

BAB IV

RANCANGAN SISTEM DAN PROGRAM USULAN

4.1 Analisis Kebutuhan Sistem

1. Kebutuhan Fungsional Pengunjung

Pengunjung memiliki kemampuan untuk melakukan pendaftaran sebagai *member*.

2. Kebutuhan Fungsional *Member*

- a. *Member* dapat masuk ke sistem menggunakan akun yang telah didaftarkan sebelumnya.
- b. *Member* bisa memilih produk yang ingin dibeli
- c. *Member* dapat membuka halaman pemesanan dan menyelesaikan proses *checkout*
- d. *Member* melakukan proses pembayaran
- e. *Member* melakukan konfirmasi atas pembayaran yang telah dilakukan
- f. *Member* mendapatkan *invoice* sebagai bukti transaksi

3. Kebutuhan Fungsional *Admin*

- a. *Admin* memiliki akses untuk *login* ke dalam sistem
- b. *Admin* dapat mengelola halaman home, tentang, berita, galeri, menu, kontak
- c. *Admin* dapat melakukan pengelolaan terhadap data pemesanan
- d. *Admin* dapat menyusun dan mengelola laporan penjualan

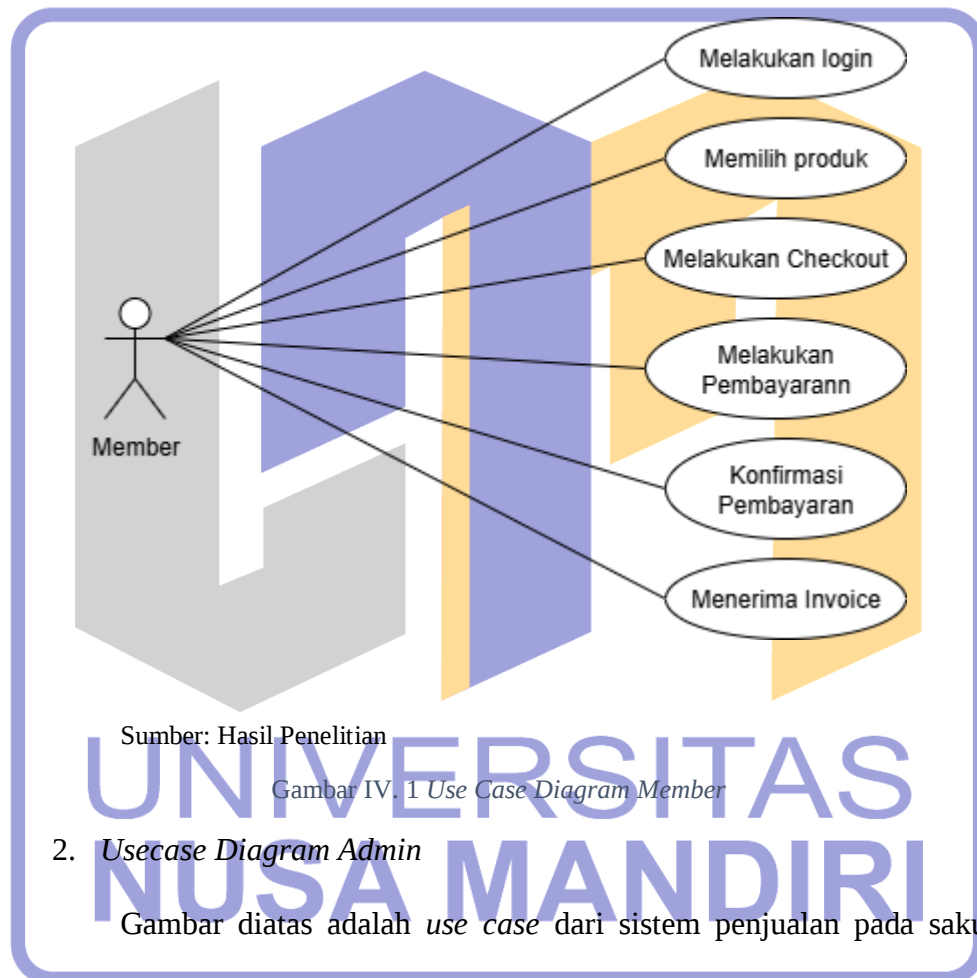
4.2 Desain

4.2.1 Desain Pemodelan Sistem

A. Pemodelan *Use Case Diagram*

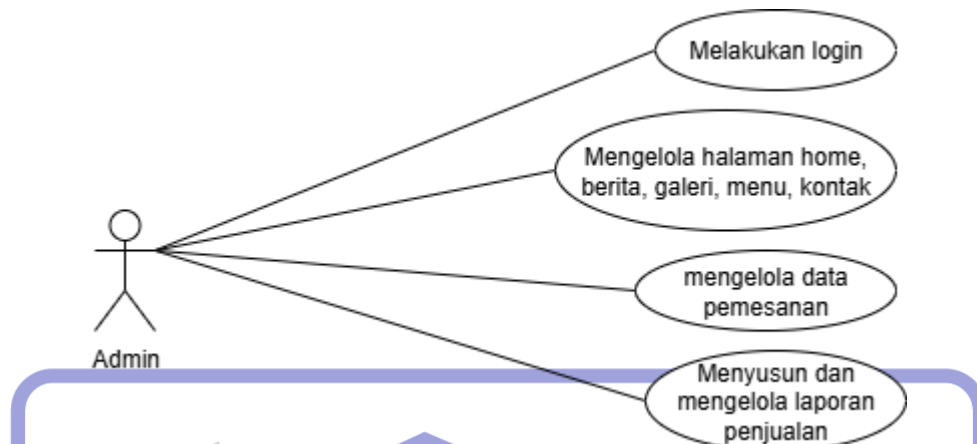
1. *Usecase Diagram Member*

Berikut ini adalah *use case* dari sistem penjualan pada sakupaleza Depok. Pada sistem penjualan ini *member* memiliki fungsi yang dapat diakses



2. *Usecase Diagram Admin*

Gambar diatas adalah *use case* dari sistem penjualan pada sakupaleza Depok. Pada sistem penjualan ini *Admin* memiliki fungsi yang dapat diakses.



Sumber: Hasil Penelitian

Gambar IV. 2 Use Case Diagram Admin

Tabel IV.1
Table Use Case Diagram Login

<i>Use Case Name</i>	:	Login sistem
<i>Use Case Deskripsi</i>	:	Member dan Admin dapat melakukan login ke dalam sistem untuk mengakses berbagai fitur yang tersedia.
<i>Actors</i>	:	Admin
<i>Pre-Condition</i>	:	Sistem harus terhubung ke jaringan Mengakses form login Masukkan <i>username</i> dan <i>password</i>
<i>Post-Condition</i>	:	Apabila proses validasi berhasil dilakukan dan akses login diterima, pengguna dapat memasuki sistem tersebut
<i>Fault Condition</i>	:	Apabila hasil validasi tidak memenuhi kriteria yang ditetapkan, sistem akan melakukan penolakan dan dengan demikian proses tersebut akan dianggap selesai
<i>Main Scenarios</i>	Serial No.	Step
Admin dan Member	1	Masukkan <i>Username</i> Masukkan <i>Password</i>
	2	Pengecekan <i>username</i> dan <i>password</i>
	3	Memberikan akses terhadap sistem berdasarkan hak yang dimiliki oleh setiap pengguna sesuai perannya.

Tabel IV. 2
Table Use Case Diagram Pilih Produk

<i>Use Case Name</i>	:	Pilih Produk
<i>Use Case Deskripsi</i>	:	Member memilih produk dari daftar menu yang tersedia dan menentukan jumlah pembelian sebelum melakukan transaksi pemesanan.
<i>Actors</i>	:	Member
<i>Pre-Condition</i>	:	Member sudah login ke sistem Data menu produk sudah tersedia di database
<i>Post-Condition</i>	:	Daftar produk yang dipilih beserta jumlah disimpan sementara pada sesi transaksi Data siap digunakan untuk proses checkout atau konfirmasi pesanan
<i>Fault Condition</i>	:	Jika data produk tidak tersedia, sistem akan menampilkan status pesanan dibatalkan
<i>Main Scenarios</i>	Serial No.	Step
<i>Member</i>	1	member memilih produk dan jumlah
	2	Member mengisi data pengiriman
	3	Sistem menghitung harga dan ongkir
	4	Member melakukan konfirmasi
	5	Sistem menyimpan transaksi

Tabel IV. 3
Table Use Case Diagram Checkout

<i>Use Case Name</i>	:	Checkout
<i>Use Case Deskripsi</i>	:	Member melakukan pengecekan akhir terhadap produk yang dipilih, mengisi data pengiriman, dan konfirmasi pesanan sebelum transaksi disimpan.
<i>Actors</i>	:	Member
<i>Pre-Condition</i>	:	Member sudah login ke sistem Produk sudah dipilih dan tersimpan di order
<i>Post-Condition</i>	:	Data pesanan tersimpan di tabel orders dengan status pending Member mendapatkan notifikasi konfirmasi pesanan
<i>Fault Condition</i>	:	Jika terjadi kesalahan koneksi database, transaksi gagal disimpan. Jika data produk tidak tersedia, sistem akan menampilkan status pesanan dibatalkan
<i>Main Scenarios</i>	Serial No.	Step
<i>Member</i>	1	Member membuka halaman order

	2	Sistem menampilkan daftar produk terpilih, jumlah, subtotal.
	3	<i>Member</i> mengisi data pengiriman
	4	Sistem menghitung ongkir berdasarkan jarak
	5	<i>Member</i> meninjau total pembayaran
	6	<i>Member</i> mengonfirmasi pesanan
	7	Sistem menyimpan transaksi

