

Kelompok 7

**Website Pemesanan Catering Warung Nasi Marsel**  
Laporan Proyek Perangkat Lunak

Nama Penyusun:

- Rolanda Evan M. (11121163)
- Fakih Syamaidzar (10121418)
- Rafif Atha Naufal (11121404)
- Muhamad Arya Pratama (10121739)
- Sindi Marselina (11121223)

Tanggal : (01/24/2025)

**Daftar Isi**

<b>1. PENDAHULUAN</b>	<b>2</b>
1.1 Latar Belakang	2
1.2 Tujuan	2
1.3 Ruang Lingkup	3
<b>2. GAMBARAN UMUM SISTEM (SRS)</b>	<b>4</b>
2.1 Fungsi Produk	4
2.2 Lingkungan Operasi	5
2.3 Kelompok Pengguna dan Karakteristik	6
<b>3. ARSITEKTUR SISTEM (SDD)</b>	<b>7</b>
3.1 Gambaran Umum Arsitektur Sistem	7
3.2 Deskripsi Dekomposisi	8
<b>4. DATA (SDD)</b>	<b>10</b>
4.1 Deskripsi Data	10
4.2 Kamus Data	11
<b>5. ANTARMUKA (SDD)</b>	<b>17</b>
5.1 Gambaran Umum Antarmuka	17
5.2 Tampilan Layar	17
5.3 Obyek Layar dan Tindakan	19
<b>6. PELAKSANAAN SRS (Gantt Chart)</b>	<b>20</b>
<b>7. PERSONIL</b>	<b>21</b>
<b>8. LAMPIRAN</b>	<b>22</b>

---

## **1. Pendahuluan**

### **1.1 Latar Belakang**

Perkembangan teknologi informasi saat ini memegang peranan yang sangat penting dalam berbagai aspek kehidupan, seperti industri, perdagangan, kesehatan, pendidikan, dan lainnya. Internet, sebagai jaringan komputer global, menjadi media komunikasi dan informasi modern yang memudahkan publik untuk mengakses berbagai data dan informasi. Berbagai lembaga, instansi pemerintah, serta perusahaan swasta memanfaatkan internet, khususnya melalui website, untuk menyampaikan profil, iklan, data, dan informasi mereka kepada masyarakat.

Catering Warung Nasi Marsel adalah salah satu contoh usaha yang masih menghadapi tantangan dalam penyampaian informasi dan layanan kepada pelanggan. Penyajian informasi yang masih sederhana menyulitkan calon pelanggan untuk mencari informasi atau melakukan pemesanan. Saat ini, masyarakat yang membutuhkan informasi mengenai layanan atau ingin memesan makanan harus datang langsung atau menghubungi pemilik secara manual. Hal ini menyebabkan proses pencarian informasi dan pemesanan menjadi kurang efektif dan efisien, serta menghambat kenyamanan pelanggan, terutama bagi mereka yang memiliki keterbatasan waktu atau lokasi.

Sistem pemesanan yang masih manual ini menjadi kendala bagi Catering Warung Nasi Marsel dalam menjangkau lebih banyak pelanggan, serta membatasi akses informasi dan layanan pemesanan. Untuk mengatasi kendala tersebut, dilakukan penelitian yang berjudul “Pembuatan Aplikasi Pemesanan Catering Warung Nasi Marsel Berbasis Website.” Diharapkan dengan adanya sistem informasi berbasis web ini, masyarakat dapat mengakses informasi dan melakukan pemesanan dengan mudah, cepat, dan hemat, kapanpun dan dimanapun tanpa kendala jarak dan waktu.

### **1.2 Tujuan**

Proyek Website Pemesanan Catering Warung Nasi Marsel adalah proyek untuk membuat website pemesanan Catering Warung Nasi Marsel agar bisa lebih dikenal dan diketahui secara luas. Serta, membangun platform digital inovatif yang memudahkan pelanggan dalam memesan makanan berkualitas, sehat, dan lezat dengan pelayanan cepat serta transparan, sehingga menjadi pilihan utama dalam

---

---

memenuhi kebutuhan catering masyarakat modern.

### 1.3 Ruang Lingkup

Sistem Pemesanan Catering Warung Nasi Marsel ini adalah sebuah platform berbasis web yang memungkinkan pengguna untuk memesan produk catering secara online dengan berbagai pilihan yang dapat disesuaikan oleh pengguna, jangkauan layanan yang lebih luas, dan meningkatkan pengalaman pelanggan dengan sistem pemesanan yang lebih efektif dan efisien.

Perangkat lunak dibuat untuk membuat catering warung nasi marsel agar lebih dikenal secara luas. Serta membangun platform digital inovatif yang memudahkan pelanggan dalam memesan makanan lezat, sehat, dan berkualitas dengan pelayanan cepat serta transparan, sehingga menjadi pilihan utama dalam memenuhi kebutuhan catering masyarakat.

Manfaat pembuatan perangkat lunak ini, antara lain:

- Pelanggan dapat memesan catering dimana saja dan kapan saja.
- Meningkatkan proses pemesanan dan pembayaran secara digital.
- Dengan adanya perangkat lunak ini dapat memperluas jangkauan pasar, memungkinkan orang-orang dari berbagai daerah dapat melihat dan memesan layanan catering.

Fungsi yang terdapat di perangkat lunak antara lain:

- Perangkat lunak dapat menampilkan berbagai menu dan catering, lengkap dengan harga, deskripsi, dan gambar untuk memudahkan pelanggan memilih sesuai kebutuhan.
- Perangkat lunak memfasilitasi pelanggan untuk melakukan pemesanan secara online melalui form pemesanan, sehingga proses order makanan lebih praktis.
- Memberikan informasi berisi status pemesanan dalam bentuk pesan dengan otomatis melalui fitur whatsapp API.
- Menyediakan sistem pembayaran dengan metode pembayaran digital seperti transfer bank atau e-wallet.

---

## **2. Gambaran Umum Sistem**

### **2.1 Fungsi Produk**

Website Pemesanan Catering Warung Nasi Marsel dirancang untuk memenuhi kebutuhan utama pelanggan dan mendukung operasional bisnis dengan fungsi utama berikut:

Fungsi Utama yang Harus Dilakukan Produk:

- 1) Fungsi Informasi
  - ❖ Menampilkan daftar menu makanan lengkap dengan deskripsi, foto, dan harga.
  - ❖ Memberikan informasi terkait promo, paket catering, dan syarat pemesanan.
  - ❖ Menyediakan informasi kontak dan Lokasi untuk pelanggan yang membutuhkan layanan tambahan.
- 2) Fungsi Pemesanan
  - ❖ Memungkinkan pelanggan memilih menu dan menambahkan pesanan ke keranjang belanja.
  - ❖ Mendukung pemesanan terjadwal untuk catering di waktu mendatang.
- 3) Fungsi Pembayaran
  - ❖ Mendukung metode pembayaran seperti QRIS untuk semua bank.
- 4) Fungsi Notifikasi
  - ❖ Memberikan pembaruan status pesanan Melalui WhatsApp
  - ❖ Memberi notifikasi Ketika melakukan pembayaran serta penyelesaian pesanan melalui WhatsApp
- 5) Fungsi Ulasan dan Interaksi
  - ❖ Mengizinkan pelanggan memberikan ulasan dan penilaian terhadap layanan dan produk yang dipesan
  - ❖ Menyediakan fitur atau konsultasi melalui WhatsApp terkait kebutuhan catering.

- 
- 6) Fungsi Pengelolaan Backend (Admin)
    - ❖ Memungkinkan pengelola bisnis untuk menambah, mengubah, atau menghapus menu yang tersedia.
    - ❖ Memantau dan mengelola pesanan masuk termasuk status pembayaran dan pengiriman.
    - ❖ Melihat data analitik untuk memahami pola pembelian pelanggan.

## 2.2 Lingkungan Operasi

Pada tahap ini dilakukan analisis mengenai kebutuhan sistem pada Pembuatan website. Dalam membuat sistem, memerlukan kebutuhan pendukung yaitu kebutuhan hardware dan software sebagai berikut :

- 1. Perangkat Keras (Hardware)
  - a. Device : VivoBook\_ASUSLaptop X415EP\_A416EP
  - b. Processor : 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
  - c. RAM : 8.00 GB
  - d. System Type : 64-bit operating system, x64-based processor
- 2. Perangkat Lunak (Software)
  - a. Figma

Figma digunakan untuk desain layout website. Aplikasi ini memungkinkan penulis untuk merancang tampilan visual website dengan mudah dan kolaboratif
  - b. Windows 11 Home Single Language

Dalam pembuatan website ini, penulis menggunakan system operasi Windows 11 Home Single Language
  - c. Visual Studio Code

Visual Studio Code sebagai kode editor
  - d. Xampp

Ini merupakan software bantu untuk membuat database pada pemrograman PHP

---

## **2.3 Pengguna dan Karakteristik**

Kelompok Pengguna Berdasarkan Fungsi Produk:

a) Admin

Karakteristik:

- 1) Memiliki hak istimewa penuh untuk mengelola produk.
- 2) Mengelola data pengguna, konten, dan pengaturan sistem.
- 3) Membutuhkan antarmuka yang mendukung efisiensi pengelolaan data, akses keamanan tinggi, serta pelaporan aktivitas pengguna.

b) Pengguna

Karakteristik:

- 1) Menggunakan fungsi utama produk seperti pembelian, pendaftaran, atau interaksi lainnya.
- 2) Memiliki hak terbatas, hanya dapat mengakses fitur yang sesuai dengan kebutuhan mereka.
- 3) Membutuhkan antarmuka yang intuitif, responsif, dan mudah dipahami.

c) Pengguna Tamu (Guest)

Karakteristik:

- 1) Tidak perlu melakukan pendaftaran ataupun login, hanya dapat melihat Informasi public (statis), dan terdapat batasan tertentu dalam hak akses.

---

## 3. Arsitektur Sistem

### 3.1 Gambaran Umum Arsitektur

Arsitektur sistem pada platform pemesanan Catering Warung Nasi Marsel dirancang menggunakan pendekatan *tiered architecture* yang terdiri dari tiga lapisan utama, yaitu *Frontend*, *Backend*, dan *Database*. Tiap lapisan memiliki tanggung jawab tertentu yang bekerja sama untuk menyediakan fungsionalitas lengkap sistem.

Sistem ini diilustrasikan dengan diagram berikut:

- Frontend (Client-Side):
  - Menggunakan framework React.js untuk membangun antarmuka yang interaktif, responsif, dan user-friendly.
  - Bertanggung jawab untuk menangani interaksi langsung dengan pengguna, seperti navigasi, form pemesanan, dan penyajian informasi menu.
- Backend (Server-Side):
  - Menggunakan framework Laravel untuk mendukung pengelolaan logika bisnis, autentikasi pengguna, dan komunikasi dengan database.
  - Menyediakan API RESTful yang menghubungkan Frontend dengan Database, termasuk fitur seperti pengelolaan menu, pemesanan, notifikasi, dan integrasi pembayaran.
- Database:
  - Menggunakan MySQL sebagai sistem manajemen basis data untuk menyimpan informasi pelanggan, pesanan, menu, dan transaksi.
  - Data disusun secara terstruktur untuk mendukung efisiensi pengambilan informasi dan skalabilitas sistem.
- Integrasi API:
  - Sistem ini terhubung dengan API eksternal, seperti WhatsApp API untuk notifikasi otomatis kepada pelanggan dan layanan pembayaran digital (e-wallet atau transfer bank).

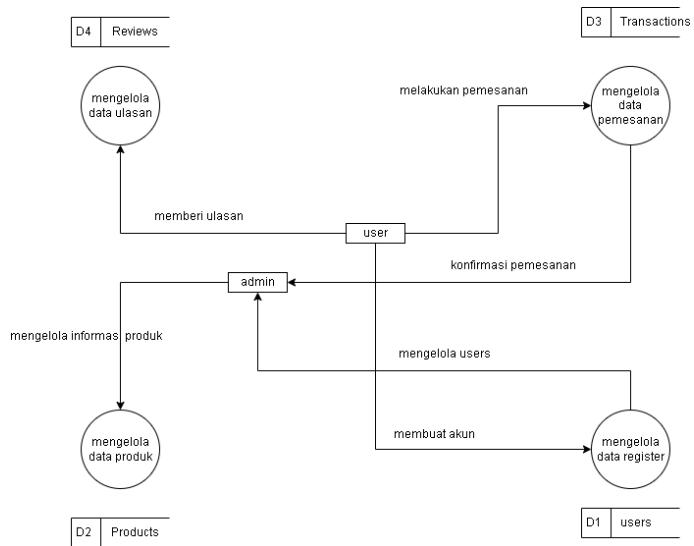
---

## 3.2 Deskripsi Dekomposisi

### Subsistem Utama

1. User Interface Subsystem (Frontend):
  - ✓ Menyediakan tampilan menu dan paket catering, form pemesanan, dan navigasi halaman.
  - ✓ Mengatur input pengguna seperti pilihan menu, jumlah porsi, dan data pemesanan.
2. Order Management Subsystem (Backend):
  - ✓ Memproses data pemesanan dari frontend dan menyimpan ke database.
  - ✓ Menyediakan informasi status pemesanan yang dikirimkan melalui WhatsApp API.
3. Inventory & Menu Management Subsystem (Backend):
  - ✓ Memantau ketersediaan menu dan bahan baku berdasarkan data yang dimasukkan oleh admin.
  - ✓ Memungkinkan pembaruan informasi menu, gambar, deskripsi, dan harga.
4. Payment Integration Subsystem (Backend):
  - ✓ Mengelola transaksi pembayaran dengan layanan pihak ketiga.
  - ✓ Menyimpan status pembayaran untuk dicatat dalam laporan penjualan.
5. Database Management Subsystem (Database):
  - ✓ Menyimpan data pelanggan, menu, pesanan, dan riwayat transaksi.
  - ✓ Mendukung kueri cepat untuk laporan penjualan dan monitoring performa bisnis.

## DFD atau Diagram Alur Data



---

## 4. Data

### 4.1 Deskripsi Data

Domain informasi sistem pemesanan catering Warung Nasi Marsel mencakup data pelanggan, menu catering, pemesanan, transaksi, dan administrasi. Struktur data dirancang untuk menyimpan, memproses, dan mengatur informasi secara efisien, mendukung kebutuhan sistem dalam memberikan layanan pemesanan yang praktis dan transparan.

Data utama disimpan dalam Database MySQL dengan tabel-tabel berikut:

1. **Tabel Pelanggan:** Menyimpan informasi pengguna yang melakukan pemesanan.
2. **Tabel Menu:** Berisi data tentang menu dan paket catering, termasuk deskripsi, harga, dan gambar.
3. **Tabel Pemesanan:** Mendokumentasikan informasi terkait pemesanan pelanggan, seperti jumlah porsi, tanggal, dan status pemesanan.
4. **Tabel Transaksi:** Menyimpan data pembayaran, termasuk metode, status, dan tanggal transaksi.
5. **Tabel Admin:** Menyimpan data admin yang bertugas mengelola konten website, menu, dan pesanan.

#### Struktur Penyimpanan Data

1. **Normalisasi Data:** Data dirancang menggunakan teknik normalisasi hingga bentuk 3NF (Third Normal Form) untuk mencegah redundansi data dan memastikan integritas referensial antar tabel.
2. **Relasi Antar Tabel:** Sistem menggunakan relasi 1-N dan N-N sesuai kebutuhan, seperti relasi antara pelanggan dan pemesanan, serta menu dan pemesanan.

#### Proses Data

1. **Penyimpanan:** Data baru dari form pemesanan pelanggan langsung disimpan ke tabel terkait melalui logika backend.
  2. **Pengambilan:** Data ditampilkan ke pengguna melalui query yang dijalankan oleh backend, misalnya, daftar menu di halaman beranda.
-

- 
3. **Pengelolaan:** Admin dapat menambah, mengubah, atau menghapus data menu atau pesanan melalui dashboard yang terhubung ke backend.

## 4.2 Kamus Data

**Tabel yang digunakan**

Entitas	Deskripsi
Users	Menyimpan informasi nama pelanggan, alamat, nomer telepon, email, dan password.
Orders	Berisi informasi nama pemesan, kapan pengguna melakukan pemesanan, menu dan jumlah yang dipesan, total harga, nomer telepon, dan status orderan.
Foods	Berisi daftar informasi nama makanan, deskripsi makanan, harga, gambar makanan, dan jenis kategori.
Categories	Berisi informasi nama kategori, dan gambar.
Payments	Berisi informasi detail tentang id order, tanggal pembayaran, dan status pembayaran.
Shopping Cart	Mengelola menu yang ingin dibeli oleh pengguna sebelum melakukan order.
Stocks	Berisi informasi nama dan jumlah stok barang.
Suggestions	Berisi informasi tentang nama pelanggan dan saran dari pelanggan tersebut.
Admins	Menyimpan informasi nama admin,

---

---

	email, dan password.
--	----------------------

▪ **Tabel Users**

<b>Nama</b>	<b>Tipe Data</b>	<b>Deskripsi</b>
id	Integer (PrimaryKey)	Berikan nomor urut secara otomatis, untuk id pelanggan.
name	Varchar (255)	Nama lengkap pelanggan.
alamat	Varchar (255)	Alamat tempat tinggal pelanggan
email	Varchar (50)	Alamat email pelanggan untuk keperluan komunikasi.
Telepon	Varchar (15)	Nomor telepon pelanggan untuk notifikasi pesanan.
Password	Varchar (10)	Beri password untuk melakukan pemesanan.

▪ **Tabel Orders**

<b>Nama</b>	<b>Tipe Data</b>	<b>Deskripsi</b>
id	Integer (PrimaryKey)	Berikan nomor urut secara otomatis, untuk id order.
user_id	Integer	Foreign key dari table users.

---

---

name	Varchar(255)	Berisi nama pelanggan yang melakukan order.
address	Varchar(255)	Berisi alamat pelanggan yang melakukan order.
phone	Varchar(15)	Berisi nomer telepon pelanggan yang melakukan order.
food_name	Varchar(255)	Berisi nama makanan yang diorder.
quantity	Integer	Berisi jumlah makanan yang disimpan.
total_amount	Integer	Berisi total harga makanan yang diorder.
status	Enum('pending', 'processing', 'completed', 'cancelled')	Berisi status pesanan.

