

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama masa magang, saya diberikan tanggung jawab utama untuk mengerjakan proyek perancangan dan pengembangan website sebagai media penyampaian informasi perusahaan. Proyek ini mencakup proses perancangan desain antarmuka, pengembangan halaman utama website, serta integrasi database untuk pengelolaan data produk dan resep. Pelaksanaan proyek dilakukan secara bertahap, dimulai dari tahap perencanaan dan desain sistem hingga proses implementasi teknis. Implementasi tersebut meliputi pengembangan sisi frontend dan backend agar website dapat berfungsi secara optimal sesuai dengan kebutuhan perusahaan.

Selama proses pengembangan website, digunakan beberapa bahasa pemrograman dan teknologi web, yaitu HTML, CSS, dan JavaScript. HTML berperan dalam membangun struktur halaman website, sedangkan CSS digunakan untuk mengatur tampilan dan tata letak agar terlihat rapi dan konsisten. JavaScript digunakan untuk mengelola berbagai logika dan fungsi pada website sehingga fitur yang tersedia dapat berjalan dengan baik. Dengan penerapan teknologi tersebut, website mampu menyajikan informasi secara optimal sesuai dengan kebutuhan perusahaan.

Pengembangan website tidak hanya berfokus pada sisi frontend, tetapi juga mencakup penanganan bagian backend. Backend website dikembangkan untuk mengelola proses pengolahan data serta integrasi dengan database. Dalam mendukung proses pengembangan tersebut, digunakan perangkat lunak Visual Studio Code sebagai editor kode. Selain itu, Supabase dimanfaatkan sebagai sistem database untuk menyimpan dan mengelola data yang digunakan pada website.

Dalam proyek ini, tanggung jawab penuh terhadap seluruh aspek pengembangan website dijalankan secara mandiri oleh saya. Saya bertanggung jawab mulai dari perancangan desain, pengembangan fitur, hingga implementasi

teknis pada sisi frontend dan backend website. Meskipun pengerjaan dilakukan secara mandiri, tetap terdapat kolaborasi terbatas dengan beberapa pihak internal perusahaan. Pihak pemasaran berperan dalam menyediakan informasi terkait produk yang akan ditampilkan pada website, termasuk deskripsi dan detail pendukung lainnya. Selain itu, pihak humas memberikan data dan informasi mengenai profil perusahaan yang diperlukan sebagai konten website. Informasi yang diperoleh dari masing-masing pihak tersebut kemudian diolah dan disesuaikan dengan struktur website yang telah dirancang. Selama proses pengembangan, saya juga secara berkala mendapatkan evaluasi dan masukan dari mentor magang. Evaluasi tersebut bertujuan untuk menyesuaikan fitur, tampilan, serta fungsionalitas website agar sesuai dengan kebutuhan dan standar perusahaan. Melalui proses ini, website dikembangkan secara bertahap hingga mencapai hasil yang diharapkan.

Selama proses pengembangan website, terdapat berbagai kendala dan tantangan yang harus saya hadapi. Tantangan tersebut antara lain berupa munculnya bug dan error pada sistem yang dikembangkan. Untuk mengatasi permasalahan tersebut, saya melakukan riset secara mandiri guna menemukan solusi yang tepat. Sebagai developer yang bekerja secara mandiri, kemampuan problem solving sangat diperlukan dalam menyelesaikan berbagai permasalahan teknis yang muncul selama proses pengembangan.

3.2 Tugas dan Uraian Kerja Magang

Selama pelaksanaan kerja magang di PT Kilang Kecap Angsa, saya diberikan tanggung jawab utama dalam pengembangan website perusahaan sebagai media penyampaian informasi dan promosi digital. Website yang dikembangkan mencakup beberapa halaman utama, seperti halaman beranda, produk, resep, detail produk, detail resep, serta halaman profil perusahaan. Pengembangan website ini bertujuan untuk membantu perusahaan dalam menyajikan informasi secara lebih terstruktur, mudah diakses, dan relevan dengan kebutuhan masyarakat. Seluruh proses pengembangan dilakukan untuk mendukung transformasi digital perusahaan melalui pemanfaatan teknologi informasi.

Tahapan awal dalam pengembangan website yaitu dengan melakukan analisis kebutuhan perusahaan secara menyeluruh. Dalam konteks PT Kilang Kecap Angsa, perusahaan membutuhkan website sebagai media informasi dan promosi untuk memperluas jangkauan pemasaran kepada masyarakat luas. Perusahaan memerlukan beberapa fitur utama seperti halaman Home, Produk, Resep sebagai pendukung promosi produk, About Us, serta halaman pendukung lainnya. Selain itu, website ini diharapkan mampu menyajikan informasi secara menarik, mudah diakses, dan meningkatkan citra profesional perusahaan.

Tahapan pengembangan website diawali dengan perancangan struktur dan tampilan antarmuka website. Pada tahap ini, saya menggunakan HTML dan CSS untuk membangun struktur halaman serta mengatur tata letak dan tampilan agar terlihat rapi dan konsisten. JavaScript digunakan untuk mendukung fungsi-fungsi tertentu pada website sehingga fitur yang disediakan dapat berjalan dengan baik. Perancangan dilakukan dengan memperhatikan kebutuhan konten yang akan ditampilkan, khususnya informasi produk dan resep yang menjadi fokus utama website.

Selain pengembangan pada sisi frontend, saya juga menangani pengembangan backend website. Backend dikembangkan untuk mengelola data produk dan resep agar dapat disimpan, ditampilkan, dan diperbarui dengan mudah. Dalam proses ini, Supabase digunakan sebagai sistem database untuk menyimpan dan mengelola data secara terpusat. Integrasi antara frontend dan database dilakukan agar website dapat menampilkan data secara dinamis sesuai dengan informasi yang tersedia di database.

Selama proses pengerjaan, saya mengerjakan proyek ini secara mandiri, mulai dari perancangan, pengembangan, hingga pengujian website. Meskipun demikian, terdapat kolaborasi dengan beberapa pihak internal perusahaan. Pihak pemasaran memberikan informasi terkait produk yang akan ditampilkan, sedangkan pihak humas menyediakan data mengenai profil dan informasi

perusahaan. Informasi tersebut kemudian saya olah dan sesuaikan dengan kebutuhan tampilan serta struktur website.

Dalam pelaksanaan proyek, saya juga mendapatkan bimbingan dan evaluasi secara berkala dari mentor magang. Evaluasi ini dilakukan untuk memastikan bahwa fitur, tampilan, dan fungsionalitas website telah sesuai dengan kebutuhan perusahaan. Masukan yang diberikan menjadi dasar dalam melakukan perbaikan dan penyempurnaan website. Proses evaluasi ini membantu saya memahami standar pengembangan website yang diterapkan di lingkungan industri.

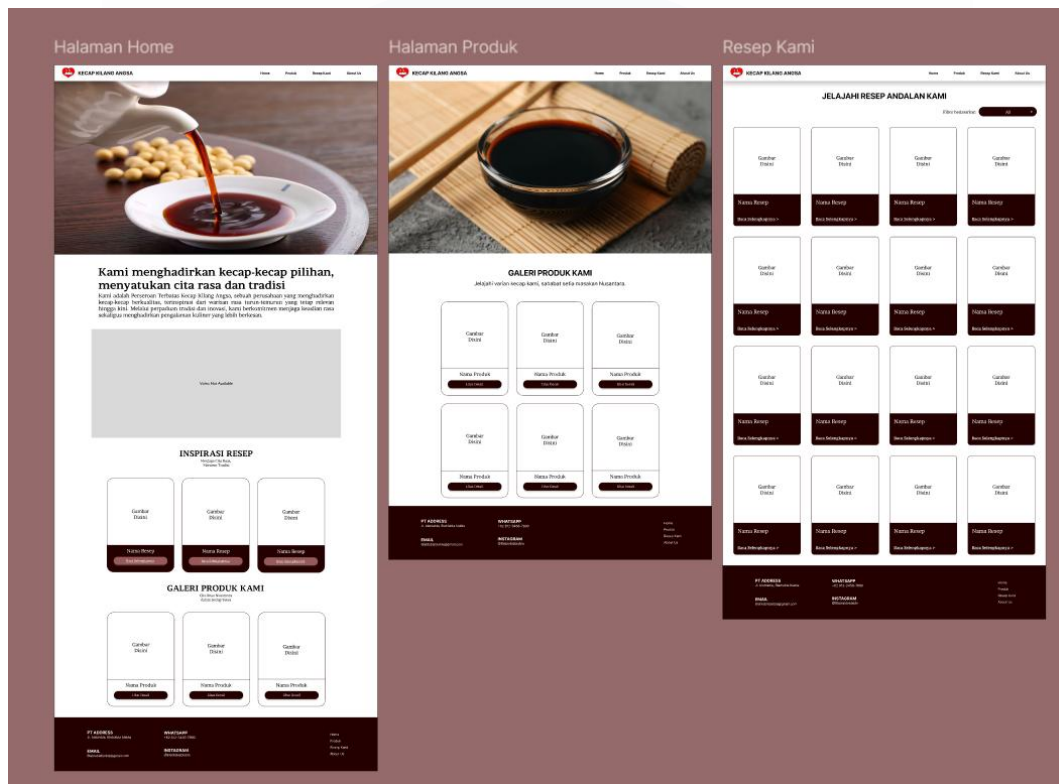
Selama proses pengembangan, beberapa kendala teknis turut dihadapi, terutama pada tahap pengkodean dan integrasi database. Beberapa bug dan error muncul selama pengujian fitur website. Untuk mengatasi kendala tersebut, saya melakukan riset secara mandiri dengan mempelajari dokumentasi, mencari referensi, serta mencoba berbagai solusi teknis yang relevan. Pendekatan ini membantu saya dalam menemukan solusi yang tepat sehingga pengembangan website dapat berjalan sesuai dengan rencana.

Untuk mendukung proses pengembangan, beberapa perangkat lunak dan alat bantu digunakan. Visual Studio Code dimanfaatkan sebagai editor kode dalam proses pengembangan frontend dan backend. Supabase digunakan sebagai database untuk pengelolaan data produk dan resep, sedangkan browser digunakan untuk melakukan pengujian dan memastikan website berjalan dengan baik. Dengan penggunaan alat bantu tersebut, proses pengembangan dapat dilakukan secara lebih efektif dan terstruktur.

Melalui pelaksanaan proyek ini, website PT Kilang Kecap Angsa berhasil dikembangkan sesuai dengan kebutuhan yang telah ditentukan. Pengalaman ini memberikan pemahaman yang mendalam mengenai proses pengembangan website secara menyeluruh, mulai dari perencanaan hingga implementasi teknis. Selain itu, pengalaman magang ini juga melatih kemampuan problem solving, manajemen waktu, serta komunikasi dengan pihak internal perusahaan. Pembelajaran yang diperoleh diharapkan dapat menjadi bekal yang bermanfaat bagi saya dalam pengembangan karier di bidang teknologi informasi dan pengembangan website di masa mendatang.

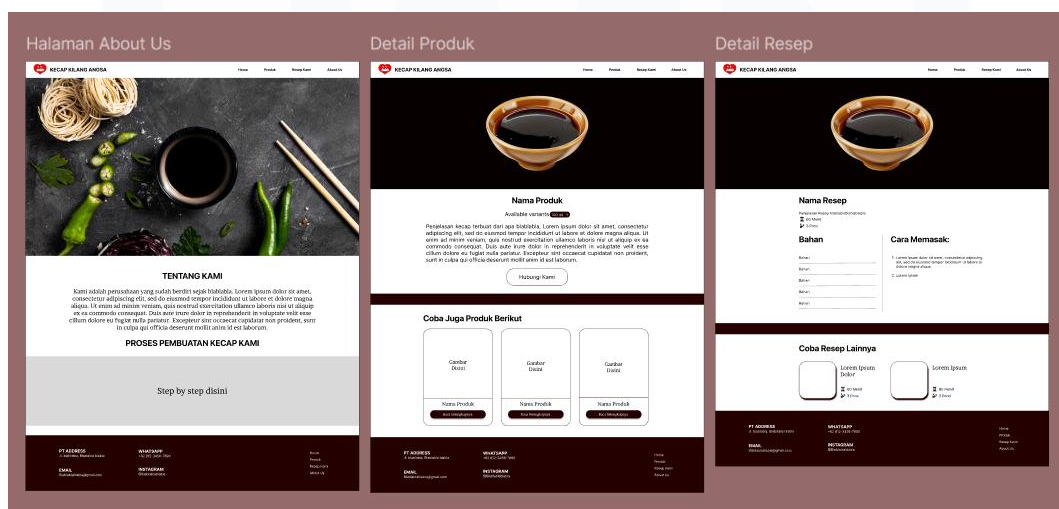
UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2.1 Desain Halaman menggunakan Figma



Gambar 3. 1 Desain Halaman Home, Produk dan Resep Kami

Gambar 3.1 merupakan desain awal dari halaman Home, Produk dan Resep Kami. Gambar-gambar tersebut memiliki beberapa unsur penting seperti header, banner, judul, penjelasan singkat, card, footer dan lain sebagainya.



Gambar 3. 2 Desain Halaman About Us, Detail Produk dan Detail Resep

Gambar 3.2 merupakan desain awal dari halaman About Us, Detail Produk, dan Detail Resep. Gambar-gambar tersebut memiliki beberapa unsur penting yang kurang lebih sama seperti gambar 3.1, yaitu header, gambar lengkap, banner, judul, penjelasan lengkap, footer dan lain sebagainya.

