

BAB III

PEMBAHASAN

3.1. Tinjauan Perusahaan

Tinjauan perusahaan dibagi menjadi sejarah perusahaan, visi, misi dan struktur organisasi perusahaan serta fungsinya. Perusahaan yang diteliti adalah UD Barokah Pamanukan, yang bergerak dibidang jasa pemotongan kayu. Kegiatan utama yang dilakukan UD Barokah Pamanukan adalah melakukan penjualan jasa pemotongan kayu. Berikut ini hal-hal yang akan diuraikan pada tinjauan perusahaan adalah sejarah berdirinya UD Barokah Pamanukan, visi dan misi, struktur organisasi serta deskripsi jabatan di UD Barokah Pamanukan.

3.1.1. Sejarah Perusahaan

UD Barokah Pamanukan adalah suatu bentuk wirausaha milik perseorangan yang menyediakan produk jasa pemotongan kayu. UD Barokah Pamanukan berlokasi di Pamanukan Jln Ion Martasasmita Desa Rancaudik RT.09 RW.02 Kabupaten Subang, didirikan oleh Alm bapak Hj Jamaludin pada tahun 1986.

Pada awal berdirinya perusahaan ini belum menggunakan gergaji mesin, tetapi masih menggunakan gergaji manual, dan kayunya tersebut awalnya dikirim dari gunung Cikeuyeup Subang berupa kayu olahan setengah jadi. Akhir tahun 1996 beralih dari yang tadinya menggunakan gergaji manual menjadi mesin gergaji *circle*, dan kayunya dikirim dari Banten berupa kayu gelondongan atau kayu yang belum diolah. Tahun 1998 perusahaan UD Barokah Pamanukan menjadi *supplier* kayu pada matrial-matrial kecil.

Pada tahun 2003 tepatnya tanggal 16 April bapak Hj Jamaludin wafat lalu perusahaan ini dikelola oleh anaknya bernama ibu Sumarni, dan pada tahun tersebut juga dilakukan pergantian alat penggergajian yang tadinya menggunakan mesin gergaji *circle* menjadi mesin gergaji pita sampai sekarang. Saat ini UD Barokah Pamanukan memperkerjakan 6 orang karyawan. Adapun visi dan misi yang ada pada UD Barokah Pamanukan sebagai berikut:

Visi

Menjadikan perusahaan terbaik dalam jasa pemotongan kayu di Kabupaten Subang.

Misi

1. Menjual harga yang kompetitif
2. Proses layanan yang cepat
3. Kualitas pemotongan yang presisi

3.1.2. Struktur Organisasi dan Fungsi

Struktur organisasi merupakan pola hubungan antara wewenang untuk memerintah dan pengambilan keputusan serta tanggung jawab dalam melaksanakan tugas yang dibebankan. Adapun struktur organisasi dan fungsi yang terdapat di UD Barokah Pamnaukan adalah sebagai berikut:



Sumber: UD Barokah Pamanukan

Gambar III.1.

Struktur Organisasi

Berikut ini adalah fungsi uraian tugas dan tanggung jawab dari masing-masing bagian pada UD Barokah Pamanukan.

1. Pemilik

- a. Memimpin dan mengawasi perkembangan perusahaan.
- b. Mengatur kebijakan perusahaan.
- c. Melihat kondisi keuangan perusahaan.
- d. Menggerakan bawahan
- e. Penanggung jawab perusahaan

2. Pengelola

- a. Mengelola keuangan perusahaan.
- b. Menjalin komunikasi yang baik dengan *costumer*.
- c. Bertanggung jawab terhadap aktifitas sehari-hari.
- d. Membuat laporan keuangan bulanan

3. Bagian Pemotongan

- a. Melakukan pemotongan kayu memakai mesin gergaji.
- b. Bertanggung jawab terhadap hasil pemotongan.
- c. Menyusun kayu hasil olahan.

4. Kuli Panggul

- a. Memuat kayu ke mobil.
- b. Melakukan bongkar kayu.
- c. Ikut pengiriman kayu

5. Bagian Pengiriman

- a. Mengirim kayu menggunakan mobil kepada konsumen yang memesan.
- b. Pengecekan kondisi mobil.
- c. Bertanggung jawab terhadap pengiriman kayu.

3.2. Tinjauan Kasus

Perkembangan teknologi saat ini sangat memberikan dukungan bagi kemajuan sistem pengolahan data perusahaan, terutama pemanfaatan teknologi berbasis komputer. Penggunaan komputer dalam dunia usaha sangat membantu dalam pengolahan data, sehingga penyajian laporan dapat dilakukan dengan cepat, tepat, dan juga akan menghemat waktu dalam prosesnya.

Dengan adanya komputer diantaranya mempermudah dalam pembuatan laporan pendapatan, sehingga pemilik perusahaan dapat melihat berapa keuntungan yang didapat dan juga pemilik perusahaan dapat mengambil keputusan yang bijak. Sampai saat ini pada UD Barokah Pamanukan dalam pengelolaan pencatatan laporan keuangannya masih dilakukan secara manual oleh pengelola perusahaan.

3.2.1. Proses Bisnis Sistem Berjalan

Proses sistem berjalan pendapatan jasa pada UD Barokah Pamanukan adalah Konsumen datang ke UD Barokah Pamanukan lalu datang ke bagian pengelola untuk melakukan pengajuan pemotongan kayu, bagian pengelola meminta no telpon dan alamat konsumen untuk dicatat dibuku data konsumen.

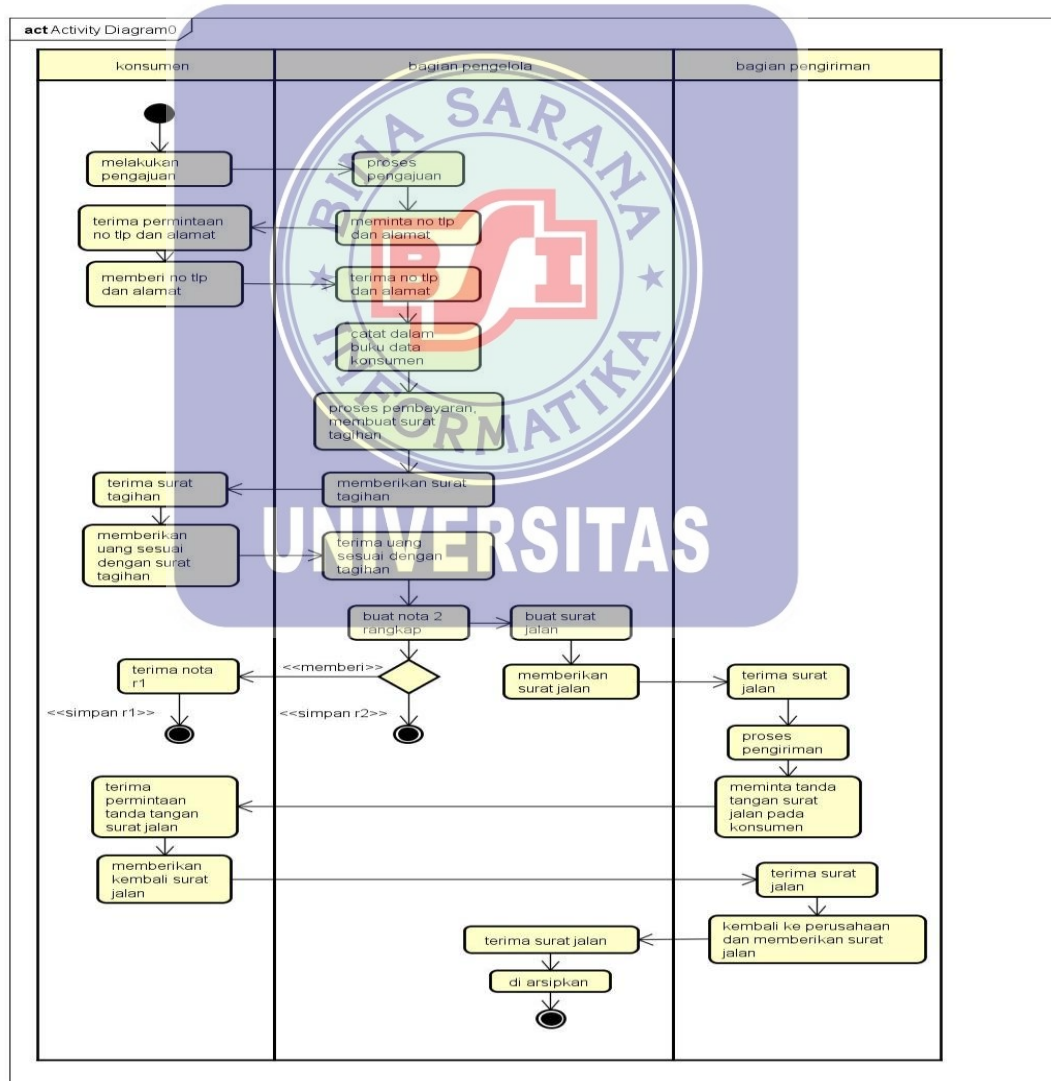
Proses pembayaran, bagian pengelola menjumlahkan total biaya yang harus di bayar dan pelanggan memberikan sejumlah uang sesuai dengan total biaya tadi. Setelah selesai pembayaran bagian pengelola membuat nota dua rangkap, rangkap pertama di berikan kepada konsumen sebagai bukti transaksi dan rangkap kedua disimpan sebagai arsip.

Bagian pengelola membuat surat jalan dan diberikan kepada bagian pengiriman agar segera melakukan proses pengiriman, bagian pengiriman meminta tanda tangan ke konsumen sebagai bukti telah melakukan pengiriman. Lalu bagian

pengiriman memberikan kembali surat jalan kepada bagian pengelola untuk di arsipkan.

3.2.2. Activity Diagram

Diagram Aktivitas atau *Activity Diagram* menggambarkan *workflow* (aliran kas) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah *Activity Diagram* sistem berjalan yang ada pada UD Barokah Pamanukan:



Gambar III.2.

Activity Diagram Berjalan

3.2.3. Dokumen masukan

Dokumen masukan merupakan proses awal dari kegiatan pada proses transaksi dalam UD Barokah Pamanukan. Adapun bentuk dokumen masukan adalah sebagai berikut.

- | | |
|-----------------|------------------------------------|
| 1. Nama Dokumen | : Data Konsumen |
| Fungsi | : Identitas Konsumen |
| Sumber | : Pengelola |
| Tujuan | : Pengelola |
| Jumlah | : Satu Lembar |
| Frekuensi | : Setiap Pengajuan Pemotongan Kayu |
| Media | : Kertas |
| Bentuk | : Lampiran A.1 |
| 2. Nama Dokumen | : Nota Rangkap 1 |
| Fungsi | : Bukti Pembayaran |
| Sumber | : Pengelola |
| Tujuan | : Pengelola |
| Jumlah | : Satu Lembar |
| Frekuensi | : Setiap Transaksi Pembayaran |
| Media | : Kertas |
| Bentuk | : Lampiran A.2 |

3.2.4. Dokumen Keluaran

Dokumen keluaran merupakan dokumen dari hasil proses kegiatan transaksi pada UD Barokah Pamanukan, Adapun bentuk dokumen keluaran adalah sebagai berikut.

