

BAB III

PEMBAHASAN

3.1. Tinjauan Perusahaan

Toko Bangunan Usaha Jaya Cikampek merupakan suatu badan usaha dagang yang bergerak dalam bidang bahan bangunan, Toko Bangunan Usaha Jaya Cikampek adalah sebuah perusahaan yang ingin maju dan berkembang. Maka perusahaan ini melakukan sistem pembelian tunai untuk mendapatkan keuntungan dan dapat terus memenuhi kebutuhan masyarakat, serta target perusahaan dimasa akan datang.

Sistem pembelian yang terjadi pada perusahaan Toko Bangunan Usaha Jaya Cikampek masih menggunakan sistem yang sederhana, khususnya dalam kegiatan transaksi, pembelian dan pencatatan belum berjalan dengan optimal sehingga masih sering terjadi kesalahan dalam pekerjaan dan dapat memperlambat proses kerja menjadi kurang efektif. Hal seperti ini jika terulang terus, maka akan dapat menyebabkan perusahaan tersebut rugi.

Kegiatan yang ada di dalam perusahaan dan untuk dapat mengurangi terjadinya kesalahan, maka langkah yang diciptakan pada Toko Bangunan Usaha Jaya Cikampek salah satunya adalah dengan menggunakan komputerisasi guna membantu kinerja suatu usaha tersebut. Selain itu menggunakan media komputer yang sudah terkomputerisasi juga dapat diperolehnya informasi yang lebih cepat dan tepat, sehingga mampu memberikan layanan yang baik pada pelanggan.

3.1.1. Sejarah Perusahaan

Toko Usaha Jaya berdiri pada tahun 1992 yang berdiri pertama kali di Jakarta, Pak Burtayo selaku pemilik Toko Usaha Jaya yang berada di Cikampek berasal dari Pontianak yang merantau ikut bersama Kakaknya, beliau dipercaya untuk membuka cabang dan diberi modal usaha, akhirnya beliau menekuni usaha material dari tahun 1995 hingga sekarang.

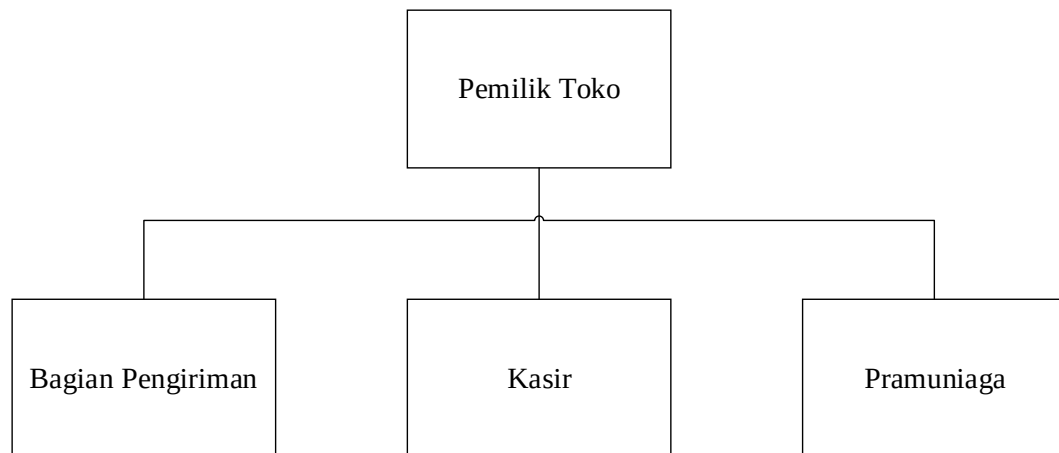
Toko Usaha Jaya adalah usaha yang bergerak dibidang Material (toko bangunan). Dengan beberapa karyawan pak Burtayo menjalankan usahanya. Dengan mengandalkan kejujuran, hemat, percaya, dan berdo'a, dan Pak Burtayo berhasil menjalankan usaha amanat dari Kakaknya.

3.1.2. Struktur Organisasi dan Fungsi

Pada umumnya dalam menjalankan sebuah perusahaan akan mempunyai susunan atau struktur organisasi, CV. Sinar Electric Jakarta mempunyai struktur organisasi, dimana dari struktur organisasi inilah kita dapat melihat jalur kegiatan atau wewenang apa saja yang dilaksanakan oleh masing-masing bagian serta kebijaksanaan yang telah ditetapkan.

Struktur organisasi menunjukkan kerangka dan susunan perwujudan pola tetap hubungan-hubungan diantara fungsi-fungsi, bagian-bagian atau posisi-posisi, maupun orang-orang yang menunjukkan kedudukan, tugas wewenang dan tanggung jawab yang berbeda-beda dalam suatu organisasi.

Adapun Struktur Organisasi Toko Bangunan Usaha Jaya Cikampek adalah sebagai berikut:



Sumber : Toko Bangunan Usaha Jaya

Gambar III.1.

Struktur Organisasi TB. Usaha Jaya Cikampek

Fungsi dari masing-masing bagian dari struktur organisasi Toko Bangunan Usaha Jaya Cikampek, dapat dilihat sebagai berikut:

1. Pemilik Toko
 - a. Sebagai penyedia dana dalam kelancaran usaha.
 - b. Mengontrol dan mengawasi kelancaran proses usaha.
 - c. Mengontrol kinerja kerja karyawan.
 - d. Bertanggung jawab atas semua barang dan data-data yang ada pada toko bangunan Usaha Jaya.
 - e. Melakukan pengecekan atas laporan-laporan yang telah dibuat oleh bagian kasir.
2. Kasir
 - a. Mencatat dan menghitung transaksi-transaksi yang terjadi pada toko bangunan Usaha Jaya.
 - b. Membuat laporan-laporan yang akan diserahkan pada pemilik toko.
 - c. Mengecek stok barang yang ada pada Gudang.

3. Bagian Pengiriman

- a. Mengirimkan barang yang telah dipesan atau dibeli oleh pelanggan.

4. Pramuniaga/karyawan

- a. Melayani pelanggan yang datang ke Toko.

3.2. Tinjauan Kasus

3.2.1. Proses Bisnis Sistem Berjalan

1. Pengecekan barang

Pramuniaga melakukan pengecekan stok barang setiap minggu nya untuk mengetahui stok barang yang sudah hampir habis, kemudian akan dicatat didalam kartu stok untuk diberikan kepada kasir dan dilakukan pemesanan kepada *supplier*.

2. Prosedur pemesanan

Setelah kasir menerima kartu stok, akan dibuatkan *Purchase Request* (PR) pemesanan barang dan kemudian kasir akan menghubungi pihak *supplier* bahwa akan melakukan pemesanan. Setelah dihubungi, pihak *supplier* akan mendatangi kasir untuk melihat PR pemesanan barang apa saja yang akan dipesan. Setelah dilihat pihak *supplier* akan memberitahu kasir bahwa proses pemesanan barang akan diproses dalam waktu 3 hari.

3. Prosedur pengiriman dan pembayaran

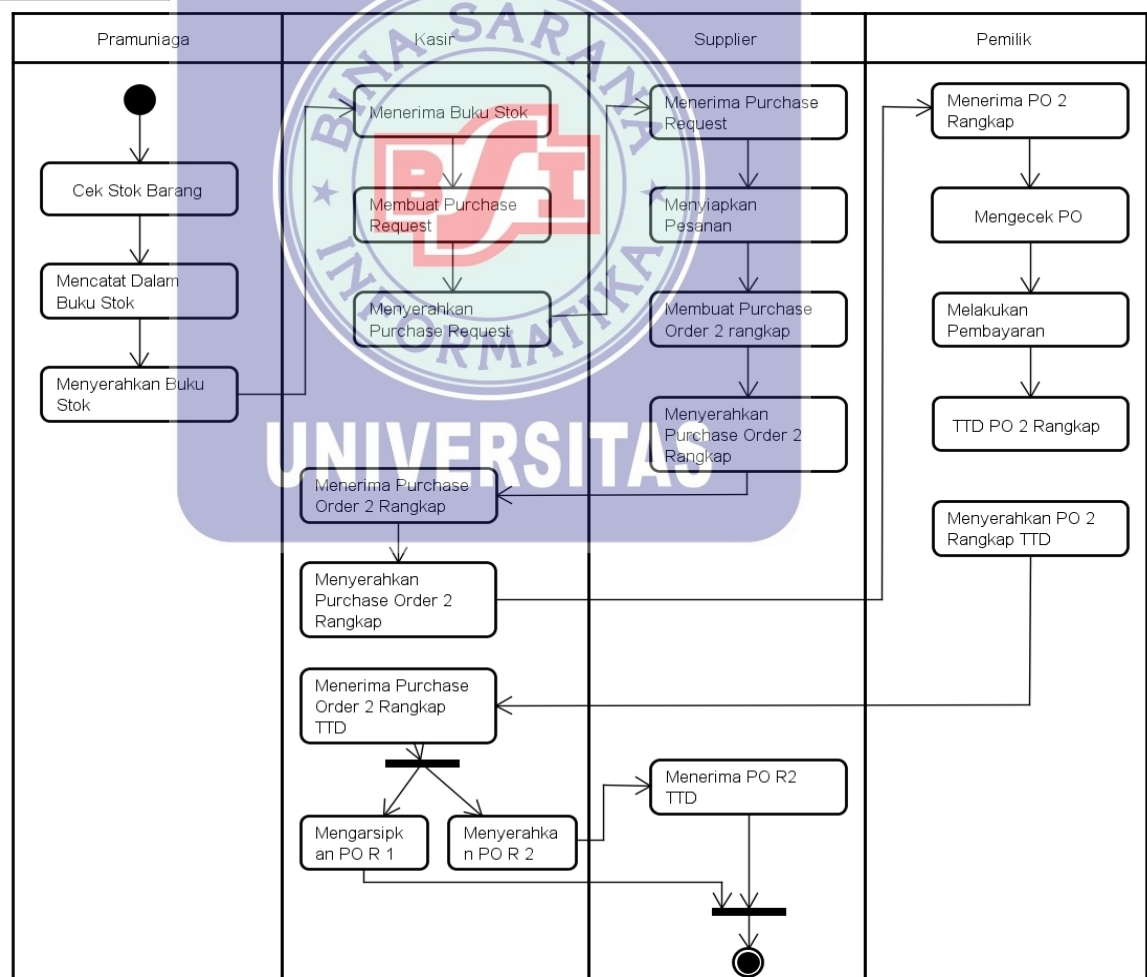
Setelah barang siap, *supplier* akan mengirimkan barang dan membuatkan *Purchase Order* (PO) 2 rangkap. Setelah barang sampai pihak *supplier* akan memberikan surat jalan kepada kasir untuk melakukan pengecekan barang. Kemudian kasir akan memberitahukan kepada pemilik bahwa barang yang dipesan telah datang dan memberikan surat jalan tersebut. Pemilik melakukan

pembayaran dan menandatangani *purchase Order* (PO) sebagai bukti bahwa pemesanan barang telah selesai. Rangkap 1 akan di simpan oleh pemilik sebagai bukti laporan pembelian dan rangkap 2 akan diberikan kepada *supplier*.

4. Prosedur laporan

Bagian kasir membuat laporan pembelian berdasarkan arsip pesanan dan arsip pembayaran untuk diserahkan kepada pemilik perusahaan.

3.2.2. Activity Diagram



Gambar III.2.

Activity Diagram Pembelian TB. Usaha Jaya Cikampek

3.2.3. Dokumen Masukan

Dokumen masukan adalah dokumen yang dimasukkan kedalam sistem untuk diproses agar sistem dapat beroperasi sehingga akan menghasilkan keluaran. Adapun dokumen masukan TB. Usaha Jaya Cikampek adalah sebagai berikut:

1. Nama Dokumen : *Purchase Request (PR)*
- Fungsi : Sebagai bukti permintaan pembelian barang
- Sumber : Bagian Kasir
- Tujuan : Supplier
- Jumlah : 1 Lembar
- Frekuensi : Setiap permintaan pembelian barang
- Media : Kertas
- Bentuk : lampiran A.1

3.2.4. Dokumen Keluaran

Dokumen keluaran adalah segala bentuk dokumen dari hasil proses pengolahan dokumen masukan yang terjadi pada suatu sistem. Adapun dokumen keluaran TB. Usaha Jaya Cikampek adalah sebagai berikut:

1. Nama Dokumen : *Purchase Order*
- Fungsi : Sebagai bukti laporan pemesanan barang
- Sumber : *Supplier*
- Tujuan : Kasir
- Jumlah : 1 Lembar
- Frekuensi : Setiap ada pemesanan barang
- Media : Kertas
- Bentuk : Lampiran B.1

3.2.5. Permasalahan Pokok

Sistem berjalan pada TB. Usaha Jaya Cikampek, penulis mencoba memberikan suatu analisa tentang sistem tersebut. Dalam hal ini penulis menemukan beberapa permasalahan yang timbul dan harus segera diadakan pemecahan, agar tidak menghambat kegiatan dalam hal menyediakan informasi yang lebih akurat khususnya dalam penanganan masalah pembelian tunai. Karena dalam penanganan pembelian tunai masih dilakukan dengan cara sederhana atau belum terkomputerisasi.