▪ **Tabel Foods**

Nama	Tipe Data	Deskripsi
id	Integer	Berisikan nomor urut secara otomatis, untuk id food.
nama	Varchar (255)	Berisi informasi nama menu.
deskripsi	Text	Berisi informasi deskripsi menu.
harga	Integer	Berisi informasi harga menu.
gambar	Varchar	Berisi gambar menu

---

---

		makanan.
category_id	Integer	Foreign key dari tabel category.

- **Tabel Categories**

Nama	Tipe Data	Deskripsi
id	Integer	Berisikan nomor urut secara otomatis, untuk id kategori.
name	Varchar(255)	Berisi informasi nama kategori.
image	Varchar	Berisi gambar kategori.

- **Tabel Payments**

Nama	Tipe Data	Deskripsi
id	Integer	Berisikan nomor urut secara otomatis, untuk id payment.
order_id	Integer	Foreign key dari table order.
payment_date	datetime	Berisi informasi tanggal pembayaran yang dilakukan oleh pengguna.
payment_status	enum('pending', 'completed', 'failed')	Berisi informasi status pembayaran.

- **Tabel Shopping Cart**

---

---

<b>Nama</b>	<b>Tipe Data</b>	<b>Deskripsi</b>
id	Integer	Berisikan nomor urut secara otomatis, untuk id cart.
user_id	Integer	Foreign key dari tabel user.
food_id	Integer	Foreign key dari tabel food.
nama_menu	Varchar(255)	Berisi informasi nama menu.
jumlah	Integer	Berisi informasi jumlah makanan yang dimasukkan ke keranjang.
harga_total	Integer	Berisi informasi harga total makanan yang dimasukkan ke keranjang.
harga_satuan	Integer	Berisi informasi harga satuan dari makanan yang dimasukkan ke keranjang.

▪ **Tabel Stocks**

<b>Nama</b>	<b>Tipe Data</b>	<b>Deskripsi</b>
id	Integer	Berisikan nomor urut secara otomatis, untuk id stok.

---

---

nama	Varchar(255)	Berisi informasi nama dari stok.
jumlah	Integer	Berisi informasi jumlah dari stok barang.

▪ **Tabel Suggestions**

Nama	Tipe Data	Deskripsi
id	Integer	Berisikan nomor urut secara otomatis, untuk id saran.
user_id	Integer	Foreign key dari tabel user.
content	Text	Berisi informasi dari saran pelanggan.

▪ **Tabel Admins**

Nama	Tipe Data	Deskripsi
id	Integer (PrimaryKey)	Berisikan nomor urut secara otomatis, untuk id admin.
name	Varchar (255)	Nama lengkap admin.
email	Varchar (50)	Alamat email admin untuk keperluan komunikasi.
Password	Varchar (10)	Berisi password admin.

---

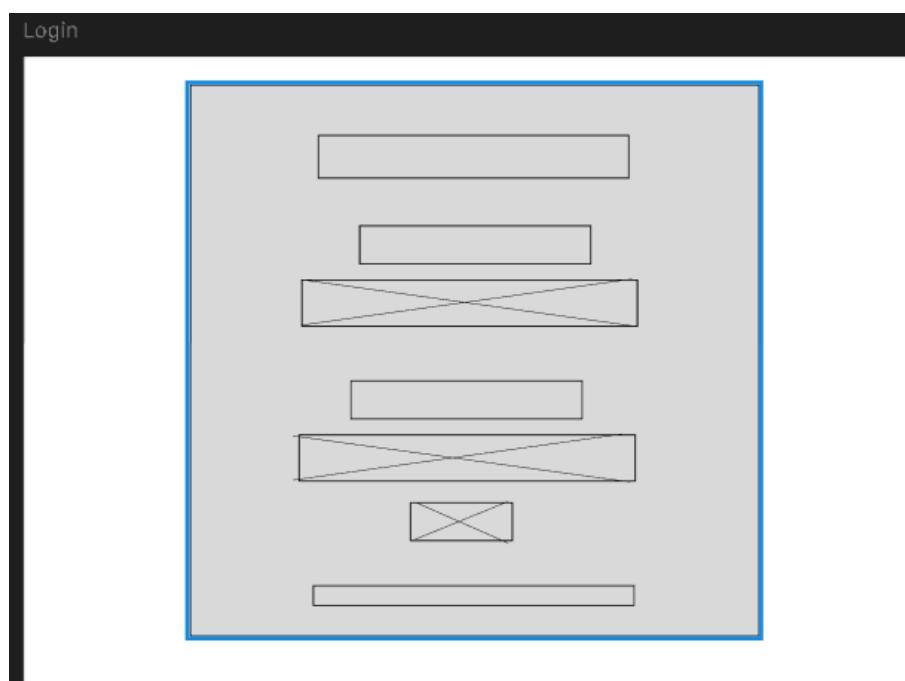
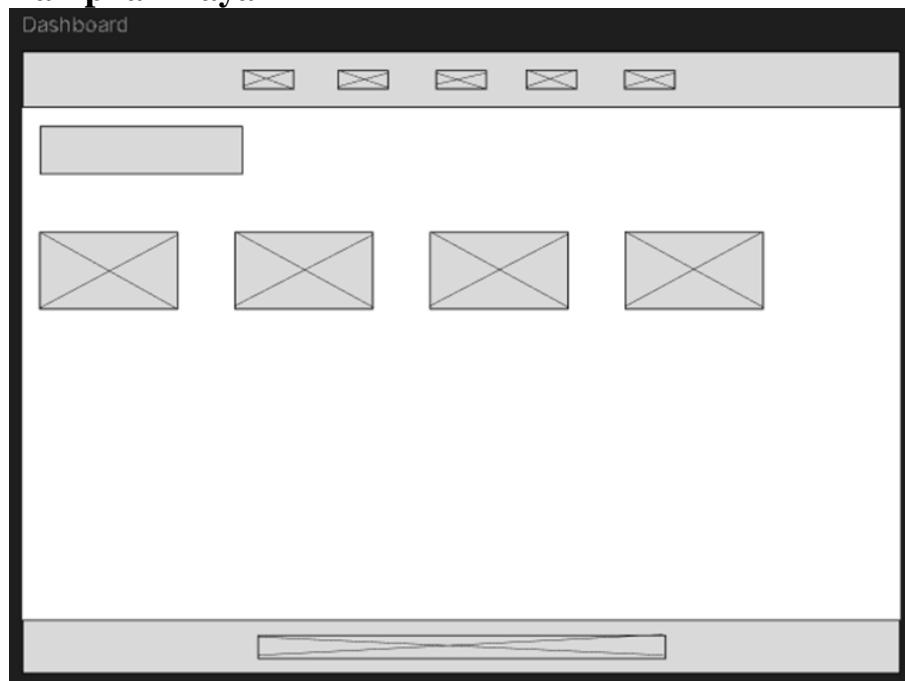
---

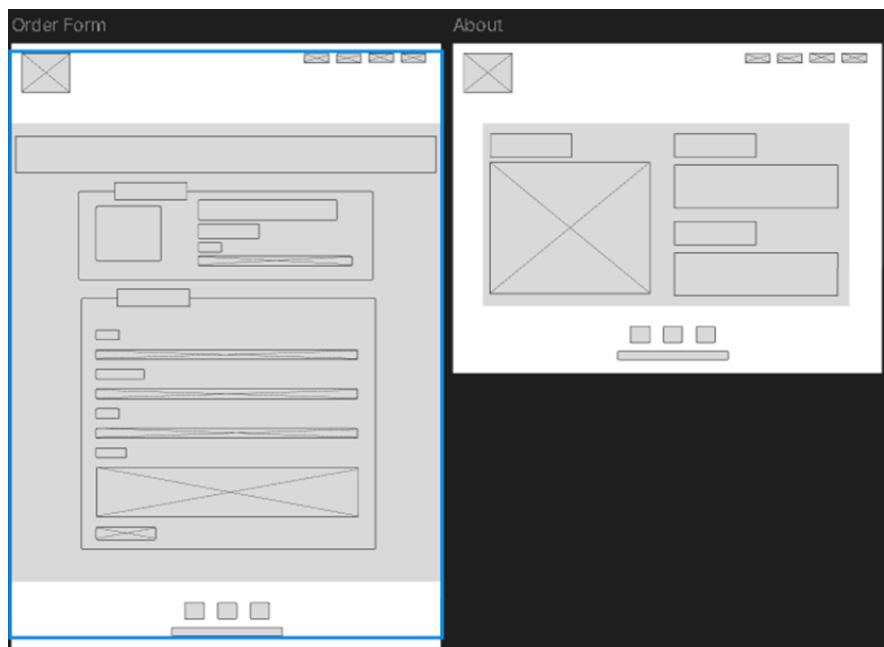
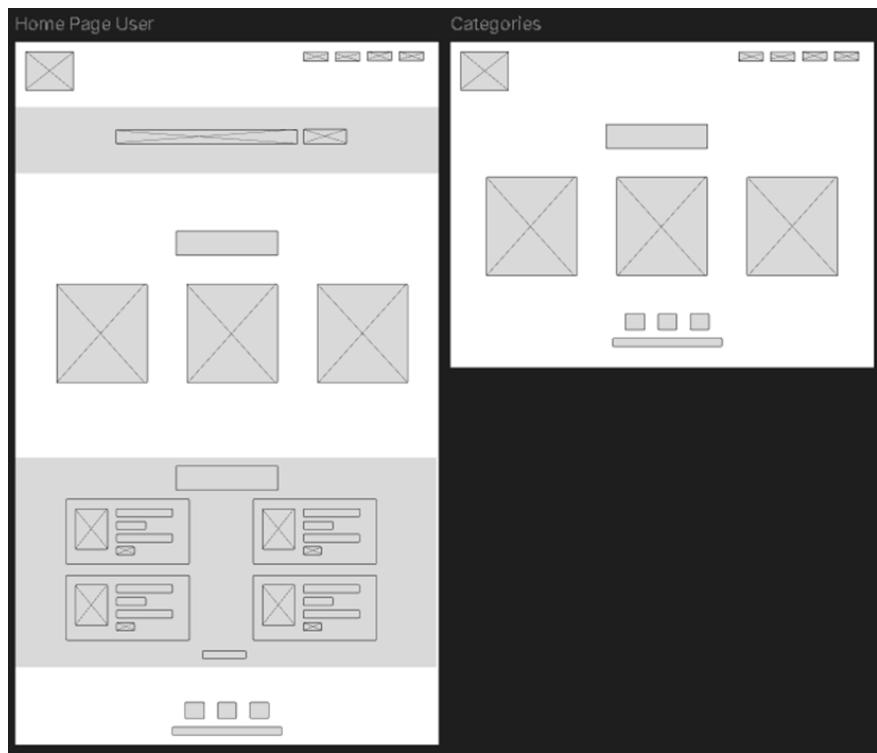
## 5. Antarmuka

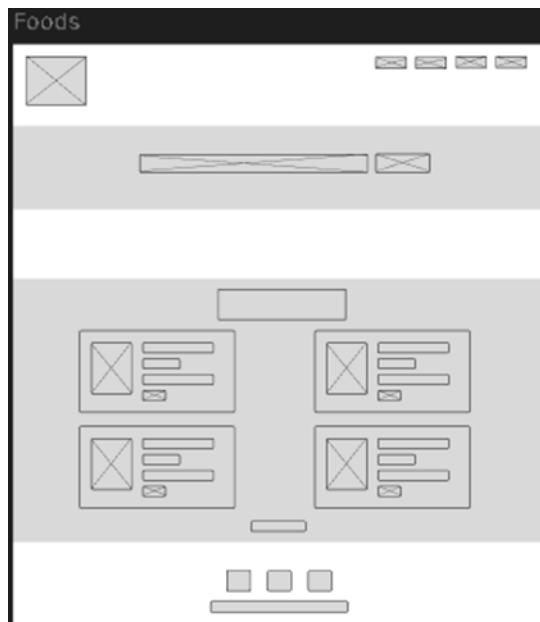
### 5.1 Gambaran Umum Antar Muka

Pada Homepage terdapat menu pada navigasi bar yang berisi Home, catalogue, About Us, dan terdapat button login.

### 5.2 Tampilan Layar







### 5.3 Objek Layar dan Tindakan

Objek	Tindakan
Header	Terdapat navbar berisi home, catalogue, About Us, dan terdapat button untuk login. Pelanggan dapat login terlebih dahulu untuk melakukan pemesanan product.
Content	bagian isi terdapat card slider untuk menampilkan catalogue, dan dapat dilihat lebih detail seperti beberapa gambar yang disediakan dan beberapa deskripsi produk.
Footer	Terdapat beberapa contact info seperti gmail, dan beberapa link yang mengarah ke profile perusahaan.

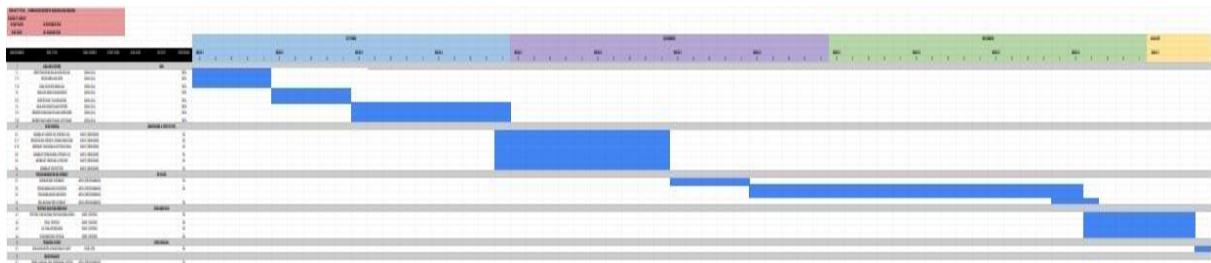
## 6. Pelaksanaan

Tanggal	Kegiatan	Hasil
11-17 Oktober 2024	<ul style="list-style-type: none"><li>• Mengadakan kickoff meeting</li><li>• Membahas kebutuhan proyek</li></ul>	Dokumen Konsep Proyek
18-24 Oktober 2024	<ul style="list-style-type: none"><li>• Melakukan diskusi dengan tim untuk menentukan kebutuhan system</li><li>• Mengidentifikasi fitur utama seperti pemesanan online, daftar menu, dan integrasi pembayaran</li><li>• Membuat dokumen kebutuhan</li></ul>	Dokumen rencana anggaran biaya, dokumen spesifikasi kebutuhan software
25-31 Oktober 2024	<ul style="list-style-type: none"><li>• Merancang struktur navigasi system</li><li>• Membuat diagram</li><li>• Membuat rancangan database</li></ul>	Dokumen diagram
1-7 November 2024	<ul style="list-style-type: none"><li>• Mendesain wireframe</li><li>• Memilih teknologi yang akan digunakan (Laravel, Javascript)</li></ul>	Wireframe diselesaikan berdasarkan masukan tim, teknologi utama ditetapkan
8-14 November 2024	<ul style="list-style-type: none"><li>• Membuat struktur database untuk sistem pemesanan</li><li>• Mendasain tabel - tabel</li></ul>	Database awal selesai dirancang
15-21 November 2024	<ul style="list-style-type: none"><li>• Memulai pengembangan Frontend</li></ul>	Halaman utama website selesai

22-28 November 2024	<ul style="list-style-type: none"> <li>Mengembangkan system backend untuk CRUD menu dan pesanan</li> <li>Menghubungkan frontend dengan backend</li> </ul>	Fitur CRUD menu dan pesanan selesai
29-5 Desember 2024	<ul style="list-style-type: none"> <li>Menambahkan fitur autentikasi pengguna (pelanggan dan admin)</li> <li>Mengimplementasikan sesi login</li> </ul>	System autentikasi selesai dan diuji coba
6-12 Desember 2024	<ul style="list-style-type: none"> <li>Mengintegrasikan system pembayaran approval (QRIS atau bank)</li> <li>Mengimplementasikan fitur notifikasi untuk status pesanan</li> </ul>	System pembayaran dan fitur notifikasi selesai
13-19 Desember 2024	<ul style="list-style-type: none"> <li>Melakukan pengujian unit dan integrasi pada seluruh fitur</li> <li>Memperbaiki bug</li> </ul>	Pengujian unit dan integrasi selesai
20-26 Desember 2024	<ul style="list-style-type: none"> <li>Melakukan revisi berdasarkan masukan</li> </ul>	Perubahan pada desain dan alur system
27-2 Januari 2025	<ul style="list-style-type: none"> <li>Mengimplementasikan fitur tambahan berdasarkan masukan seperti kategori</li> <li>Mengoptimalkan performa website</li> </ul>	Fitur tambahan selesai dan performa website meningkat
3-9 Januari 2025	<ul style="list-style-type: none"> <li>Melakukan pengujian website</li> <li>Memastikan semua fitur berfungsi sesuai kebutuhan</li> </ul>	System dinyatakan siap digunakan

---

10-16 Januari 2025	<ul style="list-style-type: none"> <li>• Melakukan deployment</li> <li>• Mengoreksi kesalahan dari masukan</li> </ul>	Kesalahan diperbaiki dan website berhasil diluncurkan
17-23 Januari 2025	<ul style="list-style-type: none"> <li>• Menyiapkan dokumentasi akhir proyek termasuk panduan pengguna dan teknis</li> <li>• Melakukan pelatihan singkat kepada tim</li> </ul> <p>Warung Nasi Marsel</p>	Dokumentasi dan pelatihan selesai,



---

## **7. Personil**

- Rolanda Evan M. (11121163) – Project Manager
- Fakih Syamaidzar (10121418) – System Analyst
- Rafif Atha Naufal (11121404) – UI/UX Designer
- Muhamad Arya Pratama (10121739) – Programmer
- Sindi Marselina (11121223) - Tester (Quality Assurance)

---

## **Lampiran**

1. Black Box Testing
2. User Acceptance Test
3. Berita Acara serah terima aplikasi
4. Dokumentasi berupa foto acara serah terima laporan dan aplikasi, serta foto pada saat mencoba aplikasi di *stakeholder*.
5. Buku manual penggunaan aplikasi
6. Kode program dari pembuatan proyek sistem informasi.