3.2.2 Header dan Footer Semua Halaman



Gambar 3. 3 Header Semua Halaman

Gambar 3.3 merupakan header website yang terdapat beberapa elemen utama yang berfungsi sebagai identitas dan navigasi pengguna. Header menampilkan logo perusahaan yang disertai dengan tulisan Kilang Kecap Angsa di sampingnya sebagai penegas identitas perusahaan. Selain itu, tersedia menu navigasi yang terdiri dari Home, Produk, Resep Kami, dan About Us yang berfungsi untuk memudahkan pengguna dalam mengakses halaman utama, melihat daftar produk, menjelajahi resep yang tersedia, serta memperoleh informasi mengenai perusahaan. Penempatan elemen-elemen tersebut dirancang secara terstruktur agar memudahkan pengguna dalam bernavigasi dan meningkatkan kenyamanan saat menggunakan website.



Gambar 3. 4 Header Responsive

Gambar 3.4 merupakan header responsive yang bisa terjadi ketika di zoom in sampai zoom tertentu. Header ini nantinya akan membuat semua navigasi masuk ke dalam bentuk logo menu pada kanan atas.



Home
Produk
Resep Kami
About Us

Gambar 3. 5 Navigasi Header Responsive

Gambar 3.5 merupakan isi dari menu dari header responsive pada gambar 3.6. Bisa dilihat di dalam menu ini terdapat navigasi yang tadinya disimpan dan ditampilkan ke dalam keadaan tulisan yang lebih besar dan jelas, yaitu Home, Produk, Resep Kami, dan About Us.



```
//Callable Header
class callableHeader extends HTMLElement {
  connectedCallback() {
    this.innerHTML = `
    <header>
      <div class="header-container">
        <!-- Header Kiri -->
        <div class="logo-area">
          
          <span class="brand">KILANG KECAP ANGSA</span>
        </div>

        <!-- Header Kanan (Desktop Nav) -->
        <nav class="desktop-nav">
          <ul>
            <li><a href="Home.html">Home</a></li>
            <li><a href="Produk.html">Produk</a></li>
            <li><a href="ResepKami.html">Resep Kami</a></li>
            <li><a href="AboutUs.html">About Us</a></li>
          </ul>
        </nav>

        <!-- Tombol Hamburger (Mobile) -->
        <div class="hamburger" id="hamburger">
          <span></span>
          <span></span>
          <span></span>
        </div>
      </div>
    </header>`
  }
}
```

Gambar 3. 6 Callable Header JS

Gambar 3.6 merupakan kode program untuk implementasi custom element menggunakan JavaScript dengan nama `callableHeader` yang berfungsi untuk menampilkan komponen header secara reusable pada setiap halaman website. Pada metode `connectedCallback()`, struktur header dibangun menggunakan HTML yang mencakup area logo dan nama perusahaan Kilang Kecap Angsa, menu navigasi utama yang terdiri dari Home, Produk, Resep Kami, dan About Us, serta tombol hamburger menu yang digunakan untuk tampilan navigasi pada perangkat mobile. Penerapan komponen ini bertujuan untuk menjaga konsistensi tampilan header dan memudahkan pengelolaan kode pada seluruh halaman website.

```

<!-- Overlay Navigasi -->
<div class="nav-overlay" id="navOverlay">
  <span class="close-btn" id="closeBtn">&times;</span>
  <ul>
    <li><a href="Home.html">Home</a></li>
    <li><a href="Produk.html">Produk</a></li>
    <li><a href="ResepKami.html">Resep Kami</a></li>
    <li><a href="AboutUs.html">About Us</a></li>
  </ul>
</div>
`;

const hamburger = this.querySelector("#hamburger");
const overlay = this.querySelector("#navOverlay");
const closeBtn = this.querySelector("#closeBtn");

hamburger.addEventListener("click", () => {
  overlay.classList.add("active");
});

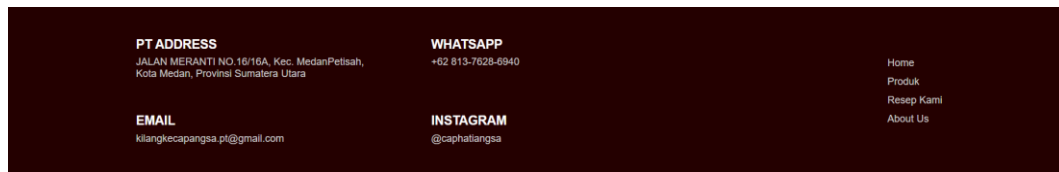
closeBtn.addEventListener("click", () => {
  overlay.classList.remove("active");
});
}

```

Gambar 3. 7 Lanjutan Callable Header JS

Gambar 3.7 merupakan kode overlay navigasi yang digunakan untuk tampilan menu pada perangkat mobile. Elemen `<div class="nav-overlay">` berfungsi sebagai layar pop-up yang muncul ketika tombol hamburger ditekan. Di dalam overlay terdapat tombol close (`×`) untuk menutup menu, serta daftar menu navigasi yang sama dengan versi desktop, yaitu Home, Produk, Resep Kami, dan About Us. Penerapan overlay ini memungkinkan navigasi yang responsif dan menjaga konsistensi menu di berbagai ukuran layar tanpa mengganggu tampilan utama halaman.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3. 8 Footer Semua Halaman

Gambar 3.8 merupakan footer website yang menampilkan informasi pendukung yang berfungsi sebagai sumber informasi dan akses tambahan bagi pengguna. Footer memuat alamat perusahaan, alamat email, serta kontak WhatsApp yang dapat diklik langsung untuk memudahkan pengguna dalam melakukan komunikasi. Selain itu, terdapat ikon media sosial Instagram yang terhubung langsung ke akun resmi perusahaan sebagai sarana informasi dan promosi. Footer juga menyediakan menu navigasi tambahan yang terdiri dari Home, Produk, Resep Kami, dan About Us, sehingga pengguna tetap dapat mengakses halaman-halaman utama website dengan mudah dari bagian bawah halaman.

```
//Callable Footer
class callableFooter extends HTMLElement {
  connectedCallback() {
    this.innerHTML = `
    <footer>
      <div class="footer-container">
        <!-- Footer Kiri -->
        <div class="footer-left">
          <div class="footer-column">
            <div class="footer-box">
              <p class="footer-title">PT ADDRESS</p>
              <p class="footer-text"> JALAN MERANTI NO.16/16A, Kec. MedanPetisah, <br>Kota Medan
            </div>
            <div class="footer-box">
              <p class="footer-title">EMAIL</p>
              <p class="footer-text"> kilangkecapangsa.pt@gmail.com</p>
            </div>
          </div>

          <div class="footer-column">
            <div class="footer-box">
              <p class="footer-title">WHATSAPP</p>
              <p class="footer-text">
                <a href="https://wa.me/6281376286940" target="_blank">+62 813-7628-6940</a>
              </p>
            </div>
            <div class="footer-box">
              <p class="footer-title">INSTAGRAM</p>
              <p class="footer-text">
                <a href="https://instagram.com/caphatiangsa" target="_blank">@caphatiangsa</a>
              </p>
            </div>
          </div>
        </div>
      </div>
    `;
  }
}
```

Gambar 3. 9 Callable Footer JS

Gambar 3.9 merupakan kode untuk implementasi custom element dengan nama callableFooter yang digunakan untuk menampilkan footer secara reusable pada seluruh halaman website. Pada metode connectedCallback(), struktur footer dibangun menggunakan HTML yang terdiri dari beberapa bagian. Bagian kiri footer (footer-left) terbagi menjadi dua kolom (footer-column). Kolom pertama menampilkan alamat perusahaan (PT Address) dan email, sedangkan kolom kedua menampilkan WhatsApp dan Instagram dengan tautan yang dapat diklik langsung untuk memudahkan pengguna menghubungi perusahaan atau mengakses media sosial.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```

<!-- Footer Kanan -->
<div class="footer-right">
  <ul>
    <li><a href="Home.html">Home</a></li>
    <li><a href="Produk.html">Produk</a></li>
    <li><a href="ResepKami.html">Resep Kami</a></li>
    <li><a href="AboutUs.html">About Us</a></li>
  </ul>
</div>
</div>
</footer>
`
;
}

```

Gambar 3. 10 Lanjutan Callable Footer JS

Gambar 3.10 menunjukkan bagian kanan footer (footer-right) berisi daftar menu navigasi yang sama dengan menu utama, yaitu Home, Produk, Resep Kami, dan About Us, sehingga pengguna tetap dapat mengakses halaman-halaman utama website dari bagian bawah halaman.

```

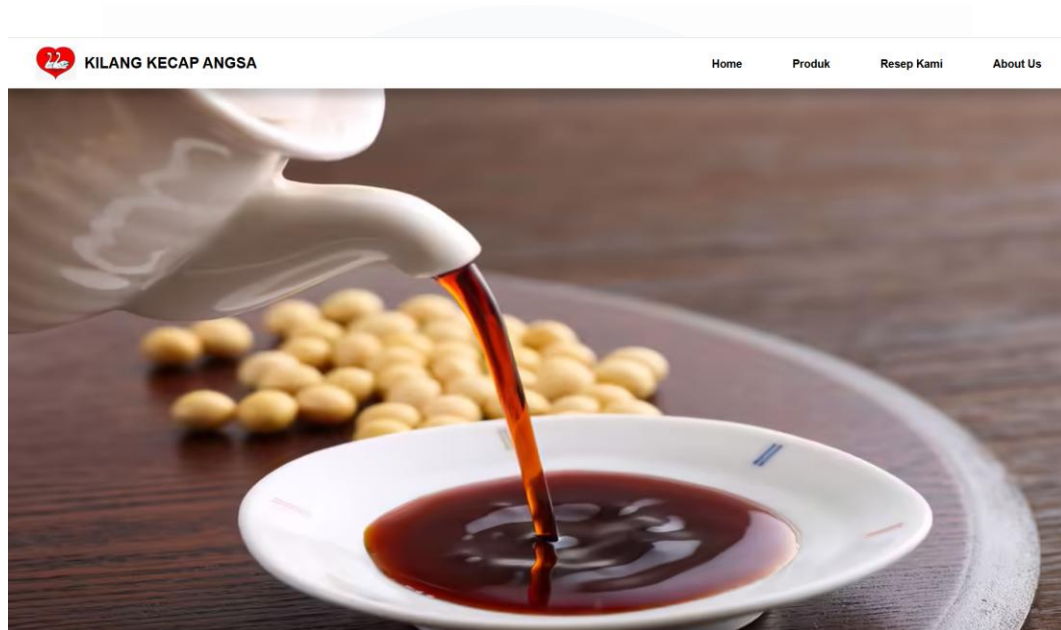
//Custom Name Callable Header
customElements.define("callable-header", callableHeader);
customElements.define("callable-footer", callableFooter);

```

Gambar 3. 11 Custom Element Define Callable Header dan Callable Footer JS

Gambar 3.11 menunjukkan kode yang digunakan untuk mendaftarkan custom element pada browser. Fungsi `customElements.define()` mengaitkan nama tag HTML kustom dengan kelas JavaScript yang telah dibuat sebelumnya, sehingga `<callable-header>` akan merujuk ke kelas `callableHeader` dan `<callable-footer>` akan merujuk ke kelas `callableFooter`. Dengan pendaftaran ini, elemen header dan footer dapat digunakan di seluruh halaman website sebagai komponen yang reusable dan konsisten.

3.2.3 Halaman Home



Gambar 3. 12 Banner Home

Pada Gambar 3.12, terlihat ada pada halaman home terdapat header website sebagai elemen navigasi utama yang telah dijelaskan pada pembahasan sebelumnya. Di bawah header tersebut, terdapat banner utama yang berfungsi sebagai tampilan awal halaman serta memberikan kesan visual yang menarik kepada pengguna sebelum menampilkan konten lainnya.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

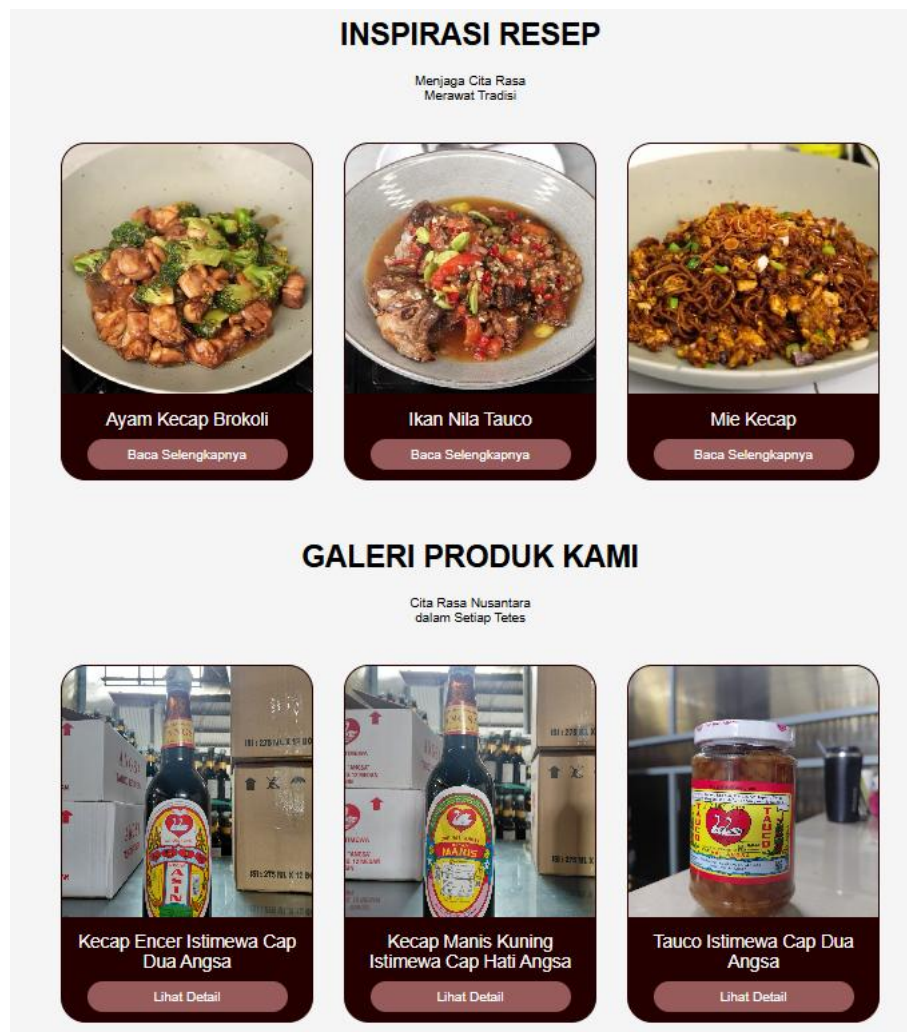
Kami menghadirkan kecap-kecap pilihan, menyatukan cita rasa dan tradisi

Kami adalah Perseroan Terbatas Kilang Kecap Angsa, sebuah perusahaan yang menghadirkan kecap-kecap berkualitas, terinspirasi dari warisan rasa turun-temurun yang tetap relevan hingga kini. Melalui perpaduan tradisi dan inovasi, kami berkomitmen menjaga keaslian rasa sekaligus menghadirkan pengalaman kuliner yang lebih berkesan.



Gambar 3. 13 Penjelasan dan Instagram Home

Setelah banner, pada Gambar 3.13 menampilkan judul “Kami menghadirkan kecap-kecap pilihan, menyatukan cita rasa dan tradisi” yang disertai dengan penjelasan singkat mengenai PT Kilang Kecap Angsa sebagai perusahaan yang menghadirkan produk kecap berkualitas melalui perpaduan tradisi dan inovasi. Pada bagian ini juga ditampilkan video Instagram yang dapat diputar langsung, serta ikon Instagram yang dapat diklik untuk mengarahkan pengguna ke akun Instagram resmi perusahaan.



Gambar 3. 14 Card Home

Setelah bagian Instagram, Gambar 3.14 menampilkan judul “Inspirasi Resep” dengan tagline “Menjaga Cita Rasa, Merawat Tradisi” yang berfungsi sebagai pengantar konten resep. Di bawahnya terdapat beberapa card resep yang menampilkan gambar, nama resep, serta tombol “Baca Selengkapnya” untuk mengarahkan pengguna ke halaman detail resep. Selanjutnya, ditampilkan bagian “Galeri Produk Kami” dengan tagline “Cita Rasa Nusantara, dalam Setiap Tetes”. Pada bagian ini disajikan card produk yang berisi gambar dan nama produk, serta tombol “Lihat Detail” yang mengarahkan pengguna ke halaman detail produk.

