

BUKU AJAR

PEMROGRAMAN WEB I

(Studi Kasus: Website Profil Kampus)

Penulis:

Roy Amrullah Ritonga, M.Kom

Gustina, M.Kom

Anita Megayanti, M.Kom



BUKU AJAR
PEMROGRAMAN WEB I

(Studi Kasus: Website Profil Kampus)

2025 I 00619

Penulis

**Roy Amrullah Ritonga, M.Kom I Gustina, M.Kom I Anita
Megayanti, M.Kom**

QRSBN : 62-0108-03015-1

Penyelia

Dr. Abdul Rahman H, M.T., C.T

Editor

Abdul Rosid, SE.,M.Hum

Desain Sampul

Lukas Liani, S.Psi.

Layout

Asep Nugraha, S.Hum.

Cetakan Pertama, Januari 2026

viii + 115 hlm ; 14.8 x 21 cm

Penerbit

Yayasan Pendidikan dan Sosial

Indonesia Maju (YPSIM) Banten

Kavling Aji Said – Muntill Permai

Blok A.12 Lingkungan Muntill

Kota Serang Provinsi Banten

E-mail: Ypsimbanten@gmail.com

Website : www.ypsimbanten.com

WhatsApp: 0815 9516 818

ANGGOTA IKAPI No. 039/BANTEN/2020

(IKATAN PENERBIT INDONESIA

*Hak Cipta Dilindungi oleh Undang-undang Dilarang mengutip
atau memperbanyak sebagian atau seluruh isi buku ini dalam
bentuk apapun juga tanpa izin tertulis dari Penerbit*

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga buku ajar Pemrograman Web ini dapat disusun dan diselesaikan dengan baik. Buku ajar ini disusun sebagai salah satu bentuk kontribusi akademik dalam mendukung proses pembelajaran pada pendidikan tinggi, khususnya pada mata kuliah Pemrograman Web.

Buku ajar ini merupakan hasil **kolaborasi akademik antara Universitas Alkhairiyah dan STTIKOM Insan Unggul Cilegon** sebagai wujud sinergi antar perguruan tinggi dalam pengembangan bahan ajar yang bermutu dan relevan dengan perkembangan teknologi informasi. Kolaborasi ini diharapkan dapat memperkaya perspektif keilmuan serta meningkatkan kualitas pembelajaran melalui penyajian materi yang sistematis, aplikatif, dan berorientasi pada capaian pembelajaran lulusan.

Penyusunan buku ajar ini melibatkan dosen dari masing-masing institusi yaitu Roy Amrullah Ritonga, M.Kom dari Universitas Alkhairisyh serta Gustina, M.Kom dan Anita Megayanti, M.Kom dari STTIKOM Insan Unggul Cilegon. Materi dalam buku ini disusun secara bertahap, dimulai dari pengenalan konsep dasar pemrograman web, penguasaan HTML, CSS, dan JavaScript, hingga penerapan workflow pengembangan dan deployment website profil kampus sebagai studi kasus terpadu.

Buku ajar ini dirancang untuk mendukung pencapaian Capaian Pembelajaran Mata Kuliah (CPMK) dan Capaian Pembelajaran Lulusan (CPL) sesuai dengan kurikulum pendidikan tinggi. Penyajian materi dilengkapi dengan

contoh, studi kasus, latihan, dan proyek akhir guna mendorong mahasiswa untuk tidak hanya memahami konsep, tetapi juga mampu mengimplementasikannya secara praktis.

Penulis menyadari bahwa buku ajar ini masih memiliki keterbatasan. Oleh karena itu, kritik dan saran yang konstruktif sangat diharapkan sebagai bahan perbaikan dan penyempurnaan pada edisi selanjutnya. Akhir kata, semoga buku ajar ini dapat memberikan manfaat bagi dosen, mahasiswa, serta seluruh pihak yang berkepentingan dalam pengembangan dan peningkatan kualitas pembelajaran pemrograman web di perguruan tinggi.

Cilegon, Januari 2026

Tim Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	iv
BAB 1	1
PENGANTAR PEMROGRAMAN WEB.....	1
1.1. Definisi Web dan Sejarah Perkembangan Web	1
1.2. Cara Kerja Website.....	2
1.3. Teknologi Front-End vs Back-End	4
1.4. Tools untuk Pemrograman Web	5
1.5. Struktur Dasar Pengembangan Website Profil Kampus	5
BAB 2	8
DASAR-DASAR HTML	8
2.1. Struktur Dokumen HTML	8
2.2. Elemen, Atribut, dan Komentar dalam HTML	9
2.3. Elemen Teks: Heading, Paragraf dan Daftar (List)	10
2.4. Link dan Anchor.....	11
2.5. Gambar & Media	12
2.6. Studi Kasus: Membuat Struktur Halaman Home Website Kampus	13
BAB 3	15
LAYOUT HTML LANJUTAN.....	15
3.1. Table, Form, dan Input	15
3.2. Semantic HTML5.....	17
3.3. Accessibility Dasar (A11y).....	19
3.4. Studi Kasus: Membuat Layout Halaman “Profil Kampus”.....	20
BAB 4.....	23
Pengenalan CSS (<i>Cascading Style Sheets</i>).....	23
4.1. Apa Itu CSS dan Cara Kerjanya	23
4.2. Cara Menyisipkan CSS.....	24
4.3. Selektor CSS Dasar	25
4.4. Properti CSS Umum	26

4.5.	Studi Kasus: Styling Halaman Beranda Sederhana.....	27
BAB 5	31
LAYOUT WEBSITE DENGAN CSS.....		31
5.1.	Box Model dalam CSS	31
5.2.	Display: block, inline, dan inline-block.....	32
5.3.	Positioning.....	33
5.4.	Float dan Clear	34
5.5.	Studi Kasus: Layout 2 Kolom untuk Halaman Profil Kampus	35
BAB 6	40
DESAIN RESPONSIF DENGAN FLEXBOX.....		40
6.1.	Konsep Responsive Design	40
6.2.	Media Query	41
6.3.	Flexbox Lengkap	41
6.4.	Studi Kasus: Membuat Menu Navigasi Responsif dengan Flexbox	43
BAB 7	47
CSS GRID UNTUK LAYOUT MODERN		47
7.1.	Dasar CSS Grid	47
7.2.	Grid Rows & Columns	48
7.3.	Penataan Konten Multiseksi dengan Grid.....	49
7.4.	Studi Kasus: Halaman Fakultas & Program Studi Menggunakan CSS Grid.....	50
BAB 8	55
DESAIN MODERN & UI/UX DASAR.....		55
8.1.	Prinsip Desain Sederhana (Simple & Clean Design) ..	55
8.2.	Pemilihan Warna dan Tipografi.....	56
8.3.	Component-Based Design (Desain Berbasis Komponen).....	58
8.4.	Animasi dan Transition CSS.....	59
8.5.	Studi Kasus: Mempercantik Website Kampus.....	60
BAB 9	65
PENGENALAN JAVASCRIPT		65
9.1.	Apa Itu JavaScript & Perannya.....	65

9.2.	Variabel, Tipe Data, dan Operator	66
9.3.	Control Flow (if, else, loop).....	67
9.4.	Function & Scope	68
9.5.	Studi Kasus: Validasi Form Pendaftaran Mahasiswa Baru	69
BAB 10	73
DOM MANIPULATION & EVENT HANDLING	73
10.1.	DOM Tree	73
10.2.	Query Selector	74
10.3.	Event Listener.....	74
10.4.	Manipulasi Elemen DOM.....	75
10.5.	Studi Kasus: Membuat Slider Gambar Kegiatan Kampus	76
BAB 11	80
JAVASCRIPT LANJUTAN	80
11.1.	Array & Object.....	80
11.2.	Local Storage.....	81
11.3.	Template Literal	82
11.4.	Modularisasi Sederhana.....	82
11.5.	Studi Kasus: Halaman Pengumuman Mengambil Data JSON Lokal	83
BAB 12	87
WORKFLOW PENGEMBANGAN WEBSITE	87
12.1.	Struktur Folder Proyek Web	87
12.2.	Optimasi Gambar.....	89
12.3.	Pengelolaan File CSS & JS.....	89
12.4.	Debugging HTML/CSS/JS	90
12.5.	Studi Kasus: Menyatukan Semua Halaman Menjadi Website Kampus Terpadu	91
BAB 13	95
DEPLOY WEBSITE PROFIL KAMPUS	95
13.1.	Pengenalan Git & GitHub.....	95
13.2.	Hosting Gratis (GitHub Pages / Netlify).....	96
13.3.	Struktur Build Sederhana.....	97

13.4.	Studi Kasus: Upload Website Kampus ke GitHub	
Pages	97
BAB 14	101
STUDI KASUS LENGKAP: WEBSITE PROFIL KAMPUS..		101
14.1.	Struktur Folder Final Website Kampus.....	101
14.2.	Template Final Website	102
14.2.1.	Template Header (navigasi).....	102
14.2.2.	Template Footer.....	103
14.2.3.	Template Layout Umum Halaman.....	103
14.3.	Halaman demi Halaman	104
14.4.	Desain Akhir (CSS Template Final)	106
14.5.	Checklist Kelulusan Praktikum Website Kampus.	107
14.6.	Rubrik Penilaian Proyek Akhir Website Kampus.	108
14.7.	Output Final Mahasiswa	108
DAFTAR PUSTAKA.....		110
TENTANG PENULIS.....		112

BAB 1

PENGANTAR PEMROGRAMAN WEB

1.1. Definisi Web dan Sejarah Perkembangan Web

Apa itu Web?

World Wide Web (WWW) adalah kumpulan halaman *web* yang saling terhubung melalui *hyperlink* dan dapat diakses melalui internet menggunakan *browser* seperti *Chrome, Firefox, Edge, atau Safari*. *Web* menggunakan protokol HTTP/HTTPS untuk bertukar informasi antara pengguna (*client*) dan server.

Sejarah Singkat Perkembangan Web

1. 1989 – 1991: Web 1.0 (*Read-Only Web*)
 - Diperkenalkan oleh Tim Berners-Lee di CERN.
 - Website bersifat statis (HTML murni).
 - Pengguna hanya dapat membaca informasi tanpa berinteraksi.
2. 2000 – 2010: Web 2.0 (*Read-Write Web*)
 - Pengguna dapat mengirim data (form, komentar, unggah konten).
 - Teknologi berkembang: JavaScript, AJAX, CSS modern.
 - Munculnya media sosial, blog, YouTube, e-commerce.
3. 2011 – Sekarang: Web 3.0 (*Semantic Web*)
 - Integrasi AI, personalisasi konten, data terdesentralisasi.

- Contoh: aplikasi pintar, rekomendasi otomatis, blockchain web.
4. Perkembangan saat ini
- Website responsif untuk semua perangkat (*mobile-first*).
 - Framework modern seperti React, Vue, Angular.
 - Komputasi awan (*cloud*) untuk hosting.

Kesimpulan: Web berevolusi dari halaman statis menjadi platform interaktif dan cerdas yang kini menjadi fondasi hampir semua layanan digital, termasuk website profil kampus.

1.2. Cara Kerja Website

Website bekerja menggunakan model komunikasi *Client-Server*, yaitu:

1. *Client* (Pengguna)

Perangkat yang mengakses website: laptop, PC, smartphone. Client menggunakan web browser untuk:

- Mengetik URL
- Mengirim permintaan (request)
- Menampilkan halaman yang diterima dari server

2. Server

Komputer khusus yang menyimpan:

- File website (HTML, CSS, JS)
- Database
- Program backend

Server menerima permintaan dari client dan mengirimkan kembali response berisi halaman web.

Alur Kerja Client–Server Sederhana

1. Pengguna mengetik URL: <https://kampus.ac.id>
 2. Browser mengirim HTTP Request ke server kampus.
 3. Server memproses permintaan.
 4. Server mengirim HTTP Response yang berisi:
 - HTML
 - CSS
 - JavaScript
 - Gambar atau data lainnya
 5. Browser merender halaman menjadi tampilan visual website.
3. Protokol HTTP dan HTTPS

HTTP (Hypertext Transfer Protocol) adalah protokol standar dalam pertukaran data di web.

HTTPS adalah versi aman dari HTTP yang dienkripsi menggunakan SSL/TLS. Website resmi kampus umumnya wajib menggunakan HTTPS untuk keamanan data.

4. Domain dan Hosting

Domain adalah alamat unik suatu website, contoh: *university.ac.id*, *unival.ac.id*, *example.com*. Domain ibarat nama dari sebuah bangunan.

Hosting adalah Tempat penyimpanan file website (server). Hosting ibarat rumah tempat file disimpan.

Ada beberapa jenis hosting:

- *Shared Hosting* (harga murah, pemula)
- *VPS (Virtual Private Server)*
- *Cloud Hosting (scalable)*
- *Dedicated Server*

Untuk website kampus, biasanya digunakan VPS atau dedicated server karena stabilitas dan keamanan lebih baik.

1.3. Teknologi Front-End vs Back-End

Front-End Development adalah bagian website yang berhubungan langsung dengan pengguna (*user interface*). Teknologi *Front-End* terdiri dari :

1. HTML (struktur)
2. CSS (desain tampilan)
3. JavaScript (interaksi dan logika di sisi pengguna)

Contoh fitur front-end:

- Menu navigasi
- Galeri foto
- Responsive layout
- Slider atau animation

Dalam mata kuliah ini, fokus utama adalah front-end web dasar.

Back-End Development adalah bagian logika server, database, dan pengolahan data. Teknologi *Back-End* terdiri dari :

1. PHP
2. Node.js
3. Python Django
4. Java Spring

Database: MySQL, PostgreSQL

Contoh fitur back-end :Sistem login mahasiswa/dosen, Penyimpanan data PMB, Pengelolaan berita kampus.

Catatan :
Back-end bukan fokus di Pemrograman Web 1,
tetapi dipelajari di pemrograman lanjutan.

1.4. Tools untuk Pemrograman Web

Tools untuk pemrograman web terdiri dari :

1. *Visual Studio Code (VS Code)*
 - Editor kode yang populer, ringan, dan mendukung:
 - Highlighting HTML, CSS, JS
 - Emmet untuk mempercepat penulisan HTML
 - Live Server (preview website)
 - Plugin tambahan (Bootstrap, Prettier, GitLens)
2. *Web Browser (Chrome, Firefox, Edge)*

Digunakan untuk menjalankan dan menguji hasil website.

