

**DPPL-01**

**DESKRIPSI PERANCANGAN PERANGKAT LUNAK  
SAYURLICIOUS**

untuk:

Tugas Besar Mata Kuliah RPL: Analisis dan Perancangan


Dipersiapkan oleh:

Kelompok L

Anggota:

M.TAUFIQ AL FIKRI	(1301223417)
SHERLY ANGELINA	(1301223370)
WILLY YEBTA PRAYOGA	(1301210521)
PUTRI AYULIA A	(1301213416)
RETNO CAHYANINGTIAS	(1301223224)

Program Studi S1 Informatika  
Fakultas Informatika  
Universitas Telkom  
2024

	Prodi S1- Informatika Universitas Telkom	Nomor Dokumen		Halaman
		DPPL-L		29
		Revisi		Tgl: 17/06/2024

### Daftar Perubahan

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

IndexTgl	-	A	B	C	D	E	F	G
DitulisOleh								
DiperiksaOleh								
DisetujuiOleh								

**Daftar Halaman Perubahan**

Halaman	Revisi	Halaman	Revisi

## Daftar Isi

Daftar Perubahan .....	2
Daftar Halaman Perubahan.....	3
Daftar Isi .....	4
Daftar Gambar.....	5
Daftar Table.....	6
1. Pendahuluan.....	7
1.1 Tujuan Penulisan Dokumen .....	7
1.2 Ruang Lingkup Masalah .....	7
1.3 Definisi dan Istilah.....	7
1.4 Referensi.....	7
2. Perancangan Global.....	8
2.1 Rencana Lingkungan Implementasi .....	8
2.2 Deskripsi Arsitektur Perangkat Lunak.....	8
3. Perancangan Rinci .....	9
3.1 Realisasi Use Case .....	9
3.2 Diagram Kelas Keseluruhan .....	22
3.3 Perancangan Detail Kelas .....	23
3.4 Perancangan Algoritma dan/ Query .....	24
4. Matrix Keruntutan .....	28

## Daftar Tabel

1. Table 1.1.....	8
2. Table 1.2.....	10
3. Table 1.3.....	11
4. Table 1.4.....	11
5. Table 1.5.....	11
6. Table 1.6.....	12
7. Table 2.1.....	15
8. Table 2.2.....	15
9. Table 2.3.....	15
10. Table 2.4.....	16
11. Table 3.1.....	17
12. Table 3.2.....	17
13. Table 3.3.....	17
14. Table 3.4.....	18
15. Table 4.1.....	19
16. Table 4.2.....	19
17. Table 4.3.....	19
18. Table 4.4.....	20
19. Table 5.1.....	21
20. Table 5.2.....	21
21. Table 5.3.....	21
22. Table 5.4.....	21
23. Table 6.1.....	22
24. Table 6.2.....	2

## Daftar gambar

1. Gambar 1.1 .....	9
2. Gambar 1.2.....	11
3. Gambar 1.3.....	13
4. Gambar 1.4.....	14
5. Gambar 1.5.....	16
6. Gambar 1.6.....	18
7. Gambar 1.7.....	20
8. Gambar 1.8.....	21
9. Gambar 1.9.....	22

## 1. Pendahuluan

### 1.1 Tujuan Penulisan Dokumen

Tujuan penulisan dokumen dari Aplikasi Belanja Produk Sayur Secara Online adalah untuk memberikan informasi kepada pengguna aplikasi mengenai keuntungan dan kemudahan yang didapat dalam membeli produk Sayur secara online, cara kerja aplikasi belanja produk Sayur secara online, jenis produk Sayur yang tersedia di dalam aplikasi belanja produk Sayur secara online, serta kebijakan privasi dan keamanan data pengguna yang diterapkan oleh aplikasi belanja produk Sayur secara online.

### 1.2 Lingkup Masalah

Ruang Lingkup masalah pada Aplikasi Belanja Produk Sayur Secara Online adalah memberikan layanan belanja produk Sayur secara online bagi pengguna yang membutuhkan. Aplikasi ini bertujuan untuk memudahkan pengguna dalam mencari dan membeli produk Sayur yang dibutuhkan tanpa harus pergi ke pasar tradisional. Masalah yang dicakup meliputi interaksi antara pengguna, toko sayur, dan admin dalam melakukan transaksi dan manajemen produk, serta masalah keamanan dan ketersediaan aplikasi. Aplikasi ini juga harus mampu menangani ketersediaan stok produk dan informasi produk yang akurat serta pembayaran yang aman dan terpercaya.

### 1.3 Definisi dan Istilah

Berikut adalah daftar definisi dan istilah penting yang digunakan dalam dokumen SKPL ini:

- SRS atau Software Requirements Specification adalah dokumen formal yang mendefinisikan kebutuhan fungsional dan non-fungsional dari sebuah perangkat lunak yang akan dikembangkan. Dokumen ini menyediakan deskripsi rinci tentang fitur, fungsi, kinerja, kinerja, antarmuka pengguna, keamanan, keterandalan, dan karakteristik lain dari perangkat lunak yang dibutuhkan untuk memenuhi kebutuhan bisnis atau teknis yang diinginkan.
- FR-XX adalah kode kebutuhan dari Functional Requirement.
- NFR-XX adalah kode kebutuhan dari Non-functional Requirement.
- Sign In adalah proses masuk ke dalam aplikasi dengan menggunakan akun yang terdaftar.
- Sign Up adalah proses pendaftaran untuk masuk ke aplikasi.

### 1.4 Referensi

Berikut beberapa referensi yang dapat membantu dalam pembuatan aplikasi Sayurlicious Mart:

#### Aplikasi Android:

- Android Developers: <https://developer.android.com/>

#### Aplikasi Pengiriman Sayur Online:

- HappyFresh: <https://www.happyfresh.id/>
- Sayurbox: <https://sayurbox.com/>

#### Metode Pembayaran:

- OVO: <https://www.ovo.id/>
- GoPay: <https://www.gopay.co.id/>

#### Website:

- Android Developers Blog: <https://android-developers.googleblog.com/>
- E-commerce Nation: <https://ecommerce-nation.com/>

#### Sumber Daya Lainnya:

- GitHub: <https://github.com>
- YouTube: <https://www.youtube.com/>

## 2. Perancangan Global

### 2.1 Rencana Lingkungan Implementasi

Lingkungan implementasi yang akan digunakan untuk pengembangan sistem Belanja Sayur Online ini adalah sebagai berikut:

*Operating System* : *Android*,  
*WindowsDBMS* : *MySQL*  
*Development Tools* : *Figma & VScode*  
*Bahasa Pemrograman* : *Python, javascript*

### 2.2 Deskripsi Arsitektur Perangkat Lunak

Aplikasi Belanja Produk Sayur Secara Online untuk memudahkan konsumen dalam memperoleh produk Sayur berkualitas dengan harga yang kompetitif. Aplikasi ini akan memberikan kemudahan bagi konsumen untuk mencari dan membeli produk Sayur yang dibutuhkan dengan lebih cepat dan efisien, serta memberikan informasi yang jelas dan lengkap mengenai produk-produk Sayur yang tersedia. Selain itu, aplikasi ini juga bertujuan untuk meningkatkan pemasaran produk Sayur yang diproduksi oleh para toko sayur lokal dan membantu mereka untuk memperluas jangkauan pasarnya, sehingga dapat meningkatkan kesejahteraan dan meningkatkan kualitas hidup para pedagang sayur.

