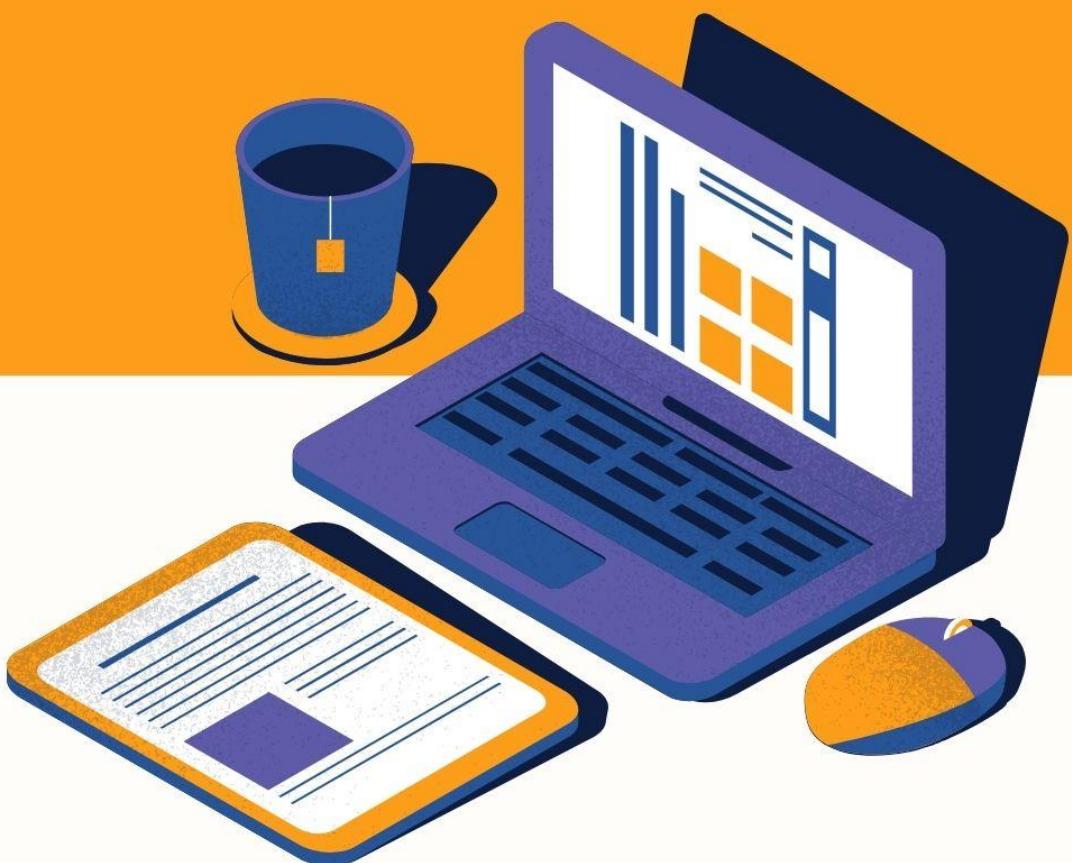


MODUL AJAR

PEMROGRAMAN

WEBSITE I



UIN K.H Abdurrahman Wahid
Fakultas Ekonomi dan Bisnis Islam
Program Studi Informatika

Dibuat Oleh:
Mohammad Reza Maulana, M.Kom



Pendahuluan

1. Deskripsi Mata Kuliah
 2. Capaian Pembelajaran
 3. Metode Pembelajaran
 4. Rencana Proyek: Website Profil UMKM
 5. Tools dan Software yang Digunakan
-

Pertemuan 1: Pengenalan Web & Proyek

1. Apa itu Website, Internet, dan Web Developer
 2. Perbedaan Frontend vs Backend
 3. Struktur Dasar Website dan Cara Kerja Browser
 4. Memulai Proyek: Struktur Folder dan File index.html
 5. Praktik: Membuat Halaman Awal UMKM
-

Pertemuan 2: HTML Dasar

1. Elemen dan Tag HTML
 2. Heading, Paragraph, dan Link
 3. Daftar Terurut dan Tak Terurut
 4. Gambar dan Atribut alt
 5. Praktik: Menambahkan Menu dan Profil UMKM
-

Pertemuan 3: HTML Media dan Tabel

1. Elemen Multimedia: , <audio>, <video>
 2. Struktur Tabel: <table>, <thead>, <tbody>
 3. Atribut colspan, rowspan, border
 4. Praktik: Menampilkan Jadwal atau Daftar Harga Layanan
-

Pertemuan 4: Formulir HTML

1. Elemen <form> dan Method

-
2. Input Text, Email, Select, Radio, Checkbox
 3. Textarea, Label, dan Button
 4. Praktik: Membuat Form Kontak Pengunjung
-

Pertemuan 5: Pengantar CSS

1. Apa itu CSS dan Cara Menyambungkan ke HTML
 2. Selektor CSS: tag, class, id
 3. Properti Dasar: warna, font, background
 4. Praktik: Memberi Style Halaman Utama
-

Pertemuan 6: Layout dan Box Model

1. Padding, Border, Margin, Width, Height
 2. Display: block, inline, inline-block
 3. Position: static, relative, absolute
 4. Praktik: Membuat Layout Grid Sederhana
-

Pertemuan 7: Flexbox dan Responsiveness

1. Apa itu Flexbox dan Container
 2. Properti justify-content, align-items, gap
 3. Media Query dan Responsive Design
 4. Praktik: Menyusun Layout Responsive Profil UMKM
-

Pertemuan 8: UTS (Proyek Parsial + Evaluasi Teori)

1. Penilaian Halaman HTML (struktur, semantik)
 2. Penilaian CSS (layout, style, responsive)
 3. Mini-Proyek: Halaman Beranda, About, dan Contact
-

Pertemuan 9: JavaScript Dasar

1. Apa itu JavaScript dan di mana menulisnya
2. Variabel: var, let, const
3. Tipe Data dan Operator

-
- 4. Praktik: Menambahkan Alert atau Console Log
-

Pertemuan 10: Kontrol Alur & Fungsi

- 1. Percabangan: if, else, switch
 - 2. Perulangan: for, while, forEach
 - 3. Fungsi dan Parameter
 - 4. Praktik: Validasi Input Sederhana
-

Pertemuan 11: Manipulasi DOM dan Event

- 1. DOM Tree dan Seleksi Elemen
 - 2. Event Listener: click, change, submit
 - 3. Mengubah Konten Halaman secara Dinamis
 - 4. Praktik: Validasi Form Interaktif
-

Pertemuan 12: Validasi Form Lanjutan

- 1. Mengambil Nilai dari Form
 - 2. Validasi: Email, Input Kosong, Panjang Karakter
 - 3. Menampilkan Pesan Kesalahan
 - 4. Praktik: Form Kontak Dinamis dan Aman
-

Pertemuan 13: ES6 dan Modular JS

- 1. Arrow Function dan Template Literal
 - 2. Destructuring dan Default Parameter
 - 3. Struktur Modular: File JS Terpisah
 - 4. Praktik: Refaktor Script.js untuk Efisiensi
-

Pertemuan 14: Library Pendukung (Bootstrap/jQuery)

- 1. Pengantar Framework CSS (Bootstrap)
- 2. Menggunakan Komponen Bootstrap: Card, Form, Navbar
- 3. Pengenalan jQuery dan Event jQuery
- 4. Praktik: Styling Cepat dan Interaktif dengan Library

Pertemuan 15: Integrasi & Finalisasi Proyek

1. Review Semua Komponen Website
 2. Finalisasi Tampilan, Validasi, dan Navigasi
 3. Dokumentasi dan Struktur Folder
 4. Praktik: Presentasi Awal Proyek Final
-

Pertemuan 16: UAS (Proyek Akhir dan Presentasi)

1. Penilaian Website Lengkap Profil UMKM
 2. Uji Navigasi, Responsiveness, Validasi
 3. Presentasi Individu
 4. Refleksi & Feedback
-

PENDAHULUAN

Program Studi: Informatika, Fakultas Ekonomi dan Bisnis, UIN K.H. Abdurrahman Wahid

Mata Kuliah: Pemrograman Web 1

SKS: 3 SKS Praktikum

Mata kuliah **Pemrograman Web 1** merupakan mata kuliah praktikum berskala 3 SKS yang dirancang untuk memberikan pemahaman dan keterampilan dasar pada pengembangan sisi *frontend* website. Mata kuliah ini merupakan fondasi bagi mahasiswa untuk memahami dan membangun website statis menggunakan teknologi dasar web seperti HTML, CSS, dan JavaScript. Melalui pendekatan berbasis proyek, mahasiswa akan belajar mengembangkan Website Profil UMKM yang mencakup halaman beranda, profil usaha, layanan/produk, galeri, hingga formulir kontak. Mata kuliah ini tidak hanya membekali keterampilan teknis, tetapi juga mendorong pengembangan logika pemrograman, kerapian struktur kode, serta pemahaman tentang user experience (UX) dan desain responsif.

1. Deskripsi Mata Kuliah

Mahasiswa akan dikenalkan pada:

- a. Arsitektur web dasar dan cara kerja HTTP request-response.
- b. Perbedaan peran pengembang *frontend* dan *backend*.
- c. Struktur dokumen HTML semantik untuk berbagai elemen (teks, media, tabel, formulir).
- d. Styling CSS: selektor, box model, layout (Flexbox & Grid), dan responsive design.
- e. Dasar JavaScript: syntax, variabel, kontrol alur, fungsi, manipulasi DOM, dan validasi form.
- f. Workflow proyek berbasis praktikum: organisasi folder, pemisahan file, dan penggunaan library/CDN sederhana.

2. Capaian Pembelajaran

Setelah menyelesaikan mata kuliah ini, mahasiswa diharapkan mampu:

- a. Menjelaskan konsep dasar internet, server, website dan cara kerja browser.
- b. Mengimplementasikan struktur HTML yang tersemantik dan mudah dipelihara.
- c. Menerapkan CSS untuk styling estetis dan layout responsif.

- d. Menggunakan JavaScript untuk membuat interaktivitas dasar dan validasi form.
- e. Mengatur alur kerja proyek web secara step-by-step dengan modular file.
- f. Memanfaatkan minimal satu library eksternal (misalnya Bootstrap atau jQuery) melalui CDN.

3. Metode Pembelajaran

- a. **Ceramah:** Dosen memberikan materi dasar kepada mahasiswa dengan penyampaian langsung.
- b. **Praktikum Step-by-Step:** Setiap pertemuan berfokus pada modul coding kecil—mahasiswa menambahkan fitur pada proyek sepotong demi sepotong, bukan hanya menyerahkan satu file utuh.
- c. **Demo Instruksi:** Dosen memandu praktik langsung, menulis kode secara live pada editor.
- d. **Tugas Mandiri:** Latihan lanjutan di rumah untuk menguatkan konsep pertemuan.
- e. **Diskusi & Review:** Sesi tanya jawab dan ulasan hasil kerja setiap minggu.
- f. **Evaluasi:** Quiz singkat, Presentasi, UTS pada Pertemuan 8 (HTML & CSS), dan UAS pada Pertemuan 16 (HTML, CSS & JavaScript).

4. Rencana Proyek: Website Profil UMKM

Mahasiswa akan membangun **Website Profil UMKM Sederhana** dengan fitur:

- a. **Struktur Dasar:** Membuat file dan folder terorganisasi (index.html, css/style.css, js/script.js).
- b. **Halaman Beranda:** Header, navigasi, hero section, dan profil singkat.
- c. **Halaman Tentang Kami:** Sejarah, visi-misi, dan tim.
- d. **Halaman Produk/Layanan:** Tabel layanan/produk dengan harga.
- e. **Halaman Galeri:** Foto galeri produk dll.
- f. **Halaman Kontak:** Formulir validasi nama, email, pesan.
- g. **Styling Responsif:** Flexbox, media queries, dan container grid sederhana.
- h. **Interaktivitas:** DOM manipulation untuk feedback pengguna.
- i. **Library/CDN:** Integrasi Bootstrap/jQuery secara bertahap.

5. Tools dan Software yang Digunakan

Kategori	Tool/Software	Keterangan
Code Editor	Visual Studio Code / Sublime	Edit kode HTML, CSS, JS
Browser	Chrome / Firefox	Debugging dan preview halaman web
Version Control	Git & GitHub (opsional)	Manajemen versi proyek
CSS Framework	Bootstrap (CDN)	Grid system dan komponen siap pakai
JS Library	jQuery (CDN, opsional)	Simplifikasi DOM dan event handling
Terminal/CLI	Git Bash / Terminal	Perintah git dan npm (jika digunakan)
Extension	Live Server, ESLint dan Prettier (opsional)	Extension VSCode untuk menjalankan server dan konsistensi kode JS & CSS

Catatan: Praktikum diutamakan menggunakan **metode step-by-step**: mahasiswa memodifikasi atau menambah kode pada file yang sama setiap pertemuan, sehingga terbiasa dengan pengembangan berkelanjutan dan versioning.

Jika ada kesalahan penulisan dan ketidaksesuaian harap menghubungi Dosen agar modul dapat diperbaiki.

Pertemuan Minggu Ke-1 - Pengenalan Web & Proyek

Tujuan Pembelajaran

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- Memahami konsep dasar dunia web,
- Menjelaskan posisi Frontend vs Backend,
- Menyiapkan proyek awal, dan
- Membuat halaman awal index.html untuk Website Profil UMKM.

1. Apa itu Website, Internet, dan Web Developer

a. Internet dan Web

- **Internet** adalah jaringan komputer global yang saling terhubung dan dapat saling bertukar data.
- **World Wide Web (Web)** adalah salah satu layanan dalam internet yang memungkinkan kita mengakses informasi berbasis **dokumen HTML** melalui **browser**.
- **Website** adalah sekumpulan halaman yang saling terhubung dan dapat diakses melalui alamat domain (misalnya: umkmku.com).

b. Siapa itu Web Developer?

Web developer adalah orang yang membangun dan memelihara website.

Terdapat 3 jenis:

- **Frontend Developer:** Bertanggung jawab terhadap tampilan yang dilihat pengguna.
- **Backend Developer:** Menangani logika di server, database, dan autentikasi.
- **Fullstack Developer:** Menguasai frontend dan backend.

2. Perbedaan Frontend vs Backend

Aspek	Frontend	Backend
Bahasa	HTML, CSS, JavaScript	PHP, Python, Node.js, dll.
Berjalan di	Browser	Server
Contoh Tugas	Layout, styling, interaksi	Proses data, autentikasi, API
Tools	VS Code, Chrome Dev Tools	PHPMyAdmin, Postman, Node.js

Dalam mata kuliah ini, kita akan fokus pada **Frontend Web Development**.

3. Struktur Dasar Website & Cara Kerja Browser

a. Struktur Dasar Website

Sebuah website sederhana biasanya terdiri dari:

- index.html: Halaman utama
- style.css: File styling
- script.js: Interaktivitas
- images/: Folder untuk gambar

b. Cara Kerja Browser

- 1) Pengguna mengetik alamat web (URL) di browser.
- 2) Browser mengirim permintaan ke server.
- 3) Server mengirimkan file HTML.
- 4) Browser membaca HTML lalu memuat CSS dan JS.
- 5) Hasil akhirnya ditampilkan ke pengguna.

4. Memulai Proyek: Struktur Folder & File index.html

a. Tools yang Digunakan:

- **VS Code**: Code editor utama.
- **Live Server Extension**: Untuk melihat hasil langsung di browser.

b. Struktur Folder Awal

```
/website-umkm/  
|  
|   index.html  
|   style.css  
|   script.js  
└   /images/
```

Tips: Simpan semua file dalam satu folder proyek agar mudah dikelola.

