

## **BAB III**

### **PEMBAHASAN**

#### **3.1. Tinjauan Perusahaan**

Semakin tingginya persaingan dunia bisnis khususnya *petshop* dalam bidang perdagangan hewan peliharaan dan perlengkapan hewan membuat banyak usaha-usaha harus melakukan kreatifitas dan inovasi dari segi produk dan pelayanan. Kegiatan yang menjadikan *customer* adalah prioritas utama, akan membuat *customer* merasa senang sehingga dapat menjadi pelanggan tetap dalam transaksi perdagangan.

Pluit Village Petshop merupakan suatu usaha yang bergerak dibidang penjualan hewan dan perlengkapan hewan peliharaan. Dalam melaksanakan kegiatan usahanya Pluit Village Petshop masih belum terkomputerisasi, salah satunya pada pembuatan laporan penjualan hewan dan barang, serta jasa pemandian hewan. Keuntungan yang didapatkan dengan adanya sistem yang telah terkomputerisasi adalah meningkatkan efisiensi kerja. Sehingga dengan menggunakan sistem yang sudah terkomputerisasi dapat memperkecil kesalahan yang akan terjadi.

##### **3.1.1. Sejarah Perusahaan**

Berdiri pada tahun 2014, didirikan oleh Budijanto. Beliau adalah pemilik Pluit Village Petshop. Petshop yang beralamatkan di Jl. Pluit Indah Raya, tepatnya di Mall Pluit Village, Jakarta Utara 14440. Berawal dari *hobby* memelihara anjing akhirnya menjadi jual beli hewan. Hingga akhirnya beliau memutuskan untuk membuka *petshop*. Toko ini merupakan *pet center* atas penjualan hewan-hewan

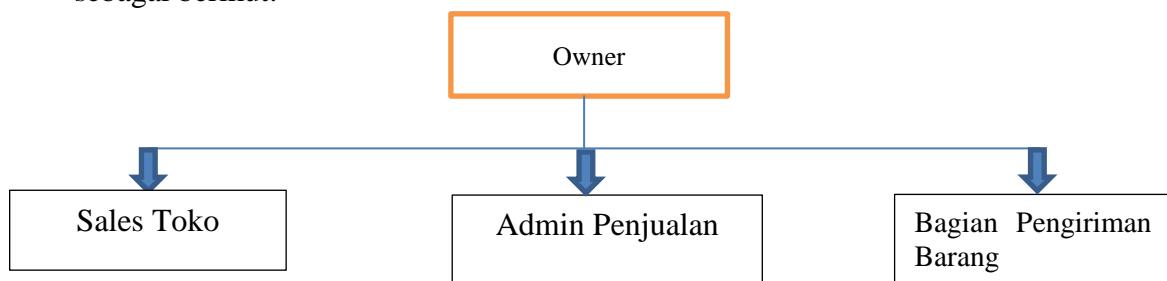
peliharaan dan perlengkapan hewan peliharaan. Sejalan dengan kebutuhan konsumen yang semakin meningkat, toko ini selanjutnya mengembangkan pameran atau event berupa playground, yaitu area bermain dengan beberapa hewan peliharaan yang bersahabat. Pameran yang diberi nama Happy Bunny ini sangat berkembang pesat karna setiap 1 bulan sekali pameran ini selalu pindah dari Mall ke Mall sekaligus mempromosikan Pluit Village Petshop. Sehingga Pluit Village Petshop begitu menerima *impact* yang baik dari pameran tersebut.

### **3.1.2. Struktur Organisasi dan Fungsi**

Struktur organisasi yang terdapat di dalam Pluit Village Petshop adalah susunan dari bagian-bagian pengelola yang didasarkan pada perbedaan kewenangan dan tanggung jawab serta dilengkapi dengan penjelasan tentang hubungan antar bagian tersebut. Struktur organisasi merupakan saran yang sangat penting untuk menjalankan fungsinya.

Adapun struktur organisasi ini sendiri dapat dikatakan sebagai suatu kerangka yang mewujudkan suatu pola tetapi dari hubungan antara kedudukan dan peranan dalam suatu kerjasama.

Bentuk struktur organisasi yang dimiliki Pluit Village Petshop adalah sebagai berikut:



**Gambar III.1. Stuktur Organisasi**

Fungsi dari tiap-tiap bagian sebagai berikut:

1. Owner
  - a. Memimpin jalannya toko
  - b. Mengontrol semua karyawan
  - b. Memeriksa laporan toko
  - c. Mengurus penggajian
2. Bagian Penjualan (Admin)
  - a. Menerima order dari *customer*
  - b. Membuat dokumen pemesanan
  - c. Membuat dokumen penjualan
  - d. Menerima pembayaran penjualan
  - e. Membuat dokumen surat jalan
  - f. Membuat laporan penjualan
3. Bagian Sales
  - a. Melakukan pelayanan langsung costumer tentang barang
  - b. Melakukan penyetoran barang yang telah di jual
  - c. Menata barang yang akan di jual
4. Bagian Pengiriman
  - a. Menyediakan barang pesanan
  - b. Melakukan pengiriman barang pesanan *customer*
  - c. Melaporkan dokumen pengiriman

### 3.2. Tinjauan Kasus

Dalam penulisan Tugas Akhir ini, penulis hanya membahas tentang Rancang Bangun Sistem Informasi Akuntansi Penjualan pada Pluit Village Petshop. Batasan dari penulisan penjualan ini dimulai dari prosedur penawaran, pemesanan, pembayaran, pengiriman,pencatatan.

#### 3.2.1 Proses Bisnis Sistem Berjalan

##### 1. Prosedur Penawaran Barang

Surat penawaran yang diajukan oleh bagian admin penjualan kepada *customer*. Apabila *customer* yang berminat melakukan pembelian akan menelpon admin penjualan untuk menanyakan tersedia atau tidaknya barang yang akan dibeli, sedangkan apabila *customer* tidak tertarik akan mengkonfirmasi melalui telpon.

##### 2. Prosedur Pemesanan Barang

Apabila barang yang dipesan tersedia, maka *customer* akan datang langsung untuk memesan barang, dan memberikan informasi barang pesanan berupa PO (*purchase order*) kepada bagian admin penjualan.

##### 3. Prosedur Pembayaran

Bagian admin penjualan langsung membuatkan invoice sesuai dengan PO yang diterima. Invoice berwarna putih diserahkan kepada *customer* sedangkan *copy* invoice disimpan sebagai arsip data pembayaran. Kemudian setelah *customer* melakukan pembayaran secara tunai kepada admin penjualan, maka admin penjualan membuatkan surat jalan dan menyerahkan surat jalan tersebut kepada bagian pengiriman.

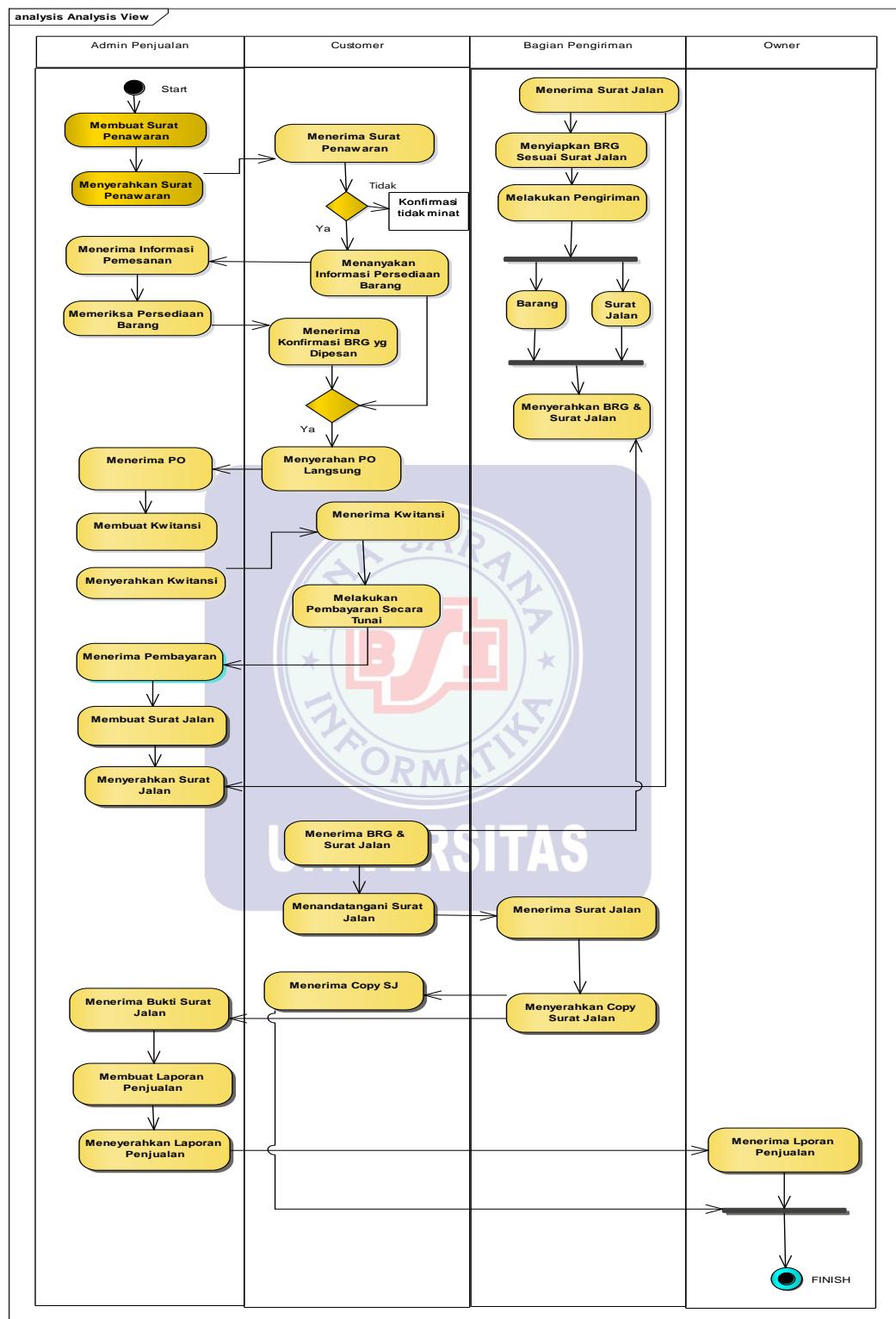
#### 4. Bagian Pengiriman

Bagian pengiriman menyiapkan barang sesuai dengan isi surat jalan dan mengirimkan barang pesanan tersebut kepada *customer*. Setelah barang pesanan sampai kepada *customer*, bagian pengiriman kemudian meminta tandatangan *customer* pada surat jalan sebagai bukti bahwa barang telah diterima langsung oleh *customer*. Bagian pengiriman menyerahkan *copy* surat jalan kepada *customer*. Surat jalan yang asli dibawa kembali oleh bagian pengiriman sebagai bukti bahwa pengiriman telah selesai dilakukan, kemudian bagian pengiriman menyerahkan surat jalan asli ke bagian admin penjualan.

#### 5. Prosedur Laporan Penjualan

Setelah menerima bukti surat jalan asli yang diserahkan oleh bagian pengiriman, admin penjualan membuatkan laporan penjualan dan menyerahkan laporan penjualan kepada owner.

### 3.2.2. Activity Diagram



Gambar III.2. Activity Diagram

### 3.2.3. Dokumen Masukan

1. Nama Dokumen : *Purchase Order*

Fungsi : Sebagai bukti transaksi pemesanan barang  
 Sumber : *Customer*  
 Tujuan : Admin penjualan  
 Media : Kertas  
 Frekuensi : Setiap terjadi transaksi pemesanan  
 Format : Lampiran A.1

### 3.2.4. Dokumen Keluaran

1. Nama Dokumen : *Invoice*

Fungsi : Sebagai bukti transaksi penjualan barang toko  
 Sumber : Admin penjualan  
 Tujuan : *Customer*  
 Media : Kertas  
 Frekuensi : Setiap terjadi transaksi penjualan  
 Format : Lampiran B.1

2. Nama Dokumen : Surat jalan

Fungsi : Sebagai bukti transaksi pengiriman barang dan dokumen  
 Sumber : Admin penjualan  
 Tujuan : Bagian Pengiriman dan Customer  
 Media : Kertas  
 Frekuensi : Setiap terjadi transaksi pengiriman  
 Format : Lampiran B.2

### 3.2.5. Permasalahan Pokok

Permasalahan pokok dalam toko ini adalah aktivitas penjualannya yang masih manual, hal ini mengakibatkan segala sistem informasi penjualan sering terjadi kesalahan terutama dalam aktivitas kerja seperti pencatatan data, pengecekan data dan laporan-laporan, yang kadangkala menimbulkan selisih pendapatan dan menyebabkan proses kerja menjadi tidak praktis dan *efisien*.

Permasalahan lain dalam toko ini yaitu penyimpanan dokumen masih manual sehingga sering kali menyusahkan saat akan mencari dan mengambil suatu dokumen yang diperlukan.

### 3.2.6. Pemecahan Masalah

Setelah melihat permasalahan yang dihadapi pada Pluit Village Petshop, maka dari permasalahan yang terjadi perlu adanya pemecahan masalah. Pemecahan-pemecahan masalah tersebut di antaranya:

1. Untuk menjaga keamanan data sebaiknya digunakan *back-up* data sebagai cadangan apabila terjadi masalah pada penyimpanan data. Agar data-data tersimpan secara teratur sehingga dokumen tidak hilang.
2. Pembuatan laporan penjualan harus menggunakan sistem yang terkomputerisasi yang terkoneksi pada komputer sehingga laporan penjualan bisa dikerjakan lebih cepat dan akurat serta meminimalisir kesalahan yang terjadi.

Semoga dengan cara pemecahan masalah ini bisa membantu permasalahan-permasalahan yang ada dalam sistem penjualan pada Pluit Village Petshop. Untuk itu, dibutuhkan sistem penjualan yang dapat memproses penjualan secara elektronik sehingga data penjualan dapat disimpan dengan baik dan lebih efisien.

### **3.3. Analisis Kebutuhan *Software***

Analisa kebutuhan perangkat lunak (*software*) adalah kondisi, kriteria, syarat atau kemampuan yang harus dimiliki oleh perangkat lunak untuk memenuhi apa yang disyaratkan atau diinginkan pemakai.

#### **3.3.1. Analisis Kebutuhan**

Halaman Admin

- A.1 Login Admin
- A.2 Admin dapat mengelola menu utama
- A.3 Admin dapat mengelola data *office*
- A.4 Admin dapat mengelola data *customer*
- A.5 Admin dapat mengelola data *items*
- A.6 Admin dapat mengolah pemesanan
- A.7 Admin dapat menginput transaksi pengiriman
- A.8 Admin dapat menginput transaksi penjualan
- A.9 Admin dapat mencetak laporan *sales order*
- A.10 Admin dapat mencetak laporan surat jalan
- A.11 Admin dapat mencetak laporan jurnal
- A.12 Admin dapat logout

Halaman Owner

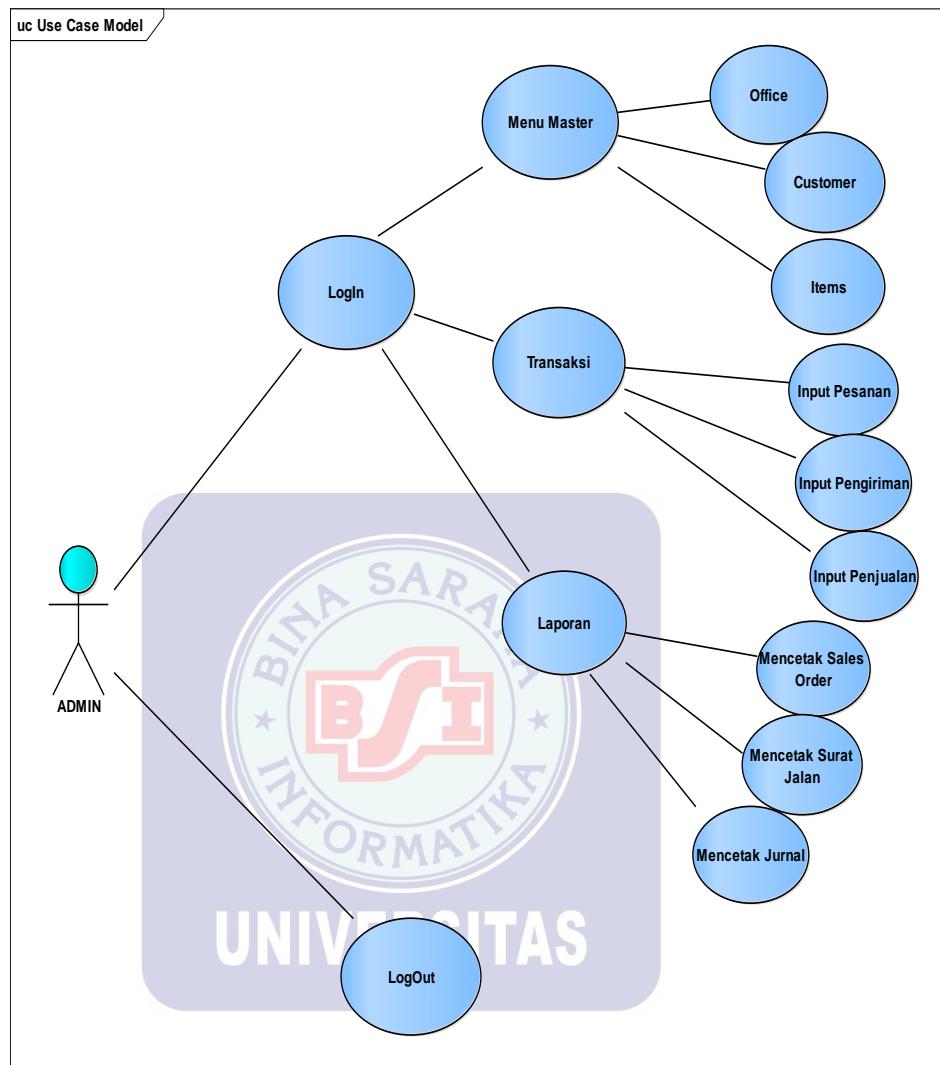
- B.1 Login Owner
- B.2 Owner dapat mengelola menu utama
- B.3 Owner dapat mengelola *office*

- B.4 Owner dapat mencetak data *office* dan *customer*
- B.5 Owner dapat mencetak laporan barang
- B.6 Owner dapat mencetak laporan pengiriman barang
- B.7 Owner dapat mencetak jurnal
- B.8 Owner dapat Logout



### 3.3.2. Use Case Diagram

#### 1. Use Case Diagram Halaman Admin

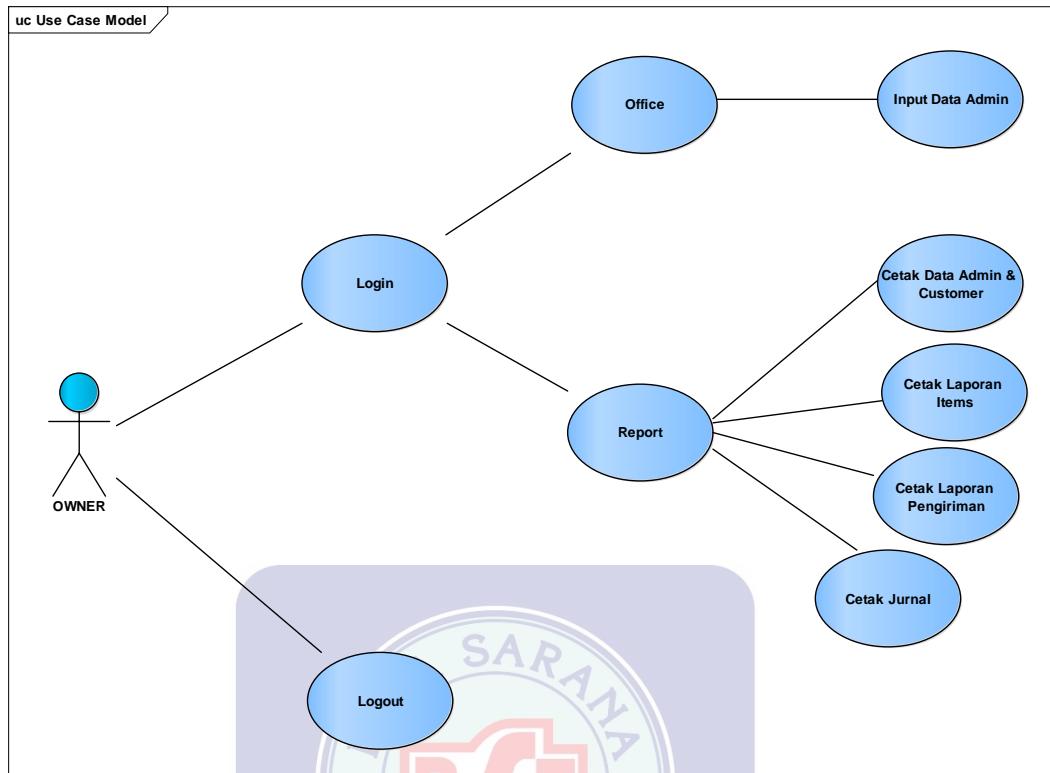


Gambar III.3. Use Case Diagram Admin

**Tabel III.1. Use Case Diagram Admin****Deskripsi Gambar Menu Utama**

<b>Use Case Narative Menu Utama</b>	
Tujuan	Admin dapat melakukan pengolahan data yang ada di menu master, menu transaksi, dan menu cetak
Deskripsi	Sistem ini memungkinkan aktor untuk mengelola sistem mulai dari melihat data, input data sampai dengan mencetak bukti transaksi
<b>Skenario Utama</b>	
Aktor	Admin
Kondisi Awal	Aktor Membuka Aplikasi Sistem dan Masuk ke Menu Utama
Aksi Aktor	Reaksi Sistem
1. Aktor Memilih tombol Menu Master	Sistem akan menampilkan submenu data pengguna, data pelanggan, data akun dan data barang
2. Aktor Memilih tombol Menu Transaksi	Sistem akan menampilkan submenu input pemesanan, pengiriman, penjualan
3. Aktor Memilih tombol Menu Laporan	Sistem akan menampilkan submenu cetak bukti <i>sales order</i> , surat jalan dan jurnal.
Kondisi Akhir	Jika perintah sesuai maka sistem akan menampilkan seperti yang diinginkan oleh si Aktor