Deskripsi Arsitektur digambarkan dengan UML Component Diagram. Berikutnya, membuat daftar modul. Modul adalah menu, fungsi sebuah kelas. Daftar modul bisaditulis dalam bentuk tabel berikut:

No	Nama Komponen	Keterangan
1	Register	Calon dapat membuat akun baru di register
2	Login	Pengguna dapat mengakses akunnya yang telah terdaftar
3	Dashboard Pengguna	Menampilkan informasi pribadi pengguna, riwayat pembelian, dan status pesanan
4	Katalog Produk	Menampilkan daftar sayur yang tersedia, lengkap dengan hargadan gambar
5	Pencarian dan Filter	Memungkinkan pengguna untuk mencari sayur berdasarkan kategori, harga, dan ketersediaan
6	Keranjang Belanja	Fitur untuk menambahkan, menghapus, dan mengubah jumlahsayur yang akan dibeli
7	Transaksi	Mendukung berbagai metode transaksi seperti transfer bank, e-wallet, maupun cash
8	Manajemen Produk	Penjualan dapat menambahkan, menghapus, dan memperbarui informasi produk yang akan dijual
9	Riwayat Pesanan	Menampilkan daftar pesanan yang pernah dilakukan oleh pengguna
10	Pengiriman	Integrasi dengan layanan pengiriman untuk melacak status pengiriman pesanan





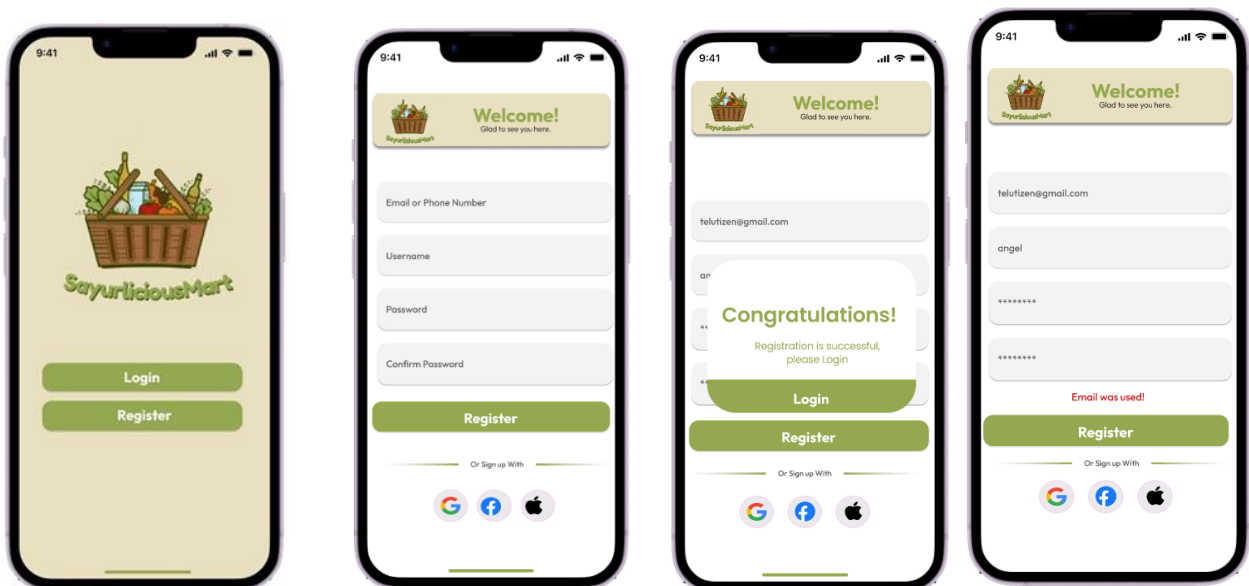
### 3.1.1.1 Use Case Scenario Register

#### Skenario Use Case Register

- i. Pre-Condition : Sistem menampilkan form data dan aktor mengisi data diri
- ii. Use Case Description
  - a. Primary Flow
    1. Calon Pengguna memilih opsi “Registrasi Akun” pada halaman login
    2. Sistem menampilkan formulir registrasi yang meminta pengguna untuk memasukan informasi data diri seperti nama, alamat email, dan kata sandi
    3. Pengguna mengisi formulir dan mengirimkannya
    4. Sistem memvalidasi informasi pengguna, memeriksa ketersediaan alamat email, dan menyimpan data akun pengguna baru ke dalam database
    5. Sistem memberikan konfirmasi pendaftaran sukses dan mengarahkan pengguna ke halaman login
    6. Pengguna dapat masuk menggunakan alamat email dan kata sandi yang telah terdaftar
  - b. Alternative Flow
    1. Jika alamat email yang dimasukkan sudah terdaftar
      - 1.1 Sistem memberikan pesan error dan meminta pengguna untuk menggunakan alamat email lain
    2. Jika terdapat kesalahan atau kegagalan pada saat registrasi
      - 2.1 Sistem memberikan pesan kesalahan dan meminta pengguna untuk mencoba lagi
- iii. Post Condition : Sistem memvalidasi data yang diisi oleh pengguna.

### 3.1.1.2 UI Design dan Deskripsi Objek UI #1 <Register>

Gambar 1.2



Tabel 1.3 Tabel Deskripsi Objek UI

<b>Id. Layar</b>	<b>Nama Layar</b>	<b>Deskripsi</b>
001	Registrasi	Fungsi ini digunakan untuk memasukan registrasi page dan pengguna dapat memulai registrasi aplikasi.

Tabel 1.4- UI Design Page Register

<b>Id_Objek</b>	<b>Jenis</b>	<b>Label*</b>	<b>Keterangan**</b>
Id_Register	Button	Registrasi	Jika diklik akan mengaktifkan fungsi untuk membuat akun baru
RTF1	RTF Box		Isi Teks yang disimpan pada File xxx
Id_Username	Text Box	Username	Pengguna memasukkan nama pengguna yang diinginkan
Id_email	Text Box	Email	Pengguna memasukkan alamat email yang valid
Id_Password	Text Box	Password	Pengguna memasukkan kata sandi yang diinginkan
Id_Confirm	Text Box	Confirm Password	Pengguna memasukkan ulang kata sandi untuk informasi
Id_Submit	Button	Submit	Jika diklik, akan mengirimkan data registrasi untuk diproses dan membuat akun baru