1. Nama Dokumen : Nota Rangkap 2
- Fungsi : Bukti Pembayaran
- Sumber : Pengelola
- Tujuan : Konsumen
- Jumlah : Satu Lembar
- Frekuensi : Setiap Transaksi Pembayaran
- Media : Kertas
- Bentuk : Lampiran B.1

2. Nama Dokumen : Surat Jalan
- Fungsi : Bukti Pembayaran
- Sumber : Pengelola
- Tujuan : Bagian Pengiriman
- Jumlah : Satu Lembar
- Frekuensi : Setiap Akan Mengirim
- Media : Kertas
- Bentuk : Lampiran B.1



3.2.5. Permasalahan Pokok

Permasalahan yang ada pada sistem pendapatan jasa pada UD Barokah Pamanukan menurut hasil analisa dari penulis, terlihat bahwa permasalahan yang dihadapi adalah:

1. Belum memiliki dokumen transaksi, sehingga pihak perusahaan tidak bisa melihat transaksi apa saja yang telah terjadi.
2. Belum memiliki catatan pendapatan, sehingga pemilik perusahaan tidak bisa melihat berapa total pendapatan yang didapat tiap bulannya.

3. Belum memiliki laporan keuangan, sehingga pihak perusahaan kesulitan dalam mencari data keuangan.

3.2.6. Pemecahan Masalah

Melihat gambaran umum permasalahan yang dihadapi oleh UD Barokah Pamanukan, penulis menyarankan bahwa perlu adanya rancang bangun sistem informasi akuntansi yang dapat membantu pencatatan laporan keuangan khususnya bulanan, agar lebih terperinci dan cepat dalam proses pembuatan laporannya.

3.3. Analisa Kebutuhan Software

Berdasarkan proses bisnis pada UD Barokah Pamanukan maka tahapan berikutnya adalah Pengelola dan Pemilik sebagai *admin*, pengelola bisa mengakses data transaksi, data konsumen, data admin dan membuat laporan. Sedangkan pemilik hanya bisa melihat data konsumen, data transaksi dan akses *form admin*.

3.3.1. Analisa Kebutuhan

1. Pengelola

- A1. Pengelola *Login*
- A2. Pengelola Mengelola Data Kayu
- A3. Pengelola Mengelola Data Konsumen
- A4. Pengelola Mengelola Transaksi
- A5. Pengelola Cetak Laporan

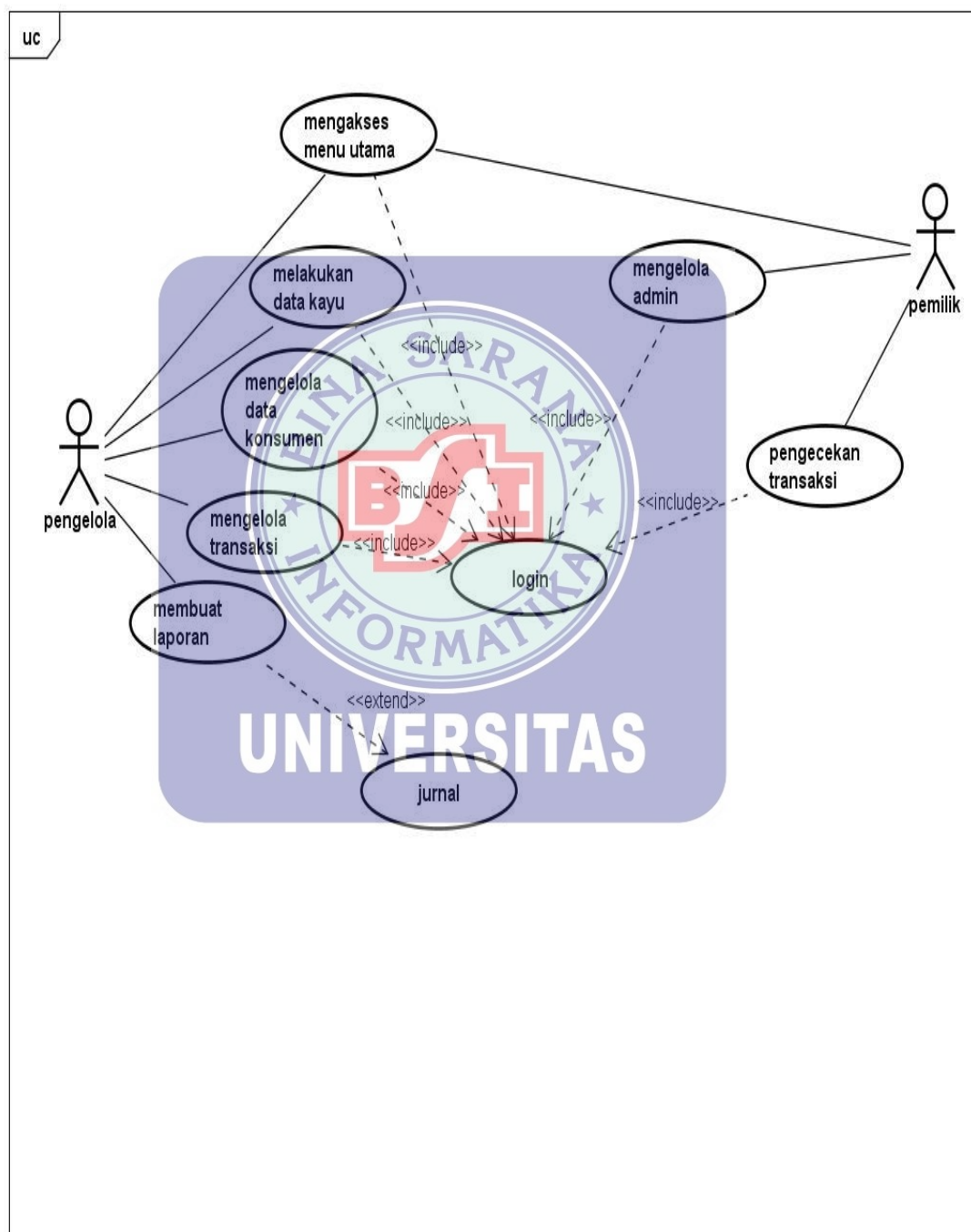
2. Pemilik

- B1. Pemilik *Login*
- B2. Pemilik Mengelola Admin

B3. Pemilik Melihat Data Konsumen

B4. Pemilik Melihat Transaksi

3.3.2. Use Case Diagram

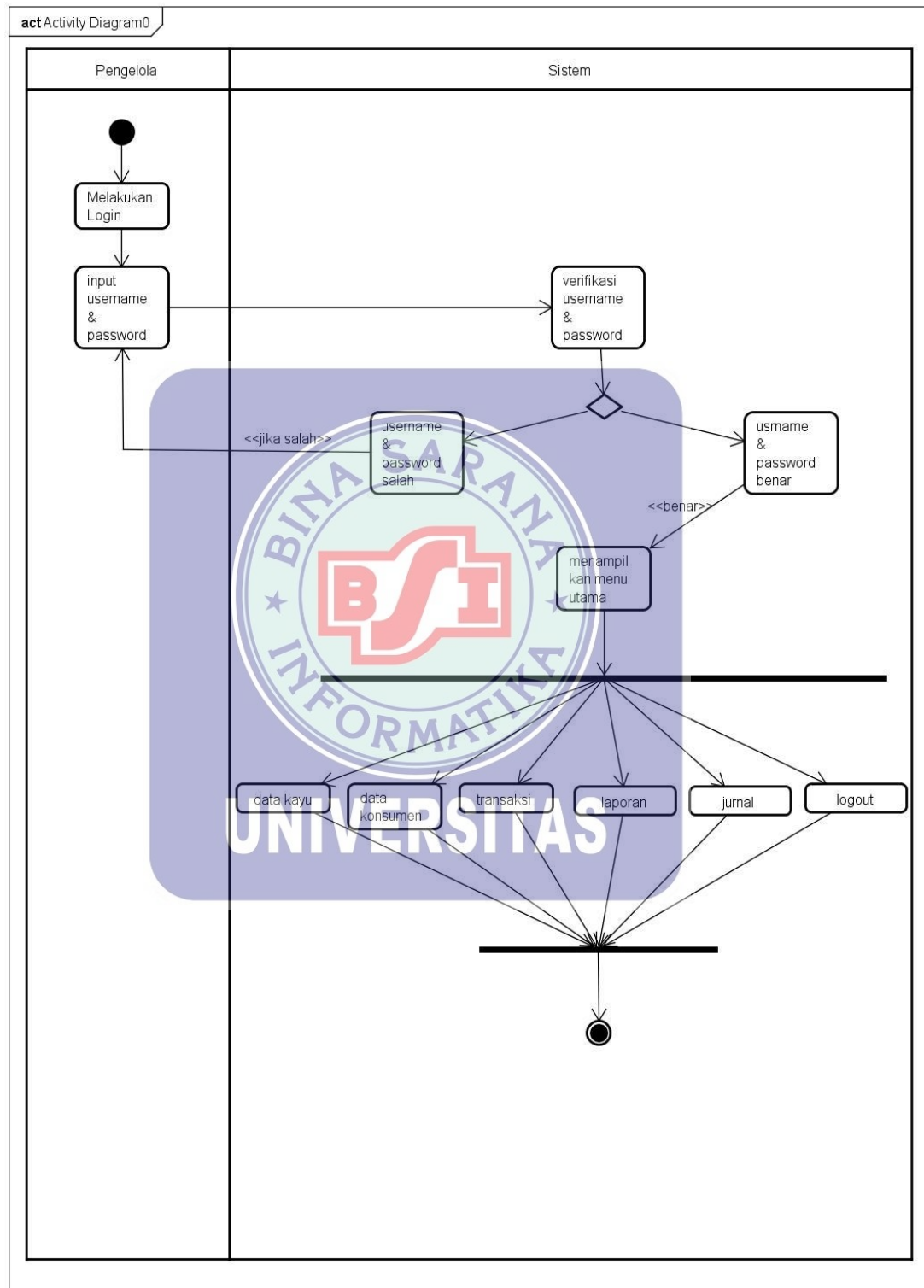


Gambar III.3.