## 2. Use Case Diagram Halaman Owner



Gambar III.4. Use Case Diagram Owner

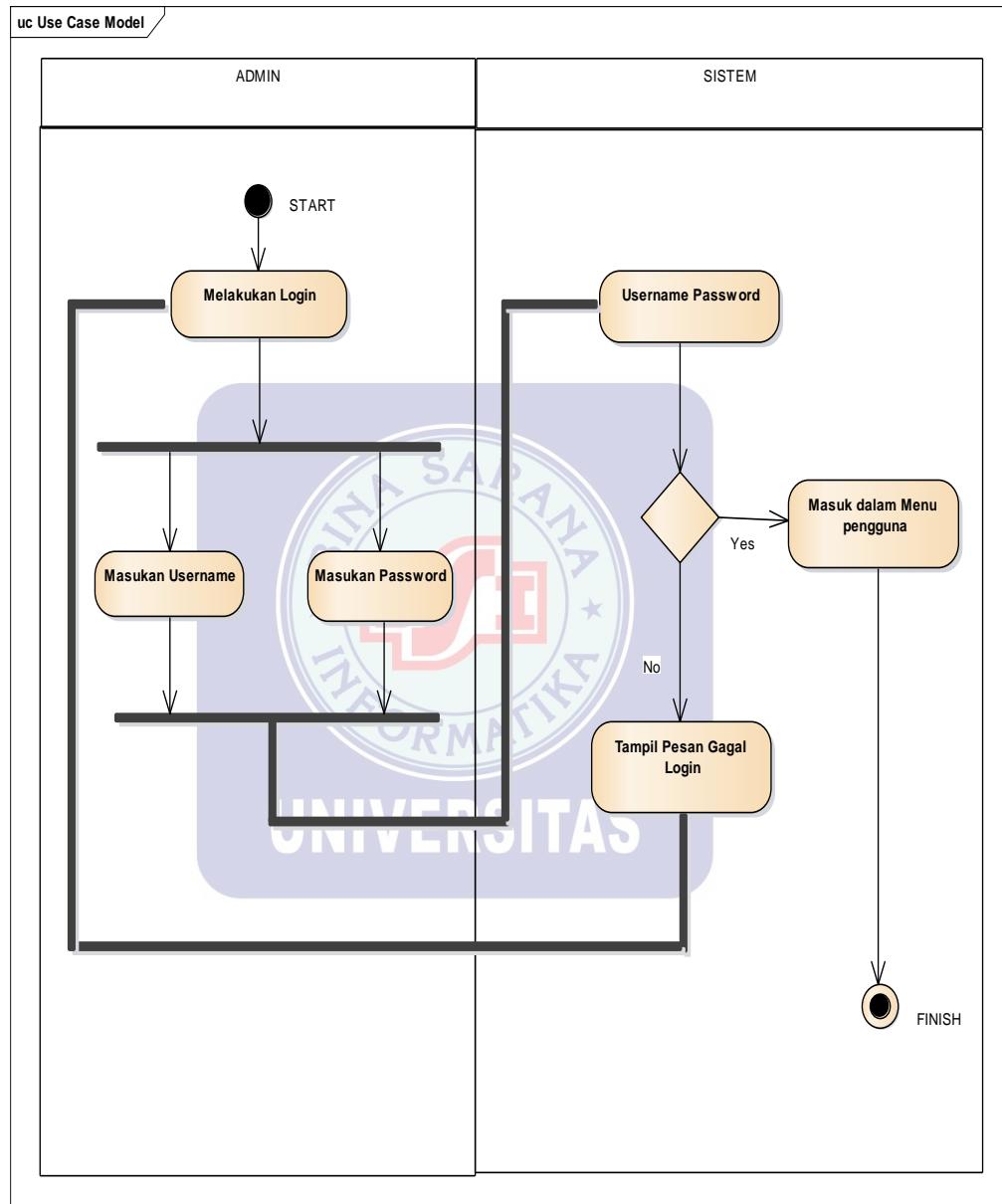
**Tabel III.2. Use Case Diagram Owner****Deskripsi Gambar Menu Utama**

<b>Use Case Narative Menu Utama</b>	
Tujuan	Owner dapat mengakses laporan yang ada di menu laporan penjualan dan laporan pengiriman
Deskripsi	Sistem ini memungkinkan aktor untuk mengakses sistem penjualan mulai dari laporan penjualan hingga laporan pengiriman
<b>Skenario Utama</b>	
Aktor	Owner
Kondisi Awal	Aktor Membuka Menu Utama
Aksi Aktor	<b>Reaksi Sistem</b>
1. Aktor memilih tombol Menu <i>Office</i> 2. Aktor memilih tombol menu <i>Report</i>	<p>Sistem akan menampilkan submenu input admin dan hapus admin.</p> <p>Sistem akan menampilkan cetakan data Admin &amp; Customer, cetakan laporan barang, laporan pengiriman, dan jurnal.</p>
Kondisi Akhir	Jika perintah sesuai maka sistem akan menampilkan seperti yang diinginkan oleh si aktor

### 3.3.3. Activity Diagram

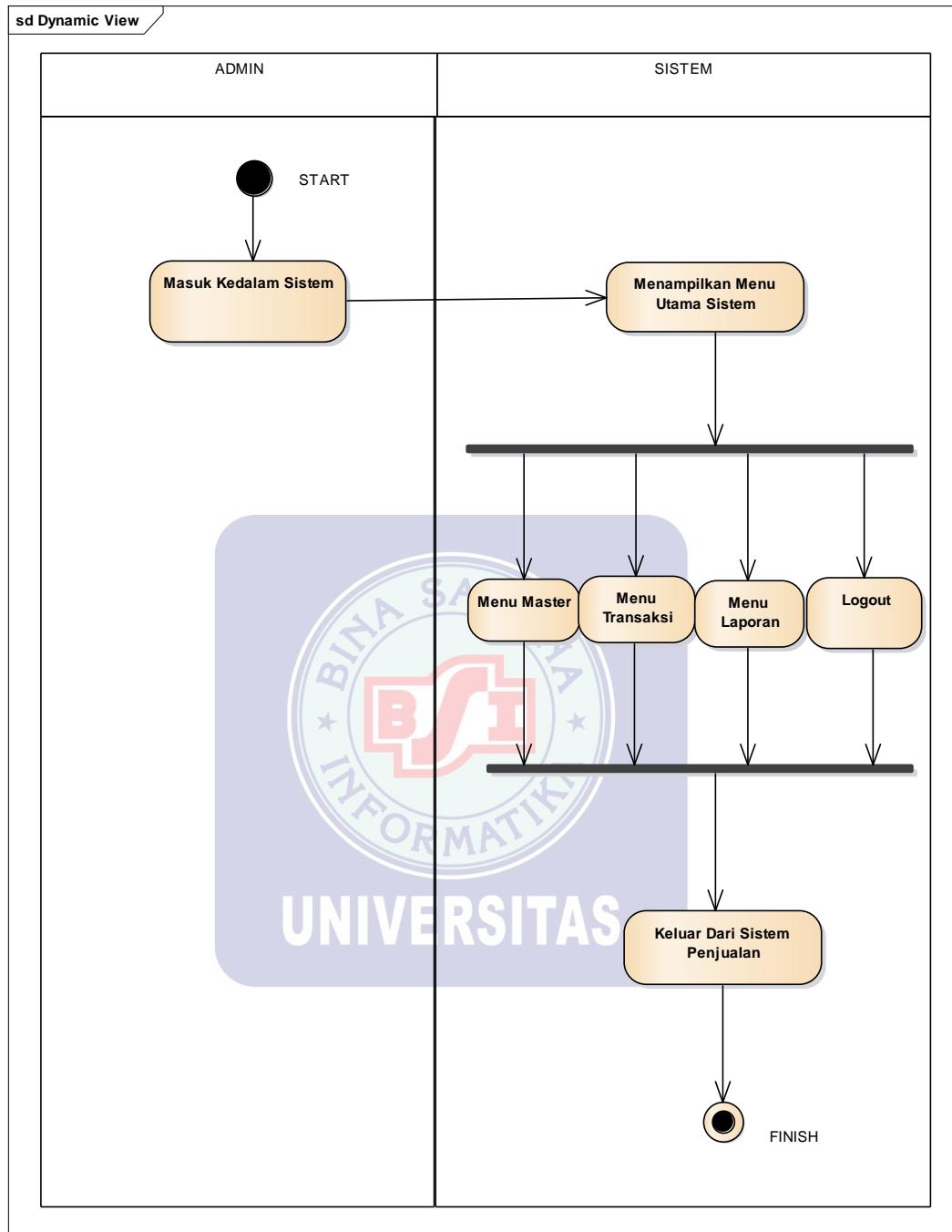
#### A. Halaman Administrasi

##### A.1 Login Admin



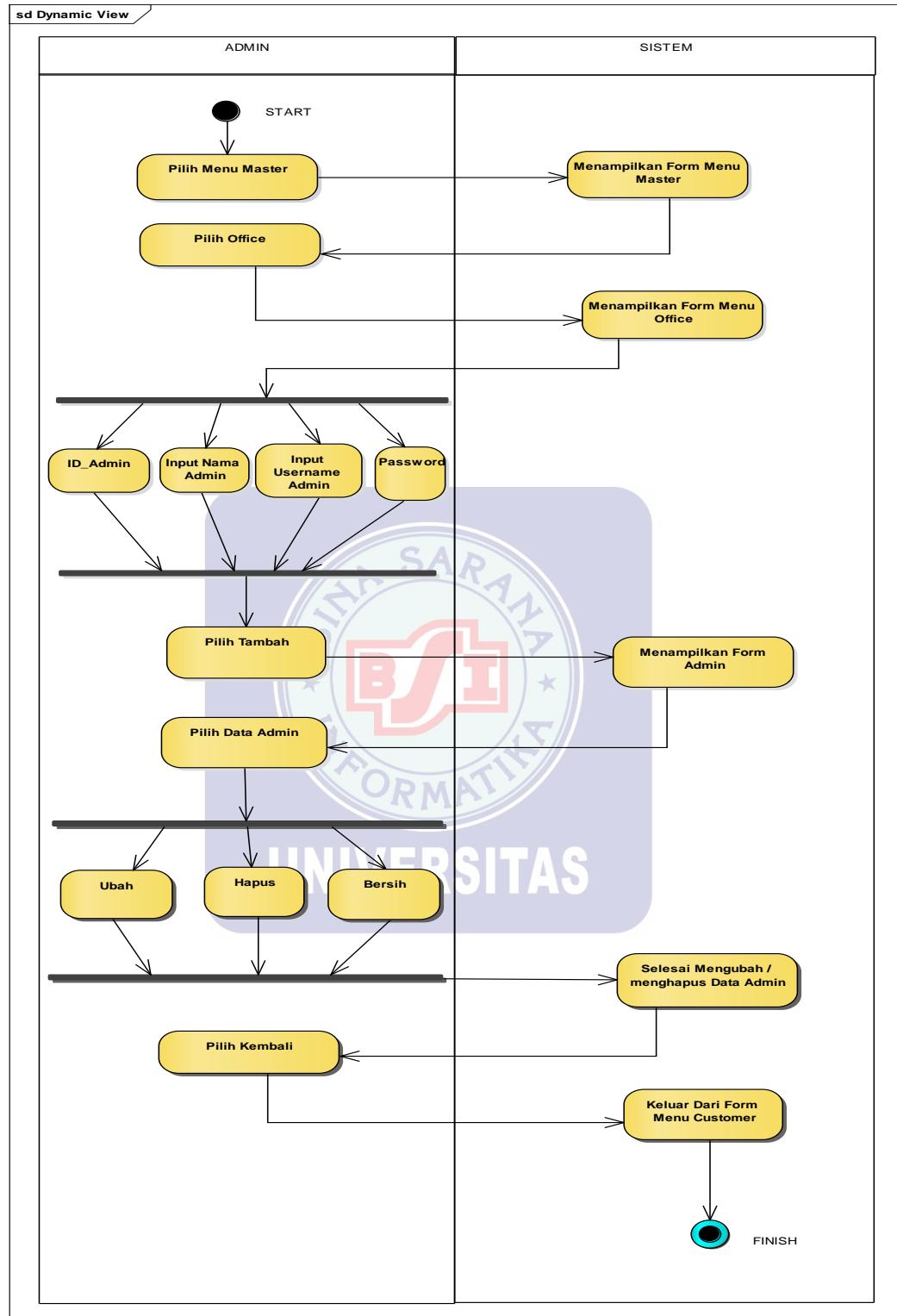
Gambar III.5. Activity Login Admin

### A.2 Menu Utama Admin



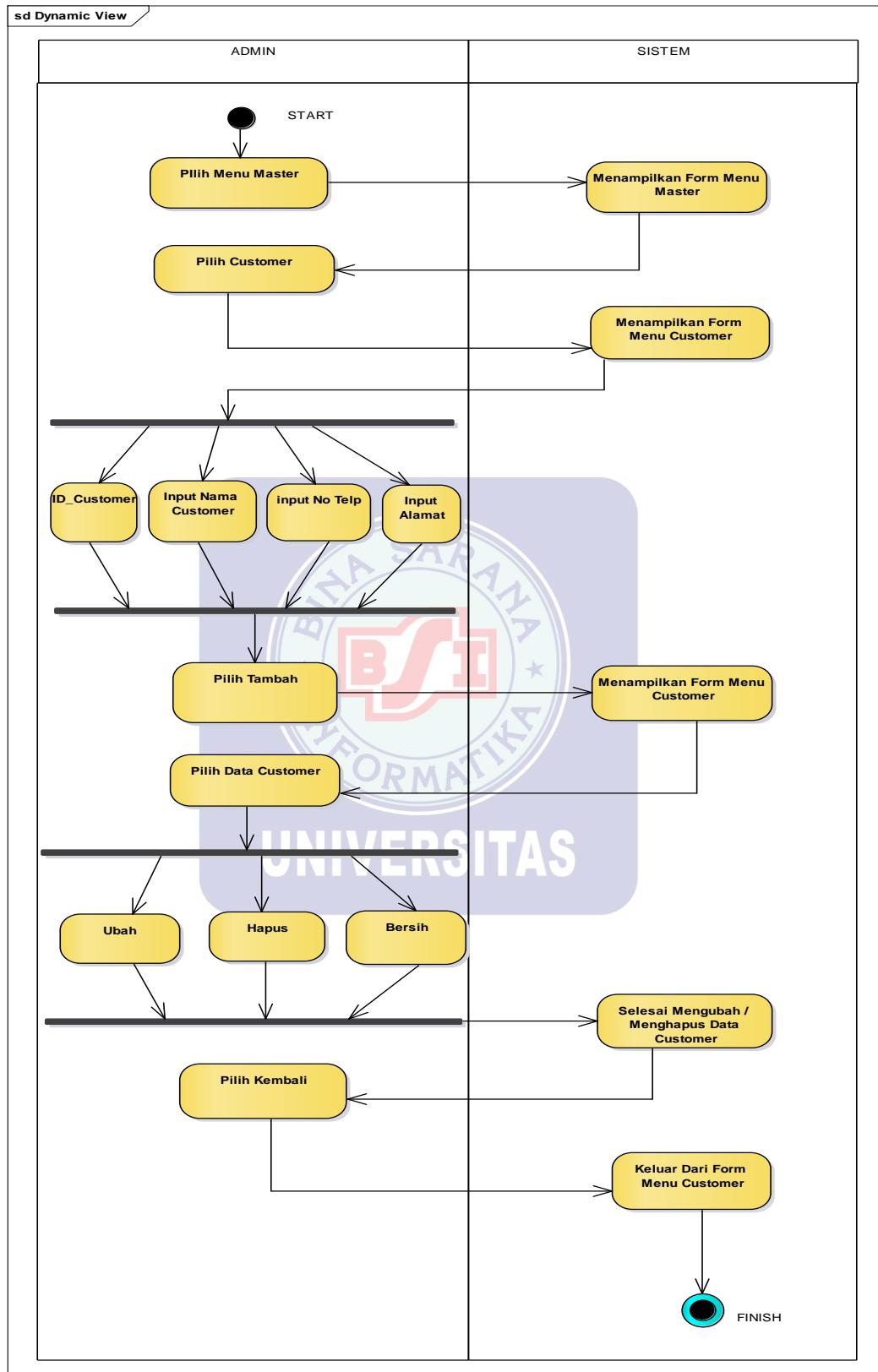
Gambar III.6. Activity Menu Utama Admin

### A.3 Menu Input Data Office / Admin



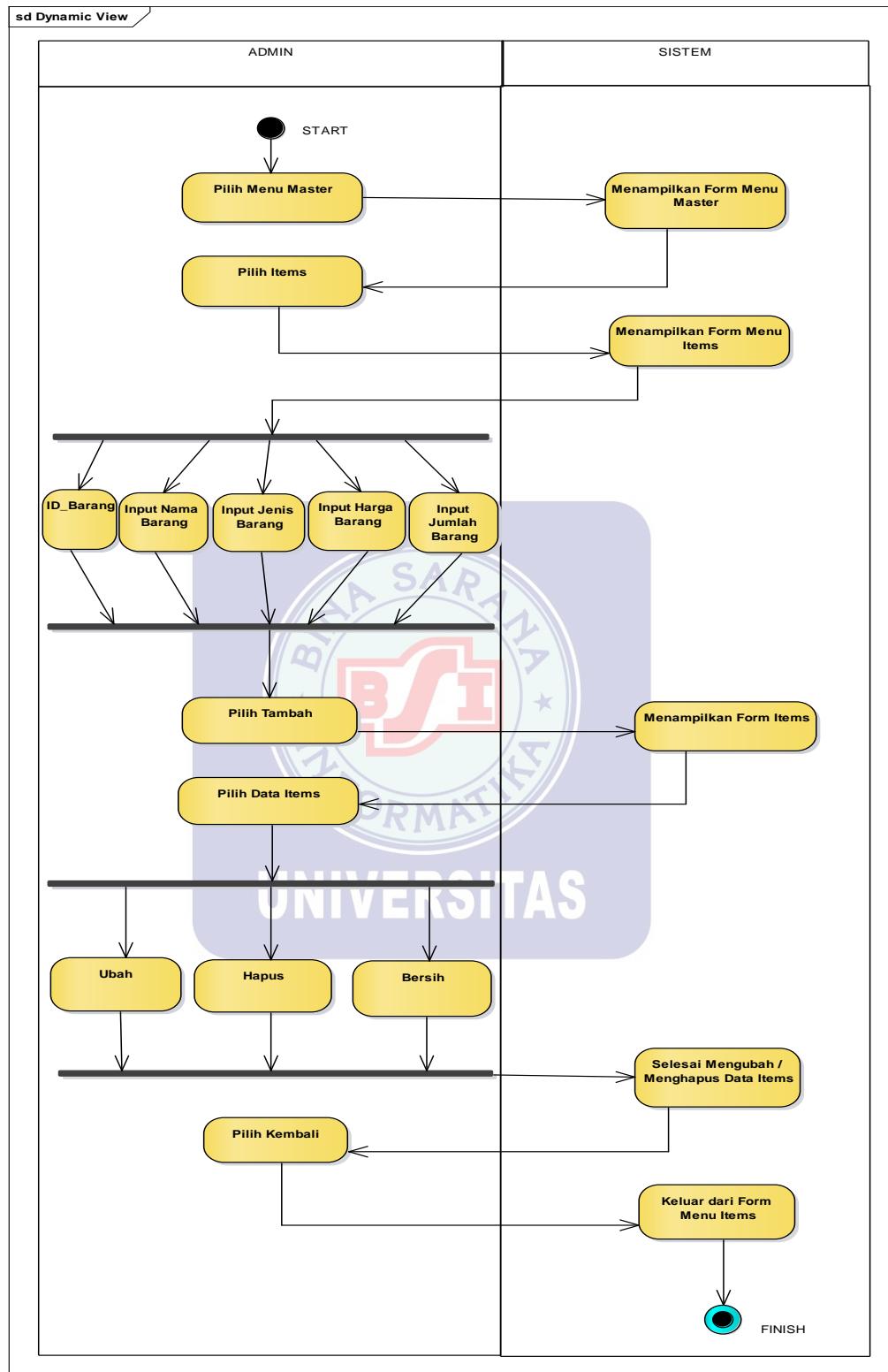
Gambar III.7. Activity Menu Input Data Admin