Tabel IV. 4

Table Use Case Diagram Melakukan Pembayaran

<i>Use Case Name</i>	:	Melakukan Pembayaran
<i>Use Case Deskripsi</i>	:	<i>Member</i> melakukan pembayaran atas transaksi pesanan yang sudah dikonfirmasi saat checkout, kemudian <i>Admin</i> memverifikasi pembayaran dan memperbarui status pesanan.
<i>Actors</i>	:	<i>Member, Admin</i>
<i>Pre-Condition</i>	:	Pesanan sudah dibuat dengan status pending Informasi total pembayaran tersedia dan dapat diakses <i>Member</i>
<i>Post-Condition</i>	:	Status pesanan diperbarui menjadi proses atau selesai setelah pembayaran diverifikasi <i>Admin</i> . <i>Member</i> menerima notifikasi status pembayaran
<i>Fault Condition</i>	:	Jika pembayaran tidak berhasil dicatat, sistem menampilkan pesan error Jika koneksi ke database gagal, status pembayaran tidak dapat diperbarui.
<i>Main Scenarios</i>	Serial No.	Step
<i>Member</i>	1	<i>Member</i> membuka halaman order
	2	Sistem menampilkan detail pesanan dan total yang harus dibayar
	3	<i>Member</i> memilih metode pembayaran
	4	<i>Member</i> melakukan pembayaran
	5	<i>Member</i> mengunggah bukti pembayaran
	6	Sistem mencatat pembayaran sebagai menunggu verifikasi
	7	<i>Admin</i> memverifikasi pembayaran
	8	<i>Admin</i> mengubah status pesanan menjadi selesai
	9	Sistem mengubah riwayat pesanan ke <i>member</i>

Tabel IV. 5
Table Use Case Diagram Konfirmasi Pembayaran

<i>Use Case Name</i>	:	Melakukan Pembayaran
<i>Use Case Deskripsi</i>	:	<i>Member</i> melakukan pembayaran atas pesanan yang telah dibuat, kemudian mengunggah bukti pembayaran untuk diverifikasi oleh <i>Admin</i> sebelum pesanan diproses lebih lanjut.
<i>Actors</i>	:	<i>Member</i>
<i>Pre-Condition</i>	:	Pesanan sudah dibuat dan berstatus <i>pending</i> . <i>Member</i> sudah login ke sistem. Informasi total pembayaran tersedia.
<i>Post-Condition</i>	:	Bukti pembayaran tercatat di sistem
<i>Fault Condition</i>	:	Jika file bukti pembayaran gagal diunggah, sistem menampilkan pesan <i>error</i>
<i>Main Scenarios</i>	Serial No.	Step
<i>Member</i>	1	<i>member</i> mengklik “pesan sekarang”
	2	Sisitem akan menampilkan detail pesanan dan total pembayaran
	3	<i>Member</i> memilih pembayaran
	4	<i>Member</i> melakukan transfer pembayaran
	5	<i>Member</i> mengunggah bukti pembayaran
	6	Sistem menyimpan data bukti pembayaran
	7	<i>Admin</i> mengubah status pesanan menjadi selesai
	8	Sistem akan mengubah status pesanan di riwayat pesanan <i>member</i>

Tabel IV. 6
Table Use Case Diagram Menerima Invoice

<i>Use Case Name</i>	:	Menerima invoice
<i>Use Case Deskripsi</i>	:	Setelah pesanan dikonfirmasi checkout, sistem secara otomatis menampilkan invoice
<i>Actors</i>	:	<i>Member</i>
<i>Pre-Condition</i>	:	Pesanan sudah dikonfirmasi checkout oleh <i>member</i> <i>Member</i> sudah login dan memiliki data transaksi
<i>Post-Condition</i>	:	Invoice berhasil dibuat dan ditampilkan kepada <i>member</i>
<i>Fault Condition</i>	:	Jika terjadi kesalahan saat membuat invoice, sistem menampilkan pesan error.
<i>Main Scenarios</i>	Serial No.	Step

<i>Member</i>	1	<i>Member menyelesaikan checkout</i>
	2	Sistem membuat invoice transaksi secara otomatis
	3	Sistem menampilkan invoice di layar

Tabel IV. 7
Table Use Case Diagram Mengelola Pesanan

<i>Use Case Name</i>	:	Mengelola Pesanan
<i>Use Case Deskripsi</i>	:	<i>Admin</i> memproses dan memverifikasi pesanan yang masuk, mengubah status pesanan sesuai tahapan pengiriman atau pembayaran, serta membatalkan pesanan jika diperlukan.
<i>Actors</i>	:	<i>admin, Admin</i>
<i>Pre-Condition</i>	:	Pesanan dibuat oleh <i>member</i> dan tersimpan di tabel orders <i>Admin</i> sudah login ke sistem
<i>Post-Condition</i>	:	Status pesanan diperbarui <i>Member</i> menerima perubahan status pesanan di halaman riwayat
<i>Fault Condition</i>	:	Jika koneksi database gagal, status pesanan tidak tersimpan
<i>Main Scenarios</i>	Serial No.	Step
<i>Member</i>	1	<i>Admin</i> membuka halaman daftar pesanan
	2	Sistem menampilkan daftar pesanan dan status terkini
	3	<i>Admin</i> memilih pesanan yang akan diperbarui
	4	<i>Admin</i> memilih status baru
	5	<i>Admin</i> menyimpan perubahan
	6	Sistem memperbarui status pesanan

Tabel IV. 8
Table Use Case Diagram Mengelola Laporan Penjualan

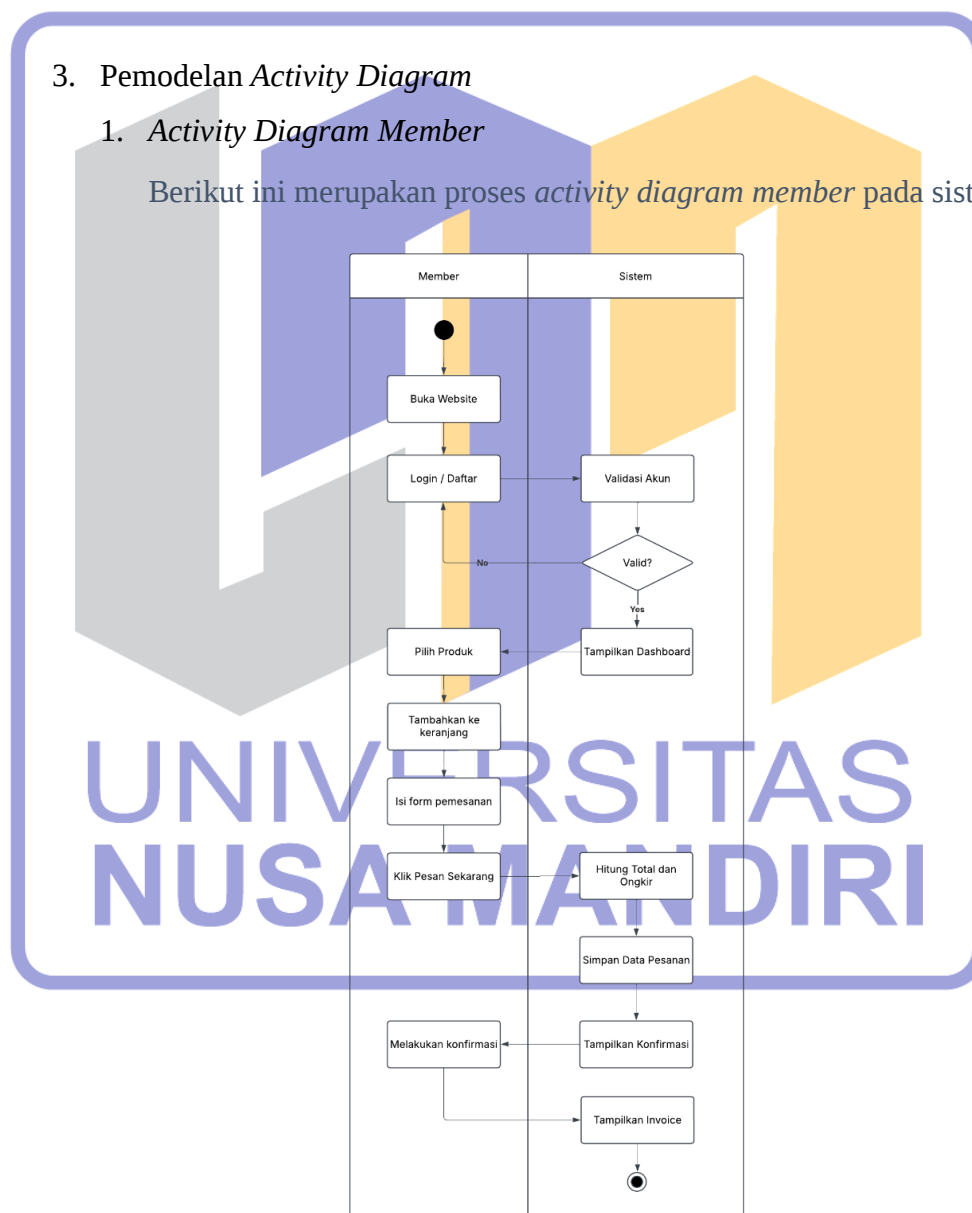
<i>Use Case Name</i>	:	Mengelola Laporan Penjualan
<i>Use Case Deskripsi</i>	:	<i>Admin</i> melihat dan mencetak laporan penjualan berdasarkan data pesanan yang sudah tersimpan di sistem dalam periode tertentu (harian, bulanan, tahunan).
<i>Actors</i>	:	<i>Admin</i>
<i>Pre-Condition</i>	:	Data pesanan sudah tersedia di tabel order <i>Admin</i> sudah login ke sistem
<i>Post-Condition</i>	:	Laporan penjualan berhasil ditampilkan di layar

<i>Fault Condition</i>	:	Jika terjadi kegagalan query data, laporan tidak bisa ditampilkan
<i>Main Scenarios</i>	Serial No.	Step
<i>Member</i>	1	<i>Admin</i> membuka halaman laporan penjualan
	2	Sistem menampilkan filter periode laporan
	3	<i>Admin</i> memilih periode waktu
	4	Sistem mengambil data transaksi sesuai filter
	5	Sistem menampilkan laporan penjualan di layar