Usecase Diagram Sistem Usulan

3.3.3. Activity Diagram

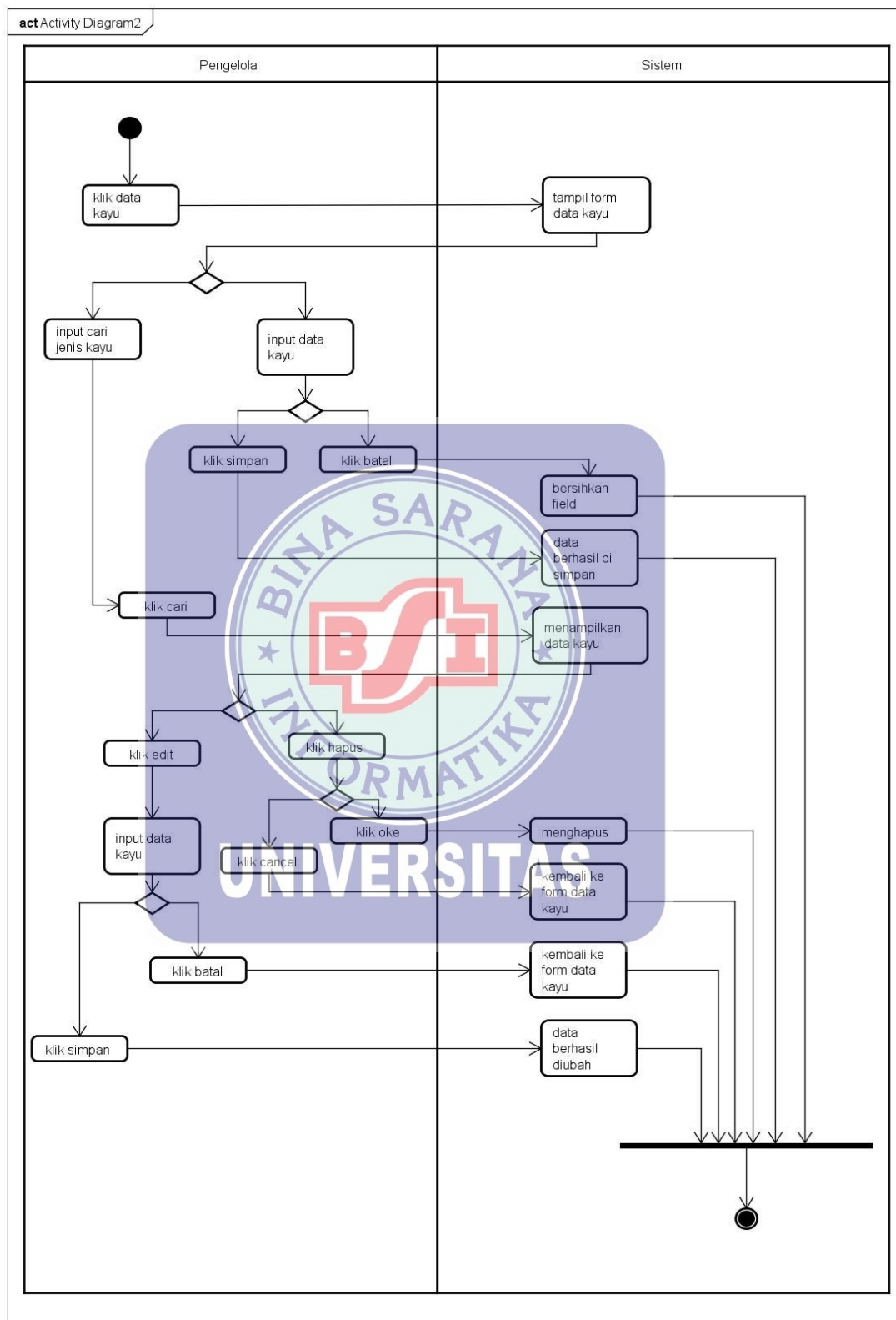
1. Activity Diagram Pengelola Login Menu Utama



Gambar III.4.

Activity Diagram Login Menu Utama

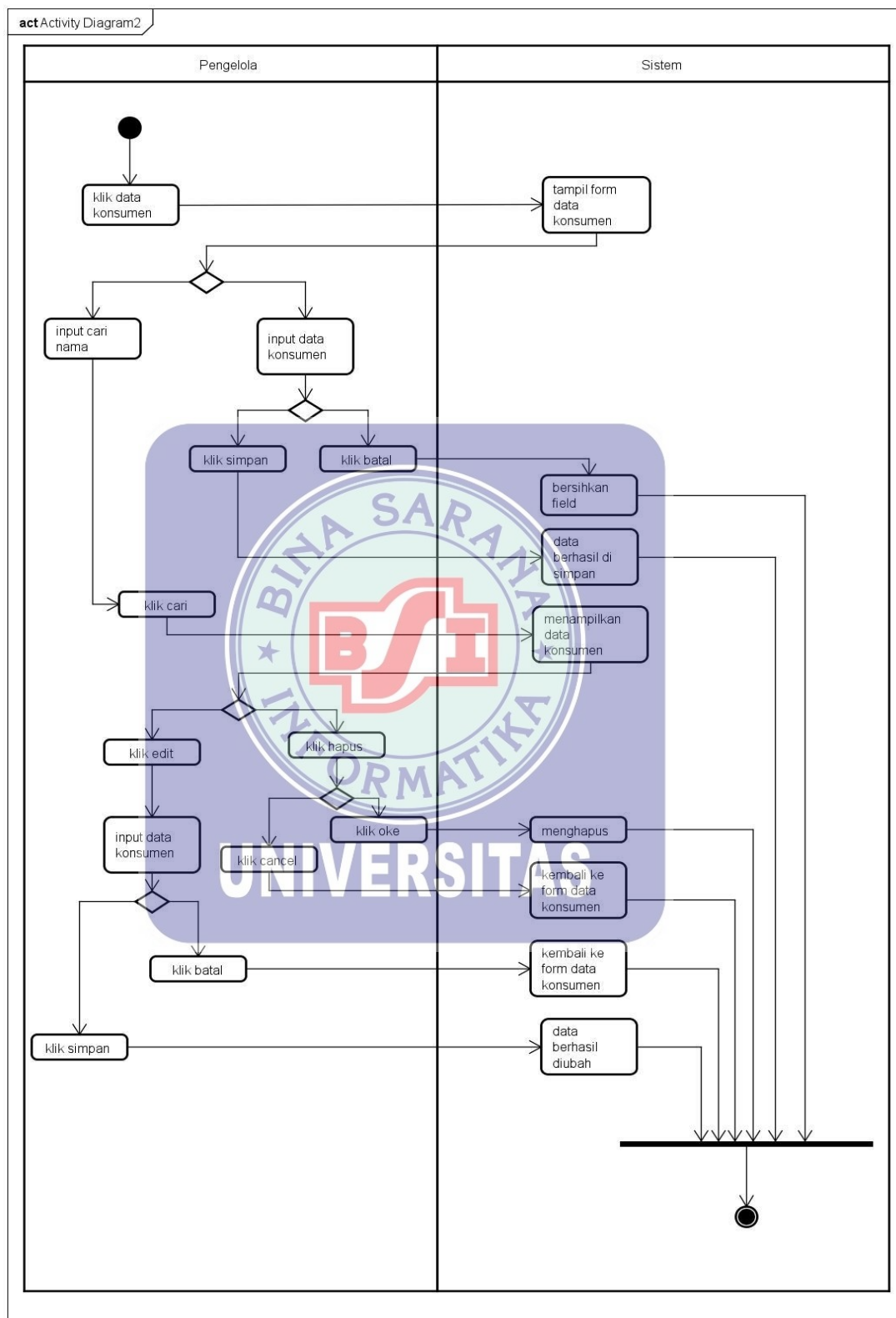
2. Activity Diagram Pengelola Data Kayu



Gambar III.5.

Activity Diagram Data Kayu

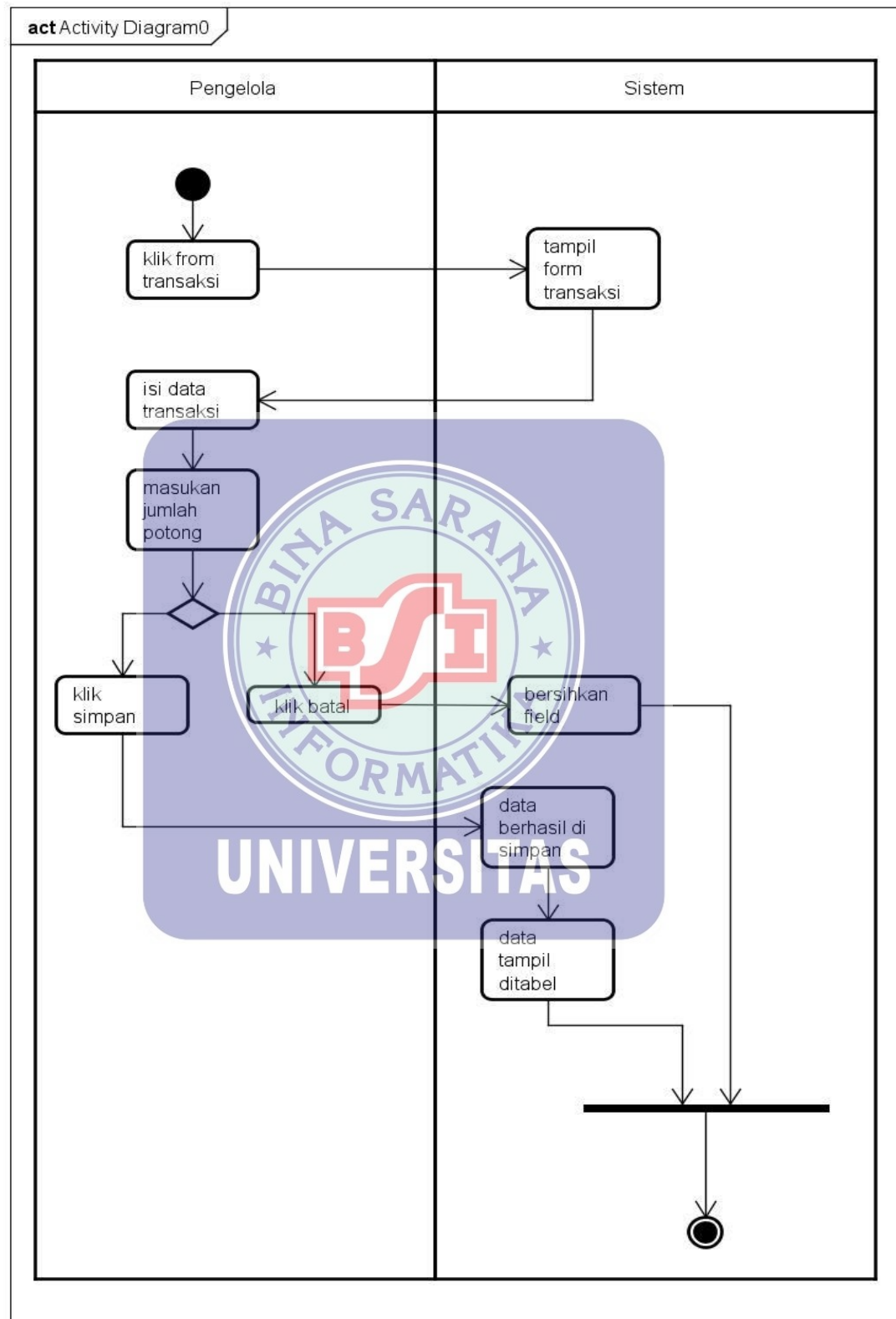
3. Activity Diagram Pengelola Konsumen



Gambar III.6.