Browser DevTools merupakan Fitur bawaan browser untuk:

- Inspect element
- Melihat struktur DOM
- Debug CSS/JS
- Melihat console log
- Memeriksa performa website

DevTools sangat penting dalam proses debugging dan pengembangan website.

1.5 Struktur Dasar Pengembangan Website Profil Kampus

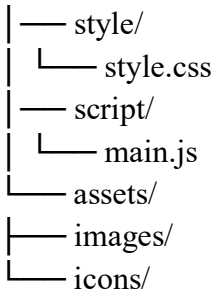
Sebelum membuat website profil kampus, kita perlu memahami struktur komponennya. Umumnya website kampus terdiri dari:

1. *Homepage*
 - Header + Logo + Navigasi
 - Gambar banner atau slider
 - Informasi singkat kampus
 - Link ke halaman penting (PMB, fakultas, kontak)
2. Profil Kampus
 - Sejarah kampus
 - Visi dan Misi
 - Struktur organisasi
3. Fakultas & Program Studi
 - Daftar fakultas
 - Informasi program studi
 - Akreditasi
4. Berita & Pengumuman
 - Artikel kegiatan kampus
 - Informasi PMB
 - Jadwal akademik
5. Galeri Kegiatan
 - Foto-foto event kampus
6. Kontak
 - Alamat
 - Formulir kontak
 - Lokasi (Google Maps embed)

Struktur Folder Proyek

/website-kampus/

```
| — index.html  
| — profil.html  
| — fakultas.html  
| — kontak.html
```



Struktur ini menjadi dasar yang dibangun sepanjang perkuliahan dalam 14 bab ke depan.

KESIMPULAN BAB 1

Pada bab ini mahasiswa memahami:

1. Sejarah dan perkembangan web
2. Cara kerja website dalam model client–server
3. Perbedaan front-end dan back-end
4. Tools dasar yang wajib dikuasai
5. Struktur dasar website profil kampus yang akan menjadi studi kasus utama

BAB 2

DASAR-DASAR HTML

HTML (**HyperText Markup Language**) adalah bahasa markup standar untuk membuat halaman web. HTML tidak bersifat pemrograman, tetapi **memberikan struktur** pada konten web seperti teks, gambar, link, tabel, maupun layout.

Pada bab ini mahasiswa akan memahami elemen dasar HTML sebagai fondasi pembuatan website profil kampus.

2.1. Struktur Dokumen HTML

Setiap halaman HTML memiliki struktur baku yang dimulai dengan deklarasi *doctype* kemudian diikuti elemen dasar `<html>`, `<head>`, dan `<body>`.

Struktur dasar dokumen HTML

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Judul Halaman</title>
</head>
<body>
  <!-- Konten halaman -->
</body>
</html>
```

Penjelasan Bagian-Bagian:

1. `<!DOCTYPE html>`

- Menandakan bahwa dokumen mengikuti standar **HTML5**.
- Wajib ditulis pada baris pertama.

2. `<html>`

- Elemen pembungkus seluruh isi halaman.
- Atribut umum:
- `lang="id"` → menandakan bahasa Indonesia.

3. `<head>`

Berisi informasi *meta* atau konfigurasi halaman, seperti:

- `<title>` judul halaman (muncul di tab browser)
- `<meta charset>` untuk menentukan karakter UTF-8
- `<meta viewport>` agar responsif di perangkat mobile
- Link stylesheet CSS
- Script JavaScript (opsional)

4. `<body>`

Berisi seluruh konten yang tampil pada browser:

- Teks
- Gambar
- Menu
- Formulir
- Footer

2.2. Elemen, Atribut, dan Komentar dalam HTML

Elemen HTML

Elemen adalah komponen pembentuk struktur HTML. Format umumnya:

```
<namatag>konten</namatag>
```

Contoh:

```
<p>Ini adalah paragraf.</p>
```

Atribut HTML

Atribut memberikan informasi tambahan pada elemen. Format:

```
<tag atribut="nilai">konten</tag>
```

Contoh:

```

```

Atribut umum:

- id → identitas unik
- class → pengelompokan elemen (CSS)
- src → sumber gambar atau media
- href → alamat link
- alt → teks alternatif gambar

Komentar HTML

Digunakan untuk menulis catatan dan tidak tampil di browser.

```
<!-- Ini adalah komentar -->
```

2.3. Elemen Teks: Heading, Paragraf dan Daftar (List)

Heading

HTML memiliki 6 tingkat heading:

```
<h1>Judul Utama</h1>  
<h2>Subjudul</h2>  
<h3>Subjudul Tingkat 3</h3>
```

- `<h1>` adalah yang terbesar dan biasanya satu kali per halaman.
- Cocok untuk: Judul kampus, nama fakultas, title halaman.

Paragraf

Menggunakan tag `<p>`:

```
<p>Kampus ABC Berdiri sejak tahun 1980 di kota  
Cilegon.</p>
```

List (Daftar)

- *Unordered List (bullet)*

```
<ul>  
  <li>Fakultas Teknik</li>  
  <li>Fakultas Ekonomi</li>  
</ul>
```

- *Ordered List (nomor)*

```
<ol>  
  <li>Program Studi Informatika</li>  
  <li>Program Studi Sistem Informasi</li>  
</ol>
```

2.4. Link dan Anchor

Link digunakan untuk menghubungkan halaman lain.

Format :

```
<a href="alamat">Teks Link</a>
```

Tipe Link terdiri dari :

1. Link ke halaman lain

```
<a href="profil.html">Profil Kampus</a>
```

2. Link ke halaman luar

```
<a href="https://kemdikbud.go.id"
target="_blank">Website Kemdikbud</a>
```

target="_blank" → membuka tab baru.

3. Link ke bagian tertentu di halaman

```
<a href="#visi">Lihat Visi Kampus</a>
<h2 id="visi">Visi Kampus</h2>
```

2.5. Gambar & Media

Menampilkan Gambar

```

```

Atribut penting:

- src → lokasi gambar
- alt → teks alternatif (SEO & aksesibilitas)
- width / height → ukuran (opsional)

Menampilkan Video

```
<video width="400" controls>
<source src="video-profil.mp4" type="video/mp4">
</video>
```

Menampilkan Audio

```
<audio controls>
<source src="lagu.mp3" type="audio/mpeg">
</audio>
```

2.6. Studi Kasus: Membuat Struktur Halaman Home Website Kampus

Berikut adalah contoh struktur HTML sederhana untuk halaman **Home** website kampus.

Contoh File: **index.html**

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Website Resmi Kampus ABC</title>
</head>
<body>

  <!-- Header dan Navigasi -->
  <header>
    <h1>Kampus ABC</h1>
    <nav>
      <a href="index.html">Beranda</a>
      <a href="profil.html">Profil</a>
      <a href="fakultas.html">Fakultas</a>
      <a href="kontak.html">Kontak</a>
    </nav>
  </header>

  <!-- Banner atau Hero Section -->
  <section>
    <h2>Selamat Datang di Kampus ABC</h2>
    <p>Kampus unggul berbasis teknologi dan
inovasi.</p>
    
  </section>
```

```
<!-- Informasi Singkat -->
<section>
  <h2>Informasi Singkat</h2>
  <p>Kampus ABC berdiri sejak 1980 dan memiliki 4
fakultas unggulan.</p>
  <ul>
    <li>Fakultas Teknik</li>
    <li>Fakultas Ekonomi</li>
    <li>Fakultas Ilmu Komputer</li>
    <li>Fakultas Keguruan</li>
  </ul>
</section>

<!-- Footer -->
<footer>
  <p>&copy; 2025 Kampus ABC. Semua Hak
Dilindungi.</p>
</footer>

</body>
</html>
```

Poin Pembelajaran dari Studi Kasus

Dari contoh di atas, mahasiswa sudah:

- Menggunakan struktur HTML lengkap
- Membuat navigasi
- Menampilkan gambar banner
- Membuat section konten
- Membuat daftar fakultas
- Membuat footer

Struktur ini akan terus dikembangkan pada bab selanjutnya dengan CS

BAB 3

LAYOUT HTML LANJUTAN

Pada bab sebelumnya, mahasiswa telah memahami struktur dasar HTML. Pada bab ini kita akan membahas komponen layout yang lebih kompleks, yaitu table, form, serta semantic HTML5 untuk membuat struktur halaman yang lebih terorganisir dan modern.

3.1. Table, Form, dan Input

1. Table

Table digunakan untuk menampilkan data tabular, misalnya data dosen, jadwal kuliah, atau akreditasi program studi.

Struktur Dasar Table

```
<table>
  <tr>
    <th>Program Studi</th>
    <th>Akreditasi</th>
  </tr>
  <tr>
    <td>Informatika</td>
    <td>A</td>
  </tr>
  <tr>
    <td>Sistem Informasi</td>
    <td>B</td>
  </tr>
</table>
```

Elemen penting dalam tabel

Elemen	Fungsi
<table>	Pembungkus tabel
<tr>	Baris tabel
<th>	Header kolom (bold & center default)
<td>	Isi sel tabel

2. Form dan Input

Form digunakan untuk mengumpulkan data dari pengguna, misalnya form **kontak**, **form pendaftaran PMB**, atau **form login**.

Struktur Form Dasar

```
<form action="proses.php" method="post">
  <label for="nama">Nama:</label>
  <input type="text" id="nama" name="nama">

  <label for="email">Email:</label>
  <input type="email" id="email" name="email">

  <label for="pesan">Pesan:</label>
  <textarea id="pesan" name="pesan"></textarea>

  <button type="submit">Kirim</button>
</form>
```

Jenis Input yang umum digunakan

Type Input	Kode	Penggunaan
Text	<code><input type="text"></code>	Nama, alamat
Email	<code><input type="email"></code>	Validasi email
Number	<code><input type="number"></code>	Umur, jumlah
Password	<code><input type="password"></code>	Login
Date	<code><input type="date"></code>	Tanggal lahir
Radio	<code><input type="radio"></code>	Pilihan tunggal
Checkbox	<code><input type="checkbox"></code>	Beberapa pilihan
File	<code><input type="file"></code>	Upload berkas

3.2. Semantic HTML5

Semantic HTML adalah elemen HTML yang memiliki *makna* untuk menjelaskan struktur halaman.

Penggunaan semantic HTML meningkatkan:

- o Struktur kode
- o SEO (pencarian Google)
- o Aksesibilitas
- o Kemudahan pemeliharaan website

Elemen-Elemen Semantic HTML5

1. `<header>`

Bagian atas halaman : Logo, Judul dan Navigasi.

2. `<nav>`

Tempat menu navigasi:

```
<nav>
  <a href="index.html">Beranda</a>
  <a href="profil.html">Profil</a>
  <a href="fakultas.html">Fakultas</a>
</nav>
```

3. *<section>*

Pembungkus konten besar:

```
<section>
  <h2>Visi Kampus</h2>
  <p>Menjadi perguruan tinggi
  unggul...</p>
</section>
```

4. *<article>*

Konten mandiri (artikel berita, posting blog):

```
<article>
  <h3>Prestasi Mahasiswa 2025</h3>
  <p>Mahasiswa Kampus ABC meraih
  juara nasional...</p>
</article>
```

5. *<aside>*

Konten sampingan:

- Informasi tambahan
- Sidebar
- Link terkait

```
<aside>
  <h3>Informasi Penting</h3>
  <p>Pendaftaran PMB dibuka sampai 30
  Juli.</p>
</aside>
```

6. `<footer>`

Bagian bawah halaman:

```
<footer>
  <p>&copy; 2025 Kampus ABC</p>
</footer>
```

3.3. Accessibility Dasar (A11y)

Aksesibilitas memastikan website dapat digunakan oleh semua orang, termasuk penyandang disabilitas.

1. Alt Text pada Gambar

Untuk pembaca layar (screen reader):

```

```

2. ARIA (Accessible Rich Internet Applications)

Digunakan jika HTML standar belum cukup.

Contoh:

```
<button aria-label="Buka Menu Navigasi">
  ≡
</button>
```

Contoh landmark ARIA:

```
<nav role="navigation">...</nav>
<main role="main">...</main>
```

3. Label yang sesuai pada Input

```
<label for="email">Email:</label>
<input type="email" id="email">
```

4. Kontras warna

Teks harus terbaca meskipun pengguna memiliki keterbatasan visual.

3.4. Studi Kasus: Membuat Layout Halaman “Profil Kampus”

Berikut contoh file lengkap **profil.html** yang menggunakan:

- Semantic HTML5
- Tabel data
- Struktur layout rapi
- Aksesibilitas dasar

Contoh File: profil.html

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Profil Kampus ABC</title>
</head>
<body>

  <!-- Header -->
  <header>
    <h1>Kampus ABC</h1>
    <nav aria-label="Menu Utama">
      <a href="index.html">Beranda</a>
      <a href="profil.html">Profil</a>
      <a href="fakultas.html">Fakultas</a>
      <a href="kontak.html">Kontak</a>
    </nav>
  </header>

  <!-- Konten utama -->
  <main role="main">

    <section>
```

```
<h2>Sejarah Kampus</h2>
<p>Kampus ABC berdiri pada tahun 1980 sebagai
pusat pendidikan unggulan di wilayah Banten.</p>
</section>
```

```
<section>
<h2>Visi dan Misi</h2>
```

```
<article>
<h3>Visi</h3>
<p>Menjadi perguruan tinggi unggul dalam
inovasi dan teknologi.</p>
</article>
```

```
<article>
<h3>Misi</h3>
<ul>
<li>Menghasilkan lulusan berkarakter dan
kompeten.</li>
<li>Mengembangkan riset unggulan.</li>
<li>Berperan aktif dalam pembangunan
masyarakat.</li>
</ul>
</article>
</section>
```

```
<aside>
<h3>Informasi Singkat</h3>
<table border="1" aria-label="Informasi Singkat
Kampus">
<tr>
<th>Tahun Berdiri</th>
<td>1980</td>
</tr>
<tr>
```

```

        <th>Akreditasi Institusi</th>
        <td>A</td>
    </tr>
    <tr>
        <th>Jumlah Fakultas</th>
        <td>4</td>
    </tr>
</table>
</aside>

<section>
    <h2>Foto Kampus</h2>
    
</section>

</main>
<!-- Footer -->
<footer>
    <p>&copy; 2025 Kampus ABC. Semua hak
dilindungi.</p>
</footer>

</body>
</html>

```

Poin Pembelajaran dari Studi Kasus

Pada studi kasus ini mahasiswa telah mempelajari:

- Cara membuat halaman profil dengan layout modern
- Implementasi semantic HTML5
- Penggunaan table untuk informasi ringkas
- Penggunaan aside untuk sidebar
- Penerapan alt text & ARIA
- Penerapan struktur halaman profesional

BAB 4

PENGENALAN CSS (*Cascading Style Sheets*)

CSS (**Cascading Style Sheets**) adalah bahasa untuk mengatur tampilan (presentasi) dari dokumen HTML. Jika HTML berfungsi sebagai **struktur**, maka CSS berfungsi sebagai **desain** atau **gaya visual**.

Dengan CSS, kita dapat mengatur:

- warna
- ukuran teks
- tata letak (layout)
- jarak antar elemen
- background
- border
- responsivitas

4.1. Apa Itu CSS dan Cara Kerjanya

CSS adalah bahasa stylesheet yang digunakan untuk memisahkan tampilan dari struktur HTML. CSS menentukan bagaimana elemen-elemen HTML ditampilkan di browser.

Fungsi utama CSS:

- Mengatur tampilan visual halaman web.
- Memisahkan struktur dan desain untuk memudahkan pemeliharaan kode.
- Meningkatkan konsistensi, karena satu file CSS dapat digunakan oleh banyak halaman.
- Memungkinkan layout modern seperti flexbox & grid.

Cara Kerja CSS

Browser membaca HTML dari atas ke bawah, dan saat menemukan CSS, browser menerapkan aturan CSS ke elemen-elemen yang cocok.

Struktur dasar CSS:

```
selektor {  
  properti: nilai;  
}  
Contoh:  
p {  
  color: blue;  
  font-size: 16px;  
}
```

Artinya: semua elemen `<p>` diberi warna biru dan ukuran font 16px.

4.2. Cara Menyisipkan CSS

Ada 3 cara menambahkan CSS ke halaman HTML:

1. CSS Inline

Ditulis langsung di atribut elemen HTML.