3. Pemodelan Activity Diagram

1. Activity Diagram Member

Berikut ini merupakan proses *activity diagram member* pada sistem

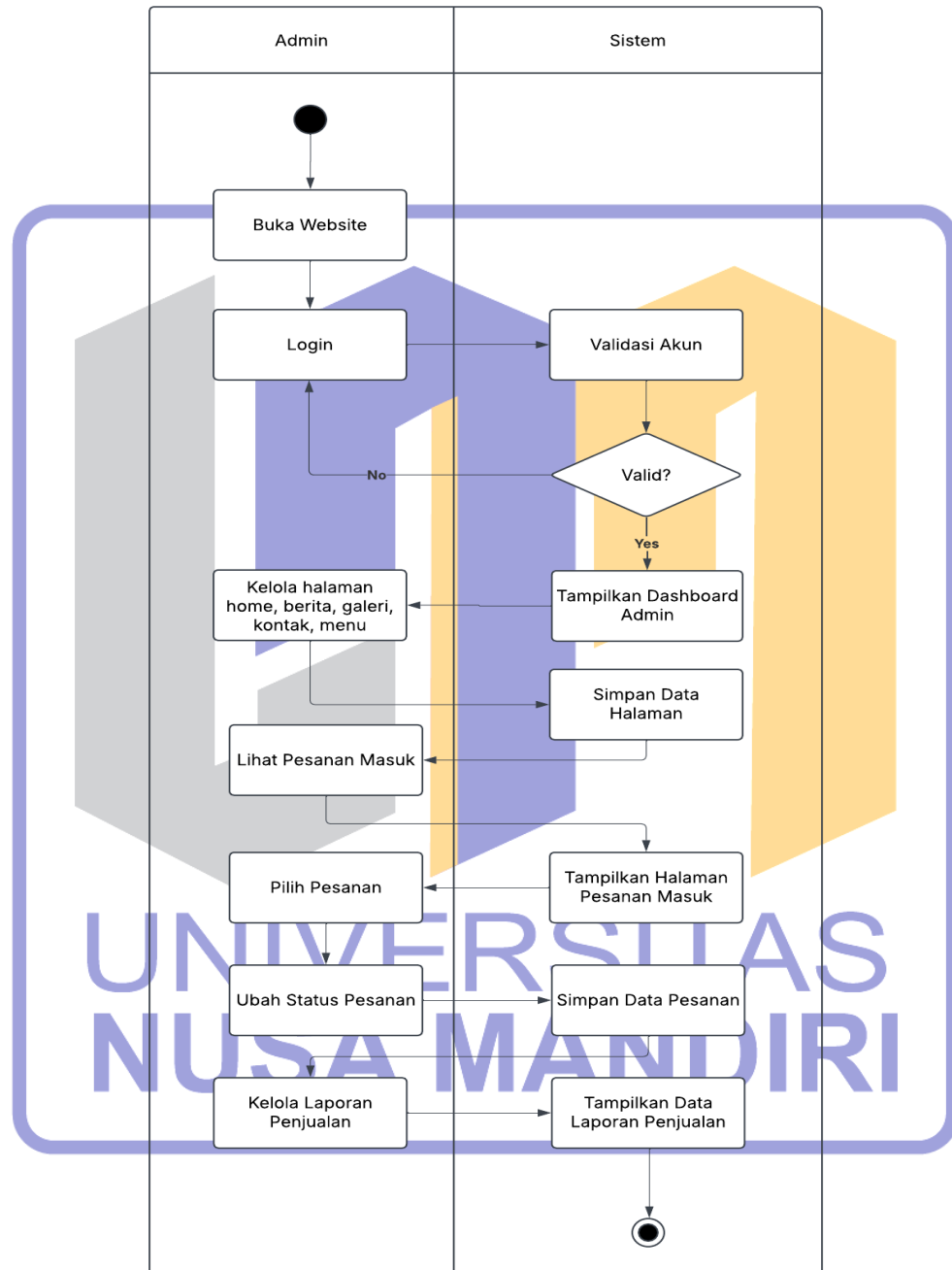


Sumber: Hasil Penelitian

Gambar IV. 3 Activity Diagram Member

2. Activity Diagram Admin

Berikut ini merupakan proses *activity diagram admin* pada sistem.

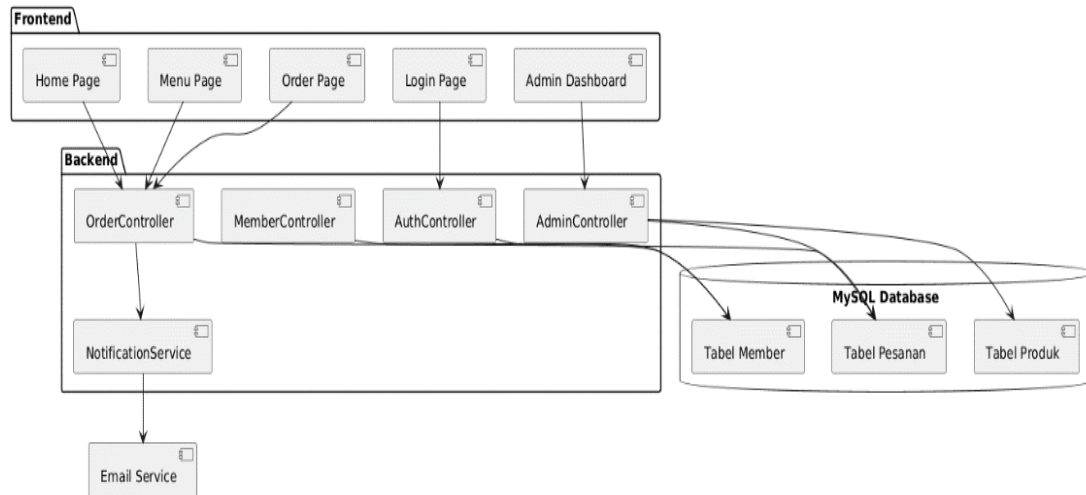


Sumber: Hasil Penelitian

Gambar IV. 4 Activity Diagram Admin

4. Component Diagram

Berikut ini merupakan ilustrasi hubungan antar komponen dalam sistem yang disajikan melalui diagram komponen (*component diagram*).

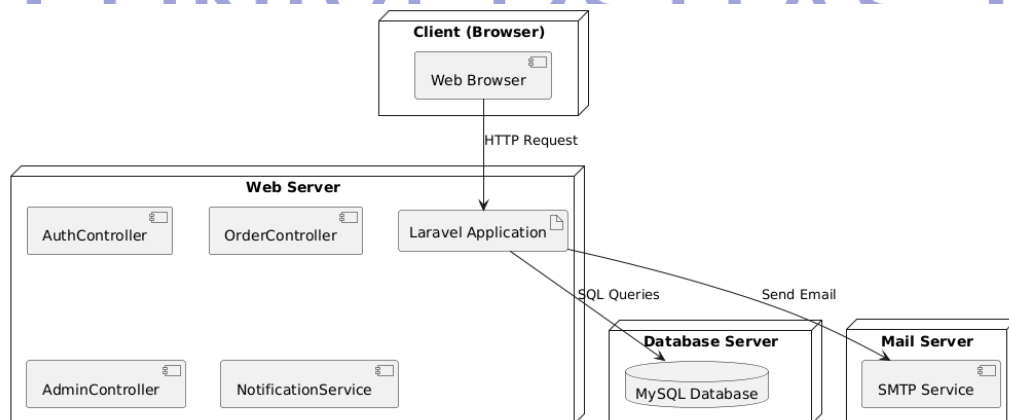


Sumber: Hasil Penelitian

Gambar IV. 5 Component Diagram

5. Deployment Diagram

Berikut merupakan representasi distribusi komponen dan perangkat keras yang digunakan dalam sistem, yang divisualisasikan melalui *deployment diagram*.



