

## BAB IV

### HASIL DAN ANALISIS PENELITIAN

#### 4.1 Analisa Perancangan Sistem

Analisis merupakan kegiatan untuk mempelajari dan mengevaluasi permasalahan atau situasi tertentu. Perancangan, di sisi lain, adalah pembuatan desain teknis berdasarkan evaluasi yang dilakukan pada tahap analisis. Sistem, dalam konteks ini, merujuk pada serangkaian elemen yang melibatkan manusia, mesin atau peralatan, prosedur, serta konsep-konsep yang digabungkan untuk mencapai tujuan bersama. Dapat disimpulkan bahwa Analisis dan Perancangan Sistem adalah suatu proses pemahaman sistem dan pembuatan desain sistem informasi berbasis komputer, dengan hasil akhir berupa sistem komputerisasi. Pengembangan sistem bertujuan untuk menyusun sistem informasi baru yang dapat mengatasi berbagai masalah di suatu organisasi dan memberikan pemahaman tentang sistem yang ada dalam organisasi tersebut serta teknik manajemen terkait sistem informasi manajemen (SIM) berbasis komputer (Indyah Hartami Santi, 2020). Analisis perancangan sistem dilakukan berdasarkan hasil analisis sistem yang sedang berjalan dan analisis kebutuhan. Dalam penelitian ini, penulis mengusulkan solusi untuk permasalahan yang ada, yakni sistem informasi pemesanan perbaikan ECU Digital berbasis web untuk menyediakan pencatatan data barang, transaksi dan laporan keuangan.

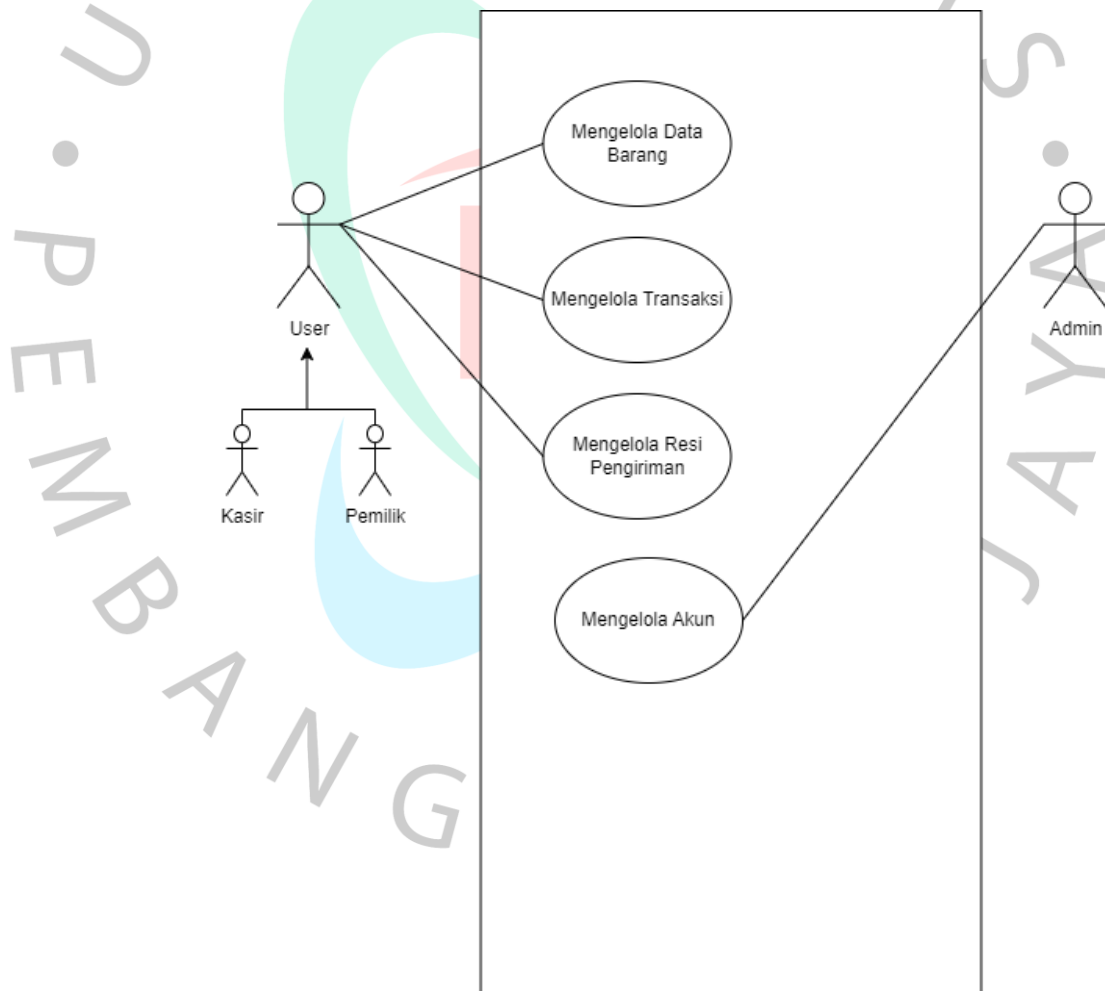
#### 4.2 Perancangan Diagram Sistem Usulan

Dalam merancang sistem, diajukan untuk menerapkan metode pendekatan *Object-Oriented Analysis and Design* (OOAD) serta memanfaatkan teknologi *Unified Modeling Language* (UML), salah satu komponen penting merupakan diagram sistem yang dilakukan untuk mengilustrasikan komponen-komponen dan berkaitan dengan komponen tersebut. beberapa diagram sistem umum dilakukan Dalam merancang sistem menggunakan pendekatan berupa: *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.

### 4.3 Requirement Analysis

#### 4.2.1 Use Case Diagram

Diagram Use Case adalah jenis diagram dalam perancangan sistem yang dipakai untuk mengilustrasikan interaksi antara aktor-aktor, baik itu pengguna atau sistem, khususnya terkait pemesanan perbaikan ECU Digital berbasis web. Semua pengguna sistem ini adalah para pekerja ECU Digital, termasuk Pemilik, Kasir, Manajer, dan Admin. Berikut merupakan ilustrasi diagram Use Case hasil perancangan sistem. **Gambar 4.1** dibawah ini merupakan gambaran interaksi.



Gambar 4.1 Use Case Diagram

#### 4.2.2 Spesifikasi Use Case

Tabel 4.1 Spesifikasi Use Case Mengelola Data Barang

<b>Use Case Name</b>	<b>Mengelola data barang</b>	
<b>Actors</b>	User (Pemilik dan Kasir)	
<b>Preconditions</b>	Login sebagai User	
<b>Postconditions</b>	User Berhasil mengelola detail toko	
	<b>Actor</b>	<b>System</b>
	1) Klik menu data barang	2) Menampilkan informasi toko berupa tabel data barang.
	3) Untuk menambahkan data barang, pengguna perlu memilih opsi "Tambah" pada antarmuka sistem	4) Menunculkan form tambah barang
	5) Untuk mengisi data barang, pengguna harus memilih opsi "Simpan" pada antarmuka sistem.	6) Memunculkan notifikasi bahwa data berhasil disimpan
	7) Jika ingin mengedit data barang maka harus memilih tombol Edit	8) Menampilkan tabel data barang yang sudah dipilih untuk di edit
	9) Jika sudah mengisi data barang maka harus memilih tombol Simpan	10) Menampilkan notifikasi bahwa data berhasil disimpan
	11) Jika ingin menghapus data barang maka harus memilih tabel data	12) Sistem akan menampilkan notifikasi bahwa

	barang lalu memilih tombol hapus	data barang berhasil dihapus
<b>Alternative Flows</b>	-	

Tabel 4.1 diatas merupakan alur yang menjelaskan mengelola data barang. Pada tabel tersebut terdapat dua aktor yaitu *User* hasil generalisasi Kasir dan Pemilik.

Tabel 4.2 Spesifikasi Use Case Mengelola Transaksi

<b>Use Case Name</b>	<b>Mengelola transaksi</b>	
<b>Actors</b>	User (Pemilik dan Kasir)	
<b>Preconditions</b>	Login sebagai User	
<b>Postconditions</b>	Berhasil mengelola barang	
<b>Succes Scenario</b>	<b>Actor</b>	<b>System</b>
	1) Klik menu transaksi	2) Menampilkan halaman utama berupa Tabel Data Transaksi
	3) Jika ingin membuat laporan pembayaran pelanggan yang sudah membayar maka harus menekan tombol tambah	4) Menampilkan Form Tambah Data Transaksi
	5) Mengisi Tabel Data Transaksi lalu menekan tombol simpan	6) Apabila formulir terisi dengan benar, maka system akan menampilkan Notifikasi

berhasil.

Apabila

pengisian

gagal, maka

akan kembali

ke bagian

Form

Tambah Data

Transaksi

makan

muncul

notifikasi

gagal

7) Ketika ingin mengedit data transaksi, maka harus menekan tombol edit

8) Menampilkan Form Edit Transaksi Keuangan yang ingin dipilih

9) Jika sudah mengisi data transaksi maka harus memilih tombol Simpan

10) Menampilkan notifikasi bahwa data berhasil disimpan

11) Jika ingin menghapus data transaksi, maka memilih data transaksi yang sudah dipilih, lalu menekan tombol Hapus

12) Jika data yang dipilih berhasil dihapus, maka sistem akan menampilkan notifikasi

<b>Alternative Flows</b>	-	“Data Berhasil Dihapus”.
--------------------------	---	--------------------------

Tabel 4.2 diatas merupakan alur yang menjelaskan mengelola data barang. Pada tabel tersebut terdapat dua aktor yaitu *User* hasil generalisasi Kasir dan Pemilik.

Tabel 4.3 Spesifikasi Use Case Mengelola Resi Pengiriman

<b>Use Case Name</b>	<b>Mengelola Resi Pengiriman</b>	
<b>Actors</b>	User (Kasir dan Pemilik)	
<b>Preconditions</b>	Login sebagai User	
<b>Postconditions</b>	Berhasil mengelola pemesanan	
<b>Succes Scenario</b>	<b>Actor</b>	<b>System</b>
	1) Masuk ke menu Resi Pengiriman	2) Menampilkan halaman utama tabel nomor resi
	3) Jika ingin memasukan nomor resi barang yang sudah dikirim maka harus menekan tombol tambah	4) Menampilkan Form Tambah Nomor Resi
	5) Ketika ingin mengedit resi, maka harus menekan tombol edit	6) Menampilkan Form Edit resi yang ingin dipilih
	7) Jika sudah mengisi data transaksi maka harus memilih tombol Simpan	8) Menampilkan notifikasi bahwa data berhasil disimpan

	9) Jika ingin menghapus data transaksi, maka memilih data transaksi yang sudah dipilih, lalu menekan tombol Hapus	10) Jika data yang dipilih berhasil dihapus, maka sistem akan menampilkan notifikasi “Data Berhasil Dihapus”.
<b>Alternative Flows</b>	-	

Tabel 4.3 diatas merupakan alur yang menjelaskan mengelola resi barang. Pada tabel tersebut terdapat dua aktor yaitu *User* hasil generalisasi Kasir dan Pemilik.

Tabel 4.4 Spesifikasi Use Case Mengelola Akun

<b>Use Case Name</b>	<b>Kelola akun</b>	
<b>Actors</b>	Admin	
<b>Preconditions</b>	Login sebagai Admin	
<b>Postconditions</b>	Berhasil Login Sebagai Admin	
<b>Succes Scenario</b>	<b>Actor</b>	<b>System</b>
	1) Login Sebagai Admin	2) Menampilkan halaman utama berupa Tabel Data User
	3) Jika Admin hendak menambahkan pengguna, langkah yang perlu diambil adalah memilih opsi "Tambah" pada antarmuka sistem.	4) Muncul Tabel Tambah User

<p>5) Jika admin ingin menambah pengguna, langkah yang harus diambil adalah dengan mengklik tombol "Tambah".</p>	<p>6) Muncul halaman tambah</p>
<p>7) Mengisi Form User lalu menekan Tombol Simpan</p>	<p>8) Berhasil menyimpan data</p>
<p>9) Ketika ingin mengubah data barang, klik Edit</p>	<p>10) Menampilkan Form Edit User, Lalu memilih data User yang ingin di Edit</p>
<p>11) Melakukan edit data pada data user yang sudah dipilih</p>	<p>12) Sistem menampilkan Notifikasi Berhasil, kembali ke bagian Form Edit</p>
<p>13) Jika Manajer ingin menghapus data user, maka Manajer harus memilih data user yang ingin dipilih, lalu memilih tombol Delete</p>	<p>14) Jika data yang dipilih dapat dihapus, sistem akan menampilkan pesan "Data Berhasil Dihapus",</p>

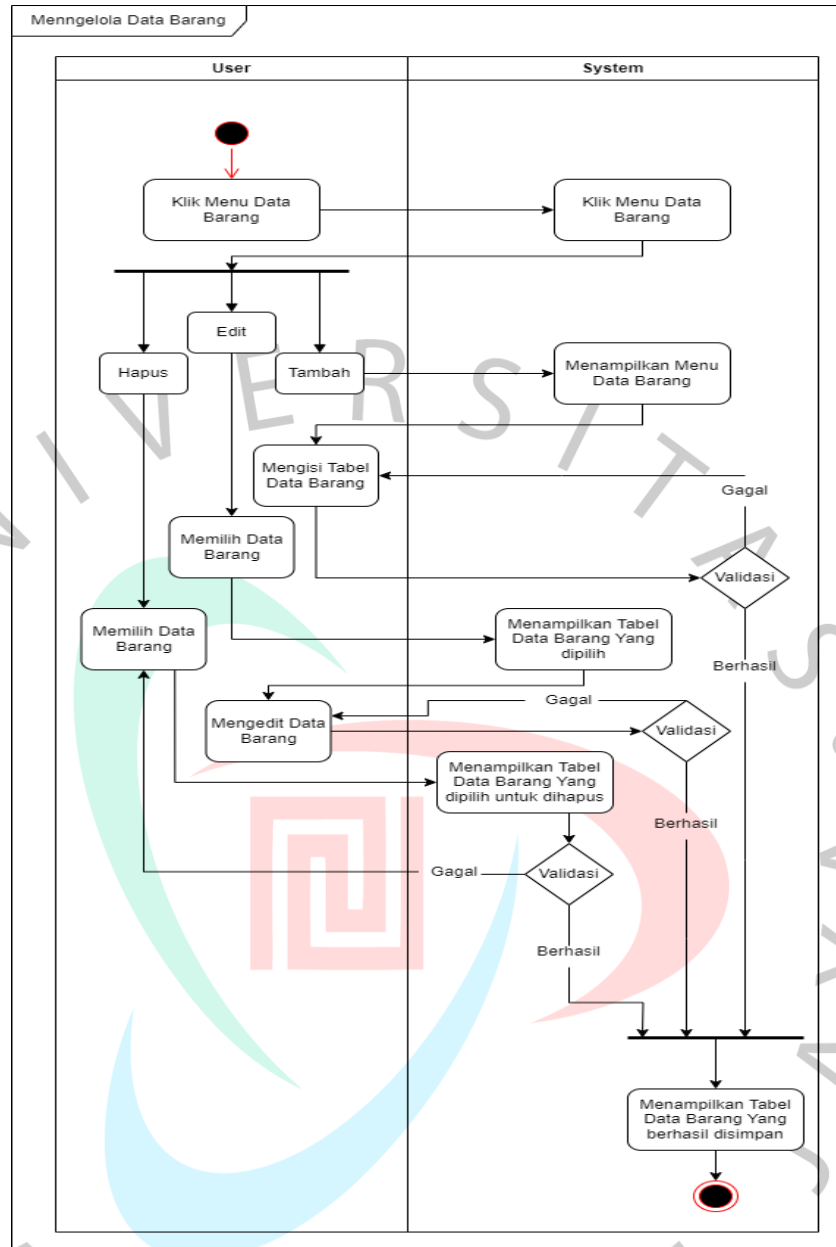


		dan kemudian kembali ke menu yang menampilkan tabel data pengguna.
<b>Alternative Flows</b>		.