5. Praktik: Membuat Halaman Awal UMKM

a. Tujuan:

Membuat halaman index.html yang berisi:

- Judul UMKM
- Paragraf deskripsi

- Gambar logo/banner
- Menu navigasi sederhana

b. Kode: index.html

```

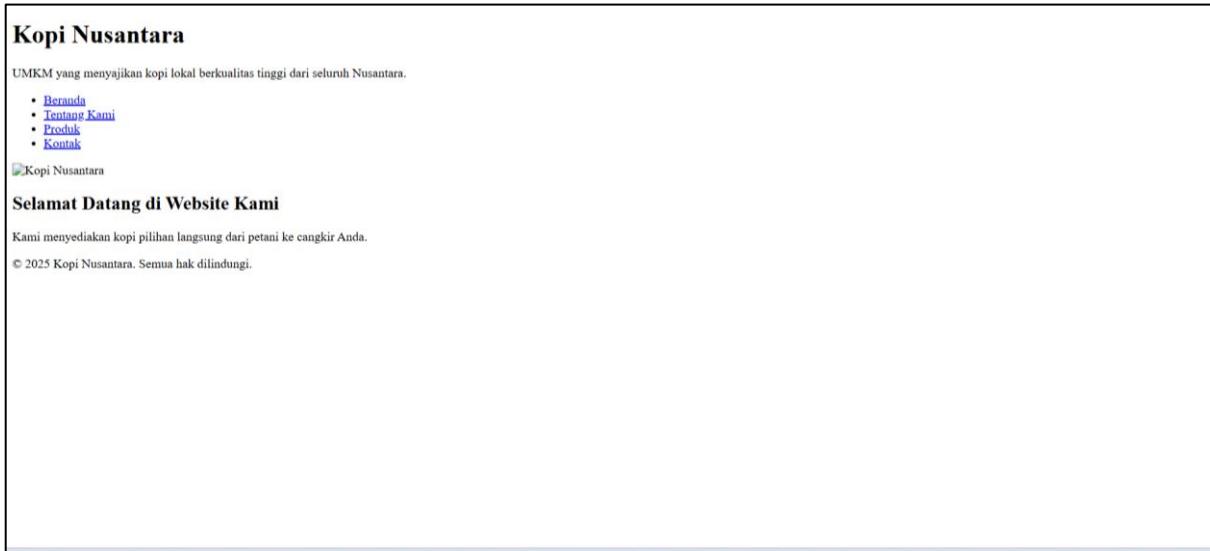
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <title>Profil UMKM - Kopi Nusantara</title>
    <link rel="stylesheet" href="style.css" />
</head>
<body>
    <header>
        <h1>Kopi Nusantara</h1>
        <p>UMKM yang menyajikan kopi lokal berkualitas tinggi dari seluruh Nusantara.</p>
        <nav>
            <ul>
                <li><a href="#">Beranda</a></li>
                <li><a href="#">Tentang Kami</a></li>
                <li><a href="#">Produk</a></li>
                <li><a href="#">Galeri</a></li>
                <li><a href="#">Kontak</a></li>
            </ul>
        </nav>
    </header>

    <main>
        
        <section>
            <h2>Selamat Datang di Website Kami</h2>
            <p>Kami menyediakan kopi pilihan langsung dari petani ke cangkir Anda.</p>
        </section>
    </main>

    <footer>
        <p>&copy; 2025 Kopi Nusantara. Semua hak dilindungi.</p>
    </footer>
</body>
</html>

```

c. Hasil Tampilan Website



The screenshot shows a simple website for 'Kopi Nusantara'. At the top, there's a navigation bar with links to 'Beranda', 'Tentang Kami', 'Produk', and 'Kontak'. Below the navigation is a logo image labeled 'Kopi Nusantara'. The main content area has a heading 'Selamat Datang di Website Kami' and a short paragraph about the company's mission to provide high-quality coffee from farmers directly to consumers. A copyright notice at the bottom states '© 2025 Kopi Nusantara. Semua hak dilindungi.'

d. Penjelasan Kode:

Tag	Fungsi
<head>	berisi metadata dan judul halaman
<header>	bagian atas, memuat judul dan menu
<nav>	menu navigasi
<main>	konten utama website
<section>	bagian konten terpisah.
<footer>	bagian bawah halaman.
<link rel="stylesheet" href="style.css" />	menghubungkan ke file CSS eksternal (akan digunakan pertemuan berikutnya).

6. Tugas Mahasiswa:

- 1) Cari produk UMKM yang dapat digunakan untuk proyek website umkm. Buat folder **umkm-namaumkm** di komputer Anda. Contoh penamaan projek misalnya umkm-pisang-keju, umkm-bagor, umkm-bajuku, dll.
- 2) Buat file index.html, style.css, script.js dan folder images/.
- 3) Tulis kode seperti index.html di atas. Sesuaikan isi, deskripsi, dll sesuai dengan UMKM yang anda angkat topiknya.
- 4) Jalankan Live Server untuk melihat hasilnya di browser.

Pertemuan Minggu Ke-2 - HTML Dasar

Tujuan Pembelajaran

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- Menjelaskan elemen dan tag dalam HTML
- Menggunakan heading, paragraf, dan link dengan benar
- Membuat daftar terstruktur (terurut dan tak terurut)
- Menampilkan gambar dan memberikan deskripsi alt
- Memperbarui halaman index.html dengan informasi profil UMKM masing-masing

1. Elemen dan Tag HTML

Pengertian:

- **HTML (HyperText Markup Language)** adalah bahasa markup untuk membuat struktur dasar halaman web.
- HTML terdiri dari **elemen** yang ditulis dengan **tag**, contoh:

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

Catatan: Semua elemen HTML dibuka dan ditutup, kecuali elemen tertentu seperti yang bersifat self-closing.

2. Heading, Paragraph, dan Link

a. Heading

Digunakan untuk judul atau subjudul.

```
<h1>Judul Besar</h1>
<h2>Subjudul</h2>
```

Gunakan heading secara hierarkis untuk SEO dan aksesibilitas.

b. Paragraf

Menuliskan teks dalam blok paragraf.

```
<p>Usaha kami berdiri sejak 2010 dengan visi memajukan kopi lokal.</p>
```

c. Link (Anchor)

Membuat tautan ke halaman lain atau website lain.

```
<a href="tentang.html">Tentang Kami</a>
<a href="https://instagram.com/kopinusantara" target="_blank">Instagram</a>
```

3. Daftar Terurut dan Tak Terurut

a. Unordered List ()

Untuk daftar yang tidak memiliki urutan.

```
<ul>
    <li>Ada kopi robusta</li>
    <li>Ada kopi arabika</li>
</ul>
```

b. Ordered List ()

Untuk daftar berurutan (angka).

```
<ol>
    <li>Petik buah kopi</li>
    <li>Sangrai</li>
    <li>Giling dan kemas</li>
</ol>
```

4. Gambar dan Atribut alt

Menambahkan gambar ke halaman:

```

```

alt digunakan untuk menjelaskan gambar, penting untuk SEO dan aksesibilitas pengguna disabilitas.

5. Praktik: Menambahkan Menu & Profil UMKM

a. Perbarui file index.html menjadi seperti berikut:

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <title>Profil UMKM - Kopi Nusantara</title>
    <link rel="stylesheet" href="style.css" />
</head>
<body>
    <header>
        <h1>Kopi Nusantara</h1>
        <nav>
            <ul>
                <li><a href="#">Beranda</a></li>
                <li><a href="#">Tentang Kami</a></li>
                <li><a href="#">Produk</a></li>
```

```
</li><a href="#">Galeri</a></li>
</li><a href="#">Kontak</a></li>
</ul>
</nav>
</header>

<main>
    
    <p>UMKM yang menyajikan kopi lokal berkualitas tinggi dari seluruh Nusantara.</p>
    <h2>Selamat Datang di Website Kami</h2>
    <p>Kami menyediakan kopi pilihan langsung dari petani ke cangkir Anda.</p>

    <section>
        <h2>Profil Usaha Kami</h2>
        <p>Kami berdiri sejak 2015, berkomitmen menghadirkan kopi terbaik langsung dari petani.</p>

        <h3>Produk Unggulan:</h3>
        <ul>
            <li>Kopi Arabika Gayo</li>
            <li>Kopi Robusta Temanggung</li>
            <li>Kopi Liberika Riau</li>
        </ul>

        <h3>Tahapan Produksi:</h3>
        <ol>
            <li>Pemetikan buah kopi segar</li>
            <li>Proses sangrai tradisional</li>
            <li>Pengemasan dan pengiriman</li>
        </ol>

        <h3>Kunjungi Kami:</h3>
        <p>
            <a href="https://instagram.com/kopinusantara" target="_blank">Instagram Kopi Nusantara</a>
        </p>
    </section>
</main>

<footer>
    <p>&copy; 2025 Kopi Nusantara. Semua hak dilindungi.</p>
</footer>
</body>
</html>
```

b. Hasil Tampilan Website

Kopi Nusantara

- Beranda
- Tentang Kami
- Produk
- Galeri
- Kontak



UMKM yang menyajikan kopi lokal berkualitas tinggi dari seluruh Nusantara.

Selamat Datang di Website Kami

Kami menyediakan kopi pilihan langsung dari petani ke cangkir Anda.

Profil Usaha Kami

Kami berdiri sejak 2015, berkomitmen menghadirkan kopi terbaik langsung dari petani.

c. Penjelasan Singkat

Tag	Fungsi
<h2>, <h3>	Subjudul dan sub-subjudul
 / 	Menyusun daftar
<a href>	Menyisipkan link ke luar atau ke dalam
	Menampilkan gambar dengan atribut alt. Agar gambar bisa muncul, silakan sediakan gambar dan rename menjadi banner.jpg, kemudian letakkan di folder images di dalam proyek website.

6. Tugas Mahasiswa

Tugas ini akan digunakan sebagai basis pengembangan di pertemuan berikutnya.

Petunjuk:

- 1) Gunakan nama dan profil UMKM **buatan sendiri** yang telah dikerjakan sebelumnya.
- 2) Modifikasi konten paragraf, daftar produk, gambar, dan link media sosial agar **sesuai dengan UMKM masing-masing**. Mahasiswa dapat memodifikasi konten dan layout sesuai kreatifitas masing-masing sesuai dengan tag yang telah diajarkan.
- 3) Simpan file dalam folder proyek website dengan struktur yang rapi.
- 4) Jalankan di Live Server dan dokumentasikan dengan screenshot tampilan awal.

Pertemuan Minggu Ke-3 – HTML Media dan Tabel

Tujuan Pembelajaran

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- Menggunakan elemen media dalam HTML (, <audio>, <video>)
- Membuat tabel HTML dengan struktur <table>, <thead>, <tbody>, dan atribut seperti colspan, rowspan
- Menyusun file produk.html sebagai halaman khusus daftar produk/layanan UMKM masing-masing

1. Elemen Multimedia: , <audio>, <video>

a. – Gambar

Digunakan untuk menampilkan gambar dari file lokal atau internet.

```

```

b. <audio> – Audio

Digunakan untuk menyisipkan suara atau musik.

```
<audio controls>
  <source src="audio/jingle.mp3" type="audio/mpeg">
    Browser Anda tidak mendukung audio.
</audio>
```

c. <video> – Video

Menampilkan video langsung di halaman.

```
<video width="320" height="240" controls>
  <source src="video/profil.mp4" type="video/mp4">
    Browser Anda tidak mendukung video.
</video>
```

Tips:

- Gunakan atribut controls agar pengguna dapat memutar/jeda media.
- Sertakan teks fallback untuk browser lama.

2. Struktur Tabel: <table>, <thead>, <tbody>

a. Contoh Tabel Produk

```
<table border="1" cellspacing="0" cellpadding="8">
  <thead>
    <tr>
```

```

<th>Produk</th>
<th>Ukuran</th>
<th>Harga</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kopi Arabika Gayo</td>
<td>250g</td>
<td>Rp 45.000</td>
</tr>
<tr>
<td>Kopi Robusta Temanggung</td>
<td>250g</td>
<td>Rp 38.000</td>
</tr>
</tbody>
</table>

```

b. Penjelasan:

Tag	Fungsi
<table>	Pembungkus utama tabel
<thead>	Bagian judul kolom
<tbody>	Data isi tabel
<th>	Table Header (judul kolom, tebal)
<td>	Table Data (isi sel biasa)
border / cellpadding / cellspacing	Pengaturan visual sederhana (nanti diganti CSS)

3. Atribut colspan, rowspan dan contoh style awal

a. Gabung Kolom (colspan)

```

<tr>
<td colspan="3" style="text-align: center; background-color: #eee;">
    Harga berlaku hingga 31 Desember
</td>
</tr>

```

b. Gabung Baris (rowspan)

```
<tr>
  <td rowspan="2">Kopi Luwak</td>
  <td>100g</td>
  <td>Rp 120.000</td>
</tr>
<tr>
  <td>250g</td>
  <td>Rp 250.000</td>
</tr>
```

4. Praktik: Membuat Halaman Produk (produk.html)

a. Struktur Folder

Sesuaikan struktur proyek dan file seperti ini, tambahkan beberapa file dan folder yang belum ada sebelumnya.

```
website-umkm/
├── index.html
├── tentang.html
└── produk.html
├── galeri.html
├── kontak.html
├── style.css
├── script.js
├── /audio/
└── /images/
    └── /videos/
```

b. Isi File: produk.html

Sebelum menulis kode, silakan siapkan asset berikut:

- Gambar produk kopi, rename file menjadi produk-1.jpg, pindahkan ke folder images
- Audio dalam format mp3, rename file menjadi cerita-umkm.mp3, pindahkan ke folder audio

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

```

<header>
    <h1>Kopi Nusantara</h1>
    <nav>
        <ul>
            <li><a href="index.html">Beranda</a></li>
            <li><a href="tentang.html">Tentang Kami</a></li>
            <li><a href="produk.html">Produk</a></li>
            <li><a href="galeri.html">Galeri</a></li>
            <li><a href="kontak.html">Kontak</a></li>
        </ul>
    </nav>
</header>

<main>
    <h2>Produk Unggulan Kami</h2>
    <p>Berikut daftar produk kopi Nusantara yang kami sediakan:</p>

    <table border="1" cellspacing="0" cellpadding="8">
        <thead>
            <tr>
                <th>Produk</th>
                <th>Ukuran</th>
                <th>Harga</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>Kopi Arabika Gayo</td>
                <td>250g</td>
                <td>Rp 45.000</td>
            </tr>
            <tr>
                <td>Kopi Robusta Temanggung</td>
                <td>250g</td>
                <td>Rp 38.000</td>
            </tr>
            <tr>
                <td>Kopi Liberika Riau</td>
                <td>250g</td>
                <td>Rp 42.000</td>
            </tr>
            <tr>
                <td colspan="3" style="text-align: center; background-color: #eee;">
                    Semua harga sudah termasuk PPN
                </td>
            </tr>
        </tbody>
    </table>

```

```

</table>

<h3>Profil Produk</h3>

<p>Kopi ini berasal dari dataran tinggi Gayo, memiliki cita rasa asam  
segar dan aroma khas.</p>

<h3>Dengarkan Cerita Kami</h3>
<audio controls>
  <source src="audio/cerita-umkm.mp3" type="audio/mpeg">
    Audio tidak tersedia di browser Anda.
</audio>
</main>