#### A.4 Menu Input Data Customer



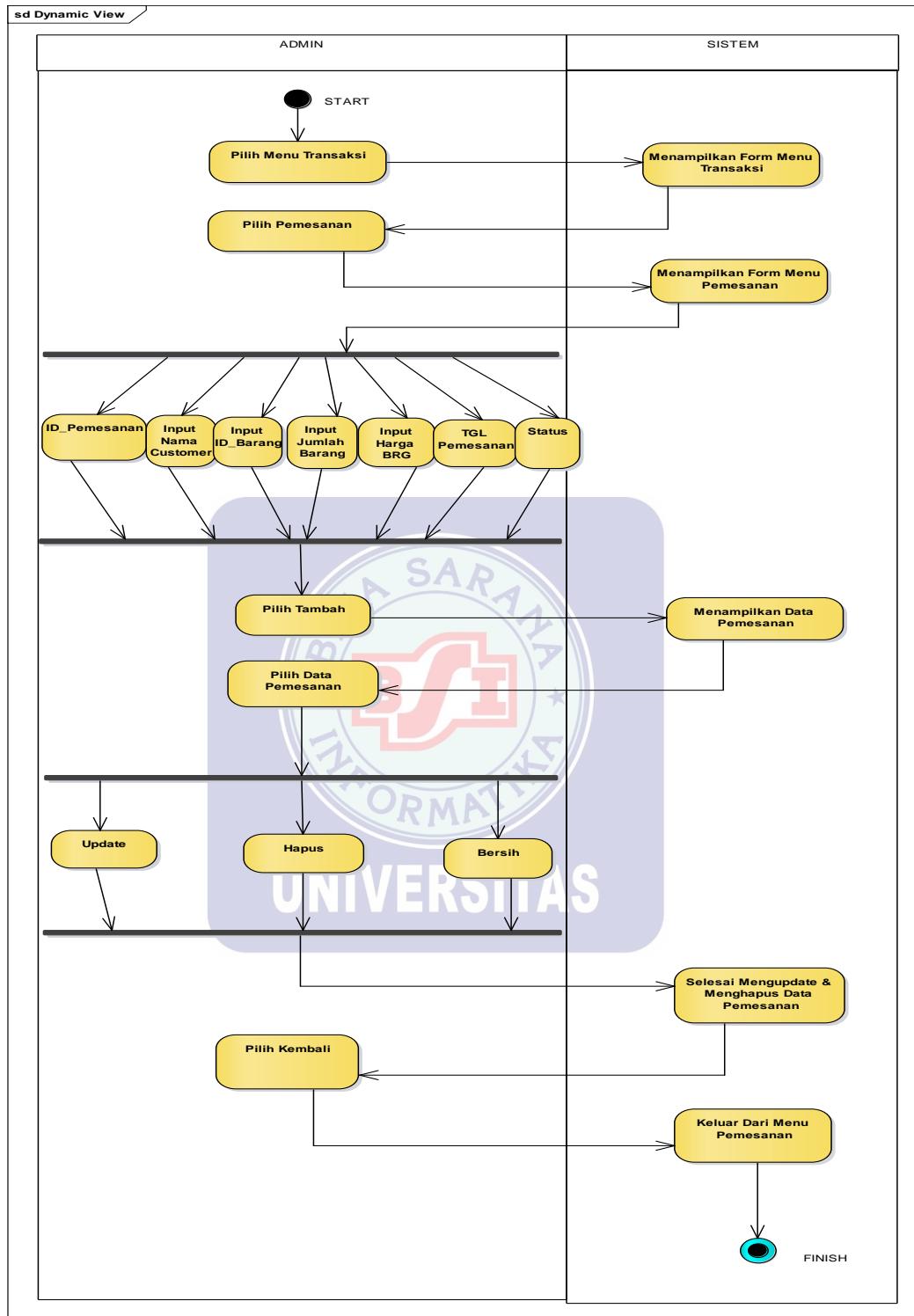
Gambar III.8. Activity Menu Input Data Customer

### A.5 Menu Input Data Barang



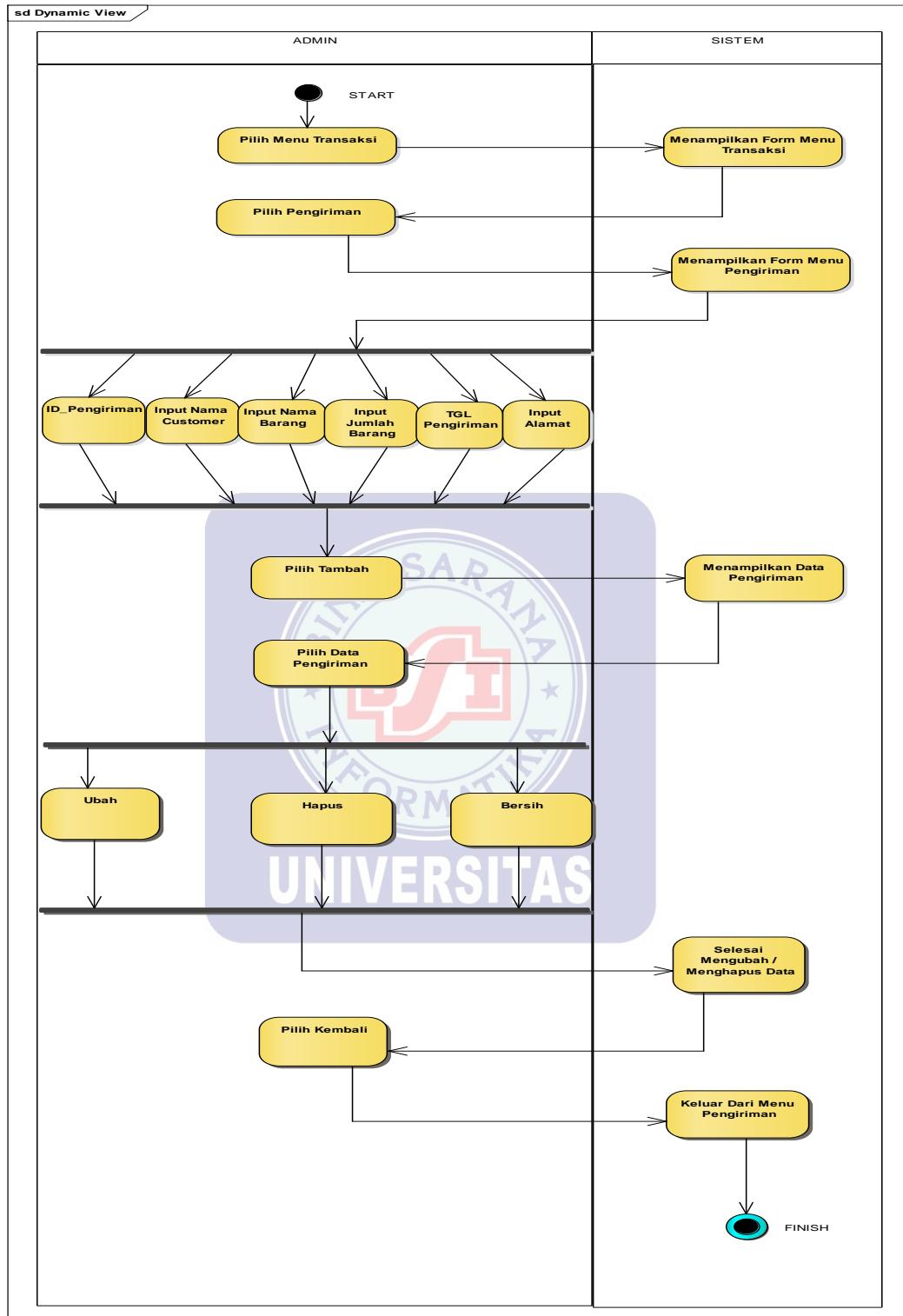
Gambar III.9. Activity Menu Input Data Barang

### A.6 Menu Input Pemesanan



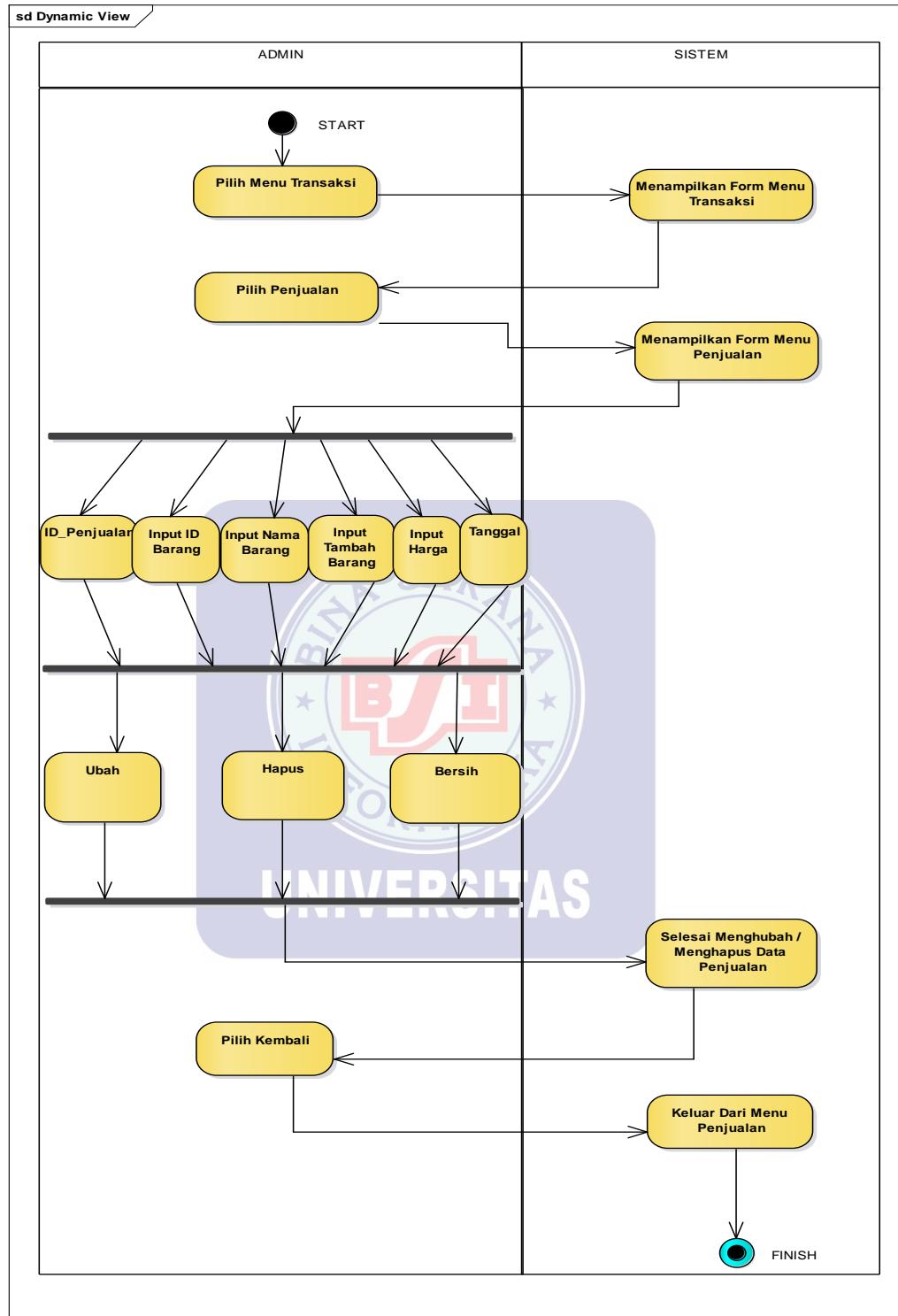
Gambar III.10. Activity Menu Input Pemesanan

### A.7 Menu Input Pengiriman



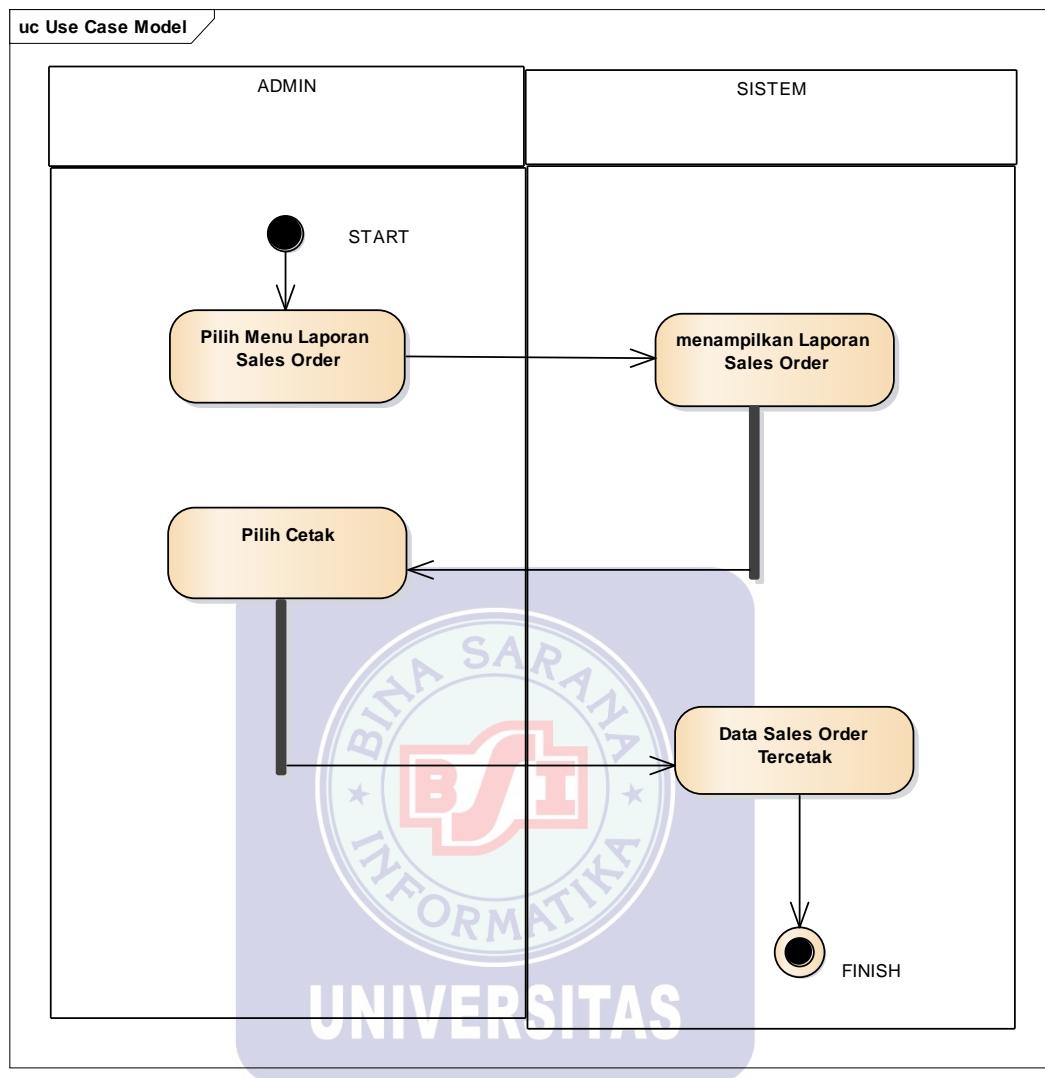
Gambar III.11. Activity Menu Input Pengiriman

### A.8 Menu Input Penjualan



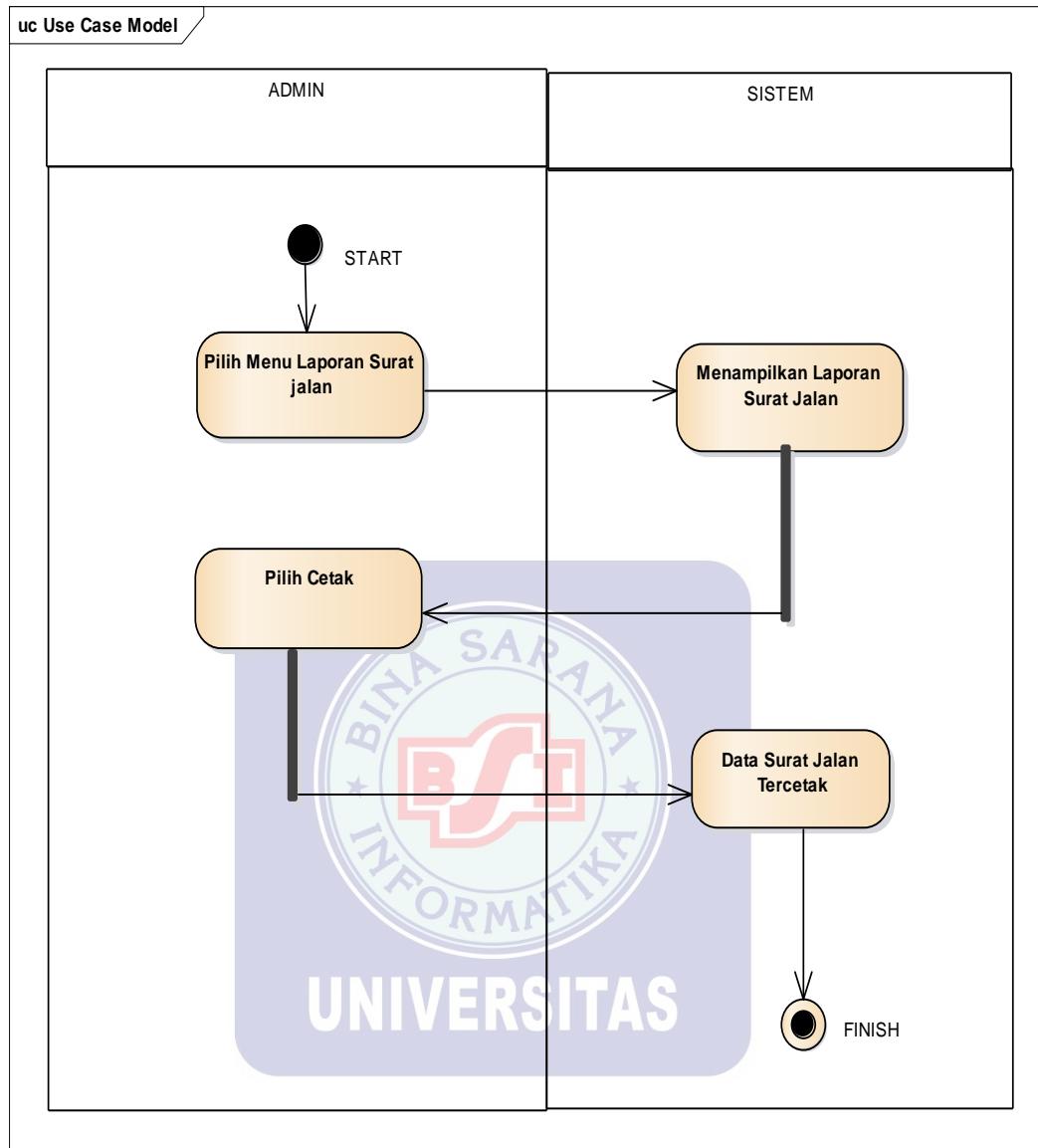
Gambar III.12. Activity Menu Input Penjualan

### A.9 Menu Cetak Sales Order



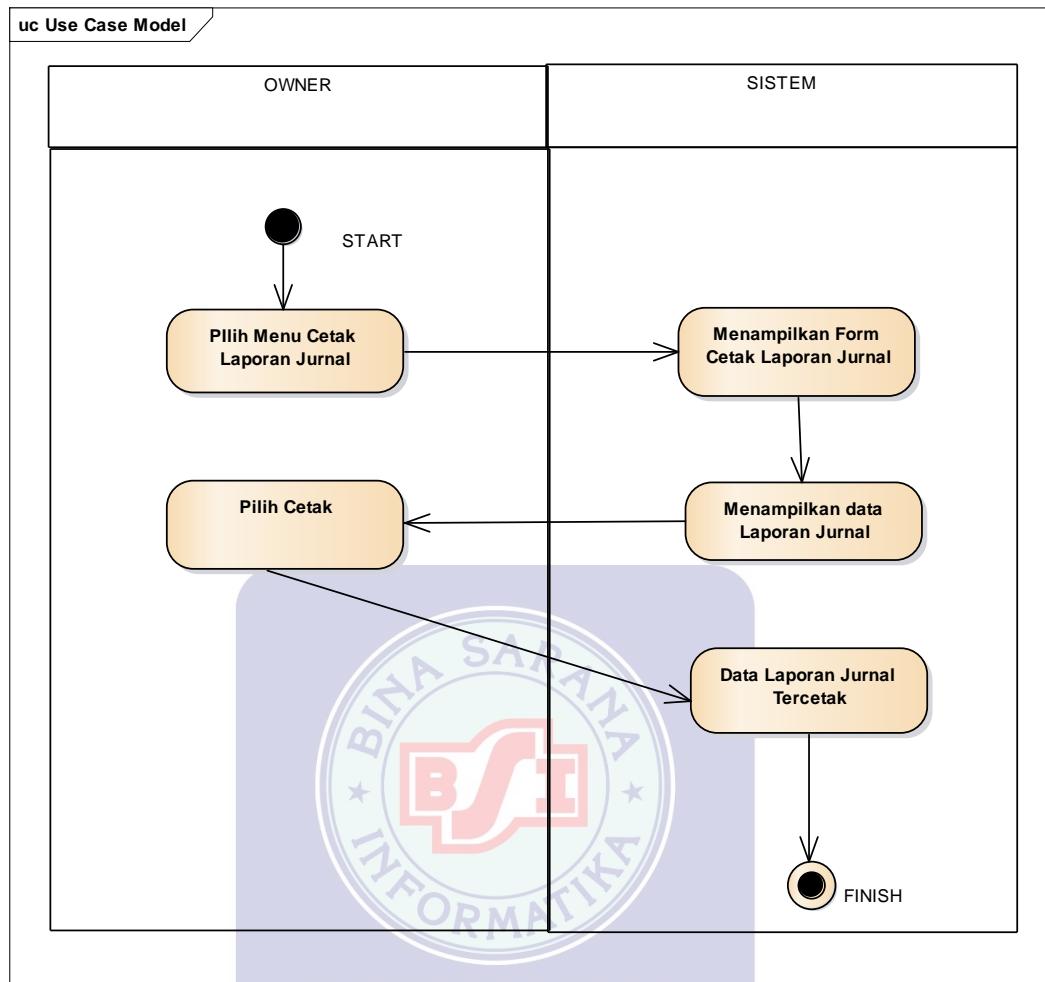
Gambar III.13. Activity Menu Cetak Sales Order