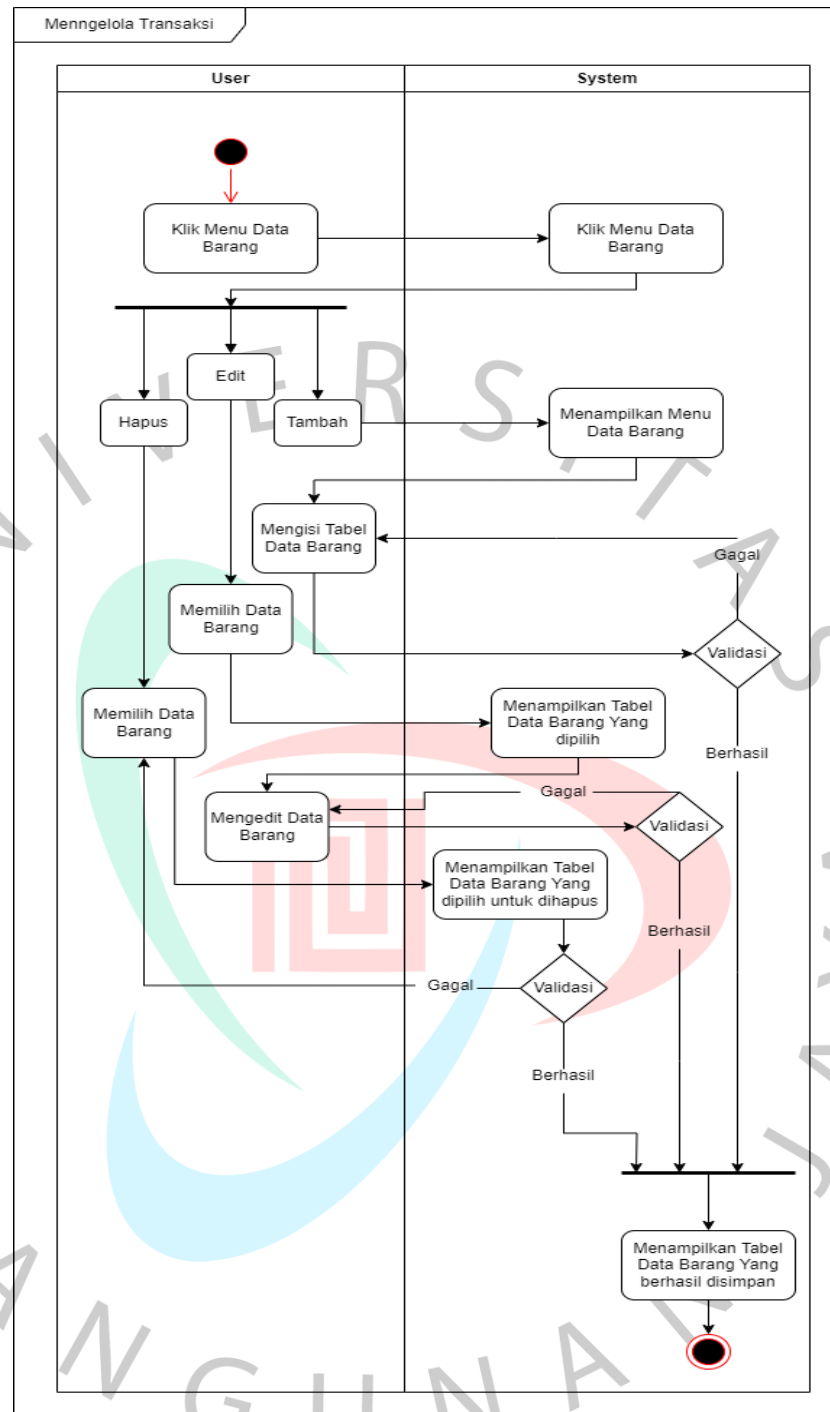
Tabel 4.4 diatas merupakan alur yang menjelaskan mengelola akun dan hak akses para pengguna. Pada tabel tersebut terdapat Manajer.

#### 4.4 Activity Diagram

Diagram aktivitas merupakan suatu diagram dalam Unified Modeling Language (UML) yang berfungsi untuk memvisualisasikan alur kerja atau urutan aktivitas dalam suatu sistem, proses bisnis, atau fungsi tertentu. Diagram ini digunakan untuk membantu pemahaman dan dokumentasi mengenai aktivitas-aktivitas yang terjadi dalam suatu konteks. membantu dalam memodelkan dinamika proses dan memberikan gambaran visual tentang bagaimana aktivitas-aktivitas tersebut dijalankan dan bagaimana mereka saling berinteraksi. Berikut ini merupakan hasil dari rancangan *Activity Diagram*. Pada gambar dibawah ini ialah *Activity* Kelola Data Barang. Pada **Gambar 4.2** dibawah ini merupakan alur user hasil dari generalisasi pemilik dan kasir ketika mengelola data barang.

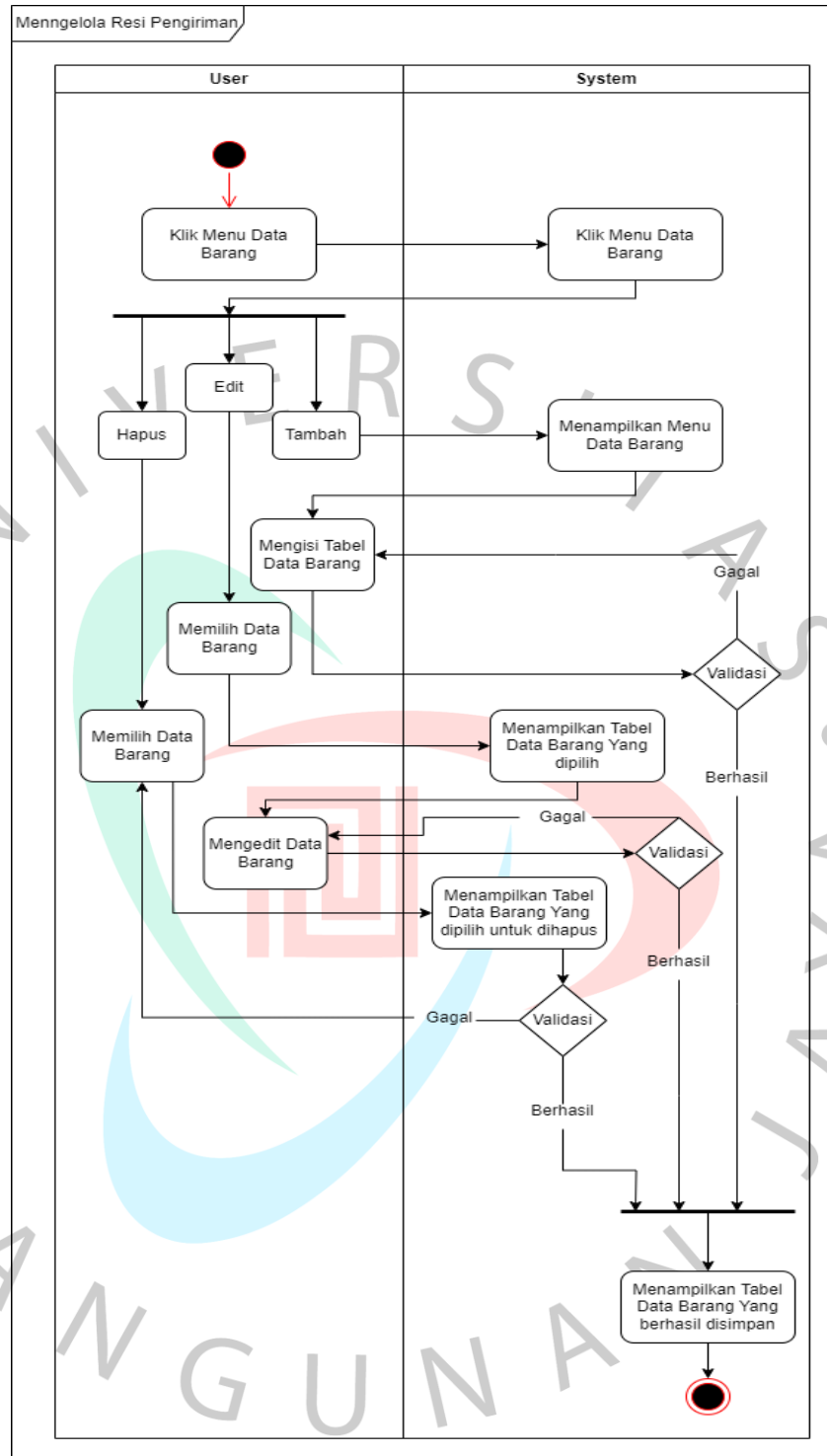


Gambar 4.2 Activity Diagram Mengelola Data Barang



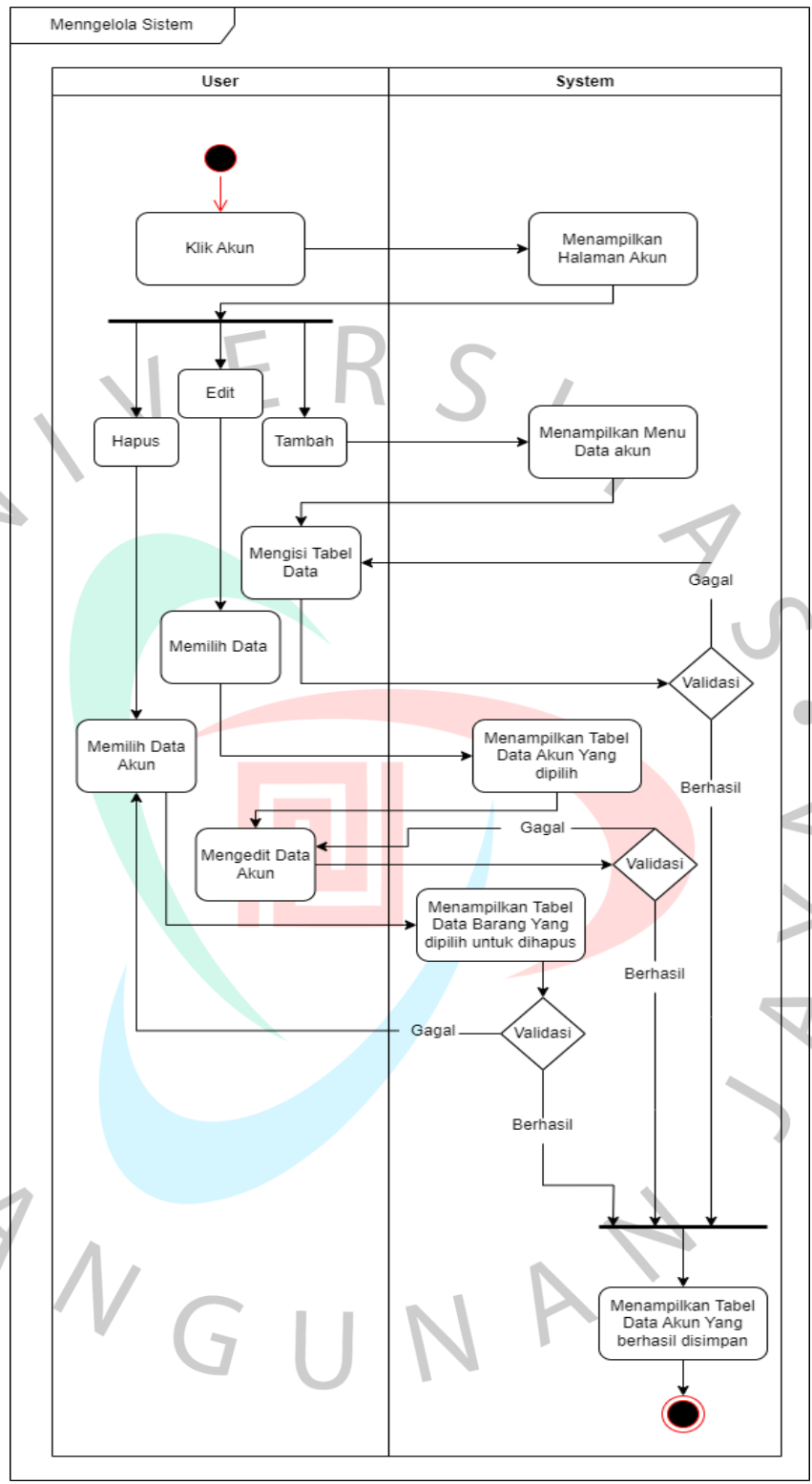
Gambar 4.3 Activity Diagram Mengelola Transaksi

Pada **Gambar 4.3** diatas merupakan sebuah *Diagram* Mengelola Transaksi yang menggambarkan alur proses penambahan transaksi, mengedit transaksi dan menghapus transaksi.



Gambar 4.4 Activity Diagram Kelola Resi

Pada **Gambar 4.4** diatas merupakan sebuah *Activity Diagram* Mengelola Resi Pengiriman yang menggambarkan alur proses penambahan nomor resi, mengedit nomor resi dan menghapus nomor resi.



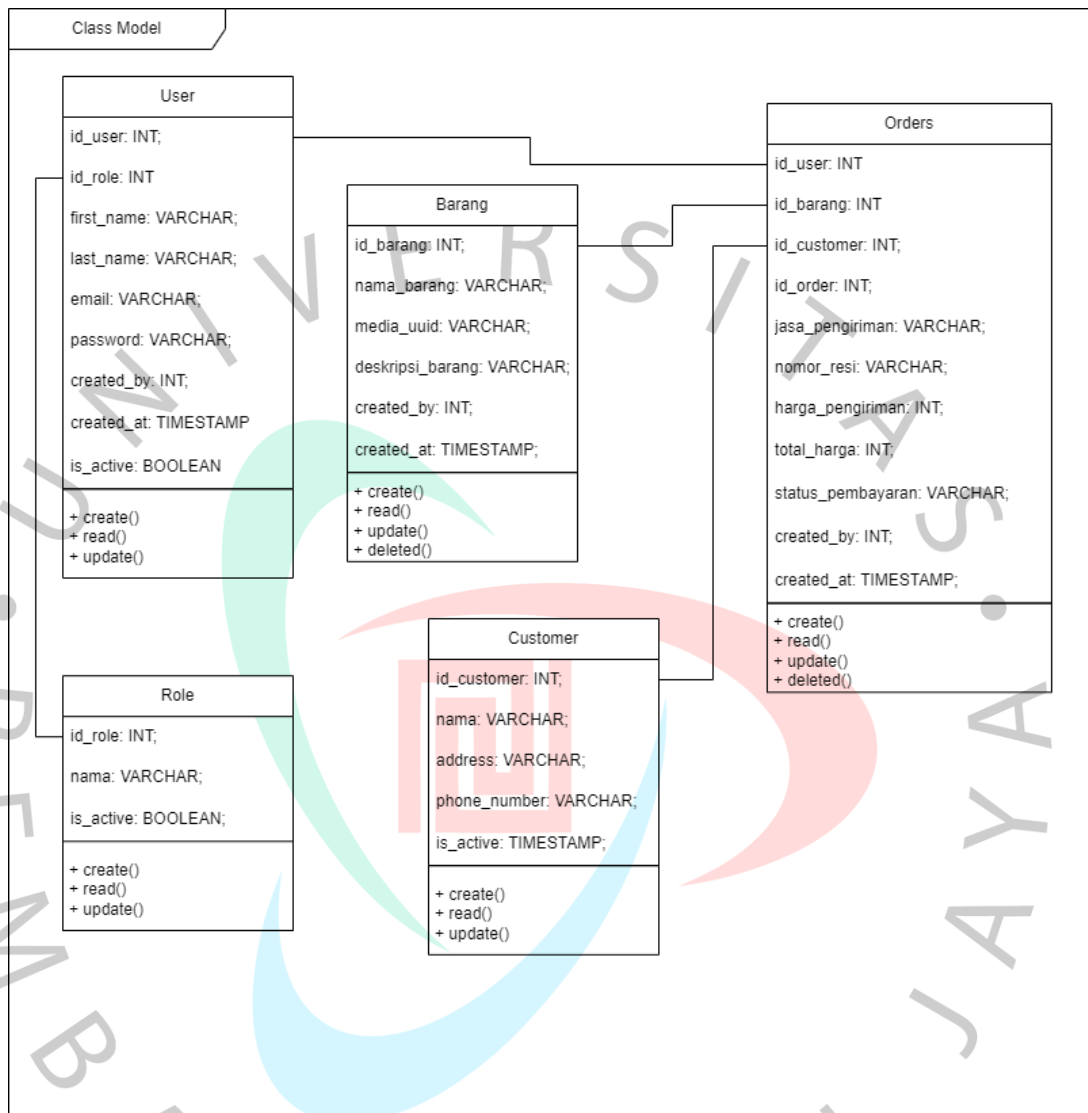
Gambar 4.5 Activity Diagram Kelola Akun

Pada **Gambar 4.5** diatas merupakan sebuah *Activity Diagram* Mengelola Akun yang menggambarkan alur proses penambahan Akun, mengedit Akun dan menghapus Akun.

#### 4.4.1 Class Diagram

Diagram kelas, juga dikenal sebagai Class Diagram, merupakan jenis diagram dalam rekayasa perangkat lunak yang bertujuan untuk mengilustrasikan struktur statis dari suatu sistem atau aplikasi berbasis objek. Diagram kelas menampilkan kelas-kelas yang terdapat dalam sistem, relasi di antara kelas-kelas ini dan properti serta metode masing-masing kelas. Diagram tersebut terdapat dalam notasi Unified Modeling Language (UML) yang digunakan untuk menggambarkan desain dan struktur perangkat lunak secara detail, meliputi informasi tentang kelas, atribut, metode, serta hubungan atau relasi di antara mereka. Fungsinya sangat membantu pengembang perangkat lunak dalam mengonseptualisasikan struktur objek dalam sistem, memahami interaksi di antara objek, dan merancang kelas-kelas. Diagram kelas sering digunakan sebagai langkah awal dalam proses pengembangan perangkat lunak untuk menguraikan struktur dasar suatu sistem sebelum melangkah ke tahap implementasi yang lebih lanjut. Setiap pengguna harus memiliki setidaknya satu peran, bisa lebih dari satu peran; setiap pesanan minimal terhubung dengan satu barang dan satu pelanggan. Sebaliknya, satu barang tidak wajib terkait dengan pesanan, dan boleh memiliki atau tidak memiliki pesanan. Pada Tabel 4.5 berikut, dijelaskan Class Diagram dari sistem yang akan dibangun.