```
<p style="color:red;">Ini teks berwarna merah.</p>
```

Kelebihan : cepat dan mudah

Kekurangan : tidak efisien, sulit di-maintain

2. CSS Internal

Ditulis dalam tag `<style>` di dalam `<head>`.

```
<head>  
<style>  
  h1 {  
    color: navy;  
  }  
</style>  
</head>
```

Kelebihan : cocok untuk halaman kecil
Kekurangan : tidak reusable ke halaman lain

3. CSS External (Dianjurkan)

Disimpan dalam file .css terpisah.

File HTML

```
<link rel="stylesheet" href="style/style.css">
```

File style.css

```
body {  
    background-color: #f0f0f0;  
}
```

Kelebihan terbesar:

- digunakan banyak halaman
- mudah dikelola
- lebih profesional

4.3. Selektor CSS Dasar

Selektor digunakan untuk memilih elemen HTML yang ingin diberi gaya.

1. Selektor Tag (Element Selector)

Mengubah semua elemen dengan nama tag tertentu.

```
p {  
    color: green;  
}
```

2. Selektor Class

Menggunakan tanda .

Dipakai untuk memberikan gaya pada banyak elemen.

HTML:

```
<p class="highlight">Teks penting.</p>
```

CSS:

```
.highlight {  
    background-color: yellow;  
}
```

3. Selektor ID

Menggunakan tanda #

Hanya untuk satu elemen unik.

HTML:

```
<h1 id="judul">Kampus ABC</h1>
```

CSS:

```
#judul {  
  color: blue;  
}
```

4. Selektor Universal

Menerapkan gaya ke semua elemen.

```
* {  
  margin: 0;  
  padding: 0;  
}
```

5. Selektor Descendant (turunan)

```
nav a {  
  color: white;  
}
```

Artinya: semua <a> di dalam <nav>.

4.4. Properti CSS Umum

Berikut properti dasar yang paling sering digunakan:

1. Color (warna teks)

```
p {  
  color: #333;  
}
```

2. Background

```
header {  
  background-color: #0033cc;  
}
```

Untuk gambar:

```
section {
```

```
background-image: url('banner.jpg');  
}
```

3. Margin (jarak luar elemen)

```
div {  
  margin: 20px;  
}
```

Margin bisa berbeda:

```
margin-top: 10px;  
margin-bottom: 20px;  
margin-left: 15px;  
margin-right: 15px;
```

4. Padding (jarak dalam elemen)

```
div {  
  padding: 15px;  
}
```

5. Border

```
p {  
  border: 1px solid black;  
}
```

Format border:

border: [ketebalan] [jenis] [warna]

4.5. Studi Kasus: Styling Halaman Beranda Sederhana

Kita akan memodifikasi halaman index.html dari bab sebelumnya menggunakan CSS.

File 1: index.html

```
<!DOCTYPE html>  
<html lang="id">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-  
width, initial-scale=1.0">  
  <title>Beranda Kampus ABC</title>  
  <link rel="stylesheet" href="style/style.css">
```

```

</head>
<body>

  <header>
    <h1>Kampus ABC</h1>
    <nav>
      <a href="index.html">Beranda</a>
      <a href="profil.html">Profil</a>
      <a href="fakultas.html">Fakultas</a>
      <a href="kontak.html">Kontak</a>
    </nav>
  </header>

  <section class="hero">
    <h2>Selamat Datang di Kampus ABC</h2>
    <p>Kampus unggul berbasis teknologi dan
inovasi.</p>
  </section>

  <footer>
    <p>&copy; 2025 Kampus ABC</p>
  </footer>

</body>
</html>

```

File 2: style/style.css

```

/* Reset margin dan padding */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* Styling body */

```

```
body {
    font-family: Arial, sans-serif;
    background-color: #f5f5f5;
}

/* Header */
header {
    background-color: #0033cc;
    color: white;
    padding: 20px;
}

header h1 {
    margin-bottom: 10px;
}

/* Navigasi */
nav a {
    color: white;
    margin-right: 15px;
    text-decoration: none;
}

nav a:hover {
    text-decoration: underline;
}

/* Hero section */
.hero {
    background-color: white;
    padding: 40px;
    margin: 20px;
    border-radius: 10px;
    text-align: center;
}
```

```
.hero h2 {
  color: #0033cc;
  margin-bottom: 10px;
}
footer {
  background-color: #0033cc;
  color: white;
  text-align: center;
  padding: 10px;
}
```

Hasil Styling

Halaman beranda kini memiliki:

- Header berwarna biru kampus
- Navigasi rapi dengan efek hover
- Section hero dengan background putih & border radius
- Layout yang lebih modern dan nyaman dilihat

KESIMPULAN BAB 4

Mahasiswa telah memahami:

1. Konsep dasar CSS dan cara kerjanya
2. Tiga cara menyisipkan CSS
3. Selektor dasar: tag, class, id, descendant
4. Properti umum: color, background, margin, padding, border
5. Menerapkan CSS untuk halaman beranda sederhana

Bab ini menjadi landasan untuk bab berikutnya:
Layout modern dengan Flexbox dan Grid.

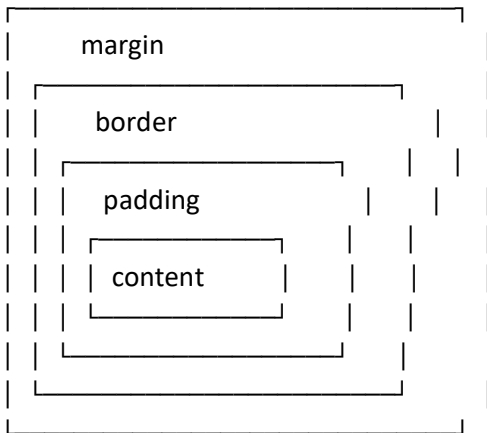
BAB 5

LAYOUT WEBSITE DENGAN CSS

Bab ini membahas konsep dasar layout yang menjadi fondasi dalam mendesain tampilan web. Mahasiswa akan mempelajari box model, display, positioning, float, dan clear, serta menerapkannya dalam membangun layout 2 kolom untuk halaman Profil Kampus.

5.1. Box Model dalam CSS

Box Model adalah konsep dasar dalam CSS yang menyatakan bahwa setiap elemen HTML dianggap sebagai sebuah kotak yang terdiri dari:



Elemen Box Model

- Content → isi elemen (teks, gambar)
- Padding → jarak antara content dan border
- Border → garis batas elemen
- Margin → jarak luar antar elemen

Contoh CSS Box Model

```
.card {  
  width: 300px;  
  padding: 20px;  
  border: 2px solid #0033cc;  
  margin: 15px;  
}
```

5.2. Display: block, inline, dan inline-block

Properti display menentukan bagaimana elemen HTML ditampilkan di halaman.

1. display: block

- Selalu memenuhi satu baris penuh (full width)
- Bisa diberi width, height, margin, padding
- Contoh tag block: <div>, <p>, <section>, <h1>

```
div {  
  display: block;  
}
```

2. display: inline

- Tidak memulai baris baru
- Lebarinya mengikuti konten
- Tidak bisa diberi width/height
- Contoh tag inline: <a>, ,

```
a {  
  display: inline;  
}
```

3. display: inline-block

- Kombinasi inline + block

- Dapat ditetapkan width/height
- Tetap tidak memulai baris baru

```
.nav-item {
    display: inline-block;
    width: 100px;
}
```

5.3. Positioning

Properti position mengatur bagaimana elemen ditempatkan relatif terhadap halaman atau elemen lainnya.

Jenis-jenis Positioning, diantaranya sebagai berikut :

1. static (default)

Posisi mengikuti alur normal dokumen.

```
div {
    position: static;
}
```

2. relative

Bergerak relatif terhadap posisi aslinya.

```
.box {
    position: relative;
    top: 10px;
    left: 20px;
}
```

3. absolute

Posisi ditentukan relatif terhadap parent yang memiliki position: relative.

```
.child {
    position: absolute;
    top: 0;
    right: 0;
```

- ```
}
4. fixed
Tetap berada di posisi tertentu meski halaman di-scroll.
```

```
.header {
 position: fixed;
 top: 0;
 width: 100%;
}
```

- ```
5. sticky  
Bergantung pada scroll: awalnya normal, lalu menempel  
saat melewati area tertentu.
```

```
nav {  
    position: sticky;  
    top: 0;  
}
```

5.4. Float dan Clear

Sebelum Flexbox dan Grid menjadi standar modern, layout website dahulu menggunakan float.

Float

Mengatur elemen agar “mengambang” ke kiri atau kanan.

```
.left {  
    float: left;  
}  
.right {  
    float: right;  
}
```

Clear

Mencegah elemen lain mengapung (float) menempel pada elemen tertentu.

```
.clearfix {  
    clear: both;  
}
```

Masalah Float

- Tidak dirancang untuk layout kompleks
- Konten bisa tumpang tindih Namun konsepnya tetap penting untuk memahami warisan layouting HTML/CSS.

5.5. Studi Kasus: Layout 2 Kolom untuk Halaman Profil Kampus

Kita akan membuat dua kolom:

- Kolom kiri → informasi profil
- Kolom kanan → sidebar (informasi singkat)

File: profil.html

```
<!DOCTYPE html>  
<html lang="id">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width,  
initial-scale=1.0">  
    <title>Profil Kampus ABC</title>  
    <link rel="stylesheet" href="style/style.css">  
</head>  
<body>  
  
    <header>  
        <h1>Kampus ABC</h1>
```

```

<nav>
  <a href="index.html">Beranda</a>
  <a href="profil.html">Profil</a>
  <a href="fakultas.html">Fakultas</a>
  <a href="kontak.html">Kontak</a>
</nav>
</header>

<main class="container">

  <section class="content">
    <h2>Profil Kampus</h2>
    <p>Kampus ABC berdiri pada tahun 1980 sebagai pusat
pendidikan unggulan di wilayah Banten.</p>
    <p>Kampus ini memiliki 4 fakultas dengan akreditasi
institusi A.</p>
  </section>

  <aside class="sidebar">
    <h3>Informasi Singkat</h3>
    <ul>
      <li>Tahun Berdiri: 1980</li>
      <li>Akreditasi: A</li>
      <li>Jumlah Fakultas: 4</li>
    </ul>
  </aside>

  <div class="clearfix"></div>

</main>

</body>
</html>

```

File: style/style.css

```
/* Layout dasar container */
.container {
    width: 90%;
    margin: 20px auto;
}

/* Kolom kiri (konten utama) */
.content {
    width: 65%;
    float: left;
    margin-right: 5%;
}

/* Kolom kanan (sidebar) */
.sidebar {
    width: 30%;
    float: left;
    background-color: #f5f5f5;
    padding: 15px;
    border-radius: 8px;
}

/* Clearfix agar float tidak merusak layout */
.clearfix {
    clear: both;
}

/* Styling tambahan */
header {
    background-color: #0033cc;
    color: white;
    padding: 20px;
}
```

```
nav a {
    color: white;
    margin-right: 15px;
    text-decoration: none;
}

main h2 {
    color: #0033cc;
}

.sidebar h3 {
    color: #0033cc;
}
```

Penjelasan Layout

1. float: left; membuat dua kolom sejajar
2. Persentase width memastikan layout responsif
3. Margin-right memberi jarak antar kolom
4. .clearfix mencegah konten di bawahnya naik ke samping float

Tampilan Akhir Layout 2 Kolom

- Kolom kiri berisi teks profil
- Kolom kanan berisi informasi singkat
- Tampilan bersih, rapi, dan mudah dipahami
- Cocok untuk halaman Profil Kampus

KESIMPULAN BAB 5

Mahasiswa telah memahami:

1. Konsep box model sebagai dasar layout
2. Perbedaan display: block, inline, inline-block
3. Cara kerja position: static, relative, absolute, fixed, sticky
4. Teknik float & clear
5. Menerapkan layout 2 kolom menggunakan float

Bab ini membuka jalan menuju layout modern seperti Flexbox dan CSS Grid.

BAB 6

DESAIN RESPONSIF DENGAN FLEXBOX

Pada era saat ini, mayoritas pengguna mengakses website melalui perangkat mobile. Oleh sebab itu, developer harus memastikan bahwa tampilan web dapat menyesuaikan ukuran layar secara otomatis. Pada bab ini, kita mempelajari **responsive design**, **media query**, dan layout modern menggunakan **Flexbox**.

6.1. Konsep Responsive Design

Responsive design adalah pendekatan desain web di mana tata letak website dapat menyesuaikan ukuran layar perangkat:

- Smartphone (320–480px)
- Tablet (768px)
- Laptop (1024px)
- Desktop (>1200px)

Tujuan responsive design:

- Tampilan seragam di berbagai perangkat
- Navigasi mudah meski layar kecil
- Menghindari scroll horizontal
- Website lebih cepat dan ramah pengguna

Prinsip-prinsip Responsive Design

- Layout menggunakan persentase, bukan pixel tetap
- Gambar responsif menggunakan max-width: 100%
- Menggunakan media query
- Menggunakan Flexbox atau CSS Grid

6.2. Media Query

Media query adalah fitur CSS yang digunakan untuk mengatur gaya berdasarkan ukuran layar atau perangkat tertentu.

Contoh Media Query Dasar