Permasalahan yang terjadi antara lain :

1. Pengecekan stok barang sering terjadi perbedaan antara stok yang tersedia digudang dengan stok yang tertulis di buku stok.
2. Pencatatan pemesanan barang sering mengalami keterlambatan karena masih menggunakan cara yang sederhana, serta dalam perhitungan total pembelian sering terjadi kesalahan karena masih menggunakan kalkulator.
3. Proses pembuatan laporan nya sering terjadi kesalahan karena kwintansi pembelian sering hilang atau rusak.

3.2.6. Pemecahan Masalah

Dengan meningkatkan mutu pelayanan serta efisiensi pada TB. Usaha Jaya Cikampek, diperlukan adanya sistem yang mendukung dan mampu mengatasi masalah-masalah diatas. Maka penulis mengusulkan dengan menggunakan sistem terkomputerisasi agar sistem berjalan lebih efektif dan efisien serta tidak membutuhkan waktu lama dalam proses pengoperasian nya. Serta penggunaan *database* agar proses penyimpanan dan pencarian data lebih mudah dan lebih tertata dengan rapih.

3.3. Analisis Kebutuhan *Software*

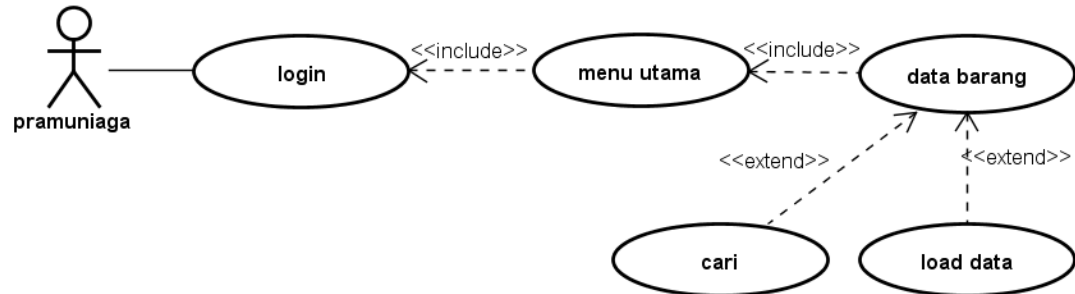
3.3.1. Analisis Kebutuhan

Analisa kebutuhan *software* untuk proram pembelian tunai pada TB. Usaha Jaya Cikampek diusulkan untuk beberapa prosedur yaitu:

1. Bagian Pramuniaga
 - A.1. Bagian Pramuniaga dapat melakukan *Login*
 - A.2. Bagian Pramuniaga dapat mengakses Data Barang
2. Bagian Kasir
 - B.1. Bagian Kasir dapat melakukan *Login*
 - B.2. Bagian Kasir dapat mengelola Data Barang
 - B.3. Bagian Kasir dapat mengelola Data *Purchase Request*
 - B.4. Bagian Kasir dapat mengelola Data *Purchase Order*
 - B.5. Bagian Kasir dapat mengelola Data *Supplier*
 - B.6. Bagian Kasir dapat mengelola Data perkiraan
 - B.7. Bagian Kasir dapat mengelola Data Jurnal
 - B.8. Bagian Kasir dapat mengelola Laporan
3. Bagian Pemilik Toko
 - C.1. Bagian Pemilik dapat melakukan *Login*
 - C.2. Bagian Pemilik dapat mengakses Data Barang
 - C.3. Bagian Pemilik dapat mengakses Data Jurnal
 - C.4. Bagian pemilik dapat mengakses Laporan

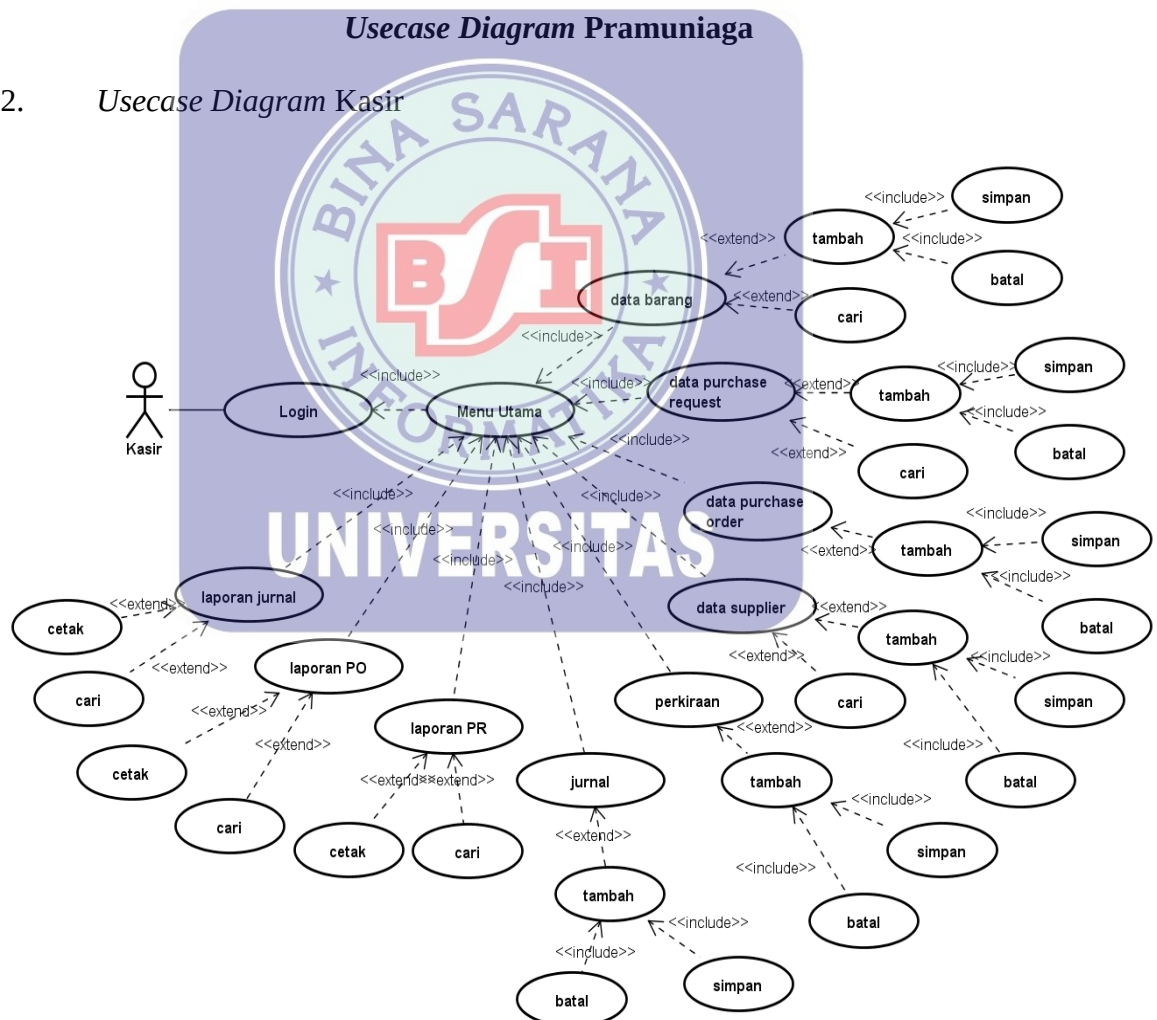
3.3.2. Usecase Diagram

1. Usecase Diagram Pramuniaga



Gambar III.3.

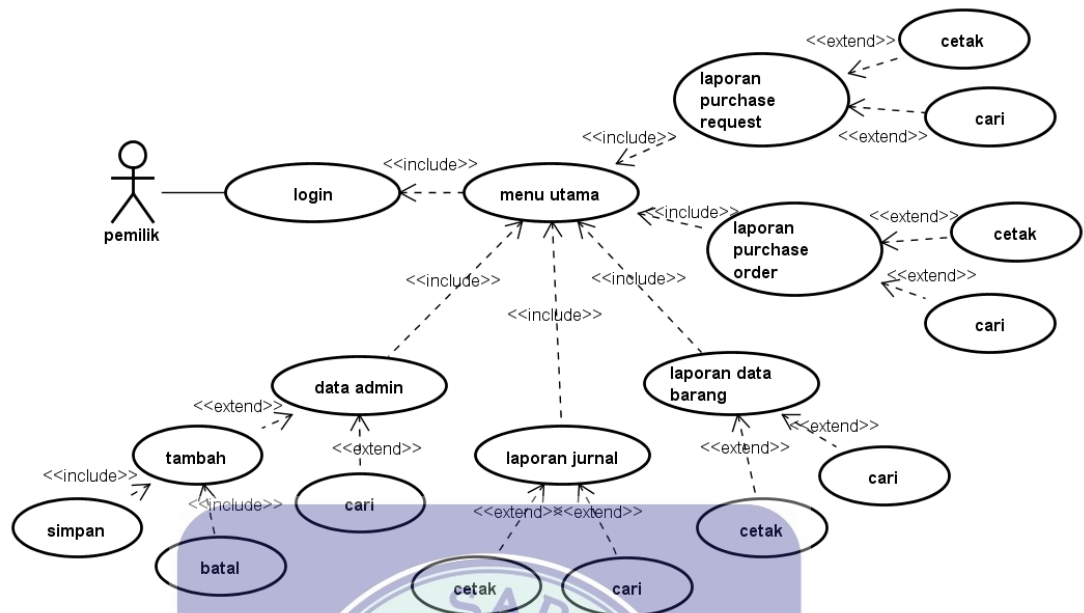
2. Usecase Diagram Kasir



Gambar III.4.

Usecase Diagram Kasir

3. Usecase Diagram Pemilik



Gambar III.5.