Tabel 4.5 Class Diagram



#### 4.2.4 Spesifikasi Class Diagram

Spesifikasi *Class Diagram* adalah dokumen yang merinci elemen-elemen dan relasi-relasi yang ada dalam diagram kelas suatu sistem perangkat lunak. Dokumen ini memberikan deskripsi rinci tentang kelas-kelas, atribut, metode, dan hubungan-hubungan di antara mereka. Spesifikasi ini membantu tim pengembangan dan pemangku kepentingan untuk memahami dengan jelas struktur statis dari sistem yang direpresentasikan dalam diagram kelas. Di bawah ini terdapat **Tabel 4.6** yang

mendeskripsikan isi dari tabel terdapat *id* yang menjadi *Primary Key* yang digunakan sebagai tempat penyimpanan data dari pengguna.

Tabel 4.6 Spesifikasi Class Diagram User

Field Name	Data Type	Size	Description
id_user	INT	-	Primary Key (Auto Increment)
id_role	INT	-	Setiap pengguna memiliki Id user
first_name	VARCHAR	255	Nama depan
last_name	VARCHAR	255	Nama belakang
email	VARCHAR	255	Nama email
password	VARCHAR	255	Password
created_by	INT	-	Nama pembuat
created_at	TIMESTAMP	-	Tanggal dibuat
is_active	BOOLEAN	-	Aktif atau tidak aktif

Tabel 4.7 Class Diagram Roles

Field Name	Data Type	Size	Description
id_role	INT	-	Primary Key (Auto Increment)
name	VARCHAR	255	Nama depan
is_active	BOOLEAN	-	Aktif atau tidak aktif

Pada **Tabel 4.7** diatas yang mendeskripsikan isi dari tabel terdapat *id* yang menjadi *Primary Key* yang digunakan sebagai tempat penyimpanan data dari pengguna. Jadi setiap karyawan yang ingin *login* harus memiliki 1 (satu) akun atau *Id* lebih dari 1 agar sistem dapat menampilkan menu halaman.



Tabel 4.8 Class Diagram barang

Field Name	Data Type	Size	Description
id_barang	INT	-	Primary Key (Auto Increment)
nama_barang	VARCHAR	255	Nama barang
media_uuid	VARCHAR	255	Jenis barang
deskripsi_barang	VARCHAR	255	Kendala yang dialami
created_by	INT	-	Nama pembuat
created_at	TIMESTAMP	-	Tanggal dibuat

Pada **Tabel 4.8** diatas yang mendeskripsikan isi dari tabel barang terdapat *id* yang menjadi *Primery Key* yang digunakan sebagai tempat penyimpanan data dari pengguna. Jadi setiap ada barang yang masuk harus mengisi tabel sesuai dengan deskripsi tabel.

Tabel 4.9 Class Diagram orders

Field Name	Data Type	Size	Description
id_order	INT	-	Primary Key (Auto Increment)
id_user	INT	-	Setiap User memiliki Id
id_barang	INT	-	Setiap barang memiliki Id barang
id_customer	INT	255	Id pelanggan
nomor_resi	VARCHAR	255	Nomor resi
jasa_pengiriman	VARCHAR	255	Ekspedisi pengiriman
harga_pengiriman	INT	255	Biaya pengiriman
total_harga	INT	-	Nama pembuat

status_pembayaran	VARCHAR	255	Tanggal dibuat
is_active	BOOLEAN	-	Aktif atau tidak aktif

Pada **Tabel 4.9** diatas yang mendeskripsikan isi dari tabel pemesanan terdapat *id* yang menjadi *Primery Key* yang digunakan sebagai tempat penyimpanan data dari pengguna. Jadi setiap ada barang yang sudah selesai diperbaiki maka harus membuat tagihan yang harus dibayar oleh pelanggan.

Tabel 4.10 Class Diagram customer

Field Name	Data Type	Size	Description
id_customer	INT	-	Primary Key (Auto Increment)
name	VARCHAR	255	Nama pelanggan
address	VARCHAR	255	Alamat pelanggan
phone_number	VARCHAR	255	Nomor telephone pelanggan
is_actived	BOOLEAN	-	Aktif atau tidak aktif

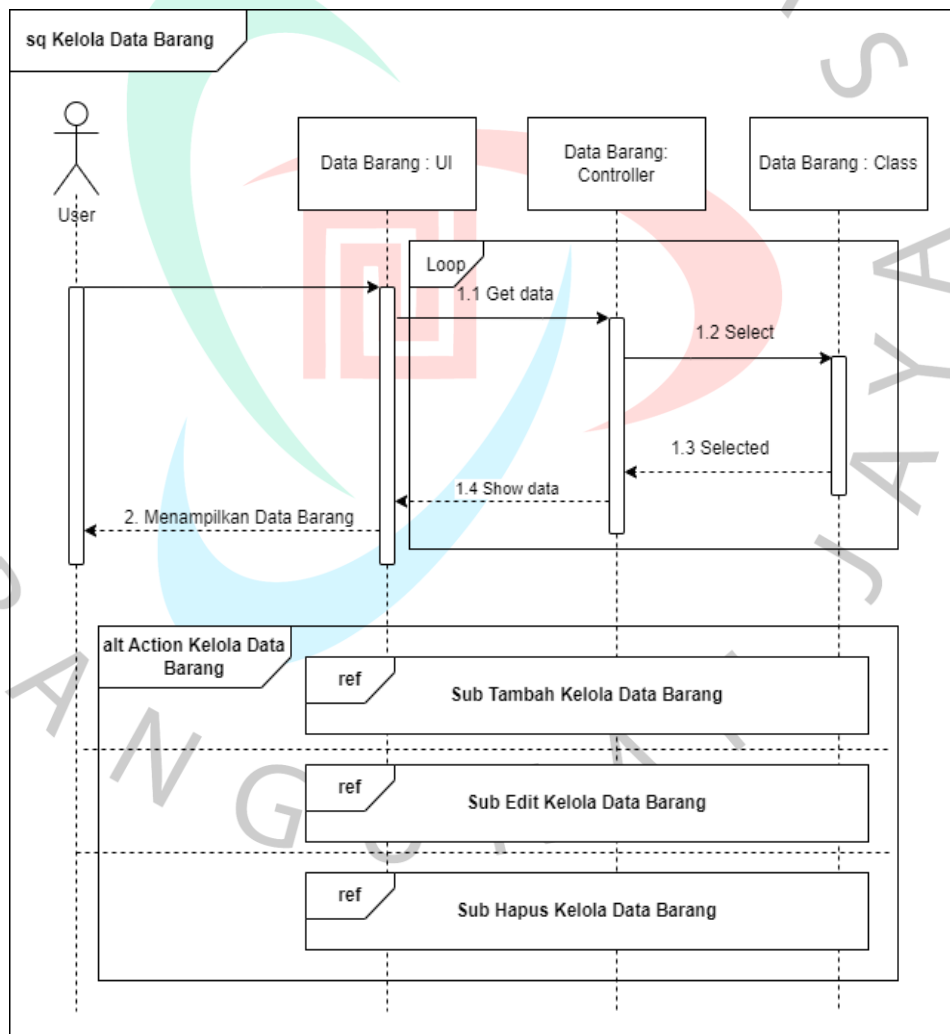
Pada **Tabel 4.10** diatas yang mendeskripsikan isi dari tabel pelanggan terdapat *id* yang menjadi *Primery Key* yang digunakan sebagai tempat penyimpanan data dari pengguna. Jadi setiap ada barang yang sudah selesai diperbaiki maka harus membuat tagihan yang harus dibayar oleh pelanggan.

#### 4.2.5 Sequence Diagram

Diagram urutan sequence diagram adalah relasi alur diagram pada *Unified Modeling Language (UML)* yang menampilkan gambaran berupa objek pada suatu sistem pada suatu waktu tertentu. Diagram urutan memberikan gambaran visual tentang bagaimana objek-objek berkomunikasi satu sama lain dan berinteraksi dalam suatu skenario tertentu atau proses bisnis. Elemen-elemen utama dalam diagram urutan melibatkan

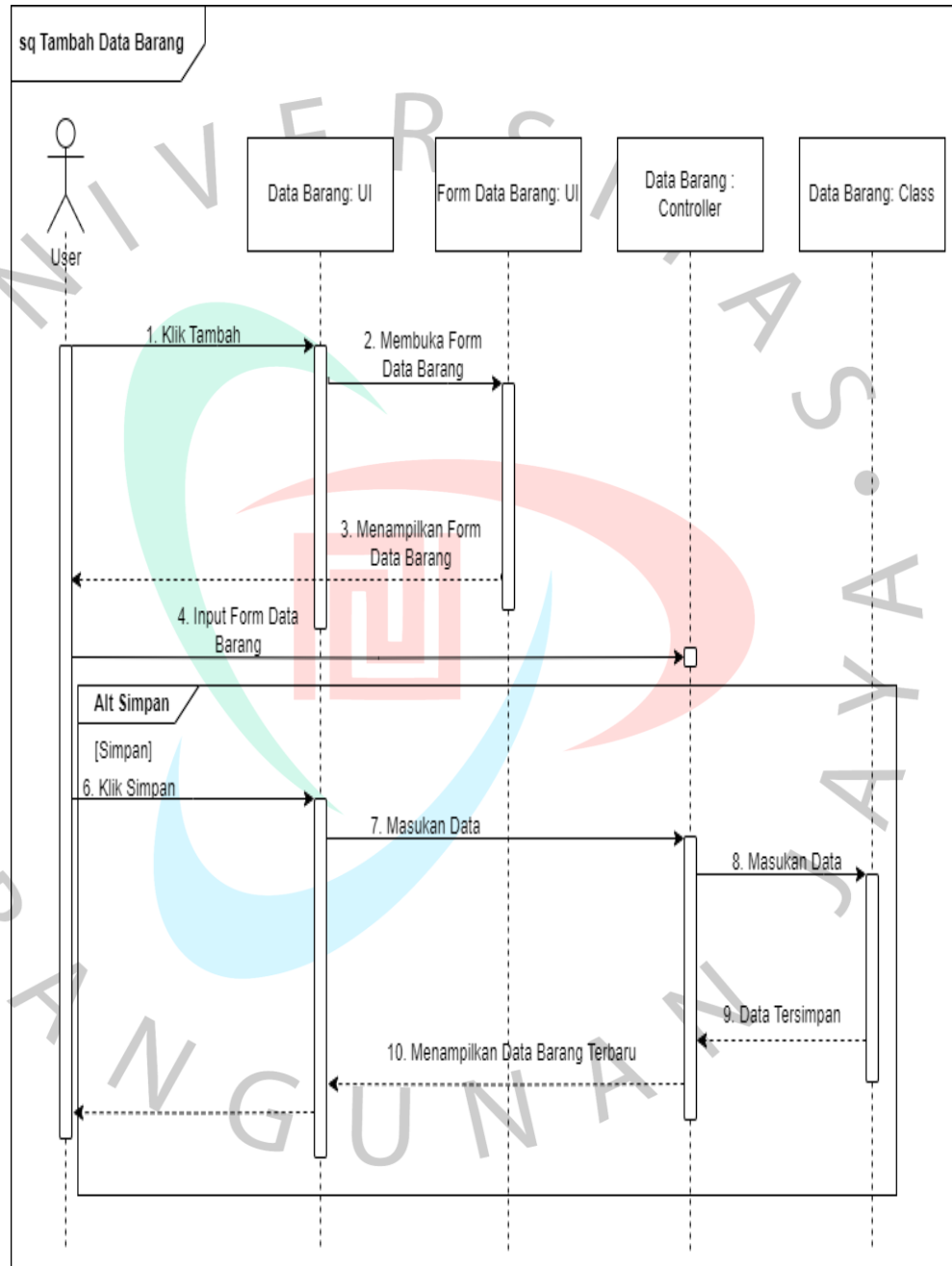
objek-objek dan pesan-pesan yang dikirim antara objek-objek tersebut terdapat Sub Tambah, Edit dan Hapus yang dibuat alternatif *flow* dibuat secara terpisah dari diagram yang dibuat. Diagram urutan membantu pengembang perangkat lunak untuk memahami bagaimana objek-objek berinteraksi dalam suatu sistem, termasuk urutan pesan yang dikirim dan *respons* yang diberikan. Ini dapat digunakan selama fase analisis dan desain untuk merinci logika dan aliran kontrol dalam suatu proses atau skenario.

a) Mengelola Data Barang



Gambar 4.6 Mengelola Data Barang

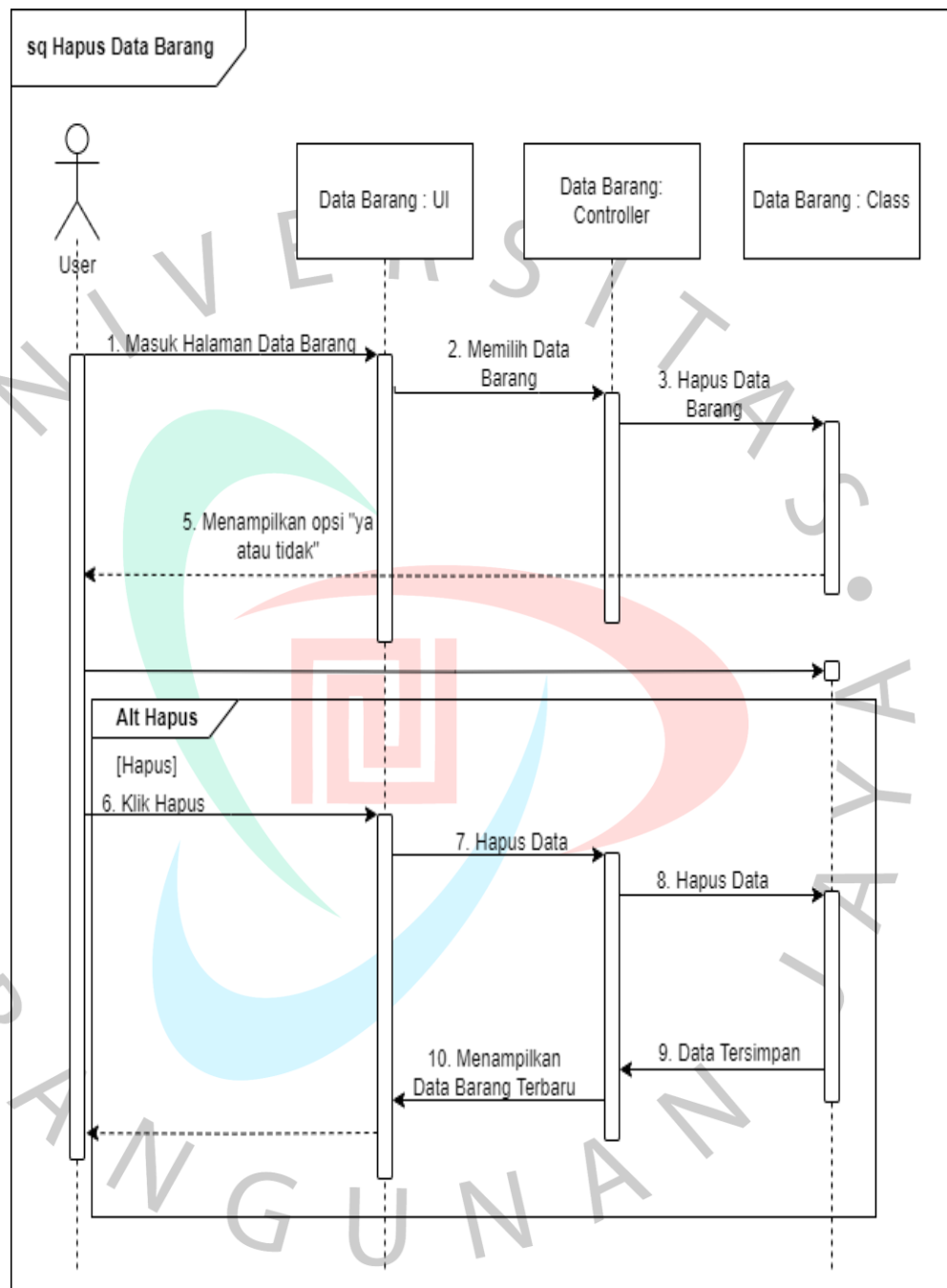
Pada **Gambar 4.6** diatas menjelaskan alur untuk mengelola data barang. Pada gambar tersebut terdapat aktor yaitu *user* yang merupakan hasil dari generalisasi kasir dan pemilik, didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.