### **Catatan :**

Laporan dan lampiran dijadikan 1 file dalam bentuk dokumen .PDF

---

## Lampiran Black Box Testing

### PENGUJIAN UNIT

#### a. Pengujian Unit Halaman Pengguna (Pembeli)

##### 1. Pengujian Halaman Dashboard Sebelum Login

Tabel 1.

Hasil Pengujian *Black Box Testing* Halaman Dashboard Sebelum Login

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan Logo Warung Nasi Marsel	Logo	Berhasil menampilkan <i>output</i> berupa tampilan halaman <i>dashboard</i> beserta <i>button</i> pemesanan makanan	Website berjalan dengan baik sesuai harapan	Sukses
2	Menekan Button “Pesanan Sekarang” untuk melihat daftar menu catering	Button “Pesanan Sekarang”	Berhasil menampilkan <i>output</i> berupa tampilan menu <i>catering</i> beserta <i>button</i> “ <i>Shopping Cart</i> ” dan <i>button</i> “Pesanan Sekarang”	Website berjalan dengan baik sesuai harapan	Sukses
3	Memilih salah satu produk dari daftar menu, kemudian mengklik “Kembali”	Menekan salah satu menu catering	Berhasil menampilkan <i>output</i> berupa tampilan halaman dari menu catering yang dipilih beserta detail dari menu	Website berjalan dengan baik sesuai harapan	Sukses
4	Memilih salah satu produk dari daftar menu, kemudian	Mengklik <i>button</i> “Kembali”	Berhasil menampilkan <i>output</i> berupa	Website berjalan dengan baik	Sukses

	mengklik <i>button</i> “Kembali”		tampilan halaman daftar menu	sesuai harapan	
5	Menekan <i>Button Shopping Cart</i> untuk memasukkan menu makanan ke dalam Cart	<i>Button Shopping Cart</i>	Berhasil menampilkan <i>output</i> berupa alert “Login Required” bahwa user harus melakukan Login/Registrasi terlebih dahulu sebelum memasukkan menu makanan ke dalam Cart	Website berjalan dengan baik sesuai harapan	Sukses
6	Menekan <i>Button “Pesan Sekarang”</i> untuk memesan menu <i>catering</i>	<i>Button “Pesan Sekarang”</i>	Berhasil menampilkan <i>output</i> berupa alert “Login Required” bahwa user harus melakukan Login/Registrasi terlebih dahulu sebelum memesan	Website berjalan dengan baik sesuai harapan	Sukses

## 2. Pengujian Halaman *Categories* Sebelum *Login*

Tabel 2.

### Hasil Pengujian *Black Box Testing* Halaman *Categories* Sebelum *Login*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan <i>Navbar Categories</i> untuk melihat kategori menu catering	<i>Navbar Categories</i>	Berhasil menampilkan <i>output</i> berupa tampilan halaman	Website berjalan dengan baik	Sukses

			“Pilihan Kategori Menu” beserta <i>button</i> “Lihat Menu”	sesuai harapan	
2	Menekan <i>Button</i> “Lihat Menu” untuk melihat kategori menu makanan yang dipilih	Button “Lihat Menu”	Berhasil menampilkan <i>output</i> berupa tampilan halaman kategori menu yang dipilih beserta harga dan detail paket menu nya	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses

### 3. Pengujian Halaman *Foods* Sebelum Login

Tabel 3.

#### Hasil Pengujian Black Box Testing Halaman *Foods* Sebelum Login

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan <i>Navbar Foods</i> untuk melihat daftar menu catering	<i>Navbar Foods</i>	Berhasil menampilkan <i>output</i> berupa tampilan halaman “Daftar Menu Catering” beserta <i>button</i> “Shopping Cart” dan <i>button</i> “Pesan Sekarang”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
2	Memilih salah satu produk dari daftar menu, kemudian mengklik “Kembali”	Menekan salah satu menu catering	Berhasil menampilkan <i>output</i> berupa tampilan halaman dari menu catering	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses

			yang dipilih beserta detail dari menu		
4	Memilih salah satu produk dari daftar menu, kemudian mengklik <i>button</i> “Kembali”	Mengklik <i>button</i> “Kembali”	Berhasil menampilkan <i>output</i> berupa tampilan halaman daftar menu	Website berjalan dengan baik sesuai harapan	Sukses
5	Menekan <i>Button Shopping Cart</i> untuk memasukkan menu makanan ke dalam Cart	<i>Button Shopping Cart</i>	Berhasil menampilkan <i>output</i> berupa alert “Login Required” bahwa user harus melakukan Login/Registrasi terlebih dahulu sebelum memasukkan menu makanan ke dalam Cart	Website berjalan dengan baik sesuai harapan	Sukses
6	Menekan <i>Button “Pesan Sekarang”</i> untuk memesan menu <i>catering</i>	<i>Button “Pesan Sekarang”</i>	Berhasil menampilkan <i>output</i> berupa alert “Login Required” bahwa user harus melakukan Login/Registrasi terlebih dahulu sebelum memesan	Website berjalan dengan baik sesuai harapan	Sukses

---

#### 4. Pengujian Halaman *About* Sebelum Login

Tabel 4.

Hasil Pengujian *Black Box Testing* Halaman *About* Sebelum Login

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan <i>Navbar About</i> untuk melihat detail dari Warung Nasi Marsel	<i>Navbar About</i>	Berhasil menampilkan <i>output</i> berupa tampilan halaman tentang detail dari Warung Nasi Marsel berupa Lokasi, Sejarah, Kontak Kami, dan <i>Form Saran</i> masukkan dari para pelanggan. Namun jika para pelanggan ingin memberikan saran mereka harus melakukan Login/Registrasi terlebih dahulu	Website berjalan dengan baik sesuai harapan	Sukses

#### 5. Pengujian *Form Sign In User*

Tabel 5.

Hasil Pengujian *Black Box Testing Form Sign In User*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Email dan Password tidak diisi	Email: (kosong)	Sistem akan menolak dan menampilkan	Website berjalan dengan	Sukses

	kemudian klik <i>Button Sign In</i>	<i>Password:</i> (kosong)	alert “Please fill out this field”	baik sesuai harapan	
2	Mengetikkan <i>Email</i> dan <i>Password</i> tidak diisi atau kosong kemudian klik <i>Button Sign In</i>	<i>Email:</i> sindimrsln@gmail.com  <i>Password:</i> (kosong)	Sistem akan menolak dan menampilkan alert “Please fill out this field”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
3	Mengetikkan <i>Email</i> dan <i>Password</i> tidak diisi atau kosong kemudian klik <i>Button Sign In</i>	<i>Email:</i> (kosong)  <i>Password:</i> Sindi1303	Sistem akan menolak dan menampilkan alert “Please fill out this field”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
4	Mengetikkan <i>Email</i> dan/atau <i>Password</i> tidak sesuai, kemudian klik <i>Button Sign In</i>	<i>Email:</i> abc123@gmail.com  <i>Password:</i> admin	Sistem akan menolak dan menampilkan pesan “Password anda salah”	<i>Website</i> berjalan dengan baik akan tetapi tidak sesuai dengan harapan	Gagal
5	Mengetikkan <i>Email</i> dan <i>Password</i> (diisi), kemudian klik <i>Button Sign In</i>	<i>Email:</i> sindimrsln@gmail.com  <i>Password:</i> sindi1303	Sistem menerima akses <i>Sign In</i> dan kemudian menampilkan halaman	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses

			<i>dashboard user</i>		
--	--	--	---------------------------	--	--

## 6. Pengujian Form Register User

Tabel 6.

Hasil Pengujian Black Box Testing Form Register User

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Full Name, Alamat, Nomer Handphone, Email Address, Password, Confirm Password, kemudian klik button <i>Sign Up</i>	Mengklik Sign In tanpa mengisi data atau hanya sebagian data diisi	Sistem akan menolak dan menampilkan alert “please fill out this field”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
2	Full Name, Alamat, Nomer Handphone, Email Address, Password, Confirm Password, kemudian klik button <i>Sign Up</i>	Mengklik <i>Sign Up</i> akan tetapi salah mengetikkan alamat <i>Email</i> (tidak sesuai dengan format penulisan <i>email</i> )	Sistem akan menolak dan menampilkan alert “Email tidak sesuai”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
3	Full Name, Alamat, Nomer Handphone, Email Address, Password, Confirm	Mengklik <i>Sign Up</i> akan tetapi mengetikkan alamat	Sistem akan menolak dan menampilkan alert “The email field”	<i>Website</i> berjalan dengan baik akan tetapi tidak sesuai	Gagal

	Password, kemudian klik <i>button Sign Up</i>	Email lebih dari 25 karakter	must not be greater than 25 characters”	dengan harapan	
4	Full Name, Alamat, Nomer Handphone, Email Address, Password, Confirm Password, kemudian klik <i>button Sign Up</i>	Mengklik <i>Sign Up</i> akan tetapi salah mengetikan <i>Password</i> (tidak sesuai dengan format penulisan password dan/atau Ulangi <i>Password</i> tidak sesuai dengan <i>Password</i> pertama)	Sistem akan menolak dan menampilkan pesan “Password do not match”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
5	Full Name, Alamat, Nomer Handphone, Email Address, Password, Confirm Password, kemudian klik <i>button Sign Up</i>	Mengklik Daftar dengan mengisi data lengkap dan benar	Sistem menerima akses pendaftaran pelanggan dan kemudian akan menampilkan pesan “Register succesfull”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses

---

## 7. Pengujian Form Lupa Password

Tabel 7.

**Hasil Pengujian Black Box Testing Form Lupa Password**

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	<i>Email</i> kemudian <i>klik Button Reset Password</i>	<i>Klik Reset Password</i> dengan mengisi <i>Email</i> yang benar	Sistem menerima akses <i>Reset Password</i> dan kemudian akan menampilkan pesan “Email terdaftar! Anda akan diarahkan untuk mengatur ulang password. Anda akan dialihkan ke halaman reset password dalam beberapa detik...”	Website berjalan dengan baik sesuai harapan	Sukses
2	<i>Password Baru, Kondirmasi Password</i> kemudian <i>klik Button Update Password</i>	Mengklik <i>Update Password</i> akan tetapi salah mengetikan <i>Password</i> (tidak sesuai dengan format penulisan password)	Sistem akan menolak dan menampilkan pesan “Password dan konfirmasi password tidak cocok”	Website berjalan dengan baik sesuai harapan	Sukses

---

		(dan/atau Ulangi Password tidak sesuai dengan Password pertama)			
3	<i>Password Baru,</i> <i>Kondirmasi</i> <i>Password</i> kemudian <i>klik</i> <i>Button Update</i> <i>Password</i>		Sistem menerima akses <i>update</i> <i>password</i> dan kemudian akan menampilkan pesan “ <i>Password</i> berhasil diubah! Anda akan dialihkan ke halaman <i>login</i> dalam beberapa detik...”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses

## 8. Pengujian Halaman Dashboard Setelah *Login*

**Tabel 8.**

**Hasil Pengujian *Black Box Testing* Halaman Dashboard Setelah *Login***

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan Logo Warung Nasi Marsel	<i>Logo</i>	Berhasil menampilkan <i>output</i> berupa tampilan halaman <i>dashboard</i> beserta <i>button</i> pemesanan makanan	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses

2	Menekan Button “Pesan Sekarang” untuk melihat daftar menu catering	<i>Button</i> ”Pesan Sekarang”	Berhasil menampilkan <i>output</i> berupa tampilan menu <i>catering</i> beserta <i>button</i> “ <i>Shopping Cart</i> ” dan <i>button</i> “Pesan Sekarang”	Website berjalan dengan baik sesuai harapan	Sukses
3	Memilih salah satu produk dari daftar menu, kemudian mengklik “Kembali”	Menekan salah satu menu catering	Berhasil menampilkan <i>output</i> berupa tampilan halaman dari menu catering yang dipilih beserta detail dari menu	Website berjalan dengan baik sesuai harapan	Sukses
4	Memilih salah satu produk dari daftar menu, kemudian mengklik <i>button</i> “Kembali”	Mengklik <i>button</i> “Kembali”	Berhasil menampilkan <i>output</i> berupa tampilan halaman daftar menu	Website berjalan dengan baik sesuai harapan	Sukses
5	Menekan <i>Button Shopping Cart</i> untuk memasukkan menu makanan ke dalam Cart	<i>Button</i> <i>Shopping Cart</i>	Berhasil menampilkan <i>output</i> berupa alert “Success” (menu berhasil ditambahkan ke <i>Cart</i> ) serta <i>Button</i> “OK”	Website berjalan dengan baik sesuai harapan	Sukses
6	Menekan <i>Button</i> “Pesan Sekarang” untuk memesan menu <i>catering</i>	<i>Button</i> “Pesan Sekarang”	Berhasil menampilkan <i>output</i> berupa tampilan halaman “ <i>Order Form</i> ”	Website berjalan dengan baik sesuai harapan	Sukses

---

## 9. Pengujian Halaman Order Form

Tabel 9.

**Hasil Pengujian Black Box Testing Order Form**

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	<i>Customer Name, Selected, Address, Phone Number, Quantity, Total Price, kemudian klik button Order Now</i>	Mengklik <i>button Order Now</i> tanpa mengisi data atau hanya sebagian data diisi	Sistem akan menolak dan menampilkan alert “please fill out this field”	<i>Website berjalan dengan baik sesuai harapan</i>	Sukses
2	<i>Customer Name, Selected, Address, Phone Number, Quantity, Total Price, kemudian klik button Order Now</i>	Mengklik <i>Order Now</i> akan tetapi mengisi “Quantity” lebih dari 1 maka “Total Price” menampilkan kalkulasi harga sesuai “quantity”	Sistem akan menampilkan alert “Success” (pesanan berhasil dibuat) serta Button “OK” yang akan melanjutkan ke halaman “Payment Gateway”	<i>Website berjalan dengan baik sesuai harapan</i>	Sukses
3	<i>Customer Name, Selected, Address, Phone Number, Quantity, Total Price, kemudian klik button Back</i>	Mengklik <i>Back</i> tanpa mengisi data atau hanya sebagian data diisi	Sistem akan kembali ke halaman awal yang menampilkan daftar menu	<i>Website berjalan dengan baik sesuai harapan</i>	Sukses

---

---

## 10. Pengujian Halaman *Payment Gateway* (*via QRIS*)

Tabel 10.

**Hasil Pengujian *Black Box Testing Payment Gateway* (*via QRIS*)**

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	<i>Customer Name, Selected, Address, Phone Number, Quantity, Total Price</i> , kemudian klik button <i>Order Now</i>	Mengklik button <i>Order Now</i>	Sistem akan menampilkan tampilan halaman “Pembayaran Qris” beserta <i>Detail Pesanan</i> (Nama, Alamat, No. Telepon, Menu, Total Pembayaran)	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
2	Scan <i>QR Code</i> untuk pembayaran	Scan QR Code menggunakan Mobile Banking dan E-Wallet lainnya	Kemudahan pada saat melakukan transaksi pembayaran	<i>Payment Gateway</i> berjalan dengan baik sesuai harapan	Sukses
3	Klik Button “Selesai Pembayaran”	Mengklik Button “Selesai Pembayaran”	Sistem akan menampilkan alert “Success” (pembayaran berhasil diselesaikan) serta Button “OK” yang akan kembali ke tampilan awal halaman dashboard	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses

---

---

## 11. Pengujian Halaman *Categories* Setelah Login

Tabel 11.

Hasil Pengujian *Black Box Testing Categories* Setelah Login

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan <i>Navbar Categories</i> untuk melihat kategori menu catering	<i>Navbar Categories</i>	Berhasil menampilkan <i>output</i> berupa tampilan halaman “Pilihan Kategori Menu” beserta <i>button</i> “Lihat Menu”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
2	Menekan <i>Button</i> “Lihat Menu” untuk melihat kategori menu makanan yang dipilih	<i>Button</i> “Lihat Menu”	Berhasil menampilkan <i>output</i> berupa tampilan halaman kategori menu yang dipilih beserta harga dan detail paket menu nya	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses

## 12. Pengujian Halaman *Foods* Setelah Login

Tabel 11.

Hasil Pengujian *Black Box Testing Foods* Setelah Login

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan <i>Navbar Foods</i> untuk melihat daftar menu catering	<i>Navbar Foods</i>	Berhasil menampilkan <i>output</i> berupa tampilan halaman “Daftar Menu”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses

---

			<i>Catering</i> ” beserta <i>button</i> “ <i>Shopping Cart</i> ” dan <i>button</i> “ <i>Pesan Sekarang</i> ”		
2	Memilih salah satu produk dari daftar menu, kemudian mengklik “Kembali”	Menekan salah satu menu catering	Berhasil menampilkan <i>output</i> berupa tampilan halaman dari menu catering yang dipilih beserta detail dari menu	Website berjalan dengan baik sesuai harapan	Sukses
3	Memilih salah satu produk dari daftar menu, kemudian mengklik <i>button</i> “Kembali”	Mengklik <i>button</i> “Kembali”	Berhasil menampilkan <i>output</i> berupa tampilan halaman daftar menu	Website berjalan dengan baik sesuai harapan	Sukses
4	Menekan <i>Button Shopping Cart</i> untuk memasukkan menu makanan ke dalam Cart	<i>Button Shopping Cart</i>	Berhasil menampilkan <i>output</i> berupa alert “Success” (menu berhasil ditambahkan ke <i>Cart</i> ) serta <i>Button</i> “OK”	Website berjalan dengan baik sesuai harapan	Sukses
5	Menekan <i>Button</i> “ <i>Pesan Sekarang</i> ” untuk memesan menu <i>catering</i>	<i>Button</i> “ <i>Pesan Sekarang</i> ”	Berhasil menampilkan <i>output</i> berupa tampilan halaman “ <i>Order Form</i> ”	Website berjalan dengan baik sesuai harapan	Sukses

---

### 13. Pengujian Halaman Order Form

Tabel 13.