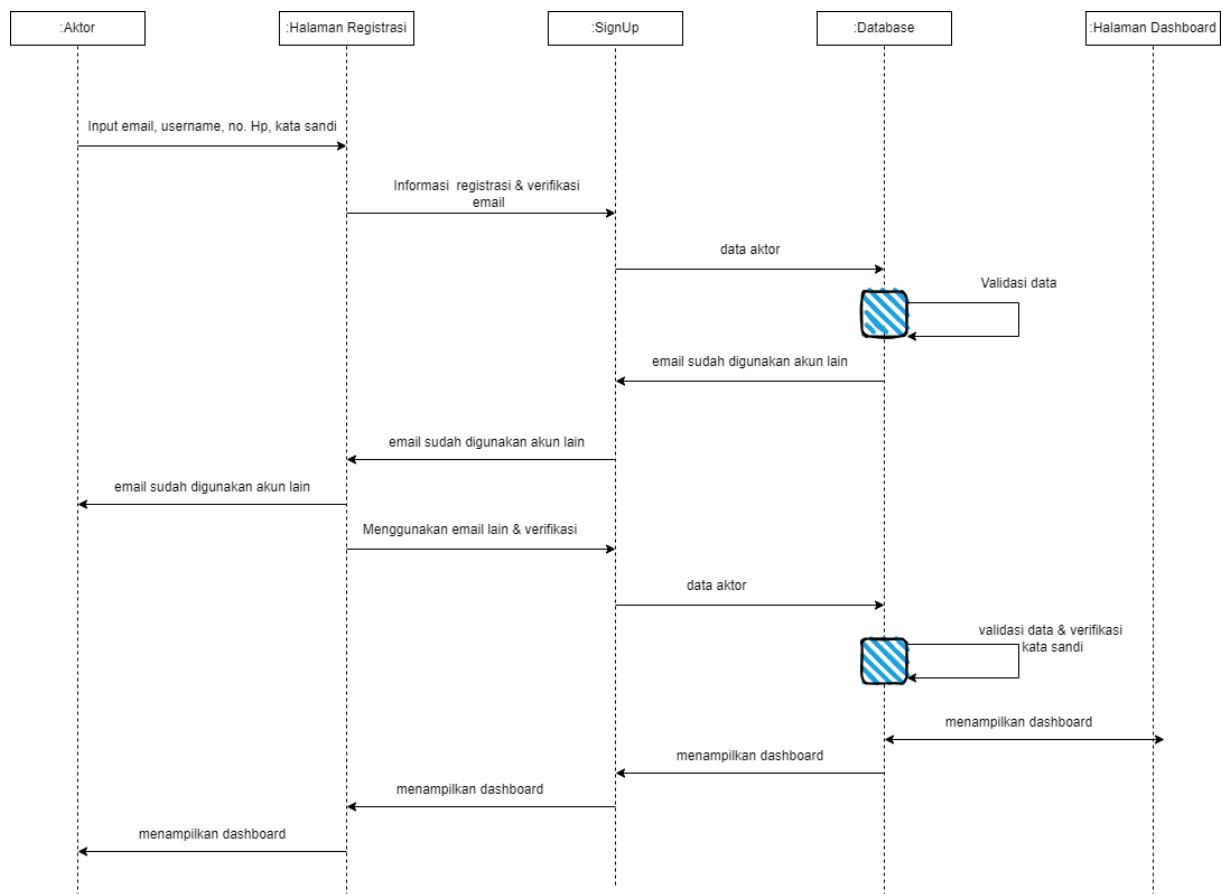
### 3.1.1.3 Identifikasi Object dan Tipe Kelas #1 <Register>

Tabel 1.5 -Tabel Object Perancangan

<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1	Input email	string
2	Input password	string

### 3.1.1.4 Sequence Diagram #2 <Register>

Gambar1.3



### 3.1.1 Use Case #2 <Login>

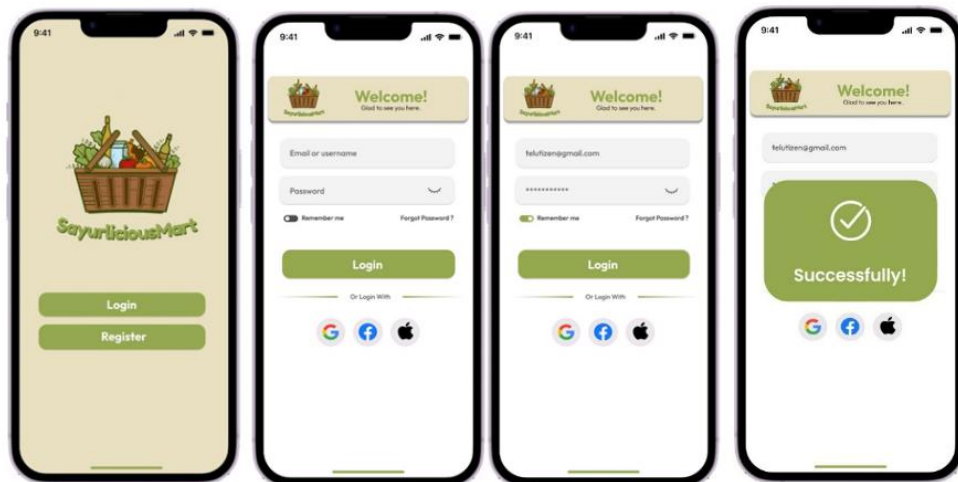
#### 3.1.1.1 Use Case Scenario #2 <Login>

Skenario Use Case login web :

- i. Pre-Condition : Pengguna membuka aplikasi
- ii. Use Case Description : Seorang pengguna ingin login ke akunnya pada Web belanjasaayuran online.
  - a. Primary Flow
    1. Pengguna membuka Web sayuran online.
    2. Pengguna memilih opsi “Login” di halaman utama.
    3. Sistem menampilkan formulir login yang meminta pengguna untuk memasukkan alamat email dan kata sandi.
    4. Pengguna memasukkan informasi login dan mengirimkannya.
    5. Sistem memvalidasi informasi login.
    6. Sistem menampilkan pesan login sukses.
    7. Sistem mengarahkan pengguna ke area dashboard sesuai dengan hak akses mereka.
  - b. Alternative Flow
    1. Jika informasi login tidak valid(invormasi tidak sesuai).
      - 1.1. Sistem menampilkan pesan error.
      - 1.2. Sistem meminta pengguna untuk memasukkan ulang informasi login yang benar.
    2. Jika pengguna telah melakukan login sebelumnya dan memilih opsi “Ingat Saya”.
      - 2.1. Sistem mengenali pengguna dan melewati proses login.
      - 2.2. Pengguna langsung diarahkan ke area dashboad sesuai dengan hak akses mereka.
    3. Jika pengguna lupa kata sandi.
      - 3.1. Pengguna memilih opsi “Lupa Kata Sandi”.
      - 3.2. Sistem meminta pengguna untuk memasukkan alamat email terdaftar.
      - 3.3. Pengguna memasukkan alamat email dan mengirimkannya.
      - 3.4. Sistem mengirimkan tautan atau intruksi reset kata sandi ke email pengguna.
      - 3.5. Pengguna mengikuti tautan dan intruksi untuk mengatur ulang kata sandi.
- iii. Post-Condition : Pengguna berhasil masuk ke dalam Web dan diarahkan kearea menu utama.

#### 3.1.1.2 UI Design dan Deskripsi Objek UI #2 <Login>

Gambar 1.4



Tabel 2-.1 Tabel Deskripsi Objek UI

<b>Id. Layar</b>	<b>Nama Layar</b>	<b>Deskripsi</b>
001	Login	User diminta memasukkan email atau username dan password yang telah terdaftar sebelumnya.