Gambar 4.7 Tambah Data Barang

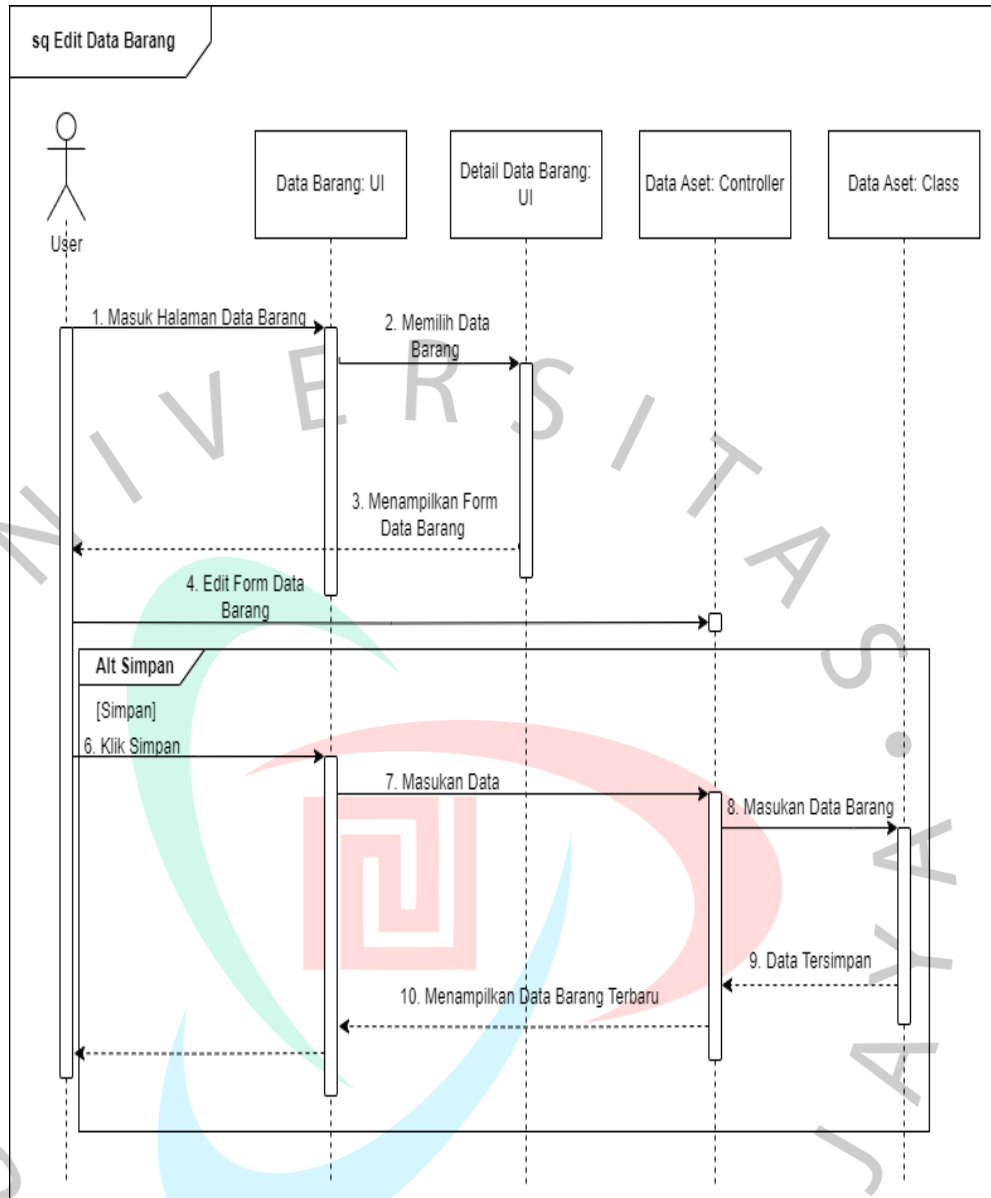
Pada **Gambar 4.7** diatas menjelaskan alur untuk tambah data barang. Pada gambar tersebut terdapat aktor yaitu *user* yang

merupakan hasil dari generalisasi kasir dan pemilik, didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.



Gambar 4.8 Hapus Data Barang

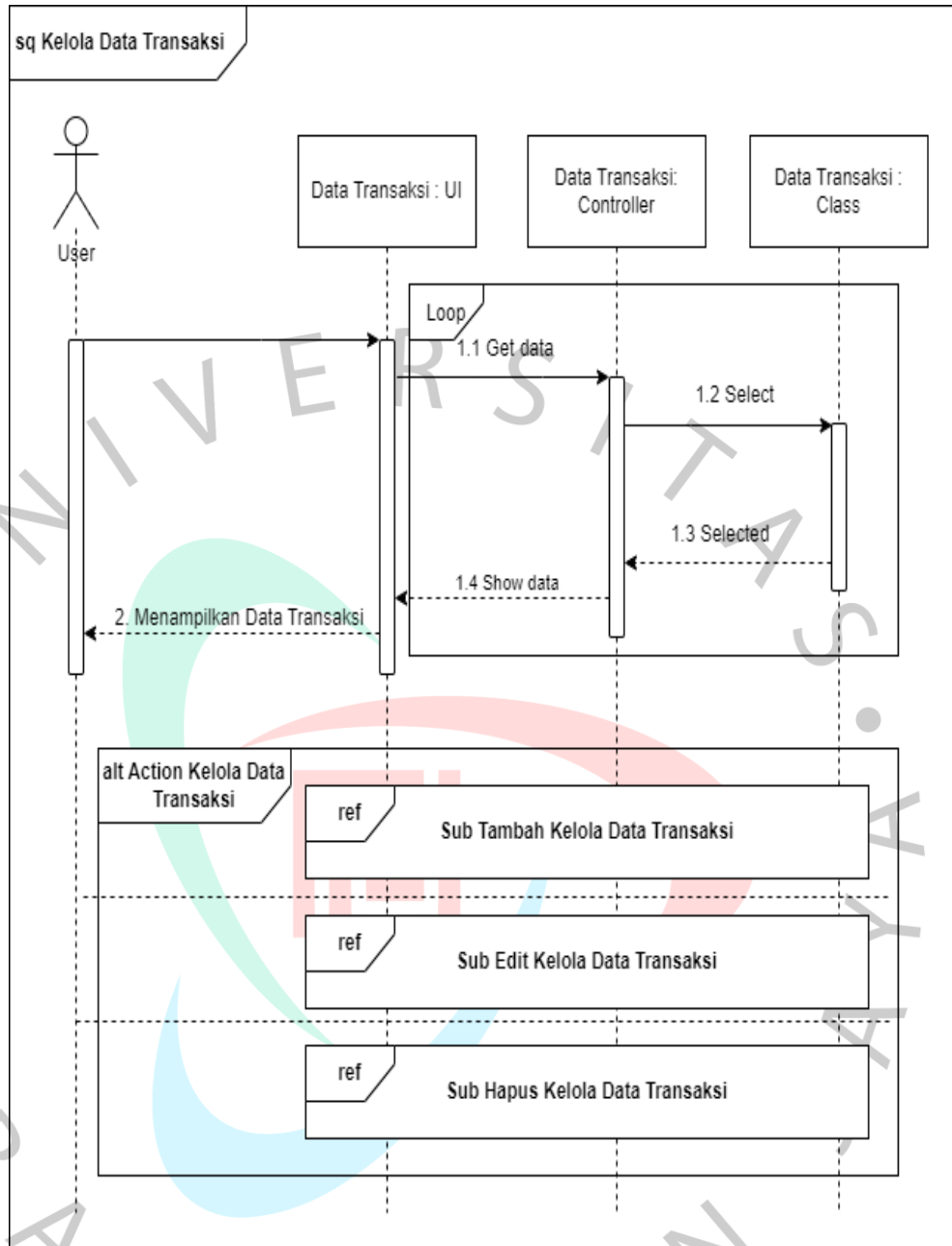
Pada **Gambar 4.8** diatas menjelaskan alur untuk hapus data barang. Pada gambar tersebut terdapat aktor yaitu *user* yang merupakan hasil dari generalisasi kasir dan pemilik, didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.



Gambar 4.9 Edit Data Barang

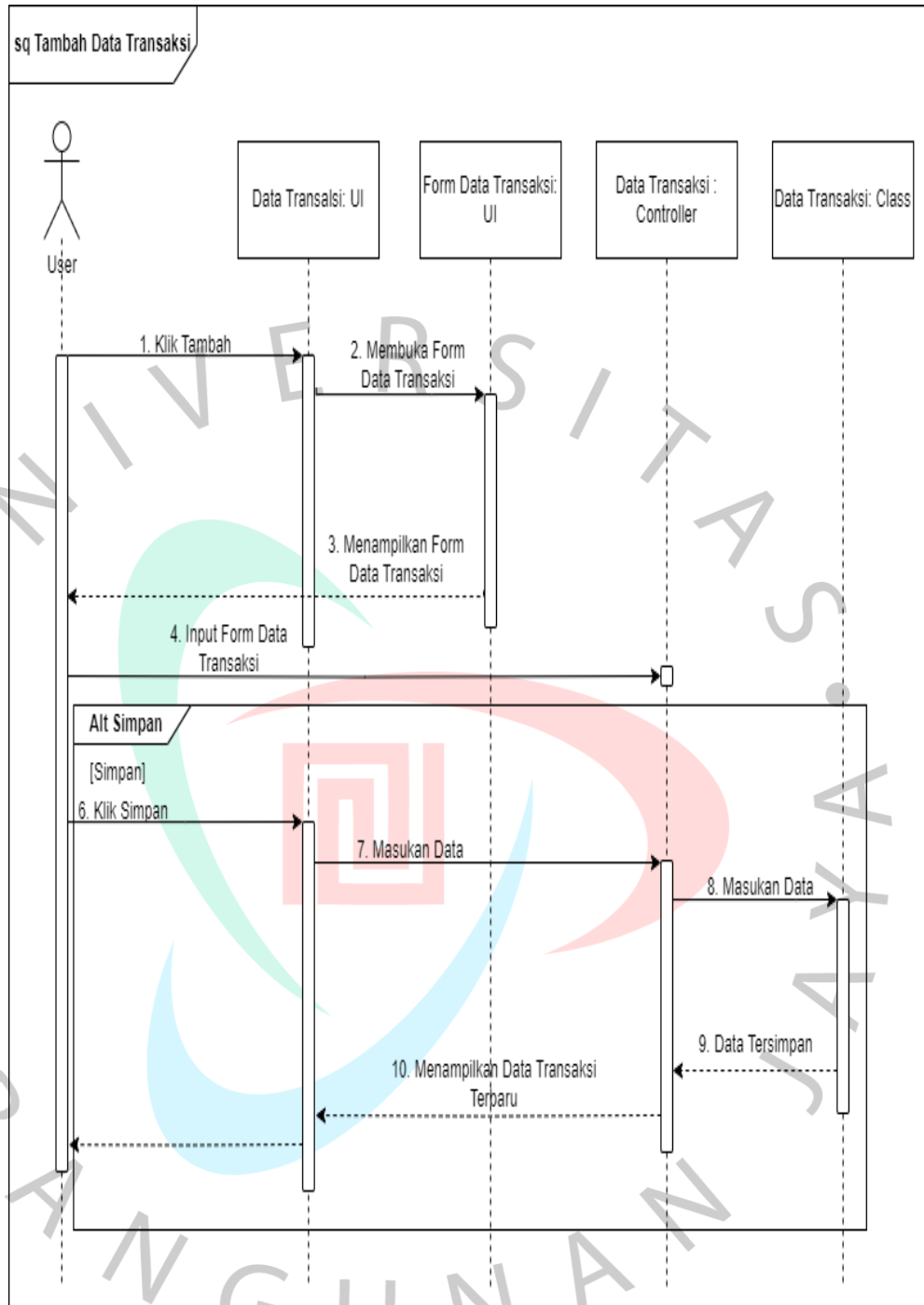
Pada **Gambar 4.9** diatas menjelaskan alur untuk edit data barang. Pada gambar tersebut terdapat aktor yaitu *user* yang merupakan hasil dari generalisasi kasir dan pemilik, didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.

b) Mengelola Transaksi



Gambar 4.10 Kelola Transaksi

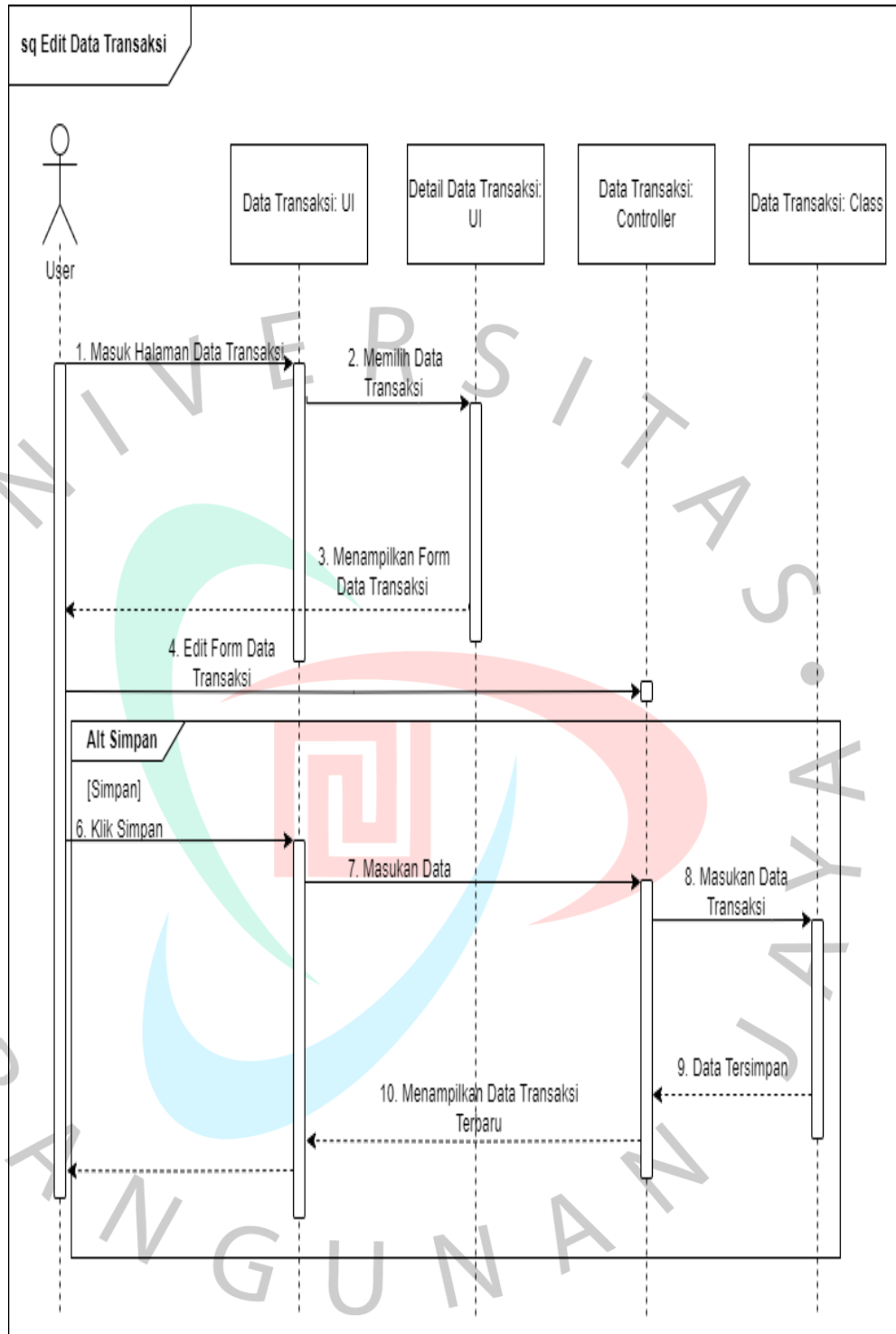
Pada **Gambar 4.10** diatas menjelaskan alur untuk mengelola data transaksi. Pada gambar tersebut terdapat aktor yaitu kasir yang memiliki akses tambah data transaksi, edit data transaksi, hapus data transaksi didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.



Gambar 4.11 Tambah Data Transaksi

Pada **Gambar 4.11** diatas mendeskripsikan proses untuk Tambah Data transaksi. Pada gambar tersebut terdapat aktor yaitu *user* hasil dari generalisasi kasir dan pemilik yang memiliki akses tambah data transaksi didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.