#### A.10 Menu Cetak Surat Jalan



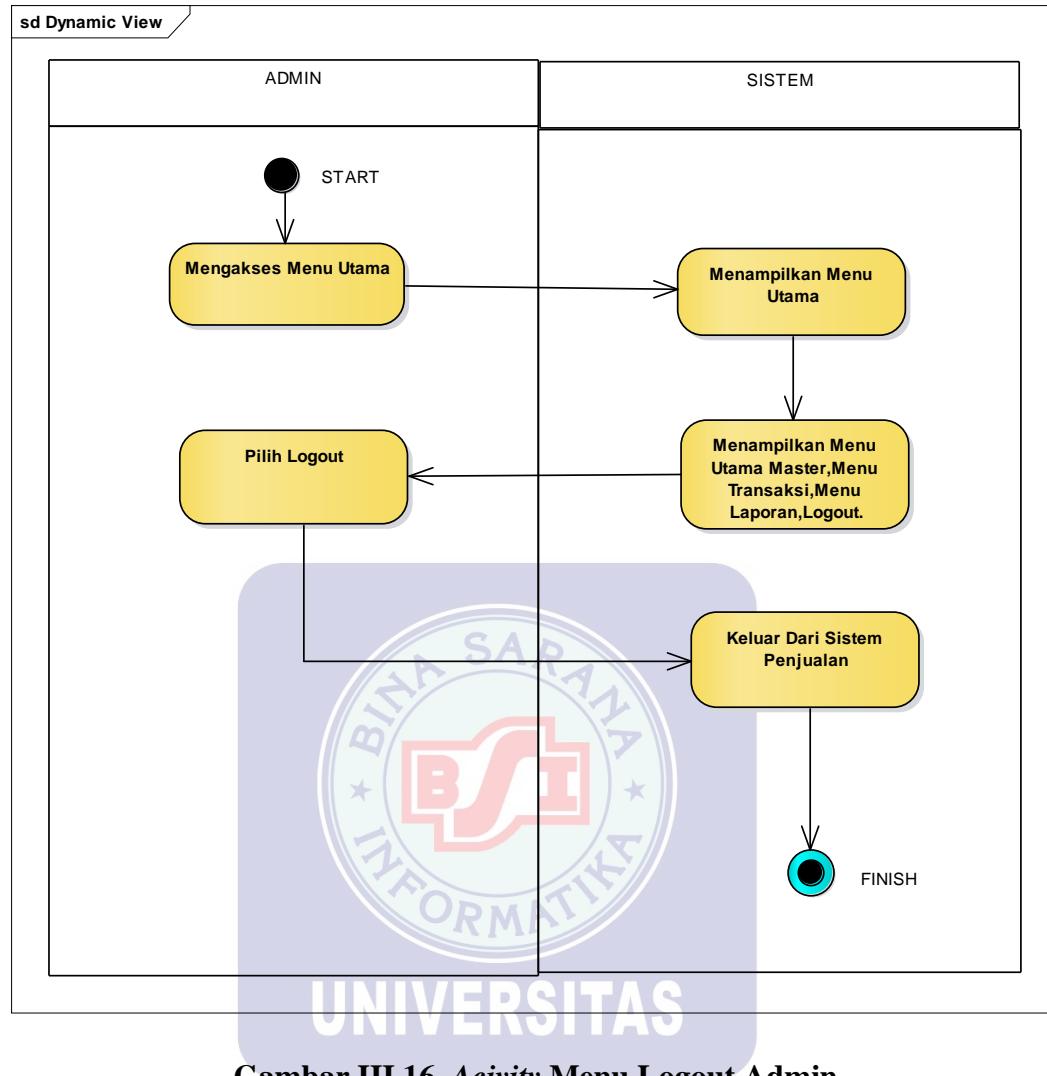
Gambar III.14. Activity Menu Cetak Surat Jalan

### A.11 Menu Cetak Jurnal



Gambar III.15. Activity Menu Cetak Surat Jalan

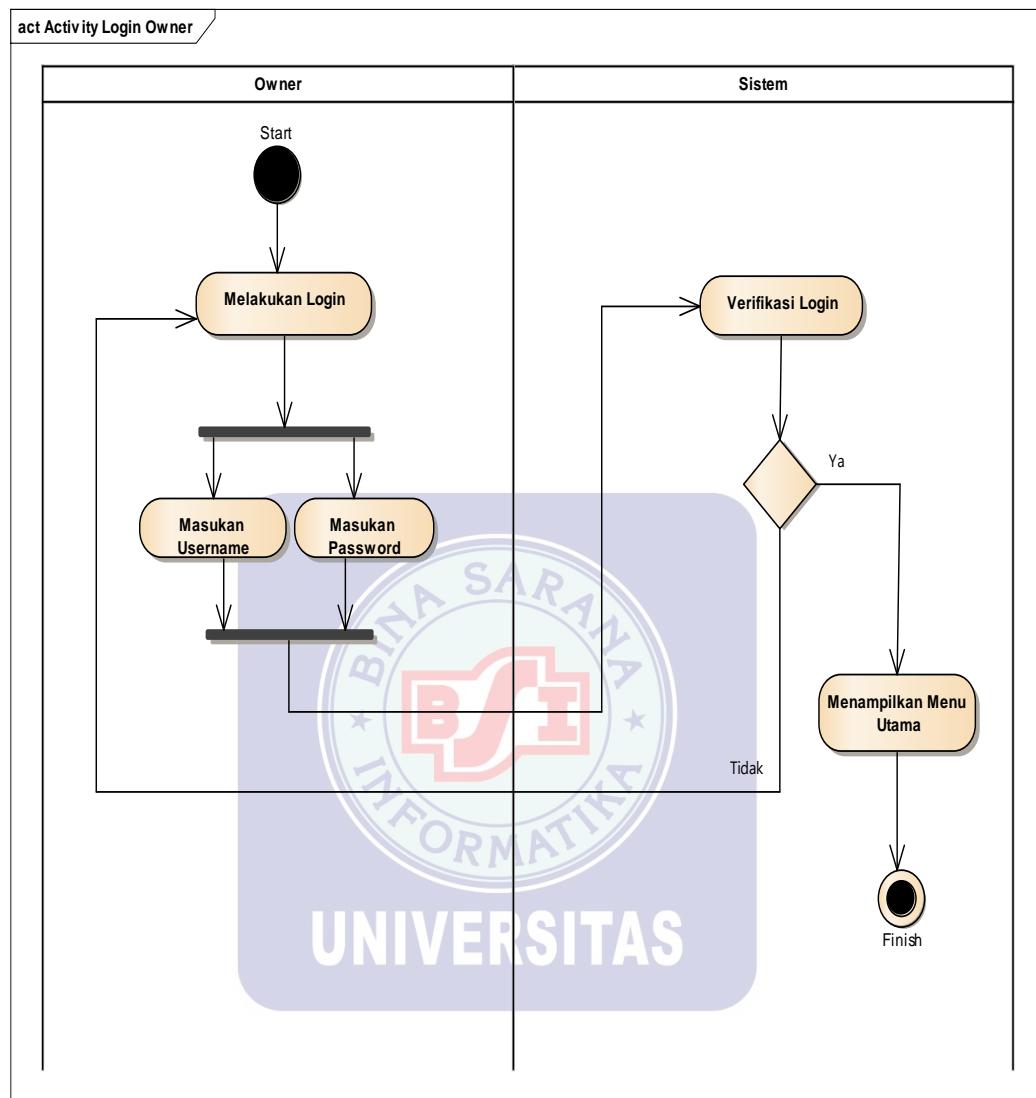
### A.12 Logout Admin



Gambar III.16. Activity Menu Logout Admin

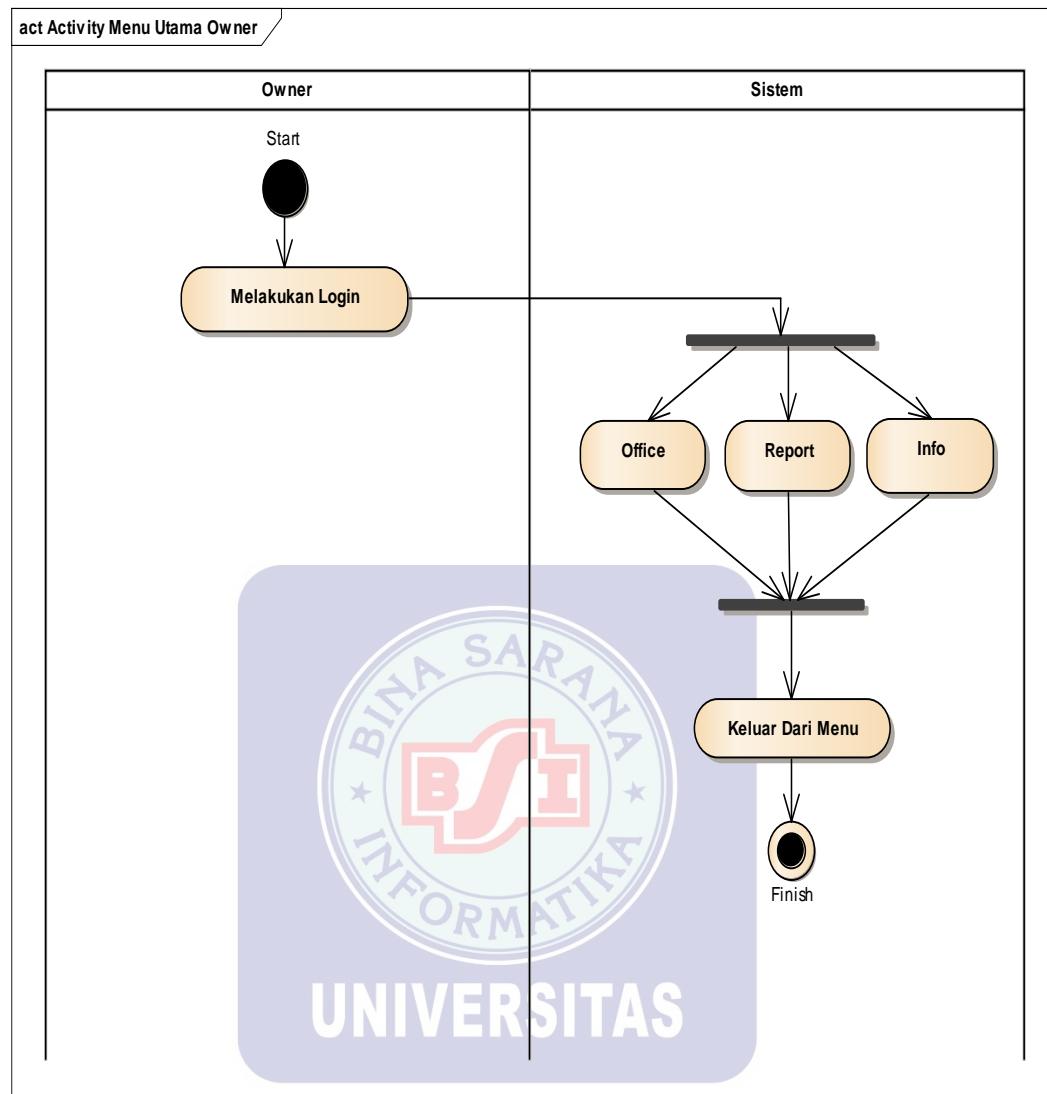
## B. Halaman Owner

### B.1 Menu Login Owner



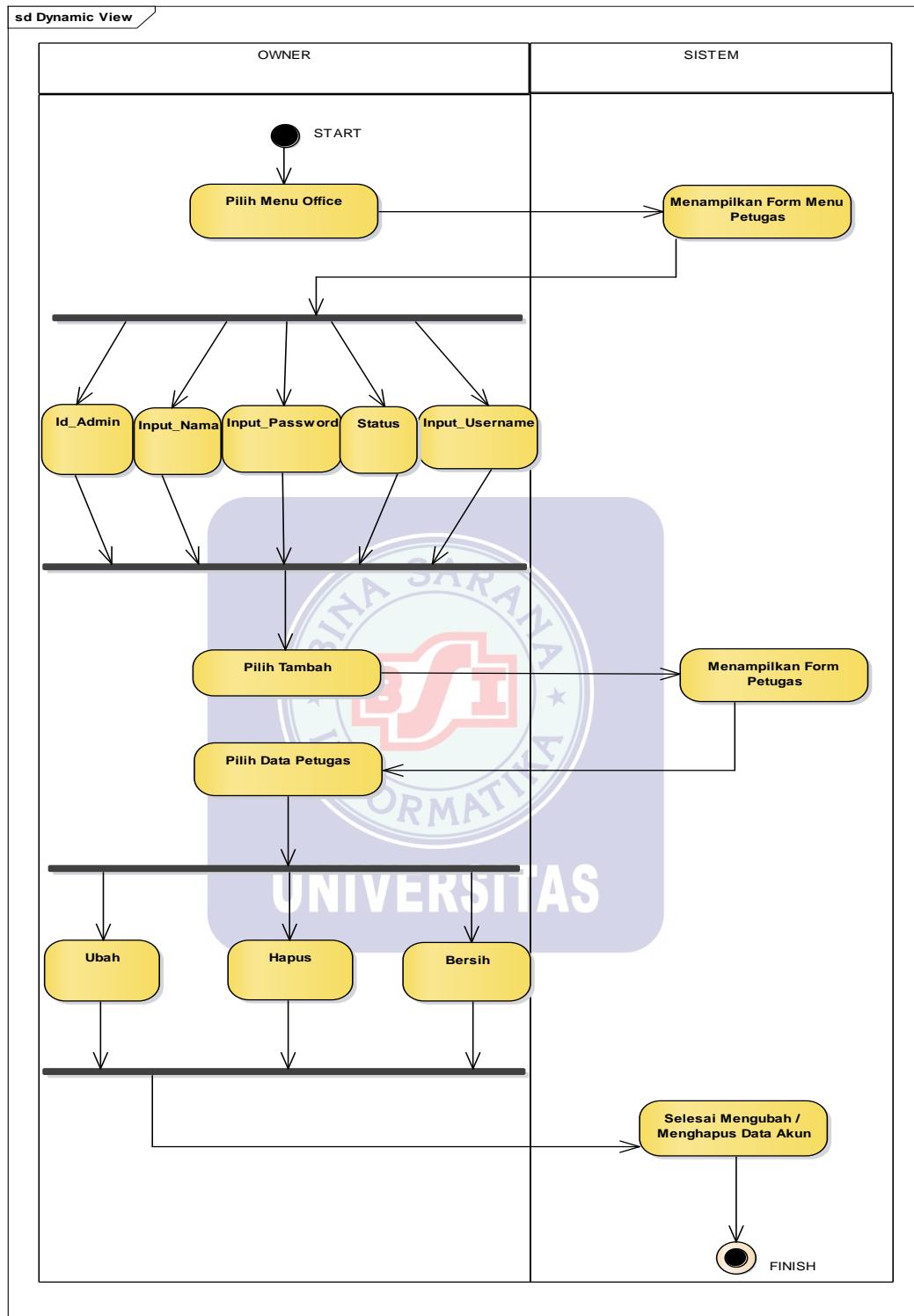
Gambar III.17. Activity Menu Login Owner

## B.2 Halaman Menu Utama Owner



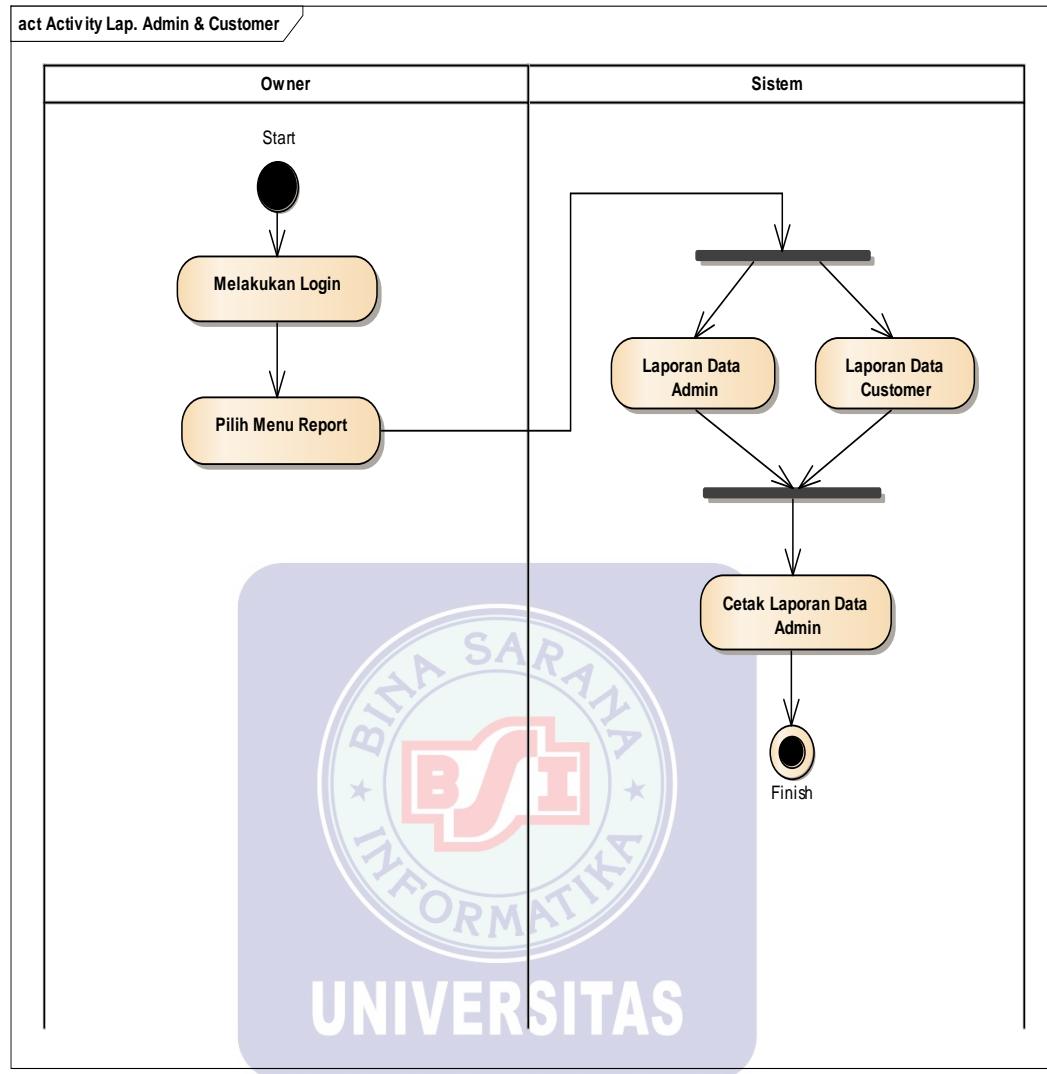
Gambar III.18. Acivity Menu Utama Owner

### B.3 Halaman Owner Input Admin



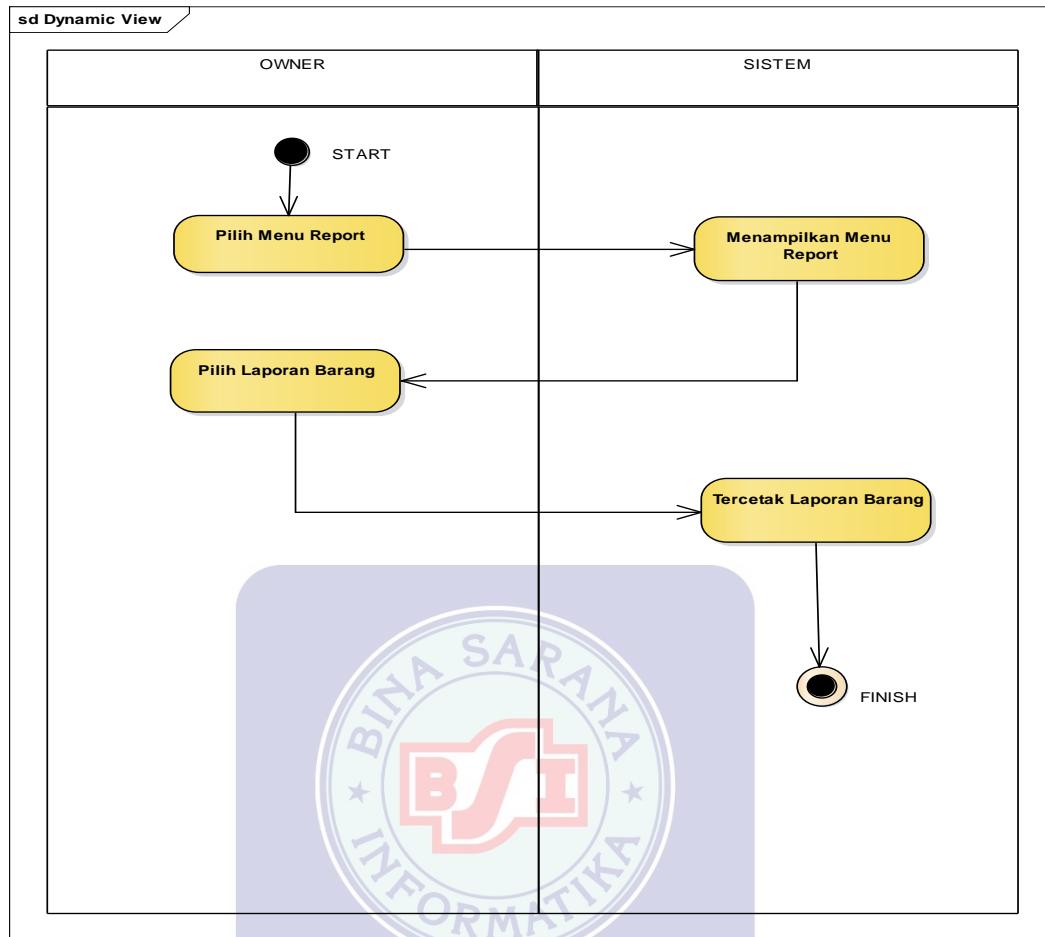
Gambar III.19. Activity Menu Input Office