```
@media (max-width: 768px) {  
  body {  
    background-color: lightgray;  
  }  
}
```

Artinya: Jika layar lebih kecil atau sama dengan 768px → ganti background menjadi abu-abu.

Media Query untuk Mobile, Tablet, Desktop

```
/* Mobile */
```

```
@media (max-width: 576px) { }
```

```
/* Tablet */
```

```
@media (min-width: 577px) and (max-width: 992px) { }
```

```
/* Desktop */
```

```
@media (min-width: 993px) { }
```

6.3. Flexbox Lengkap

Flexbox (**Flexible Box Layout**) adalah metode modern untuk mengatur tata letak elemen secara fleksibel.

Dengan Flexbox, kita dapat dengan mudah membuat:

- Layout 2 atau 3 kolom
- Navigasi responsif
- Alignment elemen secara horizontal/vertical
- Distribusi ruang otomatis

Cara Mengaktifkan Flexbox

```
.container {  
  display: flex;  
}
```

Properti Flex Container

1. flex-direction

```
flex-direction: row; /* default */
flex-direction: column;
```
2. justify-content (mengatur posisi horizontal)

```
justify-content: flex-start;
justify-content: center;
justify-content: flex-end;
justify-content: space-between;
justify-content: space-around;
```
3. align-items (mengatur posisi vertikal)

```
align-items: flex-start;
align-items: center;
align-items: flex-end;
```
4. flex-wrap

```
flex-wrap: wrap;
```

Memungkinkan elemen turun ke baris berikutnya.

Properti Flex Item

1. flex-grow
Menentukan seberapa besar elemen mengisi ruang kosong.

```
.item {
  flex-grow: 1;
}
```
2. flex-basis
Menentukan ukuran dasar elemen.

```
.item {
  flex-basis: 200px;
}
```
3. flex
Shortcut dari:

```
flex: grow shrink basis;
```

Contoh:

```
.item {  
  flex: 1 0 200px;  
}
```

6.4. Studi Kasus: Membuat Menu Navigasi Responsif dengan Flexbox

Tujuan studi kasus:

- Menu horizontal di desktop
- Menu berubah menjadi vertikal di mobile
- Tombol “hamburger” muncul di layar kecil

Struktur File: index.html

```
<!DOCTYPE html>  
<html lang="id">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-  
width, initial-scale=1.0">  
  <title>Kampus ABC</title>  
  <link rel="stylesheet" href="style/style.css">  
</head>  
<body>  
  
  <header class="header">  
    <h1>Kampus ABC</h1>  
  
    <button class="menu-toggle" aria-label="Buka  
Menu">  
      ≡  
</button>  
  
    <nav class="nav">  
      <a href="index.html">Beranda</a>  
      <a href="profil.html">Profil</a>
```

```

        <a href="fakultas.html">Fakultas</a>
        <a href="kontak.html">Kontak</a>
    </nav>
</header>

<script src="script/menu.js"></script>
</body>
</html>

```

File CSS: style/style.css

1. Styling Dasar Header

```

.header {
    background-color: #0033cc;
    color: white;
    padding: 15px;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

```

2. Menu Navigasi (Desktop)

```

.nav {
    display: flex;
    gap: 20px;
}
.nav a {
    color: white;
    text-decoration: none;
}

.menu-toggle {
    display: none;
    font-size: 24px;
    background: none;
    border: none;
    color: white;
}

```

```

    }
3. Responsif untuk Mobile (Media Query)
    @media (max-width: 768px) {
/* Tombol hamburger muncul */
    .menu-toggle {
        display: block;
    }
/* Menu disembunyikan di mobile */
    .nav {
        display: none;
        flex-direction: column;
        background-color: #0033cc;
        padding: 10px;
        margin-top: 10px;
    }
/* Menu muncul saat aktif */
    .nav.active {
        display: flex;
    }
}

```

File JavaScript: script/menu.js

Untuk menampilkan menu ketika tombol hamburger diklik.

```

const toggle = document.querySelector(".menu-toggle");
const nav = document.querySelector(".nav");

toggle.addEventListener("click", () => {
    nav.classList.toggle("active");
});

```

Penjelasan Studi Kasus

1. Desktop Mode
 - Flexbox menempatkan logo di kiri
 - Menu navigasi di kanan
 - Items berada dalam 1 baris dengan gap
2. Mobile Mode
 - Flexbox diubah menjadi column
 - Menu disembunyikan (display: none)
 - Tombol hamburger tampil
 - Menu muncul saat tombol ditekan
3. Hasil Akhir: Menu Navigasi Responsif
 - Tampilan elegan di layar besar
 - Navigasi sederhana di layar kecil
 - Menerapkan Flexbox + Media Query
 - Menggunakan JavaScript minimal

KESIMPULAN BAB 6

Mahasiswa telah memahami:

1. Memahami konsep responsive design dengan benar
2. Menulis CSS dengan media query untuk mobile, tablet, desktop
3. Menggunakan Flexbox untuk membuat layout modern
4. Membangun navigasi responsif seperti website profesional

Bab ini sangat penting karena semua website modern mengandalkan teknik Flexbox untuk struktur layout utama.

BAB 7

CSS GRID UNTUK LAYOUT MODERN

CSS Grid adalah sistem layout dua dimensi yang memungkinkan pengembang web mengatur struktur halaman secara presisi: baik secara **baris (rows)** maupun **kolom (columns)**. Grid sangat ideal untuk membuat layout kompleks seperti galeri, dashboard, katalog fakultas, dan halaman program studi.

Pada bab ini mahasiswa akan belajar konsep dasar Grid, cara mengatur baris & kolom, hingga membangun layout multiseksi menggunakan studi kasus website kampus.

7.1. Dasar CSS Grid

Untuk menggunakan CSS Grid, tentukan suatu elemen sebagai **grid container** dengan `display: grid`.

Contoh paling sederhana:

```
.container {  
  display: grid;  
}
```

Setelah menjadi grid, semua elemen di dalamnya disebut **grid item**.

Konsep Utama CSS Grid

- **Grid Container** → elemen induk yang menggunakan `display: grid`
- **Grid Items** → elemen-elemen di dalam container
- **Grid Lines** → garis pembatas baris/kolom

- **Grid Tracks** → ruang antara dua grid line
- **Grid Areas** → area besar yang dapat ditempati item

7.2. Grid Rows & Columns

CSS Grid menggunakan properti:

- `grid-template-columns`
- `grid-template-rows`

Untuk menentukan jumlah dan ukuran kolom/baris.

1. `grid-template-columns`

Contoh 2 kolom sama besar:

```
.container {
  display: grid;
  grid-template-columns: 1fr 1fr;
}
```

`fr` (fraction unit) → satuan khusus untuk grid.

Contoh 3 kolom dengan ukuran berbeda:

```
grid-template-columns: 200px 1fr 2fr;
```

2. `grid-template-rows`

```
grid-template-rows: 100px 200px auto;
```

Grid akan memiliki 3 baris.

3. Gap Antar Kolom/Baris

```
grid-gap: 20px;
```

Atau terpisah:

```
row-gap: 20px;
```

```
column-gap: 10px;
```

4. Menentukan posisi item secara manual

Column span (menggabungkan kolom)

```
.item1 {
  grid-column: 1 / 3; /* dari kolom 1 hingga sebelum
  kolom 3 */
}
```

Row span (menggabungkan baris)

```
.item1 {
  grid-row: 1 / 2;
}
```

7.3. Penataan Konten Multiseksi dengan Grid

CSS Grid sangat ideal untuk menata beberapa bagian halaman sekaligus (hero, konten utama, sidebar, footer).

Contoh layout 3 kolom responsif:

```
.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-gap: 20px;
}
```

Penjelasan:

- $\text{repeat}(3, 1\text{fr}) = 1\text{fr} + 1\text{fr} + 1\text{fr}$
- Cocok untuk menampilkan daftar program studi, kartu fakultas, atau galeri kegiatan.

Grid Areas

Dengan grid template area, desain layout menjadi lebih jelas.

CSS

```
.container {
  display: grid;
  grid-template-areas:
    "header header header"
    "sidebar content content"
    "footer footer footer";
  grid-template-columns: 200px 1fr 1fr;
  grid-gap: 20px;
}
```

```
header { grid-area: header; }
aside { grid-area: sidebar; }
main { grid-area: content; }
footer { grid-area: footer; }
```

HTML

```
<div class="container">
  <header>Header</header>
  <aside>Sidebar</aside>
  <main>Konten Utama</main>
  <footer>Footer</footer>
</div>
```

7.4. Studi Kasus: Halaman Fakultas & Program Studi Menggunakan CSS Grid

Pada studi kasus ini, kita membangun halaman fakultas.html berisi daftar fakultas dan program studi dalam bentuk kartu (card) menggunakan grid.

Struktur File: fakultas.html

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Fakultas & Program Studi – Kampus ABC</title>
  <link rel="stylesheet" href="style/style.css">
</head>
<body>

  <header class="header">
    <h1>Kampus ABC</h1>
    <nav>
```

```
    <a href="index.html">Beranda</a>
    <a href="profil.html">Profil</a>
    <a href="fakultas.html">Fakultas</a>
    <a href="kontak.html">Kontak</a>
  </nav>
</header>
```

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

```
  <h2>Fakultas & Program Studi</h2>
```

```
  <div class="grid-fakultas">
```

```
    <div class="card">
      <h3>Fakultas Teknik</h3>
      <ul>
        <li>Teknik Mesin</li>
        <li>Teknik Industri</li>
        <li>Teknik Elektro</li>
      </ul>
    </div>
```

```
    <div class="card">
      <h3>Fakultas Ekonomi</h3>
      <ul>
        <li>Manajemen</li>
        <li>Akuntansi</li>
      </ul>
    </div>
```

```
    <div class="card">
      <h3>Fakultas Ilmu Komputer</h3>
      <ul>
        <li>Informatika</li>
        <li>Sistem Informasi</li>
      </ul>
    </div>
```

```

        </ul>
    </div>

    <div class="card">
        <h3>Fakultas Keguruan</h3>
        <ul>
            <li>Pendidikan Matematika</li>
            <li>Pendidikan Bahasa Inggris</li>
        </ul>
    </div>

</div>

</main>

</body>
</html>

```

File CSS: style/style.css

1. Styling Grid Fakultas

```

.grid-fakultas {
    margin-top: 20px;
    display: grid;
    grid-template-columns: repeat(auto-fit,
minmax(250px, 1fr));
    grid-gap: 20px;
}

```

Penjelasan:

- auto-fit → grid otomatis menyesuaikan jumlah kolom berdasarkan ukuran layar
- minmax(250px, 1fr) → lebar minimal 250px, maksimal 1fr
- Layout otomatis responsif tanpa media query

2. Styling Card Fakultas

```
.card {
  background-color: #ffffff;
  border-radius: 10px;
  padding: 20px;
  border: 1px solid #dcdcdc;
  box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}

.card h3 {
  color: #0033cc;
  margin-bottom: 10px;
}

.card ul {
  padding-left: 20px;
}
```

3. Header dan Navigasi

```
.header {
  background-color: #0033cc;
  color: white;
  padding: 20px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

nav a {
  color: white;
  margin-left: 15px;
  text-decoration: none;
}
```

Penjelasan Hasil Studi Kasus

Dengan CSS Grid:

- Layout otomatis responsif
Grid akan menampilkan 4 kolom di desktop, 2 kolom di tablet, dan 1 kolom di smartphone tanpa harus menulis banyak media query.
- Tampilan modern
Kartu fakultas terlihat rapi, berjarak, dan seimbang.
- Penulisan kode efisien
Grid memungkinkan desain kompleks dengan sedikit kode.

KESIMPULAN BAB 7

Mahasiswa telah mempelajari:

1. Dasar-dasar CSS Grid
2. Mengatur baris dan kolom grid
3. Mengatur layout multi-bagian dengan grid
4. Membangun halaman Fakultas & Program Studi berbasis grid responsif

CSS Grid adalah teknologi powerful yang menggantikan teknik lama seperti float atau table layout, dan merupakan standar modern dalam web design saat ini.

BAB 8

DESAIN MODERN & UI/UX DASAR

Desain modern untuk web tidak hanya tentang visual yang menarik, tetapi juga tentang **pengalaman pengguna (User Experience/UX)**. Website kampus—sebagai portal informasi resmi—wajib memiliki tampilan yang bersih, profesional, dan mudah di-navigasi.

Pada bab ini mahasiswa belajar prinsip desain, pemilihan warna & tipografi, pendekatan *component-based*, dan penggunaan animasi/transition sederhana.

8.1.Prinsip Desain Sederhana (Simple & Clean Design)

Desain modern identik dengan:

- Minimalis
- Tata letak rapi
- Fokus pada informasi penting
- Lebih banyak ruang kosong (white space)

Sedangkan prinsip desain sederhana sebagai berikut :

1. White Space

Ruang kosong membuat konten lebih fokus dan nyaman dibaca.

```
section {  
  padding: 40px;  
}
```

2. Hierarki Visual

Gunakan ukuran font, warna, dan jarak untuk membedakan mana yang penting.

Contoh:

- a. <h1> lebih besar dari <h2>
- b. Judul menggunakan warna tegas
- c. Paragraf cukup menggunakan warna gelap tetapi soft (#333)

3. Konsistensi Desain

Gunakan:

- d. Satu gaya tombol
- e. Satu gaya kartu (card)
- f. Warna konsisten untuk brand

Contoh:

Buruk : setiap halaman menggunakan warna yang berbeda

Bagus : website kampus menggunakan warna biru navy sebagai identitas kampus

4. Navigasi Mudah Dipahami

- g. Menu harus terlihat jelas
- h. Struktur navigasi tidak membingungkan
- i. Tombol atau link penting harus mudah ditemukan

5. Mobile-first

Desain dari layar kecil ke layar besar agar responsif.