```

<body>
  <!-- Callable Header -->
  <callable-header></callable-header>

  <!-- Banner Home -->
  <section id="bannerHome">
    
  </section>

  <!-- Detail Home -->
  <section id="mainHome">
    <div id="detailHome">
      <h1 class="detail-title">
        Kami menghadirkan kecap-kecap pilihan, menyatukan cita rasa dan tradisi
      </h1>
      <p class="detail-desc">
        Kami adalah Perseroan Terbatas Kilang Kecap Angsa, sebuah perusahaan yang menghadirkan
      </p>
    </div>

    <!-- Video IG Home -->
    <div id="videoHome">
      <div class="ig-shadow">
        <blockquote class="instagram-media" data-instgrm-permalink="https://www.instagram.co
      </blockquote>
      </div>
    </div>
  </section>

```

Gambar 3. 15 Home Body HTML

Gambar 3.15 menunjukkan kode untuk Bagian <body> pada halaman Home dimulai dengan pemanggilan custom element <callable-header> yang menampilkan header website secara konsisten di seluruh halaman. Setelah header, ditampilkan banner utama pada elemen <section id="bannerHome"> sebagai tampilan visual pembuka halaman. Selanjutnya, pada <section id="mainHome"> terdapat bagian detail Home yang menampilkan judul “Kami menghadirkan kecap-kecap pilihan, menyatukan cita rasa dan tradisi” beserta deskripsi singkat mengenai PT Kilang Kecap Angsa, menekankan kualitas produk serta perpaduan tradisi dan inovasi. Di bawahnya, terdapat video Instagram yang dapat diputar langsung melalui website, dibungkus dalam <div class="ig-shadow"> untuk efek visual, sehingga pengguna dapat melihat konten media sosial perusahaan secara interaktif.

```

<!-- Card Resep Home -->
<div id="resepHome">
  <h1 class="title">INSPIRASI RESEP</h1>
  <p class="tagline"> Menjaga Cita Rasa <br> Merawat Tradisi </p>
</div>
<div class="card-container" id="resep-container">
  <!-- Informasi Card dari Supabase -->
</div>

<!-- Card Produk Home -->
<div id="galeriHome">
  <h1 class="title">GALERI PRODUK KAMI</h1>
  <p class="tagline"> Cita Rasa Nusantara <br> dalam Setiap Tetes </p>
</div>
<div class="card-container" id="produk-container">
  <!-- Informasi Card dari JS Supabase -->
</div>
</section>

<!-- Callable Footer -->
<callable-footer></callable-footer>

<!-- Pemanggilan Script -->
<script async src="https://www.instagram.com/embed.js"></script>
<script src="./headerFooterManager.js"></script>
<script src="./Home.js"></script>

```

Gambar 3. 16 Lanjutan Home Body HTML

Bagian selanjutnya pada halaman Home, Gambar 3.16 menampilkan card resep dan card produk. Pada `<div id="resepHome">`, terdapat judul “INSPIRASI RESEP” dengan tagline “Menjaga Cita Rasa, Merawat Tradisi”, diikuti `<div class="card-container" id="resep-container">` yang nantinya akan menampilkan card resep secara dinamis melalui data dari Supabase. Selanjutnya, `<div id="galeriHome">` menampilkan judul “GALERI PRODUK KAMI” dengan tagline “Cita Rasa Nusantara, dalam Setiap Tetes”, dan `<div class="card-container" id="produk-container">` menampilkan card produk secara dinamis, juga diambil dari Supabase melalui JavaScript. Di bagian paling bawah halaman, `<callable-footer>` memanggil footer yang konsisten di seluruh halaman. Selain itu, beberapa script dipanggil untuk mendukung interaktivitas, yaitu library Instagram untuk menampilkan media Instagram, `headerFooterManager.js` untuk pengelolaan header dan footer, serta `Home.js` untuk memuat data card produk dan resep secara dinamis.

MULTIMEDIA
NUSANTARA

```
const supabaseUrl = "https://wqsicwkmuwfb1scsbig.supabase.co";  
const supabaseAnonKey = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6I6Indxc2I  
const _supabase = supabase.createClient(supabaseUrl, supabaseAnonKey);
```

Gambar 3. 17 Inisialisasi Koneksi Supabase pada Home JS

Gambar 3.17 menunjukkan kode yang digunakan untuk menghubungkan website dengan layanan Supabase. Variabel `supabaseUrl` menyimpan alamat proyek Supabase, sedangkan `supabaseAnonKey` adalah kunci akses publik (anon key) yang digunakan untuk autentikasi. Fungsi `supabase.createClient(supabaseUrl, supabaseAnonKey)` membuat instansi Supabase (`_supabase`) yang memungkinkan website mengakses database dan layanan backend secara langsung, termasuk membaca dan menampilkan data resep maupun produk secara dinamis.

```

//Function fetchHomeContent
async function fetchHomeContent() {
  const { data: resepData, error: resepError } = await _supabase
    .from('resep')
    .select('id_resep, nama_resep, image_url')
    .in('id_resep', [1, 2, 3]);

  if (!resepError && resepData) {
    const resepContainer = document.getElementById('resep-container');
    resepContainer.innerHTML = '';

    resepData.forEach(item => {
      const img = item.image_url ? item.image_url : 'assets/noPicture.jpg';
      resepContainer.innerHTML += `
        <div class="card">
          <div class="card-image">
            
          </div>
          <div class="card-body">
            <p class="card-title">${item.nama_resep}</p>
            <button class="card-btn" onclick="window.location.href='DetailResep.html?id=${item.id_resep}'>Baca Selengkapnya</button>
          </div>
        </div>
      `;
    });
  }
}

```

Gambar 3. 18 Pengambilan dan Penampilan Data Resep Home JS

Gambar 3.18 menunjukkan kode yang merupakan fungsi asinkron `fetchHomeContent()` yang digunakan untuk mengambil data resep dari database Supabase dan menampilkannya di halaman Home secara dinamis. Fungsi `_supabase.from('resep').select(...)` mengambil kolom `id_resep`, `nama_resep`, dan `image_url` dari tabel `resep`, khusus untuk resep dengan `id_resep` 1, 2, dan 3. Jika tidak terjadi error (`resepError`), data akan diiterasi dan ditambahkan ke dalam elemen `<div>` dengan id `resep-container`. Setiap card menampilkan gambar resep (atau gambar default jika `image_url` null), nama resep, dan tombol “Baca Selengkapnya” yang mengarahkan pengguna ke halaman detail resep terkait. Cara yang sama juga dilakukan pada tampilan produk.

