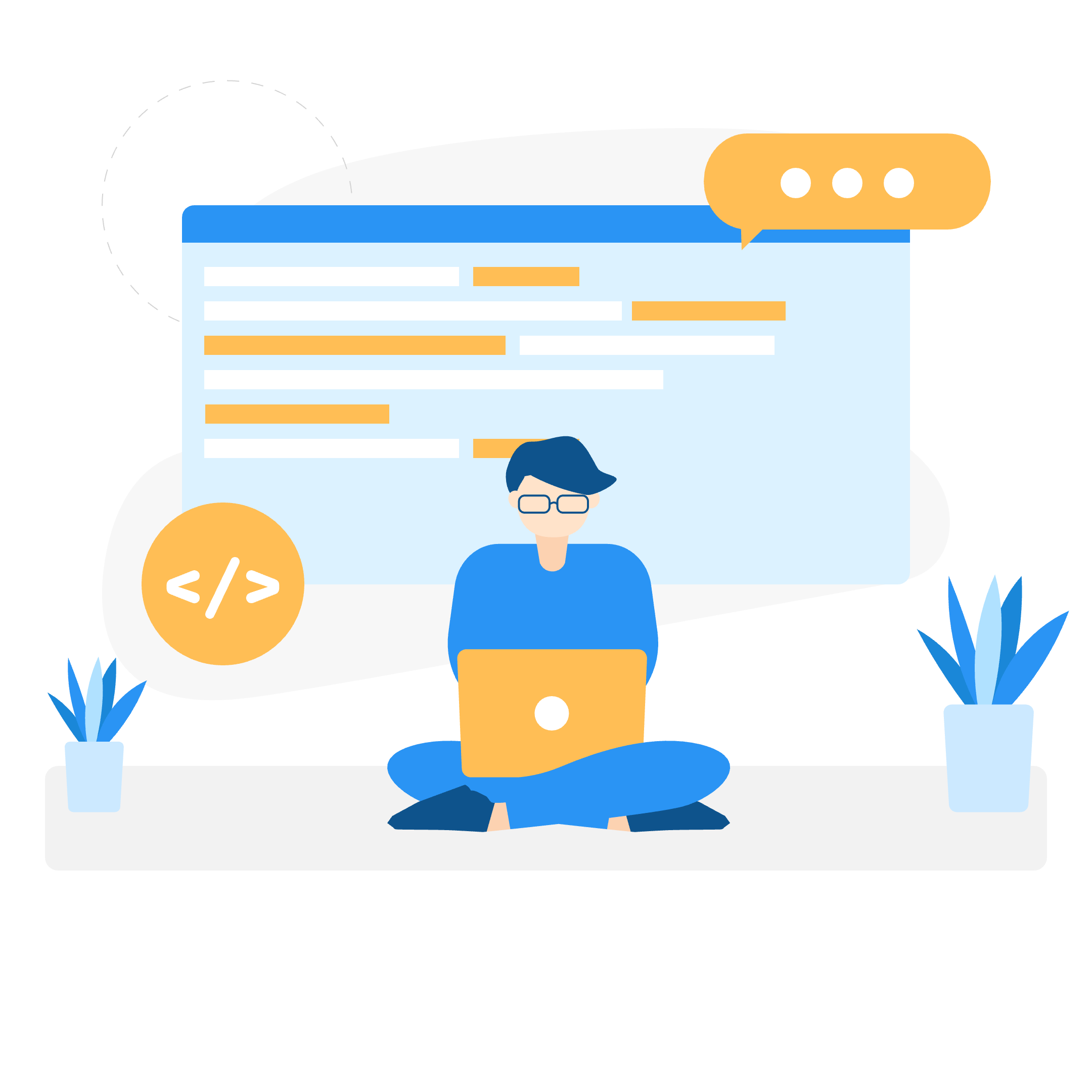
**BAB 4**

**CSS**



* 1. **Tujuan**

Setelah menyelesaikan bab ini dan mengikuti kegiatan praktikum, mahasiswa diharapkan mampu:

1. Memahami dasar-dasar CSS (*Cascading Style Sheets*), termasuk struktur, properti, dan selektor.
2. Mengimplementasikan CSS dalam pembuatan website untuk mengatur gaya (*style*), latar belakang (*background*), *web fonts*, dan membangun tata letak (*layout*) modern menggunakan CSS Grid.
   1. **Pengantar**

Jika HTML adalah kerangka (*skeleton*) dari sebuah halaman web, maka CSS (*Cascading Style Sheets*) adalah kulit, pakaian, dan penampilannya. CSS adalah bahasa yang kita gunakan untuk "menghias" elemen-elemen HTML, mulai dari warna dan ukuran teks, hingga tata letak halaman yang kompleks. Menguasai CSS adalah langkah awal untuk mengubah halaman web yang polos menjadi sebuah halaman yang menarik dan profesional.

* + 1. **Apa itu CSS?**

CSS adalah bahasa *stylesheet* yang digunakan untuk mendeskripsikan presentasi visual dari sebuah dokumen HTML. Peran utamanya adalah untuk memisahkan antara konten (HTML) dan visual (CSS). Pemisahan ini memiliki keuntungan besar seperti efisiensi, kemudahan perawatan kode, dan peningkatan aksesibilitas.

* + 1. **Sintaks dasar CSS**

Aturan CSS (*CSS Rule*) terdiri dari **Selektor** dan **Blok Deklarasi**.

1. */\* Ini adalah contoh aturan CSS \*/*
2. p {
3. color: blue;
4. font-size: 16px;
5. }

* Selektor (p): Menargetkan elemen HTML yang ingin diberi gaya.
* Blok Deklarasi ({...}): Berisi satu atau lebih deklarasi gaya.
* Deklarasi (color: blue;): Terdiri dari Properti (color) dan Nilai (blue)

### Penerapan CSS

Ada tiga metode untuk mengintegrasikan CSS ke dalam dokumen HTML Anda. Memahami ketiganya penting untuk mengetahui kapan harus menggunakan setiap pendekatan.

1. **External CSS (Metode Terbaik)** Ini adalah cara yang paling umum dan sangat direkomendasikan. Anda menulis semua aturan CSS Anda di dalam sebuah file terpisah dengan ekstensi .css (misalnya, style.css), lalu menghubungkannya ke file HTML menggunakan tag <link> di dalam <head>.

**Keuntungan:**

* + 1. **Pemisahan yang Jelas:** Kode HTML dan CSS benar-benar terpisah, membuat proyek lebih rapi dan mudah dikelola.
    2. **Dapat Digunakan Kembali:** Satu file CSS dapat digunakan oleh banyak halaman HTML. Mengubah satu file ini akan mengubah tampilan semua halaman yang terhubung.

**Contoh di index.html:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Halaman Saya</title>
5. <link rel="stylesheet" href="style.css">
6. </head>
7. <body>
8. <h1>Selamat Datang</h1>
9. <p>Ini adalah paragraf yang diberi gaya oleh file CSS eksternal.</p>
10. </body>
11. </html>

### Contoh di style.css:

1. body {
2. background-color: #f0f0f0;
3. font-family: Arial, sans-serif;
4. }
5. h1 {
6. color: navy;
7. }
8. **Internal CSS** Anda dapat menulis aturan CSS langsung di dalam file HTML menggunakan tag <style> yang ditempatkan di dalam <head>. Metode ini berguna jika sebuah halaman memiliki gaya yang sangat unik dan tidak akan digunakan di halaman lain.

**Keuntungan:**

* **Mandiri:** Semua yang dibutuhkan untuk satu halaman (HTML dan CSS) ada dalam satu file. **Kerugian:**
* Tidak dapat digunakan kembali untuk halaman lain.