8.2. Pemilihan Warna dan Tipografi

Warna dan font sangat mempengaruhi nuansa website.

1. Pemilihan Warna (Color Scheme)

- a. Warna Primer
Warna utama untuk identitas, misalnya:
 - o Biru kampus: #0033cc
 - o Biru muda: #007bff
- b. Warna Sekunder

Untuk aksen atau bagian pendukung.

c. Warna Netral

Digunakan sebagai background:

- Putih
- Abu muda #f5f5f5
- Abu gelap #333

Contoh palet warna kampus modern

- Primer: #0033cc
- Sekunder: #0055ff
- Accent: #00bfff
- Background: #f5f5f5
- Text: #333

2. Tipografi (Pemilihan Font)

Font Sans-serif modern (direkomendasikan)

- Poppins
- Open Sans
- Roboto
- Lato
- Inter

Contoh import dari Google Fonts:

```
@import  
url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&display=swap');
```

```
body {  
  font-family: 'Poppins', sans-serif;  
}
```

Aturan tipografi:

- Gunakan maksimum 2 tipe font
- Ukuran heading harus konsisten
- Jarak line-height ideal 1.5

- Gunakan bold untuk penekanan

8.3.Component-Based Design (Desain Berbasis Komponen)

Website modern dibangun dari **komponen kecil** yang dapat digunakan ulang (reusable components).

Contoh komponen:

- Card (kartu fakultas)
- Button
- Navbar
- Footer
- Hero Section
- Form input

Dengan pendekatan ini:

- Desain konsisten
- Pengembangan lebih cepat
- Mudah diperbaiki

Contoh: Komponen Card Fakultas

```
<div class="card">
  <h3>Fakultas Teknik</h3>
  <p>3 Program Studi</p>
</div>
.card {
  background-color: #ffffff;
  border-radius: 10px;
  padding: 20px;
  box-shadow: 0 3px 8px rgba(0,0,0,0.1);
  transition: transform 0.3s;
}
.card:hover {
  transform: translateY(-5px);
}
```

Dengan sedikit kode, komponen ini bisa digunakan pada 10+ halaman berbeda.

8.4. Animasi dan Transition CSS

Animasi ringan menjadikan website lebih interaktif.

1. Transition

Digunakan untuk animasi sederhana saat berubah state (hover, focus).

```
button {
  background-color: #0033cc;
  color: white;
  padding: 12px 20px;
  border-radius: 8px;
  transition: background-color 0.3s ease;
}

button:hover {
  background-color: #0055ff;
}
```

2. Transform

Untuk pergerakan & efek.

```
.card:hover {
  transform: translateY(-10px) scale(1.02);
}
```

3. Animation + Keyframes

Untuk animasi berulang.

```
@keyframes fadeln {
  from { opacity: 0; }
  to { opacity: 1; }
}

.hero {
  animation: fadeln 1s ease;
}
```

8.5. Studi Kasus: Mempercantik Website Kampus

Kita akan mempercantik halaman beranda (index.html) menggunakan prinsip modern design, tipografi, dan animasi.

File: index.html

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Kampus ABC</title>
  <link rel="stylesheet" href="style/style.css">
</head>
<body>

<header class="header">
  <h1>Kampus ABC</h1>
  <nav>
    <a href="index.html">Beranda</a>
    <a href="profil.html">Profil</a>
    <a href="fakultas.html">Fakultas</a>
    <a href="kontak.html">Kontak</a>
  </nav>
</header>

<section class="hero">
  <h2>Selamat Datang di Kampus ABC</h2>
  <p>Kampus unggul berbasis teknologi dan inovasi.</p>
</section>
<section class="info">
  <div class="card">
    <h3>Fakultas Teknik</h3>
    <p>3 Program Studi</p>
```

```

</div>

<div class="card">
  <h3>Fakultas Ekonomi</h3>
  <p>2 Program Studi</p>
</div>

<div class="card">
  <h3>Fakultas Ilmu Komputer</h3>
  <p>2 Program Studi</p>
</div>
</section>
<footer class="footer">
  <p>&copy; 2025 Kampus ABC</p>
</footer>
</body>
</html>

```

File: style/style.css

1. Google Font

```

@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&display=swap');
body {
  font-family: 'Poppins', sans-serif;
  background-color: #f5f5f5;
  margin: 0;
}

```

2. Header Modern

```

.header {
  background-color: #0033cc;
  color: white;
  padding: 20px;
  display: flex;
  justify-content: space-between;
}

```

```

        align-items: center;
    }
    .header nav a {
        color: white;
        text-decoration: none;
        margin-left: 15px;
        font-weight: 500;
    }

```

3. Hero Section dengan Animasi

```

.hero {
    text-align: center;
    padding: 60px 20px;
    background-color: white;
    margin: 20px;
    border-radius: 12px;
    animation: fadeIn 1.2s ease;
}
@keyframes fadeIn {
    from { opacity: 0; transform: translateY(20px); }
    to { opacity: 1; transform: translateY(0); }
}

.hero h2 {
    color: #0033cc;
}

```

4. Kartu Fakultas (Component)

```

.info {
    display: flex;
    gap: 20px;
    justify-content: center;
    padding: 20px;
}
.card {
    width: 250px;
    background-color: white;
}

```

```

padding: 20px;
border-radius: 12px;
text-align: center;
box-shadow: 0 4px 10px rgba(0,0,0,0.1);
transition: transform .3s ease, box-shadow .3s ease;
}
.card:hover {
transform: translateY(-8px);
box-shadow: 0 6px 15px rgba(0,0,0,0.15);
}
.card h3 {
color: #0033cc;
}

```

5. Footer

```

.footer {
background-color: #0033cc;
color: white;
text-align: center;
padding: 15px;
}

```

Manfaat Studi Kasus Ini

Dengan menerapkan desain modern ini, website kampus menjadi:

- Lebih profesional
- Lebih konsisten
- Mudah dibaca dan dipahami
- Lebih interaktif dengan animasi ringan
- Siap digunakan sebagai template portofolio mahasiswa

KESIMPULAN BAB 8

Mahasiswa telah mempelajari:

1. Prinsip desain simpel dan profesional
2. Pemilihan warna dan tipografi yang tepat
3. Konsep component-based design
4. Animasi dan transition CSS untuk interaksi modern
5. Menerapkan semua konsep pada halaman beranda website kampus

Bab ini menjadi fondasi untuk styling lanjut pada bab-bab berikutnya.

BAB 9

Pengenalan JavaScript

JavaScript (JS) adalah bahasa pemrograman yang berjalan di browser dan membuat halaman web menjadi dinamis, interaktif, dan cerdas. HTML menyediakan struktur halaman, CSS memberikan style, sedangkan JavaScript memberikan logika.

Pada bab ini mahasiswa akan mempelajari dasar-dasar JavaScript: variabel, tipe data, operator, alur kontrol (if dan looping), fungsi, dan scope. Di akhir bab, mahasiswa membuat **validasi form Pendaftaran Mahasiswa Baru (PMB)**.

9.1. Apa Itu JavaScript & Perannya

JavaScript adalah bahasa pemrograman yang berjalan di sisi klien (browser) dan dapat:

- Mengubah konten halaman secara dinamis
- Merespon input pengguna
- Mengatur animasi & interaksi
- Memvalidasi form sebelum dikirim
- Mengambil data dari server (AJAX / Fetch API)
- Membuat aplikasi web full seperti SPA (React, Vue, Angular)

Peran JavaScript dalam Website Modern

1. Interaktivitas
Contoh: slider gambar, popup, menu mobile.
2. Manipulasi DOM
Mengubah teks, gambar, dan elemen HTML saat runtime.

3. Validasi Form
Mengecek apakah input sudah benar sebelum dikirim ke server.
4. Pengelolaan Data
Menggunakan array, objek, dan penyimpanan browser (localStorage).
5. Komunikasi dengan server
Mengambil data berita, daftar fakultas, dll.

JavaScript adalah bahasa inti web modern dan wajib dikuasai oleh semua web developer.

9.2. Variabel, Tipe Data, dan Operator

Variabel

Variabel digunakan untuk menyimpan nilai. JavaScript menggunakan tiga cara deklarasi:

```
var nama = "Roy"; // cara lama
let umur = 25; // dianjurkan
const kampus = "ABC"; // nilai tetap
```

Rekomendasi modern:

- Gunakan **let** untuk nilai yang berubah
- Gunakan **const** untuk nilai tetap

Tipe Data JavaScript

Tipe Data	Contoh	Penjelasan
String	"Informatika"	Teks
Number	20, 3.14	Angka
Boolean	true, false	Logika
Array	["A", "B", "C"]	Kumpulan data
Object	{nama:"Roy", umur:25}	Data berbentuk pasangan key-value

Tipe Data	Contoh	Penjelasan
Null	null	Kosong
Undefined	undefined	Belum diberi nilai

Operator

1. Operator Aritmatika

```
let x = 10 + 5; // 15
let y = 10 * 2; // 20
```

2. Operator Perbandingan

```
5 > 3 // true
10 == "10" // true
10 === "10" // false (membandingkan tipe juga)
```

3. Operator Logika

```
true && false // false
true || false // true
!true // false
```

9.3. Control Flow (if, else, loop)

Control flow digunakan agar kode bisa mengambil keputusan atau mengulang perintah.

1. If / Else

```
let nilai = 85;

if (nilai >= 80) {
  console.log("A");
} else {
  console.log("B");
}
```

```
}
```

2. Else If

```
if (nilai >= 80) {  
    console.log("A");  
} else if (nilai >= 70) {  
    console.log("B");  
} else {  
    console.log("C");  
}
```

3. Looping

a. For Loop

```
for (let i = 1; i <= 5; i++) {  
    console.log("Perulangan ke-" + i);  
}
```

b. While Loop

```
let i = 1;  
while (i <= 3) {  
    console.log(i);  
    i++;  
}
```

c. cLooping Array

```
let prodi = ["Informatika", "SI", "DKV"];  
  
for (let item of prodi) {  
    console.log(item);  
}
```

9.4.Function & Scope

Function

Fungsi digunakan untuk mengelompokkan kode agar bisa dipanggil berulang kali.

Bentuk-bentuk function diantaranya

1. Function biasa

```
function sapa() {  
  console.log("Halo mahasiswa!");  
}
```

2. Function dengan parameter

```
function hitungLuas(p, l) {  
  return p * l;  
}
```

3. Function expression

```
const tambah = function(a, b) {  
  return a + b;  
};
```

4. Arrow Function (modern ES6)

```
const kali = (a, b) => a * b;
```

Scope

Scope menentukan **dimana variabel bisa diakses**.

1. Global scope

```
let nama = "Kampus ABC"; // bisa diakses dari mana saja
```

2. Function scope

```
function test() {  
  let x = 10; // hanya bisa diakses dalam function  
}
```

3. Block scope

```
if (true) {  
  let y = 5; // hanya di sini  
}
```

9.5.Studi Kasus: Validasi Form Pendaftaran Mahasiswa Baru

Kita akan membuat form PMB sederhana:

- o Mengecek apakah nama diisi
- o Mengecek apakah email valid
- o Mengecek apakah prodi dipilih

- o Menampilkan pesan error jika salah

File: pmb.html

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Pendaftaran Mahasiswa Baru</title>
  <link rel="stylesheet" href="style/style.css">
</head>
<body>

<h2>Form Pendaftaran Mahasiswa Baru</h2>
<form id="formPMB">
  <label for="nama">Nama Lengkap:</label>
  <input type="text" id="nama" />
  <label for="email">Email:</label>
  <input type="email" id="email" />
  <label for="prodi">Program Studi:</label>
  <select id="prodi">
    <option value="">-- Pilih Prodi --</option>
    <option value="informatika">Informatika</option>
    <option value="si">Sistem Informasi</option>
  </select>
  <button type="submit">Daftar</button>
</form>
<p id="pesan"></p>
<script src="script/validasi.js"></script>

</body>
</html>
```

File: script/validasi.js

```
document.getElementById("formPMB").addEventListener("submit", function(e) {
    e.preventDefault(); // mencegah form terkirim sebelum valid

    let nama =
document.getElementById("nama").value.trim();
    let email =
document.getElementById("email").value.trim();
    let prodi = document.getElementById("prodi").value;
    let pesan = document.getElementById("pesan");

    // Validasi nama
    if (nama === "") {
        pesan.textContent = "Nama wajib diisi!";
        pesan.style.color = "red";
        return;
    }

    // Validasi email
    if (!email.includes("@") || !email.includes(".")) {
        pesan.textContent = "Email tidak valid!";
        pesan.style.color = "red";
        return;
    }

    // Validasi prodi
    if (prodi === "") {
        pesan.textContent = "Pilih program studi!";
        pesan.style.color = "red";
        return;
    }

    // Jika semua valid
```

```
    pesan.textContent = "Pendaftaran berhasil! Selamat
datang, " + nama;
    pesan.style.color = "green";
});
```

Penjelasan Studi Kasus

- **addEventListener("submit", ...)** menangkap event submit form
- **preventDefault()** mencegah pengiriman form sebelum valid
- Validasi dilakukan menggunakan **if-else**
- Pesan kesalahan ditampilkan secara dinamis di halaman
- Website menjadi jauh lebih interaktif

KESIMPULAN BAB 9

Mahasiswa telah mempelajari:

1. Apa itu JavaScript & perannya dalam web modern
2. Variabel, tipe data, operator
3. Alur kontrol (if dan looping)
4. Fungsi dan ruang lingkup variabel (scope)
5. Penerapan JavaScript nyata: validasi form PMB

Dengan dasar ini, mahasiswa siap mempelajari manipulasi DOM & event handling di bab berikutnya.

BAB 10

DOM MANIPULATION & EVENT HANDLING

JavaScript dapat mengubah halaman web secara dinamis melalui **DOM (Document Object Model)**. Dengan DOM, JavaScript dapat:

1. Menambah elemen baru
2. Menghapus elemen
3. Mengubah teks atau style
4. Menangani interaksi pengguna (klik, ketik, hover, scroll)

Pada bab ini mahasiswa belajar dasar-dasar manipulasi DOM dan membuat **slider gambar kegiatan kampus**.
Apa Itu JavaScript & Perannya

10.1.DOM Tree

DOM adalah representasi struktur HTML dalam bentuk objek. Browser mengubah HTML menjadi:

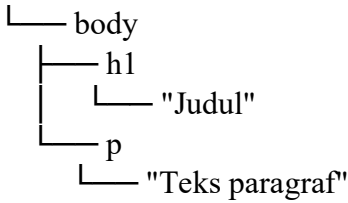
- **Nodes** → mewakili elemen, teks, atau atribut
- **Tree structure** → memiliki parent, child, sibling

Contoh HTML:

```
<body>
  <h1>Judul</h1>
  <p>Teks paragraf</p>
</body>
```

DOM Tree-nya:

```
document
├── html
```



JavaScript bisa mengakses elemen ini melalui **document**.