#### B.4 Halaman Laporan Data Admin & Customer



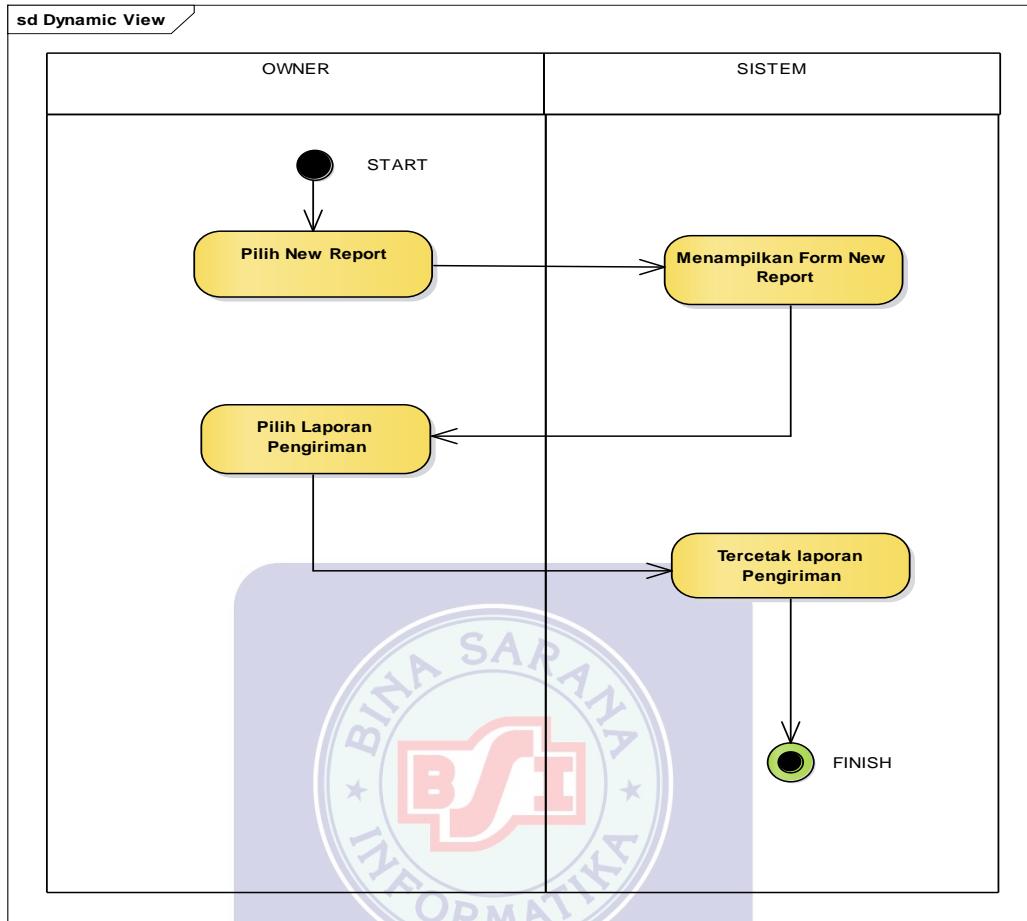
Gambar III.20. Acivity Menu Cetak Laporan Admin dan Customer

### B.5 Owner Mencetak Laporan Barang



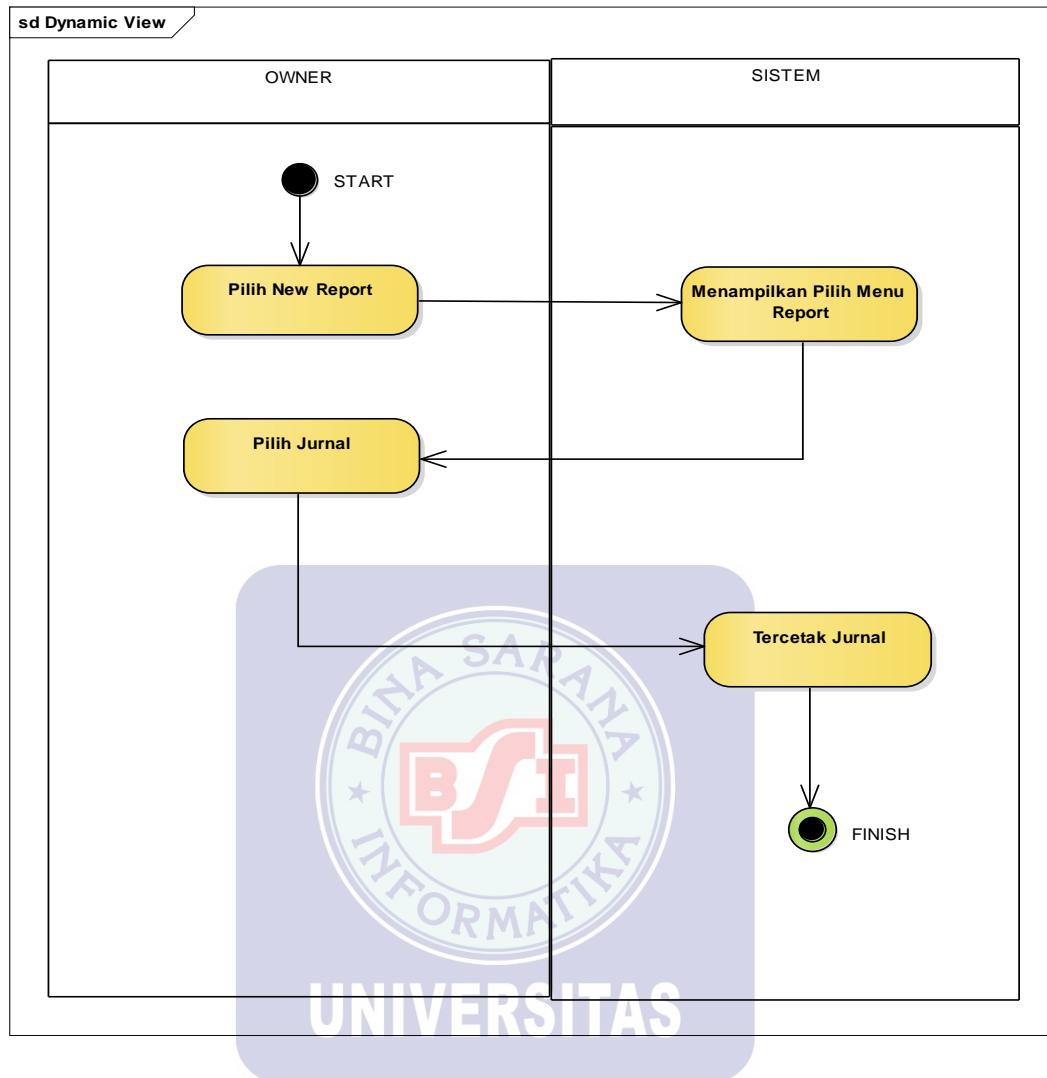
Gambar III.21. Activity Menu Cetak Laporan Barang

### B.6 Owner Mencetak Laporan Pengiriman



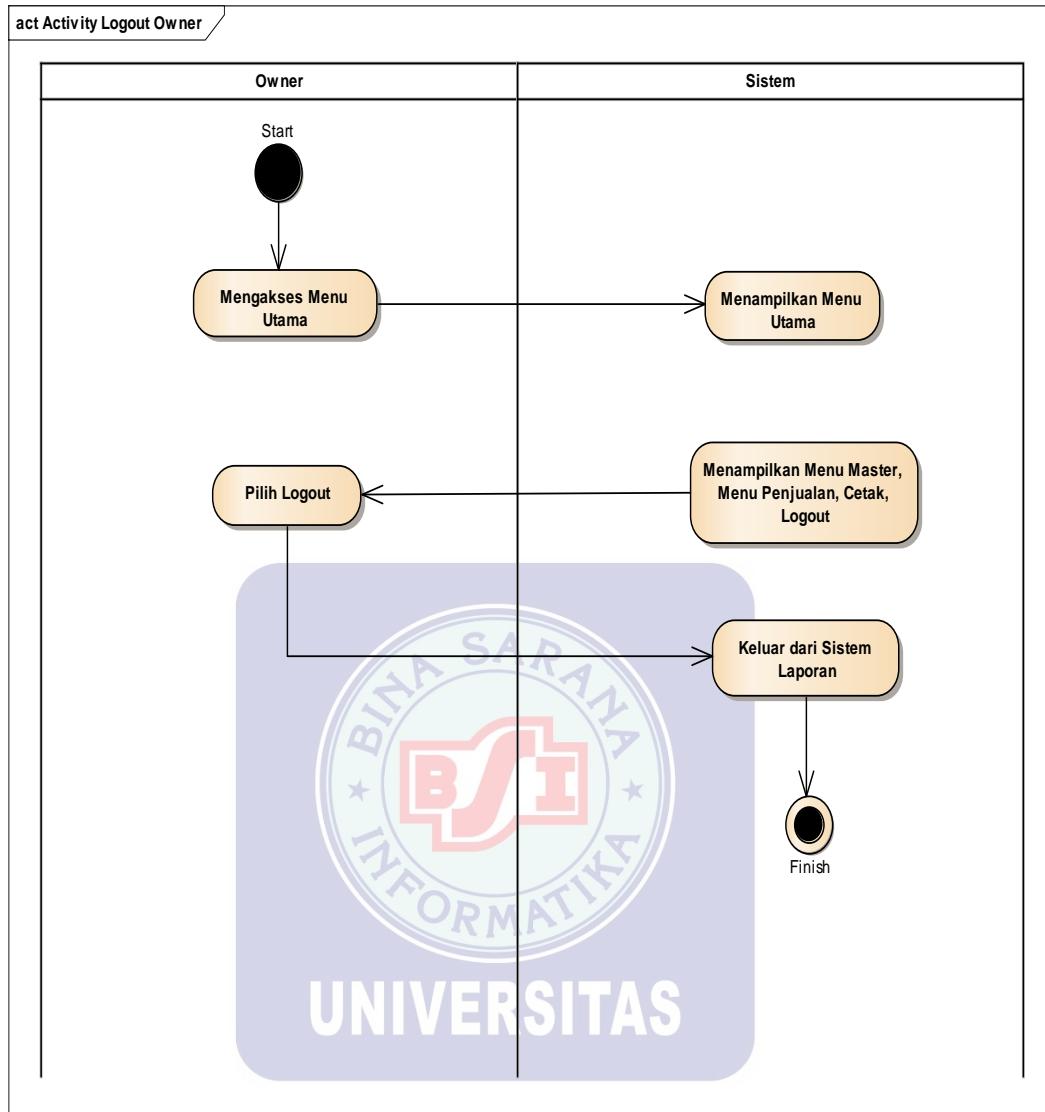
Gambar III.22. Activity Menu Cetak Laporan Pengiriman

### B.7 Halaman Laporan Cetak Jurnal



Gambar III.23. Activity Menu Cetak Jurnal

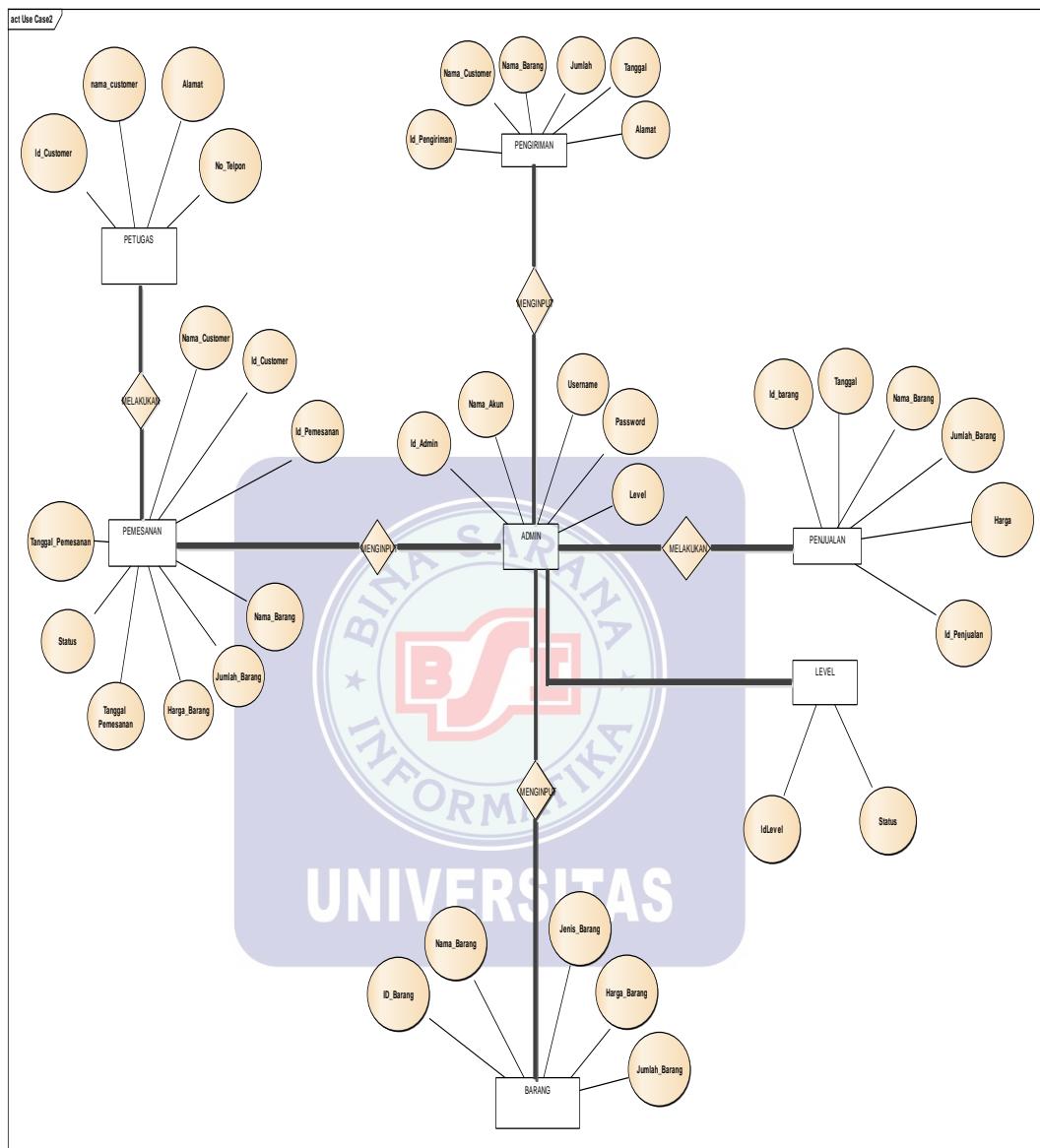
### B.8 Halaman Logout Owner



Gambar III.24. Activity Logout Owner

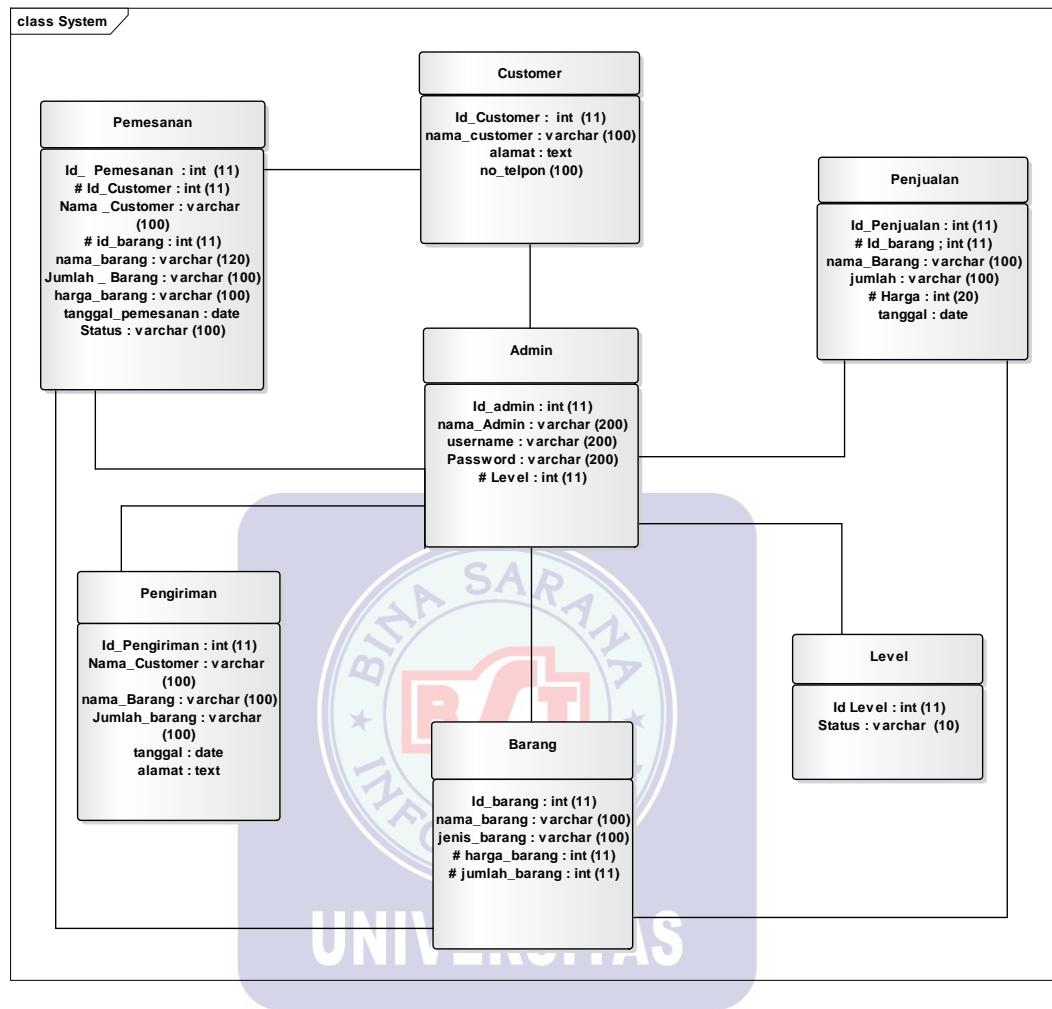
### 3.4. Desain

#### 3.4.1. Entity Relationship Diagram (ERD)



Gambar III.25. Entity Relationship Diagram

### 3.4.2. Logical Record Structure (LRS)



Gambar III.26. Logical Record Structure

### 3.4.3. Spesifikasi File

Spesifikasi *file* yang digunakan dalam perancangan program penjualan tunai ini, penulis membuat satu *file database* yaitu sistem\_db yang terdiri dari:

**Tabel III.3. Spesifikasi File Admin**

No.	Elmen data	Akronim	Tipe	Panjang	Keterangan
1	<i>Admin</i>	Id_admin	Interger	11	<i>Primary Key</i>
2	<i>Nama Admin</i>	Nama_admin	Varchar	200	
3	<i>Username</i>	Username	Varchar	200	
4	<i>Password</i>	Password	Varchar	200	
5	<i>Level</i>	Level	Interger	11	

1. Nama *File* : *Admin*
- Fungsi : Digunakan untuk mengelola data pengguna
- Akronim : Owner
- Tipe : *Master*
- Media *File* : Harddisk
- Organisasi *File* : *Index sequential*
- Akses *File* : Random
- Panjang *Record* : 611 *Character*
- Kunci *Field* : Id\_Admin
- Software* : MY SQL

**Tabel III.4. Spesifikasi *File* Barang**

No.	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1.	Kode Barang	id_barang	Integer	11	Primary key
2.	Nama Barang	nama_barang	Varchar	100	
3.	Jenis Barang	jenis_barang	Varchar	100	
4.	Harga	harga_barang	Integer	11	
5.	Jumlah Barang	jumlah_barang	Integer	11	

2. Nama *File*: *File* Barang

Fungsi

: Digunakan untuk menginput pengiriman barang

Akronim

: Pengiriman

Tipe

: *File* TransaksiMedia*File*

: Harddisk

Organisasi *File*: *Indexed sequential*Akses *File*

: Random

Panjang Record

: 233 Character

Kunci Field : Id\_Barang

Software : MySQL

**Tabel III.5. Spesifikasi File Customer**

No.	Elemen data	Akronim	Tipe	Panjang	Keterangan
1	Id_Costumer	Id_cust	Integer	11	Primary Key
2	Nama costumer	nama_cust	Varchar	100	
3	Alamat	Alamat	Text		
4	No Telpon	no_telp	Varchar	12	

3. Nama File : *File Customer*
- Fungsi : Digunakan untuk menyimpan data *customer*
- Akronim : Admin
- Tipe : *Master*
- Media File : Harddisk
- Organisasi File : *Indexed sequential*
- Akses File : Random
- Panjang Record : 121 Character
- Kunci Field : Id\_Customer

*Software* : MY SQL

**Tabel III.6. Spesifikasi File Pemesanan**

No.	Elemen data	Akronim	Tipe	Panjang	Keterangan
1	Id_pemesanan	id_pemesanan	Interger	11	<i>Primary Key</i>
2	Id_costumer	Id_cust	Interger	11	
3	Nama Customer	nama_cust	Varchar	100	
4	Id_barang	id_barang	Interger	11	
5	Nama_barang	nama_barang	Varchar	120	
6	Jumlah	jumlah_barang	varchar	100	
7	Harga	harga_barang	Varchar	100	
8	Tanggal	tanggal_barang	Date		
9	Status	Status	Varchar	100	

4. Nama File : File Pemesanan

Fungsi : Digunakan untuk menyimpan data barang

Akronim : Barang

Tipe : Master

Media File : Harddisk

Organisasi File : Indexed sequential

Akses *File* : Random

Panjang *Record* : 553 *Character*

Kunci *Field* : Id\_Pemesanan

*Software* : MY SQL

**Tabel III.7. Spesifikasi *File* Pengiriman**

No.	Elemen data	Akronim	Tipe	Panjang	Keterangan
1	Id Pengiriman	Id_pengiriman	Interger	11	Primary Key
2	Id Customer	Id_customer	Interger	11	
3	Nama Customer	nama_cust	Varchar	100	
4	Nama Barang	nama_barang	Varchar	100	
5	Jumlah	Jumlah_barang	Varchar	100	
6	Alamat	Alamat	Text		

5. Nama *File* : *File* Pengiriman

Fungsi : Digunakan untuk menyimpan data pengiriman

Akronim : Pengiriman

Tipe : *Master*

Media *File* : Harddisk

Organisasi *File* : *Indexed sequential*

Akses *File* : Random

Panjang *Record* : 322 *Character*

Kunci *Field* : Id\_Pengiriman

*Software* : MY SQL

**Tabel III.8. Spesifikasi *File* Level**

No.	Elemen data	Akronim	Tipe	Panjang	Keterangan
1	Level	Id_level	Interger	11	Primary Key
2	Status	Status	Varchar	100	