Tabel 2-.2 UI Design dan Deskripsi Page Login Web

<b>Id_Objek</b>	<b>Jenis</b>	<b>Label*</b>	<b>Keterangan**</b>
Button1	Button	Login	<i>Jika diklik, data yang diinputkan oleh pengguna akan dikirimkan untuk melakukan proses validasi.</i>
RTF 1	Text box	Email or username	<i>Pelanggan input email atau username ke dalam text box untuk divalidasi.</i>
RTF2	Text box	Password	<i>Pelanggan input password ke dalam text box untuk divalidasi.</i>
Button 2	Text button	Forgot password?	<i>Jika diklik, pelanggan akan dialihkan ke page Reset Password.</i>
Button 3	Button	Turn on/off	<i>Jika diklik, pelanggan dapat mengaktifkan atau menonaktifkan auto login.</i>
Button 4	Simbol	Eye	<i>Jika diklik, pelanggan dapat melihat atau menyembunyikan password yang telah diinput sebelum dikirimkan.</i>
Button 5	Logo	Google	<i>Jika diklik, pelanggan dapat login langsung lewat akun google.</i>
Button 6	Logo	Facebook	<i>Jika diklik, pelanggan dapat login langsung lewat akun facebook.</i>
Button 7	Logo	Apple	<i>Jika diklik, pelanggan dapat login langsung lewat akun apple.</i>

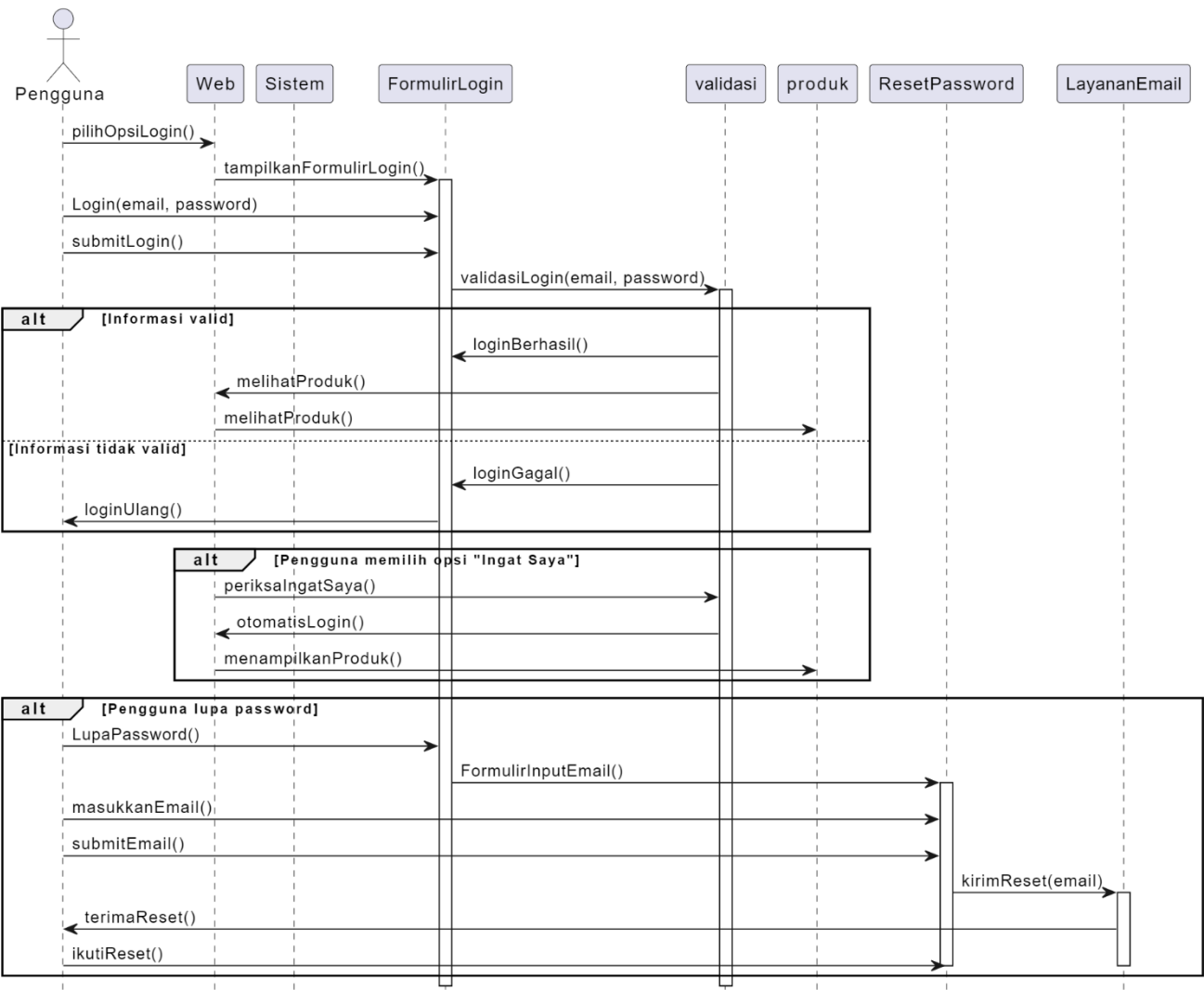
### 3.1.1.3 Identifikasi Object Baru & Tipe Kelas #2 <Login>

Tabel 2-.3 Tabel Object Perancangan

	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
	Input Email or username	String
	Input Password	String

3.1.1.4 Sequence Diagram #2 <Login>

Gambar 1.5



### 3.1.2 Use Case #3 <Melihat Produk>

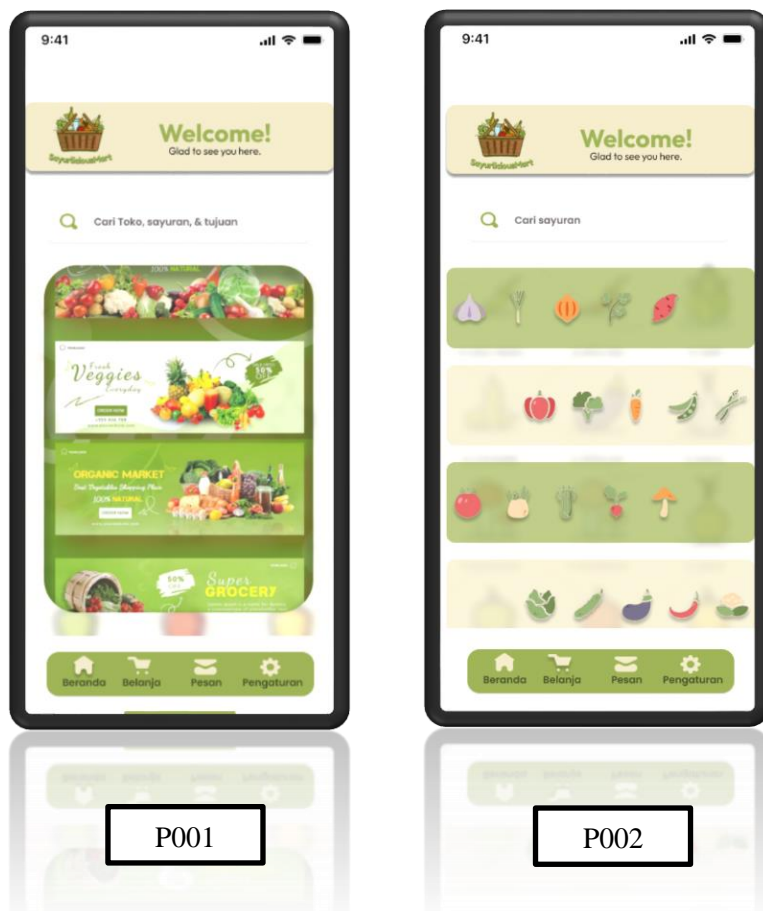
#### 3.1.2.1 Use Case Scenario Melihat Produk

