi nilai diskrit seperti ukuran baju. Ini membantu membuat program lebih terorganisir, khususnya ketika menangani banyak pilihan yang saling eksklusif.

**11. PHP Loops**

a. Penjelasan

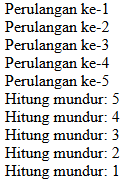
Perulangan (for, while, foreach) digunakan untuk mengeksekusi blok kode berulang kali.

b. Contoh kode file: Loops.php

|  |
| --- |
| <?php  // Perulangan for  for ($i = 0; $i < 5; $i++) {  echo "Perulangan ke-" . ($i + 1) . "<br>";  }  // Perulangan while  $angka = 5;  while ($angka > 0) {  echo "Hitung mundur: $angka <br>";  $angka--;  }  ?> |

c. Latihan Praktikum:

* Silahkan praktikkan contoh kode di atas.

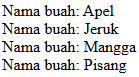


* Gunakan foreach untuk menampilkan semua elemen dari sebuah array.

Berikut kodenya:

|  |
| --- |
| <?php  $buah = ["Apel", "Jeruk", "Mangga", "Pisang"];  foreach ($buah as $item) {  echo "Nama buah: $item<br>";  }  ?> |

Dengan hasil praktiknya:



* Analisis:

**🎯 Tujuan:**

Menampilkan seluruh elemen dari array $buah menggunakan perulangan foreach.

**🔍 Logika Program:**

* $buah adalah array berisi elemen string: "Apel", "Jeruk", "Mangga", dan "Pisang".
* foreach ($buah as $item) menelusuri setiap elemen dalam array dan menyimpannya sementara di variabel $item.
* Setiap nilai $item ditampilkan menggunakan echo.

**✅ Kelebihan:**

* ✅ foreach sangat cocok digunakan untuk **array** karena tidak perlu menggunakan indeks atau panjang array secara manual.
* ✅ Membuat kode lebih **ringkas**, **mudah dibaca**, dan **minim kesalahan** saat memproses elemen satu per satu.
* ✅ Cocok untuk data dinamis seperti daftar produk, nama siswa, hasil query database, dsb.

**📌 Kesimpulan:**

foreach adalah struktur perulangan yang efisien dan praktis untuk menelusuri semua isi array. Dalam konteks pemrograman web, ini adalah salah satu cara terbaik untuk menampilkan data list secara dinamis.

**12. PHP Functions**

a. Penjelasan

Fungsi adalah blok kode yang dapat digunakan kembali. Dapat menerima parameter dan mengembalikan nilai.

b. Contoh Kode File: functions.php

|  |
| --- |
| <?php  function hitung\_diskon($harga, $persen\_diskon) {  $potongan = $harga \* ($persen\_diskon / 100);  return $harga - $potongan;  }  $harga\_baju = 150000;  $harga\_setelah\_diskon = hitung\_diskon($harga\_baju, 10); // Diskon 10%  echo "Harga akhir: Rp " . $harga\_setelah\_diskon;  ?> |

c. Latihan Praktikum

* Jika kode di atas di praktikkan:



* Buat fungsi sapa($nama, $waktu) yang menampilkan pesan seperti “Selamat Pagi, Budi!”.

Berikut codenya:

|  |
| --- |
| <?php  function sapa($nama, $waktu) {  echo "Selamat $waktu, $nama!";  }  // Contoh pemanggilan fungsi  sapa("Budi", "Pagi");  ?> |

Jika dipraktikkan:



* Analisis:

**🎯 Tujuan:**

Membuat fungsi yang menghasilkan **pesan sapaan personal** berdasarkan nama dan waktu (misal: pagi, siang, sore, malam).

**🔍 Struktur Fungsi:**

* **function sapa($nama, $waktu)**  
  Menerima dua parameter: $nama (nama orang) dan $waktu (periode waktu).
* **echo "Selamat $waktu, $nama!";**  
  Menampilkan pesan dengan format dinamis berdasarkan input.

**✅ Kelebihan:**

* ✅ Sederhana dan mudah digunakan.
* ✅ Fleksibel untuk berbagai situasi, misalnya sapa("Ani", "Malam").
* ✅ Cocok digunakan di halaman web dinamis seperti dashboard, aplikasi absen, atau e-learning.

**📌 Kesimpulan:**

Fungsi ini adalah contoh penerapan **parameter dinamis** dalam fungsi PHP. Berguna untuk membuat aplikasi yang lebih interaktif dan personal.

**13. PHP Arrays**

a. Penjelasan

Array adalah variabel yang bisa menampung banyak nilai, baik dengan indeks numerik (indexed array) maupun nama (associative array).

b. Contoh Kode File: arrays.php

|  |
| --- |
| <?php  // Associative Array  $data\_dosen = [  "nama" => "Ir. ABADI NUGROHO, S.Kom., M.Kom.",  "nidn" => "1104129002",  "matkul" => "Pemrograman Web"  ];  echo "Dosen Pengampu: " . $data\_dosen["nama"];  ?> |

c. Latihan Praktikum:

* Mempraktikkan kode di atas:

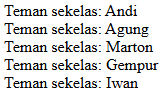


* Buat sebuah *indexed array* yang berisi daftar nama teman sekelas Anda dan tampilkan menggunakan foreach.