6. Nama *File* : *File* Level

Fungsi : Digunakan untuk menentukan status

Akronim : Level

Tipe : Master

Media *File* : Harddisk

Organisasi *File* : *Indexed sequential*

Akses *File* : Random

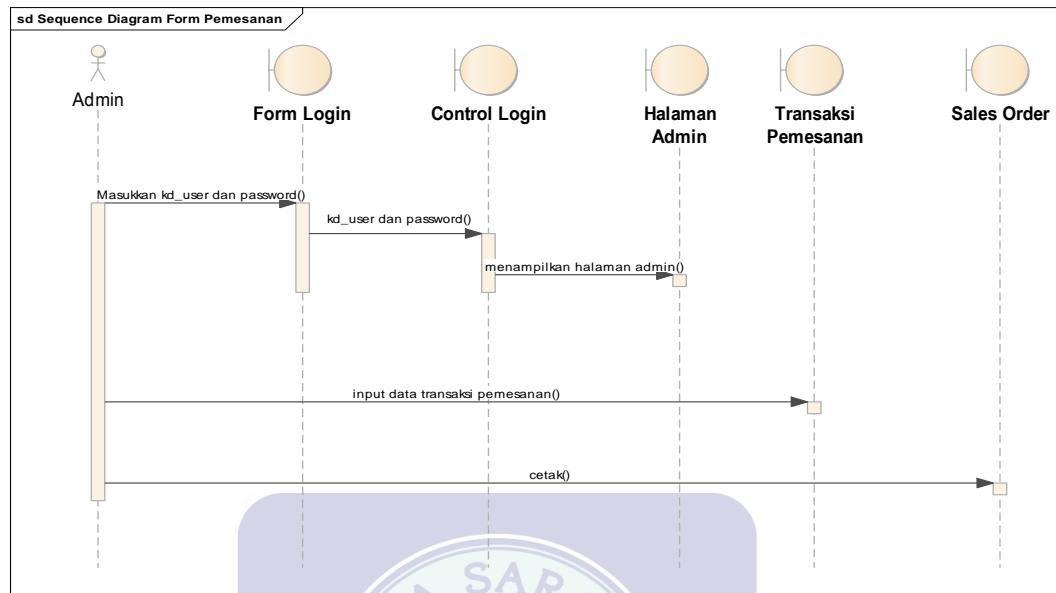
Panjang *Record* : 111 *Character*

Kunci *Field* : Id\_Level

*Software* : MY SQL

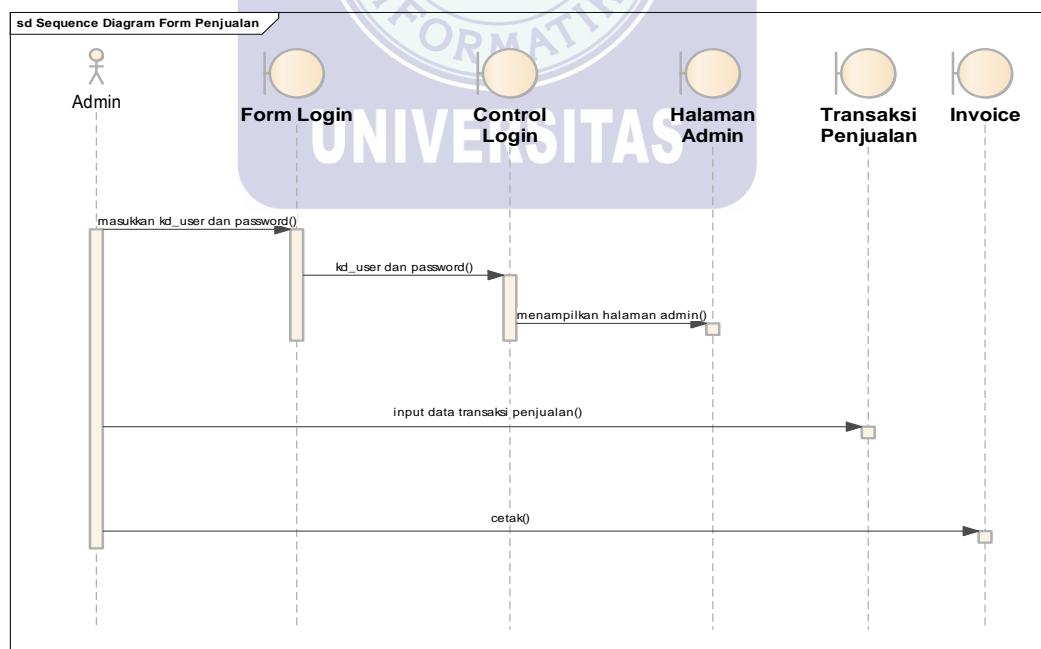
### 3.4.4. Sequence Diagram

#### 1. Sequence Diagram Transaksi Pemesanan



Gambar III.25. Deployment Diagram Transaksi Pemesanan

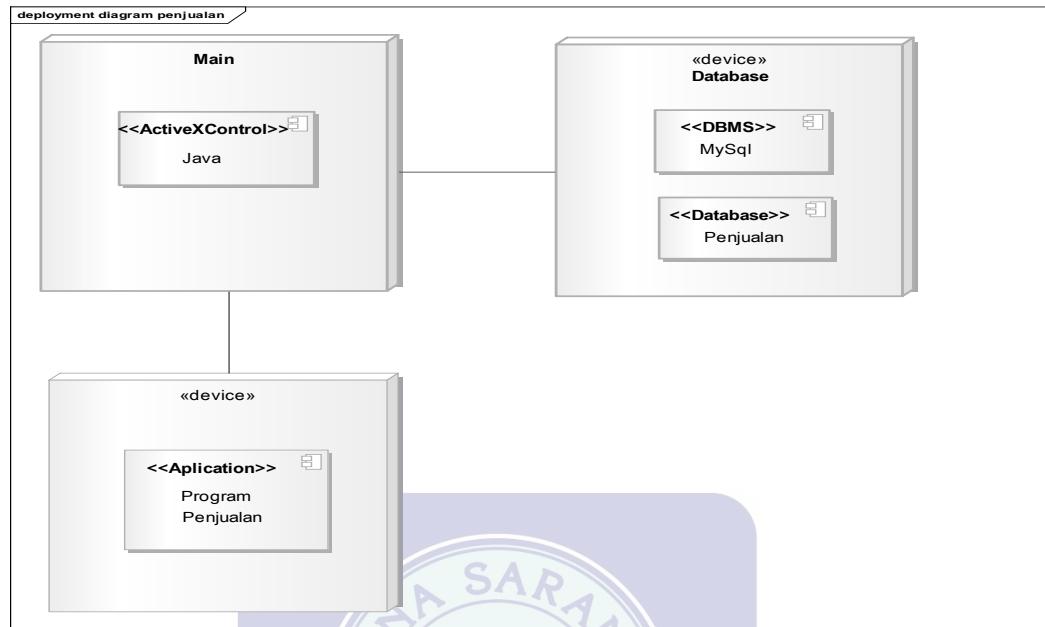
#### 2. Sequence Diagram Transaksi Penjualan



Gambar III.26. Sequence Diagram Transaksi Penjualan

### 3.4.5. Deployment Diagram

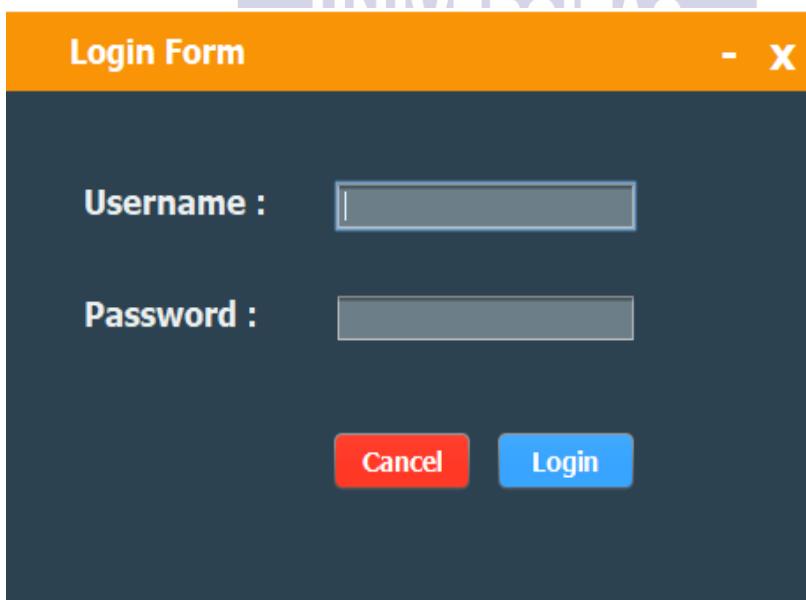
#### 1. Deployment Diagram Penjualan



Gambar III.27. Deployment Diagram Sistem Penjualan

### 3.4.6. User Interface

#### 1. User Interface Menu Login



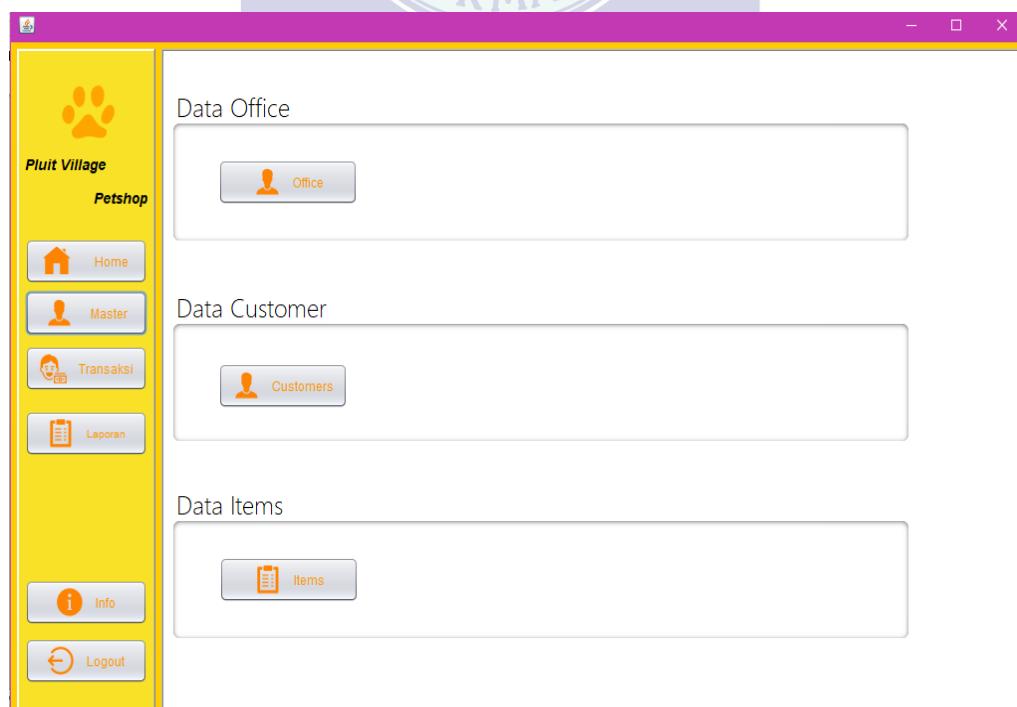
Gambar III.28. User Interface menu login

## 2. User Interface Menu Utama Admin



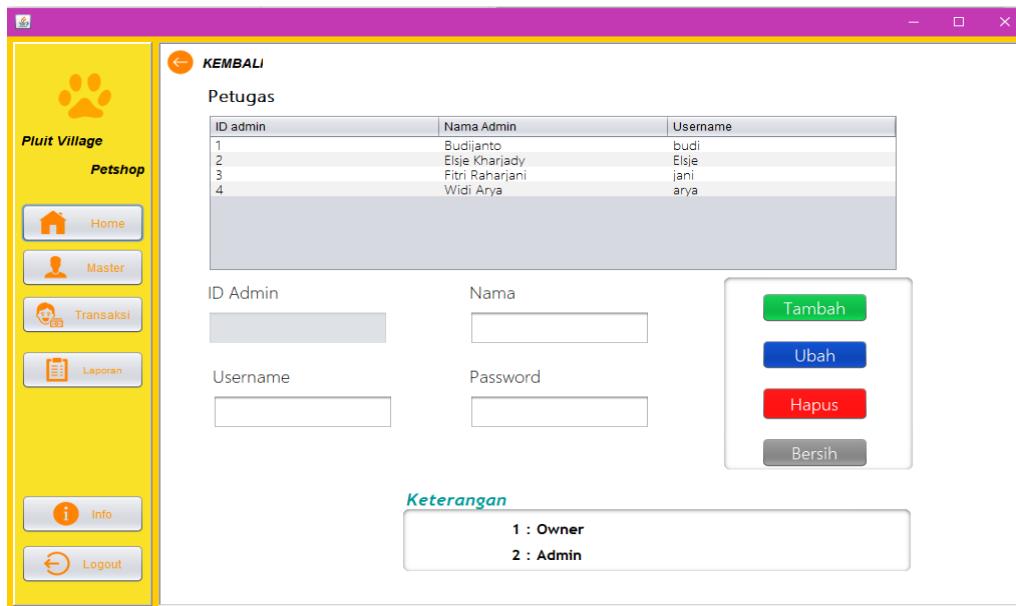
Gambar III.29. User Interface menu Utama Admin

## 3. User Interface Form Master



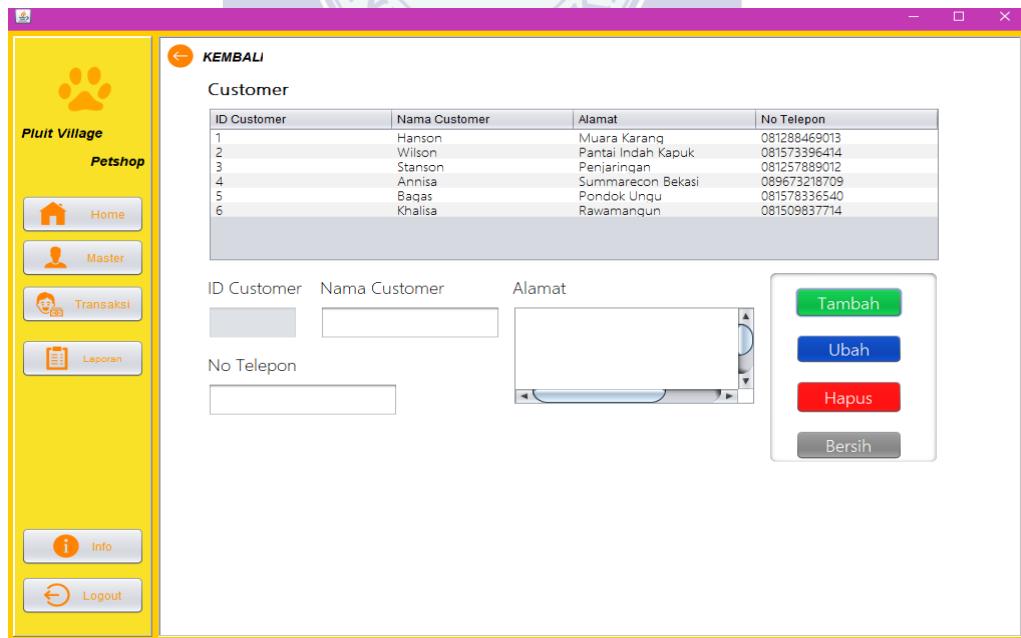
Gambar III.30. User Interface menu Master

#### 4. User Interface Input Office



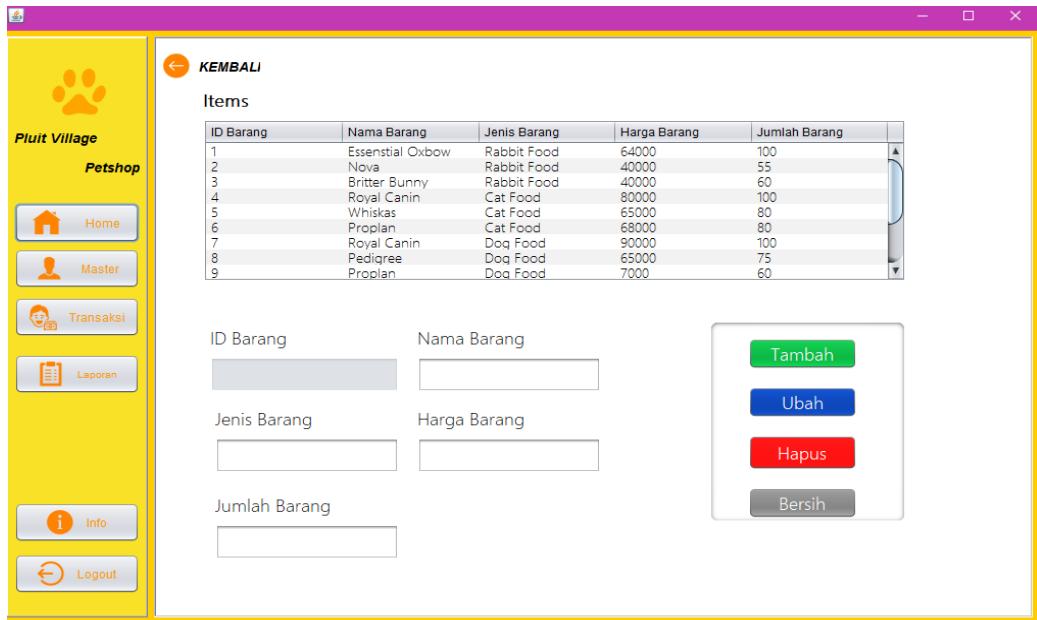
Gambar III.31. *User Interface menu Office*

#### 5. User Interface Input Customer



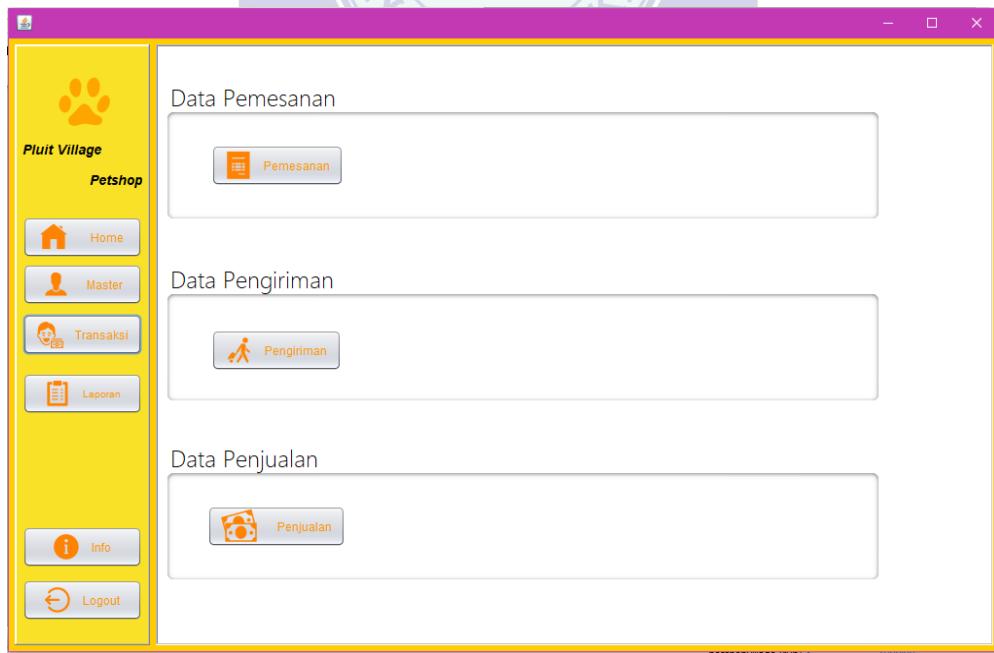
Gambar III.32. *User Interface menu Customer*

## 6. User Interface Input Items



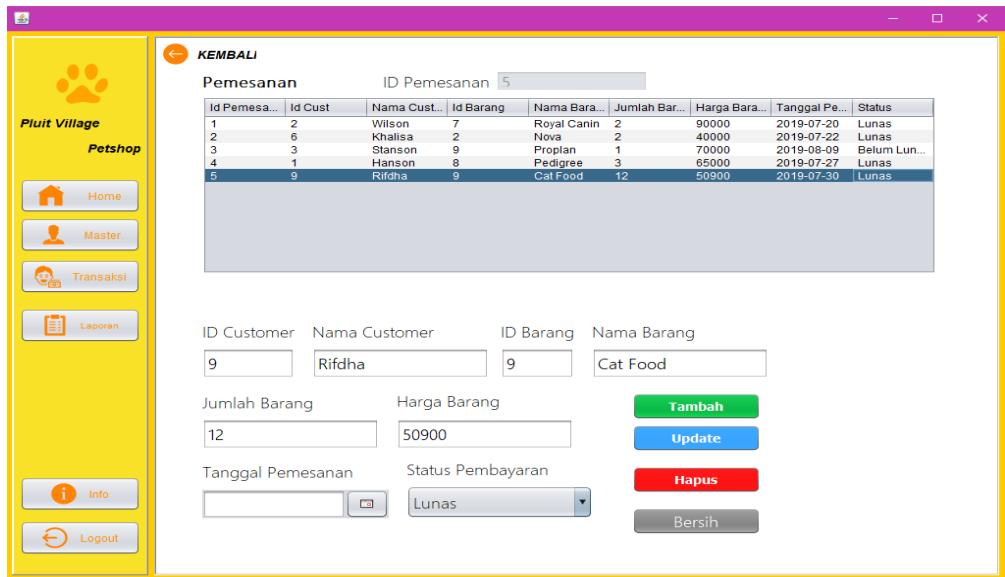
Gambar III.33. User Interface menu Barang