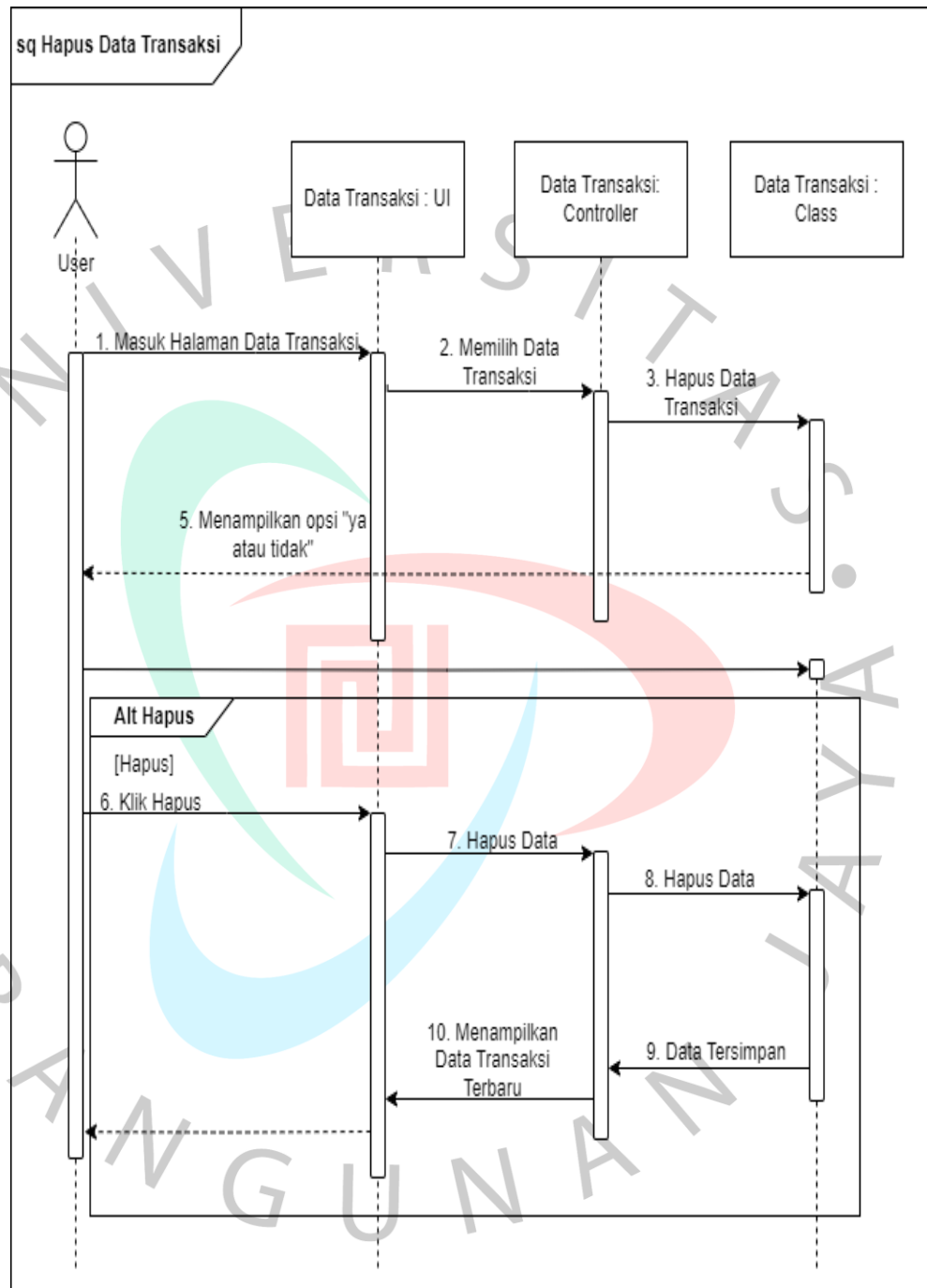




Gambar 4.12 Edit Data Transaksi

Pada **Gambar 4.13** diatas mendeskripsikan proses sistem berjalan untuk edit data transaksi. Pada gambar tersebut terdapat aktor yaitu *user* hasil dari generalisasi kasir dan pemilik yang memiliki akses

tambah data transaksi didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.

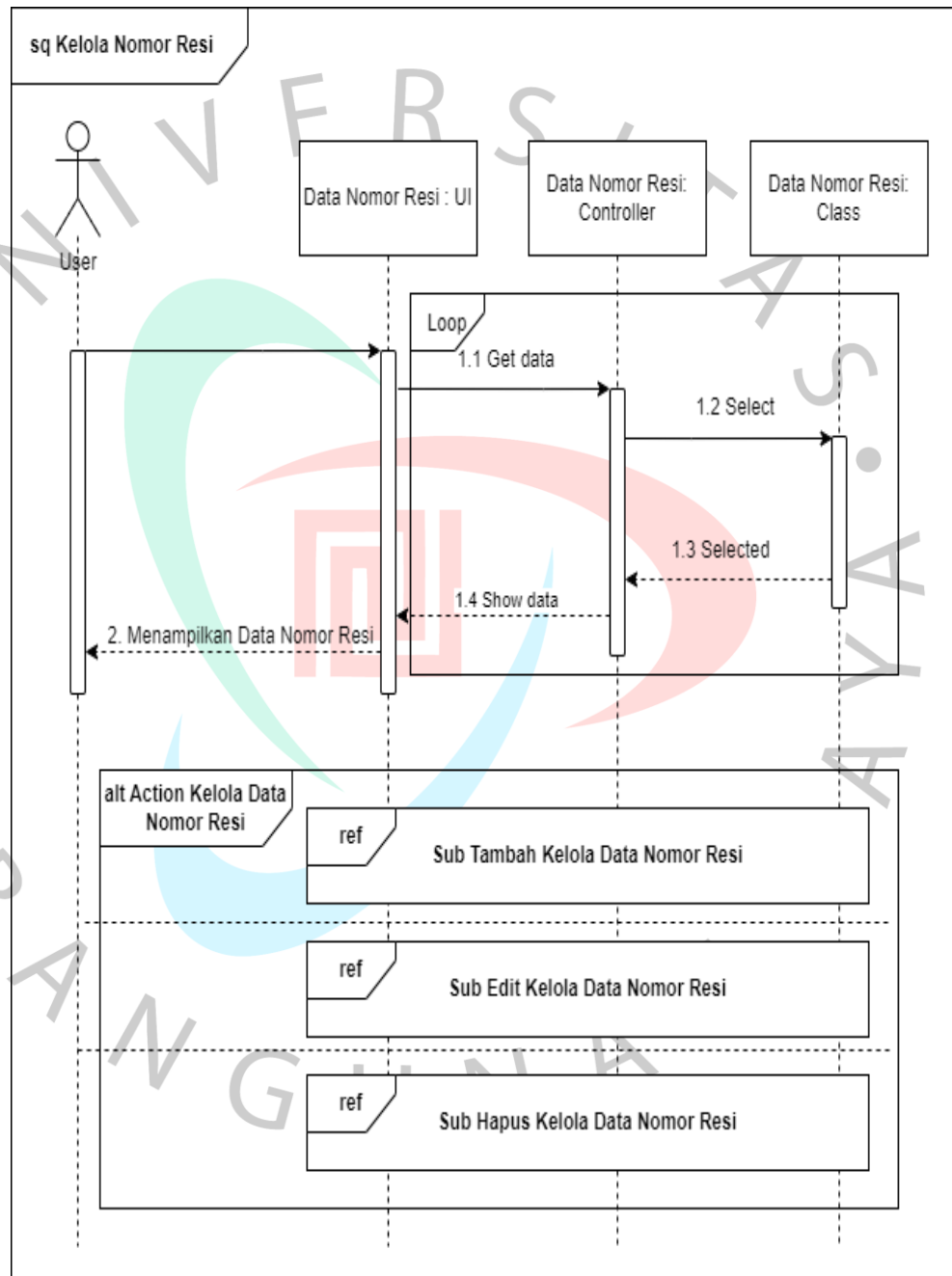


Gambar 4.13 Hapus Data Transaksi

Pada **Gambar 4.14** diatas menjelaskan alur untuk Hapus data transaksi. Pada gambar tersebut terdapat aktor yaitu *user* hasil dari

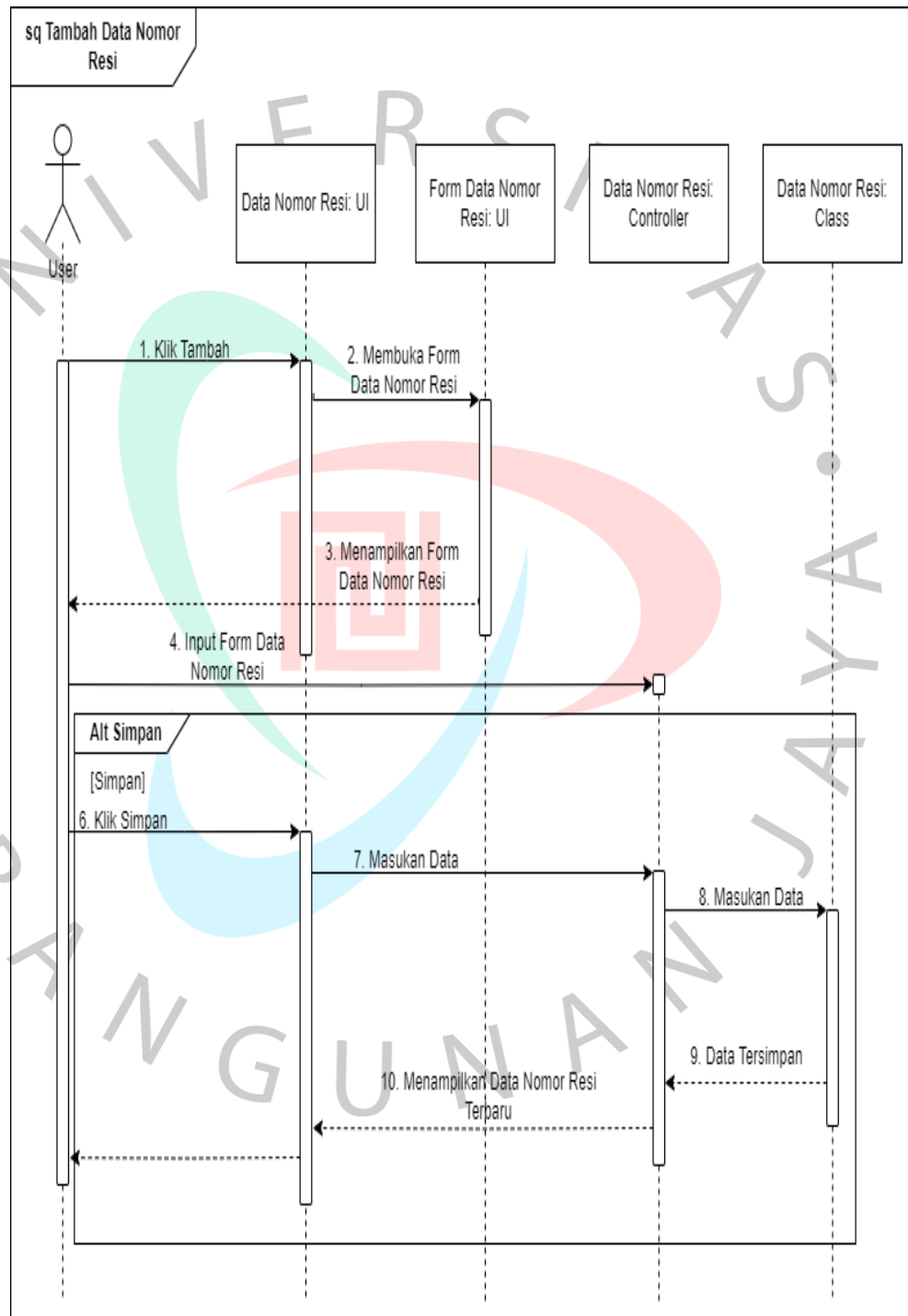
generalisasi kasir dan pemilik yang memiliki akses tambah data transaksi didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.

c) Mengelola Resi Pengiriman



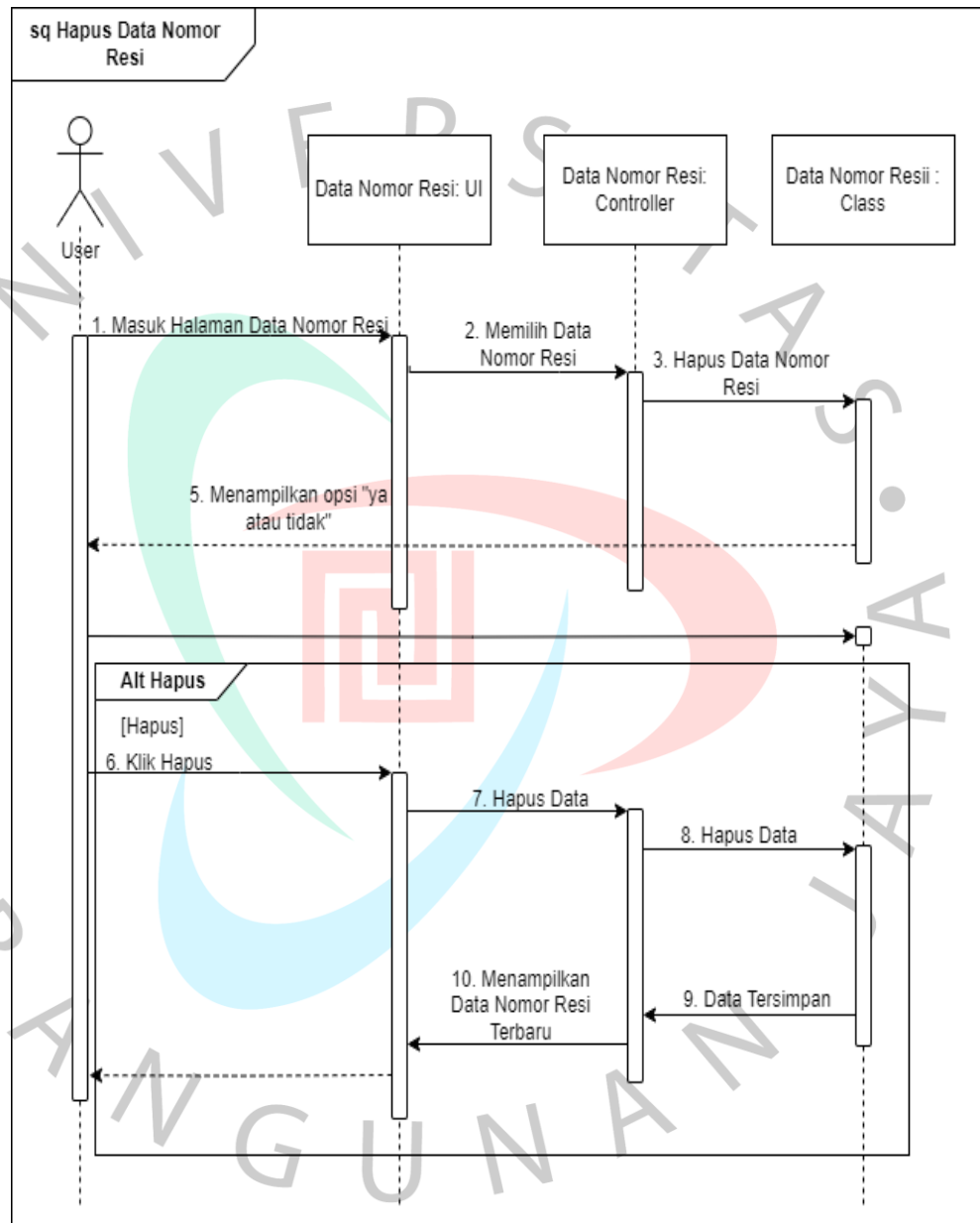
Gambar 4.14 Mengelola Resi Pengiriman

Pada **Gambar 4.15** diatas menjelaskan alur untuk Kelola data nomor resi. Pada gambar tersebut terdapat aktor yaitu *user* hasil dari generalisasi kasir dan pemilik yang memiliki akses tambah data transaksi didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.