Berikut kodenya:

|  |
| --- |
| <?php  $teman\_kelas = ["Andi", "Budi", "Citra", "Dewi", "Eko"];  foreach ($teman\_kelas as $nama) {  echo "Teman sekelas: $nama<br>";  }  ?> |

Jika dipraktikkan:



* Analisis:

**🎯 Tujuan:**

Menampilkan daftar nama teman sekelas yang disimpan dalam bentuk **indexed array** menggunakan perulangan foreach.

**🔍 Struktur Program:**

* **Array $teman\_kelas**  
  Merupakan array dengan indeks numerik otomatis (0, 1, 2, dst), berisi elemen string seperti "Andi", "Budi", dll.
* **foreach ($teman\_kelas as $nama)**  
  Mengeksekusi blok kode sebanyak jumlah elemen array. Setiap elemen disimpan sementara dalam variabel $nama dan ditampilkan.

**✅ Kelebihan:**

* ✅ Indexed array mudah digunakan saat urutan data penting atau tidak memerlukan key khusus.
* ✅ foreach menyederhanakan iterasi tanpa perlu menghitung panjang array (count()).
* ✅ Cocok digunakan dalam data daftar, absensi, maupun sistem presensi online.

**❌ Kekurangan:**

* ❌ Tidak menyimpan informasi tambahan selain nama; jika butuh data tambahan (seperti NIM, jurusan), lebih cocok pakai **associative array** atau **array multidimensi**.

**📌 Kesimpulan:**

Contoh ini menunjukkan implementasi dasar foreach dengan **indexed array**, cocok untuk pemula yang ingin memahami struktur data linear dalam PHP. Efisien untuk menampilkan list sederhana seperti nama siswa, produk, atau entri data lainnya.

**14. PHP Global Variables – Superglobals**

a. Penjelasan

Superglobals adalah variabel bawaan PHP yang selalu dapat diakses. $\_POST dan $\_GET digunakan untuk mengambil data dari form.

b. Contoh Kode FIle: Superglobals.php (form yang mengirim data ke dirinya sendiri)

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <form method="post" action="14\_superglobals.php">  Nama: <input type="text" name="nama">  <input type="submit">  </form>  <br>  <?php  if (isset($\_POST['nama'])) {  $nama = htmlspecialchars($\_POST['nama']);  echo "Halo, " . $nama . "!";  }  ?>  </body>  </html> |

c. Latihan Praktikum:

* Jika kode di atas di praktikkan:



Tombol tidak bisa di submit karena 14\_superglobals.php belum ada

* Coba ubah method form menjadi get dan ubah $\_POST menjadi $\_GET, lalu amati URL Browser

URL Browser akan menjadi:



Ketika diketikkan Martonius dan di submit

* Analisis:

**🎯 Tujuan:**

Menguji perbedaan penggunaan **POST** dan **GET** pada form HTML dalam mengirim data ke server dan mengamati dampaknya di **URL browser**.

**🔄 Perubahan yang Dilakukan:**

* method="post" diubah menjadi method="get".
* Akses variabel dari $\_POST['nama'] diubah menjadi $\_GET['nama'].

**🔍 Hasil Pengamatan:**

Setelah pengguna mengisi form dan klik submit:

* URL berubah menjadi:  
  14\_superglobals.php?nama=Martonius
* Nilai input dikirim dan **terlihat secara langsung** di address bar browser.

**⚖️ Perbandingan GET vs POST:**

| **Aspek** | **GET** | **POST** |
| --- | --- | --- |
| **Visibilitas data** | Terlihat di URL (?nama=...) | Tidak terlihat di URL |
| **Keamanan** | Kurang aman (mudah dimanipulasi) | Lebih aman untuk data sensitif |
| **Ukuran data** | Terbatas (sekitar 2KB) | Lebih besar (tergantung server) |
| **Cocok untuk** | Filter, pencarian, navigasi berbasis URL | Login, registrasi, pengisian form |
| **Bookmarkable** | ✅ Ya, karena URL menyimpan data | ❌ Tidak, karena data tidak ada di URL |

**✅ Kelebihan:**

* GET berguna saat pengguna ingin menyimpan atau membagikan hasil form via URL.
* Data lebih transparan dan mudah di-debug saat testing.