Usecase Diagram Pemilik

Definisi aktor yang ada pada usecase diatas :

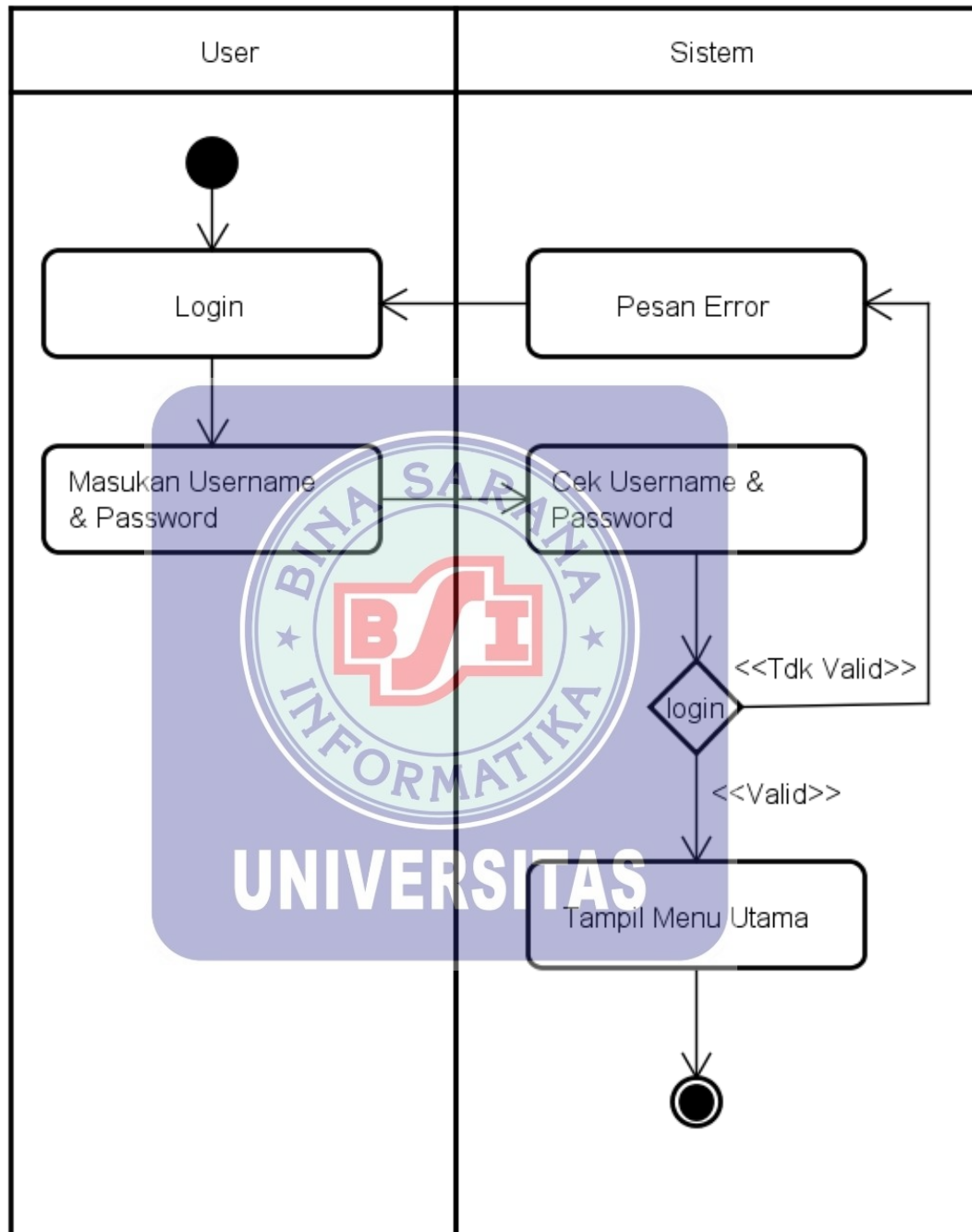
Tabel III.1.

Pendeskripsian Aktor TB. Usaha Jaya Cikampek

No	Aktor	Deskripsi
1.	Kasir	Orang yang bertugas dan memiliki hak akses untuk melakukan operasi pengelolaan data barang, <i>supplier</i> , <i>purchase request</i> , <i>purchase order</i> , perkiraan, jurnal dan laporan.
2.	Pramuniaga	Pramuniaga adalah orang yang melayani penjualan pelanggan dan melakukan cek stok barang digudang.
3	Pemilik Toko	Bertugas melakukan kontrol kerja kasir dan pramuniaga. Melakukan cek stok barang, jurnal, laporan <i>purchase request</i> dan <i>purchase order</i> .

3.3.3. Activity Diagram

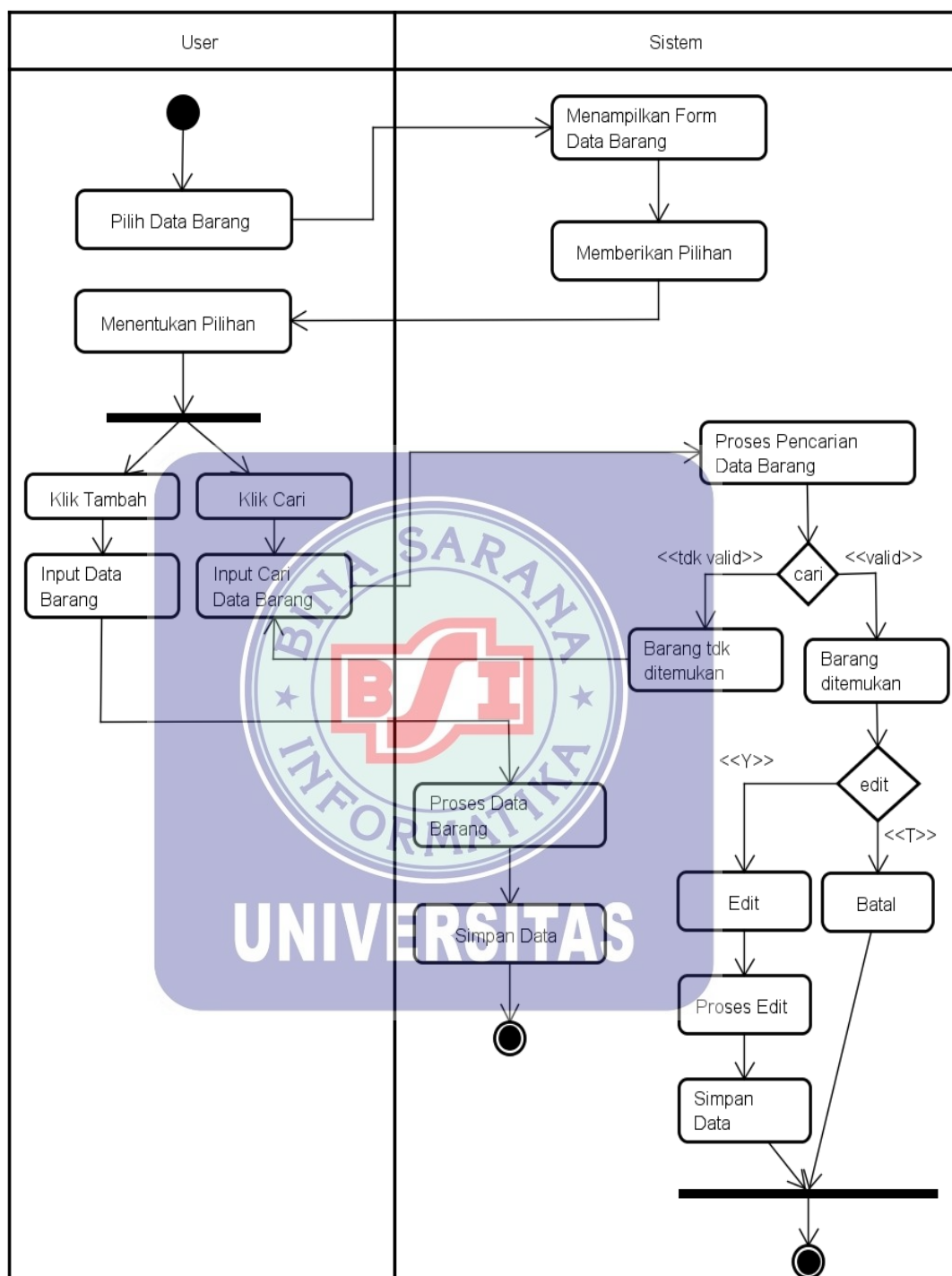
1. Activity Diagram Bagian login User



Gambar III.6.

Activity Diagram Login User

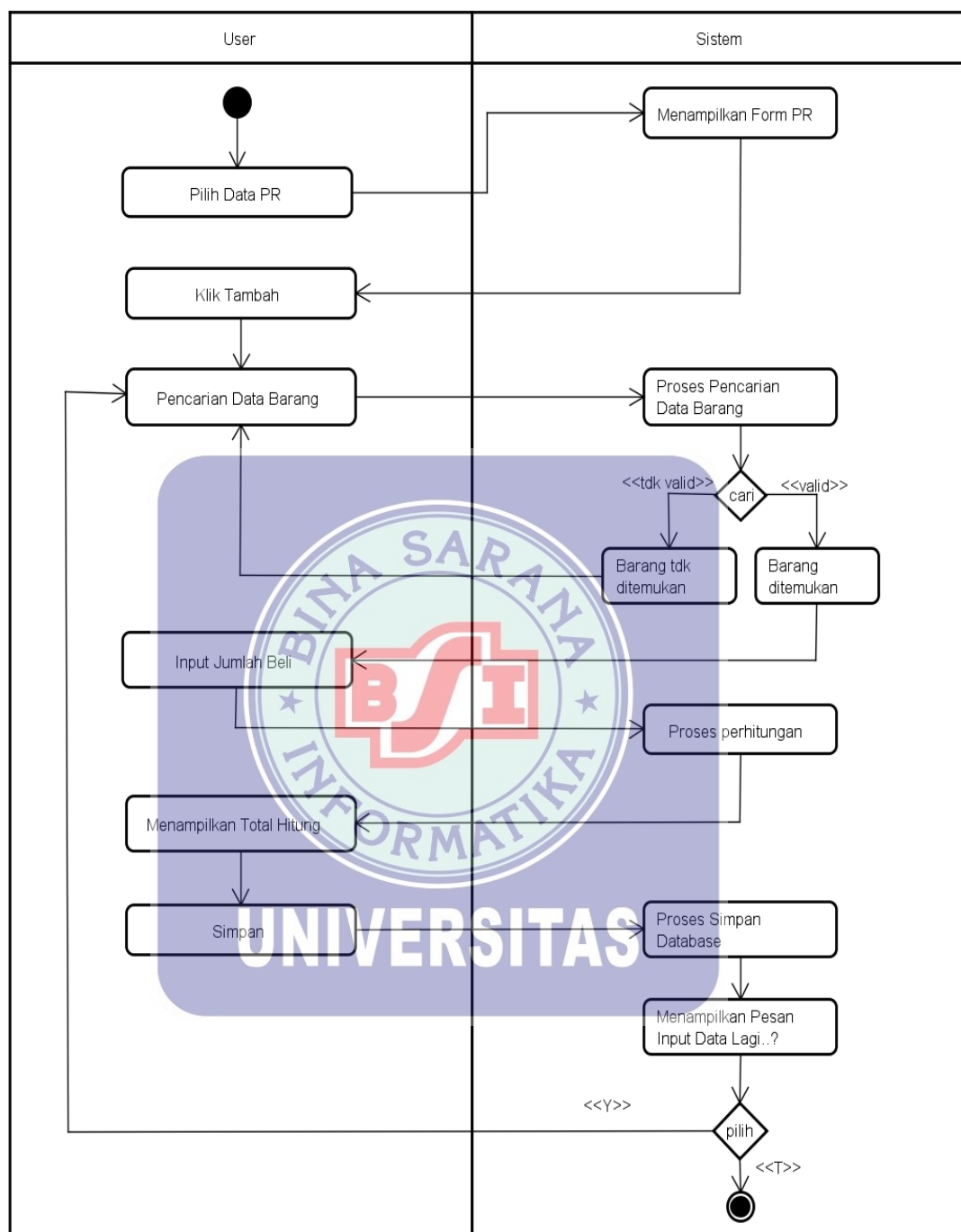
2. Activity Diagram Mengelola Data Barang



Gambar III.7.

