

## **BAB III**

### **PEMBAHASAN**

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

PT. Reykatama Kimasu Indonesia merupakan perusahaan trading yang beralamata di Jl. Pandu No. 209 Blok K Perumnas Telukjambe, Kabupaten Karawang.

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

PT. Reykatam Kimasu Indonesia merupakan perusahaan yang bergerak di bidang trading, didirikan oleh Budhi Ary Sanjaya pada tahun 2017. Awalnya perusahaan ini bergabung dengan PT. Duta Swarna Dwipa kemudian dipecah pada awal tahun 2019. Sejak dipecah PT. Reykatama Kimasu Indonesia semakin berkembang pesat.

PT. Reykatama Kimasu Indonesia bekerjasama dengan beberapa Supplier untuk memenuhi keinginan pelanggan. Diantaranya adalah PT. Duta Swarna Dwipa, Kita Utama, P. Siti Nugraha Kumala (NSK) dan lainnya. Adapun visi dan misi PT. Reykatama Kimasu Indonesia :

1. Visi
  - a. Menjadi supplier pilihan utama pelanggan
  - b. Meningkatkan pelayanan kualitas produk terbaik bagi pelanggan
2. Misi
  - a. Menciptakan lingkungan yang dapat memaksimalkan potensi karyawan dan bisnis kami.
  - b. Meningkatkan kinerja baik pribadi maupun dalam bisnis kami.

### 3.1.2. Struktur Organisasi



Sumber : PT. Reykatama Kimasu Indonesia

**Gambar III.1**

#### **Struktur Organisasi PT. Reykatama Kimasu Indonesia**

Adapun fungsi dari masing-masing bagian adalah sebagai berikut:

1. **Pemilik**  
Pemilik bertanggung jawab penuh terhadap aktifitas yang ada di perusahaan. Pemilik juga bertugas menerbitkan kebijakan-kebijakan perusahaan.
2. **Penanggung Jawab**  
Penanggung jawab bertugas membantu pemilik dalam menjalankan operasional perusahaan.
3. **Accounting**  
Accounting bertugas mengatur keuangan perusahaan serta membuat laporan keuangan.
4. **Marketing**  
Melaksanakan penjualan dan pembelian melalui telepon terhadap target konsumen ( perusahaan-perusahaan perdagangan dan industri, kantor-kantor pemerintah, asosiasi perkumpulan keagamaan, olahraga, sosial, konsultan) secara sistematis, serta melengkapi laporan kegiatan untuk setiap hubungan yang dilakukan.

5. Admin

Administrasi bertugas membuat PO (*Purchase Order*), DO (*Delivery Order*).

6. Bagian Gudang

Mengecek persediaan barang perminggu, memastikan barang yang akan dikirim tidak rusak.

7. Supir

Bertanggung jawab atas pengiriman barang kepada cumtomer, mengecek barang sebelum dikirim.

### 3.2. Tinjauan Kasus

Tinjauan kasus ini terdiri dari proses berjalan yang ada di PT. Reykatama Kimasu Indonesia. Adapun ptoses berjalannya meliputi proses:

#### 3.2.1. Proses Bisnis Sistem Berjalan

##### 1) Proses Penerimaan Barang

Bagian gudang membuat surat permintaan barang kemudian memberikan surat permintaan ke admin. Admin menerima surat permintaan barang kemudian membuat PO (*Purchase Order*). PO (*Purchase Order*) tersebut dikirimkan ke Supplier. Supplier menerima PO (*Purchase Order*) kemudian mengirimkan barang sesuai PO (*Purchase Order*). Bagian gudang menerima barang dan surat jalan dari Supplier kemudian mengecek barang, jika barang sesuai maka bagian gudang menandatangani surat jalan.

##### 2) Proses Pengeluaran Barang

Bagian marketing memberikan PO (*Purchase Order*) kepada Admin, kemudian admin menerima PO (*Purchase Order*) dan membuat surat jalan. Surat jalan tersebut itu diberikan kepada bagian gudang.

Bagian gudang menerima surat jalan kemudian mengecek ketersediaan barang dan menyiapkan barang sesuai PO (*Purchase Order*). Kemudian melakukan pengiriman barang.