```

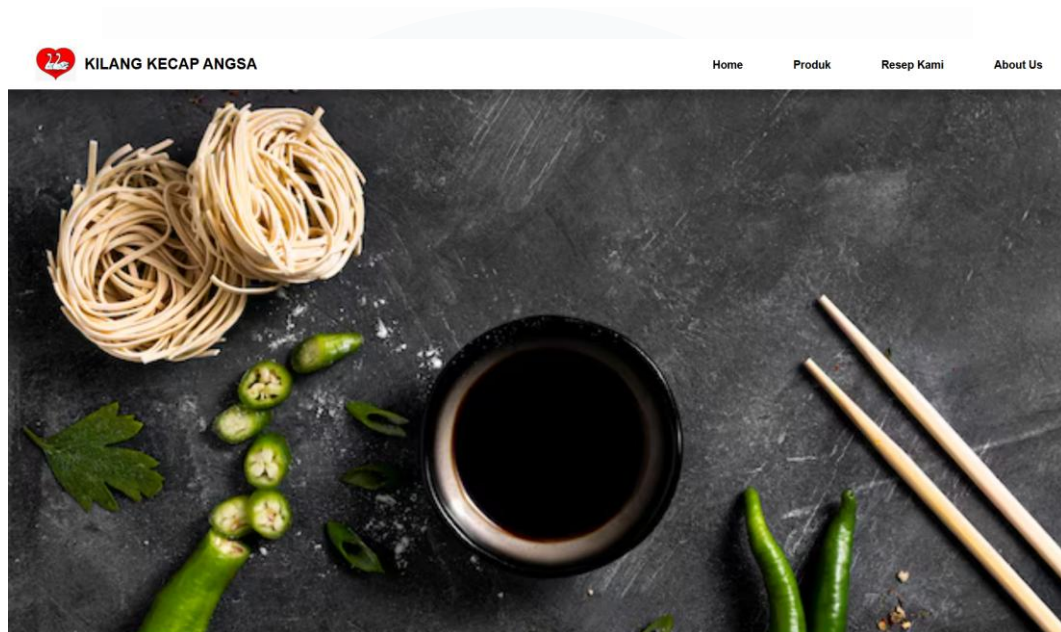
//Memanggil fetchHomeContent
document.addEventListener('DOMContentLoaded', fetchHomeContent);

```

Gambar 3. 19 Insialisasi fetchHomeContent Home JS

Gambar 3.19 menunjukkan kode yang menggunakan event listener `DOMContentLoaded` untuk memanggil fungsi `fetchHomeContent()` segera setelah seluruh konten HTML pada halaman Home selesai dimuat. Dengan demikian, data resep dari Supabase akan diambil dan ditampilkan secara dinamis pada halaman tanpa menunggu elemen-elemen lain selesai dimuat, memastikan card resep muncul dengan benar saat pengguna membuka halaman.

3.2.4 Halaman About Us



Gambar 3. 20 Banner About Us

Gambar 3.20 menampilkan banner sebagai elemen visual yang berfungsi untuk mempercantik tampilan halaman. Banner ini digunakan sebagai pendukung estetika sebelum pengguna mengakses konten utama pada halaman About Us.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



TENTANG KAMI

PT. Kilang Kecap Angsa merupakan perusahaan yang bergerak di bidang industri pangan, khususnya produksi kecap berkualitas. Perusahaan ini berdiri sejak tahun 1955 dan berkembang dengan komitmen untuk menghadirkan produk kecap yang bercita rasa khas, aman dikonsumsi, dan konsisten mutunya. Dengan pengalaman bertahun-tahun, Kecap Angsa telah menjadi salah satu merek yang dipercaya oleh masyarakat.

Gambar 3. 21 Penjelasan About Us

Di bawah banner About Us, Gambar 3.21 terdapat judul “Tentang Kami” yang berisi penjelasan mengenai PT Kilang Kecap Angsa sebagai perusahaan yang bergerak di bidang industri pangan, khususnya produksi kecap berkualitas. Pada bagian ini dijelaskan bahwa perusahaan telah berdiri sejak tahun 1955 dan berkembang dengan komitmen untuk menghadirkan produk yang bercita rasa khas, aman dikonsumsi, serta memiliki mutu yang konsisten. Informasi ini bertujuan untuk memberikan gambaran singkat mengenai latar belakang dan pengalaman perusahaan.

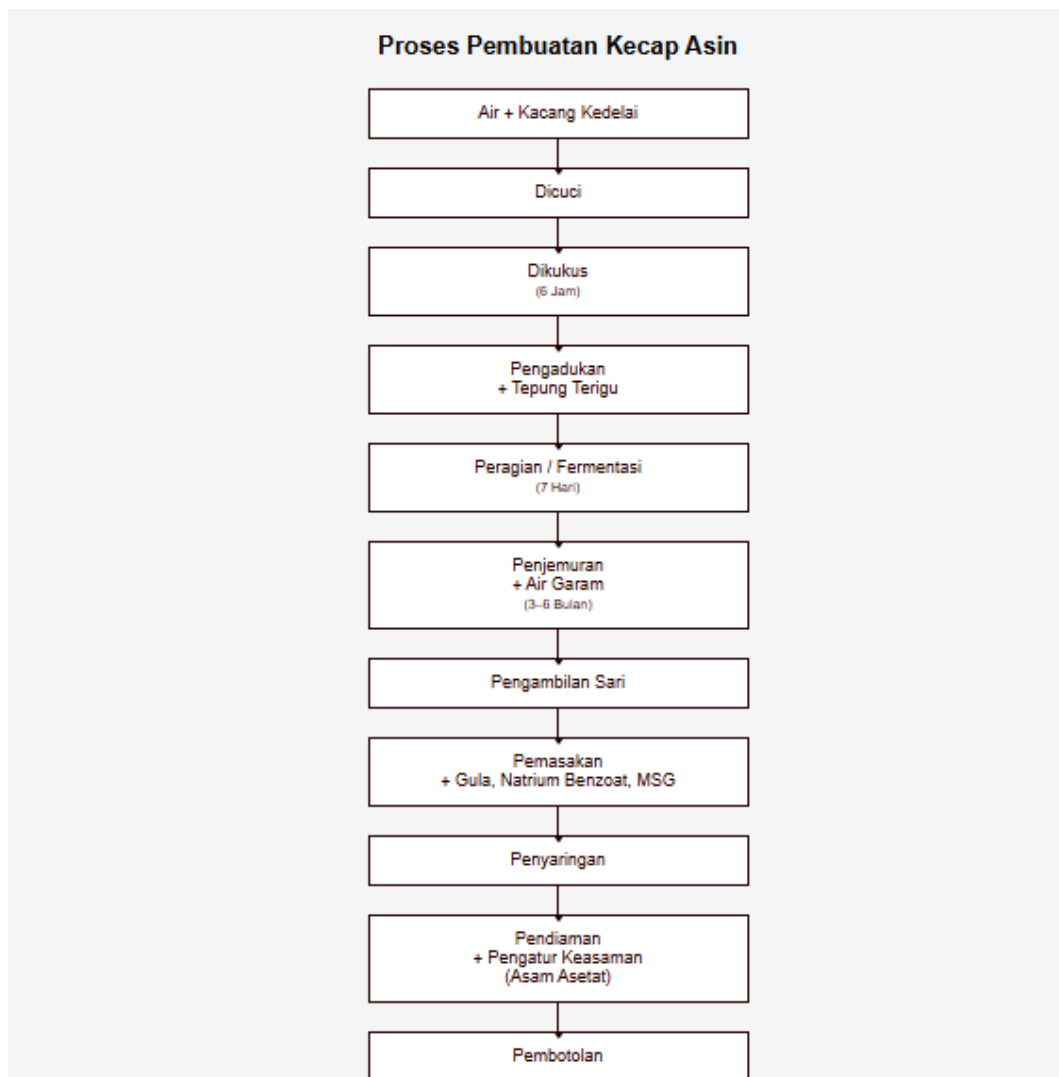
PROSES PEMBUATAN KECAP KAMI

Proses Pembuatan Kecap Manis



Gambar 3. 22 Flowchart About Us Pembuatan Kecap Manis

Di bawah penjelasan tentang kami terdapat Gambar 3.22 yang berisikan penjelasan proses pembuatan kecap manis. Flowchart ini menunjukkan pembuatan kecap manis yang dimulai dari air dan kacang kedelai yang dicuci, dikukus, diaduk dengan tepung terigu, difermentasi hingga pembotolan.



Gambar 3. 23 Flowchart About Us Pembuatan Kecap Asin

Setelah flowchart proses pembuatan kecap manis, Gambar 3.23 menunjukkan flowchart proses pembuatan kecap asin. Flowchart ini berfungsi untuk menggambarkan tahapan produksi kecap asin secara berurutan, sehingga pengguna dapat memahami perbedaan alur proses antara pembuatan kecap manis dan kecap asin.

```

<!-- Banner About Us -->
<Section id = "bannerAboutUs">
  
</Section>

<!-- Detail About Us -->
<Section id = "mainAboutUs">
  <div id = "detailAboutUs">
    <h1 class="detail-title">
      TENTANG KAMI
    </h1>
    <p class="detail-desc">
      PT. Kilang Kecap Angsa merupakan perusahaan yang bergerak di bidang industri
    </p>
  </div>

```

Gambar 3. 24 Banner dan Detail About Us HTML

Gambar 3.24 menunjukkan potongan kode yang menampilkan banner pada halaman About Us menggunakan elemen `<section id="bannerAboutUs">`, yang berfungsi sebagai elemen visual untuk mempercantik tampilan halaman. Di bawah banner, terdapat bagian detail About Us pada `<section id="mainAboutUs">` yang menampilkan judul “TENTANG KAMI” dan deskripsi singkat mengenai PT Kilang Kecap Angsa, menjelaskan latar belakang perusahaan, bidang usaha, sejarah berdiri sejak 1955, serta komitmen dalam menghadirkan produk kecap berkualitas dan terpercaya.

```

<!-- Proses About Us -->
<div id="prosesAboutUs">
<h1 class="detail-title">PROSES PEMBUATAN KECAP KAMI</h1>
<!-- Proses Pembuatan Kecap Manis -->
<h2 class="proses-title">Proses Pembuatan Kecap Manis</h2>
<div class="flowchart">
  <div class="box">Air + Kacang Kedelai</div>
  <div class="arrow"></div>

  <div class="box">Dicuci</div>
  <div class="arrow"></div>

  <div class="box">Dikukus<br><small>(6 Jam)</small></div>
  <div class="arrow"></div>

  <div class="box">Pengadukan<br>+ Tepung Terigu</div>

```

Gambar 3. 25 Flowchart Pembuatan Kecap Manis About Us HTML

Gambar 3.25 menunjukkan potongan kode yang menampilkan flowchart proses pembuatan kecap manis pada halaman About Us. Bagian ini dimulai dengan judul “PROSES PEMBUATAN KECAP KAMI” dan subjudul “Proses Pembuatan Kecap Manis”. Flowchart dibuat menggunakan `<div class="flowchart">` yang terdiri dari beberapa box untuk setiap tahap proses, seperti Air + Kacang Kedelai, Dicuci, dan Dikukus (6 Jam), serta arrow sebagai penanda alur antar-tahapan. Tampilan ini memudahkan pengguna untuk memahami urutan proses produksi secara visual dan sistematis.

```

<!-- Proses Pembuatan Kecap Asin -->
<h2 class="proses-title">Proses Pembuatan Kecap Asin</h2>
<div class="flowchart">
  <div class="box">Air + Kacang Kedelai</div>
  <div class="arrow"></div>

  <div class="box">Dicuci</div>
  <div class="arrow"></div>

  <div class="box">Dikukus<br><small>(6 Jam)</small></div>
  <div class="arrow"></div>

  <div class="box">Pengadukan<br>+ Tepung Terigu</div>
  <div class="arrow"></div>

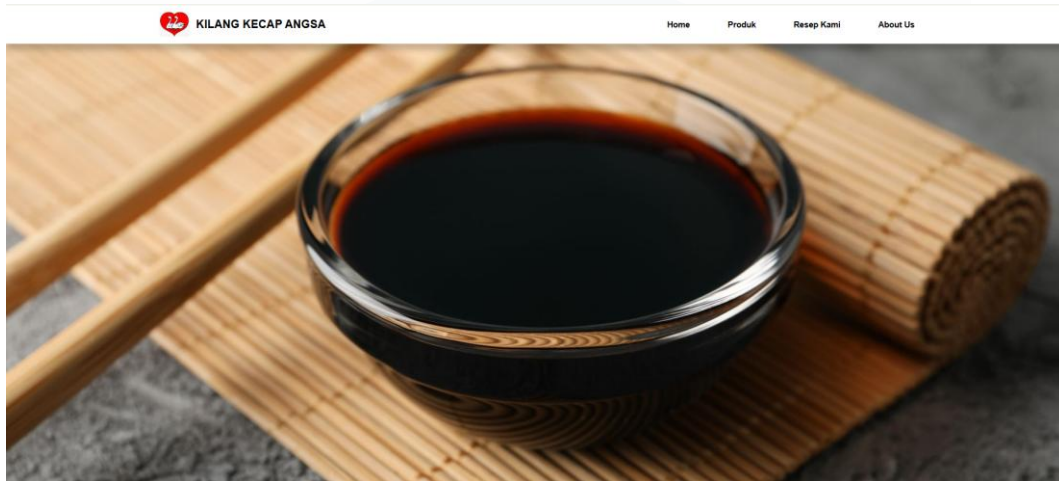
  <div class="box">Peragian / Fermentasi<br><small>(7 Hari)</small></div>
  <div class="arrow"></div>

```

Gambar 3. 26 Flowchart Pembuatan Kecap Asin About Us HTML

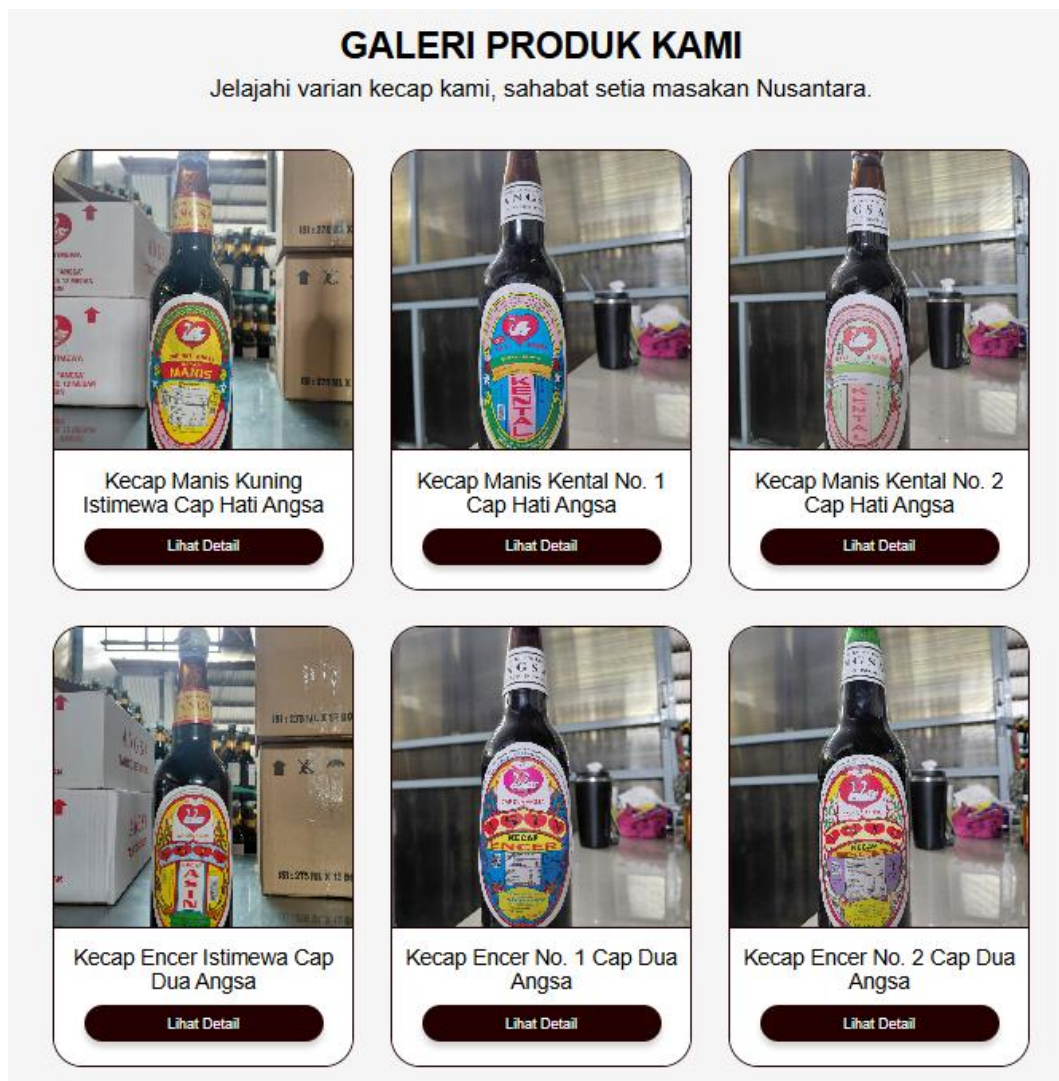
Pada gambar 3.26 sama seperti gambar 3.30 dimana disini terdapat penjelasan flowchart mengenai proses pembuatan kecap asin.

3.2.5 Halaman Produk dan Detail Produk



Gambar 3. 27 Banner Produk

Gambar 3.27 menampilkan banner sebagai elemen visual pembuka halaman. Banner ini berfungsi untuk mempercantik tampilan halaman serta memberikan pemisah visual sebelum pengguna melihat daftar produk yang tersedia.



Gambar 3. 28 Card Produk

Setelah banner pada halaman Produk, Gambar 3.28 menampilkan judul “Galeri Produk Kami” dengan tagline “Jelajahi varian kecap kami, sahabat setia masakan Nusantara”. Di bawah bagian tersebut disajikan card produk yang menampilkan gambar, nama produk, serta tombol “Lihat Detail” untuk mengarahkan pengguna ke halaman detail produk. Berbeda dengan halaman Home yang hanya menampilkan sebagian produk, pada halaman Produk seluruh produk yang tersedia ditampilkan secara lengkap.