Activity Diagram Kelola Data Barang

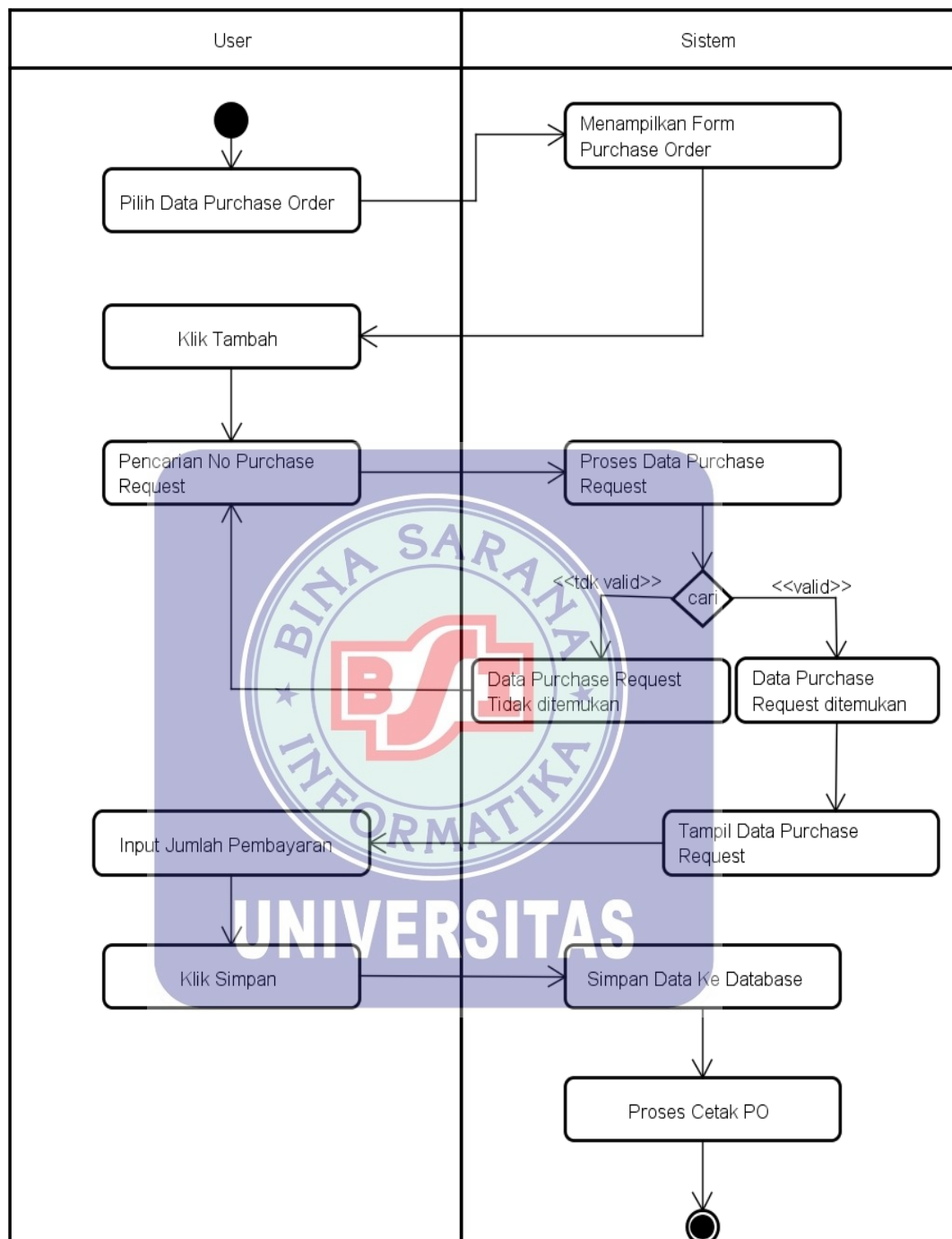
3. Activity Diagram Kelola Purchase Request (PR)



Gambar III.8.

Activity Diagram Kelola Data Purchase Request

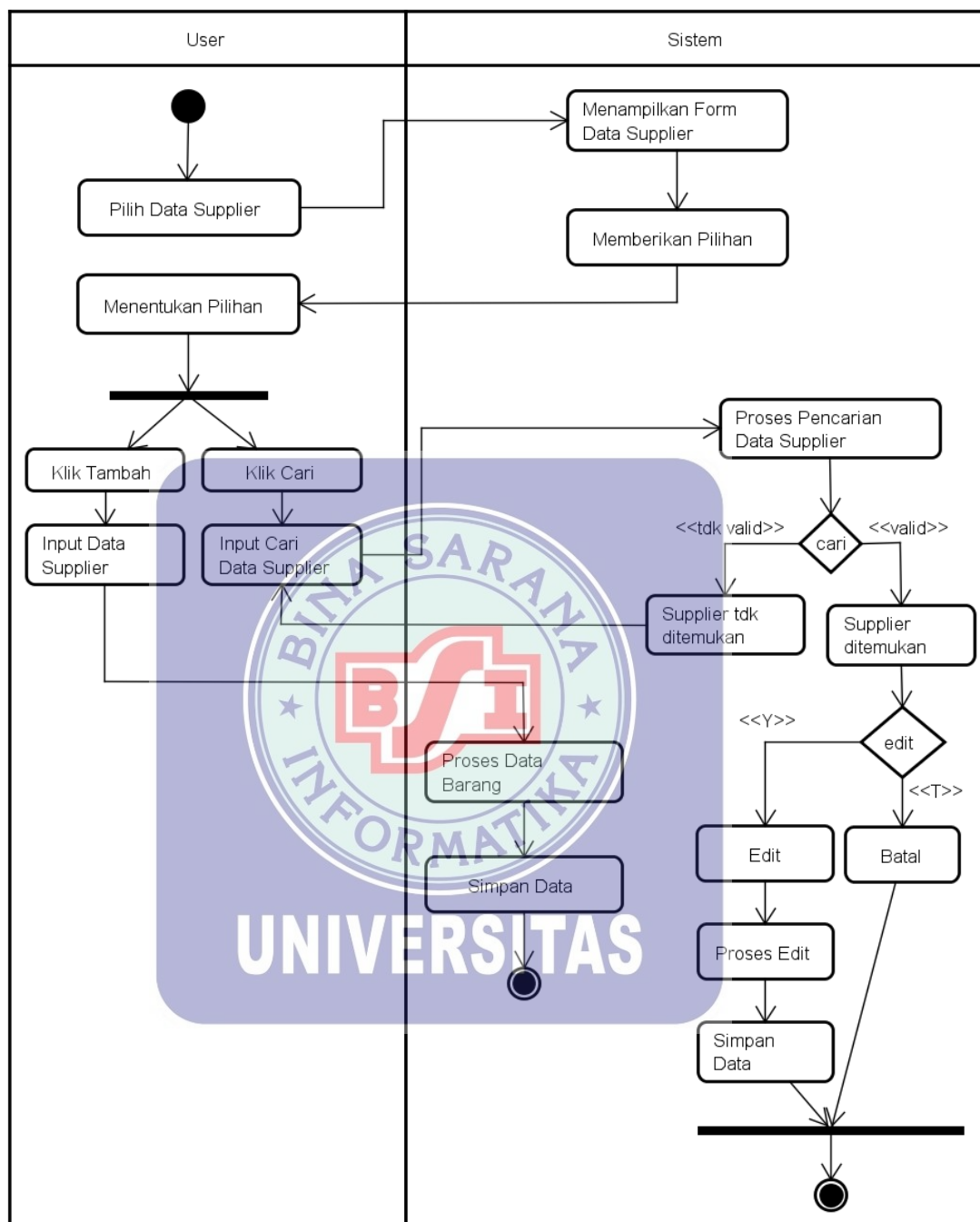
4. *Activity Diagram Kelola Data Purchase Order (PO)*



Gambar III.9.

Activity Diagram Kelola Data Purchase Order

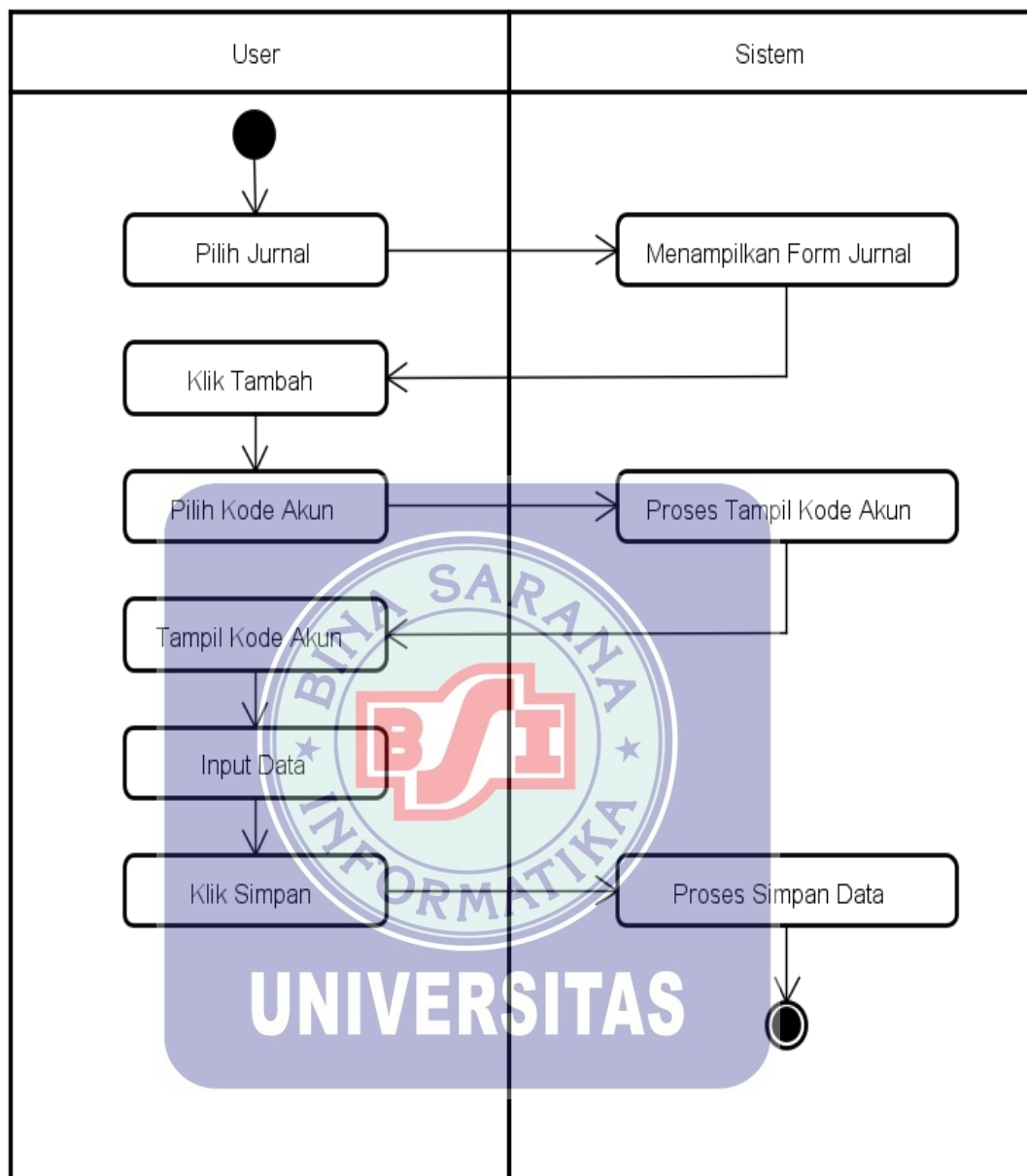
5. Activity Diagram Kelola Data Supplier



Gambar III.10.

Activity Diagram Kelola Data Supplier

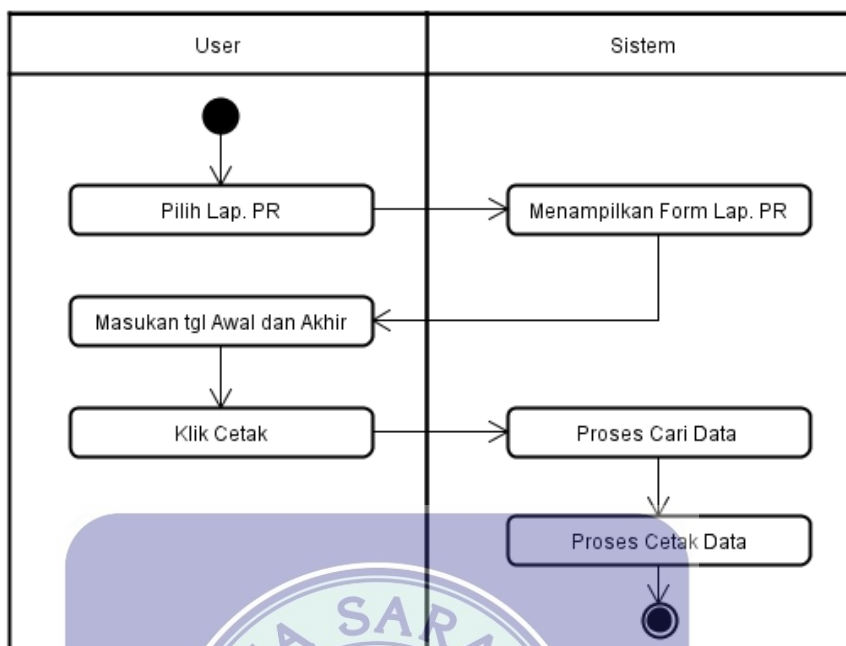
6. *Activity Diagram Kelola Data Jurnal*



Gambar III.11.