<footer>
  <p>&copy; 2025 Kopi Nusantara</p>
</footer>
</body>
</html>

```

Catatan : di bagian index.html, silakan ubah sendiri link di bagian nav seperti di file produk.html

c. Hasil Tampilan Website

Kopi Nusantara

- [Beranda](#)
- [Tentang Kami](#)
- [Produk](#)
- [Galeri](#)
- [Kontak](#)

Produk Unggulan Kami

Berikut daftar produk kopি Nusantara yang kami sediakan:

Produk	Ukuran	Harga
Kopi Arabika Gayo	250g	Rp 45.000
Kopi Robusta Temanggung	250g	Rp 38.000
Kopi Liberika Riau	250g	Rp 42.000

Semua harga sudah termasuk PPN

Profil Produk



Kopi ini berasal dari dataran tinggi Gayo, memiliki cita rasa asam segar dan aroma khas.

Dengarkan Cerita Kami

[Audio player showing 0.01 / 1.42]

© 2025 Kopi Nusantara

d. Penjelasan Singkat

Baris / Elemen	Penjelasan
<table>	Membuat tabel data produk.
<thead> / <tbody>	Memisahkan bagian header dan isi dari tabel.
<th>	Kolom header (teks dicetak tebal dan rata tengah).
<td>	Isi data dari tabel.
colspan="3"	Menggabungkan 3 kolom tabel dalam 1 sel.
	Menampilkan gambar produk dengan deskripsi alternatif (aksesibel).
<audio controls>	Menyisipkan elemen audio dengan kontrol putar.

5. Tugas Mahasiswa

Tujuan: Mahasiswa membuat halaman produk.html berdasarkan UMKM masing-masing.

Petunjuk:

- 1) Buat file produk.html dalam folder website-umkm/.
- 2) Tambahkan tabel berisi minimal 3 produk/layanan.
- 3) Tambahkan minimal satu gambar produk dengan deskripsi.
- 4) Tambahkan elemen media (audio/video) jika dimiliki, atau gunakan placeholder.
- 5) Sesuaikan semua isi dan nama produk dengan **profil UMKM masing-masing**.
- 6) Jalankan di Live Server dan dokumentasikan hasilnya.

Pertemuan Minggu Ke-4 – Formulir HTML

Tujuan Pembelajaran

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- Menjelaskan struktur form HTML dan fungsinya
- Menggunakan elemen input teks, radio, checkbox, dropdown, textarea, dan tombol
- Membuat halaman kontak.html yang memungkinkan pengunjung mengisi form komunikasi

1. Elemen <form>

Digunakan sebagai pembungkus semua elemen input/formulir. Memiliki atribut penting:

Atribut	Fungsi
action	URL tujuan saat form disubmit (sementara kosong atau "#" jika belum diproses)
method	Metode pengiriman data (get atau post)

```
<form action="#" method="post">
    <!-- elemen input -->
</form>
```

2. Elemen Input Dasar

Digunakan untuk menginput sebuah nilai (angka, huruf, dll)

Elemen	Kegunaan	Contoh
<input type="text">	Input teks umum	<input type="text" name="nama">
<input type="email">	Input email valid	<input type="email" name="email">
<textarea>	Input teks panjang (komentar/pesan)	<textarea name="pesan" rows="5" cols="40"></textarea>

```
<input type="text" name="nama" />
<input type="email" name="email" />
<textarea name="pesan" rows="5" cols="40"></textarea>
```

3. Select (Dropdown)

Memungkinkan pengguna memilih satu dari beberapa opsi.

```
<select name="kategori">
    <option value="produk">Produk</option>
    <option value="kerjasama">Kerja Sama</option>
</select>
```

4. Radio dan Checkbox

- Radio: memilih satu dari beberapa
- Checkbox: memilih lebih dari satu (opsional)

```
<input type="radio" name="gender" value="pria"> Pria
<input type="radio" name="gender" value="wanita"> Wanita

<input type="checkbox" name="langganan" value="ya"> Langganan newsletter
```

5. Label dan Tombol

<label> dikaitkan dengan input agar aksesibilitas lebih baik.

```
<label for="nama">Nama:</label>
<input type="text" id="nama" name="nama" />

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

6. Praktik: Membuat Halaman Kontak

a. Penyesuaian kode index.html dan tentang.html

Sebelum kita buat form, kita akan melakukan beberapa penyesuaian agar lebih rapi dalam penyusunan proyek web ini.

- Hapus isi kode dari tag <section>....</section> di index.html, hal ini karena akan kita pindahkan ke file tentang.html
- Tulis kode berikut di file tentang.html

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

```

<nav>
  <ul>
    <li><a href="index.html">Beranda</a></li>
    <li><a href="tentang.html">Tentang Kami</a></li>
    <li><a href="produk.html">Produk</a></li>
    <li><a href="galeri.html">Galeri</a></li>
    <li><a href="kontak.html">Kontak</a></li>
  </ul>
</nav>
</header>

<main>
  <h2>Profil Usaha Kami</h2>
  <p>Kami berdiri sejak 2015, berkomitmen menghadirkan kopi terbaik langsung dari petani.</p>

  <h3>Produk Unggulan:</h3>
  <ul>
    <li>Kopi Arabika Gayo</li>
    <li>Kopi Robusta Temanggung</li>
    <li>Kopi Liberika Riau</li>
  </ul>

  <h3>Tahapan Produksi:</h3>
  <ol>
    <li>Pemetikan buah kopi segar</li>
    <li>Proses sangrai tradisional</li>
    <li>Pengemasan dan pengiriman</li>
  </ol>

  <h3>Kunjungi Kami:</h3>
  <p><a href="https://instagram.com/kopinusantara" target="_blank">Instagram Kopi Nusantara</a></p>
</main>

<footer>
  <p>© 2025 Kopi Nusantara. Semua hak dilindungi.</p>
</footer>
</body>
</html>

```

b. Kode kontak.html

```

<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Kontak - Kopi Nusantara</title>

```

```
<link rel="stylesheet" href="style.css" />
</head>
<body>
    <header>
        <h1>Kopi Nusantara</h1>
        <nav>
            <ul>
                <li><a href="index.html">Beranda</a></li>
                <li><a href="tentang.html">Tentang Kami</a></li>
                <li><a href="produk.html">Produk</a></li>
                <li><a href="galeri.html">Galeri</a></li>
                <li><a href="kontak.html">Kontak</a></li>
            </ul>
        </nav>
    </header>

    <main>
        <h2>Hubungi Kami</h2>
        <p>Silakan isi formulir berikut untuk menghubungi tim Kopi Nusantara:</p>

        <form action="#" method="post">
            <label for="nama">Nama:</label><br>
            <input type="text" id="nama" name="nama" required /><br><br>

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

            <label for="kategori">Kategori Pesan:</label><br>
            <select id="kategori" name="kategori">
                <option value="umum">Umum</option>
                <option value="produk">Produk</option>
                <option value="kerjasama">Kerja Sama</option>
            </select><br><br>

            <label for="pesan">Pesan:</label><br>
            <textarea id="pesan" name="pesan" rows="5" cols="30"></textarea><br><br>

            <input type="checkbox" id="langganan" name="langganan" value="ya">
            <label for="langganan">Saya ingin menerima newsletter</label><br><br>

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

    <footer>
        <p>&copy; 2025 Kopi Nusantara</p>
    </footer>

```

```

</footer>
</body>
</html>

```

c. Hasil Tampilan Website

The screenshot shows a contact form for 'Kopi Nusantara'. At the top, there's a navigation menu with links to Beranda, Tentang Kami, Produk, Galeri, and Kontak. Below the menu, a section titled 'Hubungi Kami' asks users to fill out a form to connect with the team. The form fields include 'Nama' (Name), 'Email' (Email), 'Kategori Pesan' (Message Category) with a dropdown menu set to 'Umum' (General), and a large 'Pesan' (Message) text area. There's also a checkbox for newsletter subscription and a 'Kirim' (Send) button. The footer contains a copyright notice for 2025.

d. Penjelasan Kode – **Formulir HTML** di kontak.htm

Elemen/Form	Fungsi
<form action="#" method="post">	Form yang mengirim data secara POST (dummy action untuk saat ini)
<input type="text" /> email	Input teks untuk nama/email
<label for="...">	Label deskriptif terhubung ke input (aksesibilitas)
<select> + <option>	Dropdown untuk memilih kategori pesan
<textarea>	Area input pesan lebih panjang
<input type="checkbox">	Checkbox untuk langganan newsletter
<button type="submit">	Tombol untuk mengirim formulir

7. Tugas Mahasiswa

Mahasiswa membuat file kontak.html sesuai UMKM masing-masing

Petunjuk:

- 1) Gunakan elemen <form> dengan method post

- 2) Minimal 4 field: nama, email, kategori, dan pesan. Lebih banyak lebih bagus
- 3) Tambahkan checkbox dan tombol submit
- 4) Desain tetap sederhana, belum perlu styling CSS kompleks
- 5) Validasi dasar (required) sudah diterapkan
- 6) Dokumentasikan hasilnya dengan tangkapan layar

Pertemuan Minggu Ke-5 – Pengantar CSS

Tujuan Pembelajaran

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- Menjelaskan pengertian CSS dan fungsinya
- Menghubungkan CSS eksternal ke file HTML
- Menggunakan selektor dasar (tag, .class, #id)
- Menerapkan properti CSS seperti warna, font, dan background
- Memberi style awal pada halaman web

1. Apa itu CSS?

CSS (Cascading Style Sheets) adalah bahasa yang digunakan untuk mengatur **tampilan visual halaman HTML**, seperti warna, ukuran font, jarak, dan tata letak elemen.

Contoh dasar penggunaan CSS:

```
/* Selektor tag */
body {
    font-family: Arial, sans-serif;
    background-color: #f9f9f9;
    color: #333;
}
```

Penjelasan:

- body adalah **selektor** (elemen yang ditargetkan)
- font-family, background-color, color adalah **properti CSS**
- Nilai di sebelah kanan (setelah :) disebut **value**

2. Cara Menyambungkan CSS ke HTML

CSS dapat digunakan dengan 3 cara:

Cara	Penjelasan	Contoh
Inline	Ditulis langsung di elemen HTML	<p style="color: red;">Teks</p>
Internal	Di dalam tag <style> di <head>	<style>p { color: red; }</style>
Eksternal	File .css terpisah lalu dihubungkan via <link>	<link rel="stylesheet" href="style.css">

Rekomendasi terbaik: Gunakan **CSS eksternal** untuk menjaga struktur rapi dan efisien.

3. Selektor CSS

Selector CSS adalah bagian dari kode CSS yang berfungsi untuk memilih elemen HTML yang akan diberi style. Dengan kata lain, selector CSS adalah cara untuk menargetkan elemen-elemen tertentu dalam dokumen HTML agar bisa diubah tampilan atau gayanya menggunakan CSS.

Selektor	Contoh	Artinya
Tag	p {}	Semua elemen <p>
Kelas (.)	.judul {}	Semua elemen dengan class="judul"
ID (#)	#utama {}	Elemen dengan id="utama"

4. Contoh Properti Dasar CSS

Properti	Fungsi	Contoh
color	Warna teks	color: blue;
background-color	Warna latar	background-color: lightgray;
font-family	Jenis huruf	font-family: Arial;
font-size	Ukuran huruf	font-size: 16px;
text-align	Perataan teks	text-align: center;

5. Praktik: Membuat file style.css dan menerapkannya ke index.html

a. Kode CSS di file: **style.css**

```
/* Gaya umum halaman */
body {
    font-family: 'Segoe UI', sans-serif;
    background-color: #fefefe;
    color: #333;
    margin: 0;
    padding: 0;
}

/* Header */
header {
    background-color: #532a10;
    color: white;
    padding: 20px;
```

```

    text-align: center;
}

nav ul {
    list-style: none;
    padding: 0;
    display: flex;
    justify-content: center;
    gap: 20px;
}

nav ul li a {
    text-decoration: none;
    color: white;
    font-weight: bold;
}

/* Judul halaman */
h2 {
    color: #532a10;
}

/* Tabel produk */
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
}

th, td {
    border: 1px solid #ccc;
    padding: 12px;
    text-align: left;
}

th {
    background-color: #eee;
}

```

b. Jika belum ada, tambahkan ini ke semua file HTML (di <head>):

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

c. Hasil Tampilan Website

The screenshot shows the homepage of the Kopi Nusantara website. At the top, there is a dark brown header bar with the title "Kopi Nusantara" in white. Below the header is a navigation menu with links: Beranda, Tentang Kami, Produk, Galeri, and Kontak. The main content area features a large image of coffee beans in a glass jar. Below the image, there is a short paragraph about the UMKM's mission to serve high-quality coffee from various parts of Indonesia. A heading "Selamat Datang di Website Kami" is followed by a brief description of their service and copyright information.

The screenshot shows the product page of the Kopi Nusantara website. The header is identical to the homepage. The main content starts with a section titled "Produk Unggulan Kami" (Our Best Products). It includes a table showing four coffee products: Kopi Arabika Gayo, Kopi Robusta Temanggung, and Kopi Liberika Riau. Each product entry contains the name, size (250g), and price (Rp 45.000, Rp 38.000, and Rp 42.000 respectively). Below the table, a note states that all prices include VAT. The page also features a "Profil Produk" (Product Profile) section and a logo for "empocoffee" with a small image of coffee beans.

d. Penjelasan Kode – CSS Dasar

Baris / Selektor	Penjelasan
body { ... }	Mengatur font, warna latar, dan margin/padding default halaman
header { ... }	Memberi warna latar dan padding pada header
nav ul { ... }	Menghapus bullet dan mengatur menu horizontal dengan flexbox dasar
h2 { color: ... }	Warna khusus untuk subjudul halaman
table { ... }	Mengatur lebar dan tampilan tabel
th, td { ... }	Styling kolom dan isi tabel

Baris / Selektor	Penjelasan
th { background-color }	Warna latar khusus untuk header tabel

6. Tugas Mahasiswa

Mahasiswa membuat file style.css dan menerapkannya ke seluruh halaman

Petunjuk:

- 1) Buat file style.css dalam root folder
- 2) Tambahkan minimal: styling untuk body, header, nav, dan h2. Lebih banyak penerapan styling lebih bagus
- 3) Terapkan ke semua file HTML (index.html, tentang.html, dst)
- 4) Tambahkan warna latar dan font sesuai preferensi UMKM masing-masing
- 5) Dokumentasikan hasil sebelum dan sesudah styling

Pertemuan Minggu Ke-6 – Box Model & Layout Grid

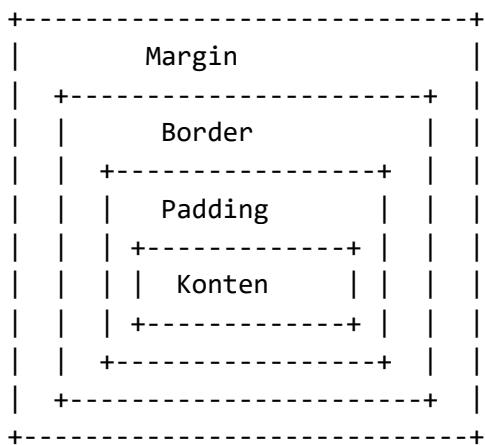
Tujuan Pembelajaran

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- Menjelaskan konsep Box Model dalam CSS
- Mengatur lebar, tinggi, margin, padding, dan border elemen
- Menggunakan display, position, dan float
- Membangun halaman galeri dengan gambar-gambar UMKM dalam tata letak grid yang bersih

1. CSS Box Model

Setiap elemen HTML dalam CSS dianggap sebagai **kotak** (box) yang terdiri dari:



```
.card {  
    width: 300px;  
    padding: 20px;  
    border: 2px solid #532a10;  
    margin: 15px;  
}
```

Properti	Penjelasan	Contoh
width, height	Ukuran konten dalam box	width: 300px;
padding	Ruang antara konten dan border	padding: 20px;

Properti	Penjelasan	Contoh
border	Garis luar elemen	border: 1px solid #ccc;
margin	Jarak antara elemen dengan elemen lain	margin: 15px;

2. Display Property

Nilai	Fungsi
block	Default elemen seperti <div>, <p>, <section>
inline	Elemen seperti , <a> – tidak memulai baris baru
inline-block	Gabungan: inline tapi bisa diberi padding/width
none	Menyembunyikan elemen

```
span {
  display: inline-block;
  padding: 10px;
}
```

3. Positioning

Nilai	Fungsi
static	Posisi default
relative	Bisa digeser dari posisi aslinya
absolute	Posisi terhadap elemen induk terdekat yang relatif
fixed	Menempel di posisi layar (contoh: sticky navbar)

```
.banner {
  position: absolute;
  top: 20px;
  left: 30px;
}
```

4. Float dan Clear (legacy)

- float: left/right; untuk menata elemen sejajar
- clear: both; mencegah elemen naik ke samping

Namun **modern layout** lebih disarankan pakai **Flexbox atau Grid**.