Activity Diagram Data Konsumen

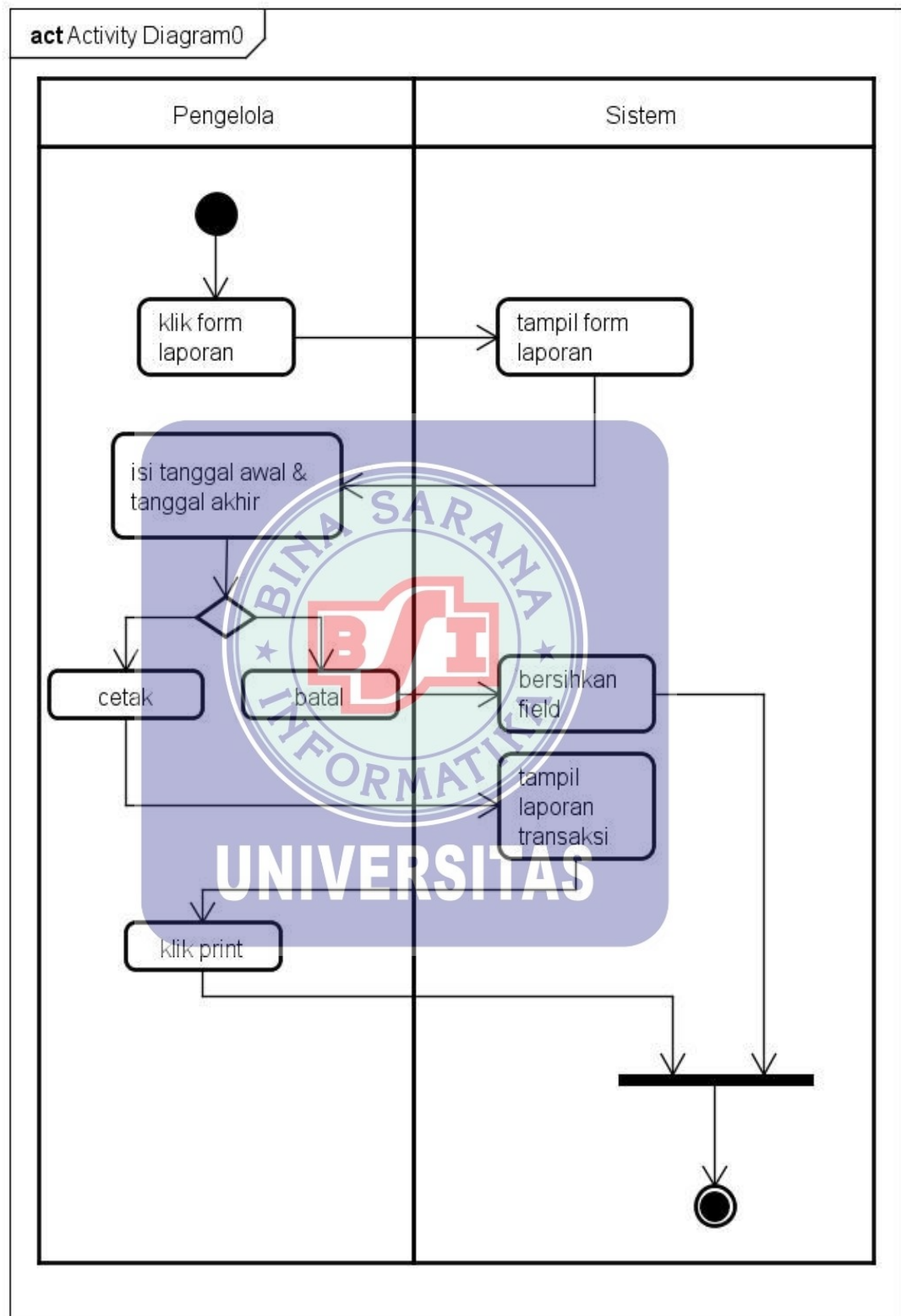
4. Activity Diagram Pengelola Transaksi



Gambar III.7.

Activity Diagram Transaksi

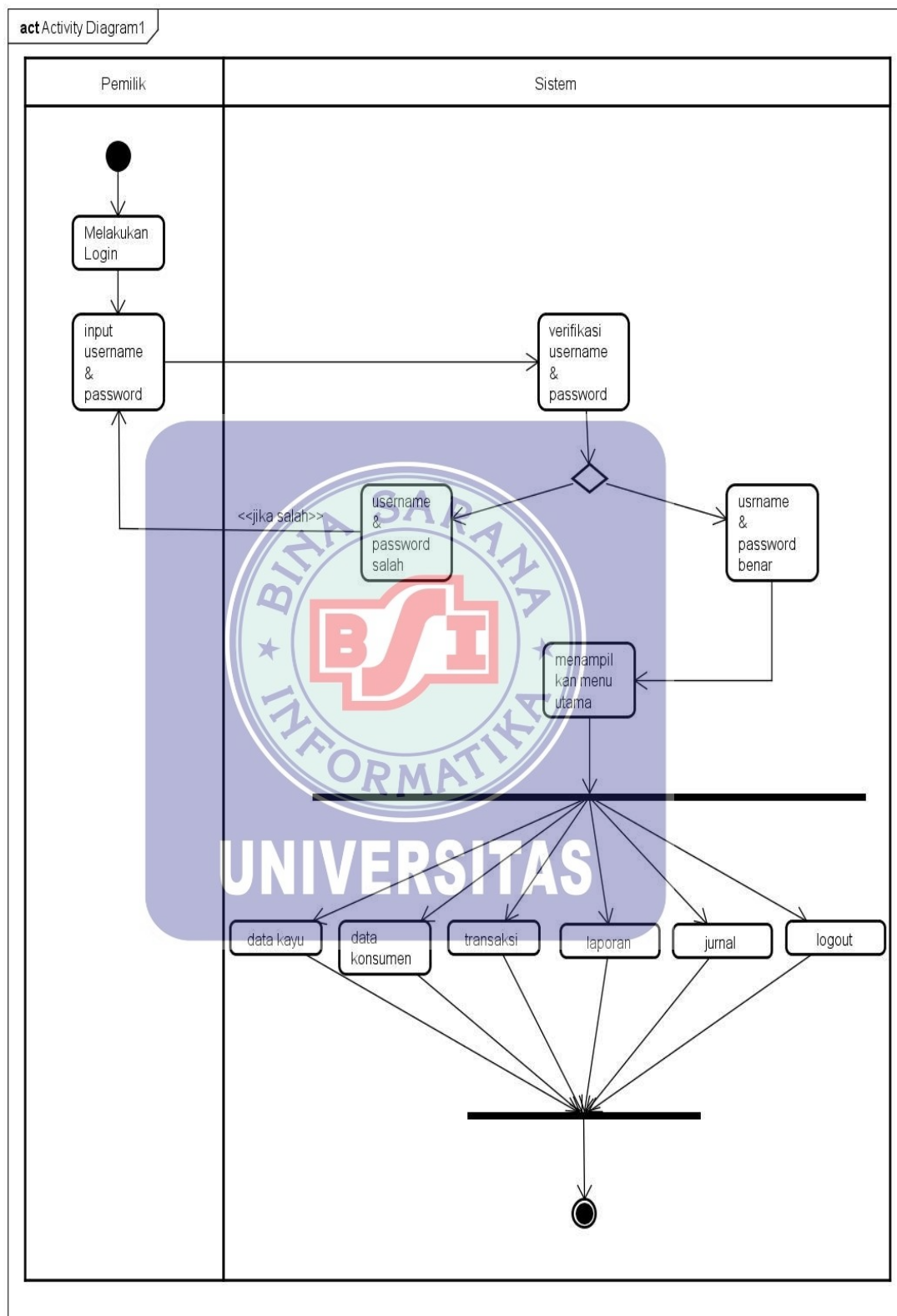
5. Activity Diagram Pengelola Laporan



Gambar III.8.

Activity Diagram Laporan

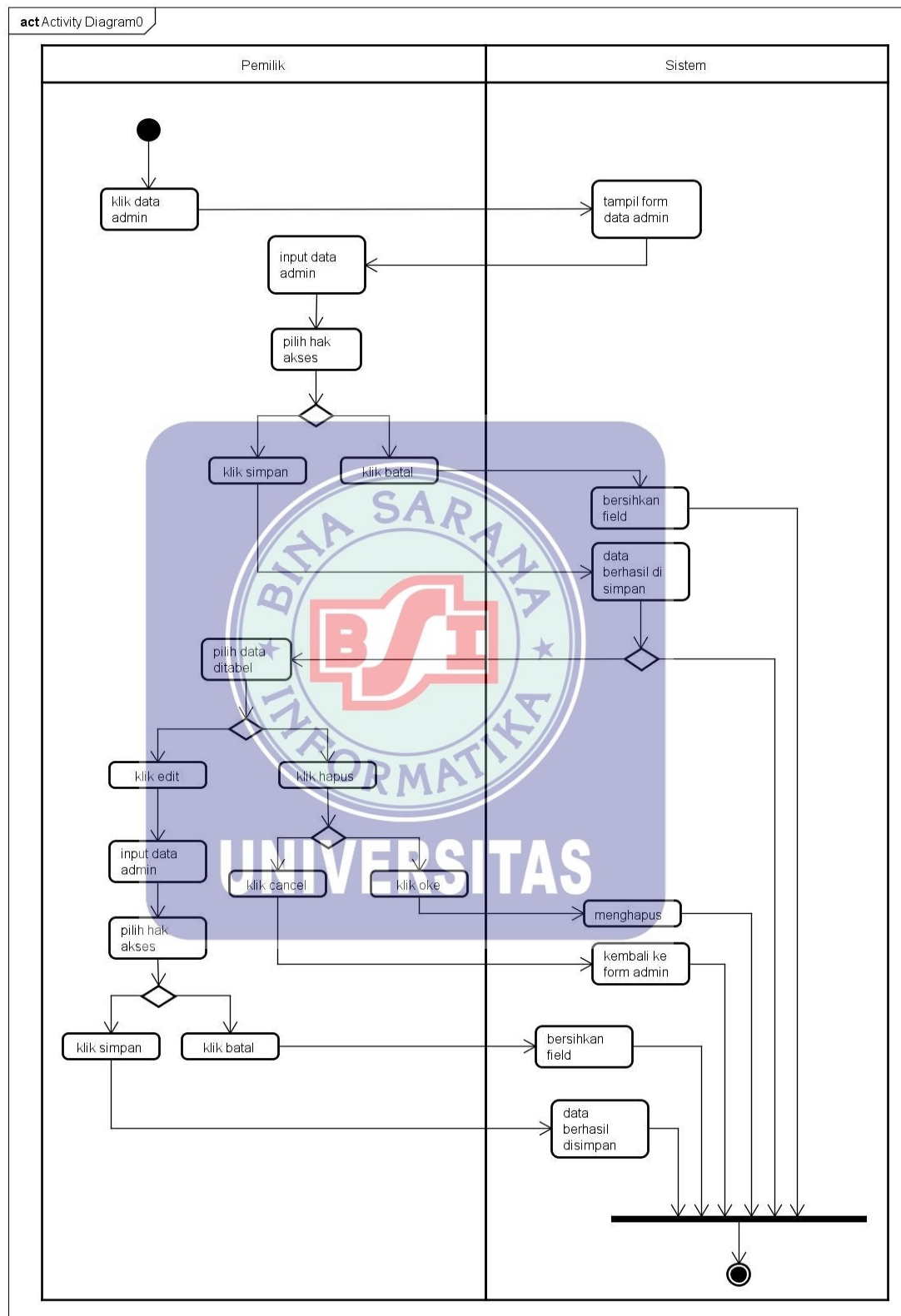
6. Activity Diagram Pemilik Login Menu Utama



Gambar III.9.