Sumber: Hasil Penelitian

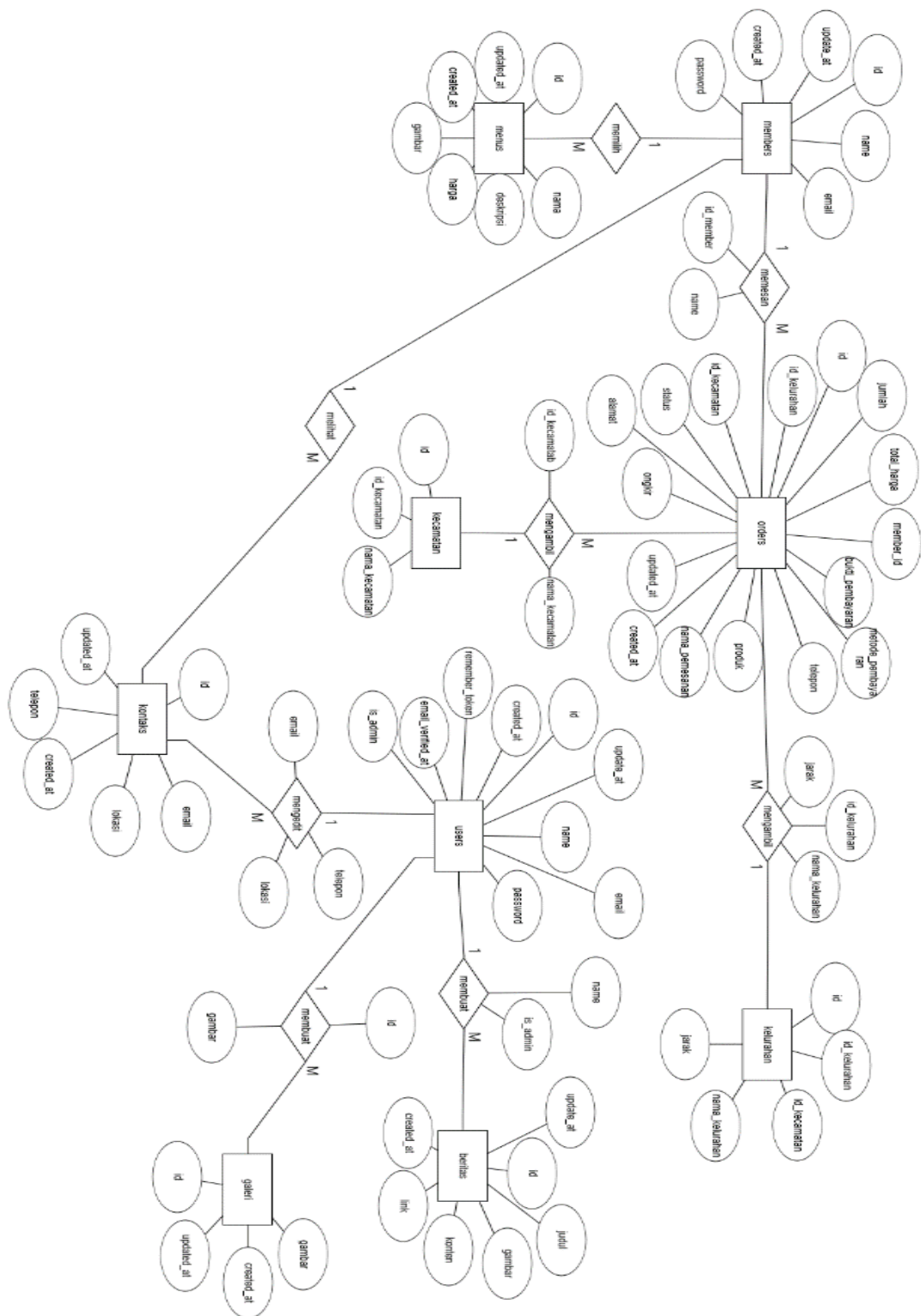
Gambar IV. 6 Deployment Diagram

4.2.2 Desain Pemodelan Data

A. *Entity Relationship Diagram*

Berikut ini merupakan hubungan entitas antar tabel dalam database sakupaleza-main. *Members* dapat memesan banyak orders, *members* dapat memilih banyak menu, *member* dapat melihat kontak. Banyak order dapat mengambil satu kelurahan dan satu kecamatan. *Users* dapat membuat banyak berita, *users* dapat membuat banyak galeri, *user* dapat mengedit banyak kontak.



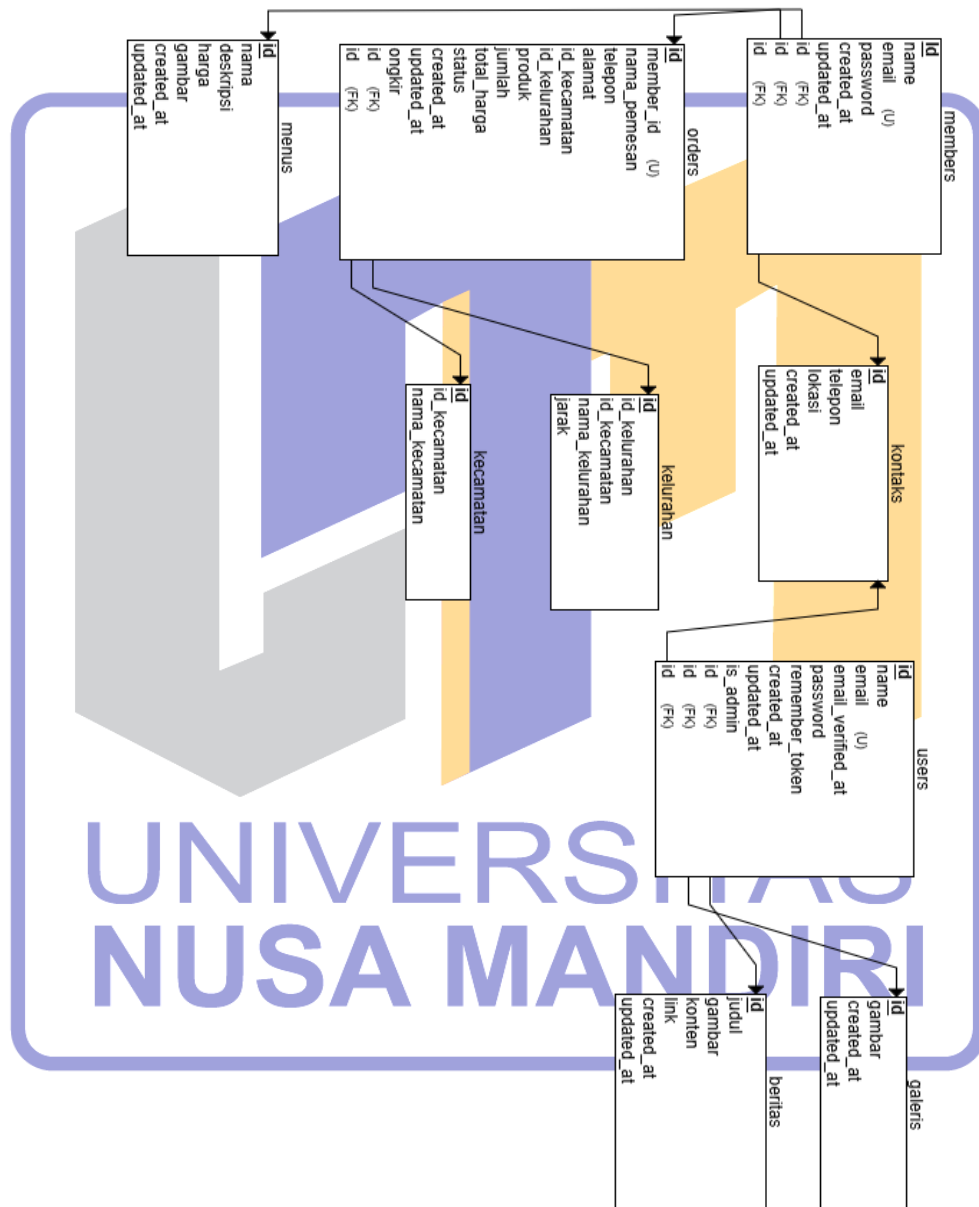


Sumber: Hasil Penelitian

Gambar IV. 7 Entity Relationship Diagram

B. Logical Record Structure

Berikut ini menjelaskan proses perubahan data dari tingkat abstrak ke representasi *logical record*.



Sumber: Hasil Penelitian

Gambar IV. 8 Logical Record Structure

C. Spesifikasi File Database

1. Deskripsi tabel *users*

Nama database : Sakupaleza-main

Nama table : *users*

Primary field : *id*

Tabel IV. 9
Tabel *Users*

No	Nama Field	Jenis Data	Besaran Field	Tipe Key
1	<i>Id</i>	Bigint	20	Primary
2	Name	Varchar	255	
3	Email	Varchar	255	
4	Email_verified_at	Timestamp	0	
5	Password	Varchar	255	
6	Remember_token	Varchar	100	
7	Created_at	Timestamp	0	
8	Updated_at	Timestamp	0	
9	Is_admin	Tinyint	1	

2. Deskripsi tabel *member*

Nama database : Sakupaleza-main

Nama table : *members*

Primary field : *id*

Tabel IV. 10
Tabel *members*

No	Nama Field	Jenis Data	Besaran Field	Tipe Key
1	<i>Id</i>	Bigint	20	Primary
2	Name	Varchar	255	
3	Email	Varchar	255	
4	Password	Varchar	255	
5	Created_at	Timestamp	0	
6	Updated_at	Timestamp	0	

3. Deskripsi tabel galeri

Nama database : Sakupaleza-main

Nama table : *galeris*

Primary field : *id*

Tabel IV. 11
Tabel *Galeris*

No	Nama Field	Jenis Data	Besaran Field	Tipe Key
1	Id	Bigint	20	Primary
2	gambar	Varchar	255	
3	Created_at	Timestamp	0	
4	Updated_at	Timestamp	0	

4. Deskripsi tabel berita

Nama database : Sakupaleza-main

Nama table : beritas

Primary field : id

Tabel IV. 12
Tabel beritas

No	Nama Field	Jenis Data	Besaran Field	Tipe Key
1	Id	Bigint	20	Primary
2	judul	Varchar	255	
3	Gambar	Text	0	
4	Konten	Text	0	
5	Link	Text	0	
6	Created_at	Timestamp	0	
7	Updated_at	Timestamp	0	

5. Deskripsi tabel kecamatan

Nama database : Sakupaleza-main

Nama table : kecamatan

Primary field : id

Tabel IV. 13
Tabel kecamatan

No	Nama Field	Jenis Data	Besaran Field	Tipe Key
1	Id	Bigint	11	Primary
2	Id_kecamatan	int	6	
3	Nama_kecamatan	Varchar	15	

6. Deskripsi tabel kelurahan

Nama database : Sakupaleza-main

Nama table : kelurahan

Primary field : id

Tabel IV. 14
Tabel Kelurahan

No	Nama Field	Jenis Data	Besaran Field	Tipe Key
1	Id	Int	11	Primary
2	Id_kelurahan	Int	11	
3	Id_kecamatan	Int	6	
4	Nama_kelurahan	Varchar	25	
5	Jarak	Varchar	5	

7. Deskripsi tabel orders

Nama database : Sakupaleza-main

Nama table : orders

Primary field : id

Tabel IV. 15
Tabel Orders

No	Nama Field	Jenis Data	Besaran Field	Tipe Key
1	Id	Bigint	20	Primary
2	Member_id	Bigint	20	
3	Nama_pemesan	Varchar	255	
4	Telepon	Varchar	255	
5	Alamat	Text	0	
6	Id_kecamatan	Int	6	
7	Id_kelurahan	Int	11	
8	Produk	Varchar	255	
9	Jumlah	Int	11	
10	Total_harga	Decimal	10,2	
11	Metode_pembayaran	Varchar	255	
12	Bukti_pembayaran	Varchar	255	
13	Status	Enum	'pending' 'proses' 'selesai' 'dibatalkan'	
14	Created_at	Timestamp	0	
15	Updated_at	Timestamp	0	
16	Ongkir	Int	11	