## 7. User Interface Menu Transaksi



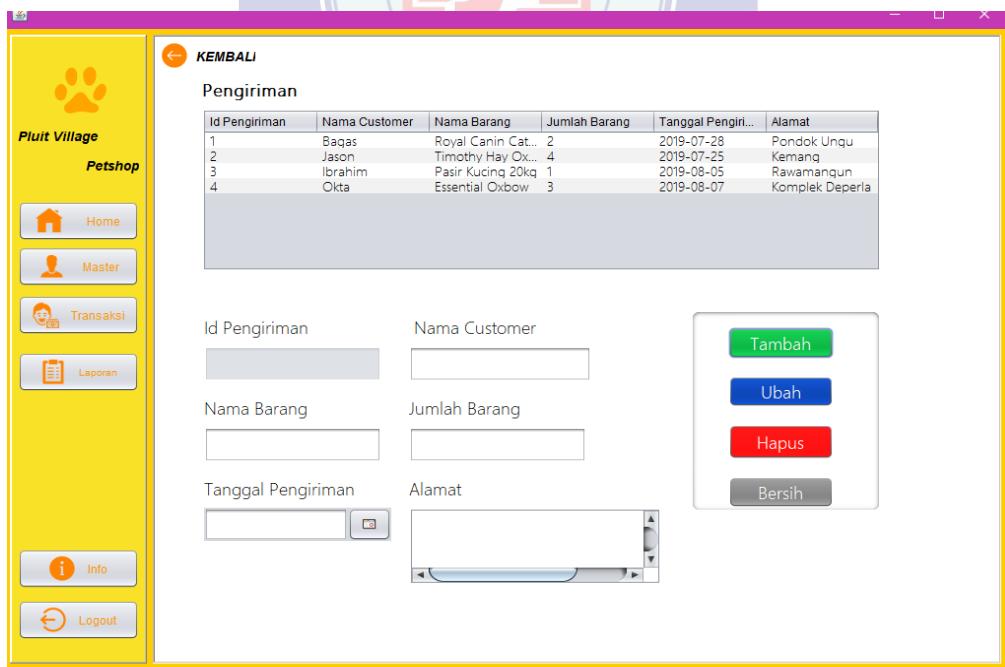
Gambar III.34. User Interface menu Transaksi

## 8. User Interface Input Pemesanan



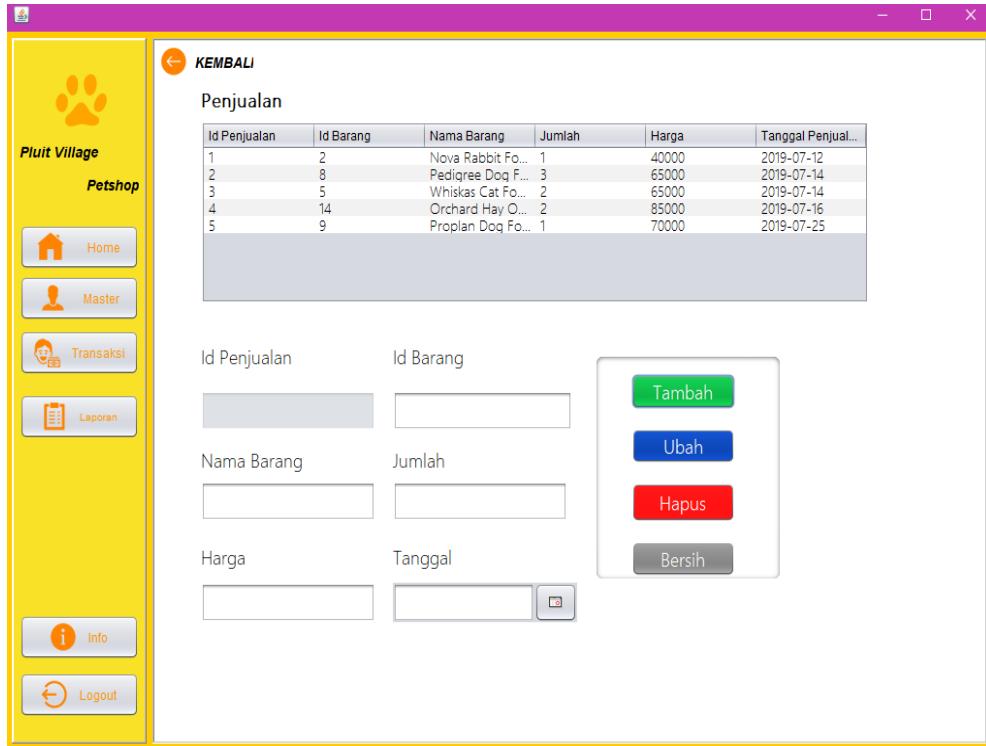
**Gambar III.35. User Interface Input Pemesanan**

## 9. User Interface Input Pengiriman



**Gambar III.36. User Interface Input Pengiriman**

### 10. User Interface Input Penjualan



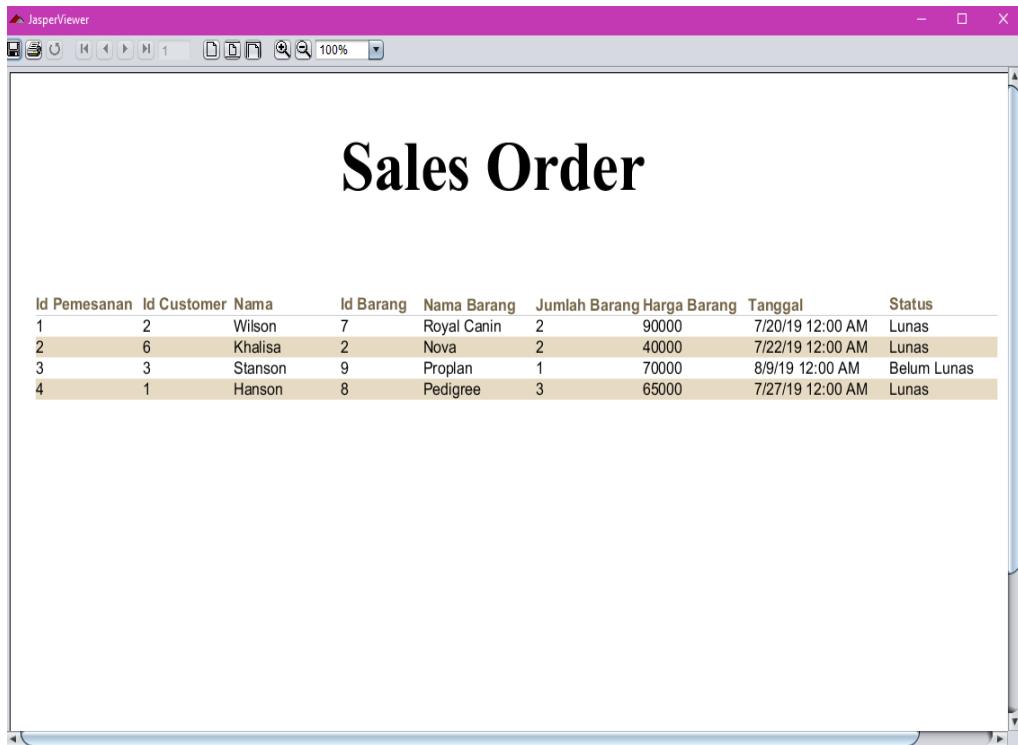
Gambar III.37. User Interface Input Penjualan

### 11. User Interface Form Laporan



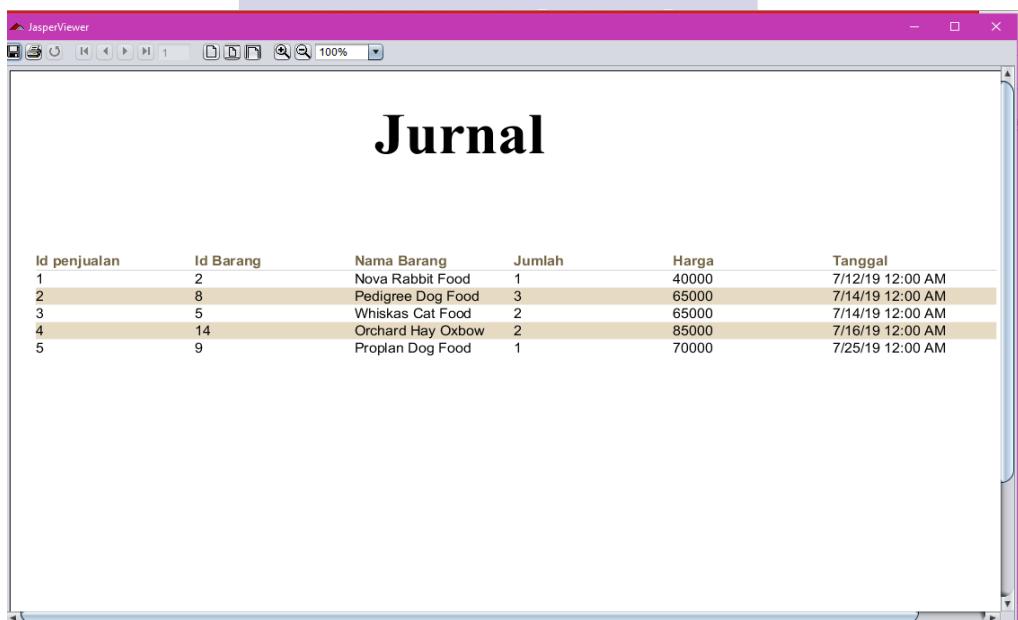
Gambar III.38. User Interface Laporan

*12. User Interface Cetak Sales Order*



**Gambar III.39. User Interface Cetak Sales Order**

*13. User Interface Cetak Jurnal*



**Gambar III.40. User Interface Cetak Jurnal**

*14. User Interface Menu Utama Owner*



**Gambar III.41. User Interface Menu Utama Owner**

*15. User Interface Owner Input Office*

ID admin	Nama Admin	Username	Password	level
1	Budijanto	budi	bud123	1
2	Elsje Kharijady	Elsje	elsje	2
3	Fitri Raharjani	jani	jani	2
4	Widi Arya	arya	arya	2

**Petugas**

ID Admin:

Nama:

Username:

Password:

Status:

Keterangan:

1 : Owner  
2 : Admin

Tambah

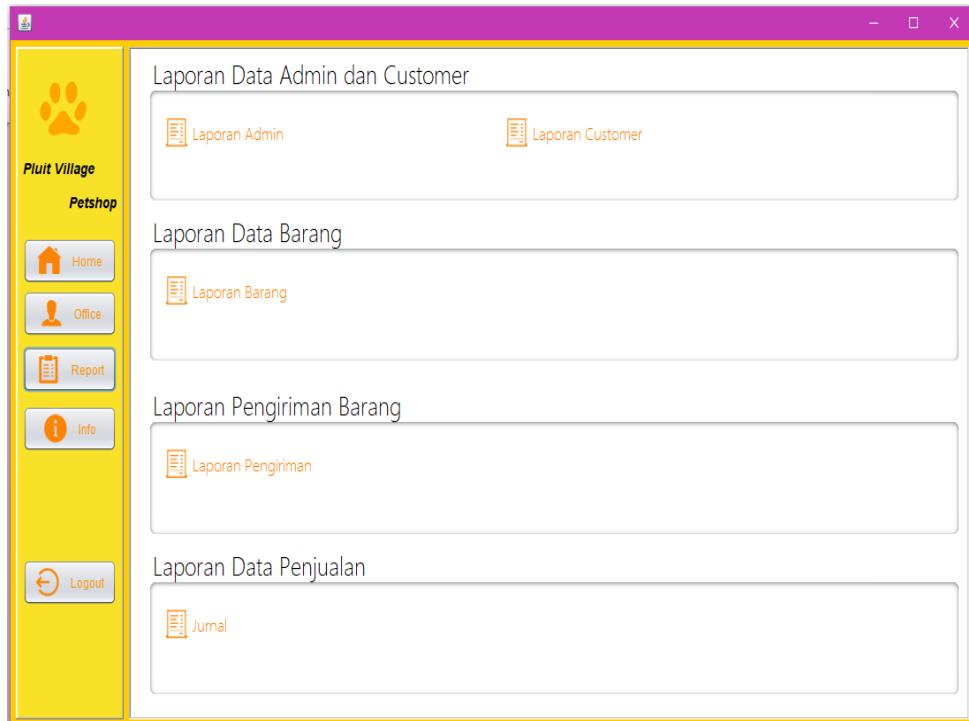
Ubah

Hapus

Bersih

**Gambar III.42. User Interface Input Office**

### 16. User Interface Report Owner



**Gambar III.43. User Interface Report Owner**

### 17. User Interface Laporan Admin

The screenshot shows the JasperViewer application window. The title bar says "JasperViewer". The main content area displays a report titled "Data Admin". The report consists of a large title "Data Admin" and a table below it. The table has columns: "id admin", "nama admin", "username", "password", and "level". The data in the table is as follows:

id admin	nama admin	username	password	level
1	Budjianto	budi	budi123	1
2	Elsje Kharjadi	Elsje	elsje	2
3	Fitri Raharjani	jani	jani	2
4	Widi Arya	arya	arya	2

**Gambar III.44. User Interface Cetak Admin**

### 18. User Interface Cetak Barang Owner

**Laporan Barang**

id barang	nama barang	jenis barang	harga barang	jumlah barang
1	Essential Oxbow	Rabbit Food	64000	100
2	Nova	Rabbit Food	40000	55
3	Briter Bunny	Rabbit Food	40000	60
4	Royal Canin	Cat Food	80000	100
5	Whiskas	Cat Food	65000	80
6	Proplan	Cat Food	68000	80
7	Royal Canin	Dog Food	90000	100
8	Pedigree	Dog Food	65000	75
9	Proplan	Dog Food	70000	60
10	Timothy Hay oxbow	Rabbit Food	85000	100
11	Alfalfa Hay Oxbow	Rabbit Food	85000	100
12	Oat Hay Oxbow	Rabbit Food	85000	100
13	Botanical Hay Oxbow	Rabbit Food	90000	100
14	Orchard Hay Oxbow	Rabbit Food	85000	100

**Gambar III.45. User Interface Cetak Barang**

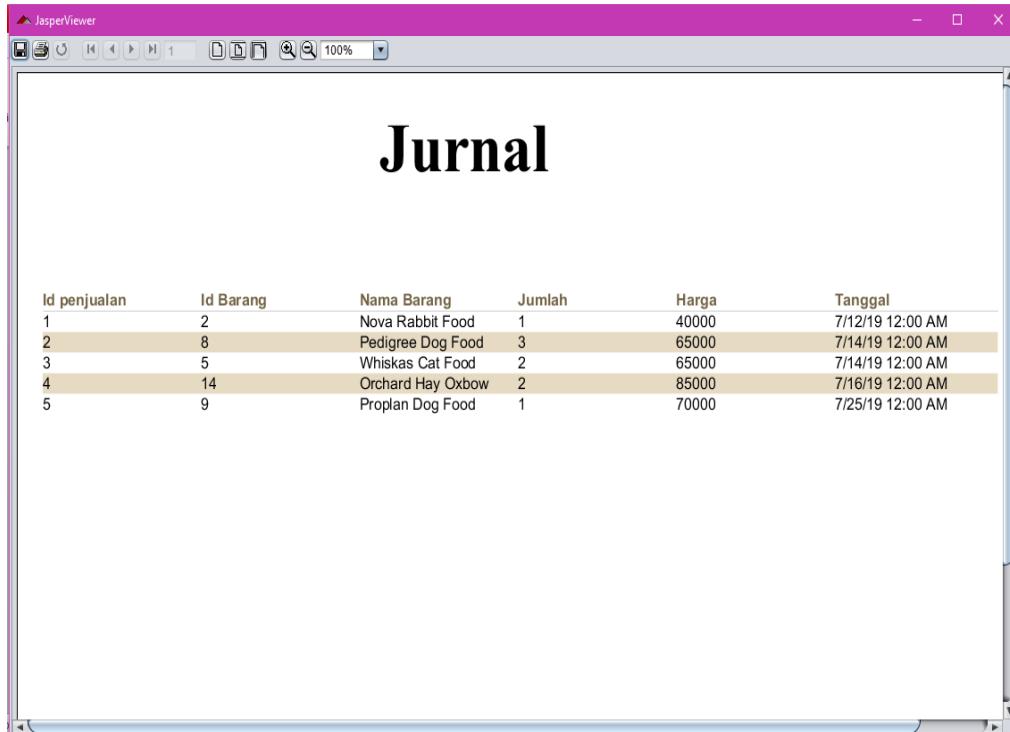
19. User Interface Cetak Pengiriman Owner

**Laporan Pengiriman**

Id Pengiriman	Nama Customer	Nama Barang	Jumlah Barang	Tanggal Pengiriman	Alamat Customer
1	Bagas	Royal Canin Catfood	2	7/28/19 12:00 AM	Pondok Ungu
2	Jason	Timothy Hay Oxbow	4	7/25/19 12:00 AM	Kemang
3	Ibrahim	Pasir Kucing 20kg	1	8/5/19 12:00 AM	Rawamangun
4	Okta	Essential Oxbow	3	8/7/19 12:00 AM	Komplek Deperla

Gambar III.46. User Interface Cetak Pengiriman

## 20. User Interface Cetak Jurnal Owner



Gambar III.47. User Interface Cetak Jurnal

### 3.5. Implementasi

#### 3.5.1. Code Generation

##### 1. Code Generation Login

```
package petshopvillage;

import javax.swing.JFrame;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
```

```
/*
 *
 * @author HP
 */
```

```
public class Login extends javax.swing.JFrame {
```

```

Connection conn;
Statement st;
ResultSet rs;
String sql;

public static String username;
public static String getUsername() {
    return username;
}
public static String password;
public static String getPassword() {
    return password;
}

/**
 * Creates new form LoginAdmin
 */
public Login() {
    initComponents();
    Koneksi DB = new Koneksi();
    DB.config();
    conn = DB.conn;
    st = DB.st;
    this.setLocationRelativeTo(null);
}

private void jLabelCloseMouseClicked(java.awt.event.MouseEvent evt) {
    System.exit(0);
}

private void jLabelMinMouseClicked(java.awt.event.MouseEvent evt) {
    this.setState(JFrame.ICONIFIED);
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        sql = "SELECT * FROM admin WHERE
username='"+jTextField1.getText()+"' AND
password='"+jPasswordField1.getText()+"'";
        rs = st.executeQuery(sql);
        if(rs.next()){
            username = rs.getString("username");
            password = rs.getString("password");
            if(rs.getString("level").equals("2")){
                new MenuAdmin().show();
            }
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}

```

```
        this.dispose();
    }else if(rs.getString("level").equals("1") ) {
        new MenuOwner().show();
        this.dispose();
    }
}else{
    JOptionPane.showMessageDialog(null,"Maaf Password atau
Username anda Salah");
    jTextField1.setText("");
    jPasswordField1.setText("");
    jTextField1.requestFocus();
}
}catch(SQLException e){
    JOptionPane.showConfirmDialog(null, e);
}
}
```

## *2. Code Generation Menu Admin*

```
package petshopvillage;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.util.HashMap;
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.view.JasperViewer;

/**
 *
 * @author HP
 */
public class MenuAdmin extends javax.swing.JFrame {
    Connection conn;
    Statement st;
    ResultSet rs;
    String sql;
```

```

private String tgl;

/*
 * Creates new form MenuAdmin
 */

public MenuAdmin() {
    initComponents();
    Koneksi DB = new Koneksi();
    DB.config();
    conn = DB.conn;
    st = DB.st;

    TampilBooking();
    TampilSending();
    TampilSales();
    TampilPetugas();
    TampilBarang();
    TampilCustomer();
    Bersih();
}

private void Bersih(){
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
    jTextField6.setText("");
    jTextField7.setText("");
    jTextArea1.setText("");
    jTextArea2.setText("");
    jTextField8.setText("");
    jTextField9.setText("");
    jTextField10.setText("");
    jTextField11.setText("");
    jTextField12.setText("");
    jTextField13.setText("");
    jTextField14.setText("");
    jTextField15.setText("");
    jTextField16.setText("");
    jTextField17.setText("");
    jTextField18.setText("");
    jTextField19.setText("");
    jTextField20.setText("");
    jTextField21.setText("");
    jTextField22.setText("");
}

```

```
jTextField24.setText("");  
jTextField25.setText("");  
jTextField26.setText("");  
jTextField27.setText("");  
jTextField28.setText("");  
jDateChooser1.setDate(null);  
jDateChooser2.setDate(null);  
jDateChooser3.setDate(null);  
}
```

```
private void TampilBooking(){
    DefaultTableModel dataTable = new DefaultTableModel();
    dataTable.addColumn("Id Pemesanan");
    dataTable.addColumn("Id Cust");
    dataTable.addColumn("Nama Customer");
    dataTable.addColumn("Id Barang");
    dataTable.addColumn("Nama Barang");
    dataTable.addColumn("Jumlah Barang");
    dataTable.addColumn("Harga Barang");
    dataTable.addColumn("Tanggal Pemesanan");
    dataTable.addColumn("Status");

    try {
        String sql = "Select * from pemesanan";
        rs = st.executeQuery(sql);
        while(rs.next()){
            dataTable.addRow(new Object[]{
                rs.getString(1),
                rs.getString(2),
                rs.getString(3),
                rs.getString(4),
                rs.getString(5),
                rs.getString(6),
                rs.getString(7),
                rs.getString(8),
                rs.getString(9)
            });
        }
        jTable4.setModel(dataTable);
    } catch(Exception e){
        System.err.print(e);
    }
}

private void TampilSending(){
    DefaultTableModel dataTable = new DefaultTableModel();
```

```

dataTable.addColumn("Id Pengiriman");
dataTable.addColumn("Nama Customer");
dataTable.addColumn("Nama Barang");
dataTable.addColumn("Jumlah Barang");
dataTable.addColumn("Tanggal Pengiriman");
dataTable.addColumn("Alamat");

try{
    String sql = "select * from pengiriman";
    rs = st.executeQuery(sql);
    while(rs.next()){
        dataTable.addRow(new Object[]{
            rs.getString("id_pengiriman"),
            rs.getString("nama_customer"),
            rs.getString("nama_barang"),
            rs.getString("jumlah_barang"),
            rs.getString("tanggal"),
            rs.getString("alamat")
        });
    }
} catch(Exception e){
    System.err.print(e);
}

private void TampilSales(){
    DefaultTableModel dataTable = new DefaultTableModel();
    dataTable.addColumn("Id Penjualan");
    dataTable.addColumn("Id Barang");
    dataTable.addColumn("Nama Barang");
    dataTable.addColumn("Jumlah");
    dataTable.addColumn("Harga");
    dataTable.addColumn("Tanggal Penjualan");

    try{
        String sql = "select * from penjualan";
        rs = st.executeQuery(sql);
        while(rs.next()){
            dataTable.addRow(new Object[]{
                rs.getString("id_penjualan"),
                rs.getString("id_barang"),
                rs.getString("nama_barang"),
                rs.getString("jumlah"),
                rs.getString("harga"),
                rs.getString("tanggal")
            });
        }
    }jTable5.setModel(dataTable);
} catch(Exception e){
}
}