Activity Diagram Kelola Data Jurnal

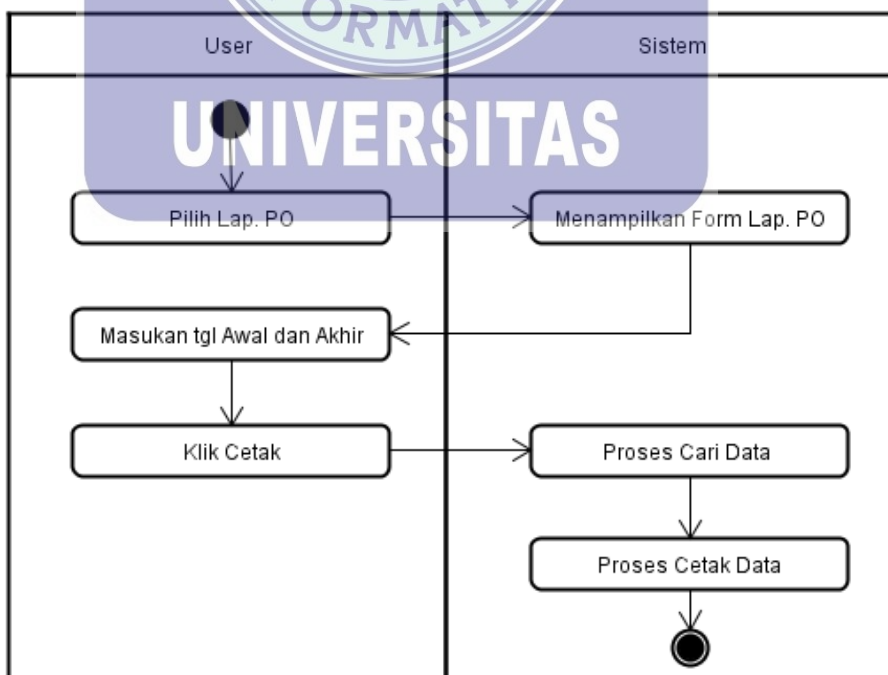
7. *Activity Diagram Kelola Laporan Purchase Request (PR)*



Gambar III.12.

Activity Diagram Kelola Laporan Purchase Request

8. *Activity Diagram Kelola Laporan Purchase Order (PO)*

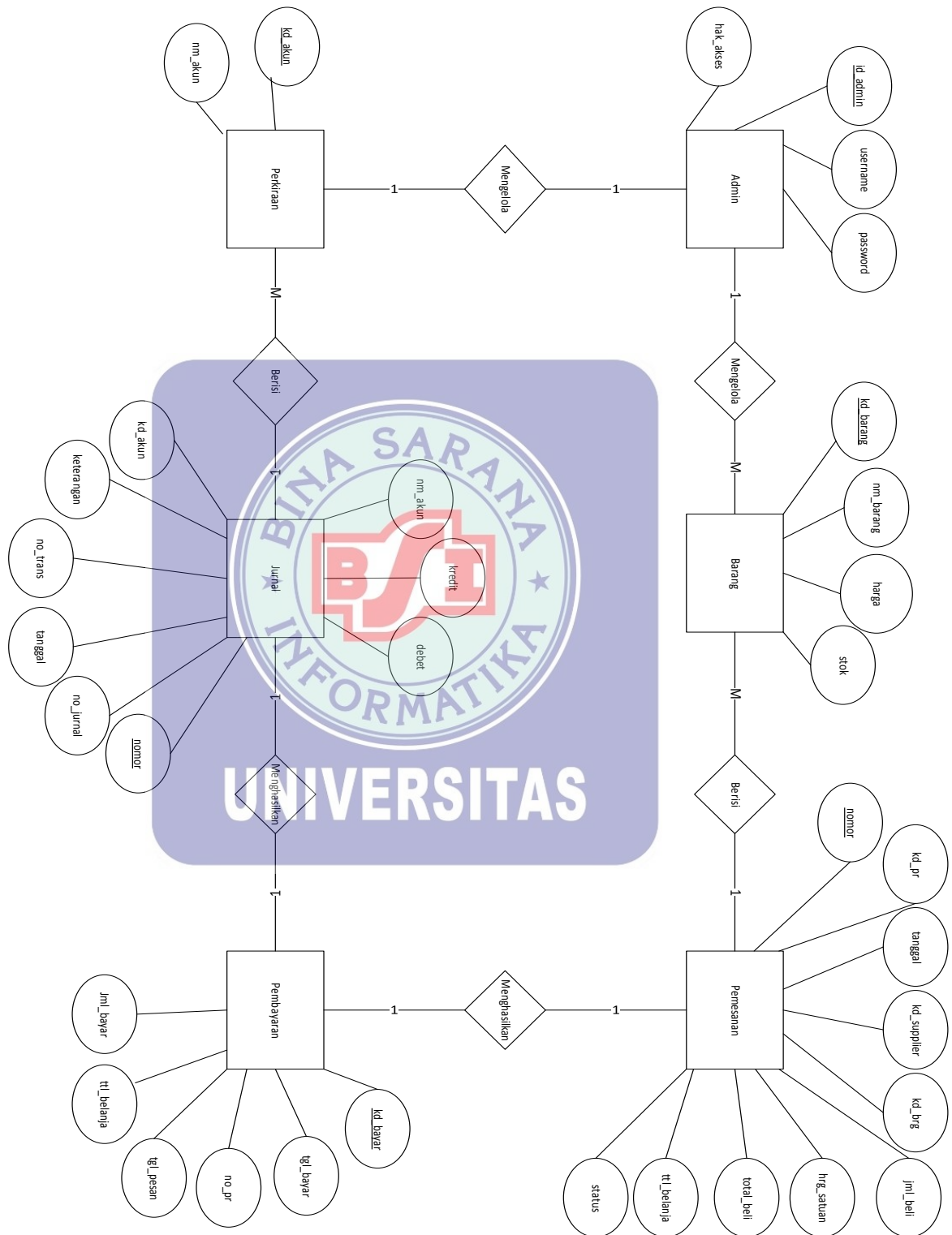


Gambar III.13.

Activity Diagram Kelola Laporan Purchase Order

3.4. Desain

3.4.1. Entity Relationship Diagram (ERD)

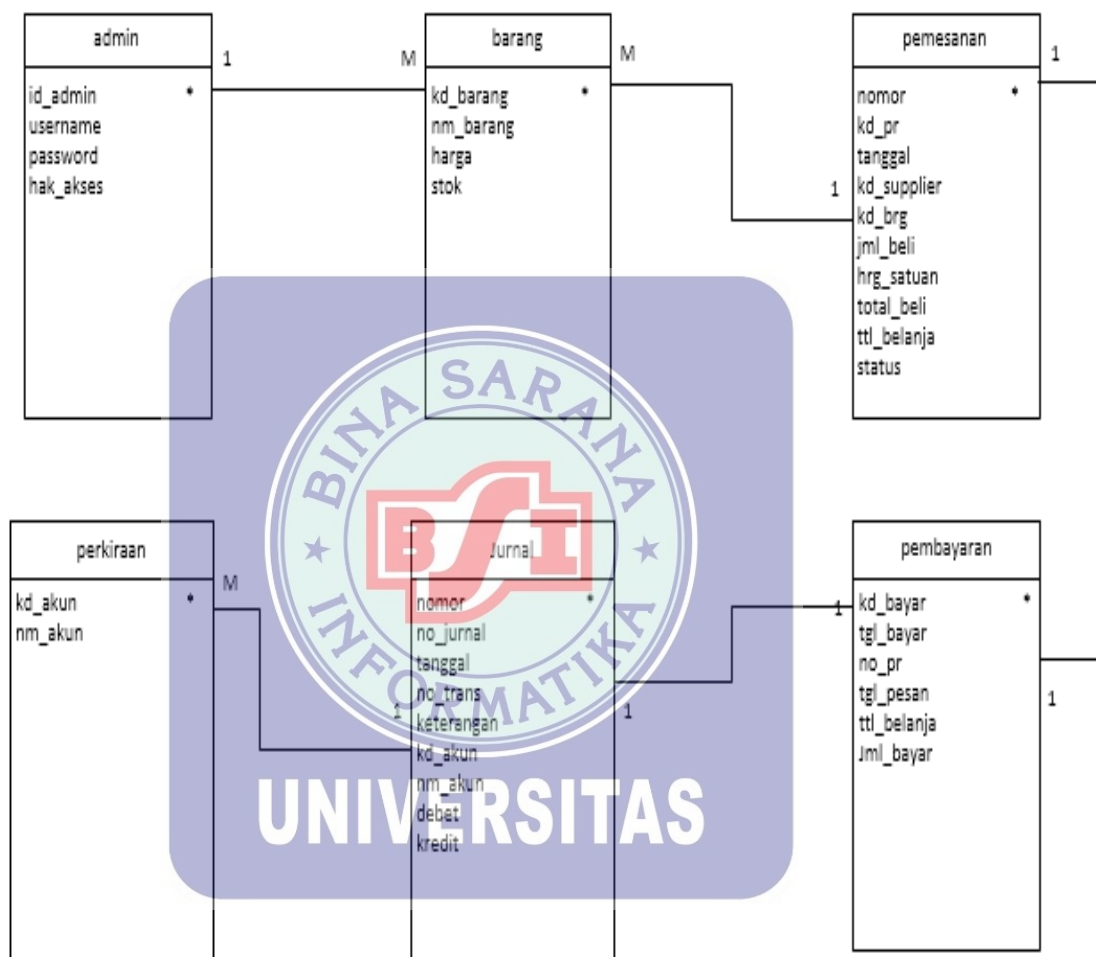


Gambar III.14.

Entity Relationship Diagram (ERD)

3.4.2. Logical Record Structure (LRS)

LRS adalah representasi dari struktur record – record pada tabel – tabel yang terbentuk dari hasil antar himpunan entitas. Menentukan kardinalitas, jumlah tabel, dan *Foreign Key* (FK). Berikut adalah gambaran LRS yang sudah dibuat:



Gambar III.15.

Logical Record Structure (LRS)

3.4.3. Spesifikasi File

File adalah komponen utama dari sebuah sistem komputerisasi. *File* memiliki fungsi sebagai data masukan dan keluaran untuk menunjang jalan nya sistem dan program yang sedang dibangun. Adapun spesifikasi *file* sebagai berikut:

1. Spesifikasi File Admin

Nama Database : pembelian_matrrial

Nama File : admin

Tipe File : File Master

Akses File : Random

Panjang Record : 108

Kunci File : id_admin

Tabel III.2.
Spesifikasi File Admin

No	Elemen Data	Nama Field	Type	Panjang	Keterangan
1	Id Petugas	Id_admin	Integer	8	Primary Key
2	Username	Username	Varchar	50	
3	Password	password	Varchar	30	
4	Hak Akses	hak_akses	Varchar	20	

2. Spesifikasi File Barang

Nama Database : pembelian_matrrial

Nama File : barang

Tipe File : File Master

Akses File : Random

Panjang Record : 88

Kunci File : kd_barang

Tabel III.3.

Spesifikasi *File* Barang

No	Elemen Data	Nama Field	Type	Panjang	Keterangan
1	Kode Barang	kd_barang	Integer	8	Primary Key
2	Nama Barang	nm_barang	Varchar	60	
3	Harga	Harga	Integer	15	
4	Stok	Stok	Integer	5	

3. Spesifikasi *File* Jurnal

Nama Database : pembelian_matrrial

Nama File : jurnal

Tipe File : File Jurnal

Akses File : Random

Panjang Record : 94

Kunci File : nomor

Tabel III.4.

Spesifikasi *File* Jurnal

No	Elemen Data	Nama Field	Type	Panjang	Keterangan
1	Nomor	nomor	Integer	5	Primary Key
2	No Jurnal	no_jurnal	Varchar	10	
3	Tanggal	tanggal	Date		
4	No Transaksi	no_trans	Varchar	20	
5	Keterangan	keterangan	Text		
6	Kode Akun	kd_akun	Varchar	9	
7	Nama Akun	nm_akun	Varchar	30	
8	Debet	debet	Integer	10	
9	Kredit	Kredit	Integer	10	

4. Spesifikasi *File* Perkiraan

Nama Database : pembelian_matrrial

Nama File : perkiraan

Tipe File : File Perkiraan

Akses File : Random

Panjang Record : 48

Kunci File : kd_akun

Tabel III.5.