5. Grid Layout Sederhana

Grid merupakan layout dua dimensi dimana cocok untuk susunan baris dan kolom sekaligus. Contoh:

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

6. Praktik: Membuat Halaman galeri.html

a. Struktur Folder:

```
website-umkm/  
    └── galeri.html  
    └── images/  
        └── galeri1.jpg  
        └── galeri2.jpg  
        └── galeri3.jpg
```

Catatan: siapkan gambar sendiri dan masukkan ke folder yang sudah ditentukan

b. File: galeri.html

Silakan lengkapi kode file galeri.html dengan kode berikut:

```
<!DOCTYPE html>  
<html lang="id">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Galeri - Kopi Nusantara</title>  
    <link rel="stylesheet" href="style.css">  
</head>  
<body>  
    <header>  
        <h1>Kopi Nusantara</h1>  
        <nav>  
            <ul>  
                <li><a href="index.html">Beranda</a></li>  
                <li><a href="tentang.html">Tentang Kami</a></li>  
                <li><a href="produk.html">Produk</a></li>  
                <li><a href="galeri.html">Galeri</a></li>  
                <li><a href="kontak.html">Kontak</a></li>  
            </ul>  
        </nav>  
    </header>
```

```

</nav>
</header>

<main>
    <h2>Galeri Produk & Kegiatan</h2>
    <section class="galeri">
        <div class="card">
            
            <p>Proses sangrai tradisional</p>
        </div>
        <div class="card">
            
            <p>Kemasan kopi premium</p>
        </div>
        <div class="card">
            
            <p>Pameran UMKM di Bandung</p>
        </div>
    </section>
</main>

<footer>
    <p>© 2025 Kopi Nusantara. Hak cipta dilindungi.</p>
</footer>
</body>
</html>

```

c. Tambahkan di style.css:

Tuliskan kode CSS berikut. Terdapat beberapa tambahan lain untuk penyesuaian tampilan agar lebih rapi dan menarik.

```

/* kode CSS Sebelumnya */

/* Footer lebih menarik */
footer {
    background-color: #532a10;
    color: white;
    text-align: center;
    padding: 20px;
    margin-top: 40px;
}

/* Styling form */
form {
    max-width: 600px;
    margin: 0 auto;
    background-color: #f4f4f4;
    padding: 20px;
}

```

```
border-radius: 8px;
box-shadow: 0 0 10px rgba(0,0,0,0.1);
}

form label {
  display: block;
  margin-bottom: 6px;
  font-weight: bold;
  color: #333;
}

form input[type="text"],
form input[type="email"],
form textarea,
form select {
  width: 100%;
  padding: 10px;
  border: 1px solid #bbb;
  border-radius: 5px;
  margin-bottom: 15px;
  font-size: 1rem;
}

form input[type="checkbox"] {
  margin-right: 5px;
}

form button[type="submit"] {
  background-color: #532a10;
  color: white;
  border: none;
  padding: 10px 20px;
  font-size: 1rem;
  border-radius: 4px;
  cursor: pointer;
  transition: background 0.3s ease;
}

form button[type="submit"]:hover {
  background-color: #75401b;
}

/* Gambar dengan card */
img {
  max-width: 100%;
  height: auto;
  display: block;
  margin: 10px auto;
```

```
border-radius: 10px;
box-shadow: 0 4px 8px rgba(0,0,0,0.1);
}

/* Card container untuk produk atau galeri */
.card {
background-color: #fff;
border-radius: 10px;
box-shadow: 0 4px 12px rgba(0,0,0,0.08);
padding: 16px;
margin: 20px auto;
max-width: 600px;
text-align: center;
}

.galeri {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
gap: 20px;
padding: 20px;
max-width: 1000px;
margin: 0 auto;
}
```

d. Sesuaikan kode bagian gambar di produk.html untuk menerapkan class card

```
<div class="card">
    
    <h3>Kopi Arabika Gayo</h3>
    <p>
        Kopi khas Aceh dengan aroma segar dan rasa asam yang halus.
    </p>
</div>
```

e. Tampilan final setelah penerapan style CSS

Kopi Nusantara

Beranda Tentang Kami Produk Galeri Kontak



UMKM yang menyajikan kopi lokal berkualitas tinggi dari seluruh Nusantara.

Selamat Datang di Website Kami

Kami menyediakan kopi pilihan langsung dari petani ke cangkir Anda.

© 2025 Kopi Nusantara. Semua hak dilindungi.

Kopi Nusantara

Beranda Tentang Kami Produk Galeri Kontak

Produk Unggulan Kami

Berikut daftar produk kopi Nusantara yang kami sediakan:

Produk	Ukuran	Harga
Kopi Arabika Gayo	250g	Rp 45.000
Kopi Robusta Temanggung	250g	Rp 38.000
Kopi Liberika Riau	250g	Rp 42.000

Semua harga sudah termasuk PPN

Profil Produk



Kopi Nusantara

Beranda Tentang Kami Produk Galeri Kontak

Galeri Produk & Kegiatan



Proses sangrai tradisional



Kemasan kopi premium



Pameran UMKM di Bandung

© 2025 Kopi Nusantara. Hak cipta dilindungi.

Beranda Tentang Kami Produk Galeri Kontak

Hubungi Kami

Silakan isi formulir berikut untuk menghubungi tim Kopi Nusantara:

Nama:

Email:

Kategori Pesan:

Pesan:

Saya ingin menerima newsletter

Kirim

f. Penjelasan Kode Tambahan Style CSS

Selektor / Properti	Penjelasan
footer { ... }	Warna latar gelap, teks putih, dan padding untuk kenyamanan baca
form { ... }	Form di tengah, padding dan shadow agar terlihat "kotak form"
form label { ... }	Label dibuat block agar sejajar dengan input
input, textarea, select { ... }	Lebar penuh, padding, dan border halus untuk tampilan bersih
button[type="submit"] { ... }	Tombol warna khas brand, efek hover untuk interaksi
img { ... }	Gambar dengan radius dan shadow ringan, auto-scale
.card { ... }	Kotak serba guna (produk, galeri) dengan padding & bayangan ringan
.galeri { ... }	Container grid, gambar tersusun rapi
grid-template-columns	Menyusun 3 kolom responsif minimal 250px
gap	Jarak antar kolom dan baris grid
.card { ... }	Kotak gambar individual dengan padding & shadow
img { ... }	Otomatis skala agar tidak pecah

7. Tugas Mahasiswa

Mahasiswa membuat file style.css dan menerapkannya ke seluruh halaman

Petunjuk:

- 1) Buat file galeri.html dengan minimal 3 gambar
- 2) Tambahkan styling CSS agar tampilan website menjadi lebih menarik
- 3) Tambahkan deskripsi singkat di bawah setiap gambar
- 4) Gunakan struktur .card untuk setiap item
- 5) Gunakan gambar UMKM sendiri atau hasil ilustrasi bebas
- 6) Simpan gambar dalam folder images/

Pertemuan Minggu Ke-7 – Responsive Web dengan Flexbox & Media Queries

Tujuan Pembelajaran

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- Menjelaskan konsep Flexbox dan Media Queries dalam CSS
- Mengatur layout elemen secara fleksibel menggunakan Flexbox
- Menyesuaikan tampilan website agar cocok untuk berbagai ukuran layar (mobile, tablet, desktop)

1. Apa itu Flexbox?

Flexbox adalah sistem layout satu dimensi yang digunakan untuk mengatur elemen secara fleksibel dalam baris atau kolom.

CSS Flexbox memiliki beberapa fungsi utama, antara lain:

- **Mengatur arah elemen:** Flexbox memungkinkan penataan elemen dalam satu arah, baik secara horizontal (baris) maupun vertikal (kolom).
- **Mendukung desain responsif:** Elemen yang menggunakan Flexbox secara otomatis dapat menyesuaikan ukuran sesuai lebar layar, sehingga tampilan menjadi lebih adaptif.
- **Mempermudah penataan posisi elemen:** Dengan properti seperti justify-content, align-items, dan align-content, elemen dapat diposisikan ke tengah, kiri, kanan, atas, bawah, atau tersebar merata dalam kontainer.
- **Membagi ruang secara fleksibel:** Properti flex-grow dan flex-shrink memungkinkan elemen berbagi ruang secara proporsional dalam satu baris atau kolom.
- **Mengubah urutan tampilan elemen:** Menggunakan properti order, kita dapat menyusun ulang posisi elemen tanpa mengubah urutan dalam HTML.
- **Menangani elemen dengan ukuran berbeda:** Flexbox mampu mengatur tata letak elemen dengan berbagai ukuran di dalam satu kontainer secara efisien.

Properti Flex Container

Properti	Penjelasan	Contoh
display: flex;	Mengaktifkan Flexbox	.container { display: flex; }
flex-direction	Arah sumbu utama (row, column, row-reverse, column-reverse)	flex-direction: row;
justify-content	Posisi horizontal (main axis)	center, space-between, space-around
align-items	Posisi vertikal (cross axis)	center, flex-start, stretch
flex-wrap	Memungkinkan baris baru jika elemen tidak muat	flex-wrap: wrap;

Contoh Flex Container

```
.container {
  display: flex;
  justify-content: center;
  gap: 20px;
}
```

Properti Flex Item

Properti	Penjelasan
flex	Kombinasi grow, shrink, basis
order	Menentukan urutan elemen
align-self	Override align-items pada item tertentu

Contoh Flex Item

```
.card {
  flex: 1 1 250px; /* grow, shrink, basis */
}
```

2. Apa itu Media Queries?

Media Queries digunakan untuk menerapkan CSS berdasarkan ukuran layar (device width).

```
@media (max-width: 768px) {
  .galeri {
    flex-direction: column;
```

```
}
```

Contoh Media Query	Fungsi
max-width: 768px	Untuk tablet ke bawah
max-width: 480px	Untuk mobile
min-width: 1024px	Untuk desktop besar

3. Praktik: Membuat Galeri Responsif

a. Tujuan Praktik

- Menjadikan .galeri responsif dengan Flexbox
- Menyusun .card agar wrap ke bawah di layar kecil

b. Tambahkan CSS Flexbox & Media Query di style.css

```
.galeri {  
    display: flex;  
    flex-wrap: wrap;  
    gap: 20px;  
    justify-content: center;  
    padding: 20px;  
}  
  
.card {  
    flex: 1 1 250px;  
    max-width: 300px;  
    background-color: #fff;  
    border: 1px solid #ccc;  
    border-radius: 10px;  
    box-shadow: 2px 2px 8px rgba(0,0,0,0.1);  
    overflow: hidden;  
    text-align: center;  
    padding: 10px;  
}  
  
.card img {  
    width: 100%;  
    height: auto;  
    border-bottom: 1px solid #ddd;  
}  
  
/* Responsif untuk layar kecil */  
@media (max-width: 768px) {  
    .galeri {  
        flex-direction: column;  
        align-items: center;
```

```
}

.card {
    max-width: 90%;
}

}
```

c. Tampilan halaman web





d. Penjelasan Kode – *Flexbox & Responsiveness*

Kode / Properti	Fungsi
display: flex	Mengubah .galeri jadi Flex container
flex-wrap: wrap	Elemen otomatis pindah baris
justify-content: center	Posisi tengah secara horizontal
flex: 1 1 250px	Elemen tumbuh, menyusut, dan lebar awal 250px
@media (max-width: 768px)	Aturan khusus saat layar \leq 768px
flex-direction: column	Layout jadi vertikal untuk mobile
max-width: 90%	Kartu hampir penuh di layar kecil

4. Perbandingan Flex VS Grid

Baik Flex maupun Grid memiliki kemampuan untuk membuat layout menjadi responsif secara otomatis. Namun, keduanya terdapat perbedaan yang perlu kita ketahui untuk memaksimalkan penggunaannya.

Aspek	Flexbox	Grid
Dimensi	Satu dimensi (row atau column)	Dua dimensi (row <i>dan</i> column)
Kapan dipakai	Layout baris, nav, form	Galeri, dashboard, grid produk

Kemudahan responsif	Sangat baik, fleksibel	Baik, tapi lebih kaku untuk beberapa kasus
Deklarasi	Berdasarkan hubungan antar item	Berdasarkan <i>grid-area</i> yang terdefinisi
Sintaks	<code>display: flex; , justify-content, align-items</code>	<code>display: grid; , grid-template-columns, grid-template-rows</code>

5. Tugas Mahasiswa

Mahasiswa membuat file style.css dan menerapkannya ke seluruh halaman

Petunjuk:

- 1) Buka file galeri.html dan style.css
- 2) Terapkan Flexbox untuk menyusun .card secara fleksibel untuk beberapa gambar
- 3) Tambahkan Media Query agar tampilan berubah di layar kecil
- 4) Uji coba pada mode developer untuk memastikan responsif
- 5) Gunakan minimal 6 gambar

Contoh Modifikasi Tampilan – Latihan Lanjutan CSS

Penyesuaian kerapian text, nav, penambahan logo, hero, card keunggulan dan card testimoni di halaman index. Untuk halaman lain bisa berkreasi sendiri.

Penyesuaian dan tambahan di file style.css (nav bagian sebelumnya ada yang dihapus dan ditambah)

```
/** CSS sebelumnya */
/* Bagian nav ada penyesuaian di bawah */

.container {
    max-width: 1000px;
    margin: 0 auto;
    padding: 0 20px;
}

nav {
    background-color: #532a10;
    position: sticky;
    top: 0;
    z-index: 1000;
    border-bottom: 1px solid #eee;
}

nav ul {
    list-style: none;
    padding: 0;
    display: flex;
    justify-content: center;
    gap: 20px;
}

nav ul li a {
    text-decoration: none;
    color: white;
    font-weight: bold;
}

.nav-container {
    max-width: 1000px;
    margin: 0 auto;
    padding: 10px 20px;
    display: flex;
    align-items: center;
    /* flex-wrap: wrap; */
}
```

```
.logo img {
  height: 100px;
  width: auto;
}

.nav-menu {
  display: flex;
  gap: 30px;
  justify-content: center;
  flex-grow: 0.75;
  list-style: none;
  padding: 0;
  margin: 0;
}

.nav-menu li {
  margin-right: 20px;
}

.nav-menu li a {
  text-decoration: none;
  color: white;
  padding: 8px 4px;
  border-bottom: 2px solid transparent;
  transition: 0.3s ease;
}

.nav-menu li a:hover,
.nav-menu li a.active {
  color: #de9e4c;
  border-bottom: 2px solid #de9e4c;
}

.hero {
  background: #e9e1cf;
  padding: 60px 0;
  text-align: center;
}

.hero h1 {
  font-size: 2.5em;
}

.keunggulan, .testimoni {
  padding: 40px 0;
}

.fitur-list {
```

```

list-style: none;
padding-left: 0;
}

.fitur-list li {
margin-bottom: 10px;
background: #e8ebec;
padding: 10px;
border-radius: 6px;
}

.testimoni .card {
background-color: #fff;
padding: 20px;
border-left: 5px solid #de9e4c;
box-shadow: 0 2px 6px rgba(0,0,0,0.1);
border-radius: 6px;
margin-top: 20px;
}

```

Penyesuaian dan tambahan di file index.html (full code) – untuk logo dipersiapkan sendiri

```

<!DOCTYPE html>
<html lang="id">

<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Profil UMKM - Kopi Nusantara</title>
<link rel="stylesheet" href="style.css" />
</head>

<body>

<nav>
<div class="nav-container">
<div class="logo">

</div>
<ul class="nav-menu">
<li><a href="index.html" class="active">Beranda</a></li>
<li><a href="tentang.html">Tentang Kami</a></li>
<li><a href="produk.html">Produk</a></li>
<li><a href="galeri.html">Galeri</a></li>
<li><a href="kontak.html">Kontak</a></li>
</ul>
</div>
</nav>

```

```
</nav>

<header class="hero">
  <div class="container">
    <h2>Selamat Datang di Website Kami</h2>
    <p>Kami menyediakan kopi pilihan langsung dari petani ke cangkir  
Anda.</p>
  </div>
</header>

<section class="keunggulan">
  <div class="container">
    <h2>Kenapa Memilih Kami?</h2>
    
    <p>UMKM yang menyajikan kopi lokal berkualitas tinggi dari seluruh  
Nusantara.</p>
    <ul class="fitur-list">
      <li>✓ Produk 100% Homemade</li>
      <li>✓ Bahan Alami & Segar</li>
      <li>✓ Pengiriman Cepat</li>
    </ul>
  </div>
</section>

<section class="testimoni">
  <div class="container">
    <h2>Testimoni Pelanggan</h2>
    <div class="card">
      <p>"Rasanya enak dan pengiriman cepat!"</p>
      <small>- Sari, Surabaya</small>
    </div>
  </div>
</section>

<footer>
  <p>&copy; 2025 Kopi Nusantara. Semua hak dilindungi.</p>
</footer>
</body>

</html>
```

Contoh Tampilan Akhir



Beranda Tentang Kami Produk Galeri Kontak

Selamat Datang di Website Kami

Kami menyediakan kopi pilihan langsung dari petani ke cangkir Anda.