**Hasil Pengujian Black Box Testing Order Form**

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	<i>Customer Name, Selected, Address, Phone Number, Quantity, Total Price, kemudian klik button Order Now</i>	Mengklik <i>button Order Now</i> tanpa mengisi data atau hanya sebagian data diisi	Sistem akan menolak dan menampilkan alert “please fill out this field”	<i>Website berjalan dengan baik sesuai harapan</i>	Sukses
2	<i>Customer Name, Selected, Address, Phone Number, Quantity, Total Price, kemudian klik button Order Now</i>	Mengklik <i>Order Now</i> akan tetapi mengisi “Quantity” lebih dari 1 atau lebih banyak maka “Total Price” menampilkan kalkulasi harga sesuai “quantity”	Sistem akan menampilkan alert “Success” (pesanan berhasil dibuat) serta Button “OK” yang akan melanjutkan ke halaman “Payment Gateway”	<i>Website berjalan dengan baik sesuai harapan</i>	Sukses
3	<i>Customer Name, Selected, Address, Phone Number, Quantity, Total Price, kemudian klik button Back</i>	Mengklik <i>Back</i> tanpa mengisi data atau hanya sebagian data diisi	Sistem akan kembali ke halaman awal yang menampilkan daftar menu	<i>Website berjalan dengan baik sesuai harapan</i>	Sukses

---

---

#### 14. Pengujian Halaman *Payment Gateway* (*via QRIS*)

Tabel 14.

Hasil Pengujian *Black Box Testing Payment Gateway* (*via QRIS*)

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	<i>Customer Name, Selected, Address, Phone Number, Quantity, Total Price</i> , kemudian klik button <i>Order Now</i>	Mengklik button <i>Order Now</i>	Sistem akan menampilkan tampilan halaman “Pembayaran Qris” beserta <i>Detail Pesanan</i> (Nama, Alamat, No. Telepon, Menu, Total Pembayaran)	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
2	Scan <i>QR Code</i> untuk pembayaran	Scan QR Code menggunakan Mobile Banking dan E-Wallet lainnya	Kemudahan pada saat melakukan transaksi pembayaran	<i>Payment Gateway</i> berjalan dengan baik sesuai harapan	Sukses
3	Klik Button “Selesai Pembayaran”	Mengklik Button “Selesai Pembayaran”	Sistem akan menampilkan alert “Success” (pembayaran berhasil diselesaikan) serta Button “OK” yang akan kembali ke tampilan awal halaman dashboard	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses

---

---

## 15. Pengujian Halaman *About* Setelah *Login*

Tabel 15.

### Hasil Pengujian *Black Box Testing* Halaman *About* Setelah *Login*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan <i>Navbar About</i> untuk melihat detail dari Warung Nasi Marsel	<i>Navbar About</i>	Berhasil menampilkan <i>output</i> berupa tampilan halaman tentang detail dari Warung Nasi Marsel berupa Lokasi, Sejarah, Kontak Kami, dan <i>Form Saran</i> masukkan dari para pelanggan	Website berjalan dengan baik sesuai harapan	Sukses

## 16. Pengujian Halaman *Shopping Cart*

Tabel 16.

### Hasil Pengujian *Black Box Testing* Halaman *Shopping Cart*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan <i>Button Shopping Cart</i> untuk mengarahkan pengguna melihat menu di keranjang sebelum melakukan “ <i>Check Out</i> ”	<i>Button Shopping Cart</i>	Berhasil menampilkan <i>output</i> berupa tampilan halaman keranjang yang berisi Harga Satuan, Jumlah, dan Harga Total.	Website berjalan dengan baik sesuai harapan	Sukses

---

2	Menekan <i>Button tambah dan kurang</i> pada <i>Shopping Cart</i> sebelum melakukan “ <i>Check Out</i> ”	<i>Button tambah dan kurang</i>	Berhasil menampilkan <i>output</i> berupa <i>button</i> untuk menambah dan mengurangi jumlah pesanan, serta kalkulasi dari jumlah yang ingin dipesan	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
3	Menekan <i>button hapus item</i> pada <i>Shopping Cart</i> sebelum melakukan “ <i>Check Out</i> ”	<i>Button hapus</i>	Berhasil menampilkan <i>output</i> tampilan berupa pesan alert “Success” (Item dihapus dari <i>cart</i> ) serta <i>Button</i> “OK” yang akan kembali ke tampilan awal halaman keranjang	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
4	Menekan <i>button “Check Out”</i> untuk menyelesaikan pesanan	<i>Button “Check Out”</i>	Berhasil menampilkan <i>output</i> berupa tampilan halaman “ <i>Order Form for Cart</i> ”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses

---

---

## 17. Pengujian Halaman *Order Form for Cart*

Tabel 17.

**Hasil Pengujian Black Box Testing Halaman *Order Form for Cart***

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan button “Check Out” untuk menyelesaikan pesanan lalu menampilkan halaman “Order Form for Cart”	<i>Order Summary</i>	Berhasil menampilkan <i>output</i> berupa ringkasan pesanan beserta kalkulasi dari jumlah pesanan serta jumlah pesanannya	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
2	<i>Customer Name, Shipping Address, Phone Number,</i> kemudian klik button <i>Place Order</i>	Mengklik button <i>Place Order</i> tanpa mengisi data atau hanya sebagian data diisi	Sistem akan menolak dan menampilkan alert “please fill out this field”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
3	<i>Customer Name, Shipping Address, Phone Number,</i> kemudian klik button <i>Place Order</i>	Mengklik button <i>Place Order</i>	Sistem akan menampilkan alert “Success” (pesanan berhasil dibuat) serta Button “OK” yang akan melanjutkan ke halaman “Payment Gateway”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
4	<i>Customer Name, Selected, Address, Phone Number,</i>	Mengklik Back tanpa mengisi data	Sistem akan kembali ke	<i>Website</i> berjalan dengan baik	Sukses

---

---

<i>Quantity, Total Price, kemudian klik button Back</i>	atau hanya sebagian data diisi	tampilan halaman keranjang	sesuai harapan	
---	--------------------------------	----------------------------	----------------	--

## 18. Pengujian Halaman *Payment Gateway (via QRIS)*

Tabel 18.

Hasil Pengujian *Black Box Testing Payment Gateway (via QRIS)*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	<i>Customer Name, Selected, Address, Phone Number, Quantity, Total Price, kemudian klik button Order Now</i>	Mengklik <i>button Order Now</i>	Sistem akan menampilkan tampilan halaman “Pembayaran Qris” beserta <i>Detail Pesanan</i> (Nama, Alamat, No. Telepon, Menu, Total Pembayaran)	<i>Website berjalan dengan baik sesuai harapan</i>	Sukses
2	<i>Scan QR Code untuk pembayaran</i>	Scan QR Code menggunakan Mobile Banking dan E-Wallet lainnya	Kemudahan pada saat melakukan transaksi pembayaran	<i>Payment Gateway berjalan dengan baik sesuai harapan</i>	Sukses
3	<i>Klik Button “Selesai Pembayaran”</i>	Mengklik <i>Button “Selesai Pembayaran”</i>	Sistem akan menampilkan alert “Success” (pembayaran berhasil diselesaikan) serta Button “OK” yang	<i>Website berjalan dengan baik sesuai harapan</i>	Sukses

---

---

			akan kembali ke tampilan awal halaman dashboard		
--	--	--	---	--	--

## 19. Pengujian Nama Pengguna Account

Tabel 19.

Hasil Pengujian *Black Box Testing* Nama Pengguna Account

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	<i>Klik Profile</i>	Mengklik <i>Profile</i>	Berhasil menampilkan <i>output</i> berupa halaman <i>profile</i> pengguna Account berupa Name, Email, Alamat, Nomer Handphone. Kemudian <i>button</i> “Edit Profile” untuk memperbarui data <i>profile</i> pelanggan	Website berjalan dengan baik sesuai harapan	Sukses
2	<i>Klik Log Out</i>	Mengklik <i>Logout</i>	Sistem akan kembali ke tampilan awal halaman <i>dashboard user</i> sebelum melakukan <i>Log In</i>	Website berjalan dengan baik sesuai harapan	Sukses

---

---

## 20. Pengujian *Edit Profile*

Tabel 20.

Hasil Pengujian *Black Box Testing Edit Profile*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	<i>Klik Profile</i> kemudian <i>klik Button “Edit Profile”</i> untuk memperbarui data <i>profile</i> pelanggan	Mengklik <i>Edit Profile</i>	Berhasil menampilkan <i>output</i> berupa halaman <i>edit profile</i>	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
2	<i>Full Name, Alamat, Nomer Handphone, Email Address,</i> kemudian <i>klik button Update Profile</i>	Mengklik <i>button Place Order</i> tanpa mengisi data atau hanya sebagian data diisi	Sistem akan menolak dan menampilkan alert “please fill out this field”	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses
3	<i>Full Name, Alamat, Nomer Handphone, Email Address,</i> kemudian <i>klik button Update Profile</i>	Mengklik <i>button Update Profile</i>	Sistem akan kembali ke tampilan halaman <i>profile</i> serta data <i>profile</i> pelanggan berhasil di perbarui	<i>Website</i> berjalan dengan baik sesuai harapan	Sukses

# Lampiran User Acceptance Test

Dokumen User Acceptance Testing		
Nama Proyek	:	Website Pemesanan Catering Warung Nasi Marsel
Studi Kasus	:	Warung Nasi Marsel
Manajer Proyek	:	Rolanda Evan Maximiliano

Proses Pengujian					
No	Use Case	Hasil Uji [Berhasil Gagal]	Nama Penguji *	Tanggal Pengujian	Catatan Penguji
1	Usecase Uji : Login Deskripsi : Melakukan verifikasi terhadap pengguna yang terdaftar dalam sistem  <b>Kasus Pengujian</b> Username : alvesdh Password : delahoya98  Hasil yang diharapkan: - Jika login berhasil akan masuk kedalam halaman dashboard user - Jika login tidak berhasil tidak akan masuk kedalam halaman dashboard, dan menampilkan pesan kesalahan melalui alert	Berhasil	Alves	15 Januari 2025	
2	Usecase Uji : Menambahkan menu ke keranjang Deskripsi : Melakukan verifikasi pengelolaan menu catering  <b>Kasus Pengujian</b> Nama Kategorie Makanan : Tampil Otomatis sesuai Kategorie menu makanan yang dipilih Harga Satuan : Rp 25.000 Jumlah : 2 Harga Total : Rp 50.000	Berhasil	Zelits	16 Januari 2025	

	Hasil yang diharapkan: - Data menu disimpan dan tampil di keranjang menu jika berhasil				
3	Usecase Uji : Membuat Pesanan/Check Out  Deskripsi : Melakukan verifikasi proses pembuatan Pesanan oleh pelanggan  Kasus Pengujian  Customer Name : Zelits  Shipping Address : Jl. Ciomas gang jeruk No.18, RT.04 RW.11  Phone Number : 6285159595299	Berhasil	Alves	15 Januari 2025	
	Hasil yang diharapkan: - Jika data berhasil dimasukkan, akan tampil pesan “Pesanan Berhasil Dibuat” dan pesanan akan tampil di daftar pesanan. - Jika data tidak berhasil dimasukkan (salah input/tidak semua diinputkan), maka akan tampil pesan “Pesanan Tidak Berhasil Dibuat” dan pengguna kembali ke form pembuatan pesanan. - Pesanan disimpan dan dapat ditampilkan di daftar pesanan	Berhasil	Zelits	16 Januari 2020	

Proses Pengujian					
No	Use Case	Hasil Uji [Berhasil Gagal]	Nama Penguji *	Tanggal Pengujian	Catatan Penguji
4	Usecase Uji : Menampilkan Daftar Pesanan  Deskripsi : Melakukan verifikasi bahwa daftar Pesanan dapat ditampilkan dengan Data yang sesuai  Kasus Pengujian  ID Pesanan : Tampil Otomatis sesuai hasil generate sistem  Tanggal Pesanan : Tampil otomatis sesuai tanggal sistem  Nama Pelanggan : Tampil sesuai data pelanggan yang memesan  Detail Pesanan : Tampil sesuai dengan menu dan jumlah yang dipesan pelanggan  Status Pesanan : Dipilih melalui drop-down (Baru, Sedang Diproses, Selesai)	Berhasil	Alves	15 Januari 2025	

	<p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Daftar pesanan menampilkan nama pelanggan, alamat pengiriman, nomor telepon, detail pesanan, total harga, dan status pesanan</li> <li>- Jika data berhasil dimasukkan, akan tampil pesan “Data Pesanan Berhasil Ditampilkan” dan pesanan terlihat pada daftar pesanan</li> <li>- Jika data tidak berhasil ditampilkan (data tidak valid), akan tampil pesan “Data Pesanan Tidak Berhasil Ditampilkan” dan pengguna kembali ke dashboard daftar pesanan</li> </ul>				
5	<p>UseCase Uji : Pembayaran</p> <p>Deskripsi : Melakukan verifikasi proses pembayaran</p> <p>Kasus Pengujian</p> <p>ID Pesanan : Input sesuai pesanan pelanggan</p> <p>Metode Pembayaran : Transfer Bank / Tunai</p>	Berhasil	Alves	15 Januari 2025	
	<p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Jika data pembayaran berhasil dimasukkan, akan tampil pesan “Pembayaran Berhasil”, status pesanan berubah menjadi “Dibayar”, dan detail pembayaran terlihat pada daftar pembayaran</li> <li>- Jika data pembayaran tidak berhasil dimasukkan (salah input/tidak semua diinputkan), maka akan tampil pesan “Pembayaran Tidak Berhasil” dan pengguna kembali ke form pembayaran.</li> <li>- Menampilkan konfirmasi pembayaran berhasil dan status pesanan menjadi “Dibayar”</li> </ul>				
6	<p>UseCase Uji : Menampilkan laporan penjualan</p> <p>Deskripsi : Melakukan verifikasi laporan penjualan Harian pada sistem</p> <p>Kasus Pengujian</p> <p>Periode Laporan : Input sesuai tanggal atau hari yang diinginkan</p> <p>Hasil yang diharapkan:</p> <ul style="list-style-type: none"> <li>- Menampilkan laporan penjualan harian dengan data pesanan yang sudah dibayar, termasuk detail seperti</li> </ul>	Berhasil	Alves	15 Januari 2025	

nama pelanggan, menu yang dipesan, jumlah, total harga, dan metode pembayaran  - Jika laporan berhasil dihasilkan, akan tampil pesan “Laporan Penjualan Harian Berhasil Ditampilkan” - Jika laporan tidak berhasil dihasilkan, akan tampil pesan “Laporan Penjualan Tidak Berhasil Ditampilkan” dan pengguna kembali ke halaman dashboard laporan				
--	--	--	--	--

---

## BERITA ACARA SERAH TERIMA WEBSITE

Berdasarkan perjanjian kerja, maka pada hari Jum'at tanggal 24 Januari 2025 dilakukan serah terima Website Pemesanan Catering Warung Nasi Marsel oleh kelompok 7 terhadap usaha catering warung nasi marsel. Adapun pihak yang terlibat dalam serah terima ini antara lain :

- 1. PIHAK PERTAMA** : Kelompok 7 selaku Pengembang Website Pemesanan Catering Warung Nasi Marsel
- 2. PIHAK KEDUA** : Ibu Eti Sumiati selaku pemilik usaha Warung Nasi Marsel

Dengan ini menyatakan bahwa **PIHAK PERTAMA** menyerahkan sebuah platform Website Pemesanan Catering Warung Nasi Marsel terhadap **PIHAK KEDUA** dalam kondisi :

- a. Berfungsi dengan baik sesuai dengan spesifikasi yang disepakati dalam dokumen perencanaan proyek.
- b. Mendukung kebutuhan operasional usaha Warung Nasi Marsel.
- c. Telah melalui proses uji coba dan perbaikan sesuai dengan laporan yang terdokumentasi.
- d. Seluruh infrastruktur pendukung telah dikonfigurasi sesuai kebutuhan operasional.
- e. Website telah terhosting maka pemeliharaan website selanjutnya akan ditanggung jawab oleh **PIHAK PERTAMA**. Maka **PIHAK KEDUA** sudah bisa menerima dan mengoperasikan website Warung Nasi Marsel melanjutkan biaya maintenance untuk jangka waktu berkepanjangan setelah serah terima ini. Hal ini karena **PIHAK PERTAMA** berkomitmen untuk menjaga kestabilan dan keamanan website selama periode tersebut, maka mengenai informasi spesifikasi hosting website akan dicantumkan dalam **Lampiran Berita Acara**.

Demikian berita acara serah terima ini dibuat dengan sebenarnya untuk dapat digunakan sebagaimana mestinya. Kedua belah pihak bertanggung jawab atas isi dokumen ini dan menandatanganinya dalam keadaan sadar tanpa paksaan dari pihak mana pun.

Pihak Pertama



( Kelompok 7 )

Pihak Kedua



( Pemilik Warung Nasi Marsel )

---

## **Lampiran Berita Acara**

### **Spesifikasi Hosting Website :**

#### **1 Akses URL :**

- Website: <https://warung-nasi-marsel.store/>

#### **2 Hosting & Domain:**

- Rumahweb :
  - Harga: Rp 110.000/bulan

---

## Dokumentasi serah terima laporan dan aplikasi



---

# **Buku Petunjuk Penggunaan Website**

## **(User Manual Book)**

### **Website Warung Nasi Marsel**



#### **Dibuat Oleh Kelompok 7 dengan Anggota:**

- Rolanda Evan M. (11121163) – Project Manager
- Fakih Syamaidzar (10121418) – System Analyst
- Rafif Atha Naufal (11121404) – UI/UX Designer
- Muhamad Arya Pratama (10121739) – Programmer (Front-End/Back-End)
- Sindi Marselina (11121223) – Tester (Quality Assurance)

**Universitas Gunadarma**  
**Fakultas Ilmu Komputer dan Teknologi Informasi**  
**Jurusan Sistem Informasi**

---

---

## **GAMBARAN UMUM WEBSITE**

Sistem Pemesanan Catering Warung Nasi Marsel ini adalah sebuah platform berbasis web yang memungkinkan pengguna untuk memesan produk catering secara online dengan berbagai pilihan yang dapat disesuaikan oleh pengguna, jangkauan layanan yang lebih luas, dan meningkatkan pengalaman pelanggan dengan sistem pemesanan yang lebih efektif dan efisien. Perangkat lunak dibuat untuk membuat catering warung nasi marsel agar lebih dikenal secara luas. Serta membangun platform digital inovatif yang memudahkan pelanggan dalam memesan makanan lezat, sehat, dan berkualitas dengan pelayanan cepat serta transparan, sehingga menjadi pilihan utama dalam memenuhi kebutuhan catering masyarakat.