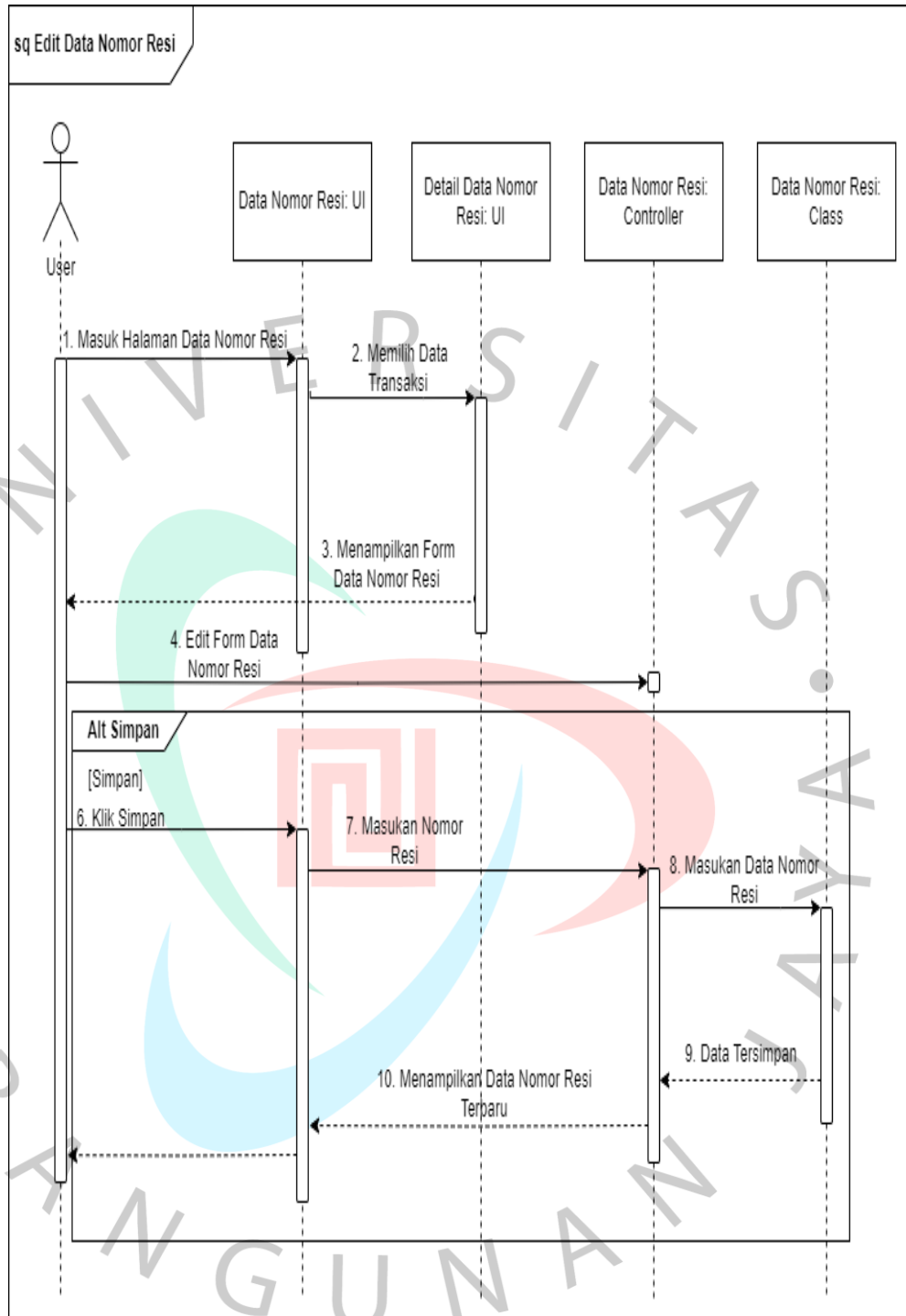
Gambar 4.15 Tambah Data Nomor Resi

Pada **Gambar 4.16** diatas menjelaskan alur untuk Tambah data nomor resi. Pada gambar tersebut terdapat aktor yaitu *user* hasil dari generalisasi kasir dan pemilik yang memiliki akses tambah data transaksi didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.



Gambar 4.16 Hapus Data Nomor Resi

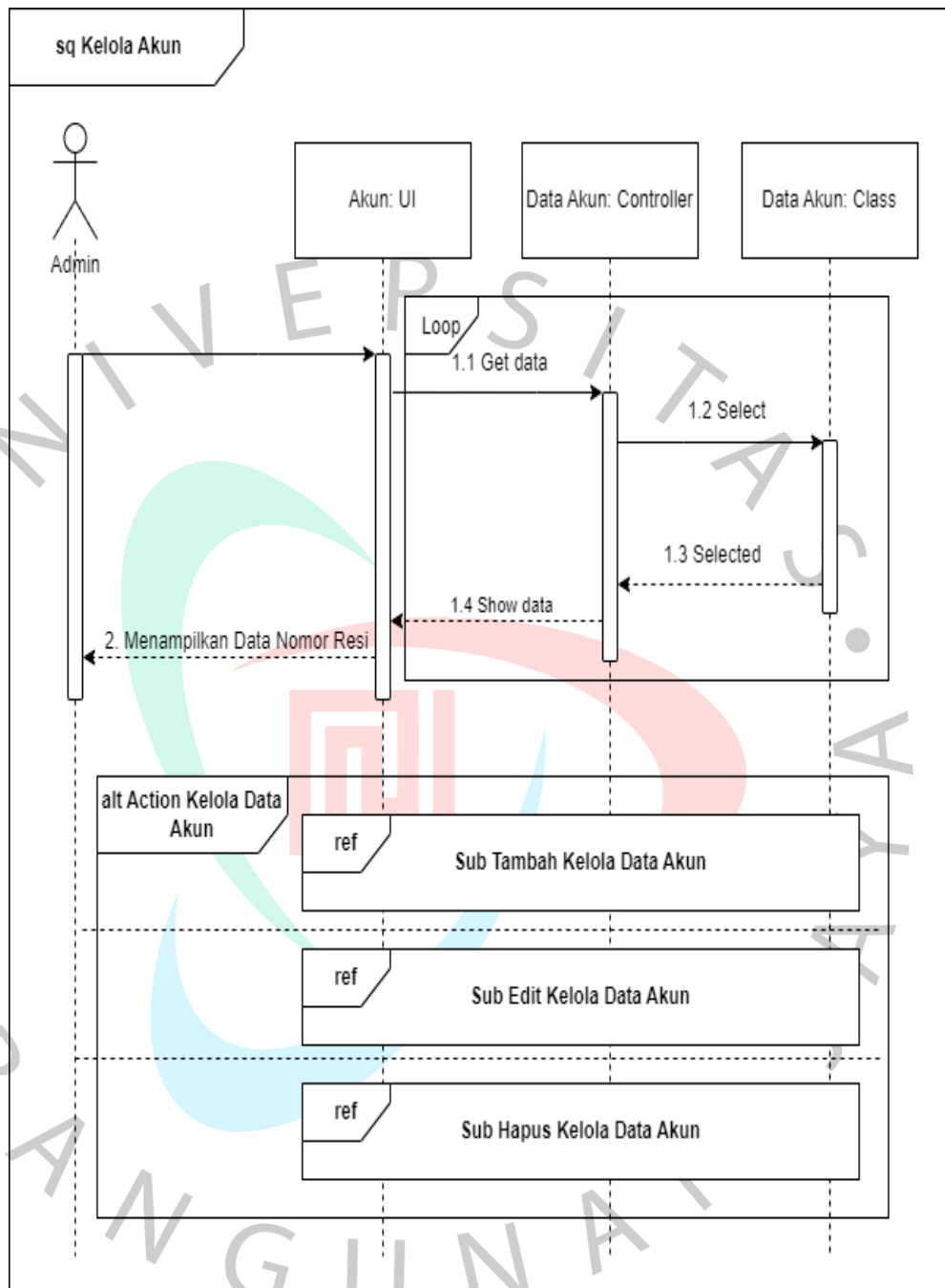
Pada **Gambar 4.17** diatas menjelaskan alur untuk Hapus data nomor resi. Pada gambar tersebut terdapat aktor yaitu *user* hasil dari generalisasi kasir dan pemilik yang memiliki akses hapus data nomor resi didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.



Gambar 4.17 Edit Data Nomor Resi

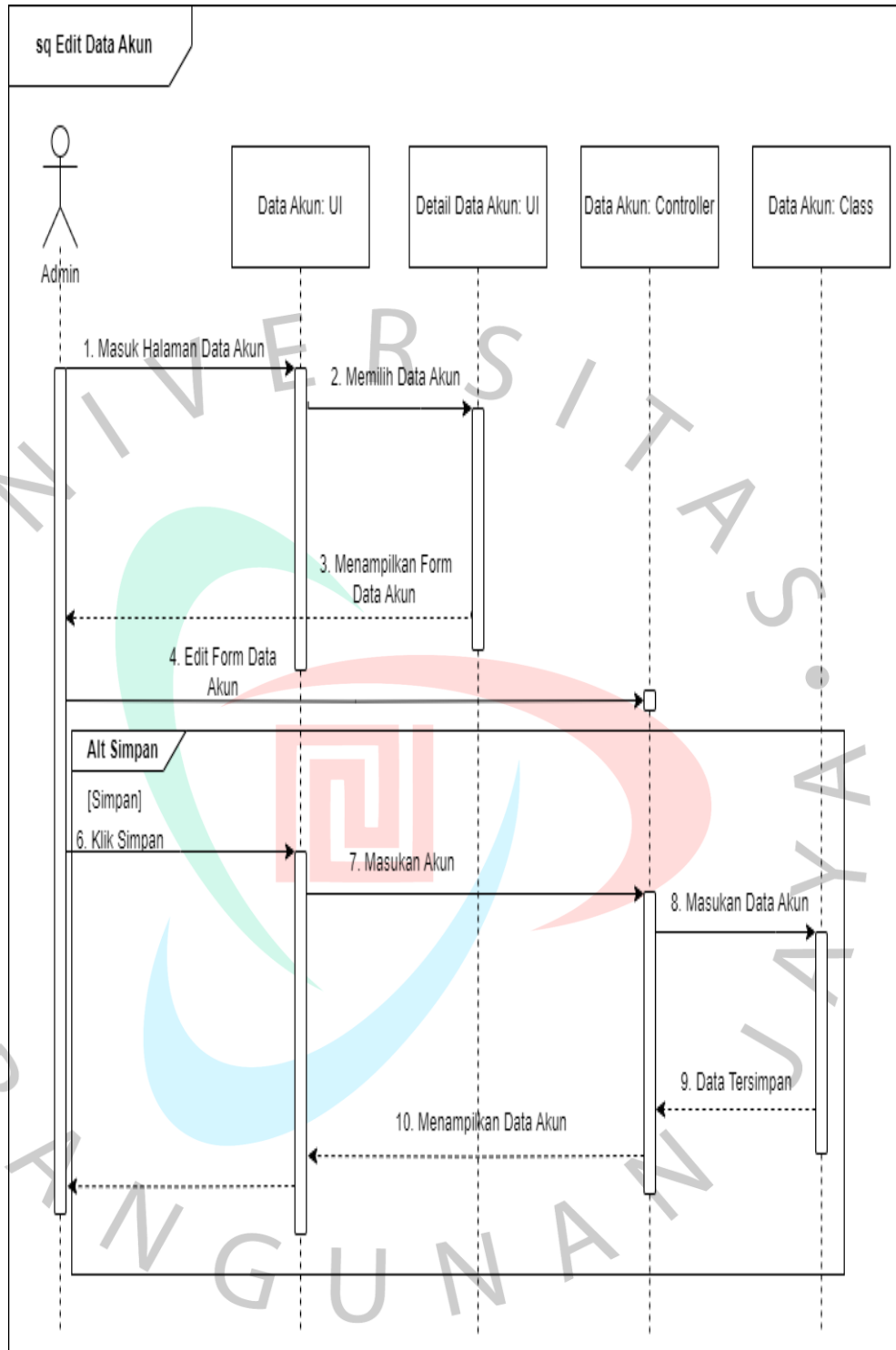
Pada **Gambar 4.18** diatas menjelaskan alur untuk Edit data nomor resi. Pada gambar tersebut terdapat aktor yaitu *user* hasil dari generalisasi kasir dan pemilik yang memiliki akses edit data nomor resi didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.

d) Kelola akun



Gambar 4.18 Kelola Akun

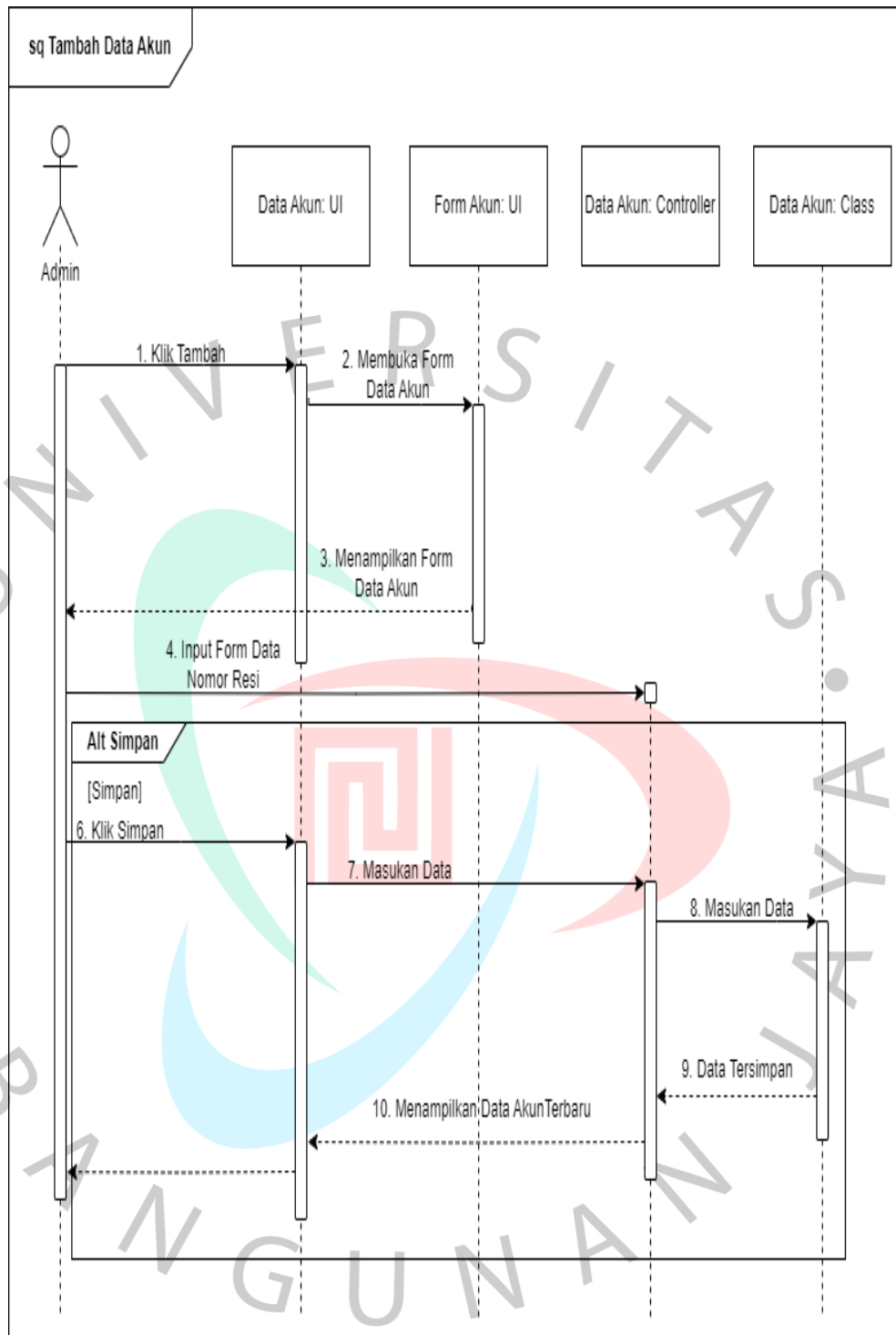
Pada **Gambar 4.19** diatas menjelaskan alur mengelola akun. Pada gambar tersebut terdapat aktor yaitu *Admin* yang memiliki akses tambah data, edit data dan hapus data akun didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.



Gambar 4.19 Edit Data Akun

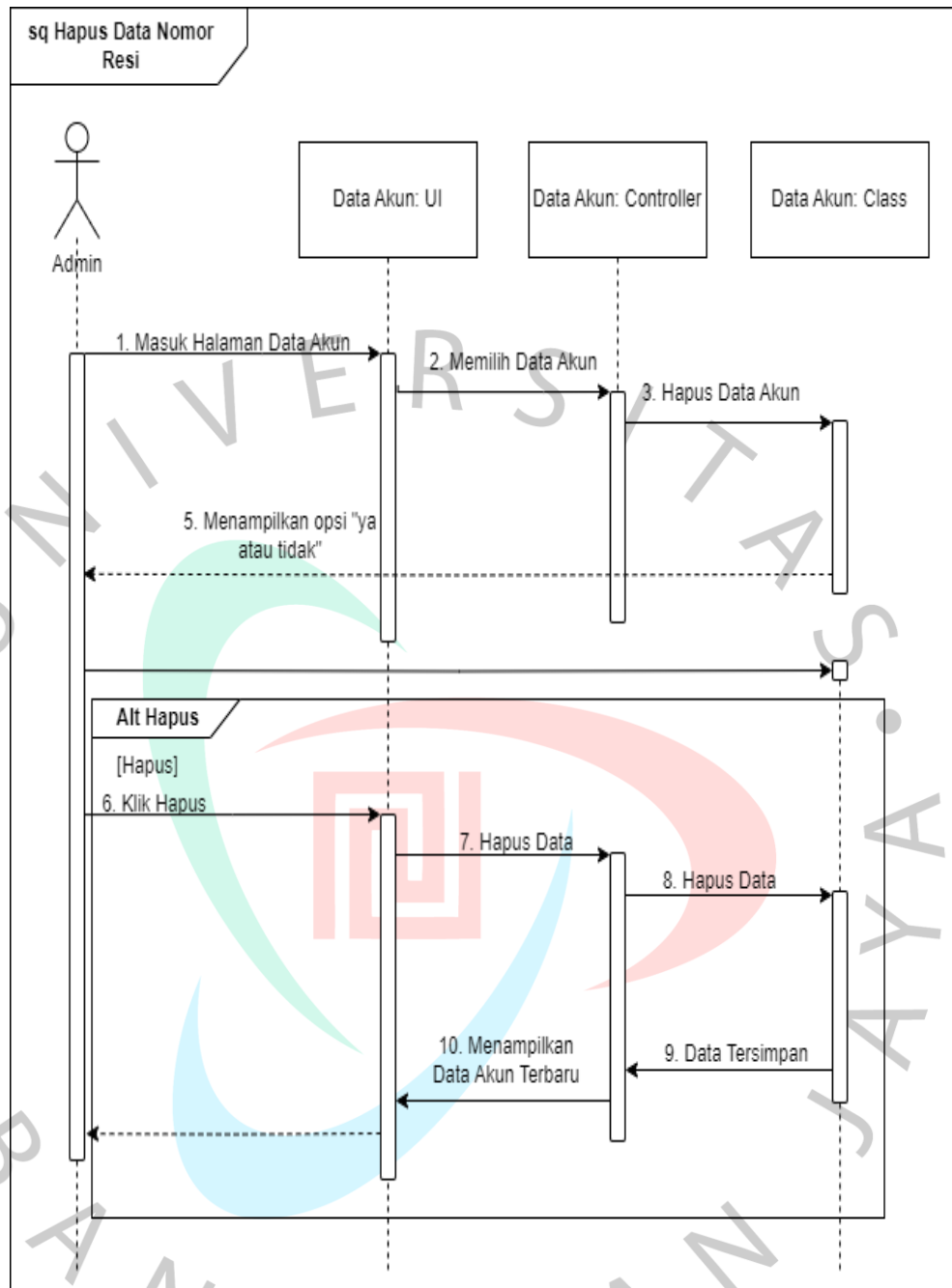
Pada **Gambar 4.20** diatas menjelaskan alur untuk Edit data akun atau pengguna. Pada gambar tersebut terdapat aktor yaitu *Admin* yang memiliki akses edit data akun, didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.