Kenapa Memilih Kami?





Beranda **Tentang Kami** Produk Galeri Kontak

UMKM yang menyajikan kopi lokal berkualitas tinggi dari seluruh Nusantara.

Produk 100% Homemade
 Bahan Alami & Segar
 Pengiriman Cepat

Testimoni Pelanggan

"Rasanya enak dan pengiriman cepat!"
— Sari, Surabaya

Pertemuan 8 – UTS

Pertemuan Minggu Ke-9 – JavaScript Dasar

Tujuan Pembelajaran

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- Memahami apa itu JavaScript dan cara memasukkannya ke dalam halaman HTML
- Menguasai deklarasi variabel dengan var, let, dan const
- Mengetahui tipe data dasar dan operator di JavaScript
- Menerapkan kode sederhana: alert() dan console.log()

1. Apa itu JavaScript dan di mana menulisnya

JavaScript adalah bahasa pemrograman yang dijalankan di browser (client-side) untuk membuat halaman web menjadi **dinamis** dan **interaktif**. Selain di browser, JS juga dapat dijalankan di server (Node.js), tapi kita fokus ke penggunaan di HTML.

Cara menulis JavaScript dalam HTML

- a. **Inline** — langsung dalam elemen HTML

```
<button onclick="alert('Halo!')">Klik Saya</button>
```

- b. **Internal** — dalam tag <script> di <head> atau sebelum </body>

```
<script>
  console.log('Halaman dimuat');
</script>
```

- c. **Eksternal** — terpisah di file .js, lalu di-link dengan <script src="script.js"></script>

```
<!-- di index.html, sebelum </body> -->
<script src="script.js"></script>
```

Rekomendasi: gunakan **file eksternal** (script.js) agar kode terstruktur rapi dan mudah dipelihara.

2. Deklarasi Variabel: var, let, const

Keyword	Scope	Reassignable?	Hoisting	Catatan
var	function/global	Ya	Di-“hoist” ke atas	Lama, berperilaku membingungkan di blok
let	block	Ya	Hoisted (TDZ)	Direkomendasikan untuk variabel yang berubah
const	block	Tidak	Hoisted (TDZ)	Untuk nilai konstan, wajib inisialisasi

Contoh:

```
<script>
  var nama = 'Andi';
  let usia = 25;
  const UMKM = 'Kopi Nusantara';

  // var bisa di-reassign
  nama = 'Budi';
  // let juga
  usia = 26;
  // const akan error jika diubah:
  // UMKM = 'Kopi Hebat'; // Uncaught TypeError
</script>
```

3. Tipe Data dan Operator

Dalam pemrograman, tipe data dan operator adalah dua komponen fundamental yang membentuk dasar dari logika dan pengolahan informasi dalam sebuah program. Di JavaScript—sebagai bahasa pemrograman yang bersifat dinamis dan fleksibel—pemahaman terhadap tipe data dan operator sangat penting untuk menulis kode yang benar, efisien, dan mudah dipahami.

a. Tipe Data

Tipe data (data types) adalah klasifikasi dari nilai-nilai yang digunakan dalam program. JavaScript secara umum memiliki dua jenis tipe data utama:

- 1) Tipe Data Primitif (Primitive Types) : Tipe data ini menyimpan nilai secara langsung dan tidak memiliki method atau properti. Tipe data primitif di JavaScript antara lain:

2) Tipe Data Non-Primitif (Reference Types) : Tipe ini menyimpan referensi ke nilai, bukan nilai itu sendiri.

Berikut contoh tipe data

Jenis Tipe Data	Nama Tipe	Contoh Nilai	Keterangan
Primitif	String	"Hello", 'JS'	Teks atau karakter diapit oleh tanda kutip
Primitif	Number	10, 3.14, -5	Semua jenis angka (bulat dan desimal)
Primitif	Boolean	true, false	Nilai logika benar atau salah
Primitif	Undefined	undefined	Variabel yang belum diberi nilai
Primitif	Null	null	Nilai kosong (diset secara eksplisit)
Primitif	BigInt	12345678901234567890n	Untuk angka yang sangat besar
Primitif	Symbol	Symbol('id')	Identifier unik untuk properti objek
Non-Primitif	Object	{nama: "Ali", usia: 20}	Koleksi pasangan key-value
Non-Primitif	Array	[1, 2, 3]	Kumpulan nilai yang disusun berdasarkan indeks
Non-Primitif	Function	function() {}	Blok kode yang bisa dipanggil kembali

b. Operator

Operator digunakan untuk melakukan operasi terhadap nilai atau variabel.

JavaScript menyediakan berbagai jenis operator berdasarkan fungsinya.

1) Operator Aritmatika

Operator	Keterangan	Contoh
+	Penjumlahan	2 + 3 // 5
-	Pengurangan	5 - 1 // 4

Operator	Keterangan	Contoh
*	Perkalian	4 * 2 // 8
/	Pembagian	8 / 2 // 4
%	Modulus (sisa)	5 % 2 // 1
**	Pangkat	2 ** 3 // 8
++	Increment	1++ // 2
--	Decrement	3-- // 2

2) Operator Perbandingan dan Logika

Operator	Keterangan	Contoh
==	Sama (banding long coercion)	5 == '5' // true
====	Sama mutlak	5 === '5' // false
!=	Tidak sama	5 != 3 // true
!==	Tidak sama mutlak	5 !== '5' // true
> / <	Lebih besar / kecil	4 > 3 // true
&&	AND	true && false // false
	OR	true false // true
!	NOT	!true // false

3) Operator Penugasan

Operator	Contoh	Keterangan
=	a = 5	Assign
+=	a += 2	a = a + 2
-=	a -= 1	a = a - 1
*=	a *= 3	a = a * 3
/=	a /= 2	a = a / 2

4. Praktik: Menambahkan alert() dan console.log()

Kita akan membuat file script.js dan memanggilnya di index.html untuk menampilkan pesan saat halaman dimuat atau tombol diklik.

a. Sesuaikan file index.html (potongan)

```
<!DOCTYPE html>
<html lang="id">
<head>
<!-- ... -->
```

```
</head>
<body>
    <!-- ... konten halaman ... -->

    <!-- panggil JavaScript di akhir body -->
    <script src="script.js"></script>
</body>
</html>
```

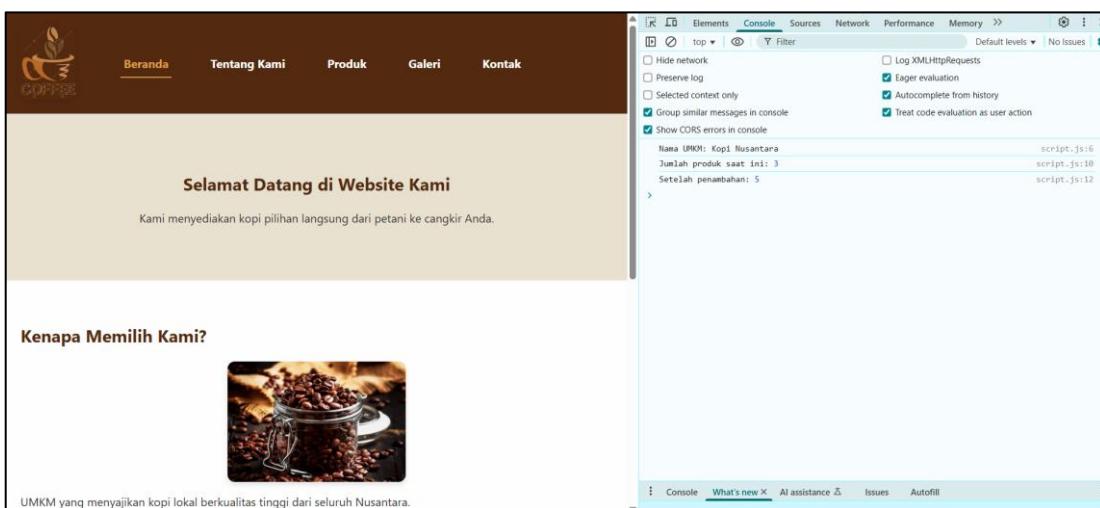
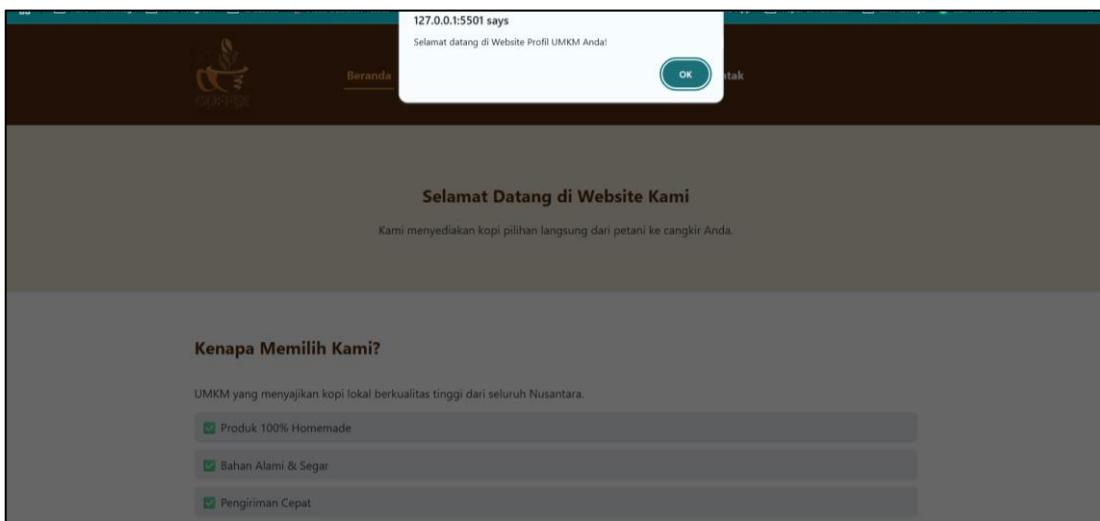
b. Tambahkan kode di file script.js

```
// Memberi tahu mahasiswa bahwa script berhasil di-load
alert('Selamat datang di Website Profil UMKM Anda!');

// Menampilkan informasi variabel di console
const namaUMKM = 'Kopi Nusantara';
console.log('Nama UMKM:', namaUMKM);

// Demonstrasi variabel dan operator
let produk = 3;
console.log('Jumlah produk saat ini:', produk);
produk += 2;
console.log('Setelah penambahan:', produk);
```

c. Tampilan halaman website



d. Penjelasan Kode Praktik

Baris Kode	Penjelasan
<code>alert('...')</code>	Menampilkan jendela pop-up saat halaman dimuat
<code>const namaUMKM = '...'</code>	Deklarasi variabel konstan
<code>console.log('Nama UMKM: ', ...)</code>	Menulis teks dan nilai variabel ke Console di DevTools
<code>let produk = 3;</code>	Deklarasi variabel bisa diubah
<code>produk += 2;</code>	Operator penugasan menambahkan 2 ke nilai produk

5. Tugas Mahasiswa

Petunjuk:

1. Buat file script.js di folder proyek dan hubungkan ke index.html.
2. Tambahkan beberapa alert() untuk menyapa pengguna dengan nama usaha masing-masing.
3. Gunakan console.log() untuk menampilkan:
 - o Nama UMKM (variabel const)
 - o Jumlah produk awal, lalu modifikasi dengan operator (let)
 - o Buat variabel untuk penerapan hal lain juga, misal jumlah produk berdasarkan jenisnya, dll
4. Buka Developer Tools (F12) untuk melihat output console.log.
5. Kirimkan tangkapan layar pop-up dan console sebagai bukti praktik.

Pertemuan Minggu Ke-10 – Kontrol Alur & Fungsi

Tujuan Pembelajaran

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- Menggunakan struktur kontrol if, else, switch dalam pengambilan keputusan.
- Mengimplementasikan perulangan for, while, forEach.
- Menulis dan memanggil fungsi sendiri untuk membuat kode modular dan reusable.
- Mengintegrasikan kontrol alur dan fungsi ke proyek Website Profil UMKM.

1. Materi Teori

1) Percabangan (Conditional Statements)

Percabangan digunakan untuk mengevaluasi suatu kondisi dan mengambil keputusan berdasarkan hasilnya.

a. if, else, else if

```
let rating = 4.5;

if (rating >= 4.5) {
    console.log("Sangat Bagus");
} else if (rating >= 3) {
    console.log("Cukup Bagus");
} else {
    console.log("Perlu Peningkatan");
}
```

b. switch

Lebih cocok saat memiliki banyak nilai tetap.

```
let kategori = "makanan";

switch (kategori) {
    case "makanan":
        console.log("Kategori Makanan");
        break;
    case "pakaian":
        console.log("Kategori Pakaian");
        break;
    default:
        console.log("Kategori Lainnya");
}
```

2) Perulangan (Looping)

Perulangan digunakan untuk mengeksekusi kode berulang kali.

a. for loop

```
for (let i = 1; i <= 3; i++) {  
    console.log("Testimoni ke-" + i);  
}
```

b. while loop

```
let stok = 5;  
  
while (stok > 0) {  
    console.log("Stok tersisa: " + stok);  
    stok--;  
}
```

c. forEach untuk array

```
let layanan = ["Pengiriman Cepat", "Garansi 7 Hari", "Dukungan  
Pelanggan"];  
  
layanan.forEach(function(item, index) {  
    console.log("Layanan " + (index + 1) + ": " + item);  
});
```

3) Fungsi (Functions)

Membantu mengelompokkan kode agar lebih mudah digunakan kembali.

a. Fungsi biasa

```
function sapa(nama) {  
    console.log("Halo, " + nama + "!");  
}  
  
sapa("Reza");
```

b. Fungsi dengan Nilai Kembali (return)

```
function hitungDiskon(harga) {  
    return harga * 0.1;  
}  
  
let potongan = hitungDiskon(100000);  
console.log("Potongan: " + potongan);
```

2. Praktikum

Buka file produk.html (lanjutan proyek UMKM). Tambahkan <script> di bawah <body> atau dalam file JS eksternal (jika sudah dipisahkan di pertemuan berikutnya)

1) Praktikum Percabangan

Tujuan: Menentukan rekomendasi produk berdasarkan rating. Tambahkan kode berikut ke file projek masing-masing.

```
<h3>Rating Produk: <span id="hasil-rating"></span></h3>
<script>
    let rating = 4;

    if (rating >= 4.5) {
        document.getElementById("hasil-rating").textContent = "★★★★★
Sangat Disarankan";
    } else if (rating >= 3) {
        document.getElementById("hasil-rating").textContent = "★★★
Rekomendasi Biasa";
    } else {
        document.getElementById("hasil-rating").textContent = "★ Tidak
Direkomendasikan";
    }
</script>
```

The screenshot displays a website interface for a coffee brand. The header features a logo with a stylized coffee cup and the word 'COFFEE'. Below the logo are navigation links: Beranda, Tentang Kami, Produk, Galeri, and Kontak. The 'Tentang Kami' link is underlined, indicating it is the current page. The main content area is titled 'Profil Produk' and shows a card for 'Kopi Arabika Gayo'. The card contains an image of coffee beans, the text 'ARABIKA GAYO', and a description: 'Kopi khas Aceh dengan aroma segar dan rasa asam yang halus.'. Below the card, it says 'Rating Produk: ★★★ Rekomendasi Biasa'. At the bottom of the card, there is a link 'Dengarkan Cerita Kami'.