```

<!-- Banner Produk -->
<Section id = "bannerProduk">
  
</Section>

<!-- Detail Produk -->
<div id = "detailProduk">
  <h1 class="detail-title">
    GALERI PRODUK KAMI
  </h1>
  <p class="detail-desc">
    Jelajahi varian kecap kami, sahabat setia masakan Nusantara.
  </p>
</div>

<!-- Card Produk -->
<div id="produkContainer">
  <div class="card-container" id="produkCards"></div>
</div>

```

Gambar 3. 29 Banner, Detail, dan Card Produk HTML

Gambar 3.29 menunjukkan potongan kode yang menampilkan banner halaman Produk menggunakan `<section id="bannerProduk">` sebagai elemen visual pembuka halaman. Di bawah banner, terdapat bagian detail Produk (`<div id="detailProduk">`) yang menampilkan judul “GALERI PRODUK KAMI” dan deskripsi singkat “Jelajahi varian kecap kami, sahabat setia masakan Nusantara” sebagai pengantar galeri produk. Selanjutnya, `<div id="produkContainer">` dengan `<div class="card-container" id="produkCards">` digunakan untuk menampilkan card produk secara dinamis, yang akan memuat gambar, nama produk, dan tombol “Lihat Detail”, memungkinkan pengguna untuk menjelajahi seluruh produk yang tersedia di halaman ini.

```
//Function loadProduk
async function loadProduk() {
    const { data, error } = await supabase
        .from("produk")
        .select("*")
        .order("id", { ascending: true });

    if (error) {
        console.error(error);
        return;
    }

    renderCards(data);
}
```

Gambar 3. 30 Pengambilan dan Pengurutan Data Produk JS

Gambar 3.30 menunjukkan kode yang merupakan fungsi asinkron `loadProduk()` yang digunakan untuk mengambil data produk dari tabel produk di database Supabase. Data diambil menggunakan metode `.select("*")` untuk mengambil seluruh kolom, dan diurutkan berdasarkan kolom id secara ascending (`.order("id", { ascending: true })`). Jika terjadi error saat pengambilan data, error akan dicetak ke konsol dan proses berhenti. Jika data berhasil diambil, fungsi `renderCards(data)` dipanggil untuk menampilkan card produk secara dinamis pada halaman Produk, memungkinkan pengguna melihat semua produk yang tersedia dengan urutan yang konsisten.

```

//Function renderCards
function renderCards(items) {
  const container = document.getElementById("produkCards");
  container.innerHTML = "";

  items.forEach(item => {
    const card = document.createElement("div");
    card.classList.add("card");

    card.innerHTML = `
      <div class="card-image">
        
      </div>
      <div class="card-body">
        <p class="card-title">${item.nama_produk}</p>
        <button class="card-btn" data-id="${item.id_produk}">Lihat Detail</button>
      </div>
    `;

    card.querySelector(".card-btn").addEventListener("click", () => {
      window.location.href = `DetailProduk.html?id=${item.id_produk}`;
    });

    container.appendChild(card);
  });
}

//Memanggil loadProduk
loadProduk();

```

Gambar 3. 31 renderCards Produk JS

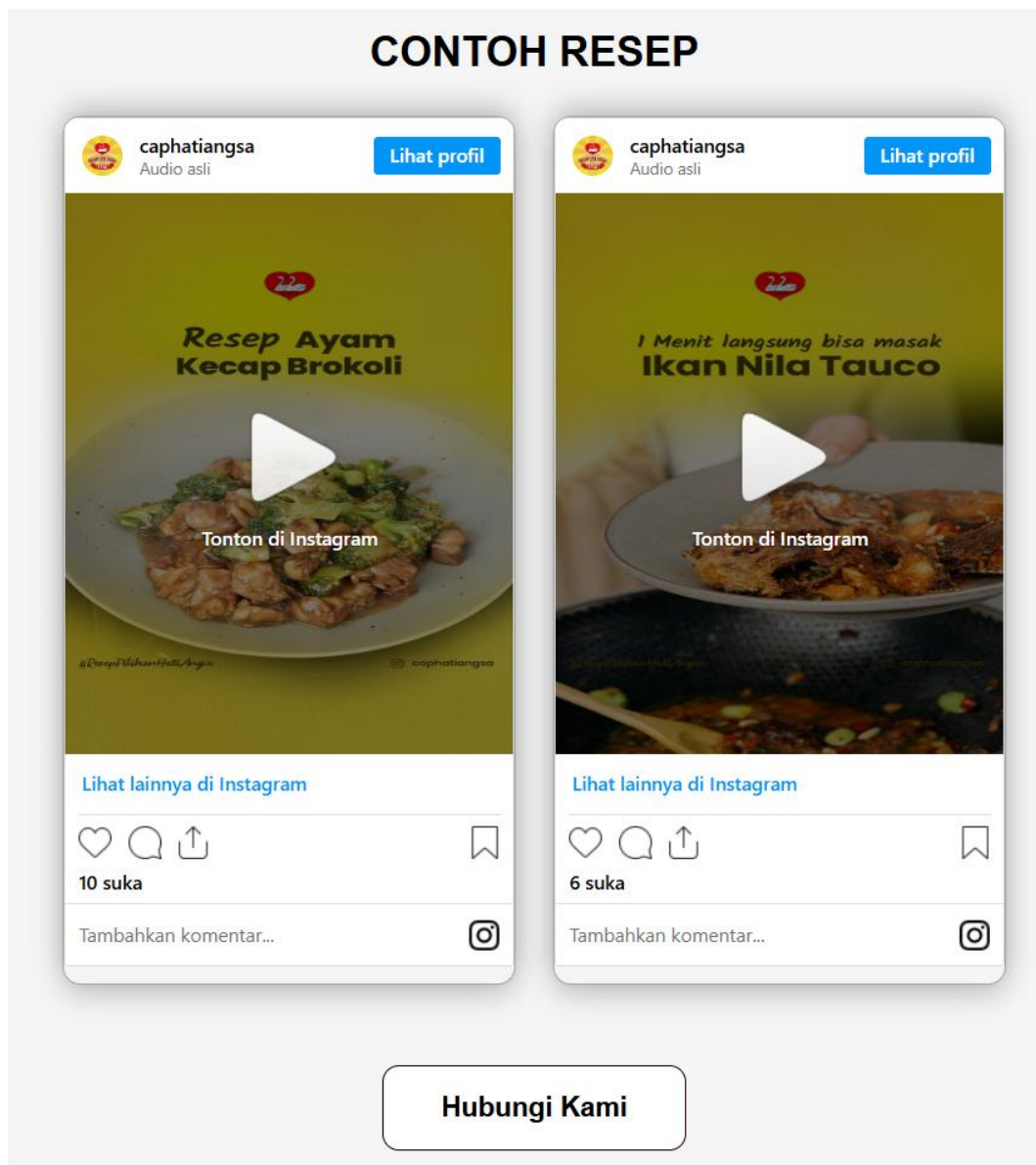
Gambar 3.31 menunjukkan kode yang merupakan fungsi `renderCards(items)` yang digunakan untuk menampilkan card produk secara dinamis pada halaman Produk. Fungsi mengambil parameter `items` berupa array data produk dari Supabase, lalu membersihkan kontainer card (`produkCards`) sebelum menambahkan elemen baru. Setiap item dibuat menjadi `<div class="card">` yang berisi gambar produk (atau gambar default jika `image_url` null), nama produk, dan tombol “Lihat Detail”. Tombol tersebut diberikan event listener sehingga ketika diklik, pengguna diarahkan ke halaman `DetailProduk.html` sesuai id produk yang bersangkutan. Fungsi ini memastikan seluruh produk ditampilkan secara interaktif dan konsisten. Di akhir kode, `loadProduk()` dipanggil untuk mengambil data dari Supabase dan memicu fungsi render, sehingga halaman Produk langsung menampilkan semua card produk secara dinamis saat dibuka.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3. 32 Tampilan Detail Produk

Pada halaman Detail Produk, Gambar 3.32 menampilkan gambar produk yang diletakkan di bagian tengah halaman sebagai fokus utama tampilan. Di bawah gambar tersebut terdapat nama produk yang diikuti dengan penjelasan singkat mengenai produk.



Gambar 3. 33 Contoh Resep pada Halaman Detail Produk

Gambar 3.33 merupakan gambar yang menampilkan 1-2 resep video Instagram yang berhubungan dengan produk yang berkaitan. Tujuan adanya video ini yaitu untuk menarik pelanggan membeli produk karena tertarik dengan resep video Instagram tersebut. Selain itu, tersedia tombol “Hubungi Kami” yang terhubung langsung ke WhatsApp untuk memudahkan pengguna menghubungi pihak perusahaan apabila tertarik melakukan pembelian dalam jumlah besar (bulking). Pada bagian paling bawah halaman, terdapat page divider yang berfungsi sebagai pemisah visual antarbagian halaman.

Coba Juga Produk Berikut



Gambar 3. 34 Coba juga Produk Lainnya pada Halaman Detail Produk

Setelah page divider, Gambar 3.34 menampilkan tulisan “Coba Juga Produk Berikut” yang berfungsi sebagai rekomendasi produk lainnya. Di bawah bagian tersebut terdapat dua card produk yang menampilkan gambar dan nama produk, serta tombol “Baca Selengkapnya” untuk mengarahkan pengguna ke halaman detail produk terkait.

```

<!-- Banner Detail Produk -->
<section id="bannerDetailProduk">
  
</section>

<!-- Detail Produk -->
<section id="mainDetailProduk">
  <div id="detailDetailProduk">
    <h1 class="detail-title">Nama Produk</h1>
    <p class="detail-desc">Deskripsi produk</p>
  </div>
</section>

<!-- Tombol Kontak -->
<section id="contactButton">
  <a href="https://wa.me/6281376286940" target="_blank" class="contact-btn">
    Hubungi Kami
  </a>
</section>

<!-- Page Divider -->
<section id="pageDivider"></section>

<!-- Produk Terkait -->
<section id="relatedProduk">
  <h2 class="related-title">Coba Juga Produk Berikut</h2>

  <div class="related-container">
    <div class="card-container" id="relatedCards">
      <!-- Informasi Card Dari JS Supabase -->
    </div>
  </div>
</section>

```

Gambar 3. 35 Banner, Detail, dan Produk Terkait Detail Produk HTML

Gambar 3.35 menunjukkan potongan kode yang menampilkan struktur halaman Detail Produk.

- <section id="bannerDetailProduk"> menampilkan banner produk di bagian atas sebagai fokus visual.
- <section id="mainDetailProduk"> berisi nama produk dan deskripsi singkat untuk memberikan informasi utama mengenai produk.
- <section id="contactButton"> menyediakan tombol “Hubungi Kami” yang terhubung langsung ke WhatsApp, memudahkan pengguna untuk melakukan pembelian dalam jumlah besar (bulk).
- <section id="pageDivider"> berfungsi sebagai pemisah visual antarbagian halaman.
- <section id="relatedProduk"> menampilkan rekomendasi produk lain dengan judul “Coba Juga Produk Berikut”, dan <div class="card-container">

id="relatedCards"> digunakan untuk menampilkan card produk terkait secara dinamis melalui data Supabase.

```
//Variabel dan Memanggil Supabase
const supabaseUrl = "https://wqsicwkmuwfblscsbig.supabase.co";
const supabaseAnonKey = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Indxc2";
const supabase = createClient(supabaseUrl, supabaseAnonKey);
const urlParams = new URLSearchParams(window.location.search);
const produkId = urlParams.get("id");
```

Gambar 3. 36 Inisialisasi Supabase dan Pengambilan Id dari URL Detail Produk JS

Gambar 3.36 terdapat Inisialisasi supabase yang melakukan pengambilan Id produk dari URL. URLSearchParams digunakan untuk membaca parameter query pada URL. Lalu, variabel produkId menyimpan nilai dari parameter id, sehingga halaman Detail Produk dapat menampilkan data spesifik berdasarkan produk yang diklik di halaman Produk atau Home.