Skenario Use Case Register

- i. Pre-Condition : Pembeli melihat daftar produk
- ii. Use Case Description : Sistem menampilkan produk yang akan dijual pada aplikasi sayuran online
  - a. Primary Flow
    1. Pengguna memilih opsi “Belanja” pada menu
    2. Sistem menampilkan produk yang tersedia pada aplikasi
  - b. Alternative Flow
    1. Jika pengguna ingin melihat produk sesuai apa yang diinginkan
      - 1.1 Pengguna memasukan nama produk pada pencarian bagian atas menu “Beranda”
      - 1.2 Sistem menampilkan produk berdasarkan pencarian pengguna
    2. Jika pengguna ingin melihat produk berdasarkan pencarian harga termurah dari beberapa merek produk yang dicari
      - 2.1 Pengguna memasukan nama produk pada pencarian kemudian menggunakan fitur “Filter” yang tersedia pada area pencarian
      - 2.2 Sistem menampilkan produk berdasarkan pencarian dan harga yang sudah di filter oleh pengguna
- iii. Post-Condition : Pengguna dapat melihat lihat produk yang tersedia pada halaman utama aplikasi

#### 3.1.2.2 UI Design dan Deskripsi Objek UI #3 <Melihat Produk>

Gambar 1.5





Tabel 3-1 Tabel Deskripsi Objek UI

<b>Id. Layar</b>	<b>Nama Layar</b>	<b>Deskripsi</b>
P001	Beranda	Tampilan Home Page dari Aplikasi
P002	Belanja	Menampilkan seluruh produk yang tersedia

Tabel 3.2- UI Design Page Register

<b>Id_Objek</b>	<b>Jenis</b>	<b>Label*</b>	<b>Keterangan**</b>
Button 1	Button	Beranda	Jika diklik, akan menampilkan halaman awal aplikasi
Search	Search Box	Cari Toko, Sayuran, ...	Isi teks akan dicari sesuai indeks pada data base
Button 2	Button	Belanja	Jika diklik, akan menampilkan halaman produk yg ditawarkan
Search	Search Box	Cari Sayuran	Isi teks berupa produk yang akan dicari oleh aktor

Tabel 3.3- UI Design Page Registrasi

<b>Id_Objek</b>	<b>Jenis</b>	<b>Label*</b>	<b>Keterangan**</b>
TextBox1	Textbox	Username	Jika diklik, akan mengaktifkan Function.....
Buuton 2	Button	SAVE	Jika diklik akan mengaktifkan .....

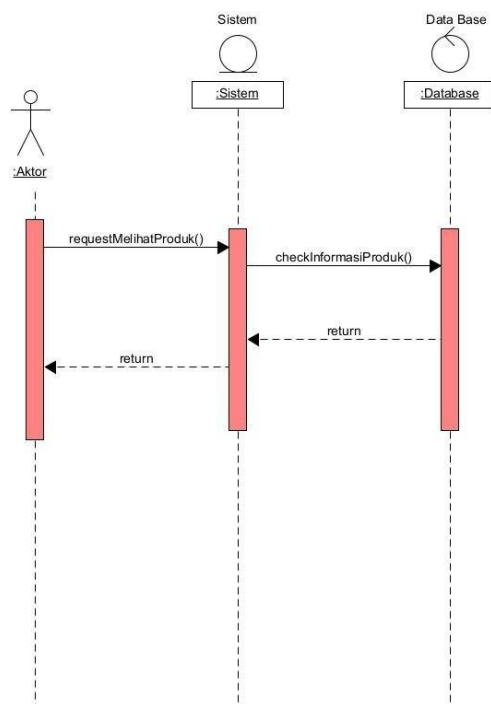
### 3.1.2.3 Identifikasi Object dan Tipe Kelas #3 <Melihat Produk>

Tabel 34- Tabel Object Perancangan

	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
	Input Toko, Sayuran, dan tujuan	String
	Input Sayuran	String

### 3.1.2.4 Sequence Diagram #3 <Melihat Produk>

Gambar 1.6



### 3.1.2.5 Sequence Diagram #4 <Memesan Produk>

#### 3.1.4.1 Use Case Scenario Memesan Produk

Skenario Use Case Register

- i. Pre-Condition : Pengguna telah login ke dalam sistem
- ii. Use Case Description
  - a. Primary Flow
    1. Pengguna memilih produk yang ingin di pesan dari katalog produk,
    2. Sistem menampilkan detail produk yang dipilih.
    3. Pengguna menekan tombol “Tambah kekeranjang”.
    4. Sistem menambahkan produk ke keranjang belanja pengguna.
    5. Pengguna membuka keranjang belanja.
    6. Sistem menampilkan isi keranjang belanja.
    7. Pengguna menekan tombol “Lanjut ke Pembayaran”
    8. Sistem menampilkan halaman pembayaran.
    9. Pengguna memasukkan informasi pembayaran dan alamat pengiriman.
    10. Pengguna menekan tombol “Konfirmasi pesanan”.
    11. Memvalidasi informasi pembayaran.
    12. Sistem menyimpan informasi pesanan dan memberikan notifikasi bahwa pesanan berhasil dilakukan .
    13. Sistem mengirim konfirmasi ke email pengguna.
  - b. Alternative Flow
    1. Produk tidak tersedia.
      - 1.1. Jika produk dipilih tidak tersedia , sistem menampilkan pesan bahwa produk habis.
      - 1.2. Pengguna memilih produk lain dari katalog
    2. Kesalahan saat pembayaran
      - 2.1. Jika teradapat kesalahan saat pembayaran, sistem menampilkan pesan error dan meminta pengguna untuk memperbaiki informasi pembayaraan.
      - 2.2. Pengguna memperbaiki informasi pembayaraan dan menekan tambol “Konfirmasi pesanan” lagi.
- iii. Post-Condition : Pesanberhasil dilakukan dan informasi pesanan disimpan di sistem.

#### 3.1.4.2 UI Design dan Deskripsi Objek UI #4 <Memesan Produk>

Tabel 3.1 Tabel Layar Memesan produk

<b>Id. Layar</b>	<b>Nama Layar</b>	<b>Deskripsi</b>
Id_Pilihproduk	Pilih Produk	Fungsi ini digunakan pengguna untuk memilih produk dari katalog.
Id_DetailProduk	Detail Produk	Fungsi ini digunakan untuk melihat detail produk yang dipilih pengguna.
Id_Keranjang	Keranjang Belanja	Fungsi ini digunakan untuk melihat dan mengelola isi keranjang belanja.
Id_Pembayaran	Pembayaran	Fungsi ini digunakan untuk memasukkan informasi pembayaran dan pengiriman.
Id_Konfirmasi	Konfirmasi Pesanan	Fungsi ini digunakan untuk menampilkan konfirmasi pesanan pengguna.