2) Praktikum Perulangan

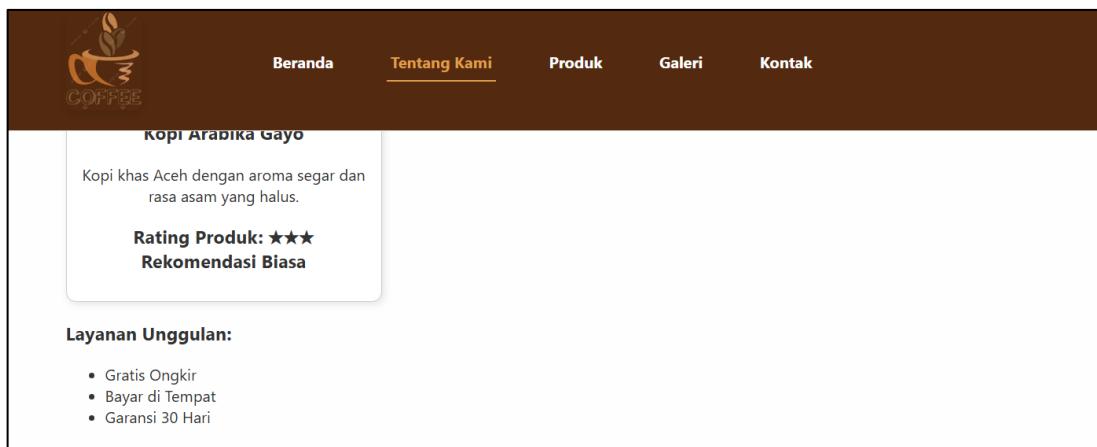
Tujuan: Menampilkan daftar layanan unggulan dari array secara dinamis.

Tambahkan kode berikut ke file projek masing-masing.

```
<h3>Layanan Unggulan:</h3>
<ul id="layanan-list"></ul>

<script>
  let layanan = ["Gratis Ongkir", "Bayar di Tempat", "Garansi 30
Hari"];

  layanan.forEach(function(item) {
    let li = document.createElement("li");
    li.textContent = item;
    document.getElementById("layanan-list").appendChild(li);
  });
</script>
```



3) Praktikum Fungsi

Tujuan: Membuat fungsi validasi sederhana pada form pemesanan.

Tambahkan kode berikut ke file projek masing-masing.

```
<h3>Form Pemesanan</h3>
<form onsubmit="return cekForm()">
  <input type="text" id="namaProduk" placeholder="Nama Produk"><br>
  <input type="number" id="jumlah" placeholder="Jumlah"><br>
  <button type="submit">Pesan</button>
</form>

<script>
  function cekForm() {
    let nama = document.getElementById("namaProduk").value;
```

```

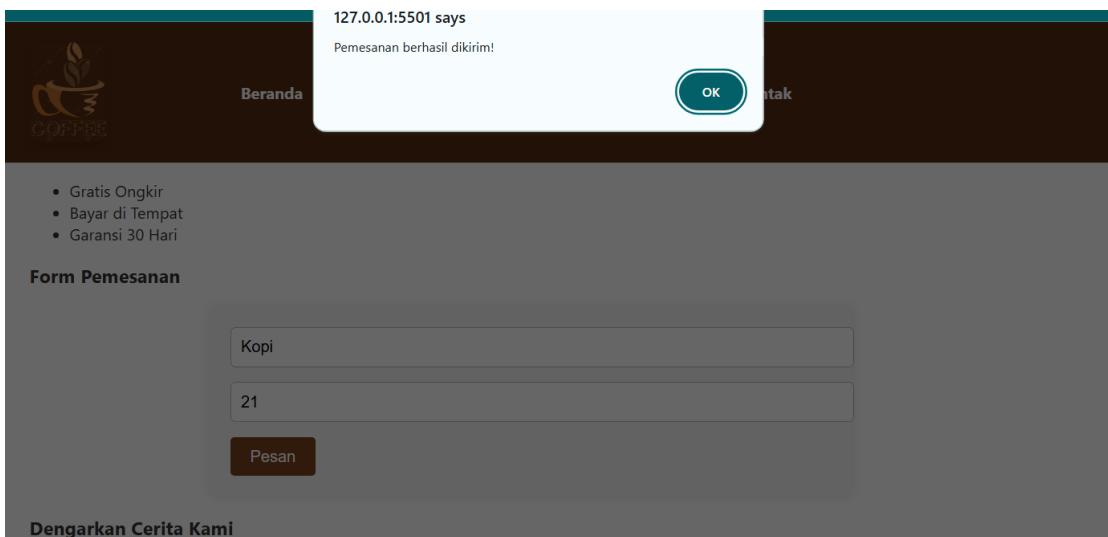
let jumlah = document.getElementById("jumlah").value;

if (nama === "" || jumlah === "") {
    alert("Semua kolom wajib diisi.");
    return false;
}

if (jumlah <= 0) {
    alert("Jumlah harus lebih dari 0");
    return false;
}

alert("Pemesanan berhasil dikirim!");
return true;
}
</script>

```



3. Penugasan Mandiri

Petunjuk:

- Lakukan penyesuaian di halaman produk untuk masing-masing projek mahasiswa dengan intruksi selanjutnya.
- Tugas Percabangan: Buat bagian/section/div yang menerima input usia pelanggan dan menampilkan status kategori:
 - Usia < 13 → "Anak-anak"
 - Usia 13–17 → "Remaja"
 - Usia 18–60 → "Dewasa"
 - Usia > 60 → "Lansia"

3. Tugas Perulangan: Buat bagian/section/div yang menampilkan 5 testimoni produk secara dinamis menggunakan array dan forEach().
4. Tugas Fungsi: Buat fungsi hitungTotal(harga, jumlah) yang menghitung total belanja, dan tampilkan hasilnya ke pengguna saat form dikirim.
5. Desain tampilan sesuai kreativitas masing-masing.
6. Kirimkan tangkapan layar sebagai bukti praktik.

Pertemuan Minggu Ke-11 – Manipulasi DOM dan Event

Tujuan Pembelajaran

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- Mahasiswa memahami konsep DOM (Document Object Model).
- Mahasiswa mampu menggunakan Event Listener untuk menangani interaksi pengguna.
- Mahasiswa dapat mengubah konten halaman secara dinamis menggunakan JavaScript.

1. Materi Teori

1) Apa Itu DOM?

DOM (Document Object Model) adalah representasi struktur halaman HTML sebagai pohon objek yang bisa dimanipulasi menggunakan JavaScript.

Ilustrasi DOM Tree:

```
<html>
  <body>
    <h1>Judul</h1>
    <p>Paragraf</p>
  </body>
</html>
```

Maka struktur DOM Tree-nya sebagai berikut:

```
html
└── body
    ├── h1
    └── p
```

2) Seleksi Elemen DOM

Terdapat berbagai cara untuk menggunakan elemen yang akan kita manipulasi. Berikut beberapa metode DOM untuk memilih elemen:

```
// Berdasarkan ID
const title = document.getElementById("judul");

// Berdasarkan class
const buttons = document.getElementsByClassName("btn");

// Seleksi modern (CSS Selector)
const input = document.querySelector("input[type='text']);
```

3) Manipulasi Konten

Manipulasi konten dapat digunakan untuk mengubah isi teks, HTML, atau atribut elemen. Berikut contoh penggunaannya.

```
title.textContent = "Judul Baru";
document.querySelector("#kontainer").innerHTML = "<p>Konten baru</p>";
input.setAttribute("placeholder", "Masukkan nama");
```

4) Event Listener

Event listener digunakan untuk menambahkan respon terhadap aksi pengguna seperti klik, input, dll.

```
const tombol = document.getElementById("submit");
tombol.addEventListener("click", function () {
    alert("Tombol diklik!");
});
```

5) Event Tipe Populer:

Event	Keterangan
click	Aksi dipicu saat elemen diklik
submit	Aksi dipicu saat form disubmit
change	Aksi dipicu saat input berubah
mouseover	Aksi dipicu saat kursor menyentuh elemen
keyup	Aksi dipicu saat tombol keyboard dilepas

2. Praktikum

Lakukan modifikasi proyek website pada bagian validasi form agar berfungsi secara interaktif.

1) Form kontak di file kontak.html

a. Langkah 1: Sesuaikan kode di bagian tag form pada Form HTML (file kontak.html)

```
<form id="formKontak" action="#" method="post">
    <label for="nama">Nama:</label><br>
    <input type="text" id="nama" name="nama" required /><br><br>

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

    <label for="kategori">Kategori Pesan:</label><br>
    <select id="kategori" name="kategori">
        <option value="umum">Umum</option>
```

```

<option value="produk">Produk</option>
<option value="kerjasama">Kerja Sama</option>
</select><br><br>

<label for="pesan">Pesan:</label><br>
<textarea id="pesan" name="pesan" rows="5"
cols="30"></textarea><br><br>

<input type="checkbox" id="langganan" name="langganan" value="ya">
<label for="langganan">Saya ingin menerima
newsletter</label><br><br>

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

<p id="errorMsg" style="color:red;"></p>
</form>

```

b. Langkah 2: Tambahkan JavaScript

```

document.addEventListener("DOMContentLoaded", function () {
    const form = document.getElementById("formKontak");
    const nama = document.getElementById("nama");
    const email = document.getElementById("email");
    const pesan = document.getElementById("pesan");
    const errorMsg = document.getElementById("errorMsg");

    form.addEventListener("submit", function (e) {
        e.preventDefault(); // mencegah reload halaman
        if (nama.value === "" || email.value === "" || pesan.value === "") {
            errorMsg.textContent = "Semua field harus diisi.";
        } else {
            errorMsg.textContent = "";
            alert("Pesanan berhasil dikirim!");
            form.reset();
        }
    });
});

```

Nama:

Email:

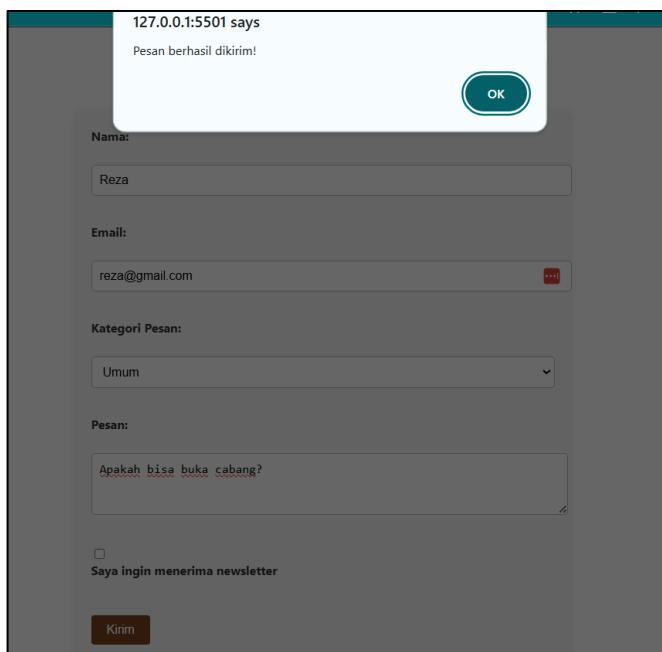
Kategori Pesan:

Pesan:

Saya ingin menerima newsletter

Kirim

Semua field harus diisi.



c. Penjelasan Kode

Bagian Kode	Penjelasan Singkat
document.getElementById()	Seleksi elemen berdasarkan ID
addEventListener("submit", ...)	Menambahkan aksi saat form dikirim
e.preventDefault()	Mencegah perilaku default (halaman reload)

Bagian Kode	Penjelasan Singkat
if (value === "")	Validasi input kosong
form.reset()	Mengosongkan seluruh field setelah submit
textContent / alert()	Menampilkan pesan atau feedback

2) Pembuatan wishlist & atur tema di file produk.html

a. Tambahkan kode berikut di file produk.html bagian bawah sebelum footer

```
<h3>Daftar Wishlist Produk</h3>

<form>
  <input type="text" id="produkInput" placeholder="Nama produk..." />
  <p>Preview: <span id="previewProduk"></span></p>
  <button id="tambahBtn">Tambah Produk</button>
</form>

<ul id="daftarProduk"></ul>

<hr />

<label for="temaSelect">Pilih Tema Warna: </label>
<select id="temaSelect">
  <option value="white">Putih</option>
  <option value="lightblue">Biru Muda</option>
  <option value="lightyellow">Kuning Muda</option>
</select>
```

b. Tambahkan javascript

```
document.addEventListener("DOMContentLoaded", function () {
  const produkInput = document.getElementById("produkInput");
  const tambahBtn = document.getElementById("tambahBtn");
  const daftarProduk = document.getElementById("daftarProduk");
  const previewProduk = document.getElementById("previewProduk");
  const temaSelect = document.getElementById("temaSelect");

  // Event 1: Tambah Produk ke daftar (click)
  tambahBtn.addEventListener("click", function (e) {
    e.preventDefault();

    const namaProduk = produkInput.value.trim();
    if (namaProduk !== "") {
      const li = document.createElement("li");
      li.textContent = namaProduk;
      daftarProduk.appendChild(li);
    }
  });
});
```

```

    // Tambahkan Event Hover (mouseover & mouseout)
    li.addEventListener("mouseover", () =>
li.style.backgroundColor = "#e0e0e0");
    li.addEventListener("mouseout", () => li.style.backgroundColor
= "");

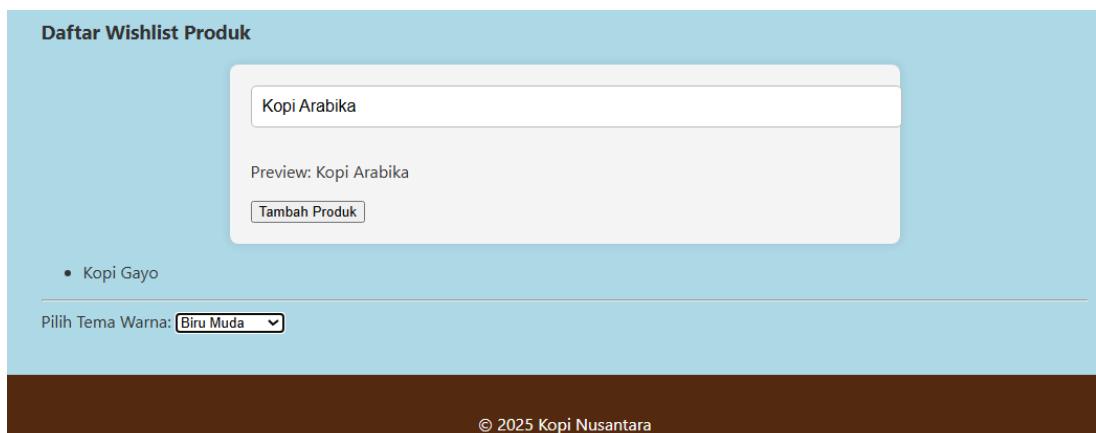
    // Tambahkan Event Hapus (dblclick)
    li.addEventListener("dblclick", () => {
        if (confirm("Yakin ingin menghapus produk ini?")) {
            li.remove();
        }
    });

    daftarProduk.appendChild(li);
    produkInput.value = "";
    previewProduk.textContent = "";
}
});

// Event 2: Preview input secara langsung (keyup)
produkInput.addEventListener("keyup", function () {
    previewProduk.textContent = produkInput.value;
});

// Event 3: Ubah tema background (change)
temaSelect.addEventListener("change", function () {
    document.body.style.backgroundColor = temaSelect.value;
});

```



c. Penjelasan kode

Event Listener	Digunakan Pada	Kegunaan
click	Tombol "Tambah Produk"	Menjalankan fungsi saat tombol diklik
mouseover	Item daftar produk	Mengubah warna latar saat mouse diarahkan
mouseout	Item daftar produk	Mengembalikan warna latar saat mouse keluar
dblclick	Item daftar produk	Menghapus item saat diklik dua kali
keyup	Input produk	Menampilkan preview real-time saat pengguna mengetik
change	Dropdown tema	Mengubah warna latar halaman

3. Penugasan Mahasiswa

Petunjuk:

- 1) Lakukan penyesuaian di halaman kontak untuk masing-masing projek mahasiswa dengan intruksi selanjutnya
- 2) Tambahkan satu input baru: **Nomor HP** dan validasi agar hanya angka.
- 3) Tambahkan fitur tombol "Clear Data" untuk menghapus semua inputan.
- 4) Buat tampilan feedback sukses di halaman seperti pesan error (bukan alert).
- 5) Tambahkan gaya (CSS) agar form tampil lebih menarik.
- 6) Kirimkan tangkapan layar sebagai bukti praktik.