Spesifikasi File Perkiraan

No	Elemen Data	Nama Field	Type	Panjang	Keterangan
1	Kode Akun	kd_akun	Integer	8	Primary Key
2	Nama Akun	nm_akun	Varchar	40	

5. Spesifikasi File Pemesanan

Nama Database : pembelian_matrial

Nama File : pemesanan

Tipe File : File Transaksi

Akses File : Random

Panjang Record : 95

Kunci File : nomor

Tabel III.6.

Spesifikasi File Pemesanan

No	Elemen Data	Nama Field	Type	Panjang	Keterangan
1	Nomor	Nomor	Integer	8	Primary Key
2	Kode PR	kd_pr	Varchar	8	
3	Tanggal	tanggal	Date		
4	Kode Supplier	kd_supplier	Varchar	8	
5	Kode Barang	kd_brg	Varchar	8	
6	Jumlah Beli	jml_beli	Integer	5	
7	Harga Satuan	hrng_satuan	Integer	9	
8	Total Beli	total_beli	Integer	9	
9	Total Belanja	ttd_belanja	Integer	10	
10	Status	status	Varchar	30	

6. Spesifikasi File Pembayaran

Nama Database : pembelian_matrial

Nama File : pembayaran

Tipe File : File Transaksi

Akses File : Random

Panjang Record : 38

Kunci File : kd_bayar

Tabel III.7.

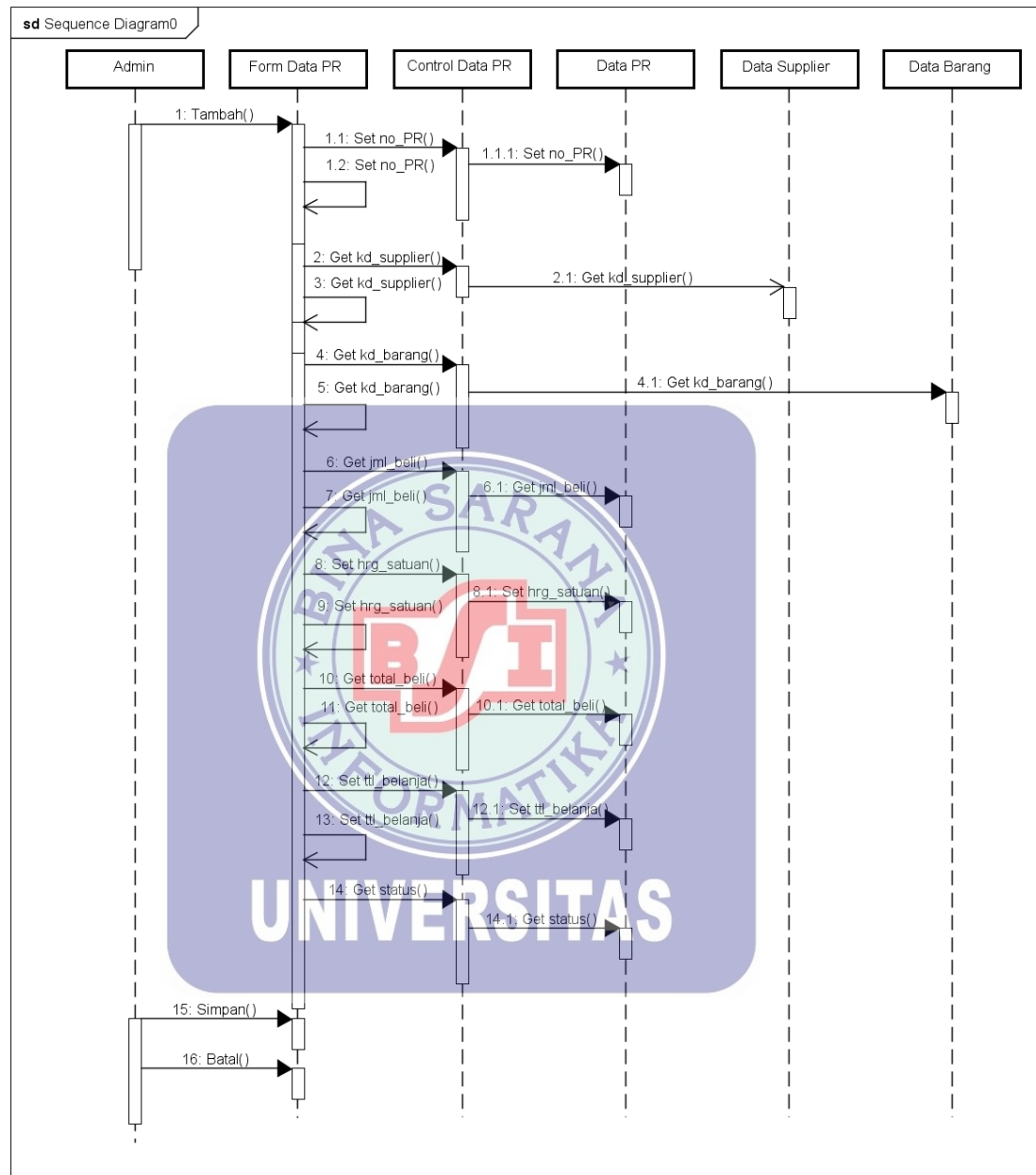
Spesifikasi File Pembayaran

No	Elemen Data	Nama Field	Type	Panjang	Keterangan
1	Kode Pembayaran	kd_bayar	Varchar	10	Primary Key
2	Tanggal Bayar	tgl_bayar	Date		
3	Nomor PR	no_pr	Varchar	10	
4	Tanggal Pesan	tgl_pesan	Date		
5	Total Belanja	tgl_belanja	Integer	9	
6	Jumlah Bayar	jml_bayar	Integer	9	



3.4.4. Sequence Diagram

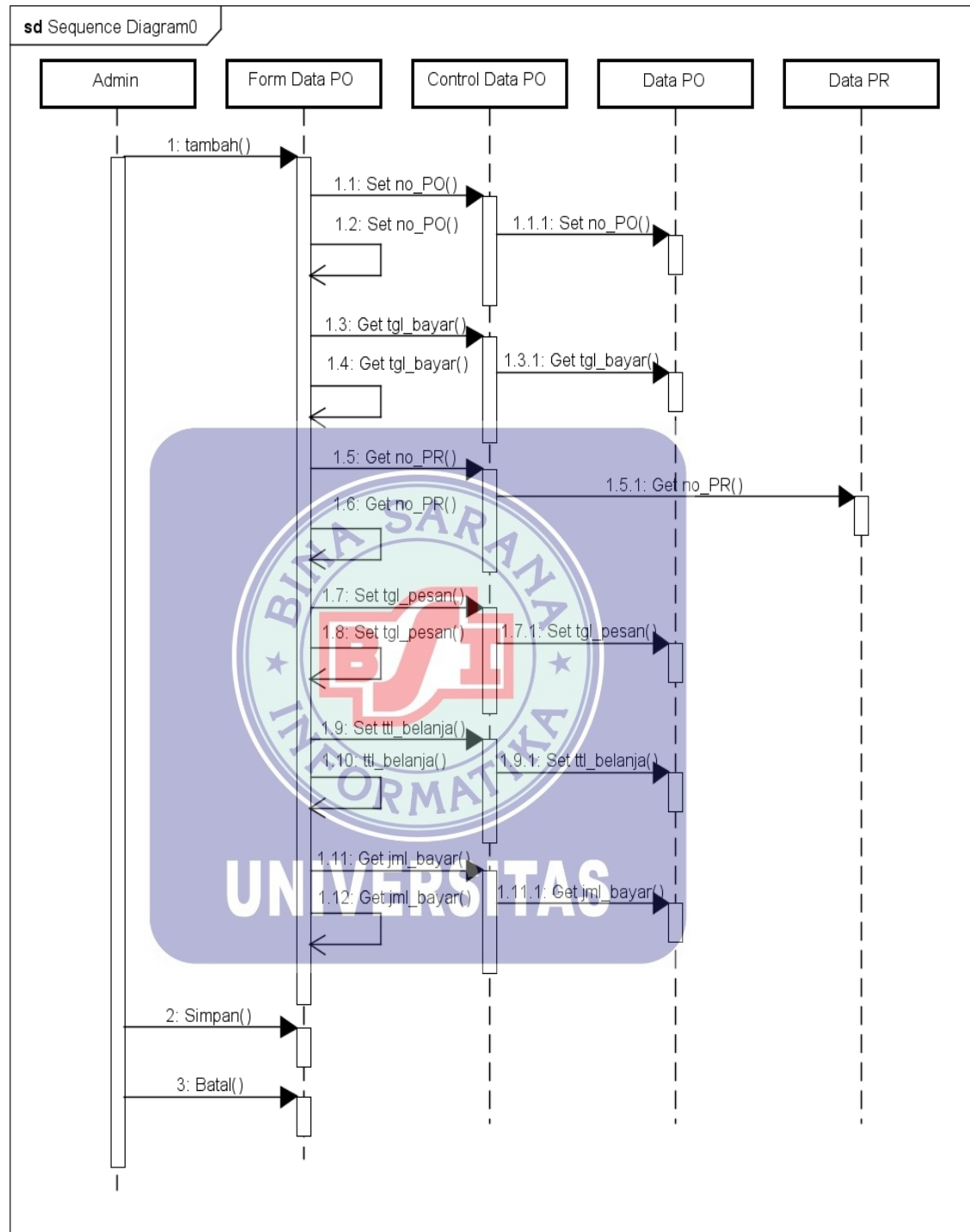
1. Sequence Diagram Purchase Request



Gambar III.16.

Sequence Diagram Purchase Request

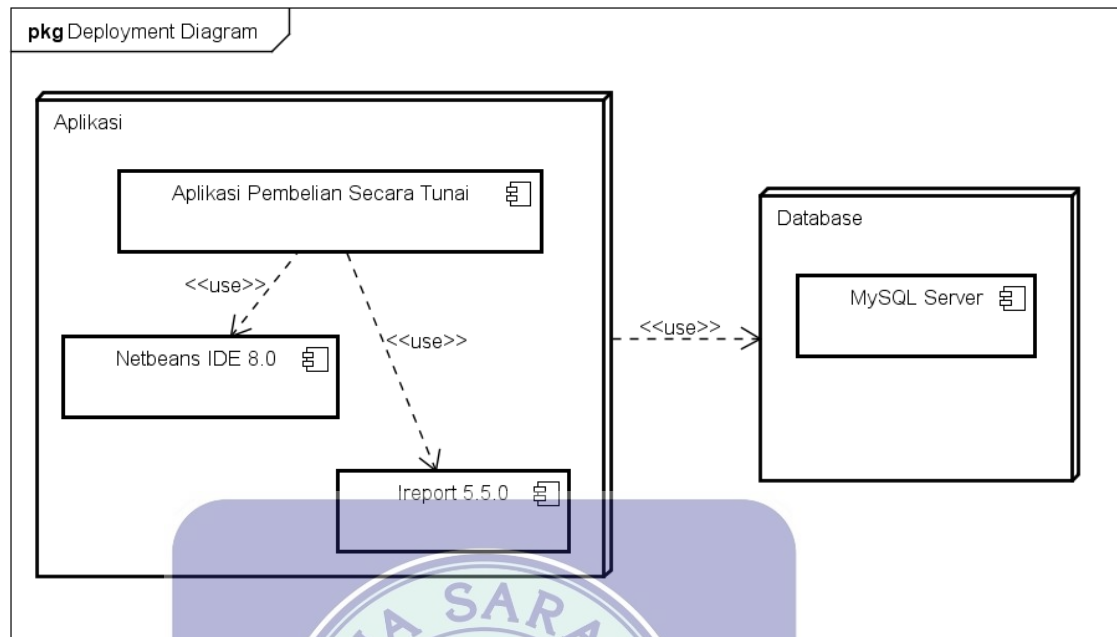
2. Sequence Diagram Purchase Order (Pembayaran)



Gambar III.17.

Sequence Diagram Purchase Order

3.4.5. Deployment Diagram



Gambar III.18.
Deployment Diagram Pembelian Tunai

3.4.6. User Interface

Berikut tampilan *user interface* dari rancangan program yang dibuat:

1. *Form Login*

The screenshot shows a login form titled '..:Form Login:.' on a teal background. It includes two input fields: 'Username : ' and 'Password : '. Below these fields is a small icon of a computer monitor with a padlock. At the bottom, there are two buttons: 'Login' and 'Keluar' (Exit).