Manfaat pembuatan perangkat lunak ini, antara lain:

- Pelanggan dapat memesan catering dimana saja dan kapan saja.
- Meningkatkan proses pemesanan dan pembayaran secara digital.
- Dengan adanya perangkat lunak ini dapat memperluas jangkauan pasar, memungkinkan orang-orang dari berbagai daerah dapat melihat dan memesan layanan catering.

Fungsi yang terdapat di perangkat lunak antara lain:

- Perangkat lunak dapat menampilkan berbagai menu dan catering, lengkap dengan harga, deskripsi, dan gambar untuk memudahkan pelanggan memilih sesuai kebutuhan.
- Perangkat lunak memfasilitasi pelanggan untuk melakukan pemesanan secara online melalui form pemesanan, sehingga proses order makanan lebih praktis.
- Memberikan informasi berisi status pemesanan dalam bentuk pesan dengan otomatis melalui fitur whatsapp API.

---

## **KEBUTUHAN HARDWARE DAN SOFTWARE**

### **Hardware**

Hardware yang digunakan untuk mengakses website ini seperti:

Laptop  
Processor 2.40GHz, RAM 4 GB

### **Software**

Software yang digunakan untuk mengakses website ini seperti:

Google Chrome, Microsoft Edge, Mozilla Firefox

---

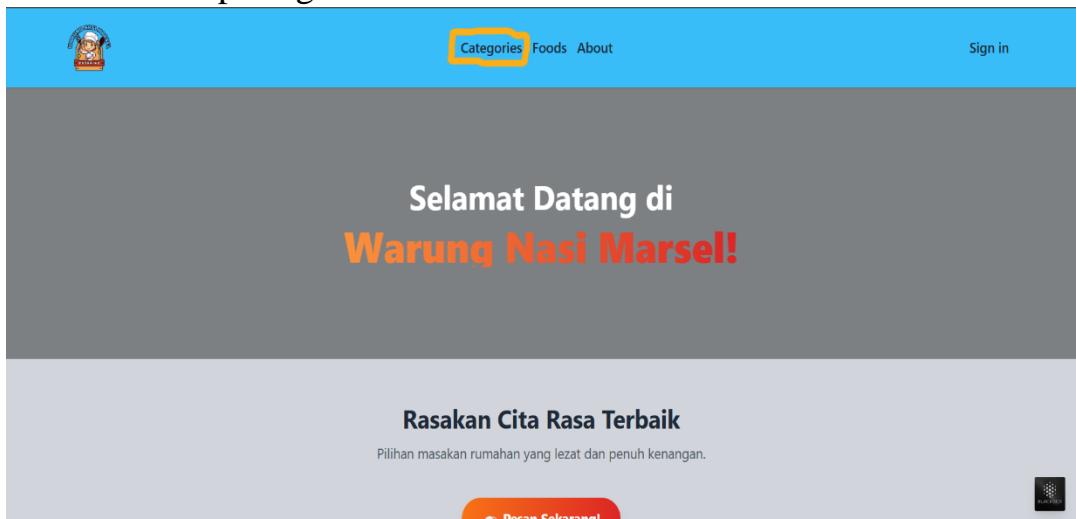
## PETUNJUK PENGGUNAAN

Untuk user yang ingin membeli atau memesan makanan di Warung Nasi Marsel lakukan langkah berikut

1. Buka browser yang anda gunakan, kemudian ketikkan alamat url berikut <https://warung-nasi-marsel.store/> .
2. Setelah mengetikkan alamat url di atas maka muncul dashboard website seperti gambar di bawah.



3. Untuk user/pelanggan bisa melihat lihat terlebih dahulu kategori makanan yang tersedia di Warung Nasi Marsel dengan mengklik Categories atau sesuai seperti gambar di bawah



Maka akan muncul kategori makanan yang tersedia seperti gambar di bawah.

## Menu Pilihan Warung Nasi Marsel

Nikmati Berbagai Pilihan Menu Istimewa Untuk Setiap Momen Spesial Anda

### Pilihan Kategori Menu

Temukan berbagai pilihan menu lezat untuk acara spesial Anda. Dari paket prasmanan hingga nasi kotak, kami siap melayani dengan cita rasa yang autentik.



Nasi Kuning



Nasi Kotak



Nasi Besek

4. Jika user/pelanggan ingin melihat menu makanan yang sesuai dengan kategori makanan dapat mengklik card kategori seperti di bawah ini.

### Pilihan Kategori Menu

Temukan berbagai pilihan menu lezat untuk acara spesial Anda. Dari paket prasmanan hingga nasi kotak, kami siap melayani dengan cita rasa yang autentik.



Nasi Kuning

Lihat Menu



Nasi Kotak

Lihat Menu



Nasi Besek

Lihat Menu



Maka akan muncul makanan yang sesuai seperti gambar di bawah

The screenshot shows a website with a blue header bar. On the left is a logo of a cartoon character. In the center are links for 'Categories', 'Foods', and 'About'. On the right is a 'Sign in' button. Below the header, the title 'Jelajahi Menu di Kategori: Nasi Kuning' is displayed in bold green text. A sub-instruction below it reads: 'Temukan hidangan lezat dari kategori Nasi Kuning. Pilih makanan favorit Anda dan tambahkan ke keranjang untuk dinikmati!' (Find delicious dishes from the Nasi Kuning category. Choose your favorite meal and add it to the cart to enjoy!). Three menu items are shown in a grid:

- Paket Nasi Kuning Tumpeng Mini**: Rp 30.000. Description: Nasi Kuning, Ayam Goreng, Telur Balado, Sambal Goreng Ati Ampela, Perkedel, Telur, Tempe Orek, Mie Goreng, Sambal, Lalapan. (Untuk 2 Orang)
- Paket Nasi Kuning Tumpeng Middle**: Rp 60.000. Description: Nasi Kuning, Ayam Goreng, Bihun, Tempe Orek, Gorengan, Sambal Goreng Ati Ampela Pete, Perkedel, Gorengan, Sambal, Lalapan. (Untuk 4 Orang)
- Paket Nasi Kuning Tumpeng Large**: Rp 150.000. Description: Nasi Kuning, Ayam, Telur, Urap, Balado Kentang, Tempe Cirek, Perkedel, Gorengan, Sambal, Lalapan. (Untuk 7 Orang)

At the bottom of the page are social media sharing icons for Facebook, Instagram, and WhatsApp.

5. Untuk melihat menu makanan yang ada di Warung Nasi Marsel user/pelanggan dapat mengklk tulisan foods yang ada di navbar website.

The screenshot shows the homepage of the 'Warung Nasi Marsel' website. The header is identical to the previous screenshot, featuring a logo, 'Categories', 'Foods' (which is highlighted in yellow), and 'About'. The main content area has a dark grey background. In the center, the text 'Selamat Datang di Warung Nasi Marsel!' is displayed in large, bold, orange and white letters. Below this, the text 'Rasakan Cita Rasa Terbaik' is shown in a smaller white font, followed by the subtitle 'Pilihan masakan rumahan yang lezat dan penuh kenangan.' A prominent orange button labeled 'Pesan Sekarang!' (Order Now!) is centered at the bottom of this section.

Akan tampil berbagai menu yang tersedia di Warung Nasi Marsel  
Selamat datang di Catering Warung Nasi Marsel

This screenshot shows the 'Daftar Menu' (Menu List) page of the website. The top part is identical to the homepage. The main content area features a heading 'Daftar Menu' in bold green text. Below it, four menu items are listed in a grid:

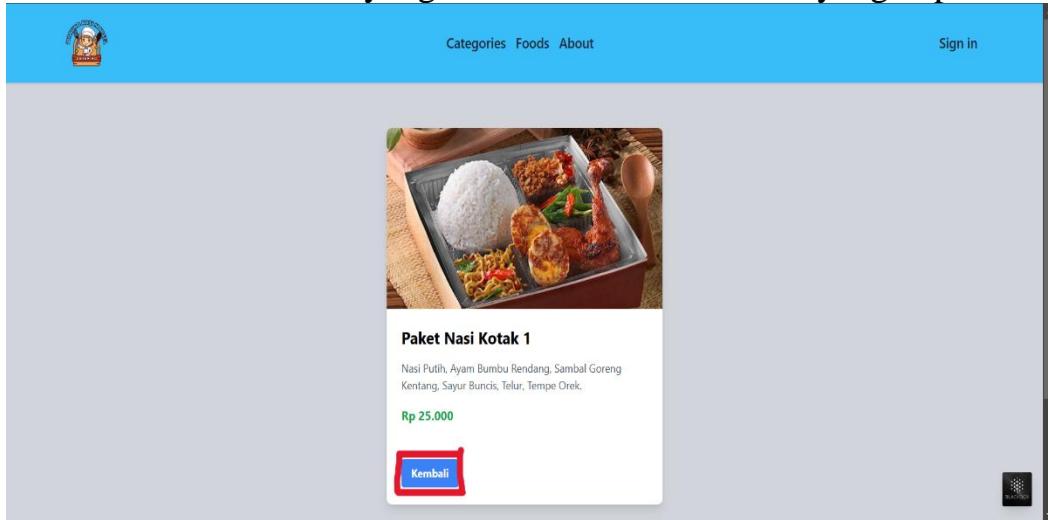
- Paket Nasi Kotak 1**: Rp 25.000. Includes images of rice, chicken, and various side dishes.
- Paket Nasi Kotak 2**: Rp 30.000. Includes images of rice, meat, and vegetables.
- Paket Nasi Kotak 3**: Rp 35.000. Includes images of rice, meat, and vegetables.
- Paket Nasi Kotak 4**: Rp 40.000. Includes images of rice, meat, and vegetables.

Each menu item has a small shopping cart icon and a 'Pesan Sekarang' (Order Now) button. Below the main grid, there are three smaller images of different meals: a bento box, a dish with rice and vegetables, and a dish with meat and rice.

- 
6. Untuk melihat deskripsi makanan, user/pelanggan dapat mengklik salah satu card menu makanan



Akan tampil deskripsi lengkap dari menu makanan yang dipilih, untuk kembali melihat menu makanan yang lain ada tombol kembali yang dapat diklik.



7. User/pelanggan dapat membeli langsung makanan atau memasukkannya ke keranjang terlebih dahulu dengan mengklik button pesan sekarang dan button keranjang, tetapi akan muncul alert agar user harus login terlebih dahulu

**Selamat datang di Catering Warung Nasi Marsel**

Kami menyediakan berbagai menu lezat dengan harga terjangkau. Jelajahi menu kami dan temukan hidangan favorit Anda.

## Daftar Menu

Paket Nasi Kotak 1	Paket Nasi Kotak 2	Paket Nasi Kotak 3	Paket Nasi Kotak 4
Rp 25.000	Rp 30.000	Rp 35.000	Rp 40.000

**Selamat datang di Catering Warung Nasi Marsel**

Kami menyediakan berbagai menu lezat dengan harga terjangkau. Jelajahi menu kami dan temukan hidangan favorit Anda.

## Daftar Menu

**Login Required**

Anda harus login terlebih dahulu untuk memesan.

**OK**

8. Untuk Halaman About berisi lokasi, sejarah, kontak, dan kolom saran. Untuk mengakses halaman about dapat mengklik tulisan About di navbar website

The screenshot shows two versions of a website for "Warung Nasi Marsel". The top version is a mobile view where the "About" tab is highlighted in yellow. The bottom version is a desktop view of the same page.

**Mobile View (Top):**

- Header: Categories, Foods, **About**, Sign in.
- Main Content: "Selamat Datang di Warung Nasi Marsel!"
- Text: "Rasakan Cita Rasa Terbaik", "Pilihan masakan rumahan yang lezat dan penuh kenangan."
- Call-to-action: "Pesan Sekarang!"
- Footer: https://warung-nasi-marsel.store, Sign in.

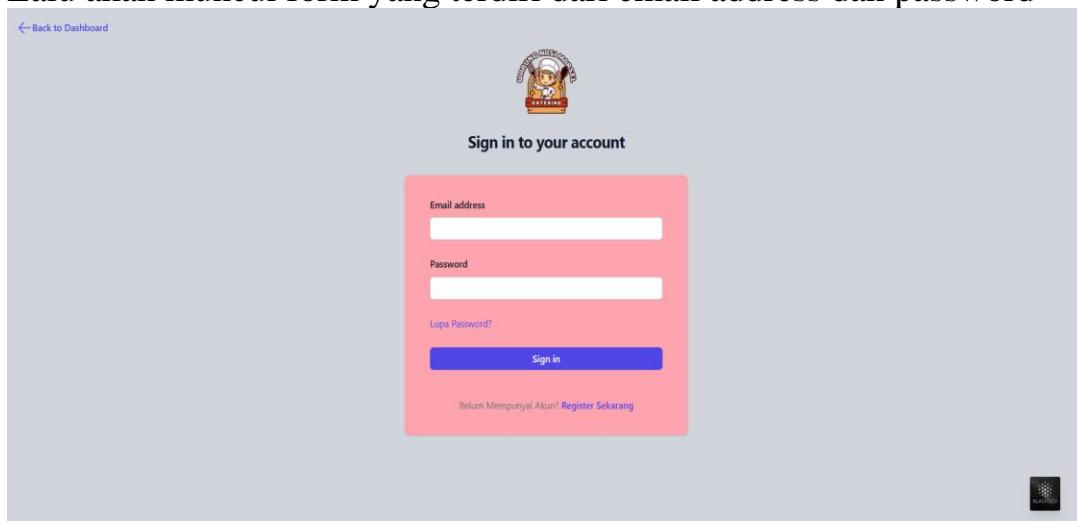
**Desktop View (Bottom):**

- Header: Categories, Foods, About, Sign in.
- Main Content:
  - Warung Nasi Marsel**
  - Lokasi**: Map showing the location of Warung Nasi Marsel on Jl. Raya Lenteng Agung Gang Bukitkrama No.35, RT.5/RW.4, Srengseng Sawah, Kec. Jagakarsa, Jakarta Selatan 12645, Daerah Khusus Ibukota Jakarta 12645. Rating 4.8 stars from 12 reviews.
  - Sejarah**: Warung Nasi Marsel adalah tempat makan yang berasal dari Kuningan Jawa Barat, menawarkan berbagai macam hidangan khas Sunda. Kami menggunakan bahan-bahan segar dan berkualitas untuk menyajikan makanan yang lezat dan bergizi.
  - Kontak Kami**:
    - Alamat: Jl. Buku Dikrama RT.05 RW.04 NO.35
    - Telepon: +62 812-9591-7717
    - Email: warungnasimarsel@gmail.com
  - Berikan Saran Anda**:
    - Silakan login terlebih dahulu untuk memberikan saran.
    - Pesan Saran:  
Silakan login terlebih dahulu untuk memberikan saran

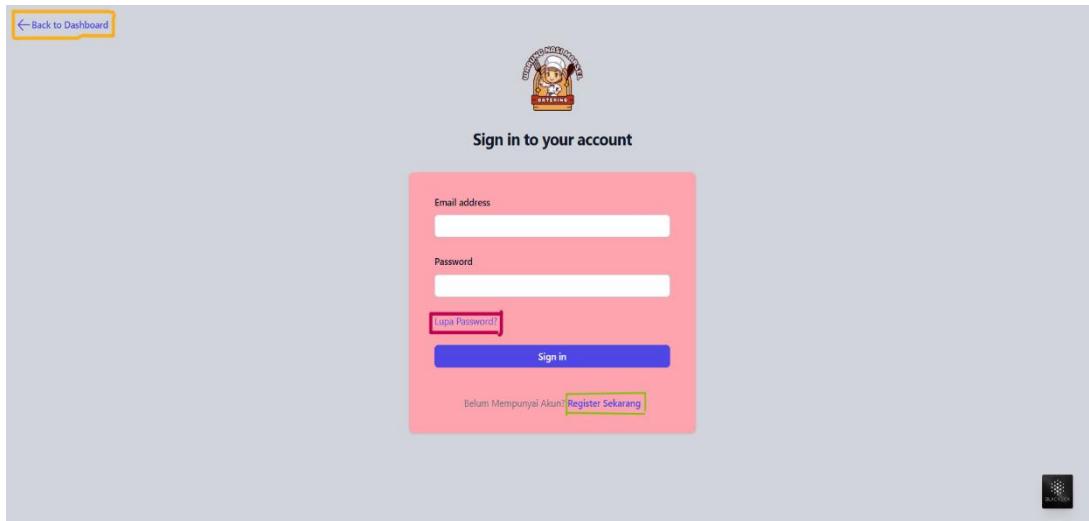
- 
9. Untuk user/pelanggan yang ingin melakukan pembelian bisa melakukan login terlebih dahulu dengan cara mengklik tulisan Sign in