**Contoh di index.html:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Halaman dengan Internal CSS</title>
5. <style>
6. body {
7. background-color: #e6f7ff;
8. }
9. p {
10. color: #005f99;
11. font-size: 18px;
12. }
13. </style>
14. </head>
15. <body>
16. <p>Paragraf ini diatur gayanya oleh Internal CSS.</p>
17. </body>
18. </html>
19. **Inline CSS** Anda menerapkan gaya langsung pada sebuah elemen HTML spesifik menggunakan atribut style. Metode ini **tidak disarankan** untuk penggunaan umum karena mencampurkan kembali konten dan presentasi, yang bertentangan dengan prinsip dasar CSS. Biasanya hanya digunakan untuk tujuan yang sangat spesifik seperti pengujian cepat atau saat gaya dihasilkan secara dinamis oleh JavaScript.

**Kerugian:**

* **Sulit Dikelola:** Membuat perawatan dan pembaruan desain menjadi sangat sulit.
* **Prioritas Tertinggi:** Gaya inline akan menimpa gaya dari internal maupun external CSS, yang bisa menyebabkan kebingungan.

**Contoh di index.html:**

1. <body>
2. <h1 style="color: red; text-align: center;">Judul ini memiliki gaya inline.</h1>
3. </body>

### Selektor CSS

Selektor adalah bagian paling kuat dari CSS. Ia adalah pola yang kita gunakan untuk "memilih" atau menargetkan elemen HTML yang ingin kita beri gaya.

* **Selektor Tipe/Elemen (Type Selector):** Menargetkan semua elemen dengan nama tag yang sama. Ini adalah selektor yang paling dasar.
* Contoh CSS :
* */\* Semua elemen <h2> akan berwarna hijau \*/*
* h2 {
* color: green;
* }
* */\* Semua paragraf akan memiliki ukuran font 16px \*/*
* p {
* font-size: 16px;
* }

**Selektor Class (Class Selector):** Menargetkan elemen berdasarkan nilai atribut class-nya. Selektor *class* diawali dengan tanda titik (.). Ini adalah selektor yang paling fleksibel dan umum digunakan karena satu *class* dapat diterapkan pada banyak elemen, dan satu elemen dapat memiliki beberapa *class*.

**Contoh CSS:**

1. */\* Semua elemen dengan class="highlight" akan memiliki latar kuning \*/*
2. .highlight {
3. background-color: yellow;
4. }
5. .text-center {
6. text-align: center;
7. }

### Contoh HTML

1. <p class="highlight">Paragraf ini akan disorot.</p>
2. <div class="highlight text-center">Div ini disorot dan teksnya di tengah.</div>

**Selektor ID (ID Selector):** Menargetkan **satu elemen unik** berdasarkan nilai atribut id-nya. Selektor ID diawali dengan tanda pagar (#). Atribut id harus unik dalam satu halaman; tidak boleh ada dua elemen dengan id yang sama.

**Contoh CSS:**

1. */\* Elemen dengan id="header-utama" akan memiliki border bawah \*/*
2. #header-utama {
3. border-bottom: 2px solid black;
4. padding: 20px;
5. }

### Contoh HTML

1. <div id="header-utama">
2. <h1>Judul Utama Website</h1>
3. </div>

### Web Fonts

Secara *default*, peramban hanya akan menggunakan font yang terinstal di sistem operasi pengguna. Dengan *web fonts*, kita dapat memuat jenis huruf (*font*) kustom dari server eksternal (seperti Google Fonts) agar desain web kita lebih unik dan konsisten di semua perangkat.