Gambar III.19.

Form Login

2. *Form Data Barang*

.:Form Data Barang:.

Kode Barang : Stok :

Nama Barang : Harga Satuan Rp :

Masukan ID / NAMA :

Title 1	Title 2	Title 3	Title 4

Gambar III.20.

Form Kelola Data Barang

3. *Form Purchase Request*

.:Form Purchase Request:.

Kode PR : Tanggal PR :

Kode Supplier :

Input Data

Kode Barang :

Nama Barang :

Harga Satuan Rp:

Jumlah Beli :

Total Rp: 0,

Data Barang

Cari Kode / Nama Barang :

Title 1	Title 2	Title 3	Title 4

Data Purchase Request

Title 1	Title 2	Title 3	Title 4	Title 5	Title 6	Title 7

Gambar III.21.

Form Purchase Request

4. *Form Purchase Order (Pembayaran)*

..Form Pembayaran (PO)..

Kode Pembayaran PO : Tanggal Pemesanan :

Tanggal Pembayaran : Total Belanja Rp:

No Purchase Request : Jumlah Bayar Rp:

Data Pembayaran

Cari Data Pembayaran :

Title 1	Title 2	Title 3	Title 4	Title 5	Title 6

Gambar III.22.

Form Purchase Order

5. *Form Jurnal Umum*

..Data Jurnal Umum..

No Jurnal :

Tanggal :

No Transaksi :

Keterangan :

Title 1	Title 2	Title 3	Title 4	Title 5	Title 6

Kode Akun :	Kode Akun :
Nama Akun :	Nama Akun :
Debet :	Kredit :

Gambar III.23.

Form Jurnal Umum

6. *Form Laporan Purchase Request*

..:Laporan Purchase Request:..

Data Purchase Request

Cari Kode / tanggal Transaksi :

Title 1	Title 2	Title 3	Title 4	Title 5	Title 6	Title 7

..:Cetak Laporan:..

Periode Awal : yyyy-MM-dd

Periode Akhir : yyyy-MM-dd



Gambar III.24.

Form Laporan Purchase Request

7. *Form Laporan Purchase Order*

..:Laporan Purchase Request:..

Data Pembayaran PO

Cari Kode / tanggal Transaksi :

Title 1	Title 2	Title 3	Title 4	Title 5	Title 6

..:Cetak Laporan:..

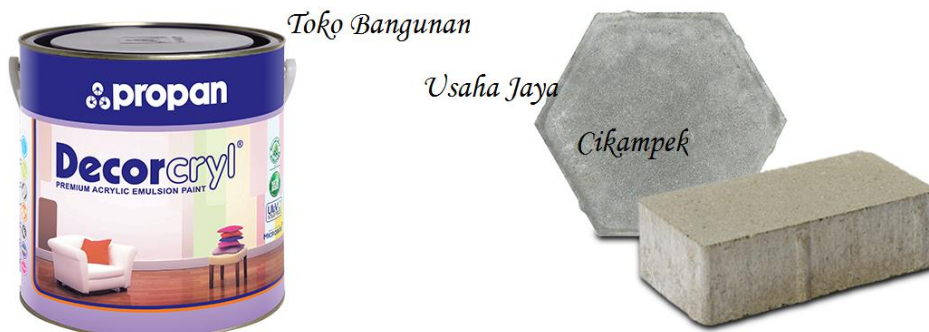
Periode Awal : yyyy-MM-dd

Periode Akhir : yyyy-MM-dd

Gambar III.25.

Form Laporan Purchase Order

8. Form Menu Utama



Gambar III.26.

Form Menu Utama

9. Form Laporan Jurnal Umum

..:Laporan Jurnal Umum:..

Data Jurnal Umum

Cari Kode / tanggal Transaksi : Cari Load Data

Title 1	Title 2	Title 3	Title 4	Title 5	Title 6

..:Cetak Laporan:..

Periode Awal : yyyy-MM-dd

Periode Akhir : yyyy-MM-dd

Cetak

Gambar III.27.

Form Laporan Jurnal Umum

3.5. Implementasi

Untuk mengimplementasikan program agar berjalan dengan baik, maka diperlukan spesifikasi sebagai berikut:

1. Bahasa Pemrograman

Program : netbeans 8.1
 Support : javaFX, web, PHP, C/C++
 Minimum resolusi monitor : 1024 x 768 pixel

2. Database

Program : xampp 3.2.2
 Bahasa pemrograman : PHP
 Server : apache
 Database : MySQL
 Ukuran : windows 7 x64 - 121MB

3. Minimum Hardware

Processor : 800MHz Intel Pentium III
 Memory : 512MB

Disk Space : 750MB free

4. Minimum Software

Sistem Operasi : windows 7 profesional

3.5.1. Code Generation

1. Kodingan Form Purchase Order

```
package program;  
  
import Database.KoneksiDatabase;
```



```

import Database.ResultSetTable;

import java.awt.event.KeyEvent;

import java.io.File;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.HashMap;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.engine.util.JRLoader;

import net.sf.jasperreports.view.JasperViewer;

```

```

public class form_pembayaran extends javax.swing.JFrame {

    String idPr;

    ResultSet rs;

    KoneksiDatabase con;

    /**
     * Creates new form form_pembayaran
     */

    public form_pembayaran() {

```

```

con = new KoneksiDatabase(new Database.Parameter().HOST_DB, new
Database.Parameter().USERNAME_DB, new
Database.Parameter().PASSWORD_DB);

initComponents();

setTitle("form pembayaran order");

loadtabel();

nonaktif();

kd_po.setEnabled(false);

jDateChooser1.setEnabled(false);

titik.setEnabled(false);

tambah.setEnabled(true);

batal.setEnabled(false);

simpan.setEnabled(false);
}

private String id() {
    String no = null;
    try {
        con.koneksiDatabase();

        String sql = "Select right(kd_bayar,4)+1 from pembayaran ";

        ResultSet rs = con.eksekusiQuery(sql);

        if (rs.next()) {
            rs.last();

            no = rs.getString(1);

            while (no.length() < 4) {
                no = "000" + no;
            }
        }
    }
}

```

```

        no = "B" + no;

        kd_po.setText(no);

    }

    } else {

        no = "B0001";

        kd_po.setText(no);

    }

    } catch (Exception e) {

    }

    return no;

}

public void itemTerpilih() {

    data_pr FPR = new data_pr();

    FPR.FBY = this;

    no_pr.setText(idPr);

    ambildata();

}

```

```
@SuppressWarnings("unchecked")
```

```
// <editor-fold defaultstate="collapsed" desc="Generated Code">
```

```

private void initComponents() {

    jPanel1 = new javax.swing.JPanel();

    jLabel1 = new javax.swing.JLabel();

    jLabel20 = new javax.swing.JLabel();

    jLabel21 = new javax.swing.JLabel();

    jLabel22 = new javax.swing.JLabel();

```

```
jLabel23 = new javax.swing.JLabel();

jPanel2 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

tbl_bayar = new javax.swing.JTable();

jLabel32 = new javax.swing.JLabel();

jLabel33 = new javax.swing.JLabel();

tcari = new javax.swing.JTextField();

cari = new javax.swing.JButton();

refresh = new javax.swing.JButton();

jLabel24 = new javax.swing.JLabel();

jLabel25 = new javax.swing.JLabel();

kd_po = new javax.swing.JTextField();

no_pr = new javax.swing.JTextField();

titik = new javax.swing.JButton();

jDateChooser1 = new com.toedter.calendar.JDateChooser();

jLabel26 = new javax.swing.JLabel();

jLabel27 = new javax.swing.JLabel();

jLabel28 = new javax.swing.JLabel();

jLabel29 = new javax.swing.JLabel();

jLabel30 = new javax.swing.JLabel();

jLabel31 = new javax.swing.JLabel();

tgl_pesan = new javax.swing.JTextField();

total_b = new javax.swing.JTextField();

bayar = new javax.swing.JTextField();

batal = new javax.swing.JButton();
```

```

        tambah = new javax.swing.JButton();

        simpan = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

        getContentPane().setLayout(new
org.netbeans.lib.awtextra.AbsoluteLayout());

        jPanel1.setBackground(new java.awt.Color(0, 153, 153));

        jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

        jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel1.setFont(new java.awt.Font("Tahoma", 0, 28)); // NOI18N
        jLabel1.setText("Form Pembayaran (PO).");
        jPanel1.add(jLabel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(490, 20, -1, -1));

        jLabel20.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
        jLabel20.setText("Kode Pembayaran PO");
        jPanel1.add(jLabel20, new
org.netbeans.lib.awtextra.AbsoluteConstraints(190, 100, -1, -1));

        jLabel21.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
        jLabel21.setText("");
        jPanel1.add(jLabel21, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 100, 10, -1));

        jLabel22.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
        jLabel22.setText("Tanggal Pembayaran");

```

```

jPanel1.add(jLabel22, new
org.netbeans.lib.awtextra.AbsoluteConstraints(190, 150, -1, -1));

jLabel23.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel23.setText(":");
jPanel1.add(jLabel23, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 150, 10, -1));

jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swin
g.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)), "Data
Pembayaran",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Tahoma", 0, 14))); // NOI18N
jPanel2.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
tbl_bayar.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(tbl_bayar);

```

```

jPanel2.add(jScrollPane1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 80, 1050, 110));

jLabel32.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel32.setText("Cari Data Pembayaran");
jPanel2.add(jLabel32, new
org.netbeans.lib.awtextra.AbsoluteConstraints(180, 40, -1, -1));

jLabel33.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel33.setText(":");
jPanel2.add(jLabel33, new
org.netbeans.lib.awtextra.AbsoluteConstraints(320, 40, 10, -1));
jPanel2.add(tcari, new
org.netbeans.lib.awtextra.AbsoluteConstraints(350, 40, 400, -1));
cari.setText("Cari");
jPanel2.add(cari, new
org.netbeans.lib.awtextra.AbsoluteConstraints(770, 40, 80, -1));
refresh.setText("Refresh");
refresh.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        refreshActionPerformed(evt);
    }
});

jPanel2.add(refresh, new
org.netbeans.lib.awtextra.AbsoluteConstraints(860, 40, 80, -1));

jPanel1.add(jPanel2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 380, 1120, 220));

```

```

jLabel24.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel24.setText("No Purchase Request");

jPanel1.add(jLabel24, new
org.netbeans.lib.awtextra.AbsoluteConstraints(190, 200, -1, -1));

jLabel25.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel25.setText(":");

jPanel1.add(jLabel25, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 200, 10, -1));

jPanel1.add(kd_po, new
org.netbeans.lib.awtextra.AbsoluteConstraints(350, 100, 190, -1));

no_pr.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        no_prActionPerformed(evt);
    }
});

jPanel1.add(no_pr, new
org.netbeans.lib.awtextra.AbsoluteConstraints(350, 200, 110, -1));

titik.setText("<....>");

titik.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        titikActionPerformed(evt);
    }
});

jPanel1.add(titik, new
org.netbeans.lib.awtextra.AbsoluteConstraints(470, 200, 70, -1));

```



```

jPanel1.add(jDateChooser1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(350, 150, 190, -1));

jLabel26.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel26.setText("Tanggal Pemesanan");
jPanel1.add(jLabel26, new
org.netbeans.lib.awtextra.AbsoluteConstraints(630, 100, -1, -1));

jLabel27.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel27.setText(":");
jPanel1.add(jLabel27, new
org.netbeans.lib.awtextra.AbsoluteConstraints(770, 100, 10, -1));

jLabel28.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel28.setText("Total Belanja      Rp");
jPanel1.add(jLabel28, new
org.netbeans.lib.awtextra.AbsoluteConstraints(630, 150, -1, -1));

jLabel29.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel29.setText(":");
jPanel1.add(jLabel29, new
org.netbeans.lib.awtextra.AbsoluteConstraints(770, 150, 10, -1));

jLabel30.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel30.setText("Jumlah Bayar      Rp");
jPanel1.add(jLabel30, new
org.netbeans.lib.awtextra.AbsoluteConstraints(630, 200, -1, -1));

jLabel31.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel31.setText(":");