10.2. Query Selector

Untuk memilih elemen pada DOM, digunakan metode:

1. **document.getElementById**

```
let judul = document.getElementById("judul");
```

2. **document.getElementsByClassName**

```
let items = document.getElementsByClassName("item");
```

3. **document.querySelector**

Memilih elemen pertama yang cocok dengan selector CSS.

```
let p = document.querySelector("p");  
let nav = document.querySelector(".menu");  
let btn = document.querySelector("#btnLogin");
```

4. **document.querySelectorAll**

Memilih semua elemen yang cocok.

```
let semua = document.querySelectorAll("li");
```

10.3. Event Listener

Event adalah aksi pengguna seperti:

- klik
- scroll
- ketik (input)
- hover
- submit
- load

Untuk menangani event, gunakan:

```
elemen.addEventListener("event", function() {  
    // kode yang dijalankan saat event terjadi  
});
```

Contoh:

```
let tombol = document.querySelector("#btn");  
tombol.addEventListener("click", function() {  
    alert("Tombol diklik!");  
});
```

10.4. Manipulasi Elemen DOM

JavaScript dapat:

1. Mengubah Konten Elemen

innerHTML

```
document.querySelector("#judul").innerHTML =  
"Profil Kampus ABC";
```

textContent

```
document.querySelector("#status").textContent =  
"Login Berhasil";
```

2. Mengubah Style

```
let box = document.querySelector(".box");  
box.style.backgroundColor = "blue";  
box.style.padding = "20px";
```

3. Menambah Elemen

```
let list = document.querySelector("#daftar");  
let baru = document.createElement("li");  
baru.textContent = "Program Studi Baru";  
list.appendChild(baru);
```

4. Menghapus Elemen

```
let hapus = document.querySelector("#item1");  
hapus.remove();
```

5. Mengubah Atribut

```
let img = document.querySelector("#foto");
img.setAttribute("src", "gambar-baru.jpg");
```

10.5.Studi Kasus: Membuat Slider Gambar Kegiatan Kampus

Kita akan membuat slider sederhana yang dapat berubah gambar ketika tombol **Next** dan **Prev** ditekan.

File: slider.html

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Slider Kegiatan Kampus</title>
  <link rel="stylesheet" href="style/slider.css">
</head>
<body>
<h2>Galeri Kegiatan Kampus</h2>
<div class="slider">
  
</div>
<div class="tombol">
  <button id="prev">Prev</button>
  <button id="next">Next</button>
</div>
<script src="script/slider.js"></script>
</body>
</html>
```

File CSS: style/slider.css

```
body {
  font-family: Arial, sans-serif;
```

```

    text-align: center;
    background-color: #f5f5f5;
}
.slider img {
    width: 600px;
    height: auto;
    border-radius: 10px;
    box-shadow: 0 5px 10px rgba(0,0,0,0.2);
    transition: opacity 0.5s;
}
.tombol button {
    padding: 10px 20px;
    background-color: #0033cc;
    color: white;
    border: none;
    border-radius: 8px;
    margin: 10px;
    cursor: pointer;
    font-size: 16px;
}
.tombol button:hover {
    background-color: #0055ff;
}

```

File JavaScript: script/slider.js

```

// Array gambar slider
let gambar = [
    "assets/img/kegiatan1.jpg",
    "assets/img/kegiatan2.jpg",
    "assets/img/kegiatan3.jpg"
];
let index = 0; // indeks gambar aktif
let img = document.getElementById("gambarSlider");
// tombol next

```

```

document.getElementById("next").addEventListener("click", function() {
    index++;
    if (index >= gambar.length) index = 0;
    img.src = gambar[index];
});
// tombol prev
document.getElementById("prev").addEventListener("click", function() {
    index--;
    if (index < 0) index = gambar.length - 1;
    img.src = gambar[index];
});

```

Penjelasan Kode Studi Kasus

- **Array gambar** menyimpan daftar gambar kegiatan kampus
- **Event listener** menangani klik tombol Next/Prev
- **img.src** diganti sesuai index gambar
- Index akan kembali ke awal jika mencapai jumlah gambar
- **Slider** berjalan tanpa plugin tambahan

Hasil Akhir Slider

Mahasiswa akan mendapatkan slider sederhana dengan:

- Tampilan modern
 - Gambar yang dapat digeser
 - Interaksi menggunakan event listener
 - Pergantian gambar dinamis melalui DOM manipulation
- Slider ini dapat dikembangkan menjadi versi otomatis, animasi fade, dot indicator, dll.

KESIMPULAN BAB 10

Mahasiswa dapat memahami:

1. Struktur DOM tree
2. Cara memilih elemen dengan query selector
3. Menangani event dengan event listener
4. Melakukan manipulasi DOM: tambah, hapus, ubah konten
5. Membuat slider gambar kampus interaktif

Bab ini sangat penting karena menjadi dasar untuk semua efek interaktif di JavaScript.

BAB 11

JAVASCRIPT LANJUTAN

Pada **Bab** ini mahasiswa akan belajar:

- Struktur data penting: **array dan object**
- Penyimpanan sisi browser: localStorage
- Penulisan string modern: template literal
- Modularisasi fungsi untuk kode yang rapi
- Studi kasus: menampilkan pengumuman kampus dari file JSON lokal

Bab ini sangat penting karena aplikasi web modern selalu bekerja dengan data terstruktur dan modul-modul kecil yang reusable.

11.1. Array & Object

JavaScript adalah bahasa berorientasi objek yang fleksibel. Dua struktur data terpentingnya adalah:

1. Array

Array digunakan untuk menyimpan banyak nilai dalam satu variabel.

Contoh array:

```
let prodi = ["Informatika", "SI", "DKV"];
```

Mengakses elemen array:

```
console.log(prodi[0]); // Informatika
```

Menambah data:

```
prodi.push("Siber");
```

Menghapus data terakhir:

```
prodi.pop();
```

Looping array:

```
for (let item of prodi) {
```

```
        console.log(item);
    }
}
```

2. Object

Object menyimpan data dalam bentuk pasangan **key** : **value**.

Contoh object:

```
let mahasiswa = {
    nama: "Budi",
    umur: 20,
    prodi: "Informatika"
};
```

Mengakses properti:

```
console.log(mahasiswa.nama);
console.log(mahasiswa["prodi"]);
```

Menambah properti baru:

```
mahasiswa.alamat = "Cilegon";
```

Array of Objects

Digunakan untuk menyimpan banyak data kompleks.

```
let daftarMahasiswa = [
    { nama: "Budi", prodi: "Informatika" },
    { nama: "Siti", prodi: "SI" }
];
```

Ini akan sangat berguna untuk studi kasus JSON nantinya.

11.2.Local Storage

localStorage adalah penyimpanan data permanen di browser.

Berbeda dari cookie, localStorage:

- Menyimpan data hingga 5–10 MB
- Tidak kedaluwarsa
- Mudah digunakan
- Dapat menyimpan string, array, atau object

Menyimpan data

```
localStorage.setItem("nama", "Roy");
```

Mengambil data

```
let nama = localStorage.getItem("nama");
```

Menghapus data

```
localStorage.removeItem("nama");
```

Menyimpan object (harus JSON stringify)

```
let user = {nama: "Budi", prodi: "Informatika"};  
localStorage.setItem("user", JSON.stringify(user));
```

Mengambil kembali

```
let dataUser = JSON.parse(localStorage.getItem("user"));  
console.log(dataUser.nama);
```

11.3.Template Literal

Template literal menggunakan **backtick** (```) sebagai pengganti tanda kutip.

Contoh:

```
let nama = "Budi";  
let teks = `Halo, nama saya ${nama}`;
```

Keuntungan:

- Bisa menyisipkan variabel dengan `${}`
- Bisa membuat string multi-baris tanpa `\n`

Contoh multi-baris:

```
let bio = `  
Nama: Budi  
Prodi: Informatika  
`;
```

11.4.Modularisasi Sederhana

Modularisasi membantu memisahkan kode agar rapi, mudah dirawat, dan reusable.

1. Membuat file JavaScript terpisah

Misal:

```
script/  
├─ data.js  
└─ app.js
```

File: data.js

```
export const pengumuman = [  
  { judul: "Pendaftaran PMB Dibuka", tanggal: "2025-07-01" },  
  { judul: "Wisuda Periode April", tanggal: "2025-04-10" }  
];
```

File: app.js

```
import { pengumuman } from "./data.js";  
console.log(pengumuman);
```

2. Cara mengaktifkan module di HTML

```
<script type="module" src="script/app.js"></script>
```

11.5.Studi Kasus: Halaman Pengumuman Mengambil Data JSON Lokal

Pada studi kasus ini mahasiswa akan belajar:

- Mengambil data dari file JSON lokal
- Menampilkannya ke dalam layout website
- Menggunakan fetch()
- Menampilkan dengan template literal

1. Siapkan File JSON: data/pengumuman.json

```
[  
  {  
    "judul": "Pendaftaran Mahasiswa Baru 2025 Telah  
Dibuka",  
    "tanggal": "2025-06-01"  
  },  
  {  
    "judul": "Libur Nasional Idul Fitri",  
    "tanggal": "2025-04-10"  
  }  
]
```

```

    },
    {
      "judul": "Seminar Teknologi Kampus ABC",
      "tanggal": "2025-03-22"
    }
  ]
}

```

2. File HTML: pengumuman.html

```

<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Pengumuman Kampus ABC</title>
  <link rel="stylesheet" href="style/style.css">
</head>

<body>

<h2>Pengumuman Terbaru</h2>

<div id="listPengumuman"></div>

<script src="script/pengumuman.js"></script>

</body>
</html>

```

3. File JavaScript: script/pengumuman.js

```

// mengambil elemen untuk menampilkan pengumuman
let list = document.getElementById("listPengumuman");

// ambil data JSON lokal
fetch("data/pengumuman.json")
  .then(response => response.json())
  .then(data => {

```

```

let html = "";

data.forEach(item => {
  html += `
    <div class="card">
      <h3>${item.judul}</h3>
      <p>Tanggal: ${item.tanggal}</p>
    </div>
  `;
});

list.innerHTML = html;
})
.catch(error => {
  list.innerHTML = `<p>Gagal memuat
pengumuman.</p>`;
  console.error(error);
});

```

4. CSS Card Opsional (style.css)

```

.card {
  background: #ffffff;
  padding: 15px;
  margin: 15px 0;
  border-radius: 10px;
  box-shadow: 0 3px 8px rgba(0,0,0,0.1);
}

.card h3 {
  color: #0033cc;
  margin-bottom: 5px;
}

```

Penjelasan Studi Kasus

- **fetch()** digunakan membaca file JSON secara asynchronous
 - **response.json()** mengubah data menjadi array object
 - Data ditampilkan menggunakan **template literal**
 - Setiap pengumuman dibuat dalam bentuk **card**
 - Jika gagal, ditampilkan pesan error
- Ini adalah langkah awal menuju aplikasi web dinamis

KESIMPULAN BAB 11

Mahasiswa telah memahami:

1. Struktur data lanjutan: array, object, array of objects
2. Penyimpanan browser menggunakan localStorage
3. Template literal untuk formatting string modern
4. Modularisasi sederhana dengan type="module"
5. Pengambilan data JSON lokal menggunakan fetch()
6. Membangun halaman pengumuman dinamis berbasis data

Bab ini menjadi jembatan menuju pengembangan aplikasi web dinamis yang lebih kompleks.

BAB 12

WORKFLOW PENGEMBANGAN WEBSITE

Workflow pengembangan web adalah langkah-langkah sistematis dalam membangun sebuah website secara efisien, rapi, dan mudah dikelola. Pada bab ini mahasiswa mempelajari standar pengorganisasian proyek web, optimasi aset, pengelolaan file CSS/JS, debugging, hingga menyatukan seluruh halaman menjadi satu website terpadu.

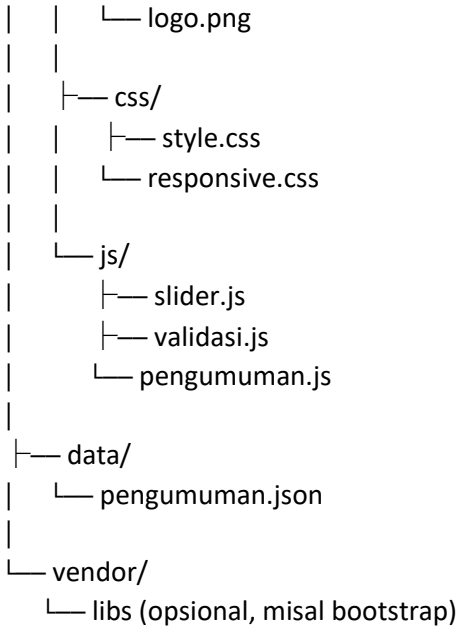
12.1. Struktur Folder Proyek Web

Website profesional wajib memiliki struktur folder rapi agar:

- o mudah dipahami oleh developer lain
- o memudahkan pemeliharaan
- o mendukung scaling (penambahan fitur)

Berikut struktur folder standar untuk website kampus:

```
/website-kampus/  
|— index.html  
|— profil.html  
|— fakultas.html  
|— pengumuman.html  
|— kontak.html  
|  
|— assets/  
|   |— img/  
|   |   |— kegiatan1.jpg  
|   |   |— kegiatan2.jpg
```



Penjelasan struktur:

Folder	Isi	Fungsi
assets/img	gambar	logo, banner, foto kegiatan
assets/css	file CSS	styling utama & file responsif
assets/js	file JS	slider, validasi form, pengambilan data
data	JSON	sumber data lokal
vendor	library	Bootstrap, jQuery, ikon, dll

Struktur seperti ini mengikuti standar produksi web modern.

12.2.Optimasi Gambar

Gambar merupakan salah satu penyebab website menjadi **lambat**. Oleh itu perlu dioptimalkan.

1. Format optimal:

- **JPG** → foto, banner
- **PNG** → logo transparan
- **WEBP** → ukuran kecil, kualitas tinggi
- **SVG** → ikon, logo vector

2. Kompresi gambar

Tools yang bisa digunakan:

- tinypng.com
- tinypng.com
- compressor.io

Mengurangi ukuran file tanpa mengurangi kualitas secara signifikan.

3. Gunakan ukuran yang sesuai

Jangan upload foto 4000px untuk ditampilkan pada ukuran 600px.

4. Gunakan Lazy Load

Menunda loading gambar hingga terlihat di viewport.

HTML:

```

```

12.3.Pengelolaan File CSS & JS

Pengelolaan file penting untuk memudahkan pemeliharaan.

1. Pisahkan CSS berdasarkan fungsi

Contoh:

`assets/css/`

`style.css` → style umum

`responsive.css` → media query

style.css (utama)

```
body { font-family: Arial; }  
header { background-color: #0033cc; }
```

responsive.css (khusus mobile)

```
@media (max-width: 768px) {  
  nav { display: none; }  
}
```

2. Pisahkan file JavaScript berdasar fitur

Misalnya:

- slider.js → untuk slider gambar
- validasi.js → untuk form pendaftaran
- pengumuman.js → untuk fetch JSON

Hal ini mencegah tumpang tindih fungsi dan memudahkan debugging.

3. Minify file CSS & JS (opsional tingkat lanjut)

File dapat diperkecil agar website lebih cepat.

Tools:

- minifier.org
- Tersedia juga plugins untuk VS Code

12.4. Debugging HTML/CSS/JS

Debugging adalah proses menemukan dan memperbaiki error. Browser menyediakan **DevTools** yang sangat membantu.

1. Debugging HTML

Gunakan **Inspect Element** untuk:

- melihat struktur DOM
- mengecek apakah elemen benar dimuat
- memeriksa atribut yang salah

2. Debugging CSS

DevTools memungkinkan:

- melihat style yang aktif
- memperbaiki style secara live
- mengecek konflik CSS

Contoh:

```
border: 1px solid red;
```

Teknik ini sering dipakai untuk test layout.

3. Debugging JavaScript

Buka Console di DevTools (F12 → Console).

Cek error:

```
console.log("Tes");  
console.error("Ini error");  
console.warn("Warning");
```

Breakpoint Debugging

Fitur Sources pada DevTools dapat menghentikan jalannya kode untuk melihat nilai variabel saat itu.

12.5. Studi Kasus: Menyatukan Semua Halaman Menjadi Website Kampus Terpadu

Pada studi kasus ini, kita membuat website kampus lengkap yang terdiri dari:

- Beranda
- Profil Kampus
- Fakultas & Program Studi
- Pengumuman (ambil data JSON)
- Kontak

Semua halaman menggunakan:

- header & footer seragam
- folder struktur yang rapi
- CSS terpadu

- o JavaScript modular

1. Header Template (header.html konsep)

```
<header class="header">
  
  <nav>
    <a href="index.html">Beranda</a>
    <a href="profil.html">Profil</a>
    <a href="fakultas.html">Fakultas</a>
    <a href="pengumuman.html">Pengumuman</a>
    <a href="kontak.html">Kontak</a>
  </nav>
</header>
```

2. Footer Template

```
<footer class="footer">
  <p>&copy; 2025 Kampus ABC. All Rights Reserved.</p>
</footer>
```

3. Integrasi ke semua halaman

Setiap halaman menyertakan file CSS & JS:

```
<link rel="stylesheet" href="assets/css/style.css">
<link rel="stylesheet" href="assets/css/responsive.css">
<script src="assets/js/slider.js"></script>
<script src="assets/js/validasi.js"></script>
```

4. Halaman Beranda

Menampilkan banner dan kartu fakultas (dari bab sebelumnya).

5. Halaman Profil

Menampilkan visi, misi, sejarah, struktur organisasi.

6. Halaman Fakultas

Menggunakan CSS Grid untuk kartu fakultas & program studi.

7. Halaman Pengumuman (dinamis menggunakan JSON)

```
fetch("data/pengumuman.json")
  .then(res => res.json())
```

```

.then(data => {
  let list = "";
  data.forEach(item => {
    list += `
      <div class="card">
        <h3>${item.judul}</h3>
        <small>${item.tanggal}</small>
      </div>
    `;
  });
});

```

```

document.getElementById("listPengumuman").innerHTML = list;
});

```

8. Halaman Kontak

Form kontak + validasi JavaScript.

9. Hasil Akhir Website Terpadu

- Folder rapi & profesional
- Halaman terhubung melalui navigasi
- Data tersentralisasi (JSON)
- Desain responsif
- Kode CSS & JS modular
- Semua fitur dari bab-bab sebelumnya dipadukan

Website kini memiliki standar pengembangan profesional, siap menjadi proyek akhir mahasiswa.

KESIMPULAN BAB 12

Mahasiswa kini telah memahami:

7. Struktur folder proyek web yang efektif
8. Teknik optimasi gambar untuk performa website
9. Pengelolaan file CSS dan JS agar modular
10. Teknik debugging HTML, CSS, dan JavaScript
11. Tahapan menyusun website kampus menjadi satu kesatuan utuh

Ini membekali mahasiswa memasuki bab berikutnya mengenai **deployment website**.

BAB 13

DEPLOY WEBSITE PROFIL KAMPUS

Setelah seluruh halaman website kampus selesai dibuat, langkah selanjutnya adalah deployment, yaitu proses mempublikasikan website ke internet. Pada bab ini mahasiswa belajar Git, GitHub, hosting gratis, dan langkah-langkah mengupload website ke GitHub Pages.

13.1. Pengenalan Git & GitHub

1. Apa itu Git?

Git adalah **version control system (VCS)** yang digunakan untuk:

- o melacak perubahan kode
- o bekerja dalam tim
- o mengembalikan versi lama jika ada kesalahan
- o menyimpan project secara terstruktur

Git digunakan oleh hampir semua perusahaan teknologi.

Perintah Git dasar:

```
git init    # membuat repository
git add .   # menambah semua file
git commit -m "pesan commit" # menyimpan perubahan
git status  # melihat status file
git push    # upload ke GitHub
```

2. Apa itu GitHub?

GitHub adalah layanan repositori Git berbasis cloud untuk:

- o menyimpan project

- berkolaborasi dengan developer lain
 - membuat dokumentasi
 - melakukan deployment (GitHub Pages, Actions, dll.)
GitHub menyediakan hosting gratis untuk website statis.
- 3. Manfaat GitHub dalam perkuliahan**
- Mahasiswa memiliki portofolio online
 - Memudahkan dosen memeriksa progres
 - Mahasiswa terbiasa bekerja dengan tools industri

13.2. Hosting Gratis (GitHub Pages / Netlify)

Website statis (HTML, CSS, JS) dapat dihosting gratis menggunakan:

1. GitHub Pages (paling direkomendasikan untuk pemula)

Fitur:

- Gratis selamanya
- Tidak perlu server
- Cocok untuk website profil kampus
- Mudah digunakan

Alamat website biasanya:

<https://username.github.io/nama-proyek/>

2. Netlify

Fitur:

- Build otomatis
- Deploy dengan drag & drop
- Custom domain tersedia
- Sangat populer di industri

Alamat website:

<https://project-name.netlify.app>

Dalam buku ajar ini kita fokus menggunakan **GitHub Pages**.

13.3. Struktur Build Sederhana

Website statis tidak memerlukan proses build yang kompleks. Struktur final yang akan di-deploy:

```
/website-kampus/  
|— index.html  
|— profil.html  
|— fakultas.html  
|— pengumuman.html  
|— kontak.html  
|  
|— assets/  
|   |— css/  
|   |— js/  
|   └─ img/  
|  
└─ data/  
    └─ pengumuman.json
```

Pastikan:

- Semua link menggunakan relative pat
- File CSS dan JS berada dalam folder asset
- Tidak ada file “.zip” atau backup lainn
- Website dapat dibuka secara lokal sebelum diupload

13.4.Studi Kasus: Upload Website Kampus ke GitHub Pages

Berikut langkah lengkap & jelas untuk deploy website profil kampus menggunakan GitHub Pages.

LANGKAH 1: Buat Akun GitHub

- Kunjungi: <https://github.com>
- Daftar menggunakan email
- Login ke dashboard GitHub

LANGKAH 2: Buat Repository Baru

- a. Klik New Repository
- b. Nama repository:
- c. website-kampus
- d. Pilih opsi:
 - Public
 - Centang "Add README.md" (*optional*)
- e. Klik **Create Repository**

LANGKAH 3: Upload File Website

Ada 2 cara: manual atau melalui Git CLI.

- a. **Cara A: Upload Manual (dianjurkan untuk pemula)**
 - Masuk ke repository
 - Klik **Add file** → **Upload files**
 - Drag semua file website ke GitHub
 - Klik **Commit changes**
 - Website sudah tersimpan di GitHub.
- b. **Cara B: Upload Menggunakan Git (untuk pengguna Git lokal)**

Di folder project:

```
git init
git add .
git commit -m "Upload website kampus"
git branch -M main
git remote add origin
https://github.com/username/website-kampus.git
git push -u origin main
```

LANGKAH 4: Aktifkan GitHub Pages

- a. Masuk ke repository → **Settings**
- b. Pilih menu **Pages**
- c. Pada “Source”, pilih:
- d. Branch: main

- e. Folder: /root
- f. Klik **Save**

GitHub akan memberikan alamat website:

<https://username.github.io/website-kampus/>

Tunggu 1–2 menit hingga aktif.

LANGKAH 5: Uji Website

- a. Buka URL tersebut dari:
 - HP
 - Laptop
 - Browser berbeda
- b. Periksa:
 - Link navigasi
 - Gambar tampil
 - CSS & JS berfungsi
 - Halaman pengumuman (JSON) tampil
- c. Jika ada gambar yang tidak muncul:
[Cek path gambar \(huruf besar/kecil sensitif di server GitHub\).](#)

LANGKAH 6: Update Website (jika ada perubahan)

Setiap kali memperbarui file:

Jika upload manual:

- Upload ulang file yang berubah
- Commit changes

Jika menggunakan Git:

```
git add .  
git commit -m "update fitur"  
git push
```

Website otomatis ter-update.

BONUS: Deploy ke Netlify (Ringkas)

- 1. Kunjungi <https://netlify.com>

2. Klik **Deploy site**
3. Drag folder project
4. Selesai!
URL akan diberikan otomatis misalnya:
<https://kampus-abc.netlify.app>

Hasil Studi Kasus

Dengan mengikuti bab ini, mahasiswa dapat:

- Membuat repository GitHub
- Mengupload website kampus ke internet
- Mengaktifkan GitHub Pages
- Melakukan update otomatis melalui Git
- Memiliki website portofolio yang dapat diakses publik
- Website kampus kini sudah **live**, profesional, dan siap dinilai.

KESIMPULAN BAB 13

Mahasiswa telah mempelajari:

1. Dasar Git & GitHub untuk version control
2. Hosting gratis menggunakan GitHub Pages atau Netlify
3. Struktur build website statis
4. Langkah teknis deploy website kampus
5. Cara update website secara profesional

Dengan ini, mahasiswa kini memiliki kemampuan lengkap untuk membuat dan mempublikasikan website.

BAB 14

STUDI KASUS LENGKAP: WEBSITE PROFIL KAMPUS

Pada bab ini mahasiswa akan membangun website profil kampus lengkap, terdiri dari:

- Homepage
- Profil Kampus
- Fakultas & Program Studi
- Galeri Kegiatan
- Berita / Pengumuman (JSON static)
- Form Kontak / PMB
- Footer & Navigasi Lengkap

14.1. Struktur Folder Final Website Kampus

Struktur folder standar produksi:

```
/website-kampus/  
|— index.html  
|— profil.html  
|— fakultas.html  
|— galeri.html  
|— pengumuman.html  
|— kontak.html  
|  
|— assets/  
|   |— css/  
|   |   |— style.css  
|   |   |— responsive.css  
|   |  
|   |
```

```

|   └─ js/
|   |   └─ slider.js
|   |   └─ pengumuman.js
|   |   └─ validasi.js
|   └─ img/
|       └─ kegiatan1.jpg
|       └─ kegiatan2.jpg
|       └─ kegiatan3.jpg
|       └─ logo.png
|       └─ fakultas/
|
|   └─ data/
|       └─ pengumuman.json
|
└─ vendor/ (opsional)

```

14.2.Template Final Website

Berikut adalah template minimal yang dapat dijadikan dasar struktur semua halaman.

14.2.1. Template Header (navigasi)

```

<header class="header">
  <div class="logo-container">
    
    <h1>Kampus ABC</h1>
  </div>

  <nav class="nav">
    <a href="index.html">Beranda</a>
    <a href="profil.html">Profil</a>
    <a href="fakultas.html">Fakultas</a>
    <a href="galeri.html">Galeri</a>
    <a href="pengumuman.html">Pengumuman</a>

```

```
        <a href="kontak.html">Kontak</a>
    </nav>
</header>
```

14.2.2. Template Footer

```
<footer class="footer">
    <p>&copy; 2025 Kampus ABC – All Rights
    Reserved</p>
</footer>
```

14.2.3. Template Layout Umum Halaman

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
    width, initial-scale=1.0">
    <title>Judul Halaman</title>
    <link rel="stylesheet" href="assets/css/style.css">
    <link rel="stylesheet"
    href="assets/css/responsive.css">
</head>

<body>

    <!-- Header -->
    <!-- (kode header di sini) -->

    <main class="container">
        <!-- konten halaman -->
    </main>

    <!-- Footer -->
    <!-- (kode footer di sini) -->

</body>
</html>
```

14.3. Halaman demi Halaman

1. Homepage (index.html)

Isi homepage:

- Banner hero
- Deskripsi singkat kampus
- 3 kartu fakultas unggulan
- Tombol ke halaman PMB / Profil

Contoh **Hero Section**:

```
<section class="hero">
  <h2>Selamat Datang di Kampus ABC</h2>
  <p>Kampus unggul berbasis teknologi dan inovasi.</p>
</section>
```

2. Halaman Profil Kampus (profil.html)

Berisi:

- Sejarah kampus
- Visi & Misi
- Struktur organisasi (opsional)
- Foto gedung kampus

Gunakan semantic HTML seperti:

```
<section>
  <h2>Visi Kampus</h2>
  <p>Menjadi perguruan tinggi unggul...</p>
</section>
```