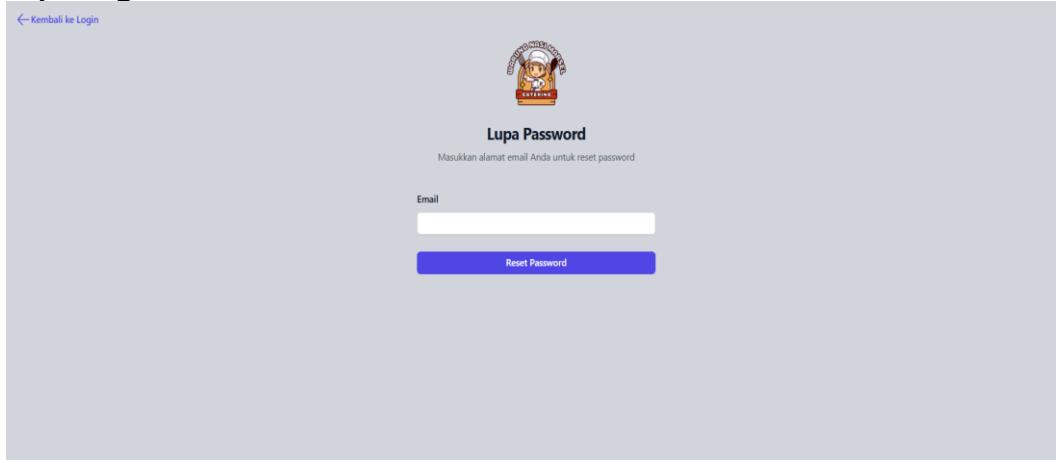
Lalu akan muncul form yang terdiri dari email address dan password



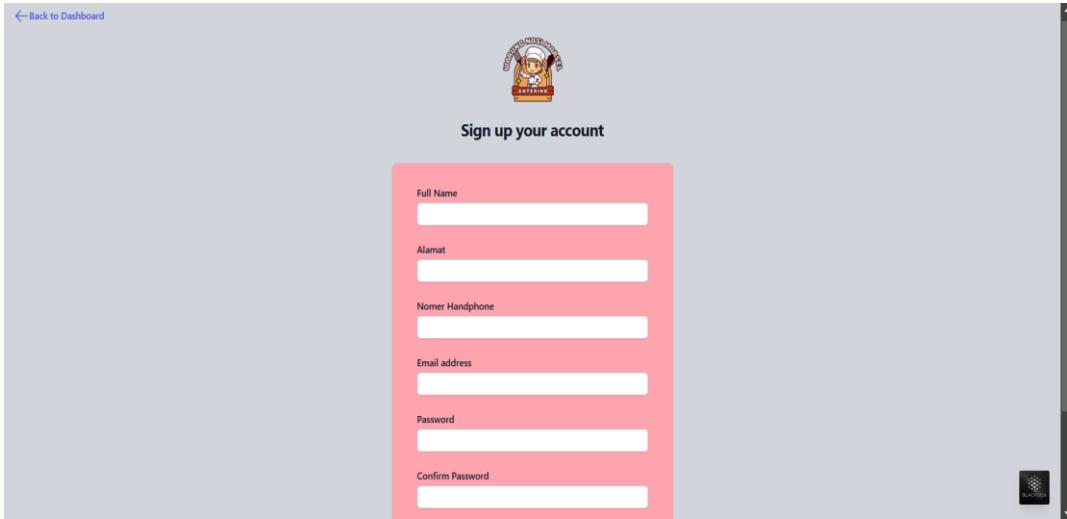
Di dalam halaman login terdapat tanda panah “Back to Dashboard”, selain itu juga terdapat tulisan “Lupa Password” , dan tulisan “Register Sekarang”.



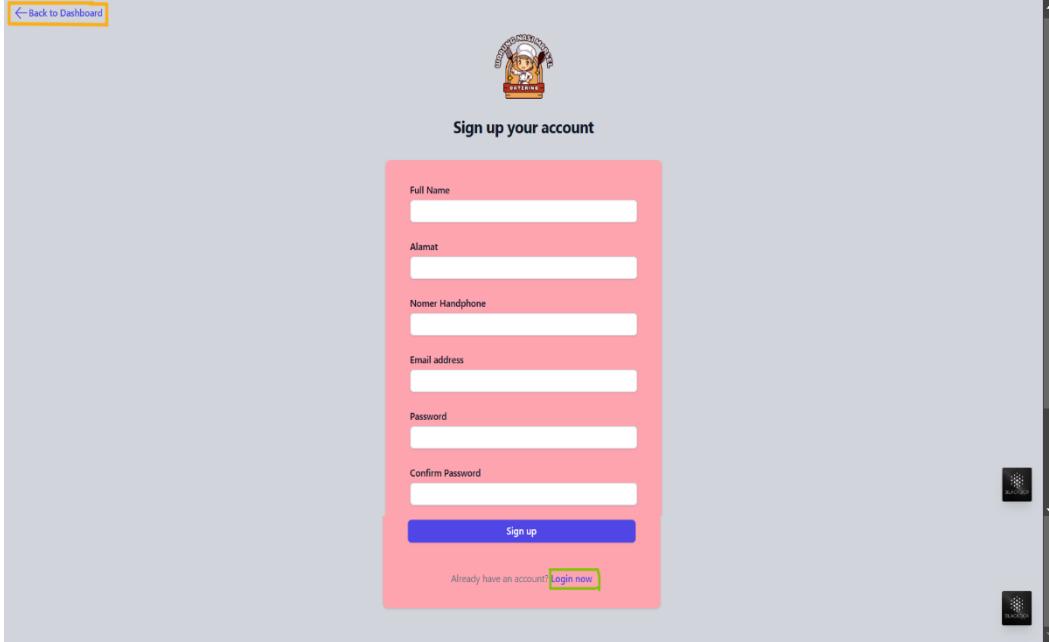
Jika user/pelanggan ingin kembali ke halaman dashboard bisa mengklik tanda panah “Back to Dashboard”. Jika user/pelanggan mengalami lupa password untuk login bisa mengklik tulisan “Lupa Password” dan akan muncul tampilan seperti gambar di bawah.



Untuk user/pelanggan yang belum mempunyai akun bisa mengklik tulisan “Register Sekarang” dan akan muncul form yang terdiri dari full name, alamat, nomer handphone, email address, password, dan confirm password.

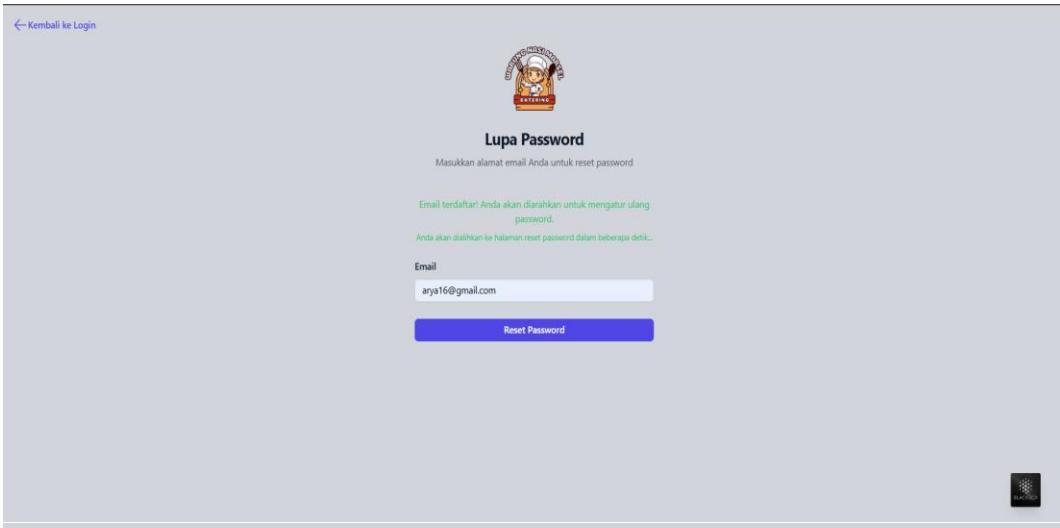


Dan di halaman register ini juga terdapat tanda panah “Back to Dashboard” dan tulisan “Login now” jika sudah memiliki akun.



- 10.Untuk halaman lupa password user/pelanggan dapat memasukkan alamat email yang ingin diubah passwordnya. Jika sudah dan mengklik button reset password akan muncul alert dan beralih ke halaman reset password seperti gambar di bawah.

---



Kembali ke Login



Lupa Password

Masukkan alamat email Anda untuk reset password.

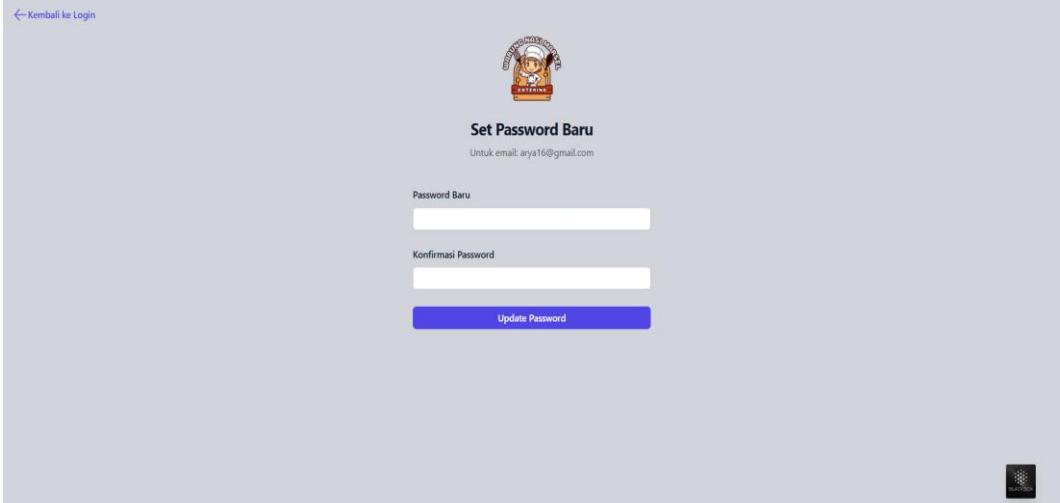
Email terdaftar! Anda akan diarahkan untuk mengatur ulang password.

Anda akan diajak ke halaman reset password dalam beberapa detik..

Email

arya16@gmail.com

Reset Password



Kembali ke Login



Set Password Baru

Untuk email: arya16@gmail.com

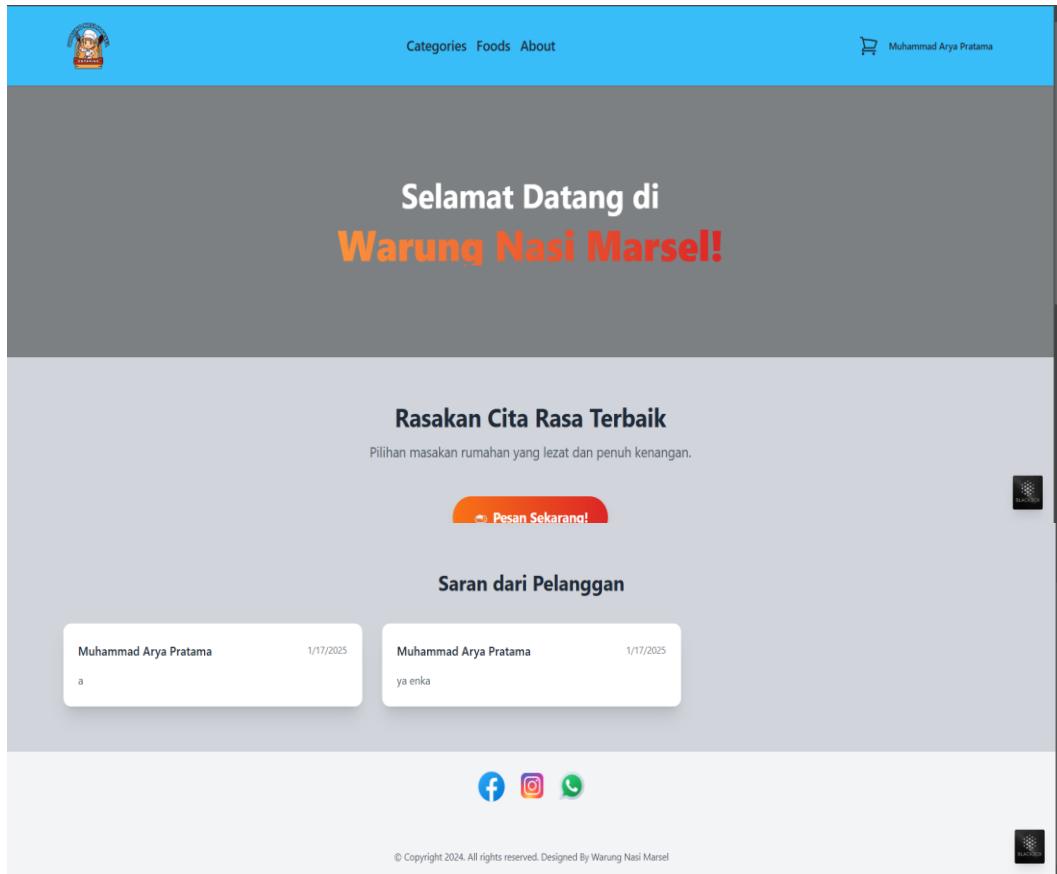
Password Baru

Konfirmasi Password

Update Password

Jika sudah memasukkan password baru dan konfirmasi password silahkan mengklik tombol update password dan akan langsung beralih ke halaman login.

11.Jika sudah melakukan login tampilan dashboard akan berubah seperti gambar di bawah.



12. Jika user/pelanggan kembali ingin memesan makanan bisa mengklik button pesan sekarang dan akan beralih ke halaman order form dan jika ingin memasukkan menu makanan ke keranjang bisa mengklik button gambar keranjang dan akan muncul notif dan gambar keranjang akan berubah.



**Order Form**

<b>Customer Name</b>	<input type="text" value="Enter your name"/>
<b>Selected Menu</b>	<input type="text" value="Paket Nasi Kotak 1"/>
<b>Address</b>	<input type="text" value="Enter your address"/>
<b>Phone Number</b>	<input type="text" value="Enter your phone number"/> Nomor HP harus diawali dengan 62 (kode negara Indonesia).
<b>Quantity</b>	<input type="text" value="1"/>
<b>Total Price</b>	<input type="text" value="Rp 25.000"/>
<b>Order Now</b>	



Categories Foods About Muhammad Arya Pratama

### Selamat datang di Catering Warung Nasi Marel

Kami menyediakan berbagai menu lezat dengan harga terjangkau. Jelajahi menu kami dan temukan hidangan favorit Anda.

### Daftar Menu

			
Paket Nasi Kotak 1 <b>Rp 25.000</b>	Paket Nasi Kotak 2 <b>Rp 30.000</b>	Paket Nasi Kotak 3 <b>Rp 35.000</b>	Paket Nasi Kotak 4 <b>Rp 40.000</b>

Silahkan isi form yang ada di halaman order, jika sudah klik button order now dan akan beralih ke halaman pembayaran.

**Pembayaran QRIS**

**Detail Pesanan:**

Nama: Rajoo  
Alamat: Kampung Durian Runtuh  
No. Telepon: 6281388946079  
Menu: Paket Nasi Kotak 1

**Total Pembayaran: Rp 25.000**

Scan QR Code untuk pembayaran:

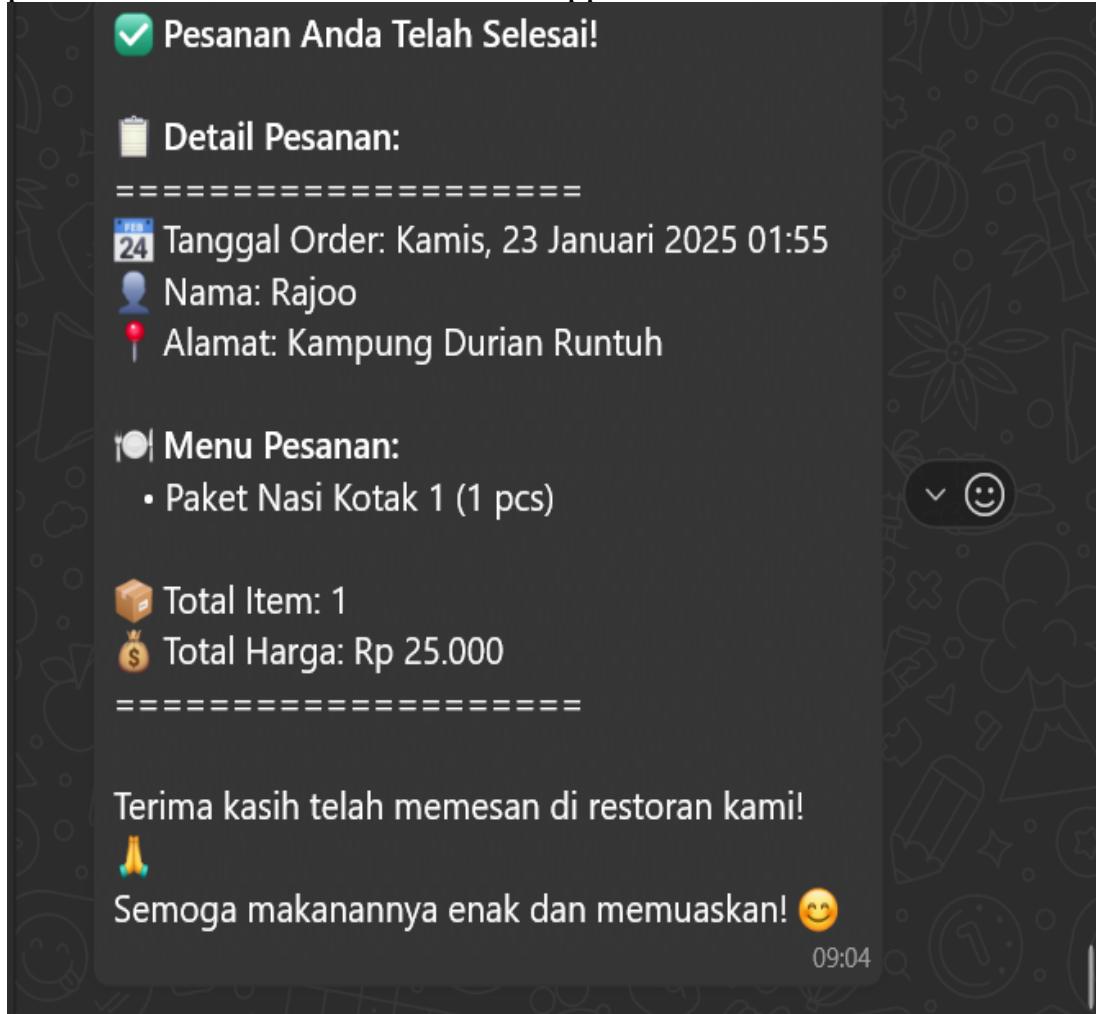


Pastikan nominal pembayaran sesuai dengan total yang tertera

Setelah melakukan pembayaran, klik tombol "Selesai Pembayaran" di bawah ini



Jika sudah melakukan pembayaran silahkan klik button “Selesai Pembayaran” dan tunggu konfirmasi admin. Jika admin sudah melakukan konfirmasi pemesanan akan muncul notif whatsapp.