```

//Function loadDetailProduk
async function loadDetailProduk() {
  if (!produkId) return;

  const { data, error } = await supabase
    .from("produk")
    .select("*")
    .eq("id_produk", produkId)
    .single();

  if (error) {
    console.error(error);
    return;
  }

  renderDetail(data);
  renderVideoProduk(data.video_url);
  loadRelatedProduk(data.id_produk);
}

```

Gambar 3. 37 Mengambil Data Produk Spesifik Detail Produk JS

Gambar 3.37 menunjukkan kode yang merupakan fungsi asinkron `loadDetailProduk()` yang digunakan untuk mengambil data produk tertentu berdasarkan `produkId` dari Supabase. Pertama, fungsi memeriksa apakah `produkId` tersedia; jika tidak, fungsi berhenti. Fungsi `_supabase.from("produk").select("*").eq("id_produk", produkId).single()` mengambil seluruh kolom dari tabel produk untuk produk dengan ID yang sesuai, dan mengembalikan satu objek data. Jika terjadi error, error dicetak ke konsol dan proses berhenti. Jika data berhasil diambil, fungsi `renderDetail(data)` dipanggil untuk menampilkan detail produk di halaman, `renderVideoProduk(data.video_url)` dipanggil untuk menampilkan video Instagram dan `loadRelatedProduk(data.id_produk)` dipanggil untuk menampilkan produk terkait. Dengan cara ini, halaman Detail Produk dapat menampilkan informasi produk yang spesifik dan menambahkan rekomendasi produk terkait secara dinamis.

```

//Function renderDetail
function renderDetail(item) {
    document.querySelector("#bannerDetailProduk .banner").src =
        item.image_url || "assets/noPicture.jpg";

    document.querySelector(".detail-title").textContent = item.nama_produk;
    document.querySelector(".detail-desc").textContent =
        item.deskripsi_produk || "Tidak ada deskripsi.";
}

//Function loadRelatedProduk
async function loadRelatedProduk(currentId) {
    const { data, error } = await supabase
        .from("produk")
        .select("id_produk, nama_produk, image_url")
        .order("id_produk", { ascending: true });

    if (error) return console.error(error);

    const index = data.findIndex(item => item.id_produk === currentId);
    let related = [];

    if (index === 0) {
        related = data.slice(1, 3);
    } else if (index === data.length - 1) {
        related = data.slice(index - 2, index);
    } else {
        related = [data[index - 1], data[index + 1]];
    }

    renderRelatedCards(related);
}

```

Gambar 3. 38 renderDetail dan loadRelatedProduk Detail Produk JS

Gambar 3.38 menunjukkan potongan kode yang terdiri dari dua fungsi utama pada halaman Detail Produk:

- **renderDetail(item)**
Fungsi ini menampilkan informasi produk spesifik pada halaman. Gambar banner produk diatur pada elemen `#bannerDetailProduk .banner` menggunakan `item.image_url`, dan akan menggunakan gambar default jika `image_url` kosong. Nama produk ditampilkan pada elemen `.detail-title`, sedangkan deskripsi produk ditampilkan pada `.detail-desc`, dengan fallback teks “Tidak ada deskripsi.” jika deskripsi kosong.

- `loadRelatedProduk(currentId)`
Fungsi ini mengambil semua data produk dari Supabase dan menentukan produk yang terkait berdasarkan posisi produk saat ini (`currentId`). Produk terkait dipilih dua produk sebelum dan/atau setelah produk saat ini untuk ditampilkan sebagai rekomendasi. Fungsi `renderRelatedCards(related)` dipanggil untuk menampilkan card produk terkait secara dinamis di halaman.

```
//Function renderVideoProduk
function renderVideoProduk(videoUrls) {
  const container = document.getElementById("videoProdukContainer");
  container.innerHTML = "";

  if (!videoUrls || videoUrls.length === 0) return;

  videoUrls.slice(0, 3).forEach(url => {
    container.innerHTML += `
      <div class="ig-wrapper">
        <blockquote class="instagram-media"
          data-instgrm-permalink="${url}"
          data-instgrm-version="14">
        </blockquote>
      </div>
    `;
  });

  if (window.instgrm) {
    window.instgrm.Embeds.process();
  }
}
```

Gambar 3. 39 *renderVideoProduk Detail Produk JS*

Gambar 3.39 menunjukkan fungsi `renderVideoProduk` digunakan untuk menampilkan video produk dari Instagram ke dalam halaman website secara dinamis. Fungsi ini menerima parameter berupa kumpulan URL video Instagram, kemudian menampilkan maksimal tiga video ke dalam elemen dengan ID `videoProdukContainer`. Sebelum menampilkan data, kontainer akan dikosongkan terlebih dahulu untuk menghindari duplikasi tampilan. Setiap URL diproses dengan membuat elemen embed Instagram menggunakan tag `blockquote` yang sesuai dengan standar Instagram. Setelah seluruh elemen berhasil ditambahkan ke halaman, fungsi `Embeds.process()` dipanggil untuk memastikan konten Instagram dapat dirender dan ditampilkan dengan benar pada halaman website.

```

//Function renderRelatedCards
function renderRelatedCards(items) {
  const container = document.getElementById("relatedCards");
  container.innerHTML = "";

  items.forEach(item => {
    const card = document.createElement("div");
    card.className = "card";

    card.innerHTML = `
      <div class="card-image">
        
      </div>
      <div class="card-body">
        <p class="card-title">${item.nama_produk}</p>
        <button class="card-btn">Baca Selengkapnya</button>
      </div>
    `;

    card.querySelector(".card-btn").onclick = () => {
      window.location.href = `DetailProduk.html?id=${item.id_produk}`;
    };

    container.appendChild(card);
  });
}

//Memanggil loadDetailProduk
loadDetailProduk();

```

Gambar 3. 40 renderRelatedCards Detail Produk JS

Gambar 3.40 menunjukkan Kode yang merupakan fungsi dari `renderRelatedCards(items)` yang digunakan untuk menampilkan card produk terkait secara dinamis di halaman Detail Produk. Fungsi membersihkan kontainer `#relatedCards` sebelum menambahkan card baru. Setiap item dari array `items` dibuat menjadi `<div class="card">` yang berisi gambar produk (atau gambar default jika `image_url` null), nama produk, dan tombol “Baca Selengkapnya”. Tombol diberi event handler yang mengarahkan pengguna ke halaman detail produk yang sesuai berdasarkan id. Dengan cara ini, pengguna dapat menelusuri produk terkait tanpa perlu navigasi tambahan, meningkatkan interaktivitas dan pengalaman pengguna. Di akhir kode, `loadDetailProduk()` dipanggil untuk memulai proses pengambilan data produk utama dan menampilkan produk terkait di halaman.

3.2.6 Halaman Resep dan Detail Resep



Gambar 3. 41 Banner Resep

Pada halaman Resep, Gambar 3.41 menampilkan banner makanan sebagai elemen visual pembuka halaman. Banner ini berfungsi untuk mempercantik tampilan halaman serta memberikan gambaran awal sebelum pengguna melihat daftar resep yang tersedia.



Gambar 3. 42 Card Resep

Setelah banner pada halaman Resep, Gambar 3.42 menampilkan judul “Jelajahi Resep Andalan Kami”. Di bawah judul tersebut terdapat fitur filter yang diletakkan di sisi kanan halaman untuk memudahkan pengguna dalam menyaring resep berdasarkan kategori tertentu. Selanjutnya, ditampilkan card resep yang berisi gambar, nama resep, serta tombol “Baca Selengkapnya” yang mengarahkan pengguna ke halaman detail resep.

```

<!-- Banner Resep -->
<section id="bannerResep">
  
</section>

<!-- Judul dan Filter Resep-->
<section id="resepHeader">
  <h1 class="resep-title">JELAJAHI RESEP ANDALAN KAMI</h1>

  <div class="filter-wrapper">
    <label for="filterResep">Filter berdasarkan</label>
    <select id="filterResep">
      <option value="All">All</option>
      <option value="ayam">Ayam</option>
      <option value="telur">Telur</option>
      <option value="mie">Mie</option>
      <option value="ikan">Ikan</option>
    </select>
  </div>
</section>

```

Gambar 3. 43 Banner dan Filter Resep HTML

Gambar 3.43 menunjukkan potongan kode yang menampilkan banner di bagian atas halaman Resep menggunakan `<section id="bannerResep">`, berfungsi sebagai elemen visual pembuka halaman. Di bawah banner, terdapat judul halaman “JELAJAHI RESEP ANDALAN KAMI” dan filter resep yang memungkinkan pengguna memilih kategori resep tertentu melalui dropdown `<select id="filterResep">`. Opsi filter mencakup berbagai jenis bahan dan masakan seperti Ayam, telur, Mie, dan ikan serta opsi All untuk menampilkan semua resep. Fitur ini memudahkan pengguna menelusuri resep sesuai kategori yang diinginkan.

```
<!-- Card Resep -->
<div id="resepContainer">
|   <div class="card-container" id="resepCards"></div>
</div>
```

Gambar 3. 44 Card Resep HTML

Gambar 3.44 menunjukkan potongan kode yang menyiapkan container untuk menampilkan card resep secara dinamis pada halaman Resep. `<div class="card-container" id="resepCards">` berfungsi sebagai tempat menampung card resep yang akan dibuat melalui JavaScript, yang berisi gambar resep, nama resep, dan tombol “Baca Selengkapnya”.

```

//Filter Resep
document.getElementById("filterResep").addEventListener("change", (e) => {
    const value = e.target.value;

    if (value === "All") {
        renderResep(allResep);
    } else {
        const filtered = allResep.filter(
            resep =>
                resep.filter_resep &&
                resep.filter_resep.toLowerCase() === value.toLowerCase()
        );