Pertemuan Minggu Ke-12 – Validasi Form Lanjutan

Tujuan Pembelajaran

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- Memahami konsep validasi form sisi klien menggunakan JavaScript.
- Menggunakan event submit, blur, input, dan change untuk validasi interaktif.
- Mengelola pesan error dinamis dan interaksi validasi secara real-time.
- Melakukan validasi custom (misalnya: email domain tertentu, panjang karakter, pilihan wajib).

1. Materi Teori: Validasi Form pada JavaScript

Validasi form adalah proses untuk memastikan data yang diisi user sesuai aturan sebelum dikirim ke server. Validasi bisa dilakukan di dua sisi:

- 1) **Client-side** (di browser, menggunakan JavaScript)
- 2) **Server-side** (di backend, misalnya PHP/Laravel)

Validasi di sisi klien membantu memberikan umpan balik cepat ke pengguna tanpa perlu request ke server.

Berikut beberapa event yang terkait validasi:

Event	Kegunaan
submit	Menangani saat form dikirim
blur	Saat elemen kehilangan fokus
input	Saat isi input berubah
change	Saat nilai input (khusus select/checkbox/radio) berubah

Beberapa Teknik Validasi Umum:

- Mengecek apakah input kosong
- Memastikan format email benar
- Panjang minimum/maksimum karakter
- Validasi pilihan tertentu (dropdown, checkbox)
- Validasi custom (misalnya: hanya menerima email @gmail.com)

2. Praktikum

Kita akan menggunakan form dari pertemuan sebelumnya, namun akan ditambahkan untuk menampilkan error di bawah masing-masing field:

```
<form id="formKontak" action="#" method="post">
    <label for="nama">Nama:</label><br>
    <input type="text" id="nama" name="nama" />
    <span class="error" id="errorNama"></span><br><br>

    <label for="email">Email:</label><br>
    <input type="email" id="email" name="email" />
    <span class="error" id="errorEmail"></span><br><br>

    <label for="kategori">Kategori Pesan:</label><br>
    <select id="kategori" name="kategori">
        <option value="">-- Pilih Kategori --</option>
        <option value="umum">Umum</option>
        <option value="produk">Produk</option>
        <option value="kerjasama">Kerja Sama</option>
    </select>
    <span class="error" id="errorKategori"></span><br><br>

    <label for="pesan">Pesan:</label><br>
    <textarea id="pesan" name="pesan" rows="5" cols="30"></textarea>
    <span class="error" id="errorPesanan"></span><br><br>

    <input type="checkbox" id="langganan" name="langganan" value="ya">
    <label for="langganan">Saya ingin menerima newsletter</label><br><br>

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

Tambahkan CSS untuk error:

```
.error {
    color: red;
    font-size: 0.8rem;
}
```

Praktikum 1: Validasi Saat Submit

```
const form = document.getElementById("formKontak");

form.addEventListener("submit", function (event) {
    let valid = true;

    // Nama harus diisi
```

```
const nama = document.getElementById("nama");
const errorNama = document.getElementById("errorNama");
if (nama.value.trim() === "") {
    errorNama.textContent = "Nama wajib diisi.";
    valid = false;
} else {
    errorNama.textContent = "";
}

// Email harus valid dan domain harus @gmail.com
const email = document.getElementById("email");
const errorEmail = document.getElementById("errorEmail");
const emailPattern = /^[^ ]+@gmail\.com$/i;
if (!emailPattern.test(email.value)) {
    errorEmail.textContent = "Email harus @gmail.com dan format benar.";
    valid = false;
} else {
    errorEmail.textContent = "";
}

// Kategori harus dipilih
const kategori = document.getElementById("kategori");
const errorKategori = document.getElementById("errorKategori");
if (kategori.value === "") {
    errorKategori.textContent = "Pilih salah satu kategori.";
    valid = false;
} else {
    errorKategori.textContent = "";
}

// Pesan minimal 10 karakter
const pesan = document.getElementById("pesan");
const errorPesam = document.getElementById("errorPesam");
if (pesan.value.trim().length < 10) {
    errorPesam.textContent = "Pesan minimal 10 karakter.";
    valid = false;
} else {
    errorPesam.textContent = "";
}

if (!valid) {
    event.preventDefault(); // Gagalkan submit
} else {
    alert("Pesan berhasil dikirim!");
    form.reset();
}

});
```

Praktikum 2: Validasi Real-time dengan Event input dan blur

```
document.getElementById("nama").addEventListener("blur", function () {
  const errorNama = document.getElementById("errorNama");
  if (this.value.trim() === "") {
    errorNama.textContent = "Nama wajib diisi.";
  } else {
    errorNama.textContent = "";
  }
});

document.getElementById("pesan").addEventListener("input", function () {
  const errorPesan = document.getElementById("errorPesan");
  if (this.value.trim().length < 10) {
    errorPesan.textContent = "Pesanan minimal 10 karakter.";
  } else {
    errorPesan.textContent = "";
  }
});
```

Praktikum 3: Validasi Checkbox

```
document.getElementById("langganan").addEventListener("change", function () {
  if (this.checked) {
    alert("Terima kasih telah berlangganan newsletter!");
  }
});
```

Penjelasan Praktikum

- Validasi dilakukan secara preventif saat form akan dikirim (submit) dan secara reaktif saat input berubah (input, blur).
- preventDefault() digunakan untuk mencegah form dikirim jika ada input tidak valid.
- Validasi regex (emailPattern) digunakan untuk mengecek pola email tertentu.

Nama:

Nama wajib diisi.

Email:
 ...
Email harus @gmail.com dan format benar.

Kategori Pesan:

Pilih salah satu kategori.

Pesan:

Pesan minimal 10 karakter.

Saya ingin menerima newsletter

Kirim

18.0.... 127.0.0.1:5501 says Web App

Terima kasih telah berlangganan newsletter!

OK

Email:
 ...
Email harus @gmail.com dan format benar.

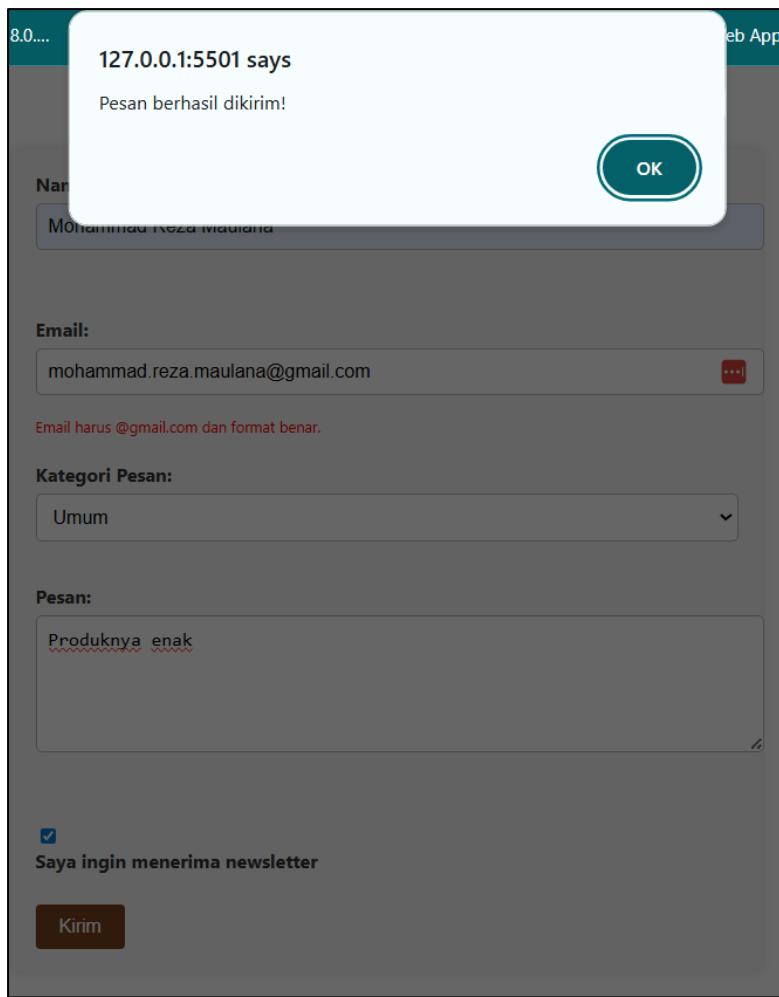
Kategori Pesan:

Pilih salah satu kategori.

Pesan:

Saya ingin menerima newsletter

Kirim



3. Penugasan Mahasiswa

Petunjuk:

- 1) Buatlah form pendaftaran webinar promosi produk yang terdiri dari:
 - Nama lengkap (wajib, minimal 3 karakter)
 - Email (wajib, harus valid dan bukan domain yahoo.com)
 - No HP (wajib, harus angka)
 - Pilihan topik webinar (dropdown, wajib pilih)
 - Checkbox persetujuan syarat & ketentuan (wajib dicentang)
- 2) Gunakan validasi lengkap menggunakan JavaScript:
 - Saat submit
 - Validasi real-time (input/blur/change)
 - Tampilkan pesan error secara dinamis
- 3) Kirimkan tangkapan layar sebagai bukti praktik.

Pertemuan Minggu Ke-13 – ES6 dan Modular JavaScript

Tujuan Pembelajaran

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- Memahami fitur-fitur baru dalam JavaScript versi ES6 (ECMAScript 2015)
- Menggunakan let, const, arrow function, template literal, destructuring, spread/rest operator, dan class
- Membuat dan mengelola kode JavaScript modular
- Mengimplementasikan konsep ES6 dan modul pada proyek Website Profil UMKM

1. Materi Teori: Fitur-Fitur Modern JavaScript (ES6)

1)let dan const

Sebelum ES6, variabel dideklarasikan menggunakan var. ES6 memperkenalkan:

- let: variabel yang dapat diubah nilainya
- const: variabel tetap (konstan)

```
let nama = "Reza";
const umur = 25;
```

2)Arrow Function

Bentuk penulisan fungsi yang lebih ringkas.

```
const sapa = (nama) => {
  return `Halo, ${nama}!`;
};
```

Jika hanya satu baris:

```
const sapa = nama => `Halo, ${nama}!`;
```

3)Template Literal

Untuk membuat string dinamis dengan \${}.

```
const nama = "Reza";
console.log(`Halo, nama saya ${nama}`);
```

4)Destructuring

Ekstrak nilai dari array atau object ke dalam variabel.

```
const user = { nama: "Reza", umur: 25 };
```

```
const { nama, umur } = user;
```

5) Spread dan Rest Operator

- Spread (...) untuk menyebarkan elemen
- Rest (...) untuk menggabungkan sisa nilai

```
const arr1 = [1, 2];
const arr2 = [...arr1, 3, 4]; // [1, 2, 3, 4]

function tampilan(...nilai) {
  console.log(nilai);
}

tampilan(1, 2, 3); // [1, 2, 3]
```

6) Class dan Constructor

Struktur seperti OOP (Object-Oriented Programming)

```
class Produk {
  constructor(nama, harga) {
    this.nama = nama;
    this.harga = harga;
  }

  tampilan() {
    return `${this.nama} - Rp${this.harga}`;
  }
}

const kopi = new Produk("Kopi Hitam", 10000);
console.log(kopi.tampilan());
```

2. Materi Teori: Modularisasi JavaScript

Mengapa Modular?

- Kode lebih terstruktur
- Lebih mudah diuji dan dikelola
- Menghindari variabel global

Cara Modularisasi (ES Modules)

Misal kita punya file:

file: utils.js

```
export function formatHarga(harga) {
  return `Rp${harga.toLocaleString("id-ID")}`;
}
```

file: produk.js

```
import { formatHarga } from "./utils.js";

const harga = 12000;
console.log(formatHarga(harga));
```

Pastikan HTML kita ditambahkan type="module":

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

3. Praktikum: Modularisasi dan ES6 pada Website Profil UMKM

Lanjutan dari proyek pertemuan sebelumnya

1) Struktur Folder Baru

Tambahkan folder dan file dengan struktur:

```
/js/
├── form.js
└── utils.js
```

2) Isi file utils.js

```
export function formatPesan(nama, kategori, pesan) {
    return `Halo, saya ${nama} ingin menyampaikan pesan mengenai
${kategori}: ${pesan}`;
}
```

3) Isi file form.js

```
import { formatPesan } from "./utils.js";

document.addEventListener("DOMContentLoaded", () => {
    const form = document.querySelector("form");

    form.addEventListener("submit", (e) => {
        e.preventDefault();

        const nama = document.getElementById("nama").value;
        const kategori = document.getElementById("kategori").value;
        const pesan = document.getElementById("pesan").value;

        const hasil = formatPesan(nama, kategori, pesan);
        alert(hasil);
    });
});
```