Activity Diagram Login Menu Utama

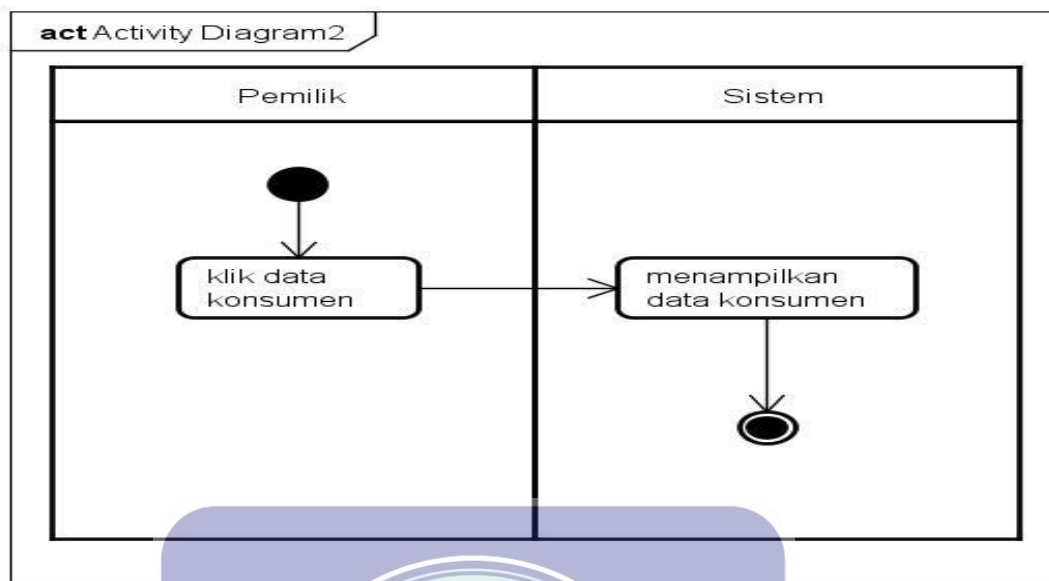
7. Activity Diagram Pemilik Data Admin



Gambar III.10.

Activity Diagram Data Admin

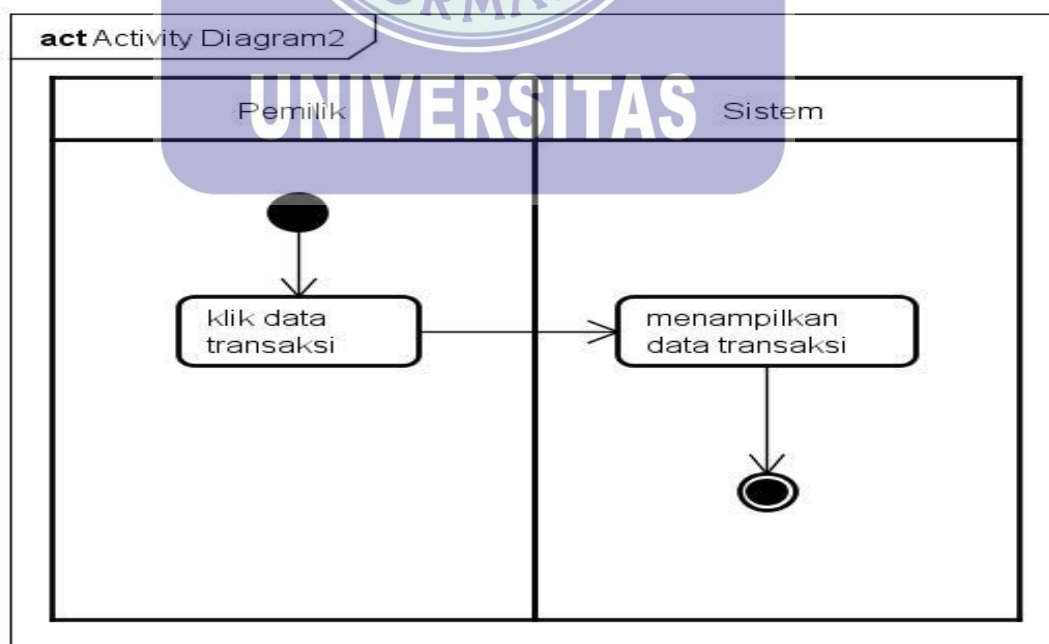
8. Activity Diagram Pemilik Lihat Data Konsumen



Gambar III.11.

Activity Diagram Lihat Data Konsumen

9. Activity Diagram Pemilik Lihat Transaksi



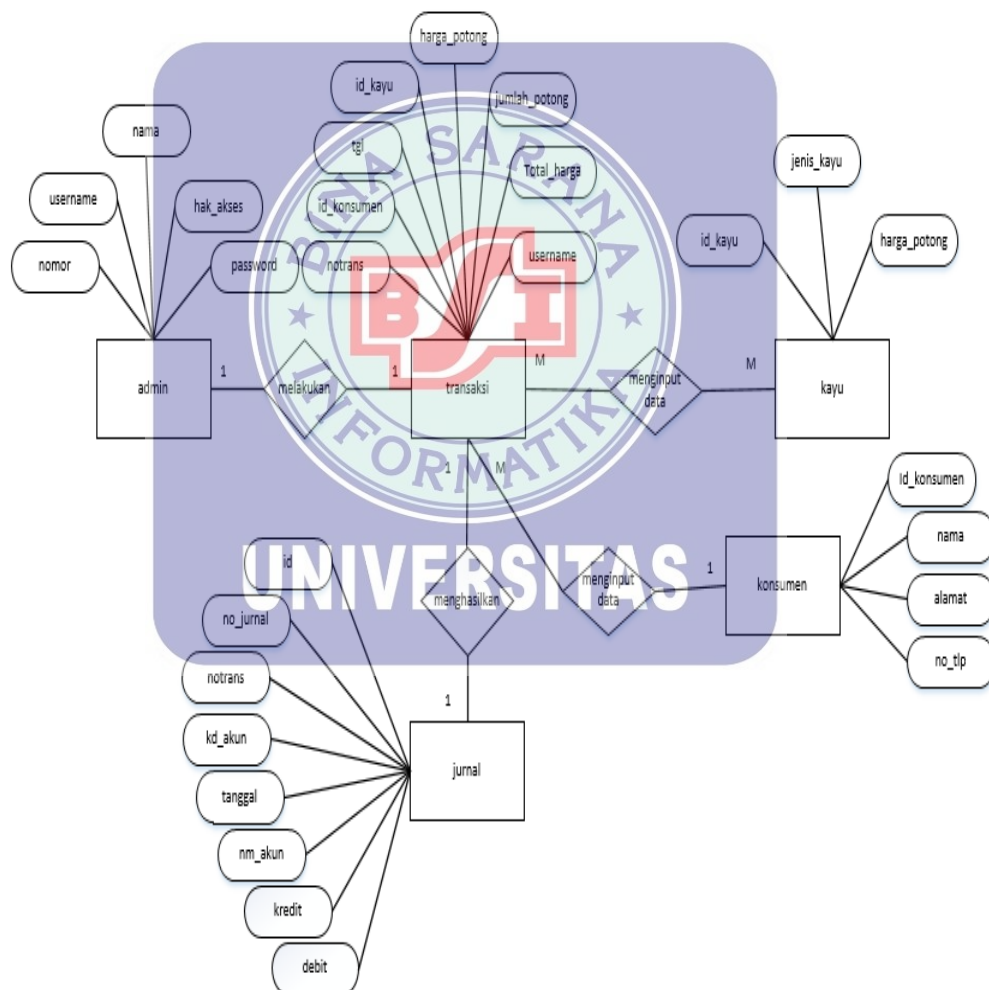
Gambar III.12.

Activity Diagram Lihat Transaksi

3.4. Desain

Merupakan suatu perencanaan atau perancangan yang dilakukan sebelum pembuatan suatu objek, sistem, komponen atau struktur, bertujuan agar objek yang diciptakan memiliki fungsi, memiliki nilai keindahan, dan berguna bagi manusia.

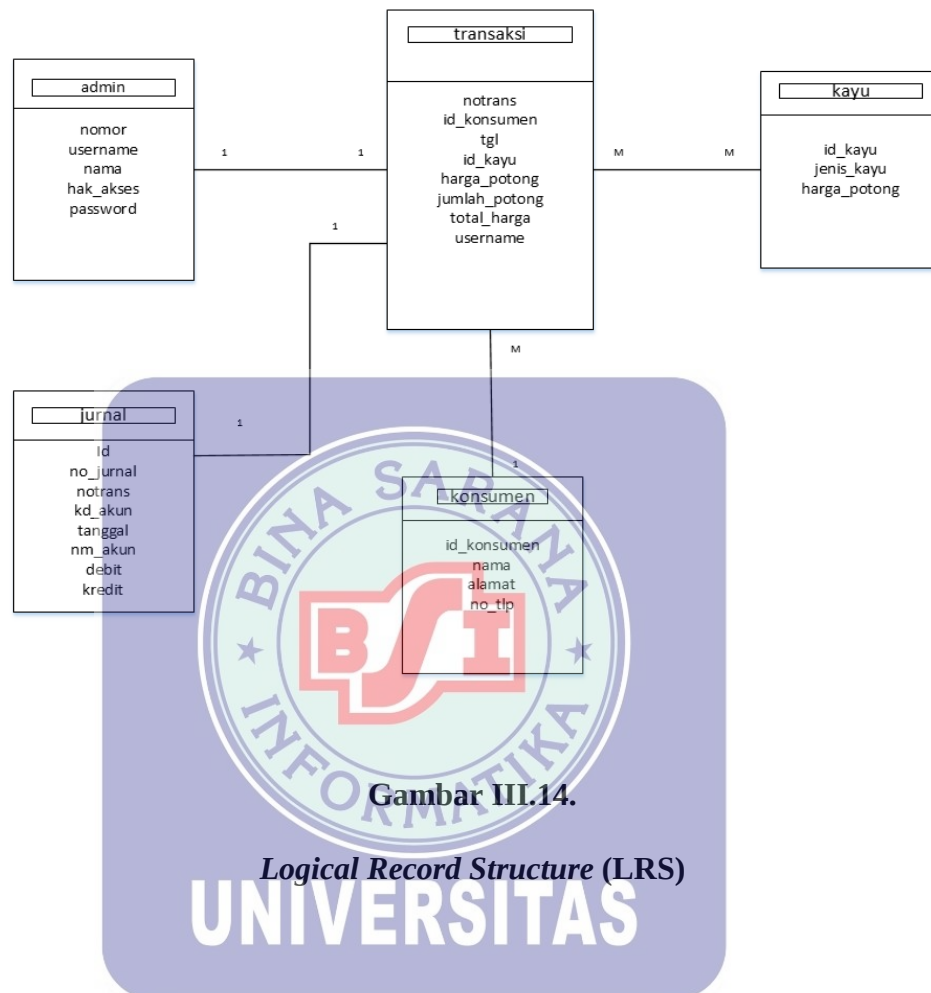
3.4.1. Entity Relationship Diagram (ERD)



Gambar III.13.