Tabel 3.2 Tabel Deskripsi Objek UI

<b>Id. Layar</b>	<b>Nama Layar</b>	<b>Deskripsi</b>
Id_PilihProduk	Pilih Produk	Fungsi ini digunakan pengguna untuk memilih produk dari katalog.
Id_DetailProduk	Detail Produk	Fungsi ini digunakan untuk melihat detail produk yang dipilih pengguna.
Id_Keranjang	Keranjang Belanja	Fungsi ini digunakan untuk melihat dan mengelola isi keranjang belanja.
Id_Pembayaran	Pembayaran	Fungsi ini digunakan untuk memasukkan informasi pembayaran dan pengiriman.
Id_Konfirmasi	Konfirmasi Pesanan	Fungsi ini digunakan untuk menampilkan konfirmasi pesanan pengguna.

UNTUK MASING – MASING ANTAR MUKA / PAGE dibuatkan spesifikasi detail  
Antarmuka XXX: {diisi dengan no. layar atau no gambar rancangan antarmuka}

Tabel 3.3 UI Design Page Pilih Produk

<b>Id_Objek</b>	<b>Jenis</b>	<b>Label*</b>	<b>Keterangan**</b>
TextBox1	TextBox	Pencarian	Pengguna memasukkan kata kunci untuk mencari produk
Button1	Button	Cari	Jika diklik, akan mengaktifkan fungsi pencarian produk berdasarkan kata kunci
ListBox1	ListBox	Daftar Produk	Menampilkan daftar produk berdasarkan hasil pencarian

Tabel 3.4 UI Design Page Detail produk

<b>Id_Objek</b>	<b>Jenis</b>	<b>Label*</b>	<b>Keterangan**</b>
Label1	Label	Nama Produk	Menampilkan nama produk
Label2	Label	Harga	Menampilkan harga produk
Label3	Label	Deskripsi	Menampilkan deskripsi produk
TextBox2	TextBox	Jumlah	Pengguna memasukkan jumlah produk yang diinginkan
Button2	Button	Tambah ke Keranjang	Jika diklik, akan menambahkan produk ke keranjang belanja

Tabel 3.5 UI Design Page Pembayaran

<b>Id_Objek</b>	<b>Jenis</b>	<b>Label*</b>	<b>Keterangan**</b>
TextBox3	TextBox	Nama Lengkap	Pengguna memasukkan nama lengkap
TextBox4	TextBox	Alamat	Pengguna memasukkan alamat pengiriman
TextBox5	TextBox	Nomor Telepon	Pengguna memasukkan nomor telepon
TextBox6	TextBox	Email	Pengguna memasukkan email
TextBox7	TextBox	Nomor Kartu	Pengguna memasukkan nomor kartu kredit/debit
TextBox8	TextBox	Tanggal Kedaluwarsa	Pengguna memasukkan tanggal kedaluwarsa kartu
TextBox9	TextBox	Kode CVV	Pengguna memasukkan kode CVV kartu
Button5	Button	Konfirmasi	Jika diklik, akan mengonfirmasi pesanan dan memproses pembayaran

Tabel 3.4 UI Design Page Konfirmasi Pesanan

<b>Id_Objek</b>	<b>Jenis</b>	<b>Label*</b>	<b>Keterangan**</b>
Label4	Label	<i>Nomor Pesanan</i>	<i>Menampilkan nomor pesanan</i>
Label5	Label	<i>Nama Produk</i>	<i>Menampilkan nama produk yang dipesan</i>
Label6	Label	<i>Jumlah</i>	<i>Menampilkan jumlah produk yang dipesan</i>
Label7	Label	<i>Total Harga</i>	<i>Menampilkan total harga pesanan</i>
Button6	Button	Selesai	<i>Jika diklik, akan menyelesaikan proses pemesanan dan kembali ke halaman utama</i>

#### Identifikasi Object dan Tipe Kelas #4 <Memesan Produk>

Identifikasi object yang terkait dengan use case tersebut. Kelas pada tahap perancangan berbeda dengan kelas pada tahap analisis. Gunakan tabel di bawah:

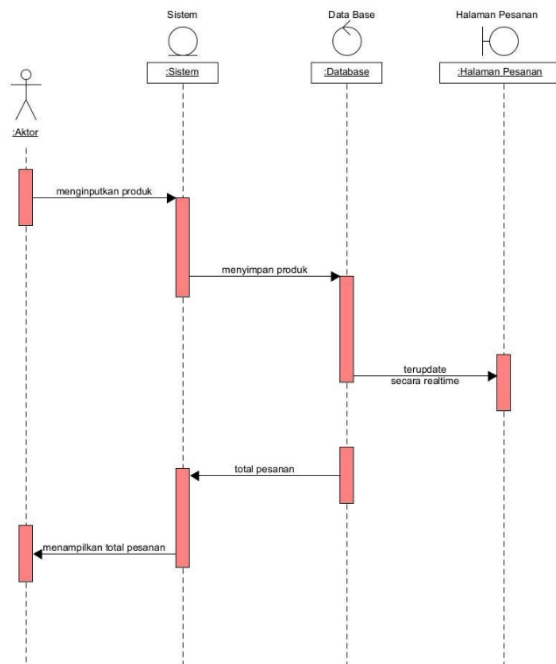
Tabel 3.4 Tabel Object Perancangan

<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1	Produk	Entity
2	Keranjang	Boundary
3	Pesanan	Entity
4	Pembayaran	Controller

\*Tipe kelas seperti Boundary (Interface), Entity (Database), Controller

#### 3.1.4.4 Sequence Diagram #4 <Memesan Produk>

Gambar 1.7



### 3.1.3 Use Case #5 <Melacak Produk>

#### 3.1.3.1 Use Case Scenario Melacak Produk

Skenario Use Case Register

- iv. Pre-Condition
- v. Use Case Description
  - a. Primary Flow
  - b. Alternative Flow
- vi. Post-Condition

#### 3.1.3.2 UI Design dan Deskripsi Objek UI #5 <Melacak Produk>

Tabel 5.1- Tabel Deskripsi Objek UI

Id. Layar	Nama Layar	Deskripsi
Id_login	Login	Fungsi ini di gunakan pengguna untuk memasuki aplikasi
Id_Register	Registrasi	Fungsi ini digunakan untuk memasukan registrasi page dan pengguna dapat memulai registrasi aplikasi.

Tabel 5.2 UI Design Page Register

Id_Objek	Jenis	Label*	Keterangan**
Id_login	Button	Login	Jika diklik, akan mengaktifkan Function.....
Id_Register	Button	Registrasi	Jika diklik akan mengaktifkan .....
RTF1	RTF Box		Isi Teks yang disimpan pada File xxx

Tabel 5.3- UI Design Page Registrasi

Id_Objek	Jenis	Label*	Keterangan**
TextBox1	Textbox	Username	Jika diklik, akan mengaktifkan Function.....
Buuton 2	Button	SAVE	Jika diklik akan mengaktifkan .....