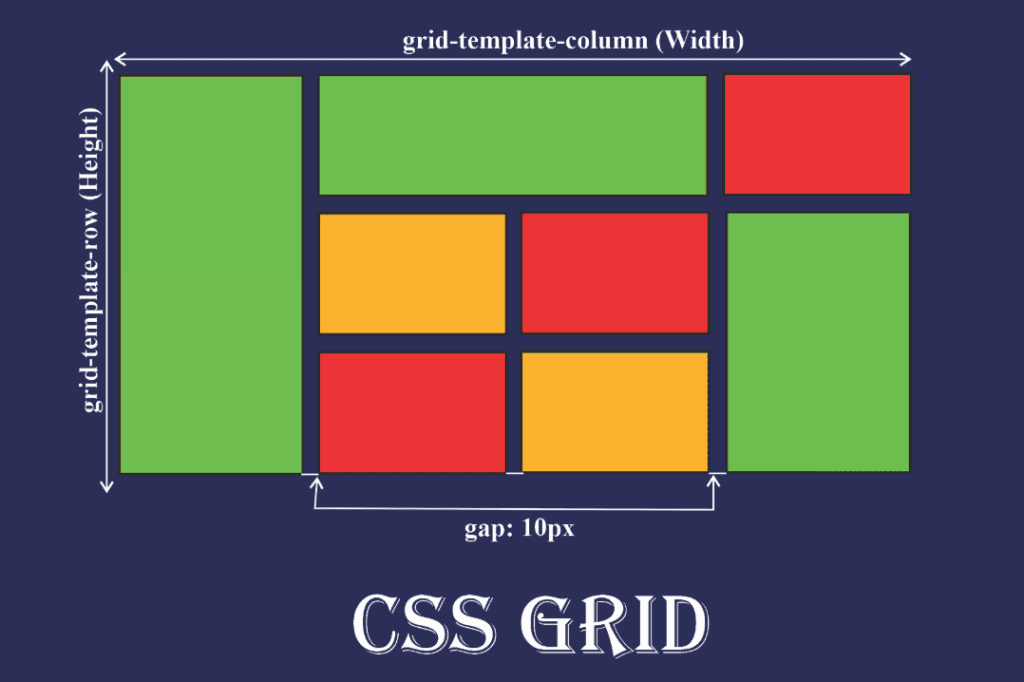
**Cara menggunakan Google Fonts:**

1. Kunjungi <https://fonts.google.com>.
2. Pilih font yang Anda suka. Pilih Get embed code.
3. Salin kode <link> yang disediakan dan tempelkan di dalam <head> file HTML Anda, atau salin kode @import dan tempelkan di baris paling atas file CSS Anda.
4. Gunakan nama font tersebut dalam properti font-family di CSS Anda.

**Contoh di CSS:**

1. @import url('https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,300..800;1,300..800&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');
2. body {
3. font-family: 'Poppins', sans-serif; */\* Poppins akan digunakan, jika gagal, sans-serif akan jadi fallback \*/*
4. }
   * 1. CSS Grid

CSS Grid adalah sistem layout dua dimensi (baris dan kolom) yang sangat kuat dan fleksibel. Ini adalah cara modern untuk merancang tata letak halaman web, menggantikan teknik lama seperti float.



**Konsep Inti:**

1. **Grid Container:** Elemen induk yang kita beri display: grid;.
2. **Grid Items:** Elemen-elemen anak langsung dari *grid container*.
3. **Grid Lines:** Garis-garis pembagi horizontal dan vertikal yang membentuk struktur grid.
4. **Grid Tracks:** Ruang di antara dua *grid lines* (membentuk kolom atau baris).
5. **Grid Cells:** Unit terkecil dari grid, ruang di antara empat *grid lines* yang bersilangan.

**Properti Dasar pada Container:**

* display: grid;: Mengaktifkan layout grid.
* grid-template-columns: Mendefinisikan jumlah dan ukuran kolom. Contoh: grid-template-columns: 200px 1fr 1fr; (membuat 3 kolom: kolom pertama 200px, dua kolom berikutnya membagi sisa ruang secara merata). Unit fr (*fractional unit*) sangat berguna untuk layout fleksibel.
* grid-template-rows: Mendefinisikan jumlah dan ukuran baris.
* gap: Menentukan jarak (celah) antara baris dan kolom. Contoh: gap: 20px;
  1. **Kegiatan Praktikum**

**Langkah 1: Membuat file CSS**

Buka folder PraktikumHTML dari pertemuan sebelumnya. Buat file baru bernama style.css. Buka file index.HTML, hubungkan style.css ke index.html menggunakan tag <link> di dalam <head>.