Gambar 4.20 Tambah Data Akun

Pada **Gambar 4.21** diatas menjelaskan alur untuk Tambah data akun. Pada gambar tersebut terdapat aktor yaitu *Admin* yang memiliki akses tambah data akun didalam objek tersebut memiliki *UI*, *Controller* dan *Class*.



Gambar 4.21 Hapus Akun

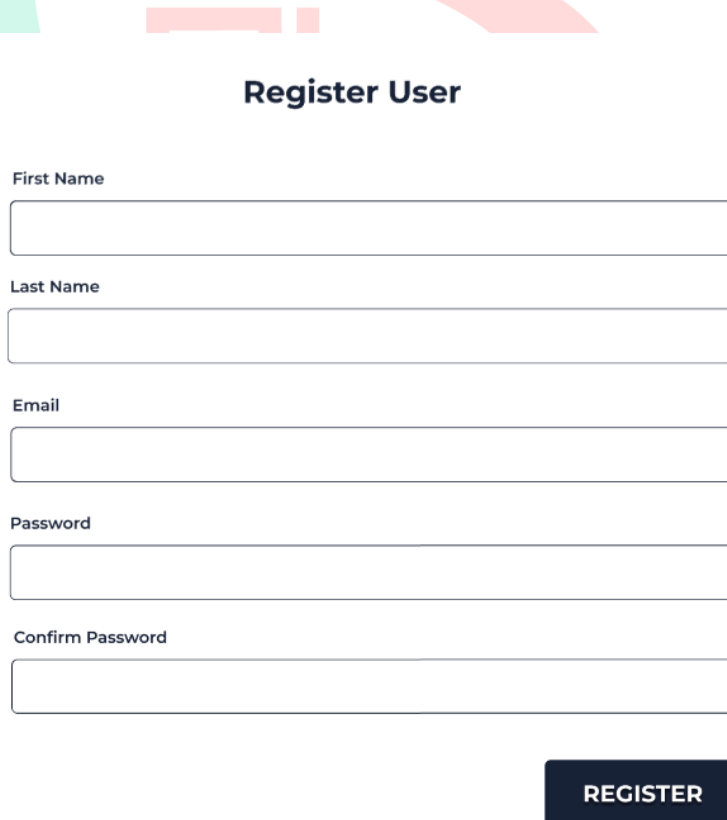
Pada **Gambar 4.22** diatas menjelaskan alur untuk Hapus data akun. Pada gambar tersebut terdapat aktor yaitu *Admin* yang memiliki akses hapus data akun didalam objek, tersebut memiliki *UI*, *Controller* dan *Class*.

#### 4.2.6 Perancangan Antar Muka Pengguna

Perancangan antarmuka (*UI Design*) adalah proses merencanakan dan membuat tampilan serta interaksi pengguna yang efektif, estetis, dan ramah pengguna dalam suatu produk atau aplikasi. Tujuannya adalah memberikan pengalaman pengguna yang baik dan memudahkan pengguna dalam berinteraksi dengan sistem. Perancangan antarmuka melibatkan pemikiran mendalam tentang bagaimana elemen-elemen visual dan fungsional akan diatur dan diimplementasikan untuk mencapai tujuan tertentu

##### 1) UI Register

Pada **Gambar 4.22** dibawah ini merupakan tabel *user* yang belum memiliki akun (*id\_role*). Untuk masuk kehalaman dashboard *User* membutuhkan *Username* dan *Password*.



**Register User**

First Name

Last Name

Email

Password

Confirm Password

**REGISTER**

Copyright (c) 2023 Ecu Digital

Gambar 4.22 Register

## 2) UI Login



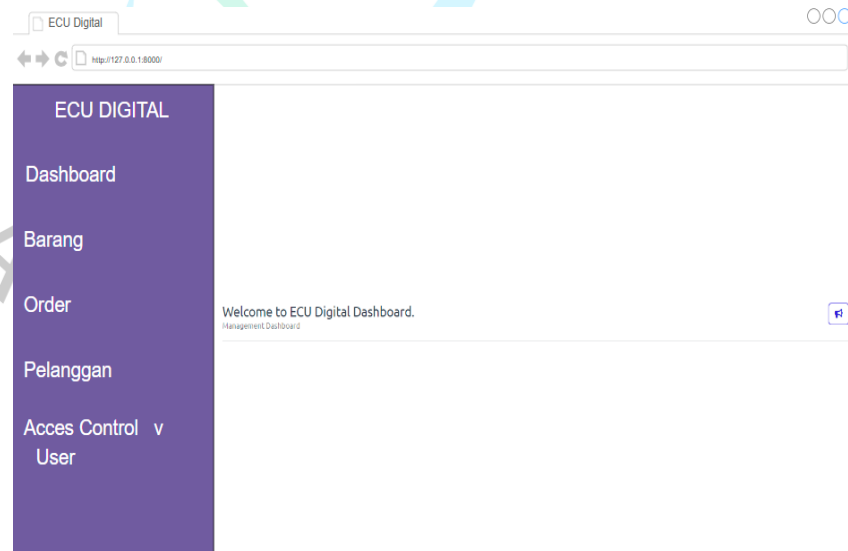
The screenshot shows a login interface with the following elements:

- Log In**: Title of the form.
- Email**: Label for the first input field.
- : Empty text input field for the email address.
- Password**: Label for the second input field.
- : Empty password input field.
- LOG IN**: A dark blue button with white text.
- Copyright (c) 2023 Ecu Digital**: Text at the bottom of the form.

Gambar 4.23 Login

Pada **Gambar 4.23** deskripsi di atas merujuk pada formulir login, di mana jika login berhasil, pengguna akan diarahkan ke menu utama, sementara jika login tidak berhasil, maka kembali ke form login dan masukan kata dengan benar.

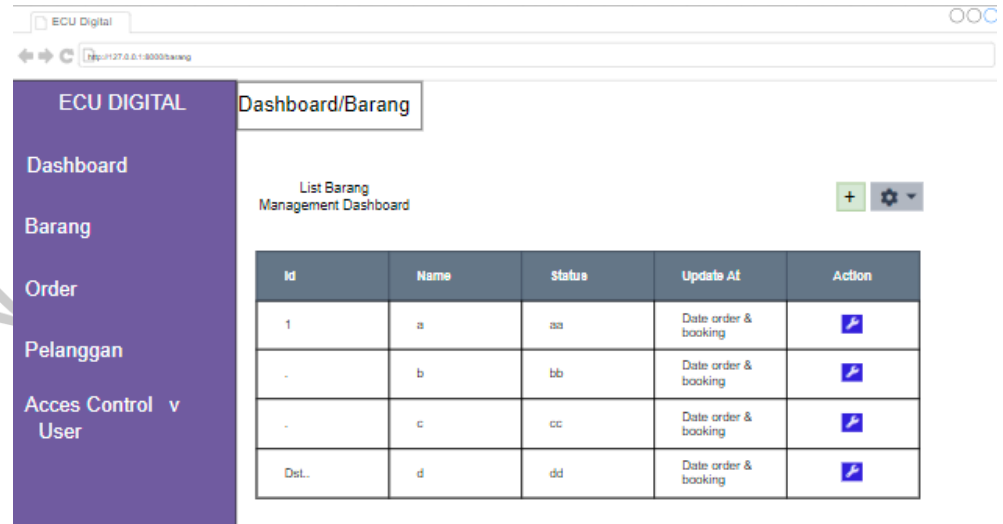
## 3) UI Index



Gambar 4.24 Dashboard

Pada **Gambar 4.24** diatas merupakan gambaran dashboard yang sudah berhasil *login* terdapat menu berupa barang, *order*, pelanggan dan *User Control*.

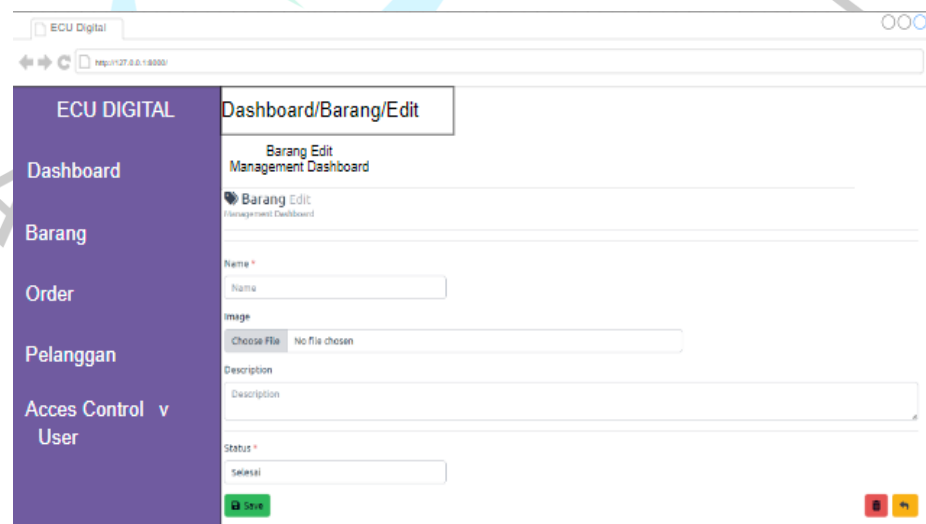
4) UI Dashboard Barang



Gambar 4.25 Dashboard Barang

Pada **Gambar 4.25** diatas adalah tabel mengelola barang yang dapat dikelola oleh kasir memiliki aksi edit dan tambah data.

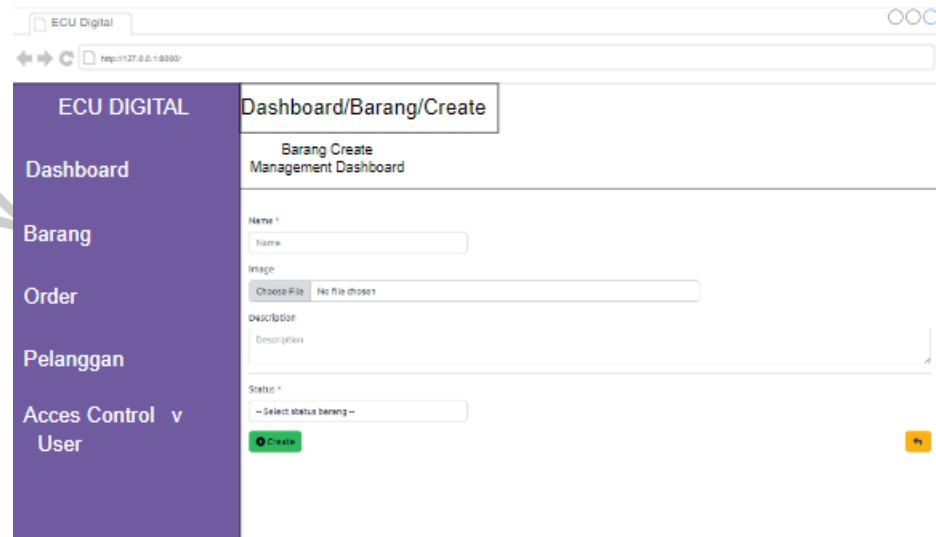
5) UI Dashboard Barang Edit



Gambar 4.26 Edit Barang

Pada **Gambar 4.26** Pernyataan di atas menggambarkan sebuah tabel untuk mengedit data barang dan juga memberikan kemampuan untuk menghapus data barang yang telah dipilih oleh kasir.

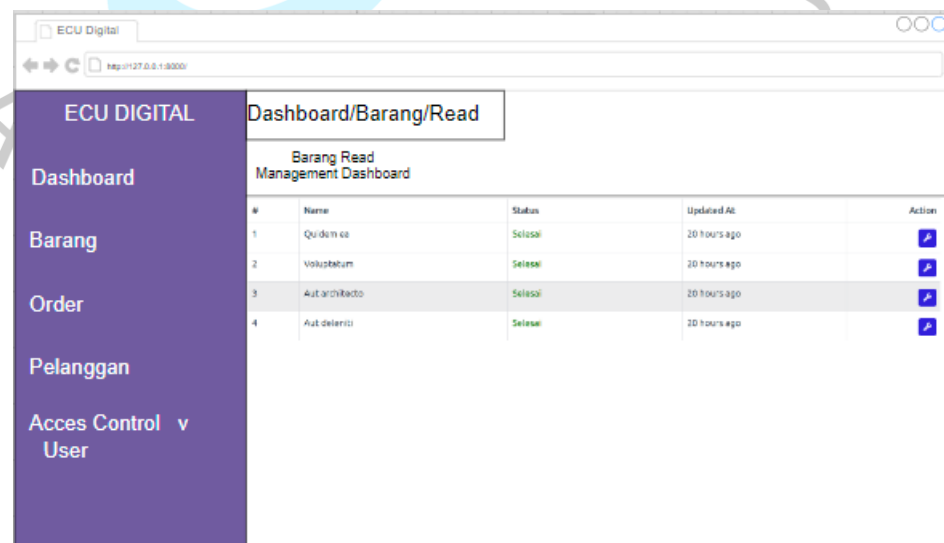
#### 6) UI Dashboard Tambah Barang



Gambar 4.27 Tambah Barang

Pada **Gambar 4.27** diatas ialah tabel tambah barang dan harus mengisi tabel yang sesuai dengan deskripsi.

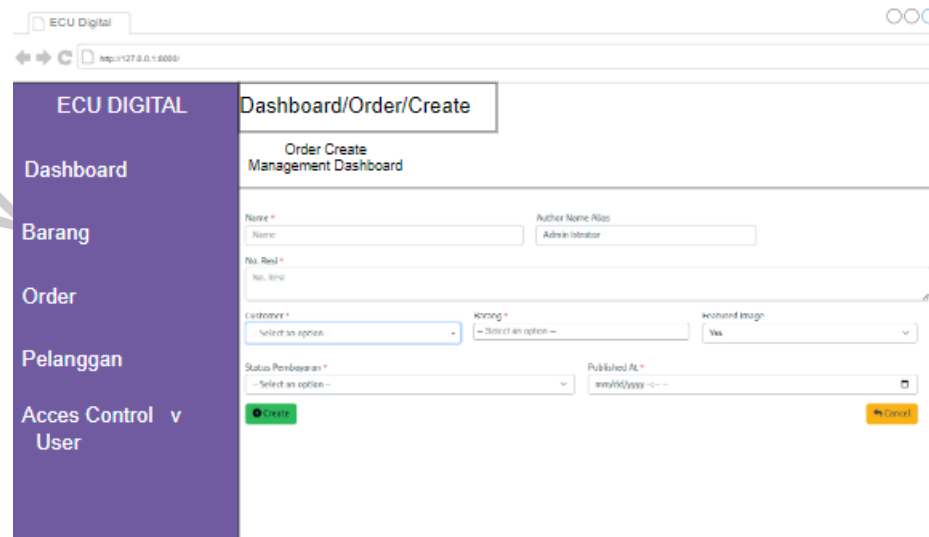
#### 7) UI Dashboard Melihat Barang



Gambar 4.28 Melihat Status Barang

Pada **Gambar 4.28** diatas menggambarkan *form* data barang yang sudah dibuat dan memiliki tanggal otomatis ketika menambah data dan mengedit data. Untuk status sendiri memiliki dua aksi yaitu *progress* dan selesai.