#### 3.1.3.3 Identifikasi Object dan Tipe Kelas #5 <Melacak Produk>

Tabel 5.4- Tabel Object Perancangan

No	Nama Object Baru	Jenis / Tipe Kelas
1		
2		
3		
4		
5		
6		

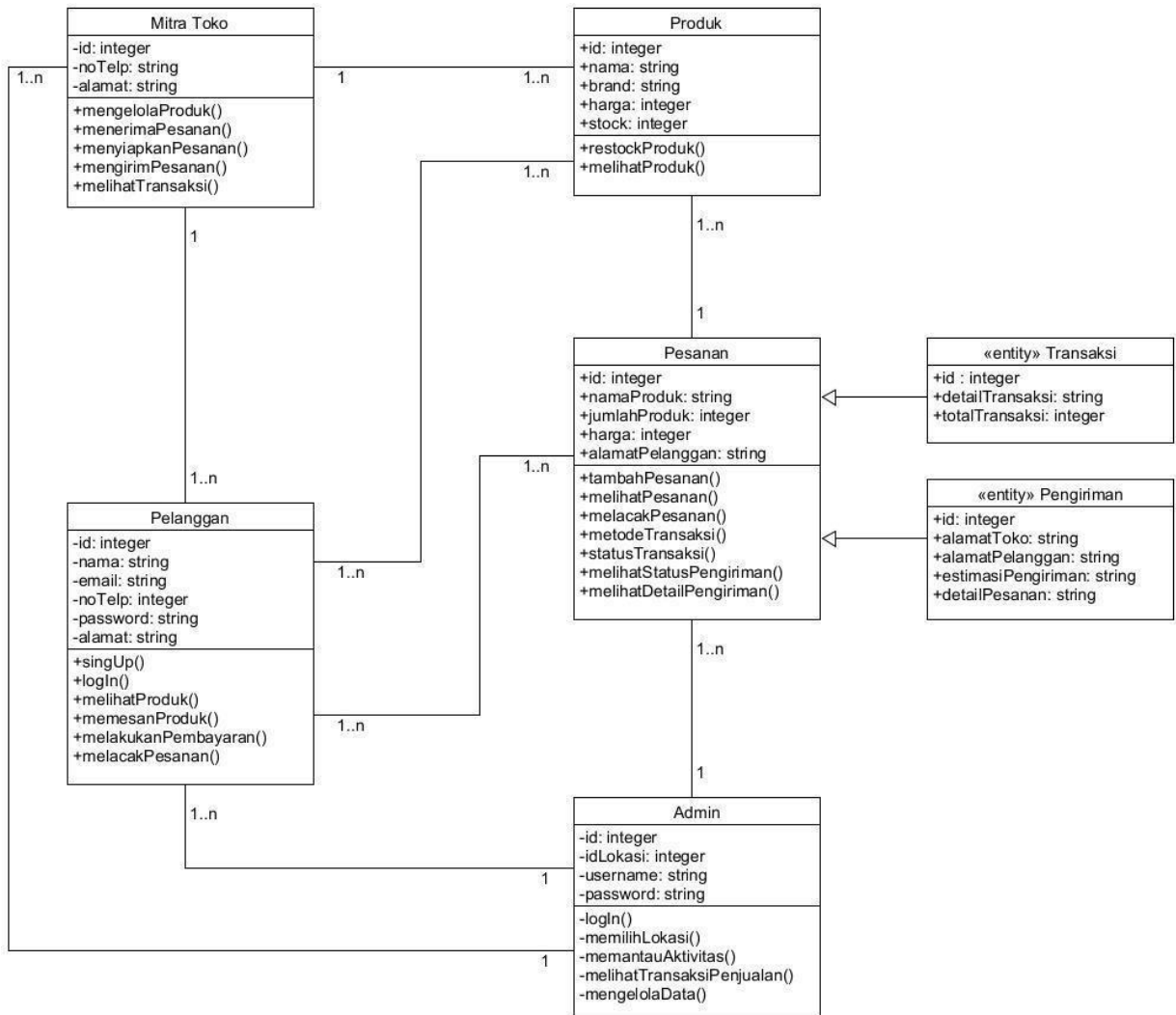
#### 3.1.3.4 Sequence Diagram #5 <Melacak Produk>

Gambar 1.8

### 3.2 Diagram Kelas Keseluruhan

Bagian ini diisi dengan diagram kelas keseluruhan yang akan digunakan dalam PL menggunakan model MVC

Gambar 1.9



### 3.3 Perancangan Detil Kelas

Bagian ini diisi dengan daftar keseluruhan kelas yang akan digunakan dalam PL. menggunakan model MVC

-Tabel 6.1 Tabel kelas

<b>Id Kelas</b>	<b>Nama Kelas Perancangan</b>	<b>Atribute (Visibility)</b>	<b>Method / Operation</b>
1	Pengguna/Pelanggan	idPengguna, noTelpPengguna, namaPengguna, emailPengguna, passwordPengguna	Daftar Produk, Cari Produk, Tambahkan Produk ke Keranjang, Lihat Keranjang, Lakukan Pembayaran, Lihat Riwayat Pesanan, Ubah Profil Pengguna
2	Mitra Toko/Toko sayur	idTokoSayur, namaTokoSayur, alamatTokoSayur	Daftar Produk, Tambahkan Produk, Ubah Produk, Hapus Produk, Lihat Pesanan, Lihat Riwayat Transaksi
3	Admin	idAdmin, namaAdmin, username, password	Kelola Akun Pengguna, Kelola Akun Toko Sayur, Pantau Aktivitas, Lihat Pesanan, Lihat Transaksi
4	Produk	idProduk, namaProduk, deskripsiProduk, hargaProduk	
5	Pesanan	IdPesanan,namaProduk,jumlahProduk,hargaProduk,alamatpelanggan	
6	Transaksi	idTransaksi, namaProduk,jumlahProduk, totalHarga	
7	Pembayaran	idPembayaran, totalPembayaran,detailPembayaran	Lakukan Pembayaran

Untuk setiap kelas:

- identifikasi operasi (mengacu pada tanggung-jawab kelas), identifikasi atribut, termasuk visibility-nya

### 3.4 Perancangan Algoritma dan/atau Query

#### Algoritma #1

Nama Kelas : PelangganNama

Operasi : Register Algoritma

```
public class RegisterPelanggan {

    public static void main(String[] args) {
        // Asumsikan data input pengguna sudah disimpan dalam variabel
        String nama = "Budi";
        String email = "budi@sayurlicious.com";

        String nomorTelepon = "08123456789";
        String password = "rahasia123";

        // Validasi input pengguna
        // ...

        // Periksa keunikan email
        if (isEmailExist(email)) {
            System.out.println("Email sudah terdaftar!");
            return;
        }

        // Menyimpan data pelanggan
        Pelanggan pelanggan = new Pelanggan(nama, email, nomorTelepon, password);
        savePelanggan(pelanggan);

        // Mengirimkan email konfirmasi
        sendKonfirmasiEmail(email);

        // Menampilkan pesan sukses
        System.out.println("Pendaftaran berhasil! Silakan periksa email Anda untuk mengaktifkan akun.");
    }

    private static boolean isEmailExist(String email) {
        // Lakukan query ke database untuk memeriksa keberadaan email
        // ...
    }

    private static void savePelanggan(Pelanggan pelanggan) {
        // Simpan objek pelanggan ke database
        // ...
    }

    private static void sendKonfirmasiEmail(String email) {
        // Buat email konfirmasi dan kirimkan ke alamat email yang diberikan
        //
```