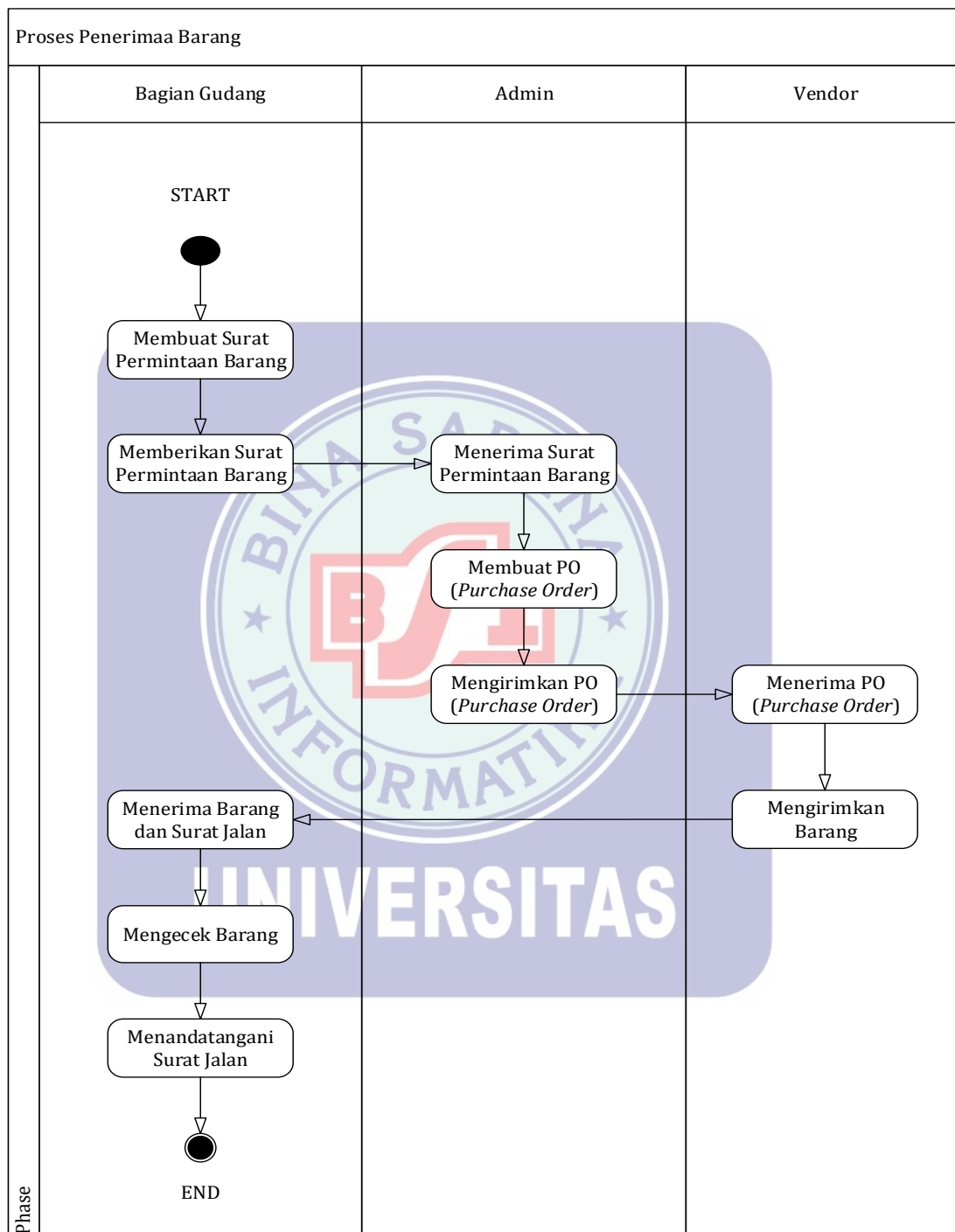
### **3) Proses Pembuatan Laporan**

Bagian gudang membuat laporan persediaan barang sesuai dengan data barang masuk dan data barang keluar kemudian di berikan kepada admin. Setelah itu admin mengecek laporan persediaan sesuai atau tidak yang ada di komputer dan penghitungan secara fisik. Jika sesuai laporan persediaan diberikan kepada bagian accounting.