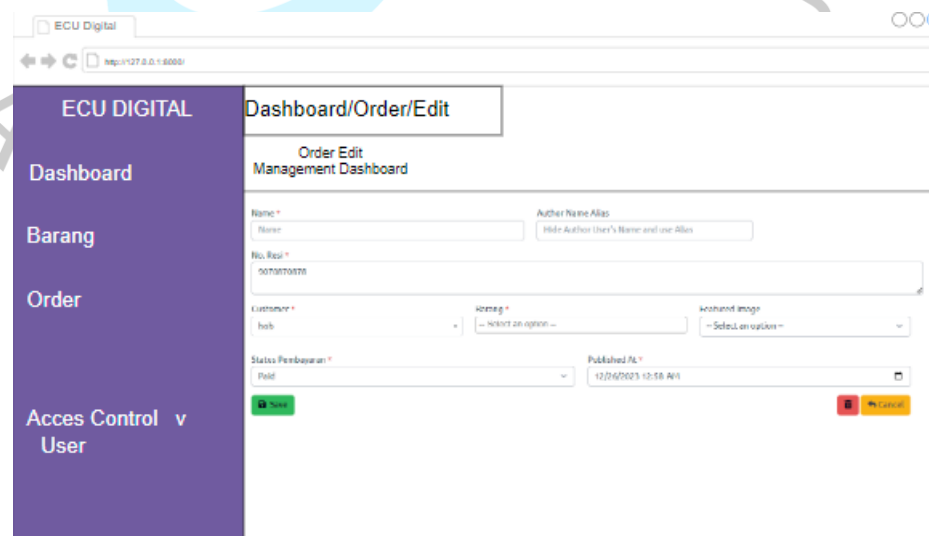
#### 8) UI Dashboard Membuat Pemesanan



Gambar 4.29 Menambah Pesanan

Pada **Gambar 4.29** diatas merupakan *form* tambah barang dan harus mengisi tabel yang sesuai dengan deskripsi.

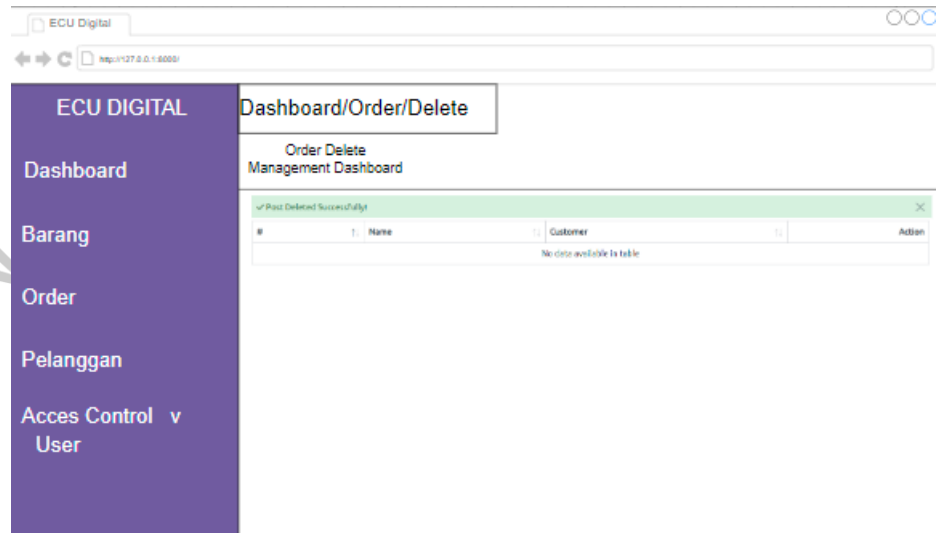
#### 9) UI Dashboard Edit Pemesanan



Gambar 4.30 Mengedit Pesanan

Pada **Gambar 4.30** diatas merupakan tabel edit data pemesanan Dan juga memungkinkan untuk menghapus data barang yang telah dipilih oleh kasir.

#### 10) UI Dashboard Hapus Pemesanan

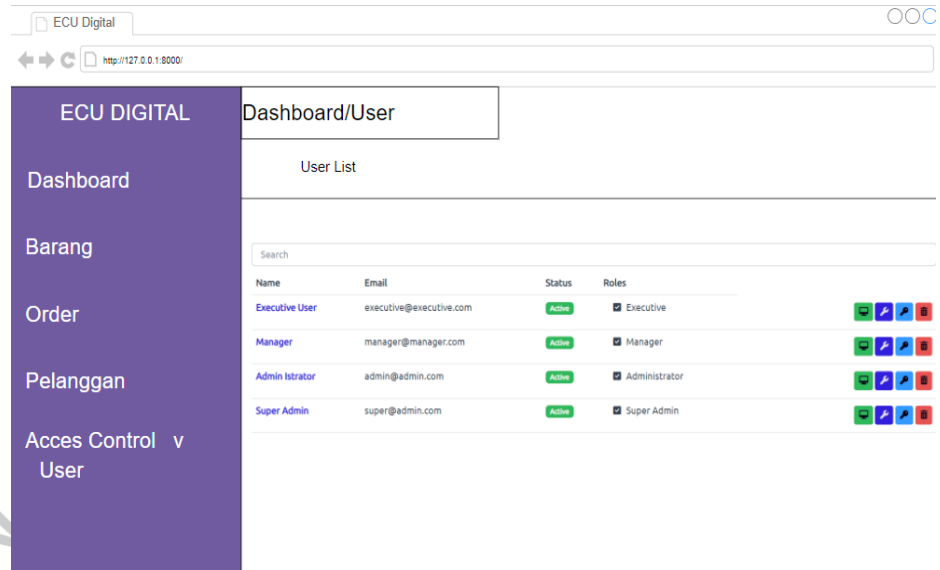


Gambar 4.31 Hapus Pemesanan

Pada **Gambar 4.31** diatas Merupakan tabel pesanan yang telah dipilih, kemudian dihapus, dan akan menampilkan pemberitahuan bahwa data berhasil dihapus..

#### 11) UI Dashboard User

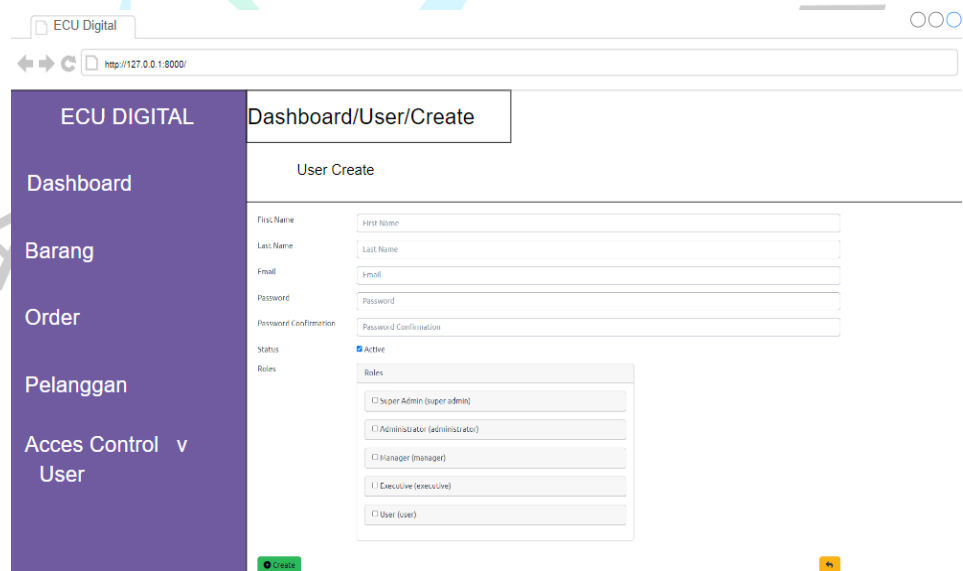




Gambar 4.32 Dashboard Admin

Pada **Gambar 4.32** diatas adalah halaman *Admin* didalamnya terdapat akun-akun *User* yang memiliki *role* atau hak akses yang berbeda. Untuk hak akses masing-masing akan dibuat oleh *Admin* didalam tabel tersebut memiliki tambah akun, edit akun dan hapus akun.

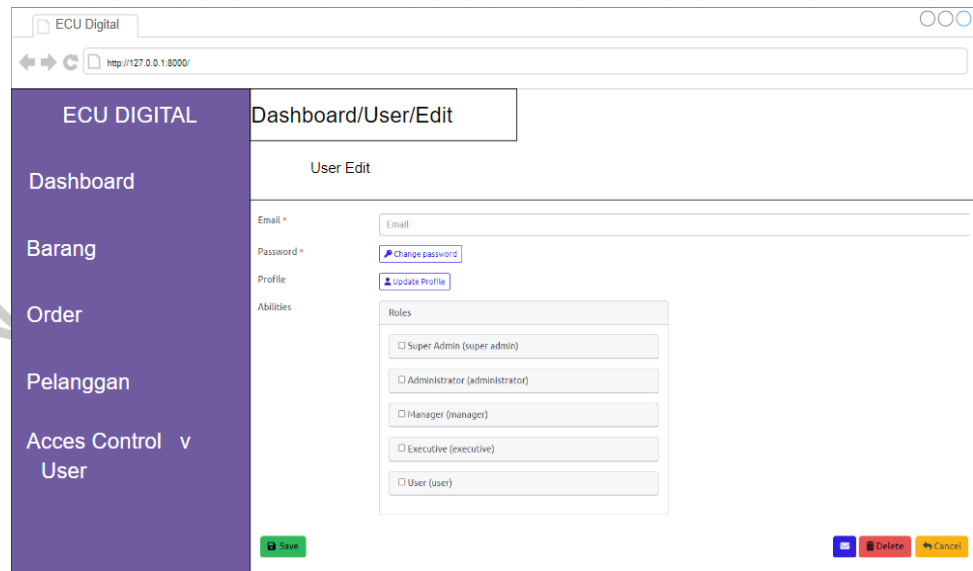
## 12) UI Dashboard Membuat User



Gambar 4.33 Tambah User

Pada **Gambar 4.33** diatas merupakan *form* tambah akun dan *Admin* harus mengisi tabel sesuai dengan deskripsi.

### 13) UI Dashboard Edit User



Gambar 4.34 Edit User

Pada **Gambar 4.34** deskripsi diatas tersebut merujuk pada tabel pengeditan pengguna yang memiliki fasilitas untuk mengubah kata sandi, foto identitas, dan hak akses..

## 4.6 Testing

Untuk tahap terakhir yang dilakukan, yaitu perancangan implementasi, Dilakukan juga pengujian untuk memverifikasi bahwa Aplikasi yang dibuat berfungsi dengan baik dan sesuai dengan yang diharapkan. Dalam proses pengujian ini, aplikasi dianalisis untuk mendeteksi potensi masalah seperti *bug* atau *error* dengan lebih efektif. Melalui tahap pengujian ini diharapkan bahwa masalah pada aplikasi dapat teridentifikasi dan diperbaiki dengan lebih mudah. Implementasi dari tahap ini diharapkan memberikan keyakinan bahwa setelah aplikasi digunakan oleh pengguna, dalam hal ini. Berikut merupakan Tabel 4.12 dibawah ini adalah rancangan *Test Case* yang telah disusun dan terdokumentasi dalam bentuk tabel untuk memandu proses pengujian.

Tabel 4.11 Perancangan Implementasi

No	Fitur	Skenario	Hasil
1.	Login	<ol style="list-style-type: none"> <li>1. Mengisi form login dengan menggunakan username dan password yang sesuai berdasarkan hak akses</li> <li>2. Masuk menu Halaman Utama</li> </ol>	Berhasil
3.	Mengelola Tambah Data Barang	<ol style="list-style-type: none"> <li>1. Masuk menu Data Barang</li> <li>2. Menampilkan Data Barang</li> <li>3. Klik Tombol Tambah</li> <li>4. Isi form</li> <li>5. Klik Tombol Simpan</li> <li>6. Data Disimpan</li> </ol>	Berhasil
7.	Mengelola Edit Data Barang	<ol style="list-style-type: none"> <li>1. Masuk menu Data Barang</li> <li>2. Menampilkan Data Barang</li> <li>3. Pilih data yang ingin dilakukan edit</li> <li>4. Isi form sesuai apa yang ingin di ubah</li> <li>5. Klik Tombol Simpan</li> <li>6. Data Disimpan</li> </ol>	Berhasil
8.	Mengelola Hapus Data Barang	<ol style="list-style-type: none"> <li>1. Masuk ke menu Data Barang</li> <li>2. Menampilkan Data Barang</li> <li>3. Pilih data yang akan dihapus</li> <li>4. Klik Button Hapus</li> <li>5. Data Disimpan</li> </ol>	Berhasil

6.	Mengelola Tambah Data Transaksi	<ol style="list-style-type: none"> <li>1. Masuk menu Data Transaksi</li> <li>2. Menampilkan Data Transaksi</li> <li>3. Klik Tombol Tambah</li> <li>4. Isi form</li> <li>5. Klik Tombol Simpan</li> <li>6. Data Disimpan</li> </ol>	Berhasil
7.	Mengelola Edit Data Transaksi	<ol style="list-style-type: none"> <li>1. Masuk menu Data Transaksi</li> <li>2. Menampilkan Data Transaksi</li> <li>3. Pilih Data Transaksi yang ingin dilakukan Edit</li> <li>4. Isi form sesuai apa yang ingin di ubah</li> <li>5. Klik Tombol Simpan</li> <li>6. Data Disimpan</li> </ol>	Berhasil
8.	Mengelola Hapus Data Transaksi	<ol style="list-style-type: none"> <li>1. Masuk menu Data Transaksi</li> <li>2. Menampilkan Data Transaksi</li> <li>3. Pilih data yang ingin dihapus</li> <li>4. Klik Tombol Hapus</li> <li>5. Data Disimpan</li> </ol>	Berhasil
6.	Mengelola Tambah Resi Pengiriman	<ol style="list-style-type: none"> <li>1. Masuk menu Data Resi Pengiriman</li> <li>2. Menampilkan Data Resi Pengiriman</li> <li>3. Klik Tombol Tambah</li> <li>4. Isi form</li> <li>5. Klik Tombol Simpan</li> <li>6. Data Disimpan</li> </ol>	Berhasil

7.	Mengelola Edit Resi Pengiriman	<ol style="list-style-type: none"> <li>1. Masuk menu Data Transaksi</li> <li>2. Menampilkan Data Transaksi</li> <li>3. Pilih Data Transaksi yang ingin dilakukan Edit</li> <li>4. Isi form sesuai apa yang ingin di ubah</li> <li>5. Klik Tombol Simpan</li> <li>6. Data Disimpan</li> </ol>	Berhasil
8.	Mengelola Hapus Resi Pengiriman	<ol style="list-style-type: none"> <li>1. Masuk menu Data Transaksi</li> <li>2. Menampilkan Data Transaksi</li> <li>3. Pilih data yang ingin dihapus</li> <li>4. Klik Tombol Hapus</li> <li>5. Data Disimpan</li> </ol>	Berhasil
9.	Mengelola Akun	<ol style="list-style-type: none"> <li>1. Masuk menu Data Akun</li> <li>2. Menampilkan Data Akun</li> <li>3. Admin Dapat menambah data akun</li> <li>4. Mengedit Akun</li> <li>5. Menghapus Akun</li> <li>6. Mengupdate Akun</li> </ol>	Berhasil
10.	Mengelola Tambah Akun	<ol style="list-style-type: none"> <li>1. Masuk menu Data Akun</li> <li>2. Menampilkan Data Akun</li> <li>3. Klik Tombol Tambah Akun</li> <li>4. Isi form Data Sesuai Deskripsi</li> <li>5. Klik Tombol Simpan</li> </ol>	Berhasil

		6. Data Disimpan	
11.	Mengelola Edit Akun	<ol style="list-style-type: none"> <li>1. Masuk menu Data Akun</li> <li>2. Menampilkan Data Akun</li> <li>3. Pilih Data Akun</li> <li>4. Klik tombol Edit dari hasil yang sudah dipilih</li> <li>5. Isi form Data</li> <li>6. Klik Tombol Simpan</li> <li>7. Data Disimpan</li> </ol>	Berhasil
12.	Mengelola Hapus Akun	<ol style="list-style-type: none"> <li>1. Masuk menu Data Akun</li> <li>2. Menampilkan Data Akun</li> <li>3. Pilih Data Akun</li> <li>4. Klik Tombol Hapus</li> <li>5. Data berhasil Disimpan</li> <li>6. Update Data Terbaru</li> </ol>	Berhasil

Perusahaan ECU DIGITAL menghadapi kendala dalam mencatat data barang, transaksi, dan pembukuan akibat *human error* atau kelalaian karyawan. Maka dari itu mencoba untuk menangani masalah tersebut, disarankan melakukan pengembangan sistem informasi pembukuan berbasis web. Langkah-langkah pembukuan melibatkan pencatatan transaksi, laporan keuangan, dan pencatatan nomor resi akan diintegrasikan dalam aplikasi pemesanan perbaikan ECU DIGITAL. Tujuannya adalah mempermudah para pekerja dalam melakukan rekap data dan memberikan solusi terhadap kendala yang ada. Implementasi sistem aplikasi pemesanan perbaikan diharapkan dapat meningkatkan kinerja perusahaan, mengurangi kesalahan, dan menyediakan dasar informasi yang akurat.