Entity Relationship Diagram (ERD)

3.4.2. Logical Record Structure (LRS)



Gambar III.14.

Logical Record Structure (LRS)

UNIVERSITAS

3.4.3. Spesifikasi File

1. Spesifikasi File Tabel Admin

Nama Database : awal

Nama File : admin

Akronim : admin.sql

Tipe file : file master

Akses file : random

Panjang record : 126 karakter

Kunci field : nomor

Tabel III.1.

Spesifikasi *File Admin*

NO	Elemen Data	Nama Field	Type	Size	Keterangan
1	Nomor	nomor	int	11	primary key
2	Username	username	varchar	25	
3	Nama	nama	varchar	35	
4	Hak akses	hak_akses	varchar	15	
5	Password	password	varchar	40	

2. Spesifikasi *File Tabel Jurnal*Nama *Database* : awalNama *File* : jurnal

Akronim : jurnal.sql

Tipe *file* : *file master*Akses *file* : *random*Panjang *record* : 61 karakterKunci *field* : id

Tabel III.2.

Spesifikasi *File Jurnal*

NO	Elemen Data	Nama Field	Type	Size	Keterangan
1	Id	id	int	11	primary key
2	No jurnal	no_jurnal	varchar	15	
3	Notrans	notrans	varchar	15	
4	Kd akun	kd_akun	int	11	

5	Tanggal	tanggal	date		
6	Nm akun	nm_akun	varchar	35	
7	Debit	debit	double		
8	Kredit	kredit	double		

3. Spesifikasi *File* Tabel Kayu

Nama *Database* : awal

Nama *File* : kayu

Akronim : kayu.sql

Tipe *file* : *file master*

Akses *file* : *random*

Panjang *record* : 41 karakter

Kunci field : id_kayu

Tabel III.3.

Spesifikasi *File* Kayu

NO	Elemen Data	Nama Field	Type	Size	Keterangan
1	Id Kayu	id_kayu	int	11	primary key
2	Jenis Kayu	jenis_kayu	varchar	30	
3	Harga Potong	harga_potong	double		

4. Spesifikasi *File* Tabel Konsumen

Nama *Database* : awal

Nama *File* : konsumen

Akronim : konsumen.sql

Tipe *file* : *file master*

Akses *file* : *random*

Panjang *record* : 61 karakter

Kunci *field* : id_konsumen

Tabel III.4.

Spesifikasi *File* Konsumen

NO	Elemen Data	Nama Field	Type	Size	Keterangan
1	Id Konsumen	id_konsumen	int	11	primary key
2	Nama	nama	varchar	35	
3	Alamat	alamat	text		
4	No Tlp	no_tlp	varchar	15	

5. Spesifikasi *File* Tabel Transaksi

Nama *Database* : awal

Nama *File* : transaksi

Akronim : transaksi.sql

Tipe *file* : file transaksi

Akses *file* : *random*

Panjang *record* : 106 karakter

Kunci *field* : notrans

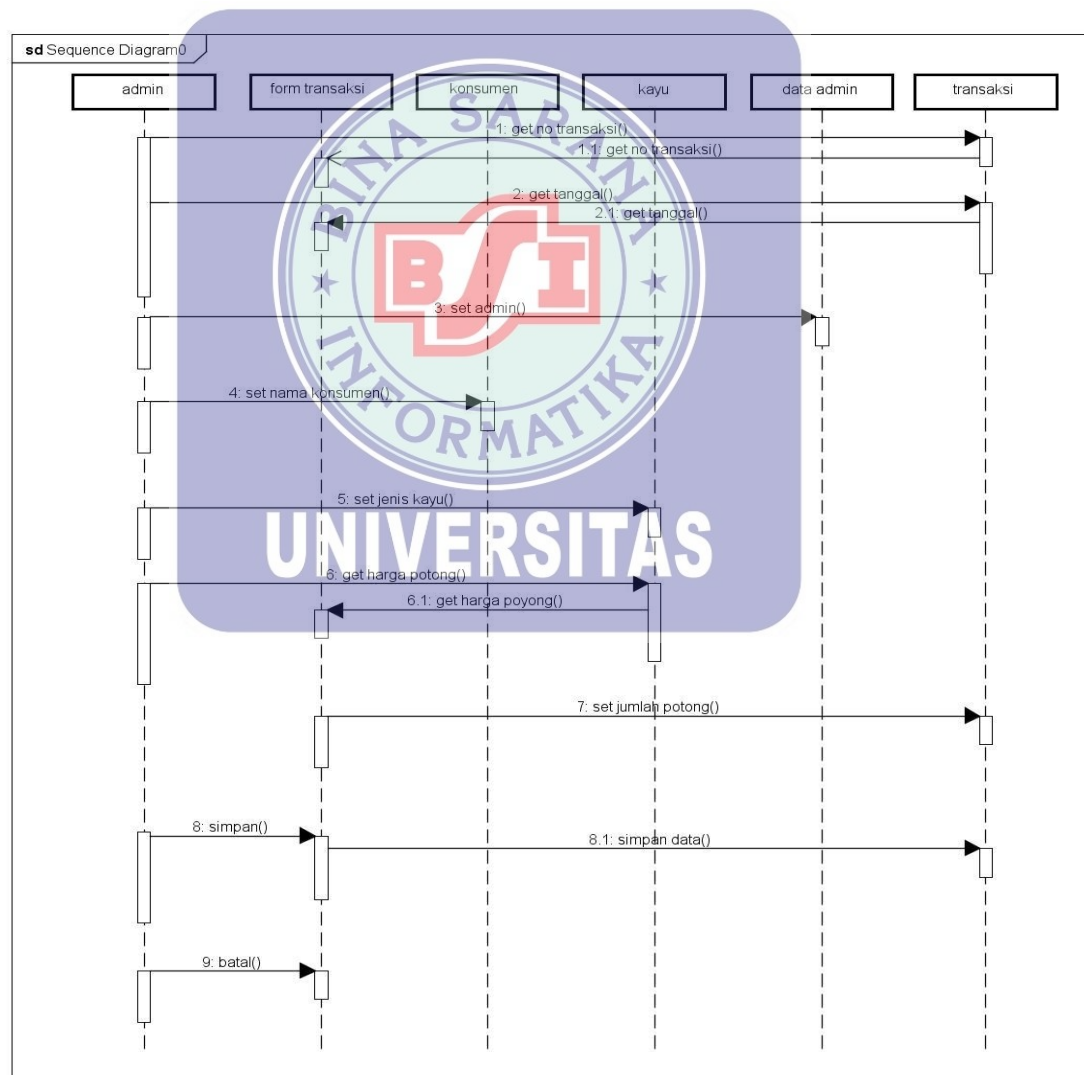
Tabel III.5.

Spesifikasi *File* Transaksi

NO	Elemen Data	Nama Field	Type	Size	Keterangan
1	Notrans	notrans	varchar	15	primary key
2	Id Konsumen	id_konsumen	varchar	30	

3	Tgl	tgl	date		
4	Id Kayu	id_kayu	varchar	30	
5	Harga Potong	harga_potong	double		
6	Jumlah Potong	jumlah_potong	double		
7	Total Harga	total_harga	double		
8	Username	username	varchar	35	

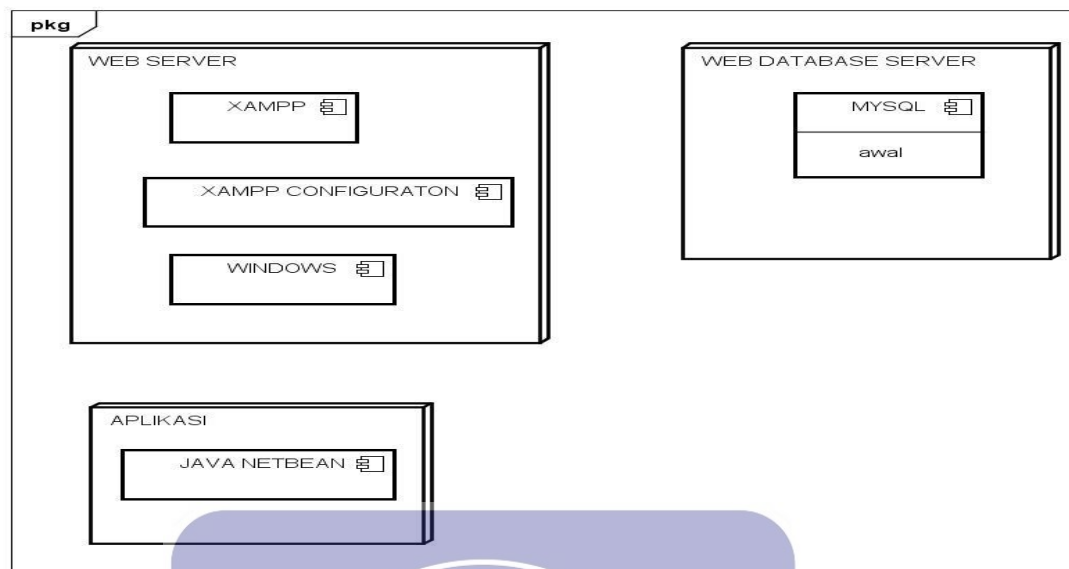
3.4.4. Sequence Diagram



Gambar III.20.

Sequence Diagram

3.4.5. Deployment Diagram



Gambar III.21.