3. Halaman Fakultas & Program Studi (fakultas.html)

Menggunakan **CSS Grid**:

```
<div class="grid-fakultas">
  <div class="card">
    <h3>Fakultas Teknik</h3>
    <ul>
      <li>Teknik Mesin</li>
      <li>Teknik Elektro</li>
    </ul>
  </div>
```

```

<div class="card">
  <h3>Fakultas Ilmu Komputer</h3>
  <ul>
    <li>Informatika</li>
    <li>Sistem Informasi</li>
  </ul>
</div>
</div>

```

4. Halaman Galeri Kegiatan (galeri.html)

Menggunakan Grid atau Flexbox:

```

<div class="galeri">
  
  
  
</div>

```

5. Halaman Pengumuman (pengumuman.html)

Mengambil data dari JSON lokal:

File: data/pengumuman.json

```

[
  {"judul": "Pendaftaran Mahasiswa Baru 2025",
  "tanggal": "2025-06-01"},
  {"judul": "Libur Idul Fitri", "tanggal": "2025-04-10"}
]

```

File: assets/js/pengumuman.js

```

fetch("data/pengumuman.json")
  .then(res => res.json())
  .then(data => {
    let html = "";
    data.forEach(item => {
      html += `
        <div class="card">
          <h3>${item.judul}</h3>
          <p>${item.tanggal}</p>
        </div>

```

```
};  
});
```

```
document.getElementById("listPengumuman").innerHT  
ML = html;  
});
```

6. Halaman Form Kontak / PMB (kontak.html)

Isi:

- Nama
- Email
- Pesan
- Validasi JavaScript sederhana

Contoh **form**:

```
<form id="formKontak">  
  <label>Nama:</label>  
  <input type="text" id="nama">  
  
  <label>Email:</label>  
  <input type="email" id="email">  
  
  <label>Pesan:</label>  
  <textarea id="pesan"></textarea>  
  
  <button type="submit">Kirim</button>  
</form>
```

14.4.Desain Akhir (CSS Template Final)

File: assets/css/style.css

Berisi:

- Warna primer kampus (#0033cc)
- Hero section
- Card desain
- Grid layout

- Footer & navigasi
- Spacing dan typography

File: assets/css/responsive.css

Berisi media query untuk:

- Tampilan mobile
- Navigasi hamburger
- Grid yang menyesuaikan ukuran layar

14.5. Checklist Kelulusan Praktikum Website Kampus

Checklist berikut digunakan sebagai syarat kelulusan praktikum:

a. Struktur Folder

- menggunakan assets/css, assets/js, assets/img
- halaman terpisah: index, profil, fakultas, galeri, pengumuman, kontak

b. HTML

- setiap halaman menggunakan semantic HTML
- navigasi konsisten di semua halaman
- footer konsisten
- tidak ada error pada HTML validator

c. CSS

- styling modern dengan warna konsisten
- penggunaan Flexbox pada layout navigasi
- penggunaan Grid pada halaman fakultas
- responsive mobile-friendly

d. JavaScript

- slider berjalan
- validasi form kontak
- pengumuman tampil dari JSON

e. Deployment

- folder lengkap diupload ke GitHub
- GitHub Pages aktif
- website dapat diakses publik

14.6. Rubrik Penilaian Proyek Akhir Website Kampus

Tabel Komponen Penilaian Proyek

Komponen	Bobot	Kriteria Penilaian
Struktur HTML	20%	Semantic, rapi, navigasi lengkap, tidak ada error
Desain CSS	25%	Konsisten, modern, responsive, UI/UX baik
JavaScript	25%	Slider berfungsi, form valid, fetch JSON bekerja
Konten & Informasi	15%	Kelengkapan halaman, teks rapi, relevansi data
Deployment GitHub Pages	15%	Website online, folder rapi, semua halaman berjalan
TOTAL	100%	–

14.7. Output Final Mahasiswa

Website profil kampus lengkap yang memenuhi:

- 6 halaman utama
- Responsif & modern
- Data dinamis JSON
- Validasi form
- Slider interaktif
- Navigasi terintegrasi
- Template profesional
- Live di GitHub Pages

Mahasiswa dapat menjadikan proyek ini sebagai **portofolio web development**.

KESIMPULAN BAB 14

Bab ini mengintegrasikan seluruh konsep dari Bab 1 sampai dengan Bab 13 menjadi proyek nyata. Mahasiswa tidak hanya mengetahui teori, tetapi mampu membangun website kampus profesional dari awal hingga deployment.

DAFTAR PUSTAKA

- Berners-Lee, T. H. (2001). *The Semantic Web. Scientific American.*
- Bootstrap Documentation.* (2023). Diambil kembali dari Components & Layout System.: <https://getbootstrap.com/>
- Canva Design School.* (2022). Diambil kembali dari Basic Principles of Layout & Typography.: <https://www.canva.com/learn/>
- Duckett, J. (2011). *HTML & CSS: Design and Build Websites.* John Wiley & Sons.
- Duckett, J. (2014). *JavaScript & jQuery: Interactive Front-End Web Development.* John Wiley & Sons.
- ECMA International.* (2023). Diambil kembali dari ECMA Script Language Specification.: <https://tc39.es/ecma262/>
- Felder, S. (2018). *Responsive Web Design with HTML5 and CSS3.* . Packt Publishing.
- Flanagan, D. (2020). *JavaScript: The Definitive Guide (7th Edition).* . O'Reilly Media.
- Freeman, E. &. (2005). *Head First HTML and CSS.* . O'Reilly Media.
- GitHub Docs.* (2023). Diambil kembali dari GitHub Pages Guides.: <https://docs.github.com/en/pages>
- Google Developers.* (t.thn.). Diambil kembali dari Web Fundamentals: Performance & SEO Basics.: <https://developers.google.com/web>
- Google Fonts.* (2023). Diambil kembali dari Typography Resources.: <https://fonts.google.com/>
- Hunt, A. (2019). UI/UX Principles for Modern Web Applications. *Journal of Web Engineering.*
- Keith, J. (2010). *HTML5 for Web Designers.* . A Book Apart.
- Marcotte, E. (2010). Responsive Web Design. *A List Apart.* Diambil kembali dari <https://alistapart.com/article/responsive-web-design/>

- Meyer, E. (2011). *CSS: The Definitive Guide*. O'Reilly Media.
- Mozilla Developer Network (MDN)*. (t.thn.). Diambil kembali dari HTML Documentation.: Mozilla Developer Network (MDN). HTML Documentation.
- Mozilla Developer Network (MDN)*. (t.thn.). Diambil kembali dari JavaScript Documentation.: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Mozilla Developer Network (MDN)*. . (t.thn.). Diambil kembali dari CSS Documentation.: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- Netlify Docs*. (2023). Diambil kembali dari Deploy a Static Site.
- React Documentation*. (t.thn.). Diambil kembali dari Thinking in Components. (optional for future learning): <https://react.dev/>
- Resig, J. &. (2013). *Secrets of the JavaScript Ninja*. . Manning Publications.
- Robbins, J. N. (2018). *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript*. . O'Reilly Media.
- Schmitt, C. (2012). Designing for the Web: Layout and Typography Principles. *Smashing Magazine*.
- W3C (World Wide Web Consortium)*. (2023). Diambil kembali dari HTML Specification.: <https://www.w3.org/Style/CSS/>
- W3Schools*. (2023). Diambil kembali dari HTML Tutorial.: <https://www.w3schools.com/html/>
- W3Schools*. (2023). Diambil kembali dari CSS Tutorial: <https://www.w3schools.com/css/>
- W3Schools*. (2023). Diambil kembali dari JavaScript Tutorial: <https://www.w3schools.com/js/>

TENTANG PENULIS



Roy Amrullah Ritonga, di Parmonangan. Ia menyelesaikan pendidikan Sarjana (S1) pada Program Studi Teknik Informatika di Universitas AMIKOM Yogyakarta pada tahun 2007, kemudian melanjutkan pendidikan Magister (S2) pada bidang Teknik Informatika di STMIK Eresha Jakarta dan lulus pada tahun 2010.

Karier akademiknya dimulai sebagai dosen di STMIK Serang pada tahun 2008. Selanjutnya, pada tahun 2010 ia dipercaya menjabat sebagai Ketua Program Studi Teknik Informatika di Universitas Serang Raya. Ia juga pernah menjabat sebagai Dekan Fakultas Ilmu Komputer Universitas Al-Khairiyah Cilegon hingga tahun 2025. Selain aktif dalam kegiatan akademik dan manajerial, ia mengampu mata kuliah Jaringan Komputer dan Pemrograman Berbasis Web. Di bidang penelitian, ia telah menghasilkan beberapa karya ilmiah, salah satunya mengenai penerapan metodologi Scrum dalam pengembangan aplikasi e-registrasi vendor. Pada tahun 2025, Roy Amrullah Ritonga menerbitkan buku ajar berjudul “Jaringan Komputer (Konsep Dasar dan Implementasi)” sebagai kontribusi dalam pengembangan literatur dan pembelajaran di bidang teknologi informasi.



Gustina seorang akademisi di bidang teknologi informasi yang memiliki latar belakang pengalaman profesional sebagai praktisi sebelum menekuni dunia pendidikan tinggi. Sebelum aktif sebagai dosen, ia bekerja di beberapa perusahaan yang berlokasi di Kota Cilegon, sehingga memiliki pemahaman yang kuat terhadap

penerapan teknologi informasi di dunia industri.

Pendidikan formal ditempuh dengan menyelesaikan program Sarjana (S1) pada Program Studi Sistem Informasi di salah satu perguruan tinggi swasta di Kota Cilegon. Selanjutnya, ia melanjutkan pendidikan Magister (S2) pada Program Studi Teknik Informatika guna memperdalam kompetensi akademik dan keilmuannya di bidang teknologi informasi. Saat ini, Gustina berprofesi sebagai dosen aktif di STTIKOM Insan Unggul. Dalam kegiatan tridarma perguruan tinggi, ia terlibat dalam pelaksanaan pendidikan dan pengajaran, serta terus mengembangkan keilmuan melalui kegiatan penelitian dan pengabdian kepada masyarakat. Dengan perpaduan pengalaman praktis dan akademik yang dimiliki, ia berkomitmen untuk berkontribusi dalam pengembangan sumber daya manusia yang kompeten di bidang teknologi informasi.



Anita Megayanti, menyelesaikan pendidikan S1 dalam bidang Sistem Informasi di STMIK Muhammadiyah Jakarta pada tahun 2004. Beliau kemudian melanjutkan studi dan meraih gelar S2 dengan jurusan yang sama di STMIK Eresha Jakarta pada tahun 2010. Saat ini berprofesi sebagai dosen di STTIKOM Insan Unggul Cilegon.

Ia mengampu beberapa mata kuliah, antara lain Teknologi Jaringan I, Organisasi dan Arsitektur Komputer, serta Perancangan dan Pemrograman Web. Selain kegiatan pengajaran, ia aktif dalam bidang penelitian dan publikasi ilmiah. Beberapa karya ilmiah yang telah dipublikasikan meliputi Sistem Informasi Eksekutif Aplikasi Rumah Sakit Berbasis Web dan Mobile System (Studi Kasus: RS. Tugurejo Semarang) yang diterbitkan dalam Jurnal Informatika, Penerapan Tools Manajemen Proyek pada PT.

Krakatau IT Cilegon yang juga dipublikasikan dalam Jurnal Informatika, Perancangan Enterprise Architecture E-Procurement Framework Menggunakan Metode TOGAF dalam Jurnal Sainteks, Sistem Pakar Identifikasi Penyakit Pernapasan Menggunakan Metode Certainty Factor dalam Jurnal IFTECH, serta Sistem Informasi Manajemen Aset Berbasis Web yang diterbitkan dalam Jurnal Saintek. Pada tahun 2025 bersama Roy Amrullah Ritonga menerbitkan buku ajar berjudul “Jaringan Komputer (Konsep Dasar dan Implementasi)” melalui Penerbit YPSIM, sebagai kontribusi dalam pengembangan bahan ajar dan literatur di bidang teknologi informasi.

Pemrograman Web I

Studi Kasus : Website profile kampus

Buku ajar Pemrograman Web ini disusun sebagai panduan pembelajaran bagi mahasiswa dalam memahami dan menerapkan konsep pemrograman web secara bertahap dan terstruktur. Materi disajikan mulai dari pengenalan konsep dasar web hingga pengembangan dan publikasi website profil kampus yang utuh dan fungsional. Pembahasan diawali dengan pengantar pemrograman web yang mencakup definisi dan sejarah web, cara kerja website berbasis arsitektur client–server, serta pengenalan teknologi front-end dan back-end beserta perangkat pendukung pengembangannya. Selanjutnya, mahasiswa diperkenalkan dengan HTML sebagai fondasi utama dalam membangun struktur halaman web, mulai dari elemen dasar, pengelolaan teks, media, hingga penerapan semantic HTML dan prinsip aksesibilitas dasar.

Pada tahap berikutnya, buku ini membahas Cascading Style Sheets (CSS) sebagai sarana pengaturan tampilan dan tata letak website. Materi meliputi konsep dasar CSS, box model, teknik layout konvensional, desain responsif menggunakan Flexbox, serta pengembangan layout modern dengan CSS Grid. Selain aspek teknis, buku ini juga memperkenalkan prinsip desain antarmuka dan pengalaman pengguna (UI/UX) dasar untuk menghasilkan tampilan website yang informatif, konsisten, dan menarik.

Interaktivitas website dibahas melalui pengenalan JavaScript, mulai dari konsep dasar pemrograman, manipulasi Document Object Model (DOM), penanganan event, hingga pengolahan data menggunakan array, object, dan penyimpanan lokal. Materi JavaScript dilengkapi dengan studi kasus yang relevan dengan kebutuhan website profil kampus, seperti validasi form, galeri interaktif, dan halaman pengumuman berbasis data statis.

Selain penguasaan teknis, buku ini juga membekali mahasiswa dengan pemahaman workflow pengembangan website yang mencakup pengelolaan struktur proyek, optimasi aset, teknik debugging, serta penggunaan sistem kontrol versi. Pada bagian akhir, mahasiswa diarahkan untuk melakukan deployment website menggunakan layanan hosting gratis sehingga hasil pengembangan dapat diakses secara publik.

Sebagai puncak pembelajaran, buku ini menyajikan studi kasus lengkap berupa pembuatan website profil kampus terpadu. Proyek akhir ini dirancang untuk mengintegrasikan seluruh materi yang telah dipelajari dan dilengkapi dengan template final, checklist praktikum, serta rubrik penilaian sebagai acuan evaluasi capaian pembelajaran mahasiswa.