```

```

jPanel1.add(jLabel31, new
org.netbeans.lib.awtextra.AbsoluteConstraints(770, 200, 10, -1));

jPanel1.add(tgl_pesanan, new
org.netbeans.lib.awtextra.AbsoluteConstraints(790, 100, 200, -1));

total_b.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        total_bKeyTyped(evt);
    }
});

jPanel1.add(total_b, new
org.netbeans.lib.awtextra.AbsoluteConstraints(790, 150, 200, -1));

bayar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        bayarKeyTyped(evt);
    }
});

jPanel1.add(bayar, new
org.netbeans.lib.awtextra.AbsoluteConstraints(790, 200, 200, -1));

batal.setText("Batal");

batal.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        batalActionPerformed(evt);
    }
});

```

```

jPanel1.add(batal, new
org.netbeans.lib.awtextra.AbsoluteConstraints(780, 290, 210, 60));

tambah.setText("Tambah");

tambah.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tambahActionPerformed(evt);
    }
});

jPanel1.add(tambah, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 290, 210, 60));

simpan.setText("Simpan");

simpan.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        simpanActionPerformed(evt);
    }
});

jPanel1.add(simpan, new
org.netbeans.lib.awtextra.AbsoluteConstraints(510, 290, 210, 60));

getContentPane().add(jPanel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1250, 660));

pack();

} // </editor-fold>

private void refreshActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

```

```

}

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

    // TODO add your handling code here:

    data_pr FPR = new data_pr();

    FPR.FBY = this;

    FPR.setVisible(true);

    FPR.setResizable(false);

}

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

    // TODO add your handling code here:

    id();

    jDateChooser1.setEnabled(true);

    titik.setEnabled(true);

    tambah.setEnabled(false);

    simpan.setEnabled(true);

    batal.setEnabled(true);

    aktif();

}

private void total_bKeyTyped(java.awt.event.KeyEvent evt) {

    // TODO add your handling code here:

    if (!Character.isDigit(evt.getKeyChar())) {

        evt.consume();

    }

}

private void bayarKeyTyped(java.awt.event.KeyEvent evt) {

```

```

// TODO add your handling code here:

if (!Character.isDigit(evt.getKeyChar())) {

    evt.consume();

}

}

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

    // TODO add your handling code here:

}

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

    // TODO add your handling code here:

    tambah.setEnabled(true);
    simpan.setEnabled(false);
    batal.setEnabled(false);
    titik.setEnabled(false);
    nonaktif();
    clear();
}

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

    // TODO add your handling code here:

    simpan();

}

/**
 * @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_pembayaran.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

    java.util.logging.Logger.getLogger(form_pembayaran.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);

```

```

    } catch (IllegalAccessException ex) {

ava.util.logging.Logger.getLogger(form_pembayaran.class.getName()).log(ja
va.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(form_pembayaran.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 form_pembayaran().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton batal;

private javax.swing.JTextField bayar;

private javax.swing.JButton cari;

private com.toedter.calendar.JDateChooser jDateChooser1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel20;

private javax.swing.JLabel jLabel21;

private javax.swing.JLabel jLabel22;

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 jLabel28;

private javax.swing.JLabel jLabel29;

private javax.swing.JLabel jLabel30;

private javax.swing.JLabel jLabel31;

private javax.swing.JLabel jLabel32;

private javax.swing.JLabel jLabel33;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextField kd_po;

private javax.swing.JTextField no_pr;

private javax.swing.JButton refresh;

private javax.swing.JButton simpan;

private javax.swing.JButton tambah;

private javax.swing.JTable tbl_bayar;

private javax.swing.JTextField tcari;

private javax.swing.JTextField tgl_pesang;

private javax.swing.JButton titik;

private javax.swing.JTextField total_b;

// End of variables declaration

private void loadtabel() {

```



```

String namaKolom[] = {"kd_bayar", "tgl_bayar", "no_pr", "tgl_pesan",
    "ttl_belanja", "jml_bayar"};

rs = con.querySelect(namaKolom, "pembayaran");

tbl_bayar.setModel(new ResultSetTable(rs));
}

private void nonaktif() {

    kd_po.setEnabled(false);

    jDateChooser1.setEnabled(false);

    no_pr.setEnabled(false);

    tgl_pesan.setEnabled(false);

    total_b.setEnabled(false);

    bayar.setEnabled(false);
}

private void aktif() {

    kd_po.setEnabled(false);

    jDateChooser1.setEnabled(true);

    no_pr.setEnabled(false);

    tgl_pesan.setEnabled(false);

    total_b.setEnabled(false);

    bayar.setEnabled(true);
}

private void clear() {

    kd_po.setText("");

    total_b.setText("");

    no_pr.setText("");

```

```

        bayar.setText("");

        tgl_pesan.setText("");
    }

    private void ambildata() {

        String st = (String) no_pr.getText();

        ResultSet rst = con.querySelectAll("pemesanan", "kd_pr='" + st + "'OR
tanggal='" + st + "'");

        try {

            while (rst.next()) {

                this.tgl_pesan.setText(rst.getString("tanggal"));

                this.total_b.setText(rst.getString("ttl_belanja"));

            }
        } catch (SQLException ex) {

            Logger.getLogger(form_pembayaran.class.getName()).log(Level.SEVERE,
            null, ex);

        }

    }

    private void simpan() {

        try {

            if (!kd_po.getText().isEmpty() && !no_pr.getText().isEmpty() &&
            !total_b.getText().isEmpty()

                && !tgl_pesan.getText().isEmpty() &&
            !bayar.getText().isEmpty()) {

                String kolom[] = {"kd_bayar", "tgl_bayar", "no_pr",
                "tgl_pesan", "ttl_belanja", "jml_bayar"};

```

```

        java.util.Date tgl = (java.util.Date) this.jDateChooser1.getDate();

        String isi[] = {kd_po.getText(), new
java.sql.Date(tgl.getTime()).toString(), no_pr.getText(),

        tgl_pesan.getText(), total_b.getText(), bayar.getText()};

        System.out.println(con.queryInsert("pembayaran", kolom, isi));

        JOptionPane.showMessageDialog(this, "Data Berhasil Disimpan,
Kwitansi Sedang di Cetak");

        update_stts();

```

```

        cetak();
        loadtabel();
        nonaktif();
        clear();
        tambah.setEnabled(true);
        batal.setEnabled(false);
        simpan.setEnabled(false);
        titik.setEnabled(false);
    } else {

```



```

        JOptionPane.showMessageDialog(this, "Data isian ada yang
kosong");
    }

    } catch (Exception e) {

        System.out.println("salah");

    }

}

```

```

private void cetak() {

    try {

        String>NamaFile = "src/laporan/kwitansi_po.jasper";

        Class.forName("com.mysql.jdbc.Driver").newInstance();

        Connection koneksi =

DriverManager.getConnection("jdbc:mysql://localhost/pembelian_matrial",

"root", "");

        HashMap hash = new HashMap();

        //Mengambil parameter dari ireport

        hash.put("kode", kd_po.getText());

        File file = new File>NamaFile);

        JasperReport jasperReport = (JasperReport)

JRLoader.loadObject(file.getPath());

        JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport,

hash, koneksi);

        JasperViewer.viewReport(jasperPrint);

    } catch (Exception ex) {

        javax.swing.JOptionPane.showMessageDialog(rootPane,

ex.getMessage());

    }

}

private void update_stts() {

    String kolom[] = {"kd_pr", "status"};

    String isi[] = {no_pr.getText(), "dibayar & diterima"};

```

```

con.queryUpdate("pemesanan", kolom, isi, "kd_pr=" +
String.valueOf(no_pr.getText()) + "");
}
}

```

3.5.2. Blackbox Testing

1. Pengujian Terhadap Halaman Login

Tabel III.8.

Hasil Black Box Testing Halaman Login

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Klik login tanpa menginput <i>username</i> dan <i>password</i>	<i>Username</i> (kosong) <i>Password</i> (kosong)	Sistem akan menolak dan menampilkan pesan “login gagal “	Sesuai harapan	Valid
2	Menginputkan <i>username</i> dan <i>password</i> kemudian klik login	<i>Username</i> (nama user) <i>Password</i> (password user)	Sistem akan menerima akses dan menampilkan pesan “selamat datang “	Sesuai harapan	Valid

2. Pengujian Terhadap Form Data Barang

Tabel III.9.

Hasil Black Box Testing Halaman Data Barang

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Klik tambah kemudian data isian barang tidak diisi, klik tombol simpan	kode barang, nama barang, stok, harga (kosong)	Sistem akan menolak akses dan menampilkan pesan “ data isian ada yang kosong “	Sesuai harapan	Valid
2	Klik tambah kemudian mengisi semua	kode barang, nama	Sistem akan menerima akses dan	Sesuai harapan	Valid

	data buku lalu klik tombol simpan	barang, stok, harga (data terisi)	menampilkan pesan “ data berhasil disimpan “		
3	Masukan pencarian pada tabel barang kemudian klik hapus	Konfirmasi penghapusan data barang	Sistem akan menerima dan menampilkan pesan “ yakin menghapus data “	Sesuai harapan	Valid
4	Masukan pencarian pada table buku, klik tombol edit dan klik tombol simpan	Ubah data barang	Sistem akan menerima dan menampilkan pesan “ data berhasil diedit “	Sesuai harapan	Valid

3. Pengujian terhadap *form Purchase Request*

Tabel III.10.

Hasil Black Box Testing Halaman *Purchase Request*

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Klik tambah kemudian data isian pemesanan tidak diisi, klik simpan	nomor, kode_pr, tanggal, kd_supplier, kd_brg, jml_beli, hrg_satuan, total_beli, ttl_belanja, status	Sistem akan menolak akses dan menampilkan pesan “ data isian ada yang kosong “	Sesuai harapan	Valid
2	Klik tambah kemudian mengisi semua data pemesanan, klik tombol simpan	nomor, kode_pr, tanggal, kd_supplier, kd_brg, jml_beli, hrg_satuan, total_beli, ttl_belanja, status	Sistem akan menerima akses dan menampilkan pesan “ data berhasil disimpan “	Sesuai harapan	Valid

4. Pengujian Terhadap *Form Purchase Order***Tabel III.11.****Hasil Black Box Testing Halaman Purchase Order**

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Klik tambah kemudian data isian pembayaran tidak diisi, klik simpan	kd_bayar, tgl_bayar, no_pr, tgl_pesan, ttl_belanja, jml_bayar	Sistem akan menolak akses dan menampilkan pesan “ data isian ada yang kosong “	Sesuai harapan	Valid
2	Klik tambah kemudian mengisi semua data pembayaran, klik tombol simpan	kd_bayar, tgl_bayar, no_pr, tgl_pesan, ttl_belanja, jml_bayar	Sistem akan menerima akses dan menampilkan pesan “ data berhasil disimpan “	Sesuai harapan	Valid

5. Pengujian Terhadap *Form Jurnal Umum***Tabel III.12.****Hasil Black Box Testing Halaman Jurnal Umum**

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Klik tambah kemudian data isian jurnal umum tidak diisi, klik simpan	Nomor, no_jurnal, tanggal, no_trans, keterangan, kd_akun, nm_akun, debet, kredit	Sistem akan menolak akses dan menampilkan pesan “ data isian ada yang kosong “	Sesuai harapan	Valid
2	Klik tambah kemudian mengisi semua data jurnal umum, klik tombol simpan	Nomor, no_jurnal, tanggal, no_trans, keterangan, kd_akun, nm_akun, debet, kredit	Sistem akan menerima akses dan menampilkan pesan “ data berhasil disimpan “	Sesuai harapan	Valid

3.5.3. Spesifikasi *Hardware* dan *Software*

1. Spesifikasi *Hardware*

Spesifikasi *Hardware* yang digunakan penulis adalah sebagai berikut :

- a. Laptop : Acer One
- b. *Processor* : 1.4 GHz Core I3
- c. *RAM* : 2 GB
- d. *Harddisk* : 500 gb
- e. Monitor : SVGA 14"
- f. *Keyboard* : 108 ke
- g. *Mouse* : Standard Mouse

2. Spesifikasi *Software*

Spesifikasi *Software* yang digunakan penulis adalah sebagai berikut :

- a. Sistem Operasi : Windows 10
- b. Bahasa Pemograman : java
- c. DBMS : *PhpmyAdmin*
- d. Antivirus : SMADAV
- e. Aplikasi : *Netbeans* IDE 8.0