Deployment Diagram

3.4.6. User Interface

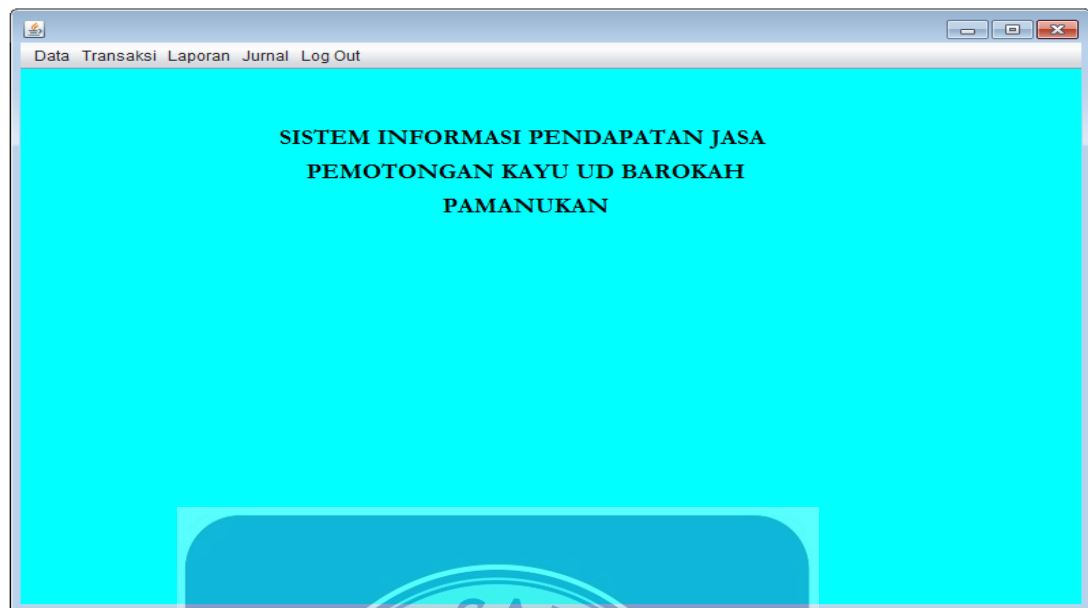
1. Form Login

The screenshot shows a Java Swing window titled 'UNIVERSITAS Form Login'. The window has a light gray background and a blue title bar. It contains two text input fields: one for 'Username :' and one for 'Password :'. Below the input fields are two buttons: 'Login' and 'Batal'. The window is overlaid with a large, semi-transparent watermark of the Bina Sarana Informatika (BSI) logo and the text 'UNIVERSITAS BINA SARANA INFORMATIKA'.

Gambar III.22.

Form Login

2. Form Menu Utama



Gambar III.23.

Form Menu Utama

3. Form Data Kayu

The screenshot shows a window titled 'Data Kayu'. It contains input fields for 'Jenis Kayu' and 'Harga Potong', and buttons for 'Simpan', 'Batal', 'Cari', 'Edit', and 'Hapus'. Below the input fields is a table with the following data:

ID	Jenis Kayu	Harga Potong
2	Pinus	15000
3	jati	20000

Gambar III.24.

Form Data Kayu

4. Form Data Konsumen

Data Konsumen

Nama Konsumen :

No Tlp :

Alamat :

Simpan

Batal

Hapus

Cari Nama : Cari Edit

ID	Nama	Alamat	No Telp
2	Aldy	Cikampek	0829129929

Gambar III.25.

Form Data Konsumen

5. Form Admin

Form Admin

Username : Hak Akses : **Pengelola**

Nama :

Password :

Ulangi Password :

Simpan

Batal

Edit Hapus

ID	Username	Nama	Hak Akses
4	aldy	aldy julian	Pemilik
5	dede	dede aja	Pengelola
6	edo	edo h	Pengelola

Gambar III.26.

Form Admin

6. Form Transaksi

Form Transaksi

No Transaksi : Admin :

Nama Konsumen : Tanggal :

Jenis Kayu :

Harga Potong :

Jumlah Potong :

Total Harga :

ID	No Trans...	ID Konsu...	Tanggal	Jenis Kayu	Harga Pot...	Jumlah P...	Total Harga	Admin
1	TR08071...	Aldy	2019-07-08	Pinus	15000	2	30000	Ropik
2	TR08071...	Aldy	2019-07-08	Pinus	15000	3	45000	Ropik
3	TR09071...	Aldy	2019-07-09	Pinus	15000	5	75000	Ropik
4	TR09071...	Aldy	2019-07-09	Pinus	15000	7	105000	Ropik
5	TR09071...	Aldy	2019-07-09	Pinus	15000	12	180000	Ropik
6	TR09071...	Aldy	2019-07-09	jati	20000	3	60000	dede aja

Gambar III.27.

Form Transaksi

7. Laporan

Tanggal Awal :

Tanggal Akhir :

ID	No Tra...	ID Kon...	Tanggal	Jenis ...	Harga ...	Jumla...	Total H...	Admin
1	TR080...	Aldy	2019-0...	Pinus	15000	2	30000	Ropik
2	TR080...	Aldy	2019-0...	Pinus	15000	3	45000	Ropik
3	TR090...	Aldy	2019-0...	Pinus	15000	5	75000	Ropik
4	TR090...	Aldy	2019-0...	Pinus	15000	7	105000	Ropik
5	TR090...	Aldy	2019-0...	Pinus	15000	12	180000	Ropik
6	TR090...	Aldy	2019-0...	jati	20000	3	60000	dede aja

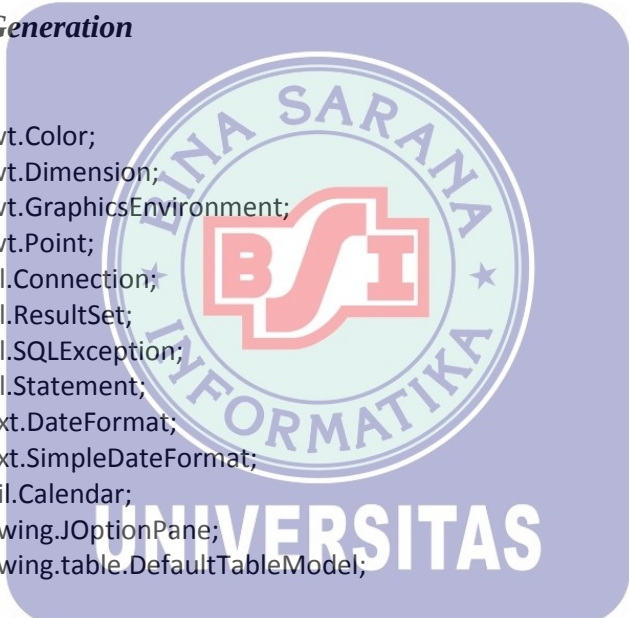
Gambar III.28.

Laporan

3.5. Implementasi

Implementasi merupakan kelanjutan dari kegiatan perancangan sistem dan dapat dipandang sebagai usaha untuk mewujudkan sistem yang dirancang. Langkah-langkah dan proses implementasi adalah urutan dari kegiatan awal sampai kegiatan akhir yang harus dilakukan dalam mewujudkan sistem yang dirancang. Hasil dari tahapan implementasi ini adalah suatu sistem pengolahan data yang sudah dapat berjalan dengan baik.

3.5.1. Code Generation



```
package Awal;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.GraphicsEnvironment;
import java.awt.Point;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
public class form_transaksi extends javax.swing.JFrame {
    private DefaultTableModel model;
    /**
     * Creates new form form_transaksi
     */
    public form_transaksi() {
        initComponents();
        initUI();
        notrans();
        txtid.setVisible(false);
        this.getContentPane().setBackground(Color.white);
        txtadmin.setText(SesiAdmin.getU_nama());
        DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
```

```

Calendar cal = Calendar.getInstance();
txttgl.setText(dateFormat.format(cal.getTime()));
model = new DefaultTableModel ();
jTable1.setModel(model);
model.addColumn("ID");
model.addColumn("No Transaksi");
model.addColumn("ID Konsumen");
model.addColumn("Tanggal");
model.addColumn("Jenis Kayu");
model.addColumn("Harga Potong");
model.addColumn("Jumlah Potong");
model.addColumn("Total Harga");
model.addColumn("Admin");
GetData();
}
private void initUI(){
getContentPane().setBackground(new Color(245, 245, 245));
Dimension windowSize = getSize();
GraphicsEnvironment ge = GraphicsEnvironment.getLocalGraphicsEnvironment();
Point centerPoint = ge.getCenterPoint();
int dx = centerPoint.x - windowSize.width / 2;
int dy = centerPoint.y - windowSize.height / 2;
setLocation(dx, dy);
}
public void clear(){
txtid.setText("");
txtnmkons.setText("");
txtjenis.setText("");
txtharga.setText("");
txtjml.setText("");
txttotal.setText("");
}
private void GetData(){
model.getDataVector().removeAllElements();
model.fireTableDataChanged();
try{
Statement stat = (Statement) koneksi.getKoneksi().createStatement();
String sql = "Select * from transaksi";
ResultSet res = stat.executeQuery(sql);
while(res.next()){
Object[] obj = new Object[9];
obj[0] = res.getString("id");
obj[1] = res.getString("notrans");
obj[2] = res.getString("id_konsumen");
obj[3] = res.getString("tgl");
obj[4] = res.getString("jenis_kayu");
obj[5] = res.getString("harga_potong");
obj[6] = res.getString("Jumlah_potong");
obj[7] = res.getString("total_harga");
obj[8] = res.getString("admin");
model.addRow(obj);
}
}

```

```

}
}catch(SQLException err){
JOptionPane.showMessageDialog(null, err.getMessage() );
}
}

public String notrans(){
String urutan=null;
java.util.Date skrg = new java.util.Date();
java.text.SimpleDateFormat kal=new java.text.SimpleDateFormat("ddMMyy");
String tgl=(kal.format(skrg));
try{
Connection con = koneksi.getKoneksi();
Statement st=con.createStatement();
java.sql.ResultSet sql = st.executeQuery("SELECT  (right (notrans,3)+1) as nomor FROM
transaksi where notrans like '%" +tgl+"%' order by notrans desc");
if(sql.next())
{
urutan=sql.getString(1);
while (urutan.length()<3)
urutan="0"+urutan;
urutan="TR"+tgl+urutan;
}else
{
urutan="TR"+tgl+"001";
}
txtnotrans.setText(urutan);
}
catch(Exception e){
// JOptionPane.showMessageDialog(null,e);
}
return urutan;
}

public String nmkons;
public String jeniskayu;
public String hargapotong;
public String getnmkons(){
return nmkons;
}

public String getjeniskayu(){
return jeniskayu;
}

public String gethargapotong(){
return hargapotong;
}
}
/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">

```

```

private void initComponents() {
jTextField11 = new javax.swing.JTextField();
jPanel1 = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();
txtid = new javax.swing.JTextField();
txtnmkons = new javax.swing.JTextField();
txtnotrans = new javax.swing.JTextField();
txtjenis = new javax.swing.JTextField();
txtjml = new javax.swing.JTextField();
txtharga = new javax.swing.JTextField();
txtadmin = new javax.swing.JTextField();
txttotal = new javax.swing.JTextField();
jLabel3 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();
jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
jLabel15 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
btnsimpan = new javax.swing.JButton();
btnbatal = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();
txtcarikayu = new javax.swing.JButton();
txtcarikons = new javax.swing.JButton();
txttgl = new javax.swing.JTextField();
jTextField11.setText("jTextField11");
setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
jLabel1.setFont(new java.awt.Font("Garamond", 1, 18)); // NOI18N
jLabel1.setText("Form Transaksi");
jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 10, -1, -1));
jLabel2.setFont(new java.awt.Font("Garamond", 1, 11)); // NOI18N
jLabel2.setText("Total Harga      :");
jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 300, 90, 20));
jLabel6.setFont(new java.awt.Font("Garamond", 1, 11)); // NOI18N
jLabel6.setText("Tanggal      :");
jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(360, 100, 80, 20));
jLabel10.setFont(new java.awt.Font("Garamond", 1, 11)); // NOI18N
jLabel10.setText("Admin      :");
jPanel1.add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(360, 60, 80, 20));
txtid.setEditable(false);
jPanel1.add(txtid, new org.netbeans.lib.awtextra.AbsoluteConstraints(270, 60, 20, -1));
txtnmkons.setEditable(false);
jPanel1.add(txtnmkons, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 100, 150, -1));

txtnotrans.setEditable(false);