8. Deskripsi tabel kontak

Nama database : Sakupaleza-main
 Nama table : kontak
 Primary field : id

Tabel IV. 16
Tabel Kontak

No	Nama Field	Jenis Data	Besaran Field	Tipe Key
1	Id	Bigint	20	Primary
3	Email	Varchar	255	
5	Telepon	Varchar	255	
6	lokasi	Varchar	100	
7	Created_at	Timestamp	0	
8	Updated_at	Timestamp	0	

9. Deskripsi tabel menu

Nama database : Sakupaleza-main
 Nama table : menus
 Primary field : id

Tabel IV. 17
Tabel Menus

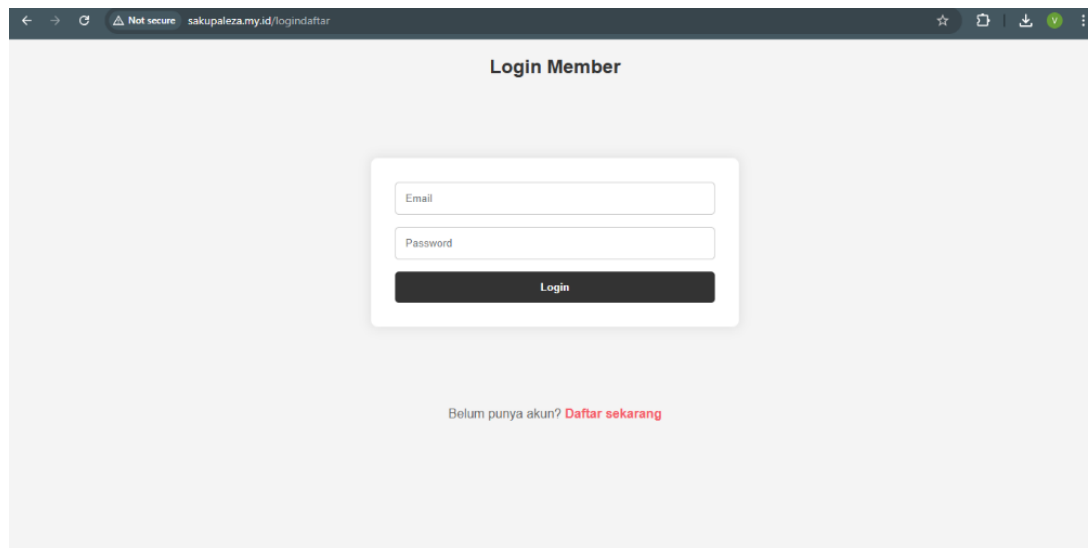
No	Nama Field	Jenis Data	Besaran Field	Tipe Key
1	Id	Bigint	20	Primary
2	Nama	Varchar	255	
3	Deskripsi	Text	255	
4	Harga	Decimal	10,2	
5	gambar	Varchar	255	
7	Created_at	Timestamp	0	
8	Updated_at	Timestamp	0	

4.2.3 Desain User Interface

A. Halaman login member

Berikut ini adalah tampilan halaman *Login Member* pada sistem Sakupaleza. Halaman ini berfungsi sebagai pintu masuk bagi pengguna yang ingin melakukan pemesanan produk, di mana proses login menjadi syarat utama sebelum dapat mengakses fitur pemesanan. Pengguna diminta untuk

mengisi email dan kata sandi yang telah terdaftar. Bagi yang belum memiliki akun, tersedia tautan pendaftaran untuk membuat akun baru.



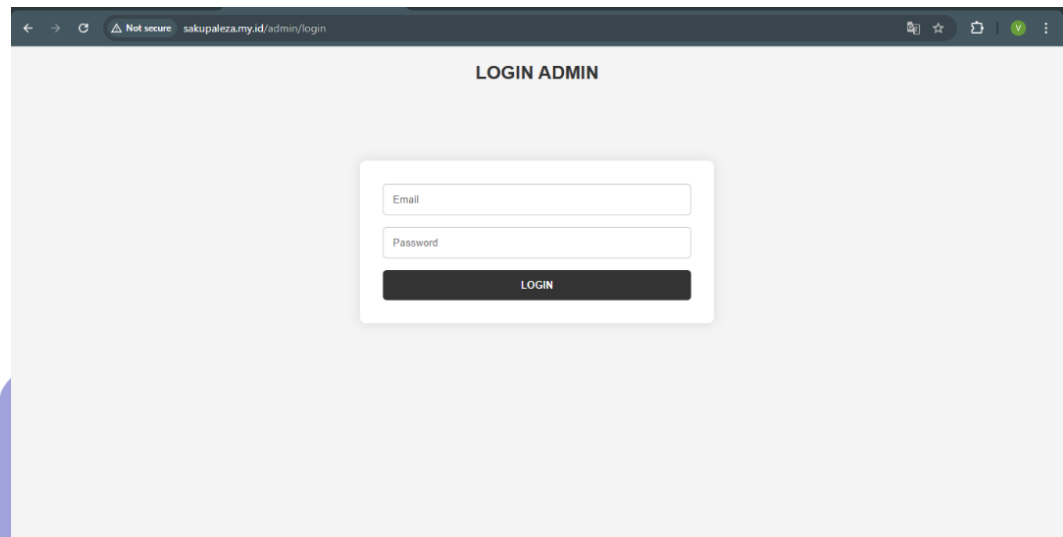
The screenshot shows a web browser window with the address bar displaying 'sakupaleza.my.id/logindaftar'. The page title is 'Login Member'. The main content area contains a white login form with two input fields labeled 'Email' and 'Password', and a black 'Login' button. Below the form, there is a text link: 'Belum punya akun? [Daftar sekarang](#)'.

Sumber: Hasil Penelitian

Gambar IV. 9 Login Member

B. Halaman *Login Admin*

Berikut ini merupakan tampilan halaman *Login Admin* pada sistem Sakupaleza. Halaman ini berfungsi sebagai gerbang masuk bagi pengguna dengan peran sebagai *administrator*, yang bertanggung jawab dalam mengelola data dan memantau aktivitas sistem. Untuk dapat mengakses *dashboard admin*, pengguna diwajibkan mengisi alamat email dan kata sandi yang telah terdaftar sebelumnya. Proses login ini dirancang untuk memastikan keamanan dan membatasi akses hanya kepada pihak yang memiliki otorisasi.

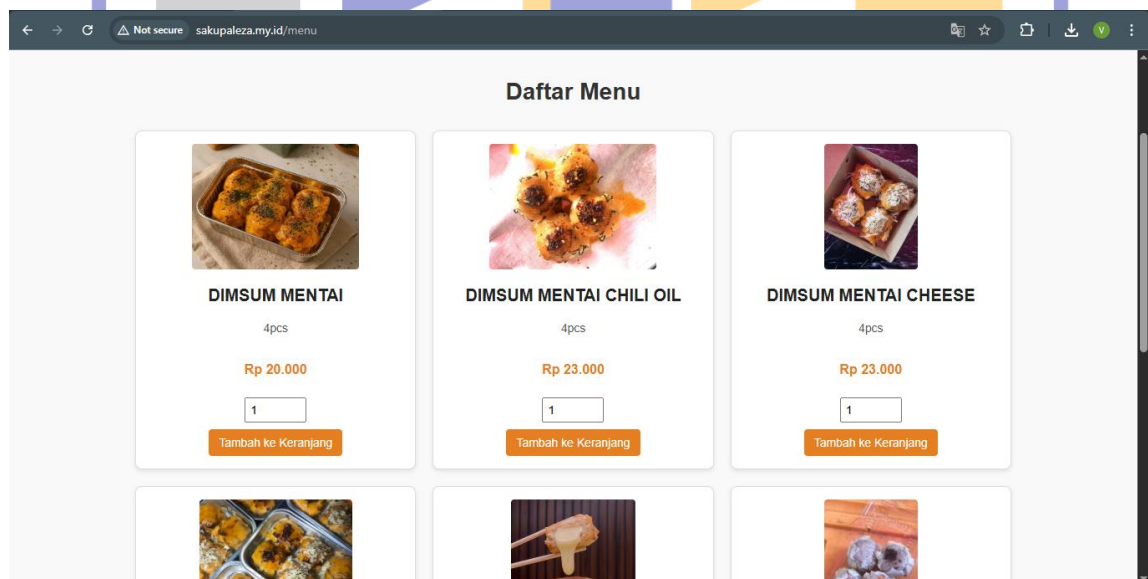


Sumber: Hasil Penelitian

Gambar IV. 10 Login Admin

C. Halaman Menu

Berikut ini adalah halaman daftar menu yang dapat diakses oleh *member*.



Sumber: Hasil Penelitian

Gambar IV. 11 Menu

D. Halaman Form Pemesanan

Berikut ini merupakan halaman form pemesanan yang dapat diakses *member* yang sudah login.

Form Pemesanan

Produk berhasil ditambahkan ke keranjang.

[+ Tambah Menu Lagi](#)

Produk	Harga	Jumlah	Subtotal
DIMSUM MENTAI CHILI OIL	Rp 23.000	1	Rp 23.000
Total			Rp 23.000
Ongkir			Rp 6.000
Grand Total			Rp 29.000

aditya 0895333026460

Jl. Pinang 3 No 9A RT.002 RW.003 Beji Kelurahan: Beji

Pesan Sekarang

Sumber: Hasil Penelitian

Gambar IV. 12 Form Pemesanan

E. Halaman Pembayaran

Berikut ini merupakan halaman intruksi pembayaran untuk *member* setelah melakukan checkout.