Buka style.css. Di baris paling atas, impor font "Poppins" dari Google Fonts.

yang sudah dikerjakan pada Bab 2. Tambahkan kode berikut dibawah bagian terakhir yang dibuat dan diatas tag penutup <body>

1. *<!-- ... kode sebelumnya ... -->*
2. </ol>
3. <h2>Jadwal Kuliah Semester Ini</h2>
4. <table border="1">
5. <thead>
6. <tr>
7. <th>Hari</th>
8. <th>Mata Kuliah</th>
9. <th>Jam</th>
10. </tr>
11. </thead>
12. <tbody>
13. <tr>
14. <td>Senin</td>
15. <td>Praktikum Pemrograman Web</td>
16. <td>10:00 - 12:00</td>
17. </tr>
18. <tr>
19. <td>Selasa</td>
20. <td>Basis Data</td>
21. <td>08:00 - 10:00</td>
22. </tr>
23. <tr>
24. <td>Rabu</td>
25. <td>Struktur Data</td>
26. <td>13:00 - 15:00</td>
27. </tr>
28. </tbody>
29. </table>

Simpan file kemudian buka atau *refresh* file index.html

### Langkah 2: Menambahkan form kontak

Setelah tag penutup </table>, tambahkan kode berikut

1. *<!-- ... kode sebelumnya ... -->*
2. </table>
3. <h2>Hubungi Saya</h2>
4. <form>
5. <label for="nama">Nama Lengkap:</label><br>
6. <input type="text" id="nama" name="nama"><br><br>
7. <label for="email">Alamat Email:</label><br>
8. <input type="email" id="email" name="email"><br><br>
9. <label for="pesan">Pesan:</label><br>
10. <textarea id="pesan" name="pesan" rows="4" cols="50"></textarea><br><br>
11. <textarea id="pesan" name="pesan" rows="4" cols="50"></textarea><br><br>
12. <input type="submit" value="Kirim Pesan">
13. </form>

Simpan file (Ctrl+S) kemudian buka atau *refresh* file index.html.

* 1. **Rangkuman**
* Tabel HTML (<table>) digunakan untuk menyajikan data tabular. Struktur utamanya terdiri dari <tr> (baris), <th> (sel header), dan <td> (sel data). Penggunaan <thead>, <tbody>, dan <tfoot> sangat disarankan untuk struktur yang semantik.
* Formulir HTML (<form>) adalah cara standar untuk mengumpulkan input dari pengguna. Elemen <form> membungkus berbagai jenis elemen input.
* Elemen <input> adalah elemen serbaguna yang tipenya dapat diubah melalui atribut type (misalnya, text, email, radio, submit).
* Penggunaan <label> yang dihubungkan dengan id input melalui atribut for sangat krusial untuk aksesibilitas dan pengalaman pengguna yang baik.
  1. **Tugas**

Kerjakan tugas-tugas di bawah ini untuk memperkuat pemahaman dan keterampilan Anda.

1. Jelaskan perbedaan mendasar antara metode GET dan POST pada sebuah formulir HTML. Kapan sebaiknya kita menggunakan masing-masing metode?
2. Apa fungsi dari atribut colspan dan rowspan pada sel tabel?
3. Lanjutkan file index.html yang telah Anda buat :
   * Pada **tabel jadwal kuliah** di file index.html, tambahkan satu baris baru. Pada baris tersebut, buat satu sel yang membentang sepanjang 3 kolom (menggunakan colspan) dan beri teks "Waktu Istirahat".
   * Pada **formulir kontak**, tambahkan sebuah *dropdown menu* (<select>) untuk menanyakan "Tujuan Menghubungi" dengan opsi: "Pertanyaan", "Kerja Sama", dan "Lainnya".