```

```

jPanel1.add(txtnotrans, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 60, 150, -
1));
txtjenis.setEditable(false);
jPanel1.add(txtjenis, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 140, 150, -1));
txtjml.addKeyListener(new java.awt.event.KeyAdapter() {
public void keyPressed(java.awt.event.KeyEvent evt) {
txtjmlKeyPressed(evt);
}
public void keyReleased(java.awt.event.KeyEvent evt) {
txtjmlKeyReleased(evt);
}
public void keyTyped(java.awt.event.KeyEvent evt) {
txtjmlKeyTyped(evt);
}
});
jPanel1.add(txtjml, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 220, 150, -1));
txtharga.setEditable(false);
jPanel1.add(txtharga, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 180, 150, -
1));
txtadmin.setEditable(false);
jPanel1.add(txtadmin, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 60, 140, -1));
txttotal.setEditable(false);
jPanel1.add(txttotal, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 300, 180, -1));
jLabel3.setFont(new java.awt.Font("Garamond", 1, 11)); // NOI18N
jLabel3.setText("Nama Konsumen :");
jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 100, 100, 20));
jLabel12.setFont(new java.awt.Font("Garamond", 1, 11)); // NOI18N
jLabel12.setText("No Transaksi :");
jPanel1.add(jLabel12, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 60, 100, 20));
jLabel13.setFont(new java.awt.Font("Garamond", 1, 11)); // NOI18N
jLabel13.setText("Jenis Kayu :");
jPanel1.add(jLabel13, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 140, 100, 20));
jLabel14.setFont(new java.awt.Font("Garamond", 1, 11)); // NOI18N
jLabel14.setText("Jumlah Potong :");
jPanel1.add(jLabel14, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 220, 100, 20));
jLabel15.setFont(new java.awt.Font("Garamond", 1, 11)); // NOI18N
jLabel15.setText("Harga Potong :");
jPanel1.add(jLabel15, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 180, 100, 20));
jLabel4.setText("=====
=====");
jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 270, 700, -1));
btnsimpan.setFont(new java.awt.Font("Garamond", 1, 18)); // NOI18N
btnsimpan.setText("Simpan");
btnsimpan.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
btnsimpanActionPerformed(evt);
}
});
jPanel1.add(btnsimpan, new org.netbeans.lib.awtextra.AbsoluteConstraints(390, 200, 110,
50));
btnbatal.setFont(new java.awt.Font("Garamond", 1, 18)); // NOI18N

```

```

btnbatal.setText("Batal");
btnbatal.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnbatalActionPerformed(evt);
    }
});
jPanel1.add(btnbatal, new org.netbeans.lib.awtextra.AbsoluteConstraints(520, 200, 110, 50));
jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null, null},
        {null, null, null, null, null, null},
        {null, null, null, null, null, null},
        {null, null, null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4", "Title 5", "Title 6"
    }
));
jScrollPane1.setViewportView(jTable1);
jPanel1.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 330, 670, 150));
txtcarikayu.setText("jButton3");
txtcarikayu.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtcarikayuActionPerformed(evt);
    }
});
jPanel1.add(txtcarikayu, new org.netbeans.lib.awtextra.AbsoluteConstraints(290, 140, 30, 25));
txtcarikons.setText("jButton3");
txtcarikons.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtcarikonsActionPerformed(evt);
    }
});
jPanel1.add(txtcarikons, new org.netbeans.lib.awtextra.AbsoluteConstraints(290, 100, 30, 25));
txtttl.setEditable(false);
jPanel1.add(txtttl, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 100, 140, -1));
getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 690, 510));
pack();
} // </editor-fold>
private void btnsimpanActionPerformed(java.awt.event.ActionEvent evt) {
    String row_id = txtid.getText();
    String row_txtnotrans = txtnotrans.getText();
    String row_txtidkons = txtnmkons.getText();
    String row_txtjenis = txtjenis.getText();
    String row_txtharga = txtharga.getText();
    String row_txtjml = txtjml.getText();

```

```

String row_txtadmin = txtadmin.getText();
String row_txttotal = txttotal.getText();
String row_txttgl = txttgl.getText();
int c_kode = 0;
if(!"".equals(row_txtidkons) && !"".equals(row_txtjenis) && !"".equals(row_txtjml) &&
!"".equals(row_txtadmin)){
try {
Connection con = koneksi.getKoneksi();
Statement stm=con.createStatement();
java.sql.ResultSet sql = stm.executeQuery("SELECT COUNT(transaksi.id) as count FROM
transaksi WHERE transaksi.notrans='"+row_txtnotrans+"'");
sql.next();
c_kode = sql.getInt("count");
con.close();
}catch (SQLException e) {
JOptionPane.showMessageDialog(null, "Error " + e);
}
if("".equals(row_id)){
if(c_kode == 0){
try {
Connection con = koneksi.getKoneksi();
Statement stm=con.createStatement();
stm.executeUpdate("INSERT INTO transaksi(notrans, id_konsumen, tgl, jenis_kayu,
harga_potong, jumlah_potong, total_harga, admin) VALUES ('" + row_txtnotrans + ", '" +
row_txtidkons + ", '" + row_txttgl + ", '" + row_txtjenis + ", '" + row_txtharga + ", '" +
row_txtjml + ", '" + row_txttotal + ", '" + row_txtadmin + "')");
JOptionPane.showMessageDialog(null, "Data berhasil disimpan");
btnbatal.doClick();
con.close();
GetData();
notrans();
}catch (SQLException e) {
JOptionPane.showMessageDialog(null, "Error " + e);
}
}else{
}
}else{
}
}else{
JOptionPane.showMessageDialog(null, "Terdapat inputan kosong");
}
}

private void btnbatalActionPerformed(java.awt.event.ActionEvent evt) {
clear();
jTable1.clearSelection();
}

private void txtcarikonsActionPerformed(java.awt.event.ActionEvent evt) {
boolean closable = true;
carikonsumen a = new carikonsumen(this, rootPaneCheckingEnabled);
a.kons=this;
a.setVisible(true);

```

```

if(!a.isVisible()){
    txtnmkons.setText(nmkons);
}
}

private void txtcarikayuActionPerformed(java.awt.event.ActionEvent evt) {
    boolean closable = true;
    carikayu a = new carikayu(this, rootPaneCheckingEnabled);
    a.kayu=this;
    a.setVisible(true);
    if(!a.isVisible()){
        txtjenis.setText(jeniskayu);
        txtharga.setText(hargapotong);
    }
}

private void txtjmlKeyReleased(java.awt.event.KeyEvent evt) {
    int harga=Integer.parseInt(txtharga.getText());
    int jumlah=Integer.parseInt(txtjml.getText());
    int total;
    total = jumlah * harga;
    txttotal.setText(String.valueOf(total));
}

private void txtjmlKeyPressed(java.awt.event.KeyEvent evt) {
}

private void txtjmlKeyTyped(java.awt.event.KeyEvent evt) {
}

/**
 * @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(form_transaksi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(form_transaksi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

```

```

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

```

```
//</editor-fold>
```

```
/* Create and display the form */
```

```

java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new form_transaksi().setVisible(true);
}
});
}

```

```
// Variables declaration - do not modify
```

```

private javax.swing.JButton btnbatal;
private javax.swing.JButton btnsimpan;

private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextField11;
private javax.swing.JTextField txtadmin;
private javax.swing.JButton txtcarikayu;
private javax.swing.JButton txtcarikons;
private javax.swing.JTextField txtharga;
private javax.swing.JTextField txtid;
private javax.swing.JTextField txtjenis;
private javax.swing.JTextField txtjml;
private javax.swing.JTextField txtnmkons;
private javax.swing.JTextField txtnotrans;
private javax.swing.JTextField txttgl;
private javax.swing.JTextField txttotal;

```

```

// End of variables declaration
}

```

3.5.2. Black Box Testing

1. Pengujian Form Login

Tabel III.6.

Hasil Pengujian Form Login

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Username dan Password tidak diisi kemudian klik tombol login	Username: (kosong) Password: (kosong)	Sistem akan menolak dan menampilkan pesan "Username atau Password belum terdaftar"	Sesuai Harapan	Valid
2	Mengetikkan Username, dan password tidak diisi atau kosong kemudian klik tombol login	Username: admin Password: (kosong)	Sistem akan menolak dan menampilkan pesan "Username atau Password belum terdaftar"	Sesuai Harapan	Valid
3	Mengetikkan Password, dan username tidak diisi atau kosong kemudian klik tombol login	Username: (kosong) Password: admin	Sistem akan menolak dan menampilkan pesan "Username atau Password belum terdaftar"	Sesuai Harapan	Valid

4	Mengetikka Username dan/atau password tidak sesuai, kemudian klik tombol login	Username: dede Password: dede1234	Sistem akan menolak dan menampilkan pesan "Username atau Password belum terdaftar	Sesuai Harapan	Valid
5	Mengetikka username dan password (diisi), kemudian klik tombol login	Username: dede Password: dede	Sistem menerima akses login dan kemudian menampilkan halaman utama	Sesuai Harapan	Valid

2. Pengujian Form Transaksi

Tabel III.7

Hasil Pengujian Form Transaksi

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Input data transaksi (Data yang di input tidak lengkap) lalu klik simpan	Sistem tidak akan menyimpan jika kolom tidak terisi semua, maka akan menampilkan pesan "Terdapat Inputan Kosong"	Sesuai Harapan	Valid
2	Input data transaksi (Data yang di input lengkap) lalu klik simpan	Sistem sukses input data transaksi dan menampilkan pesan "Data berhasil disimpan"	Sesuai Harapan	Valid

3.5.3. Spesifikasi *Hardware* dan *Software*

Spesifikasi *software* (Perangkat Lunak) adalah sesuatu yang berhubungan dengan program komputer. Dalam membuat program aplikasi ini, penulis menggunakan beberapa *software* yaitu:

1. Sistem Operasi : *Microsoft Windows*
2. Bahasa Pemrograman : *Java*
3. Aplikasi Pendukung : *NetBeans IDE 8.1, XAMPP v3.2.3, JASPERREPORT*
4. DBMS/ Basis Data : *MySQL*
5. Browser : *Google Chrome v75.0.3770.142 (64-bit)*

Spesifikasi *Hardware* merupakan perangkat keras komputer, seperti *monitor*, *printer*, *keyboard*, *mouse*, dan lain-lain. Minimal dalam spesifikasi *file* untuk dapat membuat dan menjalankan aplikasi ini adalah sebagai berikut:

1. *Processor* : Intel Core i3
2. *RAM* : 2 GB
3. *Monitor* : 14 inc
4. *Mouse* : CLIPtec
5. *Printer* : HP DeskJet Ink Advantage 2135
6. *Hardisk* : 500 GB