Instruksi Pembayaran

Total Bayar:
Rp 29.000

Waktu tersisa untuk bayar:
14:39

Pilih Metode Pembayaran:
GoPay

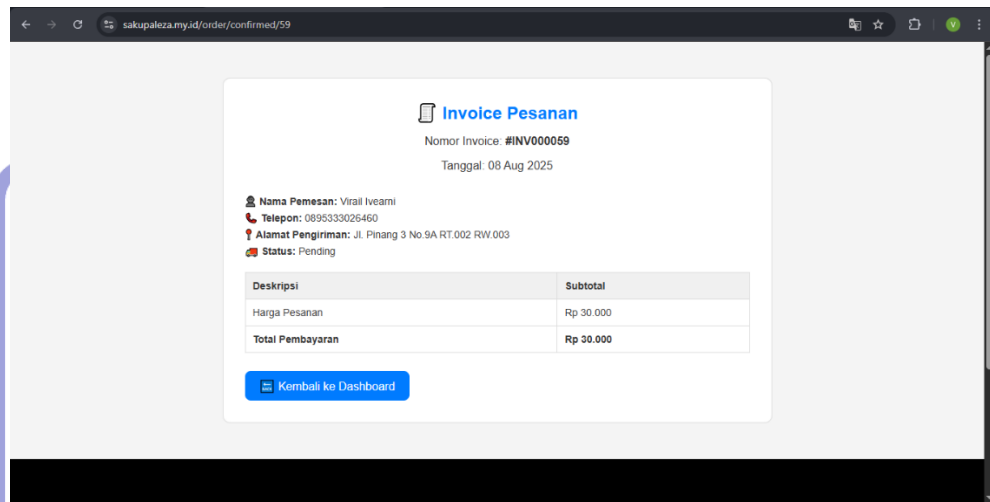
Pembayaran via GoPay:
Nomor: 0895-3330-26460
Atau scan QR:

Sumber: Hasil Penelitian

Gambar IV. 13 Pembayaran

F. Halaman Invoice

Berikut ini adalah halaman invoice yang didapat oleh *member* setelah melakukan pembayaran.

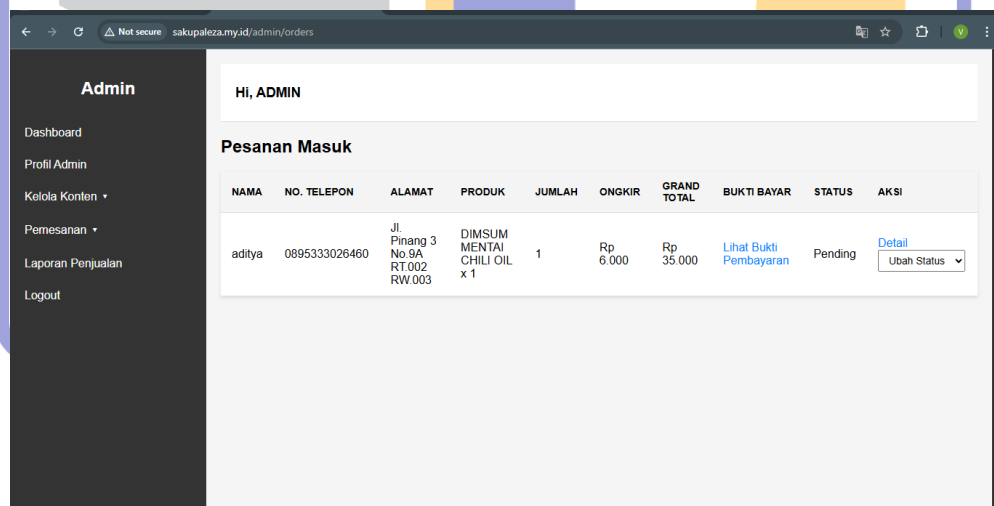


Sumber: Hasil Penelitian

Gambar IV. 14 Invoice

G. Halaman Pesanan Masuk

Berikut ini merupakan halaman pesanan masuk yang dapat dikelola oleh *admin*.



Sumber: Hasil Penelitian

Gambar IV. 15 Pesanan Masuk

H. Halaman Riwayat Pesanan

Berikut ini adalah halaman riwayat pesanan yang dapat diakses oleh *admin*.

Hi, ADMIN

Riwayat Pesanan

Filter Status: -- Semua Status --

Nama Pemesan	Produk	Jumlah	Total Harga	Status	Tanggal Update
aditya	DIMSUM MENTAI CHILI OIL x 1	1	Rp 29.000	Selesai	11-07-2025 17:21
janu	DIMSUM MENTAI CHILI OIL x 1	1	Rp 29.000	Dibatalkan	11-07-2025 17:19
lala	DIMSUM MENTAI CHILI OIL x 1	1	Rp 29.000	Selesai	11-07-2025 17:14
nisa	DIMSUM MENTAI CHILI OIL x 1	1	Rp 29.000	Dibatalkan	09-07-2025 14:53
lala	dimsum x 1	1	Rp 36.000	Selesai	09-07-2025 05:57
janu	dimsum x 2	2	Rp 60.000	Selesai	08-07-2025 12:16
arfian	dimsum x 1	1	Rp 30.000	Selesai	08-07-2025 12:16
arfian	esteh x 1, dimsum x 1	2	Rp 40.000	Selesai	08-07-2025 12:16
arfian	dimsum x 1	1	Rp 36.000	Selesai	08-07-2025 12:16

Sumber: Hasil Penelitian

Gambar IV. 16 Riwayat Pesanan

I. Halaman Laporan Penjualan

Berikut ini adalah halaman laporan penjualan yang dapat diakses oleh *admin*.

Hi, ADMIN

Laporan Penjualan

Dari Tanggal: Sampai Tanggal: Filter

Total Pendapatan: Rp 1.826.000

Tanggal: 30-06-2025

Nama	Produk	Jumlah	Total
pirel	Produk 2 x 1, Produk 1 x 1	2	Rp 40.000
pirel	Produk 3 x 1	1	Rp 20.000
Total Tanggal Ini			Rp 60.000

Tanggal: 01-07-2025

Nama	Produk	Jumlah	Total
pirel	Produk 3 x 1	1	Rp 20.000

Sumber: Hasil Penelitian

Gambar IV. 17 Laporan Penjualan

4.3 Code Generation

A. OrderController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Order;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;

class OrderController extends Controller
{
    public function index()
    {
        $orders = Order::where('member_id', Auth::guard('member')->id())
            ->orderBy('created_at', 'desc')
            ->get();

        return view('order.index', compact('orders'));
    }

    public function create()
    {
        $cart = session('cart', []);
        $kecamatans = DB::table('kecamatan')->get();
        return view('order.create', compact('cart', 'kecamatans'));
    }

    public function store(Request $request)
    {
        if (!session()->has('order_preview')) {
            return redirect()->route('order.create')->with('error', 'Session pemesanan tidak
ditemukan.');
```

```

}

$data = session('order_preview');

$request->validate([

    'metode_pembayaran' => 'required|in:bca,ovo,gopay,shopeepay,gris',

    'bukti_pembayaran' => 'required|image|mimes:jpg,jpeg,png|max:2048',

]);

$bukti = $request->file('bukti_pembayaran')->store('bukti-pembayaran', 'public');

// Ambil kelurahan

$kecamatan = DB::table('kecamatan')->where('id_kecamatan', $request->kecamatan)->first();

if (!$kecamatan) {

    return back()->withErrors(['kecamatan' => 'Data kecamatan tidak ditemukan.']);

}

// Ambil kelurahan

$kelurahan = DB::table('kelurahan')->where('id_kelurahan', $request->kelurahan)->first();

if (!$kelurahan) {

    return back()->withErrors(['kelurahan' => 'Data kelurahan tidak ditemukan.']);

}

// Hitung ongkir

$tarifPerKm = 1000;

$ongkir = $kelurahan->jarak * $tarifPerKm;

$produk = [];

$totalHarga = 0;

$totalQty = 0;

foreach ($data['menu'] as $i => $menu) {

    $harga = $data['harga'][$i];

    $jumlah = $data['jumlah'][$i];

```

```

$produk[] = "{$menu} x {$jumlah}";

$totalHarga += $harga * $jumlah + $ongkir;

$totalQty += $jumlah;
}

if (!isset($data['menu'], $data['harga'], $data['jumlah'], $data['nama_pemesan'],
$data['telepon'], $data['kelurahan'], $data['kecamatan'], $data['alamat'])) {

    return redirect()->route('order.create')->with('error', 'Data pemesanan tidak
    lengkap.');
```



```

}

// Simpan ke orders
$order = Order::create([
    'member_id' => Auth::guard('member')->id(),
    'nama_pemesan' => $data['nama_pemesan'],
    'telepon' => $data['telepon'],
    'alamat' => $data['alamat'],
    'id_kelurahan' => $data['kelurahan'],
    'id_kecamatan' => $data['kecamatan'],
    'produk' => implode(', ', $produk),
    'jumlah' => $totalQty,
    'total_harga' => $totalHarga,
    'ongkir' => $ongkir,
    'status' => 'dibayar',
    'metode_pembayaran' => $request->metode_pembayaran,
    'bukti_pembayaran' => $bukti,
]);

//Hapus session agar tidak kembali ke payment
session()->forget(['order_preview', 'cart', 'payment_deadline']);

```

```

        return redirect()->route('order.confirmed', ['id' => $order->id]);
    }

    public function history()
    {
        $orders = Order::where('member_id', Auth::guard('member')->id())->latest()->get();

        return view('member.dashboard', compact('orders'));
    }

    public function previewPayment(Request $request) {
        $request->validate([
            'menu' => 'required|array',
            'harga' => 'required|array',
            'jumlah' => 'required|array',
            'nama_pemesan' => 'required|string',
            'telepon' => 'required|string',
            'kelurahan' => 'required|exists:kelurahan,id_kelurahan',
            'kecamatan' => 'required|exists:kecamatan,id_kecamatan',
            'ongkir' => 'required|numeric|min:0',
        ]);
        session([
            'order_preview' => $request->all()
        ]);
        return redirect()->route('order.payment.page');
    }

    public function payment(Order $order)
    {
        return view('order.payment', compact('order'));
    }

```

```

public function paymentPage()
{
    // Cek dulu: ada session order atau tidak?
    if (!session()->has('order_preview')) {
        return redirect()->route('order.create')->with('error', 'Data pembayaran tidak
ditemukan.');
```