```

```

        System.err.print(e);
    }
}

private void TampilPetugas(){
    DefaultTableModel dataTable = new DefaultTableModel();
    dataTable.addColumn("ID admin");
    dataTable.addColumn("Nama Admin");
    dataTable.addColumn("Username");

    try{
        String sql = "Select * from admin";
        rs = st.executeQuery(sql);
        while(rs.next()){
            dataTable.addRow(new Object[]{
                rs.getString(1),
                rs.getString(2),
                rs.getString(3),
            });
        }
        jTable1.setModel(dataTable);
    }catch(Exception e){
        System.err.print(e);
    }
}

private void TampilBarang(){
    DefaultTableModel dataTable = new DefaultTableModel();
    dataTable.addColumn("ID Barang");
    dataTable.addColumn("Nama Barang");
    dataTable.addColumn("Jenis Barang");
    dataTable.addColumn("Harga Barang");
    dataTable.addColumn("Jumlah Barang");

    try{
        String sql = "Select * from barang";
        rs = st.executeQuery(sql);
        while(rs.next()){
            dataTable.addRow(new Object[]{
                rs.getString(1),
                rs.getString(2),
                rs.getString(3),
                rs.getString(4),
                rs.getString(5)
            });
        }
        jTable2.setModel(dataTable);
    }catch(Exception e){
        System.err.print(e);
    }
}

```

```
}
```

```
private void TampilCustomer(){
    DefaultTableModel dataTable = new DefaultTableModel();
    dataTable.addColumn("ID Customer");
    dataTable.addColumn("Nama Customer");
    dataTable.addColumn("Alamat");
    dataTable.addColumn("No Telepon");

    try{
        String sql = "select * from customer";
        rs = st.executeQuery(sql);
        while(rs.next()){
            dataTable.addRow(new Object[]{
                rs.getString("id_cust"),
                rs.getString("nama_cust"),
                rs.getString("alamat"),
                rs.getString("no_telp")
            });
        }
    }catch(Exception e){
        System.err.print(e);
    }
}

private void TampilSales(){
    DefaultTableModel dataTable = new DefaultTableModel();
    dataTable.addColumn("Id Penjualan");
    dataTable.addColumn("Id Barang");
    dataTable.addColumn("Nama Barang");
    dataTable.addColumn("Jumlah");
    dataTable.addColumn("Harga");
    dataTable.addColumn("Tanggal Penjualan");

    try{
        String sql = "select * from penjualan";
        rs = st.executeQuery(sql);
        while(rs.next()){
            dataTable.addRow(new Object[]{
                rs.getString("id_penjualan"),
                rs.getString("id_barang"),
                rs.getString("nama_barang"),
                rs.getString("jumlah"),
                rs.getString("harga"),
                rs.getString("tanggal")
            });
        }
    }catch(Exception e){
```

```

        System.err.print(e);
    }
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt)
{
    int app;
    if((app = JOptionPane.showConfirmDialog(null,"Ingin  
Menambahkan Data ?  
"+ "?","Perhatian",JOptionPane.YES_NO_OPTION))==0);
    try{
        String sql = " insert into barang values (null,  
"+jTextField6.getText()+"','"+jTextField7.getText()+"','"+  
+jTextField8.getText()+"','"++jTextField9.getText()+"');  
st.executeUpdate(sql);  
JOptionPane.showMessageDialog(null,"Data Berhasil di  
tambah");  
Bersih();  
TampilBarang();  
}catch(SQLException e){  
    System.err.print(e);
}
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt)
{
    int app;
    if((app = JOptionPane.showConfirmDialog(null,"Ingin Mengubah  
Data ? "+ "?","Perhatian",JOptionPane.YES_NO_OPTION))==0);
    try{
        String sql = "update barang set"  

        + " id_barang = "+ jTextField5.getText()+" , "  

        + " nama_barang = "+ jTextField6.getText()+" , "  

        + " jenis_barang = "+ jTextField7.getText()+" , "  

        + " harga_barang = "+ jTextField8.getText()+" , "  

        + " jumlah_barang = "+ jTextField9.getText()+" where"  

        + " id_barang = "+ jTextField5.getText()+" "";  

        st.executeUpdate(sql);  

JOptionPane.showMessageDialog(null,"Data Berhasil di Ubah");  

Bersih();  

TampilBarang();  
}catch(SQLException e){  

    JOptionPane.showMessageDialog(null,"Data Tidak dapat di  
hapus !");
}
}

```

```

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
    int app;
    if((app = JOptionPane.showConfirmDialog(null, "Ingin Menghapus Data ? "+"?", "Perhatian", JOptionPane.YES_NO_OPTION))==0);
    try{
        String sql = "Delete from barang where id_barang =
        "+jTextField5.getText()+"";
        st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data Berhasildi Hapus");
        Bersih();
        TampilBarang();
    }catch(SQLException e){
        JOptionPane.showMessageDialog(null, "Data Tidak dapat di hapus !");
    }
}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
    Bersih();
}

private void jTable2MouseClicked(java.awt.event.MouseEvent evt) {
    int table = jTable2.getSelectedRow();
    jTextField5.setText(jTable2.getValueAt(table, 0).toString());
    jTextField6.setText(jTable2.getValueAt(table, 1).toString());
    jTextField7.setText(jTable2.getValueAt(table, 2).toString());
    jTextField8.setText(jTable2.getValueAt(table, 3).toString());
    jTextField9.setText(jTable2.getValueAt(table, 4).toString());
}

private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
    SimpleDateFormat dFormat = new SimpleDateFormat("yyyy-MM-dd");
    int app;
    if((app = JOptionPane.showConfirmDialog(null, "Ingin Update Data ? "+"?", "Perhatian", JOptionPane.YES_NO_OPTION))==0);
    try{
        String sql = "update pemesanan set"
        + " id_pemesanan = "+jTextField13.getText()+" , "

```

```
+ " id_cust = "+jTextField16.getText()+" , "
+ " nama_customer = "+jTextField3.getText()+" , "
+ " id_barang = "+jTextField17.getText()+" , "
+ " nama_barang = "+jTextField10.getText()+" , "
+ " jumlah_barang = "+jTextField11.getText()+" , "
+ " harga_barang = "+jTextField12.getText()+" , "
+ " tanggal_pemesanan =
"+dFormat.format(jDateChooser2.getDate())+", "
+ " status = "+jComboBox1.getSelectedItem()+" where"
+ " id_pemesanan = "+jTextField13.getText()+"";
st.executeUpdate(sql);
JOptionPane.showMessageDialog(null,"Data Berhasil di
Update");
Bersih();
TampilBooking();
}catch(SQLException e){
JOptionPane.showMessageDialog(null,"Data Tidak dapat di
hapus !");
}
}
```

### *3. Code Generation Menu Owner*

```
package petshopvillage;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.util.HashMap;

import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.view.JasperViewer;
/***
 * Creates new form MenuOwner
 */

```

```
public class MenuOwner extends javax.swing.JFrame {  
    Connection conn;  
    Statement st;  
    ResultSet rs;  
    String sql;
```

```

        public MenuOwner() {
    initComponents();

    Koneksi DB = new Koneksi();
    DB.config();
    conn = DB.conn;
    st = DB.st;
    TampilPetugas();
    Bersih();
}

private void Bersih(){
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jPasswordField1.setText("");
    jTextField4.setText("");
    jTextField1.requestFocus();
}

private void TampilPetugas(){
    DefaultTableModel dataTable = new DefaultTableModel();
    dataTable.addColumn( "ID admin" );
    dataTable.addColumn( "Nama Admin" );
    dataTable.addColumn( "Username" );
    dataTable.addColumn( "Password" );
    dataTable.addColumn( "level" );

    try{
        String sql = "Select * from admin";
        rs = st.executeQuery(sql);
        while(rs.next()){
            dataTable.addRow(new Object[]{
                rs.getString(1),
                rs.getString(2),
                rs.getString(3),
                rs.getString(4),
                rs.getString(5)
            });
        }
        jTable1.setModel(dataTable);
    }catch(Exception e){
        System.err.print(e);
    }
}
@SuppressWarnings("unchecked")

private void initComponents() {

    bodyPanel = new javax.swing.JPanel();
}

```

```

MenuPanel = new javax.swing.JPanel();
btnHome = new javax.swing.JButton();
btnReport = new javax.swing.JButton();
btnInfo = new javax.swing.JButton();
btnLogout = new javax.swing.JButton();
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
btnpetugas = new javax.swing.JButton();
MainPanel = new javax.swing.JPanel();
HomePanel = new javax.swing.JPanel();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
PetugasPanel = new javax.swing.JPanel();
jTextField2 = new javax.swing.JTextField();
jTextField4 = new javax.swing.JTextField();
jTextField1 = new javax.swing.JTextField();
jPasswordField1 = new javax.swing.JPasswordField();
jLabel32 = new javax.swing.JLabel();
jLabel33 = new javax.swing.JLabel();
jLabel34 = new javax.swing.JLabel();
jLabel35 = new javax.swing.JLabel();

private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //remove panel
    MainPanel.removeAll();
    MainPanel.repaint();
    MainPanel.revalidate();

    //setelah itu add Panel
    MainPanel.add(HomePanel);
    MainPanel.repaint();
    MainPanel.revalidate();
}

private void btnReportActionPerformed(java.awt.event.ActionEvent evt) {
    MainPanel.removeAll();
    MainPanel.repaint();
    MainPanel.revalidate();

    //setelah itu add Panel
    MainPanel.add(Report);
    MainPanel.repaint();
}

```

```

MainPanel.revalidate();
//remove panel

    // TODO add your handling code here:
}

private void btnInfoActionPerformed(java.awt.event.ActionEvent evt) {

    AboutView n = new AboutView();
    n.setVisible(true);
    // TODO add your handling code here:
}

private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
    // TODO add your handling code here:
}

private void
btnpetugasActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //remove panel
    MainPanel.removeAll();
    MainPanel.repaint();
    MainPanel.revalidate();

    //setelah itu add Panel
    MainPanel.add(PetugasPanel);
    MainPanel.repaint();
    MainPanel.revalidate();
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    int app;
    if((app = JOptionPane.showConfirmDialog(null, "Ingin
Menambahkan Data?" + "?",
"Perhatian", JOptionPane.YES_NO_OPTION))==0);
    try{

        String sql = " insert into admin values (null,
" +jTextField1.getText() + "','" +jTextField2.getText() + "','" +
+jPasswordField1.getText() + "','" +jTextField3.getText() + ")";
        st.executeUpdate(sql);
    }
}

```

```

        JOptionPane.showMessageDialog(null,"Data Berhasil di
tambah");
        Bersih();
        TampilPetugas();
    }catch(SQLException e){
        System.err.print(e);
    }
    // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent
evt) {
    int app;
    if((app = JOptionPane.showConfirmDialog(null,"Ingin Mengubah
Data ? "+ "?", "Perhatian", JOptionPane.YES_NO_OPTION))==0);
    try{
        String sql = "update admin set"
        + " id_admin = "+jTextField4.getText()+" , "
        + " nama_admin = "+jTextField1.getText()+" , "
        + " username = "+jTextField2.getText()+" , "
        + " password = "+jPasswordField1.getText()+" , "
        + " level = "+jTextField3.getText()+" where"
        + " id_admin = "+jTextField4.getText()+" ";
        st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null, "Data Berhasil di Ubah"
);
        Bersih();
        TampilPetugas();
    }catch(SQLException e){
        JOptionPane.showMessageDialog(null, "Data Tidak dapat di
hapus !");
    }
    // TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent
evt) {
    int app;
    if((app = JOptionPane.showConfirmDialog(null, "Ingin
Menghapus Data? "+ "?", "Perhatian",
JOptionPane.YES_NO_OPTION))==0);
    try{
        String sql = "Delete from admin where id_admin =
"+jTextField4.getText()+"";
        st.executeUpdate(sql);
    }
}

```

```

        JOptionPane.showMessageDialog(null, "Data Berhasil di
Hapus");
        Bersih();
        TampilPetugas();
    }catch(SQLException e){
        JOptionPane.showMessageDialog(null, "Data Tidak dapat di
hapus !");
    }
    // TODO add your handling code here:
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    Bersih();
}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int table = jTable1.getSelectedRow();
    jTextField4.setText(jTable1.getValueAt(table, 0).toString());
    jTextField1.setText(jTable1.getValueAt(table, 1).toString());
    jTextField2.setText(jTable1.getValueAt(table, 2).toString());
    jPasswordField1.setText(jTable1.getValueAt(table, 3).toString());
    jTextField3.setText(jTable1.getValueAt(table, 4).toString());
}

private void jLabel20MouseClicked(java.awt.event.MouseEvent evt)
{
    try{
        String NamaFile = "barangg.jasper";
        Connection koneksi =
DriverManager.getConnection("jdbc:mysql://localhost:3306/village","r
oot","");
        HashMap param = new HashMap();

        JasperPrint Jprint = JasperFillManager.fillReport(NamaFile ,
param , koneksi );
        JasperViewer.viewReport(Jprint);
    } catch (SQLException | JRException ex){
        System.out.println(ex);
    }
}

private void jLabel21MouseClicked(java.awt.event.MouseEvent evt)
{
    try{

```

```

        String NamaFile = "customerr.jasper";
        Connection koneksi =
DriverManager.getConnection("jdbc:mysql://localhost:3306/village","r
oot","");
        HashMap param = new HashMap();

        JasperPrint Jprint = JasperFillManager.fillReport(NamaFile ,
param ,koneksi );
        JasperViewer.viewReport(Jprint);
    }catch(SQLException | JRException ex){
        System.out.println(ex);
    }
}

private void jLabel25MouseClicked(java.awt.event.MouseEvent evt)
{
try{
    String NamaFile = "itemm.jasper";
    Connection koneksi =
DriverManager.getConnection("jdbc:mysql://localhost:3306/village","r
oot","");
    HashMap param = new HashMap();

    JasperPrint Jprint = JasperFillManager.fillReport(NamaFile ,
param ,koneksi );
    JasperViewer.viewReport(Jprint);
}catch(SQLException | JRException ex){
    System.out.println(ex);
}
// TODO add your handling code here:
}

private void jLabel26MouseClicked(java.awt.event.MouseEvent evt)
{
try{
    String NamaFile = "Jurnal.jasper";
    Connection koneksi =
DriverManager.getConnection("jdbc:mysql://localhost:3306/village","r
oot","");
    HashMap param = new HashMap();

    JasperPrint Jprint = JasperFillManager.fillReport(NamaFile ,
param ,koneksi );
    JasperViewer.viewReport(Jprint);
}catch(SQLException | JRException ex){
    System.out.println(ex);
}
}

```

```

private void jLabel27MouseClicked(java.awt.event.MouseEvent evt) {
    try{
        String NamaFile = "Pengiriman.jasper";
        Connection koneksi =
DriverManager.getConnection("jdbc:mysql://localhost:3306/village","root","");
        HashMap param = new HashMap();

        JasperPrint Jprint = JasperFillManager.fillReport(NamaFile ,
param ,koneksi );
        JasperViewer.viewReport(Jprint);
    }catch(SQLException | JRException ex){
        System.out.println(ex);
    }
    // TODO add your handling code here:
}

/*
 * @param args the command line arguments
 */

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
     * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MenuOwner.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

```

```

java.util.logging.Logger.getLogger(MenuOwner.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MenuOwner.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MenuOwner.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new MenuOwner().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JPanel HomePanel;
private javax.swing.JPanel MainPanel;
private javax.swing.JPanel MenuPanel;
private javax.swing.JPanel PetugasPanel;
private javax.swing.JPanel Report;
private javax.swing.JPanel bodyPanel;
private javax.swing.JButton btnHome;
private javax.swing.JButton btnInfo;
private javax.swing.JButton btnLogout;
private javax.swing.JButton btnReport;
private javax.swing.JButton btnpetugas;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JPanel jKet;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel27;

```

```

private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel32;
private javax.swing.JLabel jLabel33;
private javax.swing.JLabel jLabel34;
private javax.swing.JLabel jLabel35;
private javax.swing.JLabel jLabel36;
private javax.swing.JLabel jLabel37;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JPanel jPanel12;
private javax.swing.JPanel jPanel5;
private javax.swing.JPanel jPanel6;
private javax.swing.JPanel jPanel7;
private javax.swing.JPanel jPanel8;
private javax.swing.JPasswordField jPasswordField1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
// End of variables declaration
}

```

### 3.5.2. Blackbox Testing

#### 1. Form Login Admin

**Tabel III.9.**  
**Hasil Black Box Testing Form Login**

No.	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	<i>Username</i> tidak diinput	Username: Kosong	Sistem akan menolak akses	Sesuai Harapan	Valid
2.	Mengimput username dan tidak <i>input password</i>	Username: (jani) Password: (kosong)	Sistem tidak bisa verifikasi dan tombol login tidak aktif	Sesuai Harapan	Valid

3.	Menginput <i>username</i> dan <i>password</i> sesuai kemudian <i>login</i>	Kode User: (jani) Password: (1985)	Sistem akan memverifikasi dan tampil menu utama	Sesuai Harapan	Valid
----	--	---------------------------------------	---	----------------	-------

## 2. Form Master Data Customer

**Tabel III.10.**

**Hasil Black Box Testing Form Customer**

No.	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Id <i>Customer</i> , Nama <i>Customer</i> , Alamat, dan No Telp tidak di input	Id_cust: (kosong) Nama Customer: (kosong) Alamat: (kosong) No.Telp: (kosong)	Sistem akan menolak akses	Sesuai Harapan	Valid
2.	Menginput Id <i>Customer</i> saja, Nama <i>Customer</i> , Alamat, dan No Telp di kosongkan	Nama Pelanggan: Eka Alamat (kosong) No. Telp: (kososng)	Sistem akan menolak akses dan menampilkan pesan “Ingin menambahkan data?”	Sesuai Harapan	Valid

3.	Menginput Alamat dan Nama Pelanggan, No. Telp di kosongkan	Nama Pelanggan: (kosong) Alamat: Muara Karang No. Telp: (kosong)	Sistem akan menolak akses dan menampilkan pesan “Ingin menambahkan data?”	Sesuai Harapan	Valid
4.	Meninput No.Telp dan Nama Pelanggan, Alamat di kosongkan	Nama Pelanggan: (kosong) Alamat: (kosong) No. Telp: 085682801919	Sistem akan menolak akses dan menampilkan pesan “Ingin menambahkan data?”	Sesuai Harapan	Valid
5.	Menginput Nama Pelanggan, Alamat dan No. Telp	Nama Pelanggan: Bernard Alamat: Muara Karang No. Telp: 085682801919	Sistem akan menerima akses dan menampilkan pesan “Data Customer Berhasil Disimpan”	Sesuai Harapan	Valid

### 3. Form Transaksi Data Pengiriman

**Tabel III.11.**

#### **Hasil Black Box Testing Transaksi Pengiriman**

No.	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Id pengiriman, Id customer, Nama customer, Nama barang jumlah barang, alamat tidak dinput	Kode Penjualan (kosong) Kode pesanan: (kosong) Nama pelanggan: (kosong)	Sistem akan menolak akses dan menampilkan pesan “Ingin menambahkan data?”	Sesuai harapan	Valid

		Nama barang: (kosong) Tanggal: (kosong) Jumlah qty: (kosong) Harga: (kosong)			
2.	Menginput nama customer dan nama barang, jumlah barang, alamat tidak diinput	Nama customer: Bernard Nama Barang: (kosong) Jumlah barang; (kosong) Alamat (kosong)	Sistem akan menolak akses dan menampilkan pesan “Ingin membahakan data?”	Sesuai harapan	Valid
3.	Menginput Nama barang dan nama pelanggan, jumlah barang, dan alamat tidak diinput	Nama customer: (kosong) Nama barang: Catfood wishkas Jumlah barang: (kosong) Alamat: (Kosong)	Sistem akan menolak akses dan menampilkan pesan “Ingin menambahkan data?”	Sesuai harapan	Valid
4.	Menginput alamat saja, nama customer, nama barang, jumlah barang, tidak diinput	Nama customer: (kosong) Nama barang (kosong) Jumlah barang (kosong) Alamat: Muara Karang	Sistem akan menolak akses dan menampilkan pesan “Ingin menambahkan data?”	Sesuai harapan	Valid
5.	Menginput Nama	Nama Pelanggan:	Sistem akan menerima	Sesuai harapan	Valid

	Pelanggan, Nama barang, Tanggal, Jumlah qty , harga dengan benar, kemudian klik Tambah	Hanson Nama Barang: Oxbow Hay Alfalfa Tanggal: 08 agustus 2019 Jumlah qty: 15 Harga: Rp. 85000	akses dan menampilkan pesan “ <i>Data Pemesanan Berhasil Disimpan</i> ”		
--	---	---	---	--	--



### 3.5.3. Spesifikasi Hardware dan Software

Dalam bahasan mengenai penggunaan perangkat lunak (*software*) dan perangkat keras (*hardware*) yang akan digunakan dalam sistem usulan.

*Hardware* yang dimaksud adalah suatu peralatan fisik yang terdapat pada komputer. Untuk menunjang program aplikasi yang dibuat *hardware* yang dibutuhkan untuk menerapkan komputerisasi sistem yang diusulkan harus mempunyai tingkat kecepatan yang tinggi, kapasitas yang besar, dan daya tahan yang kuat.

Sedangkan Perangkat lunak (*software*) adalah program yang berisi perintah-perintah untuk melakukan pengolahan data. Adapun spesifikasi *hardware* dan *software* untuk perancangan sistem ini yaitu:

**Tabel III.12. Spesifikasi Hardware dan Software**

Kebutuhan	Keterangan
Sistem Operasi	Microsoft Windows 10
Processor	Intel(R) Celeron(R) CPU B830 @1.80GHz
RAM	2.00 GB
Harddisk	500 GB
Monitor	14' inch
Keyboard	Standar
Printer	Canon
Mouse	Standard
Software	Bahasa Pemograman : NetBeans IDE 8.1 DBMS : MySQL Anti virus : Avast Free Antivirus