## Algoritma #2

Nama Kelas : PelangganNama

Operasi : Login Algoritma :

```
public class LoginPelanggan {  
  
    public static void main(String[] args) {  
        // Asumsikan data input pengguna sudah disimpan dalam variabelString  
        email = "budi@sayurlicious.com";  
        String password = "rahasia123";  
  
        // Validasi input pengguna  
        // ...  
  
        // Memeriksa kredensial pengguna  
        Pelanggan pelanggan = login(email, password);if  
        (pelanggan == null) {  
            System.out.println("Email atau password salah!");return;  
        }  
  
        // Membuat sesi pengguna  
        createSession(pelanggan);  
  
        // Mengalihkan pengguna ke halaman utama  
        System.out.println("Login berhasil! Selamat datang, " + pelanggan.getNama() + ".");  
        // ... (redirect to main page)  
    }  
  
    private static Pelanggan login(String email, String password) {  
        // Lakukan query ke database untuk mengambil data akun pelanggan  
        // ...  
  
        // Bandingkan password yang dimasukkan dengan password yang tersimpan  
        // ...  
  
        // Jika kredensial valid, kembalikan objek pelanggan  
        // ...  
        return null; // Replace with actual return statement  
    }  
  
    private static void createSession(Pelanggan pelanggan) {  
        // Simpan informasi pengguna di sesi  
        // ...  
    }  
}
```

### Algoritma #3

Nama Kelas: Pelanggan

Nama Operasi : melihat produk Algoritma

```
public class LihatProdukPelanggan {

    public static void main(String[] args) {
        // Asumsikan pengguna sudah login dan berada di halaman daftar produk

        // Mengambil daftar produk
        List<Produk> produkList = getProdukList(kategori, kriteriaUrutan, halaman);

        // Menampilkan daftar produk
        for (Produk produk : produkList) {
            System.out.println("Gambar: " + produk.getGambarHighRes());
            System.out.println("Nama: " + produk.getNama()); System.out.println("Harga: " +
            produk.getHargaFormatted());

            System.out.println("Deskripsi: " + produk.getDeskripsiSingkat()); System.out.println("Rating: " +
            produk.getRating() + " (" + produk.getJumlahUlasan() + "
ulasan)");
            System.out.println("Tombol: Tambah ke Keranjang");
            System.out.println("-----");
        }

        // Menampilkan filter produk
        showFilterProduk(kategori, kriteriaUrutan);

        // Menampilkan paginasi
        showPaginasi(halaman, totalProduk);

        // Menangani kesalahan
        // ...
    }

    private static List<Produk> getProdukList(String kategori, String kriteriaUrutan, int halaman) {
        // Lakukan query ke database untuk mengambil data produk
        // Filter produk berdasarkan kategori dan kriteria urutan
        // Paginate daftar produk
        // ...

        // Kembalikan daftar produk
        // ...
        return null; // Replace with actual return statement
    }

    private static void showFilterProduk(String kategori, String kriteriaUrutan) {
        // Tampilkan daftar kategori produk yang tersedia
        // Tampilkan daftar kriteria pengurutan
        // ...
    }

    private static void showPaginasi(int halaman, int totalProduk) {
        // Tampilkan nomor halaman
        // Izinkan pengguna untuk berpindah antar halaman
        // Tampilkan informasi jumlah produk yang ditampilkan dan total produk
        // ...
    }
}
```

#### Algoritma #4

Nama Kelas : Pelanggan

Nama Operasi : Transaksi pembayaran

```
public class PembayaranPelanggan {

    public static void main(String[] args) {
        // Asumsikan pengguna sudah login, selesai melihat ringkasan transaksi, dan memilih metode pembayaran

        // Menampilkan informasi pembayaran
        showTransaksiSummary();

        // Menghubungi gateway pembayaran
        String response = processPayment(metodePembayaran);

        // Menangani respon gateway pembayaran
        if (response.equals("SUCCESS")) {
            handlePaymentSuccess();
        } else {
            handlePaymentFailure(response);
        }

        // Menangani kesalahan
        // ...
    }

    private static void showTransaksiSummary() {
        // Tampilkan kembali ringkasan transaksi kepada pengguna
        // ...
    }

    private static String processPayment(String metodePembayaran) {
        // Lakukan komunikasi dengan gateway pembayaran
        // Kirimkan informasi transaksi yang diperlukan
        // ...

        // Kembalikan respon dari gateway pembayaran (SUCCESS atau error message)
        // ...
        return null; // Replace with actual return statement
    }

    private static void handlePaymentSuccess() {
        // Perbarui status transaksi, kurangi stok produk, kirim email konfirmasi
        // ...

        System.out.println("Pembayaran berhasil!");
    }

    private static void handlePaymentFailure(String errorMessage) {
        // Tampilkan pesan error kepada pengguna, jelaskan penyebab kegagalan
        // ...

        System.out.println("Pembayaran gagal: " + errorMessage);
    }
}
```

#### 4 Matriks Keruntutan (Requirement Traceability Matrix)

Mapping requirement dengan Use Case yang direalisasikan

Tabel 6.2 Tabel kelas

Kode FR	Nama Functional Requirement	ID Use-case	Nama Use Case
FR1	Pengguna dapat mendaftar dan membuat akun baru pada aplikasi.	UC1	Register
FR2	Pengguna dapat masuk ke akun yang telah terdaftar pada aplikasi.	UC2	Login
FR3	Pengguna dapat melihat daftar produk yang tersedia pada aplikasi.	UC3	Lihat Daftar Produk
FR4	Pengguna dapat memilih produk dan melakukan pembelian pada aplikasi.	UC4	Melakukan Transaksi
FR5	Pengguna dapat melihat riwayat pesanan mereka, termasuk detail pesanan, status pengiriman, dan informasi pembayaran.	UC5	Melihat Riwayat Pesanan
FR6	Toko sayur dapat mendaftar sebagai penjual dengan mengisi formulir pendaftaran yang meminta informasi seperti nama, alamat email, dan data Sayur yang mereka kelola.	UC6	Mendaftar Sebagai Penjual
FR7	Toko sayur dapat mengelola produk yang ingin dijual pada aplikasi, seperti menambahkan, mengedit, atau menghapus produk.	UC7	Mengelola Produk

FR8	Toko sayur dapat melihat pesanan yang diterima dari pengguna, termasuk detail pesanan dan informasi pengiriman.	UC8	Melihat Pesanan
FR9	sayur dapat melihat laporan penjualan mereka, termasuk informasi seperti total penjualan, produk terlaris.	UC9	Melihat Transaksi Penjualan
FR10	Admin dapat mengelola akun pengguna dan toko sayur, termasuk pendaftaran, pengaktifan, atau penonaktifan akun.	UC10	Mengelola akun pengguna dan toko sayur
FR11	Admin dapat memantau aktivitas di aplikasi, termasuk pesanan dan transaksi.	UC11	Memantau aktivitas