```

    }
    // Ambil datanya setelah dipastikan ada
    $order = session('order_preview');
    // Timer batas pembayaran: 15 menit dari awal (bukan dari sekarang terus)
    if (!session()->has('payment_deadline')) {
        session(['payment_deadline' => now()->addMinutes(15)->timestamp]);
    }
    $deadline = session('payment_deadline');
    $remaining = $deadline - now()->timestamp;
    if ($remaining <= 0) {
        session()->forget(['payment_deadline', 'order_preview', 'cart']);
        return redirect()->route('order.create')->with('error', 'Waktu pembayaran telah
habis.');
```

```

    }
    // Lanjut render view
    $items = [];
    $total = 0;

    foreach ($order['menu'] as $i => $nama) {
        $harga = $order['harga'][$i];
        $jumlah = $order['jumlah'][$i];
        $subtotal = $harga * $jumlah;
        $total += $subtotal;
    }
}

```

```

        $items[] = compact('nama', 'harga', 'jumlah', 'subtotal');
    }
    $grandTotal = $total + $order['ongkir'];
    return view('order.payment', compact('order', 'items', 'total', 'grandTotal',
    'remaining'));
}

public function confirmed($id)
{
    $order = Order::findOrFail($id);
    // Pastikan user adalah pemilik pesanan
    if ($order->member_id !== Auth::guard('member')->id()) {
        abort(403);
    }
    return view('order.confirmed', compact('order'));
}

public function confirmPayment(Request $request)
{
    if (!session()->has('order_preview')) {
        return redirect()->route('order.create')->with('error', 'Session pemesanan tidak
        ditemukan.');
```



```

    }
    $data = session('order_preview');
    // Validasi input
    $request->validate([
        'metode_pembayaran' => 'required|in:bca,ovo,gopay,shopeepay,qrisk',
        'bukti_pembayaran' => 'required|image|mimes:jpg,jpeg,png|max:2048',
    ]);
    //Simpan bukti pembayaran
    $bukti = $request->file('bukti_pembayaran')->store('bukti-pembayaran', 'public');
```

```

//Validasi kelurahan dan kecamatan

$kecamatan = DB::table('kecamatan')->where('id_kecamatan', $data['kecamatan'])->first();

if (!$kecamatan) {
    return back()->withErrors(['kecamatan' => 'Data kecamatan tidak ditemukan.']);
}

$kelurahan = DB::table('kelurahan')->where('id_kelurahan', $data['kelurahan'])->first();

if (!$kelurahan) {
    return back()->withErrors(['kelurahan' => 'Data kelurahan tidak ditemukan.']);
}

//Hitung ongkir berdasarkan jarak
$tarifPerKm = 1000;
$ongkir = $kelurahan->jarak * $tarifPerKm;
//Hitung produk
$produk = [];
$totalHarga = 0;
$totalQty = 0;
foreach ($data['menu'] as $i => $menu) {
    $harga = $data['harga'][$i];
    $jumlah = $data['jumlah'][$i];
    $produk[] = "{$menu} x {$jumlah}";