        renderResep(filtered);
    }
});

```

Gambar 3. 45 Filter Resep JS

Gambar 3.45 menunjukkan kode yang menambahkan event listener pada dropdown filter resep (#filterResep) untuk menangani perubahan pilihan kategori oleh pengguna. Ketika nilai dropdown berubah (change), variabel value menyimpan kategori yang dipilih. Jika pengguna memilih All, fungsi renderResep(allResep) dipanggil untuk menampilkan semua resep. Jika kategori tertentu dipilih, array allResep difilter berdasarkan properti filter_resep yang sesuai dengan kategori tersebut, dan hasilnya ditampilkan menggunakan renderResep(filtered). Dengan mekanisme ini, pengguna dapat menampilkan resep sesuai kategori yang dipilih secara interaktif tanpa memuat ulang halaman.



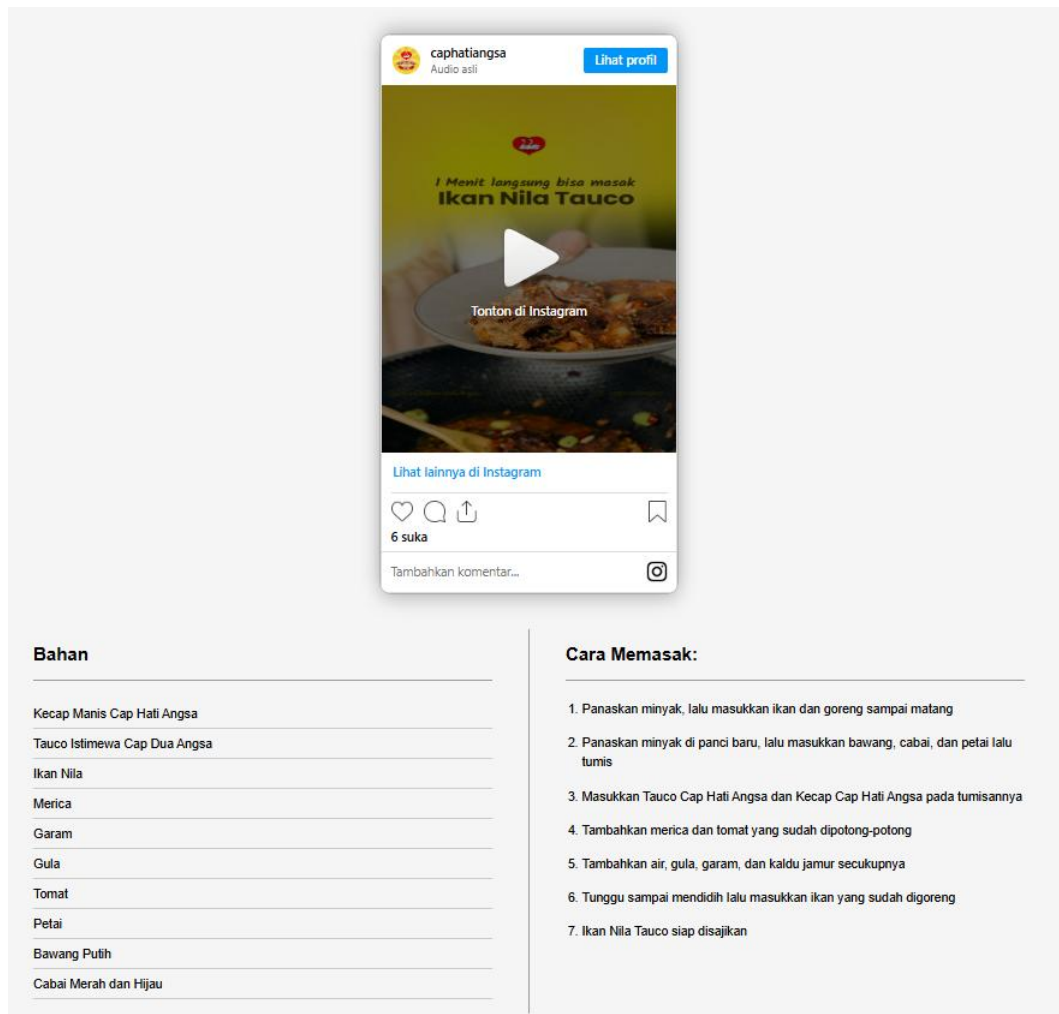
Ikan Nila Tauco

Ikan Nila Tauco merupakan masakan khas dengan cita rasa gurih dan sedikit asin dari tauco pilihan. Dipadukan dengan kecap manis Cap Hati Angsa dan aneka rempah, hidangan ini cocok untuk pecinta masakan tradisional yang kaya rasa.

🕒 - Menit

Gambar 3. 46 Banner dan Penjelasan Detail Resep

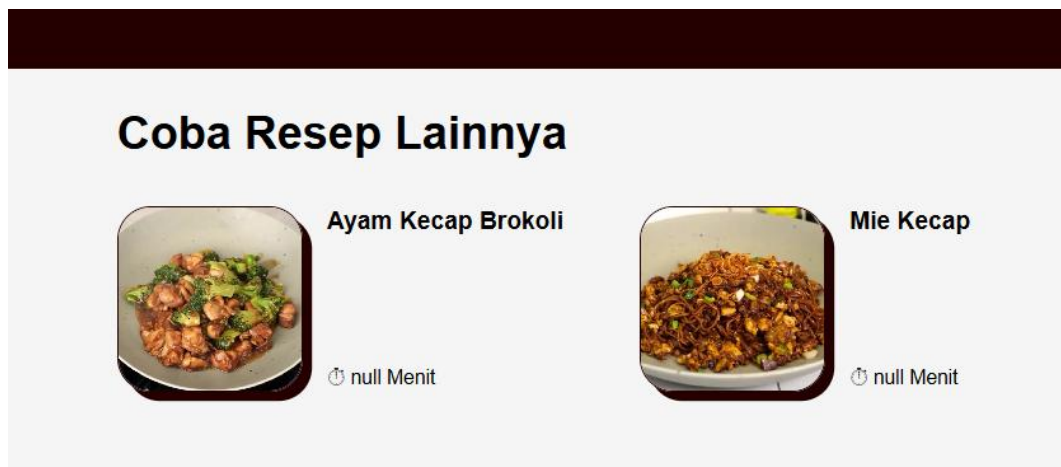
Pada halaman Detail Resep, Gambar 3.46 menampilkan gambar resep yang diletakkan di bagian tengah halaman sebagai fokus utama. Di bawah gambar tersebut terdapat nama resep beserta deskripsi singkat.



Gambar 3. 47 Instagram dan Detail Cara Masak pada Halaman Detail Resep

Pada halaman Detail Resep, Gambar 3.47 menampilkan video IG singkat yang menampilkan demo cara memasak. Selanjutnya, ditampilkan informasi mengenai estimasi waktu memasak dalam satuan menit. Pada bagian bawah halaman, terdapat dua section yang disusun secara berdampingan, yaitu bagian kiri yang berisi daftar bahan dan bagian kanan yang menampilkan cara memasak, sehingga memudahkan pengguna dalam mengikuti resep secara terstruktur.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3. 48 Coba Resep Lainnya pada Halaman Detail Resep

Setelah page divider, Gambar 3.48 menampilkan tulisan “Coba Resep Lainnya” sebagai rekomendasi resep tambahan. Di bawah bagian tersebut terdapat beberapa card resep yang menampilkan gambar, nama resep, serta estimasi waktu memasak dalam satuan menit. Seluruh card pada bagian ini dapat diklik langsung tanpa menggunakan tombol, sehingga pengguna dapat menuju halaman detail resep dengan lebih mudah.

```

<body>
  <!-- Callable Header -->
  <callable-header></callable-header>

  <!-- Banner Detail Resep-->
  <section id="bannerDetailResep">
    
  </section>

  <!-- Detail Resep -->
  <section id="detailResepContainer">
    <!-- Penjelasan Resep -->
    <h1 class="resep-title">Nama Resep</h1>
    <p class="resep-desc">Penjelasan Resep</p>
    <div class="resep-info">
      <div class="info-item">
        ⌚ <span class="lama-resep">0</span> Menit
      </div>
    </div>
  </section>

```

Gambar 3. 49 Callable Header, Banner, dan Penjelasan Resep Detail Resep HTML

Gambar 3.49 menunjukkan potongan kode yang menampilkan bagian atas halaman Detail Resep:

- <callable-header> memanggil header yang konsisten di seluruh halaman.
- <section id="bannerDetailResep"> menampilkan banner resep sebagai elemen visual utama.
- <section id="detailResepContainer"> berisi informasi resep, termasuk: Judul resep (.resep-title), Deskripsi singkat resep (.resep-desc), dan Informasi waktu memasak (.lama-resep) yang menunjukkan durasi resep dalam menit.

```

<!-- Video IG Resep -->
<div id="videoResep">
  <div class="ig-shadow" id="videoResepContainer"></div>
</div>

```

Gambar 3. 50 Video IG Resep Detail Resep HTML

Gambar 3.50 menunjukkan potongan code yang menampilkan video demo cara memasak melalui Instagram setelah penjelasan resep.

```

<!-- Grid Bahan & Cara Memasak -->
<div class="resep-grid">
  <!-- Detail Bahan -->
  <div class="resep-bahan">
    <h2>Bahan</h2>
    <div class="divider"></div>
    <ul id="bahanList"></ul>
  </div>

  <!-- Vertical Divider -->
  <div class="vertical-divider"></div>

  <!-- Detail Cara Memasak -->
  <div class="resep-step">
    <h2>Cara Memasak:</h2>
    <div class="divider"></div>
    <ol id="stepList"></ol>
  </div>
</div>
</section>

<!-- Page Divider -->
<section id="pageDivider"></section>

<!-- Coba Resep Lainnya -->
<section id="resepLainnya">
  <h2 class="resep-lainnya-title">Coba Resep Lainnya</h2>

  <div class="resep-lainnya-container" id="resepLainnyaContainer">
    <!-- Informasi Card dari JS Supabase -->
  </div>

```

Gambar 3. 51 Bahan, Cara Memasak, Page Divider dan Coba Resep Lainnya Detail Resep HTML

Gambar 3.51 menunjukkan potongan kode yang menampilkan struktur utama bagian detail resep pada halaman Detail Resep:

- `<div class="resep-grid">` membagi halaman menjadi dua kolom: Kolom kiri (.resep-bahan) menampilkan daftar bahan-bahan yang diperlukan resep menggunakan `<ul id="bahanList">`. Kolom kanan (.resep-step) menampilkan langkah-langkah memasak menggunakan `<ol id="stepList">`. Kedua kolom dipisahkan oleh `<div class="vertical-divider">` untuk tampilan visual yang rapi.
- `<section id="pageDivider">` berfungsi sebagai pemisah visual antarbagian halaman.

- `<section id="resepLainnya">` menampilkan rekomendasi resep lain dengan judul “Coba Resep Lainnya”, dan `<div id="resepLainnyaContainer">` akan diisi secara dinamis melalui JavaScript dari data Supabase.

```
const supabaseUrl = "https://wqsicwkmuwfblscsbig.supabase.co";
const supabaseAnonKey = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6IjI6Indxc2";
const supabase = createClient(supabaseUrl, supabaseAnonKey);
const params = new URLSearchParams(window.location.search);
const resepId = params.get("id");
```

Gambar 3. 52 Inisialisasi Supabase dan Pengambilan Id dari URL Detail Resep JS

Gambar 3.52 ini sama seperti gambar 3.36 dimana pada codingan tersebut terdapat inisialisasi supabase dan pengambilan id dari url untuk menampilkan halaman melalui id. Dengan adanya sistem ini, pengguna dapat melihat detail resep sesuai dengan yang diinginkan.

```

//Function renderDetailResep
function renderDetailResep(resep) {

    //Banner
    document.querySelector("#bannerDetailResep .banner").src =
        resep.image_url || "assets/noPicture.jpg";

    //Judul & Deskripsi
    document.querySelector(".resep-title").textContent = resep.nama_resep;
    document.querySelector(".resep-desc").textContent = resep.penjelasan_resep || "";
    document.querySelector(".lama-resep").textContent = resep.lama_resep || "-";

    //Video Instagram
    const videoContainer = document.getElementById("videoResepContainer");
    videoContainer.innerHTML = "";

    if (resep.video_url) {
        videoContainer.innerHTML = `
            <blockquote
                class="instagram-media"
                data-instgrm-permalink="${resep.video_url}"
                data-instgrm-version="14">
            </blockquote>
        `;

        //Proses embed IG
        if (window.instgrm) {
            window.instgrm.Embeds.process();
        }
    }
}

```

Gambar 3. 53 renderDetailResep Detail Resep JS

Pada gambar 3.53 terdapat fungsi `renderDetailResep(resep)` digunakan untuk menampilkan detail data resep ke dalam halaman website secara dinamis berdasarkan data yang diterima dari parameter `resep`. Pada bagian awal, fungsi ini mengatur banner gambar resep dengan mengambil nilai `image_url`, dan jika data gambar tidak tersedia maka akan digunakan gambar cadangan. Selanjutnya, fungsi mengisi elemen judul, deskripsi, dan lama memasak resep dengan data yang sesuai menggunakan manipulasi DOM. Selain itu, fungsi ini juga menangani penampilan video resep dari Instagram dengan membuat elemen `blockquote` secara dinamis apabila URL video tersedia. Proses embed video kemudian diproses menggunakan fitur Instagram Embed API agar video dapat ditampilkan dengan benar di halaman. Dengan adanya fungsi ini, tampilan detail resep menjadi lebih interaktif, terstruktur, dan dapat menyesuaikan konten berdasarkan data yang tersedia.

```

//Bahan
const bahanList = document.getElementById("bahanList");
bahanList.innerHTML = "";
resep.bahan_resep.forEach(bahan => {
    const li = document.createElement("li");
    li.textContent = bahan;
    bahanList.appendChild(li);
});

//Step Memasak
const stepList = document.getElementById("stepList");
stepList.innerHTML = "";
resep.step_resep.forEach(step => {
    const li = document.createElement("li");
    li.textContent = step;
    stepList.appendChild(li);
});

loadResepLainnya(resep.id_resep);
}

```

Gambar 3. 54 Lanjutan renderDetailResep Detail Resep JS

Gambar 3.54 merupakan bagian lanjutan fungsi renderDetailResep, sistem menampilkan daftar bahan dan langkah memasak berdasarkan data resep yang diterima. Data bahan resep (bahan_resep) ditampilkan dengan cara mengosongkan elemen daftar terlebih dahulu, kemudian setiap bahan diproses menggunakan perulangan forEach untuk dibuatkan elemen secara dinamis dan ditambahkan ke dalam daftar bahan. Proses ini memastikan bahwa seluruh bahan resep ditampilkan secara rapi dan sesuai dengan data yang tersedia. Selanjutnya, langkah-langkah memasak (step_resep) ditampilkan dengan mekanisme yang serupa. Elemen daftar langkah memasak dikosongkan terlebih dahulu untuk menghindari duplikasi data, kemudian setiap langkah dimasukkan ke dalam elemen dan ditambahkan ke dalam daftar langkah memasak. Dengan cara ini, urutan proses memasak dapat ditampilkan secara sistematis dan mudah dipahami oleh pengguna. Pada bagian akhir, fungsi loadResepLainnya dipanggil dengan parameter id_resep untuk menampilkan rekomendasi resep lain yang berkaitan, sehingga meningkatkan interaksi dan pengalaman pengguna dalam menjelajahi konten resep.

NUSANTARA

3.2.7 Supabase