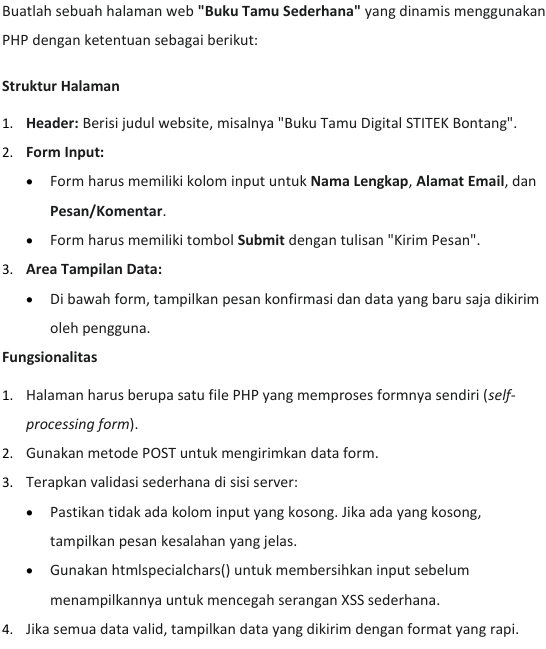
**❌ Kekurangan:**

* Tidak cocok untuk mengirim **data pribadi atau sensitif**.
* Mudah dimodifikasi di address bar, sehingga rawan manipulasi.

**📌 Kesimpulan:**

Perubahan metode form dari POST ke GET mengungkapkan perbedaan penting terkait **keamanan**, **transparansi**, dan **kegunaan URL**. Pilihan antara keduanya harus disesuaikan dengan konteks: gunakan GET untuk keperluan ringan yang tidak sensitif, dan POST untuk data penting atau rahasia.

## D. Tugas Praktikum





Jawab: berikut full kode tugasphp.php

|  |
| --- |
| <!DOCTYPE html>  <html lang="id">  <head>  <meta charset="UTF-8">  <title>Buku Tamu Digital STITEK Bontang</title>  <style>  body {  font-family: Arial, sans-serif;  background: #f4f4f4;  margin: 0;  padding: 20px;  }  .container {  max-width: 600px;  background: #fff;  margin: 0 auto;  padding: 30px;  border-radius: 10px;  box-shadow: 0 2px 10px rgba(0,0,0,0.1);  }  h1 {  text-align: center;  color: #2c3e50;  }  label {  display: block;  margin-top: 15px;  font-weight: bold;  }  input[type="text"], input[type="email"], textarea {  width: 100%;  padding: 10px;  margin-top: 5px;  border-radius: 5px;  border: 1px solid #ccc;  }  textarea {  resize: vertical;  }  .btn {  margin-top: 20px;  background-color: #2c3e50;  color: #fff;  padding: 10px 20px;  border: none;  border-radius: 5px;  cursor: pointer;  }  .btn:hover {  background-color: #34495e;  }  .output {  margin-top: 30px;  background: #ecf0f1;  padding: 20px;  border-radius: 5px;  }  .error {  color: red;  }  </style>  </head>  <body>  <div class="container">  <h1>Buku Tamu Digital STITEK Bontang</h1>  <form method="post" action="">  <label for="nama">Nama Lengkap</label>  <input type="text" id="nama" name="nama">  <label for="email">Alamat Email</label>  <input type="email" id="email" name="email">  <label for="pesan">Pesan / Komentar</label>  <textarea id="pesan" name="pesan" rows="5"></textarea>  <button type="submit" class="btn">Kirim Pesan</button>  </form>  <?php  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  $nama = trim($\_POST['nama']);  $email = trim($\_POST['email']);  $pesan = trim($\_POST['pesan']);  $errors = [];  if (empty($nama)) $errors[] = "Nama tidak boleh kosong.";  if (empty($email)) $errors[] = "Email tidak boleh kosong.";  if (empty($pesan)) $errors[] = "Pesan tidak boleh kosong.";  if (!empty($errors)) {  echo '<div class="output error">';  foreach ($errors as $e) {  echo "<p>• $e</p>";  }  echo '</div>';  } else {  // Sanitasi input  $nama = htmlspecialchars($nama);  $email = htmlspecialchars($email);  $pesan = htmlspecialchars($pesan);  echo '<div class="output">';  echo "<h3>Terima kasih atas pesan Anda!</h3>";  echo "<p><strong>Nama:</strong> $nama</p>";  echo "<p><strong>Email:</strong> $email</p>";  echo "<p><strong>Pesan:</strong><br>$pesan</p>";  echo '</div>';  }  }  ?>  </div>  </body>  </html> |

Berikut tampilannya jika dijalankan dan diisi form dan disubmit:



Berikut penjelasan video pada link github: [Link/Tautan](https://github.com/Martonius3/Modul5/blob/main/Tugas%20Modul%205.mp4)

# DAFTAR PUSTAKA

- 3C. (n.d.). *HTML5 Specification*. World Wide Web Consortium (W3C). Retrieved from <https://www.w3.org/TR/html5/>

- MDN Web Docs. (n.d.). *HTML forms - <form>, <input>, and other form elements*. Mozilla Developer Network. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/form>

- W3Schools. (n.d.). *HTML Forms*. Retrieved from https://www.w3schools.com/html/html\_forms.asp

- Beyer, S. (2017). *HTML and CSS: Design and Build Websites*. Wiley.

- Shay, R. (2014). *HTML5 for Web Designers*. A Book Apart.

- Google Web Fundamentals. (n.d.). *Forms and Validation*. Google Developers. Retrieved from https://developers.google.com/web/fundamentals/forms