    $totalHarga += $harga * $jumlah;
    $totalQty += $jumlah;
}

//Tambahkan ongkir ke total bayar
$grandTotal = $totalHarga + $ongkir;

```

```

//Simpan order ke DB
$order = Order::create([
    'member_id'      => Auth::guard('member')->id(),
    'nama_pemesan'   => $data['nama_pemesan'],
    'telepon'        => $data['telepon'],
    'alamat'         => $data['alamat'],
    'id_kelurahan'   => $data['kelurahan'],
    'id_kecamatan'   => $data['kecamatan'],
    'produk'         => implode(' ', $produk),
    'jumlah'         => $totalQty,
    'total_harga'    => $grandTotal,
    'ongkir'         => $ongkir,
    'status'         => 'pending',
    'metode_pembayaran' => $request->metode_pembayaran,
    'bukti_pembayaran' => $bukti,
]);
//Bersihkan session
session()->forget(['order_preview', 'cart', 'payment_deadline']);
return redirect()->route('order.confirmed', ['id' => $order->id]);
}
}

```

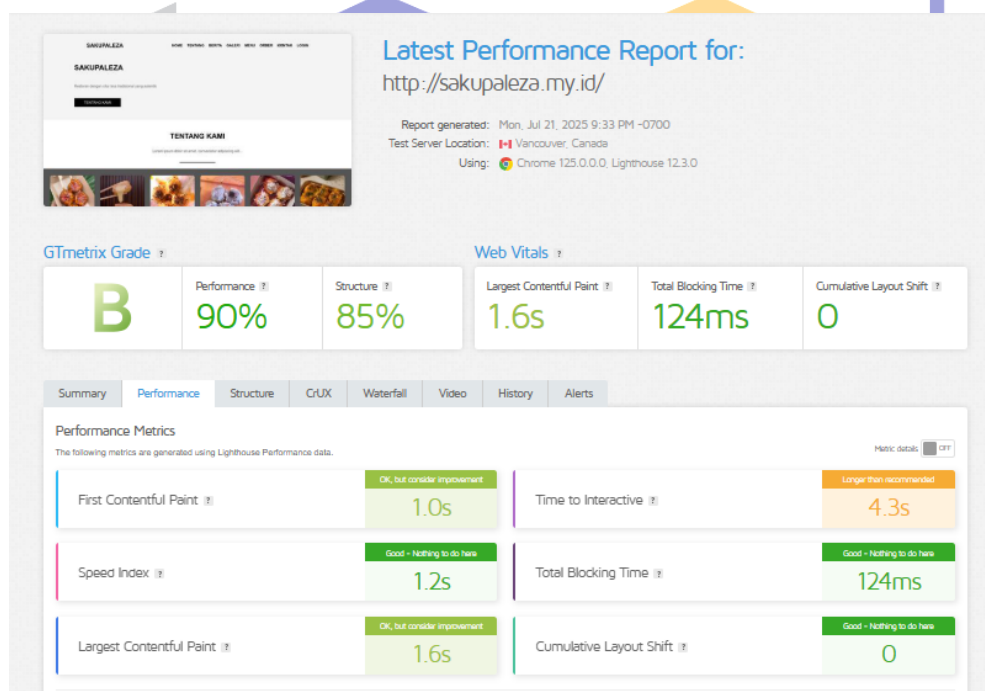
4.4 Testing

4.4.1 Tahap Pengujian Aplikasi

a. Pengujian *Performance*

Hasil pengujian performa *website* sakupaleza.my.id melalui *GTmetrix* menunjukkan kinerja yang baik dengan *GTmetrix Grade B, Performance* 90%, dan *Structure* 85%. *First Contentful Paint* tercapai dalam 1,0 detik,

Speed Index 1,2 detik, dan *Largest Contentful Paint* 1,6 detik. Namun, *Time to Interactive* mencapai 4,3 detik akibat pemrosesan skrip yang cukup berat. Sementara itu, *Total Blocking Time* hanya 124 ms dan *Cumulative Layout Shift* sebesar 0, menunjukkan tampilan yang stabil. Secara umum, *website* memiliki performa baik, namun masih perlu dioptimalkan pada aspek interaktivitas.



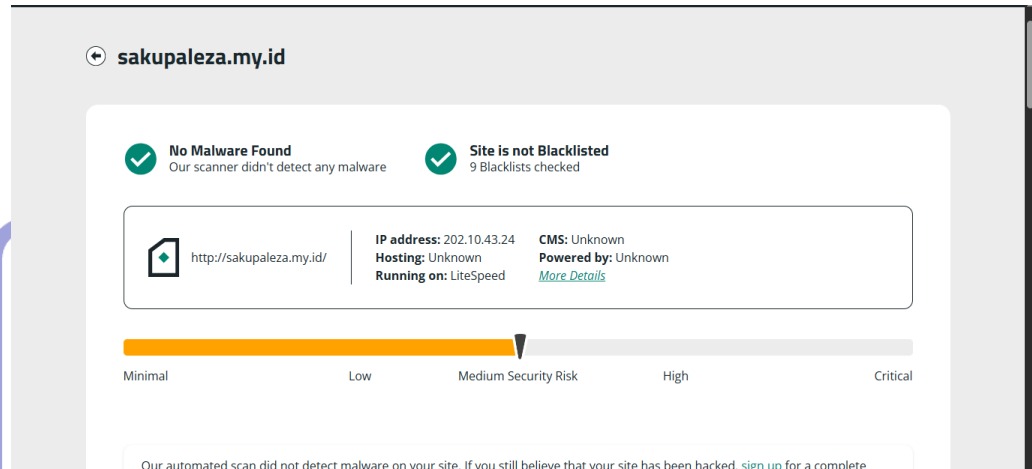
Sumber: Hasil Penelitian

Gambar IV. 18 Pengujian Performance

b. Pengujian Keamanan

Hasil pemindaian keamanan terhadap *website* sakupaleza.my.id menunjukkan bahwa situs ini aman dari *malware* dan tidak masuk daftar blacklist. Namun, *website* tergolong memiliki tingkat risiko keamanan sedang (*Medium Security Risk*), yang mengindikasikan adanya potensi kerentanan yang perlu diwaspadai. Situs menggunakan server *LiteSpeed*, tetapi detail

hosting dan CMS tidak terdeteksi, sehingga disarankan untuk meningkatkan keamanan *server* dan sistem secara keseluruhan.



Sumber: Hasil Penelitian

Gambar IV. 19 Pengujian Keamanan

4.4.2 Tahap Pengujian Kenerimaan Sistem

Tabel IV. 18
User Acceptance Testing

Dokumen <i>User Acceptance Testing</i>			
Nama Proyek	:	Perancangan Sistem Informasi Penjualan Berbasis Web pada Sakupaleza Depok	
Mitra	:	Sakupaleza	
Manajer Proyek	:	Virail Ivearni Septia Virgie Darwanto	

Proses Pengujian					
No	Usecase	Hasil Uji [Berhasil Gagal]	Nama Penguji	Tanggal Pengujian	Catatan Pengujian
1.	Usecase uji: Login Deskripsi: Melakukan verifikasi data login <i>user</i> yang sudah terdaftar Kasus pengujian <i>Username:</i> <i>admin@admin.com</i>	BERHASIL	Afif - <i>Admin</i>	30 Juli 2025	Kurang <i>Toggle Password Visibility</i> dan pemberitahuan <i>username</i> atau password salah

	<i>Password: password</i> Hasil yang diharapkan: 1. Jika <i>user</i> terdaftar dan <i>username</i> serta <i>password</i> sesuai maka berhasil <i>login</i> dan masuk ke aplikasi sesuai dengan <i>role</i>-nya 2. Jika <i>username</i> atau <i>password</i> salah atau tidak ditemukan maka tidak bisa masuk ke sistem.				
2	Usecase uji: Pilih Produk Deskripsi: <i>Member</i> memilih produk dari daftar produk yang tersedia untuk dimasukkan ke keranjang. Kasus pengujian 1. <i>Member</i> dapat melihat daftar produk. 2. <i>Member</i> dapat memilih salah satu produk. 3. Produk yang dipilih masuk ke keranjang. Hasil yang diharapkan: 1. Sistem menampilkan produk secara lengkap. 2. Produk berhasil ditambahkan ke keranjang.	BERHASIL	Syifa - <i>Member</i>	2 Agustus 2025	Tampilan transaksi masih perlu diperbaiki
3.	Usecase uji: CheckOut	BERHASIL	Syifa - <i>Member</i>	02 Agustus 2025	-

	Deskripsi: <i>Member</i> melakukan proses checkout dari produk yang sudah ada di keranjang. Kasus pengujian 1. <i>Member</i> mengakses halaman checkout. 2. Mengisi data pemesanan dan alamat. 3. Mengkonfirmasi pesanan Hasil yang diharapkan: Sistem menyimpan data pemesanan dan menampilkan total pembayaran.				
4.	Usecase uji: Pembayaran Deskripsi: <i>Member</i> melakukan pembayaran sesuai dengan instruksi yang diberikan oleh sistem. Kasus pengujian 1. <i>Member</i> memilih metode pembayaran. 2. Melakukan transfer sesuai nominal. Hasil yang diharapkan: Sistem menampilkan instruksi pembayaran dan jumlah tagihan.	BERHASIL	Syifa - Member	02 Agustus 2025	Tampilan transaksi masih perlu diperbaiki
5.	Usecase uji: Konfirmasi Pembayaran Deskripsi: <i>Member</i> mengunggah bukti		Syifa - Member	02 Agustus 2025	Tampilan masih perlu diperbaiki

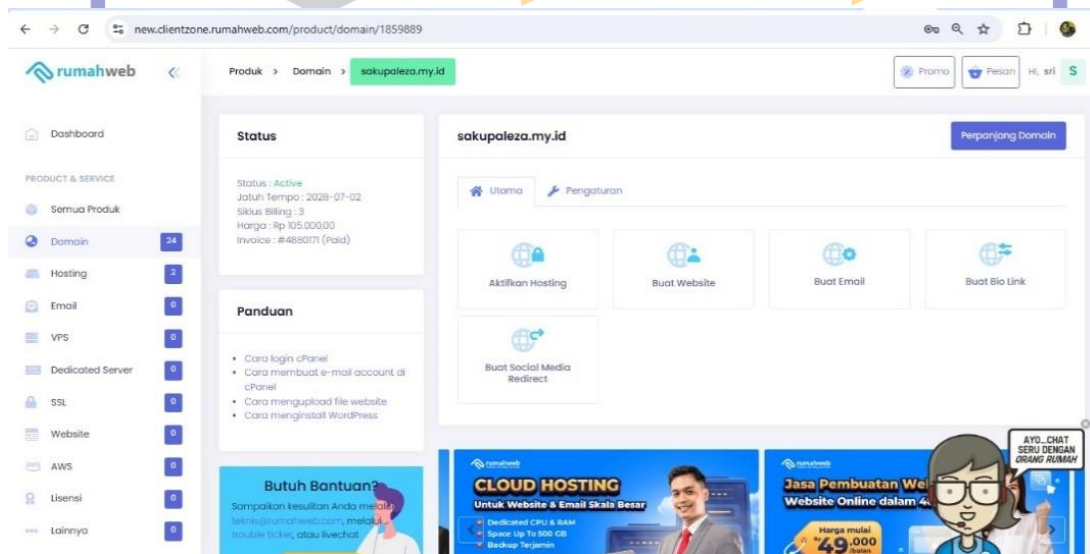
	pembayaran melalui formulir konfirmasi Kasus pengujian 1. Mengunggah bukti transfer. 2. Menyimpan data konfirmasi. Hasil yang diharapkan: 1. Sistem menyimpan konfirmasi 2. Menampilkan pemberitahuan bahwa bukti pembayaran berhasil di unggah				
6.	Usecase uji: Invoice Deskripsi: <i>Member</i> dapat melihat <i>invoice</i> pemesanan setelah pembayaran dikonfirmasi. Kasus pengujian 1. <i>Member</i> melihat detail transaksi. Hasil yang diharapkan: Sistem menampilkan invoice lengkap dengan status pembayaran.	BERHASIL	Syifa - <i>Member</i>	02 Agustus 2025	-
7.	Usecase uji: Mengelola Pesanan Masuk Deskripsi: <i>Admin</i> dapat melihat dan mengubah status pesanan <i>member</i>. Kasus pengujian 1. <i>Admin</i> melihat daftar pesanan. 2. Mengubah status menjadi "selesai" atau "dibatalkan". Hasil yang diharapkan:	BERHASIL	Afif - <i>Admin</i>	30 Juli 2025	-

	Sistem menampilkan daftar pesanan dan menyimpan status terbaru.				
8.	Usecase uji: Laporan Penjualan Deskripsi: <i>Admin</i> melihat laporan penjualan Kasus pengujian 1. <i>Admin</i> membuka menu laporan. 2. Menyaring laporan berdasarkan tanggal. Hasil yang diharapkan: Laporan tampil sesuai filter	BERHASIL	Afif - <i>Admin</i>	30 Juli 2025	-

4.5 Support

4.5.1 Publikasi Web

Website sakupaleza.my.id didaftarkan melalui layanan Rumahweb dengan status domain yang saat ini aktif dan masa berlaku hingga tanggal 2 Juli 2028. Domain ini dibeli dengan harga sebesar Rp105.000,00 untuk siklus billing selama tiga tahun.



Sumber: Hasil Penelitian

Gambar IV. 20 Publikasi Web

4.5.2 Spesifikasi *Hardware* dan *Software*

Spesifikasi *Hardware* yang digunakan oleh pengguna:

<i>Processor</i>	Core i7
RAM	512GB
<i>Harddisk</i>	GDDR5 4GB

Spesifikasi *software* yang digunakan oleh peneliti:

Sistem Operasi	Windows 10
Aplikasi	Visual Studio Code
Web Browser	Chrome
Database Server	MySQL
Web Server	Apache

4.6 Spesifikasi Dokumen Sistem Usulan

Berikut ini adalah spesifikasi dokumen dan sistem yang diusulkan:

- Nama Masukan : Form Pemesanan

Fungsi : Menerima data pesanan dari *Member*

Sumber : *Member*

Media : Formulir elektronik

Frekuensi : Setiap kali *Member* melakukan pemesanan

Keterangan : menyimpan transaksi dan menghasilkan invoice

Format : Lampiran B1
- Nama Masukan : Invoice

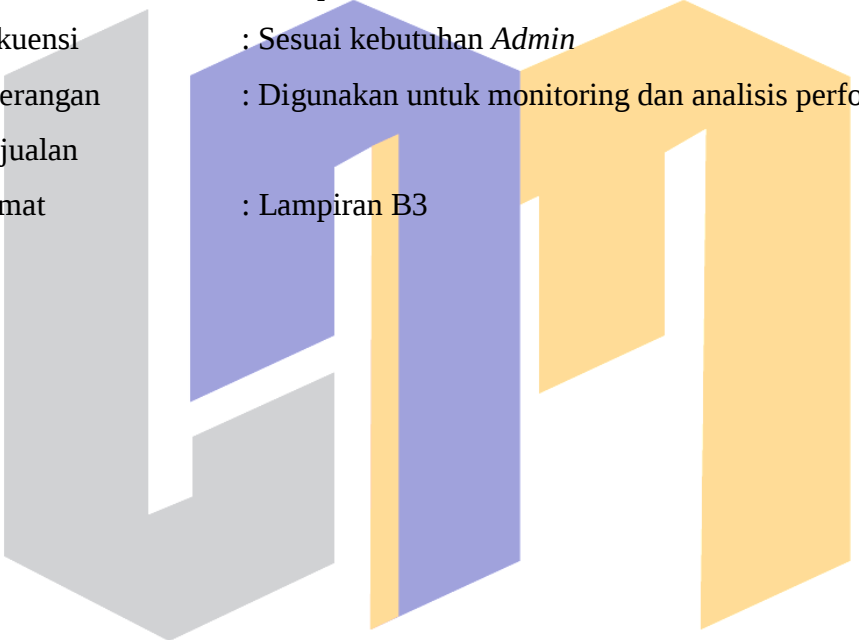
Fungsi : Menyajikan rincian transaksi pembelian sebagai bukti pesanan

Sumber : *Member* / sistem

Media : Tampilan di halaman web

- Frekuensi : Setiap kali transaksi berhasil dikonfirmasi checkout
 Keterangan : Berisi informasi pesanan
 Format : Lampiran B2
 3. Nama Masukan : Laporan Penjualan
 Fungsi : Menyajikan data rekap transaksi yang telah dilakukan dalam periode tertentu
 Sumber : *Admin* / sistem

Media : Tampilan halaman *admin*
 Frekuensi : Sesuai kebutuhan *Admin*
 Keterangan : Digunakan untuk monitoring dan analisis performa penjualan
 Format : Lampiran B3



UNIVERSITAS
NUSA MANDIRI