---

## Lampiran Kode Program

```
Login.jsx
import { useState } from "react";
import { Link, useNavigate } from "react-router-dom";
import axiosClient from "../axiosClient";
import Logo from "../assets/logo.png";

export default function Login() {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [error, setError] = useState(null);
  const [isLoading, setIsLoading] = useState(false);
  const navigate = useNavigate();

  const handleLogin = async (e) => {
    e.preventDefault();
    setError(null);
    setIsLoading(true);

    try {
      const response = await axiosClient.post("/auth/login", {
        email,
        password,
      });

      if (response.data.status === "success") {
        const { access_token } = response.data;
        localStorage.setItem("access_token", access_token);
        navigate("/");
      }
    } catch (err) {
      // Display the specific error message from the backend
      setError(
        err.response?.data?.message ||
        err.response?.data?.error ||
        "Login gagal. Cek kembali kredensial Anda."
      );
    } finally {
      setIsLoading(false);
    }
  };

  return (
    <div className="flex min-h-full flex-1 flex-col justify-center px-6 py-12 lg:px-8">
      <div className="absolute top-4 left-4">
        <Link
          to="/"
          className="text-indigo-600 hover:text-indigo-500 flex items-center"
        >
          <svg
            xmlns="http://www.w3.org/2000/svg"
            fill="none"
            viewBox="0 0 24 24"
            strokeWidth="1.5"
            stroke="currentColor"
            className="size-6"
          >
            <path
              strokeLinecap="round"
              strokeLinejoin="round"
              d="M10.5 19.5 3 12m0 0 7.5-7.5M3 12h18"
            />
          </svg>
        <span className="text-sm font-semibold">Back to Dashboard</span>
      </Link>
    </div>
    <div className="sm:mx-auto sm:w-full sm:max-w-sm">
```

---

```
<img alt="Your Company" src={Logo} className="mx-auto h-30 w-28" />
<h2 className="mt-3 text-center text-2xl font-bold tracking-tight text-gray-900">
  Sign in to your account
</h2>
</div>

<div className="mt-8 sm:mx-auto sm:w-full sm:max-w-md">
  <div className="bg-rose-300 py-8 px-4 shadow sm:rounded-lg sm:px-10">
    <form className="space-y-6" onSubmit={handleLogin}>
      {error && (
        <div className="p-3 rounded bg-red-100 border border-red-400 text-red-700 text-sm text-center">
          {error}
        </div>
      )}
      <div>
        <label
          htmlFor="email"
          className="block text-sm font-medium text-gray-900"
        >
          Email address
        </label>
        <div className="mt-2">
          <input
            id="email"
            name="email"
            type="email"
            required
            autoComplete="email"
            value={email}
            onChange={(e) => setEmail(e.target.value)}
            className="block w-full rounded-md border-0 py-1.5 text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300 placeholder:text-gray-400 focus:ring-2 focus:ring-inset focus:ring-indigo-600 sm:text-sm"
          />
        </div>
      </div>

      <div>
        <label
          htmlFor="password"
          className="block text-sm font-medium text-gray-900"
        >
          Password
        </label>
        <div className="mt-2">
          <input
            id="password"
            name="password"
            type="password"
            required
            autoComplete="current-password"
            value={password}
            onChange={(e) => setPassword(e.target.value)}
            className="block w-full rounded-md border-0 py-1.5 text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300 placeholder:text-gray-400 focus:ring-2 focus:ring-inset focus:ring-indigo-600 sm:text-sm"
          />
        </div>
      </div>

      <div className="flex items-center">
        <Link
          to="/forgot-password"
          className="text-sm text-indigo-600 hover:text-indigo-500"
        >
          Lupa Password?
        </Link>
      </div>

      <div>
        <button
          type="submit"

```

```
disabled={isLoading}
className={`flex w-full justify-center rounded-md px-3 py-1.5 text-sm font-semibold text-white shadow-sm focus-visible:outline focus-visible:outline-2 focus-visible:outline-offset-2 focus-visible:outline-indigo-600 ${
  isLoading
    ? "bg-indigo-400 cursor-not-allowed"
    : "bg-indigo-600 hover:bg-indigo-500"
`}`}
>
{isLoading ? "Signing in..." : "Sign in"}
</button>
</div>
</form>
<p className="mt-10 text-center text-sm text-gray-500">
Belum Mempunyai Akun?{" "}
<Link
to="/signup"
className="font-semibold text-indigo-600 hover:text-indigo-500"
>
  Register Sekarang
</Link>
</p>
</div>
</div>
</div>
);
}
```

```
Signup.jsx
import { useState } from "react";
import { Link, useNavigate } from "react-router-dom";
import axiosClient from "../axiosClient";
import Logo from "../assets/logo.png";

export default function Signup() {
  const [formData, setFormData] = useState({
    name: "",
    alamat: "",
    telephone: "",
    email: "",
    password: "",
    passwordConfirmation: "",
  });

  const [errors, setErrors] = useState([]);
  const [successMessage, setSuccessMessage] = useState("");
  const [showNotification, setShowNotification] = useState(false);
  const navigate = useNavigate();

  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData((prevData) => ({
      ...prevData,
      [name]: value,
    }));
  };

  const handleSubmit = async (e) => {
    e.preventDefault();

    // Reset errors dan success message sebelum submit
    setErrors([]);
    setSuccessMessage("");

    // Validasi frontend: Pastikan password dan konfirmasi cocok
    if(formData.password !== formData.passwordConfirmation) {
      setErrors(["Passwords do not match."]);
      return;
    }

    try {
```

```
// Kirim data ke backend dengan `password_confirmation`
const response = await axiosClient.post("/auth/register", {
  name: formData.name,
  alamat: formData.alamat,
  telephone: formData.telephone,
  email: formData.email,
  password: formData.password,
  password_confirmation: formData.passwordConfirmation,
});

if (response.status === 201) {
  setShowNotification(true);

  // Gunakan delay untuk memberikan waktu pengguna melihat pesan
  setTimeout(() => {
    setSuccessMessage(""); // Hapus pesan sukses sebelum navigasi
    setShowNotification(false); // Pastikan notifikasi juga disembunyikan
    navigate("/login");
  }, 3000);
}

} catch (error) {
  if (error.response) {
    const backendErrors = error.response.data.message;

    // Backend mengirim validasi error dalam bentuk objek
    if (typeof backendErrors === "object") {
      const errorMessages = Object.values(backendErrors).flat();
      setErrors(errorMessages);
    } else {
      setErrors([backendErrors]);
    }
  } else {
    setErrors(["Something went wrong. Please try again."]);
  }
}

};

return (
<div className="flex min-h-full flex-1 flex-col justify-center px-6 py-12 lg:px-8">
<div className="absolute top-4 left-4">
<Link
  to="/"
  className="text-indigo-600 hover:text-indigo-500 flex items-center"
>
<svg
  xmlns="http://www.w3.org/2000/svg"
  fill="none"
  viewBox="0 0 24 24"
  strokeWidth="1.5"
  stroke="currentColor"
  className="size-6"
>
<path
  strokeLinecap="round"
  strokeLinejoin="round"
  d="M10.5 19.5 3 12m0 0 7.5-7.5M3 12h18"
/>
</svg>
<span className="text-sm font-semibold">Back to Dashboard</span>
</Link>
</div>
<div className="sm:mx-auto sm:w-full sm:max-w-sm">
<img alt="Your Company" src={Logo} className="mx-auto h-30 w-28" />
<h2 className="mt-3 text-center text-2xl font-bold tracking-tight text-gray-900">
  Sign up your account
</h2>
</div>

<div className="mt-8 sm:mx-auto sm:w-full sm:max-w-md">
```

---

```
<div className="bg-rose-300 py-8 px-4 shadow sm:rounded-lg sm:px-10">
  <form className="space-y-6" onSubmit={handleSubmit}>
    <div>
      <label
        htmlFor="name"
        className="block text-sm font-medium text-gray-900"
      >
        Full Name
      </label>
      <div className="mt-1 relative rounded-md shadow-sm">
        <input
          id="name"
          name="name"
          type="text"
          value={formData.name}
          onChange={handleChange}
          required
          maxLength={255}
          className="block w-full rounded-md border-0 py-1.5 text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300 placeholder:text-gray-400 focus:ring-2 focus:ring-inset focus:ring-indigo-600 sm:text-sm"
        />
      </div>
    </div>
    <div>
      <label
        htmlFor="alamat"
        className="block text-sm font-medium text-gray-900"
      >
        Alamat
      </label>
      <div className="mt-1 relative rounded-md shadow-sm">
        <input
          id="alamat"
          name="alamat"
          type="text"
          value={formData.alamat}
          onChange={handleChange}
          required
          maxLength={255}
          className="block w-full rounded-md border-0 py-1.5 text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300 placeholder:text-gray-400 focus:ring-2 focus:ring-inset focus:ring-indigo-600 sm:text-sm"
        />
      </div>
    </div>
    <div>
      <label
        htmlFor="telephone"
        className="block text-sm font-medium text-gray-900"
      >
        Nomer Handphone
      </label>
      <div className="mt-1 relative rounded-md shadow-sm">
        <input
          id="telephone"
          name="telephone"
          type="text"
          value={formData.telephone}
          onChange={handleChange}
          required
          maxLength={255}
          className="block w-full rounded-md border-0 py-1.5 text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300 placeholder:text-gray-400 focus:ring-2 focus:ring-inset focus:ring-indigo-600 sm:text-sm"
        />
      </div>
    </div>
    <div>
      <label
        htmlFor="email"
        className="block text-sm font-medium text-gray-900"
      >
```

```
>
  Email address
</label>
<div className="mt-1 relative rounded-md shadow-sm">
  <input
    id="email"
    name="email"
    type="email"
    value={formData.email}
    onChange={handleChange}
    required
    maxLength={255}
    className="block w-full rounded-md border-0 py-1.5 text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300 placeholder:text-gray-400 focus:ring-2 focus:ring-inset focus:ring-indigo-600 sm:text-sm"
  />
</div>
</div>

<div>
  <label
    htmlFor="password"
    className="block text-sm font-medium text-gray-900"
  >
    Password
  </label>
  <div className="mt-1 relative rounded-md shadow-sm">
    <input
      id="password"
      name="password"
      type="password"
      value={formData.password}
      onChange={handleChange}
      required
      autoComplete="current-password"
      className="block w-full rounded-md border-0 py-1.5 text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300 placeholder:text-gray-400 focus:ring-2 focus:ring-inset focus:ring-indigo-600 sm:text-sm"
    />
  </div>
</div>

<div>
  <label
    htmlFor="passwordConfirmation"
    className="block text-sm font-medium text-gray-900"
  >
    Confirm Password
  </label>
  <div className="mt-1 relative rounded-md shadow-sm">
    <input
      id="passwordConfirmation"
      name="passwordConfirmation"
      type="password"
      value={formData.passwordConfirmation}
      onChange={handleChange}
      required
      className="block w-full rounded-md border-0 py-1.5 text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300 placeholder:text-gray-400 focus:ring-2 focus:ring-inset focus:ring-indigo-600 sm:text-sm"
    />
  </div>
</div>

{errors.length > 0 && (
  <div className="text-red-500 text-sm space-y-1">
    {errors.map((error, index) => (
      <p key={index}>{error}</p>
    )))
  </div>
)}

{successMessage && (
```

```
<div className="text-green-500 text-sm">{successMessage}</div>
)}

<div>
  <button
    type="submit"
    className="flex w-full justify-center rounded-md bg-indigo-600 px-3 py-1.5 text-sm font-semibold text-white shadow-sm hover:bg-indigo-500 focus-visible:outline focus-visible:outline-2 focus-visible:outline-offset-2 focus-visible:outline-indigo-600"
  >
    Sign up
  </button>
</div>
</form>
<p className="mt-10 text-center text-sm text-gray-500">
  Already have an account?{" "}
  <Link
    to="/login"
    className="font-semibold text-indigo-600 hover:text-indigo-500"
  >
    Login now
  </Link>
</p>
</div>
</div>

{showNotification && (
  <div className="fixed top-4 left-1/2 transform -translate-x-1/2 bg-green-500 text-white px-4 py-2 rounded shadow-lg">
    Registration successful!
  </div>
)
);
}
```

```
FoodsMenu.jsx
import { useState, useEffect } from "react";
import Swal from "sweetalert2";
import Navbar from "../components/Navbar";
import Footer from "../components/Footer";
import axiosClient from "../axiosClient";
import { useNavigate } from "react-router-dom";
import "../styles/animations.css";

function FoodsMenu() {
  const [cartCount, setCartCount] = useState(0);
  const [foods, setFoods] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);
  const navigate = useNavigate();

  const formatCurrency = (amount) => {
    return new Intl.NumberFormat("id-ID", {
      style: "currency",
      currency: "IDR",
      minimumFractionDigits: 0,
    }).format(amount);
  };

  const fetchCartCount = async () => {
    try {
      const token = localStorage.getItem("access_token");
      if (token) {
        const cartResponse = await axiosClient.get("/cart");
        if (cartResponse.data && cartResponse.data.data) {
          const cartData = cartResponse.data.data;
          const totalCount = cartData.reduce(
            (total, item) => total + item.jumlah,
            0
          );
          setCartCount(totalCount);
        }
      }
    } catch (error) {
      setError(error.message);
    }
  };
}

export default FoodsMenu;
```

```
        }
    }
} catch (error) {
    console.error("Failed to fetch cart count:", error);
    // Don't show error to user as this is not critical
}
};

useEffect(() => {
    const fetchData = async () => {
        try {
            setLoading(true);
            setError(null);

            const foodsResponse = await axiosClient.get("/foods");

            if (foodsResponse.data && foodsResponse.data.data) {
                setFoods(foodsResponse.data.data);
                await fetchCartCount();
            } else {
                throw new Error("Invalid data format received from server");
            }
        } catch (error) {
            console.error("Error fetching foods:", error);
            setError(error.response?.data?.message || "Gagal memuat data menu.");
        }
    }

    Swal.fire({
        title: "Error",
        text: error.response?.data?.message || "Gagal memuat data menu.",
        icon: "error",
        confirmButtonText: "OK",
    });
} finally {
    setLoading(false);
}
};

fetchData();
}, []);

const goToDetail = (foodId) => {
    navigate(`/foods/${foodId}`);
};

const handleOrderNow = async (food, e) => {
    e.stopPropagation(); // Prevent event bubbling

    try {
        const token = localStorage.getItem("access_token");
        if (!token) {
            Swal.fire({
                title: "Login Required",
                text: "Anda harus login terlebih dahulu untuk memesan.",
                icon: "warning",
                confirmButtonText: "OK",
            });
            navigate("/foods");
            return;
        }
    }

    navigate("/order-form", {
        state: {
            orderType: "direct",
            foodData: {
                id: food.id,
                quantity: 1,
            },
        },
    });
} catch (error) {
```

```
console.error("Order failed:", error);
Swal.fire({
  title: "Error",
  text: "Gagal memproses pesanan.",
  icon: "error",
  confirmButtonText: "OK",
});
};

const handleCartClick = async (food, e) => {
e.stopPropagation(); // Prevent event bubbling

const token = localStorage.getItem("access_token");
if (!token) {
  Swal.fire({
    title: "Login Required",
    text: "Anda harus login terlebih dahulu untuk mengakses keranjang.",
    icon: "warning",
    confirmButtonText: "OK",
  });
  navigate("/login");
  return;
}

try {
  const response = await axiosClient.post("/cart", {
    food_id: food.id,
    jumlah: 1,
  });

  if (response.data.status === "success") {
    await fetchCartCount();

    Swal.fire({
      title: "Success",
      text: `${food.nama} telah ditambahkan ke keranjang.`,
      icon: "success",
      confirmButtonText: "OK",
    });
  }
} catch (error) {
  console.error("Cart error:", error);
  const errorMessage =
    error.response?.data?.message || "Gagal menambahkan item ke keranjang.";

  Swal.fire({
    title: "Error",
    text: errorMessage,
    icon: "error",
    confirmButtonText: "OK",
  });
}

if (error) {
  return (
    <div className="min-h-screen flex flex-col">
      <Navbar cartCount={cartCount} />
      <div className="container mx-auto py-12 px-4 md:px-8 flex-grow">
        <div className="text-center text-red-600">
          <h2 className="text-2xl font-bold mb-4">Error Loading Menu</h2>
          <p>{error}</p>
        </div>
      </div>
      <Footer />
    </div>
  );
}
}
```

```
return (
<div className="min-h-screen flex flex-col">
  <Navbar cartCount={cartCount} />
  <div className="container mx-auto py-12 px-4 md:px-8 flex-grow">
    <div className="text-center space-y-6 animate-fadeIn">
      <h2 className="text-4xl font-bold text-teal-700 animate-slideDown">
        Selamat datang di Catering Warung Nasi Marsel
      </h2>
      <p className="text-xl text-gray-600 animate-slideUp">
        Kami menyediakan berbagai menulezat dengan harga terjangkau.
        Jelajahi menu kami dan temukan hidangan favorit Anda.
      </p>
      <h1 className="text-5xl font-extrabold text-teal-800 animate-bounce">
        Daftar Menu
      </h1>
    </div>

  {loading ? (
    <div className="text-center py-20">
      <div className="animate-spin rounded-full h-12 w-12 border-b-2 border-teal-500 mx-auto"></div>
      <p className="mt-4 text-gray-600">Loading menu...</p>
    </div>
  ) : (
    <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-8 mt-12">
      {foods.map((food, index) => (
        <div
          key={food.id}
          className="bg-white rounded-lg overflow-hidden shadow-lg hover:scale-105 hover:shadow-2xl transition-all duration-500 cursor-pointer
          animate-fadeInUp"
          style={{ animationDelay: `${index * 150}ms` }}
          onClick={() => goToDetail(food.id)}
        >
          <div className="h-56 overflow-hidden relative">
            <img
              className="w-full h-full object-cover transform hover:scale-110 transition-transform duration-500"
              src={`${import.meta.env.VITE_API_BASE_URL}${food.gambar_url}`}
            >
            alt={food.nama}
            onError={(e) => {
              e.target.src = "/placeholder-food.jpg"; // Add a placeholder image
              e.target.className = "w-full h-full object-cover";
            }}
          </div>
        </div>
        <div className="p-6 flex flex-col justify-between space-y-4">
          <h3 className="text-xl font-bold text-teal-700 text-center animate-pulse">
            {food.nama}
          </h3>
          <div className="font-bold text-2xl text-center text-teal-600 animate-slideUp">
            {formatCurrency(food.harga)}
          </div>
          <div className="flex items-center justify-center space-x-4">
            <button
              className="bg-teal-500 hover:bg-teal-600 text-white font-bold py-3 px-4 rounded-lg shadow-md transform hover:scale-105
              transition-all duration-300"
              onClick={(e) => handleCartClick(food, e)}
            >
              <svg
                xmlns="http://www.w3.org/2000/svg"
                fill="none"
                viewBox="0 0 24 24"
                strokeWidth="1.5"
                stroke="currentColor"
                className="size-6"
              >
                <path
                  strokeLinecap="round"
                  strokeLinejoin="round"
                  d="M2.25 3h1.386c.51 0 .955.343 1.087.835l.383 1.437M7.5 14.25a3 3 0 0 0-3 3h15.75m-12.75-3h11.218c1.121-2.3 2.1-4.684
                </path>
              </svg>
            </button>
          </div>
        </div>
      </div>
    </div>
  )
)
```