```
});  
});
```

4) Isi file newsletter.js

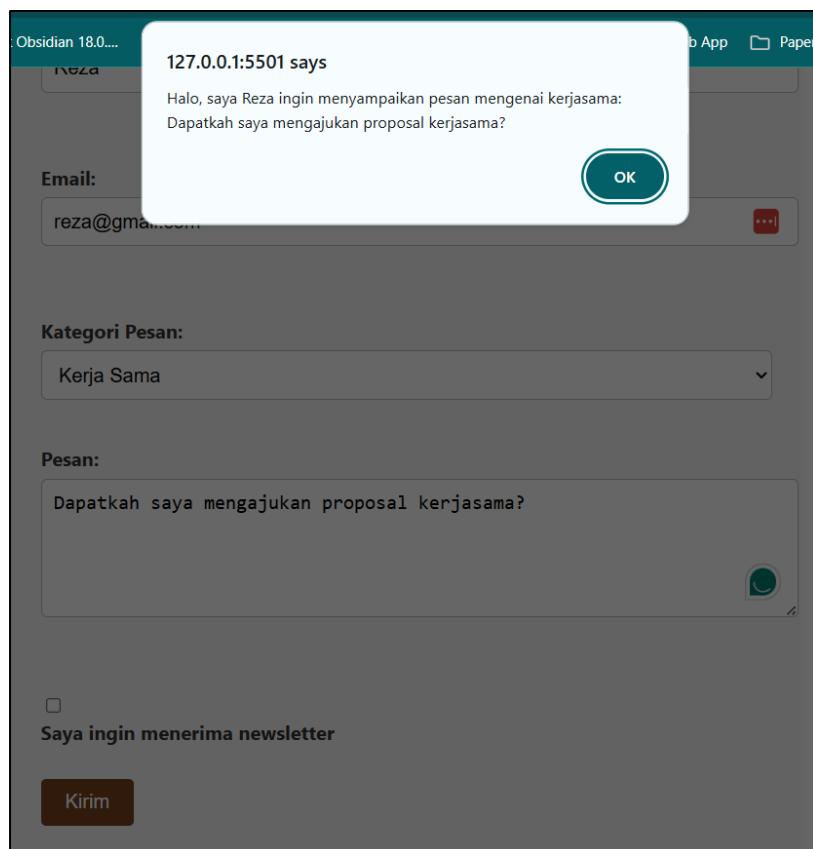
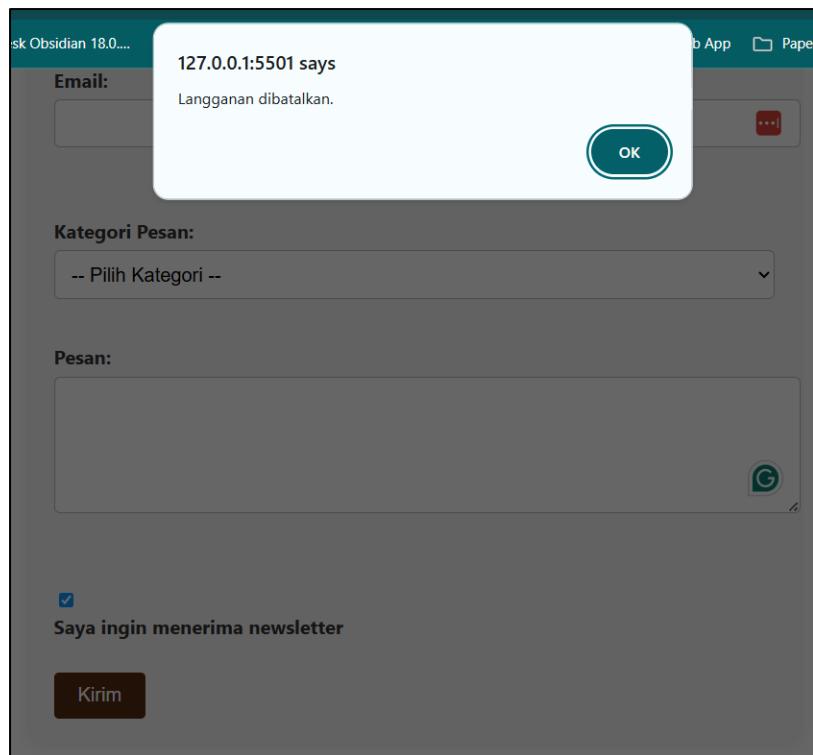
```
document.addEventListener("DOMContentLoaded", () => {  
    const checkbox = document.getElementById("langganan");  
  
    checkbox.addEventListener("change", () => {  
        if (checkbox.checked) {  
            alert("Terima kasih telah berlangganan!");  
        } else {  
            alert("Langganan dibatalkan.");  
        }  
    });  
});
```

5) Update file kontak.html

Tambahkan sricpt dan juga update beberapa kode di bawah ini.

```
.....  
.....  
  
<script type="module" src="js/form.js"></script>  
<script type="module" src="js/newsletter.js"></script>  
  
.....  
.....  
  
if (!valid) {  
    event.preventDefault(); // Gagalkan submit  
} else {  
    // beri komentar pada else disini karena sudah menggunakan form.js  
    // alert("Pesanan berhasil dikirim!");  
    // form.reset();  
}  
.....  
.....
```

Catatan : Jika ingin lebih efisien dan clean code, kode penanganan form yang ada sebelumnya bisa dijadikan satu dengan kode yang ada di form.js (silakan lakukan modifikasi sendiri).



Penjelasan Praktikum

- **formatPesan()** diekspor dari utils.js sebagai fungsi modular
- File form.js hanya menangani submit form, tidak mencampur logika
- File newsletter.js fokus pada interaksi dengan checkbox
- Modularisasi meningkatkan keterbacaan dan pengelolaan kode

4. Penugasan

Petunjuk:

- 1) Buat modul JavaScript baru bernama galeri.js untuk menangani interaksi berikut:
 - Saat gambar diklik, tampilkan nama produk di bawahnya.
 - Tambahkan efek hover menggunakan JavaScript classList.
- 2) Terapkan minimal 5 fitur ES6 di dalam modul galeri.js.
- 3) Integrasikan modul galeri.js dengan halaman galeri pada proyek UMKM Anda.
- 4) Kirimkan tangkapan layar sebagai bukti praktik.

Pertemuan Minggu Ke-14 – Library Pendukung (Bootstrap/jQuery)

Tujuan Pembelajaran

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- Memahami konsep dan manfaat penggunaan library/framework CSS seperti Bootstrap.
- Menggunakan komponen Bootstrap untuk mempercepat pembuatan antarmuka.
- Memahami dasar penggunaan jQuery untuk mempermudah manipulasi DOM dan event.
- Mengintegrasikan Bootstrap dan jQuery ke dalam proyek UMKM yang sudah dibuat sebelumnya.

1. Materi Teori

1) Apa itu Library dan Framework CSS?

Library adalah kumpulan kode siap pakai yang membantu mempercepat pengembangan (contoh: jQuery). Sedangkan framework CSS adalah sekumpulan aturan dan komponen CSS siap pakai yang memudahkan membuat desain responsif (contoh: Bootstrap, Tailwind CSS)

Kelebihan menggunakan library/framework:

- Cepat dan konsisten dalam desain.
- Sudah teruji di berbagai perangkat.
- Tidak perlu menulis CSS dari nol.

Kekurangan:

- Tampilan bisa generik jika tidak dimodifikasi.
- Bisa menambah ukuran file website.

2) Pengenalan Bootstrap

Bootstrap adalah framework front-end yang populer, gratis, dan open-source yang digunakan untuk mengembangkan situs web dan aplikasi web yang responsif dan mobile-first. Secara sederhana, Bootstrap menyediakan kumpulan kode HTML, CSS, dan JavaScript siap pakai yang memudahkan pengembang membangun tampilan website yang menarik dan konsisten di berbagai perangkat. Versi terbaru dari bootstrap saat ini adalah versi 5.x (diakses bulan Agustus 2025). Untuk informasi lebih lanjut silakan kunjungi <https://getbootstrap.com/>.

Terdapat beberapa cara menggunakan bootstrap, salah satunya menggunakan CDN seperti kode berikut:

```
<!-- CSS Bootstrap -->
<link
  href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
<!-- JS Bootstrap -->
<script
  src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
```

Beberapa fitur utama dari Bootstrap, yaitu:

- Grid System → Membuat layout responsif dengan container, row, col.
- Komponen siap pakai → Navbar, Card, Button, Modal, dll.
- Utility Classes → Margin (m-3), Padding (p-2), Warna (bg-primary, text-light).

Berikut contoh potongan kode bootstrap

```
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Nama Produk</h5>
    <p class="card-text">Deskripsi singkat produk UMKM Anda.</p>
    <a href="#" class="btn btn-primary">Beli Sekarang</a>
  </div>
</div>
```

Penjelasan kode diatas:

Class Bootstrap	Fungsi
.card	Membuat wadah card
.card-img-top	Menempatkan gambar di atas card
.card-body	Konten utama card
.btn .btn-primary	Membuat tombol dengan warna utama

3) Pengenalan jQuery

jQuery adalah library (pustaka) JavaScript yang populer yang mempermudah dan mempercepat pengembangan web. Pustaka ini menyediakan berbagai fungsi dan metode yang menyederhanakan tugas-tugas seperti manipulasi DOM (Document Object Model), penanganan peristiwa (event handling), animasi, dan AJAX. Dengan jQuery,

pengembang dapat menulis kode JavaScript yang lebih ringkas dan efisien, serta kompatibel di berbagai browser.

Seperti halnya bootstrap, terdapat beberapa cara untuk menggunakan JQuery, salah satunya via CDN sebagai berikut:

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

Berikut contoh perbandingan antara penggunaan JQuery dan tanpa JQuery untuk memberikan warna pada komponen:

```
// tanpa JQuery
document.getElementById("judul").style.color = "red";

// dengan JQuery
$("#judul").css("color", "red");
```

Berikut contoh penggunaan event handling (contoh: saat tombol diklik):

```
$(document).ready(function() {
    $("#tombol").click(function() {
        alert("Tombol diklik!");
    });
});
```

2. Praktikum

1) Langkah Integrasi ke Proyek UMKM

- a. Tambahkan link CDN Bootstrap dan jQuery ke file HTML utama (index.html, tentang.html, produk.html, kontak.html).
- b. Ubah navbar dan footer agar menggunakan komponen Bootstrap.
- c. Modifikasi halaman produk untuk menggunakan Card Bootstrap.
- d. Tambahkan event jQuery sederhana untuk interaksi.

2) Contoh Implementasi di Proyek

Sebelum implementasi, silakan copykan projek latihan ini. Kita akan menggunakan projek copy-an untuk mempraktekkan penggunaan Bootstrap. Silakan sesuaikan navbar di masing-masing file seperti berikut:

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark sticky-top">
    <div class="container">
        <a class="navbar-brand" href="#">
            
        </a>
        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav">
```

```

        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ms-auto">
            <li class="nav-item"><a class="nav-link active" href="index.html">Beranda</a></li>
            <li class="nav-item"><a class="nav-link" href="tentang.html">Tentang Kami</a></li>
            <li class="nav-item"><a class="nav-link" href="produk.html">Produk</a></li>
            <li class="nav-item"><a class="nav-link" href="galeri.html">Galeri</a></li>
            <li class="nav-item"><a class="nav-link" href="kontak.html">Kontak</a></li>
        </ul>
    </div>
</div>
</nav>

```



Kemudian lakukan penyesuaian card produk di file produk.html serta tambahan kode javascript dalam penggunaan JQuery

```

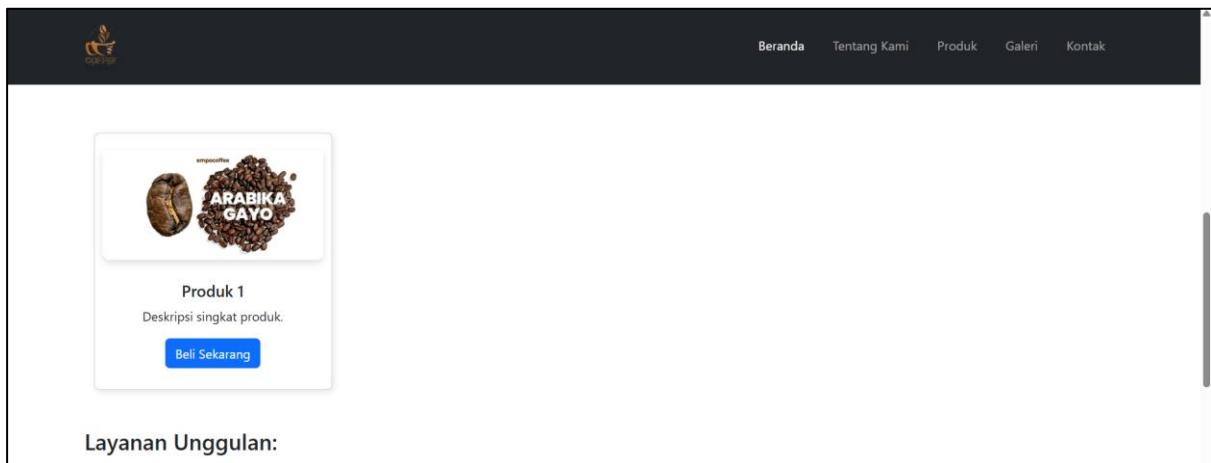
.....
.....
<div class="container my-5">
    <div class="row g-4">
        <div class="col-md-4">
            <div class="card">
                
                <div class="card-body">
                    <h5 class="card-title">Produk 1</h5>
                    <p class="card-text">Deskripsi singkat produk.</p>

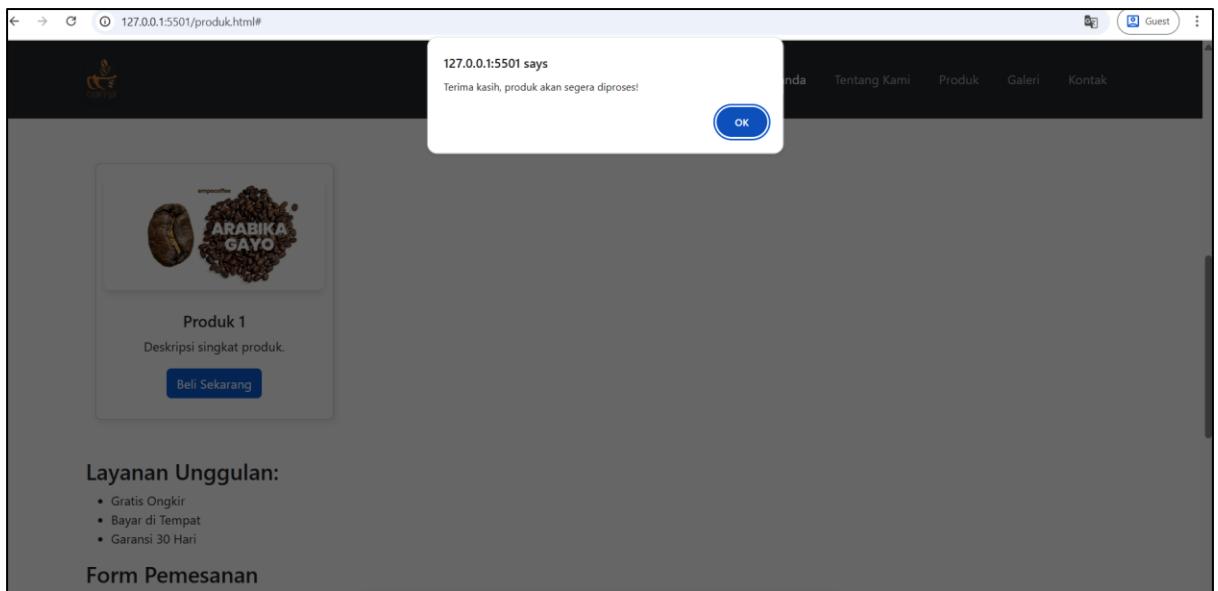
```

```
        <a href="#" class="btn btn-primary">Beli Sekarang</a>
    </div>
</div>
</div>
<!-- Tambahkan card lain di sini --&gt;
&lt;/div&gt;
&lt;/div&gt;

.....
.....

&lt;script&gt;
$(document).ready(function(){
    $(".btn-primary").click(function(e){
        e.preventDefault();
        alert("Terima kasih, produk akan segera diproses!");
    });
});
&lt;/script&gt;</pre>
```





3. Penugasan

Petunjuk:

- 1) Copy projek anda masing-masing. Kemudian projek yang telah dicopy lakukan penyesuaian sesuai ketentuan selanjutnya.
- 2) Ubah semua halaman proyek UMKM menjadi menggunakan Bootstrap untuk layout dan styling.
- 3) Gunakan minimal 5 komponen Bootstrap yang berbeda (contoh: Card, Navbar, Alert, dll).
- 4) Tambahkan minimal 3 event jQuery untuk interaksi pengguna (contoh: klik tombol, hover gambar, dll).
- 5) Dokumentasikan perubahan yang dilakukan dan jelaskan manfaatnya.

Pertemuan 16 – UAS