```
1 create table produk (  
2   id_produk bigint generated by default as identity primary key,  
3   created_at timestamp without time zone default now(),  
4   nama_produk text not null,  
5   deskripsi_produk text,  
6   image_url text,  
7   video_url text[]  
8 );  
9  
10 create table resep (  
11   id_resep bigint generated by default as identity primary key,  
12   lama_resep integer,  
13   created_at timestamp with time zone default now(),  
14   image_url text,  
15   filter_resep text,  
16   bahan_resep text[],  
17   step_resep text[],  
18   video_url text,  
19   nama_resep text not null,  
20   penjelasan_resep text  
21 );  
22
```

Gambar 3. 55 Pembuatan Tabel Produk dan Resep pada Supabase

Pada gambar 3.55 terdapat sql query untuk membuat tabel produk dan resep. Tabel produk sendiri memiliki struktur dari id_produk, created_at, nama_produk, deskripsi_produk, image_url, dan video_url. Untuk tabel resep memiliki struktur, yaitu id_resep, lama_resep, created_at, image_url, filter_resep, bahan_resep, step_resep, video_url, nama_resep, dan penjelasan_resep.

Gambar 3. 56 Gambar 3.56 Pembuatan Bucket pada Supabase

Pada gambar 3.56 terdapat cara membuat bucket untuk menampung data-data seperti gambar pada bucket produk dan resep. Pembuatan bucket ini dapat ditemukan melalui navigasi dari kiri yang bernama storage lalu create new bucket.

Adding new policy to resep

Policy name: 4/50

Allowed operation: ☒ SELECT ☒ INSERT ☒ UPDATE ☒ DELETE

Based on the operations you have selected, you can use the highlighted functions in the [client library](#).

SELECT has been auto selected as UPDATE and DELETE require it

upload download list update move copy remove createSignedUrl createSignedUrls getPublicUrl

Target roles: Defaults to all (public) roles if none selected

Policy definition: 1 bucket_id = 'resep'

View templates Review

Gambar 3. 57 Policy pada Bucket Supabase

Gambar 3.57 menunjukkan proses pembuatan policy pada bucket yang diinginkan. Pada tahap ini, saya mengaktifkan policy select, insert, update, dan delete pada bucket produk dan resep. Untuk menampilkan pengaturan policy tersebut, pengguna dapat menekan tombol New Policy. Tombol ini dapat ditemukan setelah memilih menu Policies yang berada di bagian kanan atas halaman bucket yang telah dibuat.

id_produk	nama_produk	deskripsi_produk	image_url	video_url	created_at	timestamp
1	Kecap Manis Kuning Istimewa Cap Hati Angsa	Kecap manis dengan cita rasa manis seimbang	https://picswkuwvflsccbig.supabase.com/reel/D5H5F31E6L	https://www.instagram.com/reel/D5H5F31E6L	NULL	NULL
2	Kecap Manis Kental No. 1 Cap Hati Angsa	Kecap manis kental dengan rasa manis ku	https://picswkuwvflsccbig.supabase.com/reel/D5H5F31E6L	https://www.instagram.com/reel/D5H5F31E6L	NULL	NULL
3	Kecap Manis Kental No. 2 Cap Hati Angsa	Varian kecap manis kental dengan tingkat	https://picswkuwvflsccbig.supabase.com/reel/D5H5F31E6L	https://www.instagram.com/reel/D5H5F31E6L	NULL	NULL
4	Kecap Encer Istimewa Cap Dua Angsa	Kecap encer berkualitas dengan rasa guri	https://picswkuwvflsccbig.supabase.com/reel/D5H5F31E6L	https://www.instagram.com/reel/D5H5F31E6L	NULL	NULL
5	Kecap Encer No. 1 Cap Dua Angsa	Kecap encer dengan kesetimbangan rasa i	https://picswkuwvflsccbig.supabase.com/reel/D5H5F31E6L	https://www.instagram.com/reel/D5H5F31E6L	NULL	NULL
6	Kecap Encer No. 2 Cap Dua Angsa	Pilihan ekonomis dengan rasa khas kecap	https://picswkuwvflsccbig.supabase.com/reel/D5H5F31E6L	https://www.instagram.com/reel/D5H5F31E6L	NULL	NULL
7	Kecap Refill Manis Besar Cap Hati Angsa	Kemasan isi ulang ukuran besar yang prat	https://picswkuwvflsccbig.supabase.com/reel/D5H5F31E6L	https://www.instagram.com/reel/D5H5F31E6L	NULL	NULL
8	Kecap Refill Manis Kecil Cap Hati Angsa	Kemasan refill ukuran kecil yang praktis d	https://picswkuwvflsccbig.supabase.com/reel/D5H5F31E6L	https://www.instagram.com/reel/D5H5F31E6L	NULL	NULL
9	Kecap Refill Asin Cap Dua Angsa	Kecap asin isi ulang dengan rasa asin guri	https://picswkuwvflsccbig.supabase.com/reel/D5H5F31E6L	https://www.instagram.com/reel/D5H5F31E6L	NULL	NULL
10	Kecap Asin Premium Cap Dua Angsa	Kecap asin berkualitas premium dengan r	https://picswkuwvflsccbig.supabase.com/reel/D5H5F31E6L	https://www.instagram.com/reel/D5H5F31E6L	NULL	NULL
11	Kecap Asin Istimewa Cap Dua Angsa	Kecap asin dengan rasa seimbang dan ku	https://picswkuwvflsccbig.supabase.com/reel/D5H5F31E6L	https://www.instagram.com/reel/D5H5F31E6L	NULL	NULL
12	Tauco Istimewa Cap Dua Angsa	Tauco fermentasi dengan tekstur kental d	https://picswkuwvflsccbig.supabase.com/reel/D5H5F31E6L	https://www.instagram.com/reel/D5H5F31E6L	NULL	NULL

Gambar 3. 58 Mengisi Data Produk pada Tabel Produk Supabase

Gambar 3.58 menunjukkan tabel produk yang berisi berbagai macam data. Pengisian data pada tabel tersebut dapat dilakukan dengan cara melakukan double click pada kolom yang diinginkan. Metode ini merupakan cara yang digunakan oleh saya dalam menginput data produk.

```

1 INSERT INTO resep (
2   nama_resep,
3   penjelasan_resep,
4   bahan_resep,
5   step_resep,
6   video_url
7 ) VALUES
8
9 -- 1. Ayam Kecap Brokoli
10 (
11   'Ayam Kecap Brokoli',
12   'Ayam Kecap Brokoli adalah hidangan rumah yang memadukan ayam empuk dengan brokoli segar dalam balutan kecap manis khas Cap Hati Angsa. Rasanya gurih, manis, dan lezat, cocok disajikan sebagai menu makan siang maupun makan malam keluarga.',
13   ARRAY[
14     'Kecap Manis Kuning Istimewa Cap Hati Angsa',
15     'Kecap Manis Kental Cap Hati Angsa',
16     'Ayam',
17     'saus tiram',
18     'tepung maizena',
19     'bawang putih',
20     'brokoli'
21   ],
22   ARRAY[
23     'Lumuri ayam yang sudah dicuci dan dipotong dengan Kecap Manis Kuning Istimewa Cap Hati Angsa dan Kecap Manis Kental Cap Hati Angsa dalam mangkuk',
24     'Tambahkan saus tiram dan tepung maizena dalam mangkuk',
25     'Ratakan semua bumbu bersama daging ayam',
26     'Panaskan minyak dan tumis bawang putih dalam panci',
27     'Masukkan brokoli dan air secukupnya',
28     'Pindahkan brokoli yang sudah ditumis ke dalam mangkuk',
29     'Panaskan minyak dan masukkan ayam yang sudah dimarinasi ke dalam panci',
30     'Masukkan kemudi brokoli yang sudah ditumis',
31     'Campur hingga matang dan Ayam Kecap Brokoli siap disajikan'
32   ],
33   'https://www.instagram.com/reel/D5H5F31E6L/?utm_source=ig_web_copy_link&igsh=rR10081MF12A=='
34 ),
35

```

Gambar 3. 59 Mengisi Data Resep pada Tabel Resep Melalui Sql Query Supabase

Gambar 3.59 menunjukkan SQL query yang digunakan untuk mengisi data pada tabel resep. Alasan saya menggunakan SQL query dibandingkan dengan metode input langsung seperti pada Gambar 3.58 adalah karena data yang terdapat pada tabel resep memiliki isi yang lebih panjang dan kompleks dibandingkan dengan tabel produk.

3.3 Kendala yang Ditemukan

Selama masa magang di PT Kilang Kecap Angsa, saya menghadapi beberapa kendala yang berkaitan dengan aspek teknis maupun non-teknis. Salah satu kendala utama adalah kompleksitas pengembangan website, terutama saat mengintegrasikan data produk dan resep ke dalam database Supabase. Beberapa fitur membutuhkan logika khusus agar data dapat ditampilkan secara dinamis sesuai kategori dan filter yang dipilih pengguna. Tantangan lainnya muncul dalam memastikan tampilan halaman tetap konsisten dan rapi di berbagai browser.

Dalam menghadapi kendala teknis tersebut, saya melakukan riset secara mandiri dengan mempelajari dokumentasi Supabase dan mencari referensi dari komunitas pengembang. Interaksi dengan database mencakup pengambilan data produk dan resep secara real-time serta pengelolaan data melalui operasi insert, update, dan delete. Hal ini memungkinkan website menampilkan informasi yang selalu terbaru dan relevan bagi pengguna. Pendekatan ini membantu saya menemukan solusi yang tepat untuk menyelesaikan berbagai masalah teknis.

Kendala lain yang saya temui adalah keterbatasan waktu dan sumber daya dalam menyelesaikan seluruh aspek pengembangan website secara mandiri. Saya bertanggung jawab mulai dari perancangan desain, pengembangan frontend, hingga backend dan integrasi database, tanpa adanya anggota tim lain yang membantu. Hal ini menuntut saya untuk mengatur waktu secara efektif dan memprioritaskan tugas-tugas yang paling penting. Selain itu, koordinasi dengan pihak pemasaran dan humas kadang mengalami hambatan karena perbedaan jadwal dan kebutuhan informasi yang berubah selama proses pengembangan.

Selama pengerjaan proyek, saya juga menghadapi kendala terkait pengujian dan validasi fitur website. Beberapa fitur memerlukan pengujian berulang untuk memastikan data dari Supabase tampil dengan benar dan tidak terjadi kesalahan saat pengguna melakukan interaksi. Saya melakukan penyesuaian kode secara bertahap berdasarkan hasil pengujian tersebut agar semua fungsi berjalan sesuai rencana. Proses ini mengajarkan pentingnya ketelitian dan kesabaran dalam mengembangkan sistem yang andal dan stabil.

Selain kendala teknis, komunikasi lintas divisi menjadi tantangan tersendiri. Saya perlu berkoordinasi dengan pihak pemasaran dan humas untuk memastikan konten produk, resep, dan informasi perusahaan akurat dan sesuai kebutuhan website. Kadang terjadi perbedaan jadwal atau perubahan informasi yang membutuhkan penyesuaian cepat dari pihak saya. Pengalaman ini melatih kemampuan adaptasi dan komunikasi saya, sehingga saya dapat tetap menyelesaikan tugas sesuai target waktu dan standar perusahaan.

3.4 Solusi atas Kendala yang Ditemukan

Selama menghadapi kendala teknis pada integrasi data produk dan resep, saya mengatasi masalah tersebut dengan memecah fitur menjadi bagian-bagian kecil dan menguji setiap bagian secara bertahap. Saya menggunakan dokumentasi Supabase secara intensif untuk memahami fungsi query dan pengelolaan data yang tepat. Selain itu, saya mencoba beberapa metode pengolahan data hingga berhasil menampilkan informasi sesuai kategori dan filter yang diinginkan. Pendekatan bertahap ini membantu mengurangi risiko kesalahan dan mempercepat penyelesaian pengembangan fitur.

Untuk kendala terkait riset mandiri dan interaksi database, saya membuat catatan langkah-langkah troubleshooting setiap kali menemukan bug atau error. Saya juga memanfaatkan forum dan komunitas pengembang untuk menemukan solusi yang terbukti efektif. Dengan demikian, setiap masalah teknis dapat diselesaikan tanpa mengganggu alur pengembangan utama. Proses ini meningkatkan kemampuan problem solving saya secara signifikan.

Dalam menghadapi keterbatasan waktu dan sumber daya, saya menyusun jadwal kerja harian dan prioritas tugas secara rinci. Setiap tugas penting ditangani lebih dahulu, sedangkan tugas yang tidak mendesak dijadwalkan ulang agar tidak menumpuk. Pendekatan manajemen waktu ini memastikan semua aspek pengembangan dapat diselesaikan sesuai target. Saya juga menggunakan alat bantu manajemen proyek sederhana untuk memantau progres secara berkala.

Untuk kendala pengujian dan validasi fitur, saya melakukan pengujian secara berulang pada setiap modul website. Saya menyiapkan skenario pengujian untuk memastikan data dari database tampil benar dan tidak ada kesalahan saat interaksi pengguna. Setiap bug yang ditemukan dicatat dan diperbaiki sebelum melanjutkan ke modul berikutnya. Metode ini memastikan website stabil dan fungsional saat selesai dikembangkan.

Mengatasi kendala komunikasi lintas divisi, saya membangun jadwal koordinasi rutin dengan pemasaran dan humas. Informasi produk, resep, dan konten perusahaan dikompilasi dan diverifikasi sebelum dimasukkan ke website. Saya juga membuat catatan perubahan dan update agar mudah menyesuaikan website ketika terjadi perubahan informasi. Strategi ini memastikan kolaborasi tetap efektif dan website selalu menampilkan konten yang akurat dan relevan.

Melalui penerapan berbagai strategi dan pendekatan tersebut, saya berhasil mengatasi kendala teknis maupun non-teknis selama pengembangan website PT Kilang Kecap Angsa. Pendekatan bertahap, riset mandiri, pengujian berulang, manajemen waktu, dan koordinasi lintas divisi terbukti efektif dalam memastikan fitur dan konten website berjalan sesuai rencana. Pengalaman ini meningkatkan kemampuan problem solving, adaptasi, dan komunikasi saya secara signifikan. Dengan demikian, saya mampu menyelesaikan proyek pengembangan website secara mandiri dan optimal, sekaligus memperoleh pemahaman mendalam tentang proses pengembangan sistem informasi di lingkungan industri.