---

```
2.924-7.138a60.114 60.114 0 0 0-16.536-1.84M7.5 14.25 5.106 5.272M6 20.25a.75.75 0 1 1-1.5 0 .75.75 0 0 1 1.5 0Zm12.75 0a.75.75 0 1 1-1.5 0
.75.75 0 0 1 1.5 0Z"
    />
    </svg>
</button>
<button
    className="bg-blue-500 hover:bg-blue-600 text-white font-bold py-3 px-6 rounded-lg shadow-md transform hover:scale-105
transition-all duration-300"
    onClick={(e) => handleOrderNow(food, e)}
    >
    Pesan Sekarang
    </button>
</div>
</div>
</div>
    )}
)
</div>
<Footer />
</div>
);
}

export default FoodsMenu;

CategoryMenu.jsx
import { useState, useEffect } from "react";
import { useNavigate } from "react-router-dom";
import Navbar from "../components/Navbar";
import Footer from "../components/Footer";
import axiosClient from "../axiosClient";
import "./styles/animations.css";

function CategoryMenu() {
    const [categories, setCategories] = useState([]);
    const [cartCount, setCartCount] = useState(0);
    const navigate = useNavigate();

    useEffect(() => {
        const fetchCategories = async () => {
            try {
                const response = await axiosClient.get("/categories");
                setCategories(response.data.data);
            } catch (error) {
                console.error("Failed to fetch categories:", error);
            }
        };
        const fetchCartCount = async () => {
            const token = localStorage.getItem("access_token");
            if (!token) return;
            try {
                const response = await axiosClient.get("/cart");
                const totalCount = response.data.data.reduce(
                    (total, item) => total + item.jumlah,
                    0
                );
                setCartCount(totalCount);
            } catch (error) {
                console.error("Failed to fetch cart count:", error);
            }
        };
        fetchCategories();
        fetchCartCount();
    }, []);
}

const handleCategoryClick = (categoryId) => {
```

---

```
navigate(`/categories/${categoryId}`);
};

return (
<div className="min-h-screen flex flex-col">
<Navbar cartCount={cartCount} />

 {/* Hero Section with Animation */}
<div className="bg-gradient-to-r from-orange-50 to-orange-100 py-16 px-4">
<div className="container mx-auto text-center">
<h1 className="text-5xl font-bold text-orange-800 mb-6 animate-slideDown">
    Menu Pilihan Warung Nasi Marsel
</h1>
<p className="text-xl text-gray-700 mb-8 animate-fadeIn">
    Nikmati Berbagai Pilihan Menu Istimewa Untuk Setiap Momen Spesial
    Anda
</p>
</div>
</div>

 {/* Categories Section with Animation */}
<div className="container mx-auto py-16 px-4">
<div className="text-center mb-16">
<h2 className="text-4xl font-bold text-gray-800 mb-6 animate-bounce">
    Pilihan Kategori Menu
</h2>
<p className="text-lg text-gray-600 max-w-2xl mx-auto animate-fadeIn">
    Temukan berbagai pilihan menulezat untuk acara spesial Anda. Dari
    paket prasmanan hingga nasi kotak, kami siap melayani dengan cita
    rasa yang autentik.
</p>
</div>

<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8">
{categories.map((category, index) => (
<div
    key={category.id}
    onClick={() => handleCategoryClick(category.id)}
    className="bg-white rounded-xl shadow-lg hover:shadow-2xl transition-all duration-500 overflow-hidden cursor-pointer transform hover:translate-y-2 animate-fadeInUp"
    style={{ animationDelay: `${index * 200}ms` }}
>
<div className="relative h-48 overflow-hidden group">
<img
    src={`${import.meta.env.VITE_API_BASE_URL}${category.image_url}`}
    alt={category.name}
    className="w-full h-full object-cover transition-transform duration-500 group-hover:scale-110"
    onError={(e) => {
        e.target.onerror = null;
        e.target.src = "/placeholder-image.jpg";
    }}
/>
<div className="absolute inset-0 bg-black bg-opacity-0 group-hover:bg-opacity-20 transition-all duration-300" />
</div>

<div className="p-6">
<h3 className="text-2xl font-bold text-gray-800 mb-3 animate-pulse">
    {category.name}
</h3>
<button className="mt-4 bg-gradient-to-r from-orange-500 to-orange-600 text-white px-8 py-3 rounded-full hover:from-orange-600
    hover:to-orange-700 transition-all duration-300 w-full transform hover:scale-105 font-semibold">
        Lihat Menu
</button>
</div>
</div>
))}>
</div>
</div>
```

---

```
<Footer />
</div>
);
}

export default CategoryMenu;

Dashboard.jsx
import { useState, useEffect } from "react";
import Navbar from "../components/Navbar";
import Footer from "../components/Footer";
import axiosClient from "../axiosClient";
import { useNavigate } from "react-router-dom";

export default function Dashboard() {
  const [cartCount, setCartCount] = useState(0);
  const [suggestions, setSuggestions] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);
  const navigate = useNavigate();

  useEffect(() => {
    const fetchCartCount = async () => {
      const token = localStorage.getItem("access_token");
      if (!token) {
        setCartCount(0);
        return;
      }
      try {
        const response = await axiosClient.get("/cart");
        const totalJumlah = response.data.data.reduce(
          (total, item) => total + item.jumlah,
          0
        );
        setCartCount(totalJumlah);
      } catch (error) {
        console.error("Failed to fetch cart count:", error);
        if (error.response && error.response.status === 401) {
          localStorage.removeItem("access_token");
        }
        setCartCount(0);
      }
    };
    const fetchSuggestions = async () => {
      setLoading(true);
      setError(null);
      try {
        const response = await axiosClient.get("/suggestions");
        setSuggestions(response.data.data);
      } catch (error) {
        console.error("Failed to fetch suggestions:", error);
        if (error.code === "ECONNABORTED") {
          setError("Request timeout. Please try again.");
        } else if (error.response && error.response.status >= 500) {
          setError("Server is currently unavailable. Please try again later.");
        } else {
          setError("Failed to load suggestions. Please check your connection.");
        }
      } finally {
        setLoading(false);
      }
    };
    fetchCartCount();
    fetchSuggestions();
  }, []);

  return (

```

---

```
<div className="min-h-screen flex flex-col">
<Navbar cartCount={cartCount} />
<div
  className="hero-section bg-cover bg-center h-96 flex items-center justify-center text-white relative overflow-hidden"
  style={{ 
    backgroundImage:
      "url('https://source.unsplash.com/1600x900/?restaurant,food')",
  }}>
  <div className="absolute inset-0 bg-black bg-opacity-40"></div>
  <div className="relative z-10 text-center">
    <h1 className="text-5xl font-bold mb-4">Selamat Datang di</h1>
    <div className="text-6xl font-extrabold text-transparent bg-clip-text bg-gradient-to-r from-orange-400 to-red-600">
      Warung Nasi Marsel!
    </div>
  </div>
</div>
</div>

<div className="container mx-auto px-4 py-16">
  <div className="text-center mb-12">
    <h2 className="text-4xl font-bold text-gray-800 mb-4">
      Rasakan Cita Rasa Terbaik
    </h2>
    <p className="text-xl text-gray-600">
      Pilihan masakan rumahan yang lezat dan penuh kenangan.
    </p>
  </div>

  <div className="text-center">
    <button
      className="bg-gradient-to-r from-orange-500 to-red-600 text-white px-8 py-4 rounded-full text-xl font-bold hover:from-orange-600
      hover:to-red-700 transition-all shadow-lg transform hover:scale-105 hover:-translate-y-1"
      onClick={() => navigate("/foods")}>
      🍽 Pesan Sekarang!
    </button>
  </div>

  <div className="mt-20">
    <h2 className="text-3xl font-bold text-gray-800 mb-10 text-center">
      Saran dari Pelanggan
    </h2>

    {loading && (
      <div className="text-center py-4">
        <div className="animate-spin rounded-full h-12 w-12 border-b-2 border-orange-500 mx-auto"></div>
      </div>
    )}

    {error && (
      <div className="bg-red-100 text-red-700 p-4 rounded-lg mb-4 text-center">
        {error}
      </div>
    )}

    {!loading && !error && suggestions.length === 0 && (
      <p className="text-gray-600 text-center">
        Belum ada saran dari pelanggan.
      </p>
    )}

    {!loading && !error && suggestions.length > 0 && (
      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8">
        {suggestions.map((suggestion, index) => (
          <div
            key={suggestion.id}
            className="bg-white rounded-lg shadow-lg p-6 transform hover:scale-105 transition-all duration-300">
            <div className="flex justify-between items-center mb-4">
              <div className="text-lg text-gray-800 font-semibold">
```

```
{suggestion.user?.name || "Anonymous"}  
      </div>  
      <div className="text-sm text-gray-500">  
        {new Date(suggestion.created_at).toLocaleDateString()}  
      </div>  
    </div>  
    <p className="text-gray-600 leading-relaxed">  
      {suggestion.content}  
    </p>  
  </div>  
))}  
      </div>  
    )}  
  </div>  
</div>  
);  
}
```

#### About.jsx

```
import { useState, useEffect } from "react";  
import Navbar from "../components/Navbar";  
import Footer from "../components/Footer";  
import axiosClient from "../axiosClient";  
import { useNavigate } from "react-router-dom";  
  
function About() {  
  const [cartCount, setCartCount] = useState(0);  
  const [suggestion, setSuggestion] = useState("");  
  const [submitStatus, setSubmitStatus] = useState(null);  
  const [isLoggedIn, setIsLoggedIn] = useState(false);  
  const [userData, setUserData] = useState(null);  
  const navigate = useNavigate();  
  
  useEffect(() => {  
    const token = localStorage.getItem("access_token");  
    setIsLoggedIn(!token);  
  
    // Fetch user data if logged in  
    const fetchUserData = async () => {  
      if (token) {  
        try {  
          const response = await axiosClient.post("/auth/me");  
          setUserData(response.data);  
        } catch (error) {  
          console.error("Failed to fetch user data:", error);  
          if (error.response?.status === 401) {  
            handleLogout();  
          }
        }
      }
    };
  );  
  
  const fetchCartCount = async () => {  
    if (!token) {  
      setCartCount(0);  
      return;
    }
  );
  try {
    const response = await axiosClient.get("/cart");
    const totalCartCount = response.data.data.reduce(
      (total, item) => total + item.jumlah,
      0
    );
    setCartCount(totalCartCount);
  } catch (error) {
    console.error("Failed to fetch cart count:", error);
  }
}
```

---

```
        if (error.response?.status === 401) {
          handleLogout();
        }
      };
    }

    fetchUserData();
    fetchCartCount();
  }, []);
}

const handleLogout = () => {
  localStorage.removeItem("access_token");
  setIsLoggedIn(false);
  setCartCount(0);
  setUserData(null);
};

const handleSubmit = async (e) => {
  e.preventDefault();

  if (!isLoggedIn) {
    const confirmLogin = window.confirm(
      "Anda harus login terlebih dahulu untuk memberikan saran. Apakah Anda ingin login sekarang?"
    );
    if (confirmLogin) {
      navigate("/login");
    }
    return;
  }

  try {
    const response = await axiosClient.post("/suggestions", {
      user_id: userData.id,
      content: suggestion,
    });

    if (response.status === 201) {
      setSubmitStatus({
        type: "success",
        message: "Terima kasih! Saran Anda telah berhasil dikirim.",
      });
      setSuggestion("");
      setTimeout(() => setSubmitStatus(null), 3000);
    }
  } catch (error) {
    console.error("Failed to submit suggestion:", error);
    if (error.response?.status === 401) {
      handleLogout();
      const confirmLogin = window.confirm(
        "Sesi Anda telah berakhir. Apakah Anda ingin login kembali?"
      );
      if (confirmLogin) {
        navigate("/login");
      }
    } else {
      setSubmitStatus({
        type: "error",
        message: "Maaf, terjadi kesalahan. Silakan coba lagi.",
      });
      setTimeout(() => setSubmitStatus(null), 3000);
    }
  }
};

return (
  <>
  <Navbar cartCount={cartCount} />
  <div className="container mx-auto px-4 py-8">
    <div className="header text-center mb-8">
      <h1 className="text-3xl font-bold">Warung Nasi Marsel</h1>
```

---

```
</div>

<div className="main-content flex flex-col md:flex-row md:space-x-8 bg-orange-100 p-8 rounded-lg">
  <div className="location md:w-1/2 mb-8 md:mb-0">
    <h2 className="text-2xl font-semibold text-center mb-4">Lokasi</h2>
    <iframe
      src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3965.44811654303!2d106.83169077453273!3d-6.335953161992128!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x2e69edaea5fe6f59%3A0x26f70cea80aa96b1!2sWarung%20Nasi%20Marsel!5e0!3m2!1sid!2sid!4v1721817577383!5m2!1sid!2sid"
      width="100%"
      height="450"
      style={{ border: 0 }}
      allowFullScreen
      loading="lazy"
      referrerPolicy="no-referrer-when-downgrade"
      className="rounded-lg"
    ></iframe>
  </div>

  <div className="about-us md:w-1/2">
    <h2 className="text-2xl font-semibold mb-4">Sejarah</h2>
    <p className="mb-6">
      Warung Nasi Marsel adalah tempat makan yang berasal dari Kuningan Jawa Barat, menawarkan berbagai macam hidangan khas Sunda. Kami menggunakan bahan-bahan segar dan berkualitas untuk menyajikan makanan yang lezat dan bergizi.
    </p>
    <div className="contact mb-8">
      <h2 className="text-2xl font-semibold mb-4">Kontak Kami</h2>
      <div className="contact-details space-y-2">
        <p>
          <strong>Alamat:</strong> Jl. Buku Dikrama RT.05 RW.04 NO.35
        </p>
        <p>
          <strong>Telepon:</strong> +62 812-9591-7717
        </p>
        <p>
          <strong>Email:</strong> warungnasimarsel@gmail.com
        </p>
      </div>
    </div>
  </div>

  <div className="suggestion-form bg-white p-6 rounded-lg shadow-sm">
    <h2 className="text-2xl font-semibold mb-4">
      Berikan Saran Anda
    </h2>

    { !isLoggedIn && (
      <div className="mb-4 p-3 bg-yellow-100 text-yellow-700 rounded">
        Silakan login terlebih dahulu untuk memberikan saran.
      </div>
    )}

    { submitStatus && (
      <div
        className={`mb-4 p-3 rounded ${
          submitStatus.type === "success"
            ? "bg-green-100 text-green-700"
            : "bg-red-100 text-red-700"
        }`}
      >
        {submitStatus.message}
      </div>
    )}
  </div>

  <form onSubmit={handleSubmit} className="space-y-4">
    <div>
      <label
        className="block text-sm font-medium mb-2"
      >
```

```
    htmlFor="suggestion"
  >
  Pesan Saran
</label>
<textarea
  id="suggestion"
  value={suggestion}
  onChange={(e) => setSuggestion(e.target.value)}
  className="w-full p-3 border rounded-lg focus:ring-2 focus:ring-blue-500 focus:border-blue-500"
  rows="4"
  placeholder={
    isLoggedIn
      ? "Tulis saran Anda di sini..."
      : "Silakan login terlebih dahulu untuk memberikan saran"
  }
  required
  disabled={!isLoggedIn}
></textarea>
</div>
<button
  type="submit"
  className={`w-full px-4 py-2 rounded-lg transition-colors ${
    isLoggedIn
      ? "bg-blue-500 text-white hover:bg-blue-600"
      : "bg-gray-300 text-gray-500 cursor-not-allowed"
  `}
  disabled={!isLoggedIn}
>
  {isLoggedIn ? "Kirim Saran" : "Login untuk Kirim Saran"}
</button>
</form>
</div>
</div>
</div>
<Footer />
</>
);
}

export default About;
```