### 3.2.2. Activity Diagram Sistem Berjalan

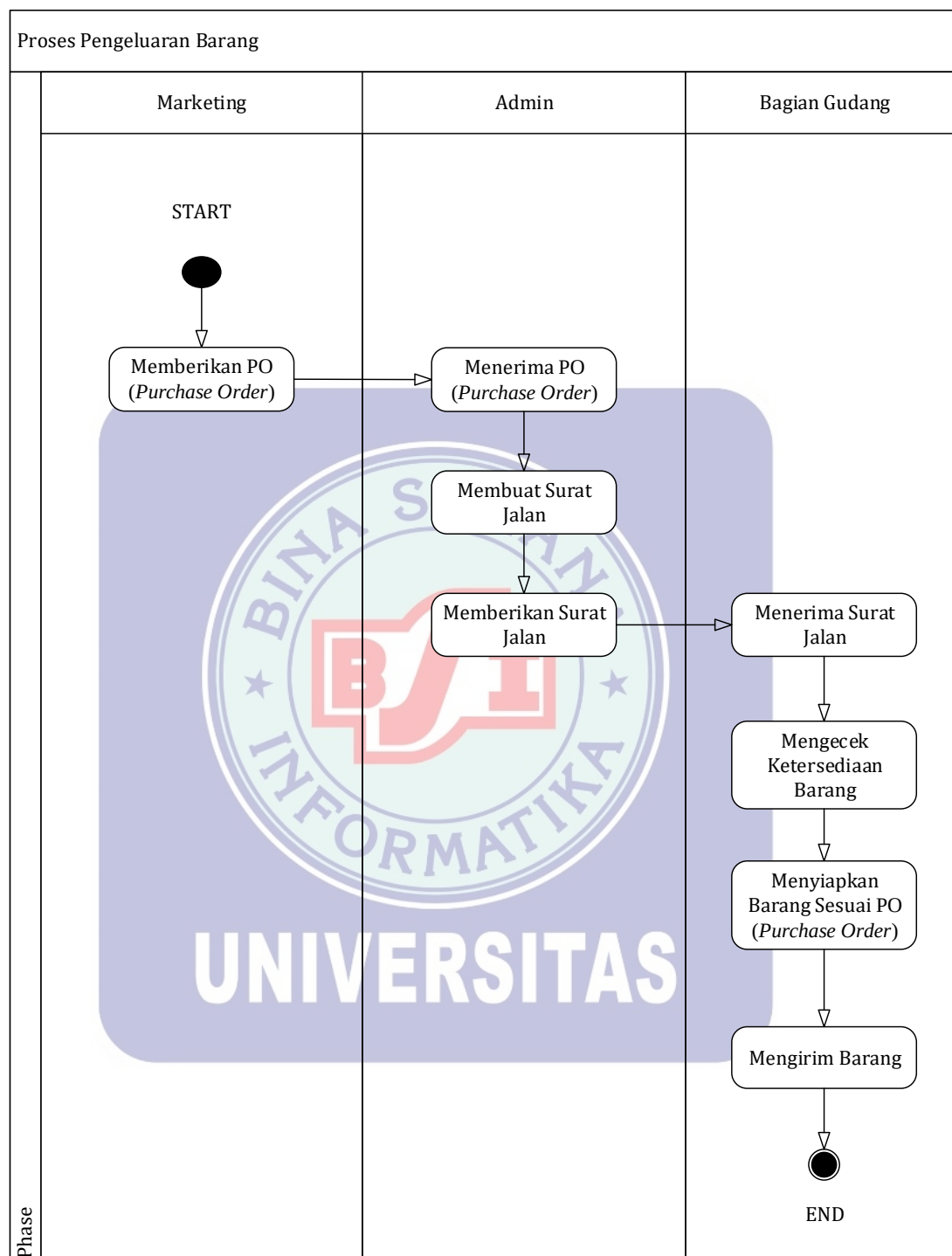
#### 1. Proses Penerimaan Barang



**Gambar III.2.**

**Proses Penerimaan Barang**

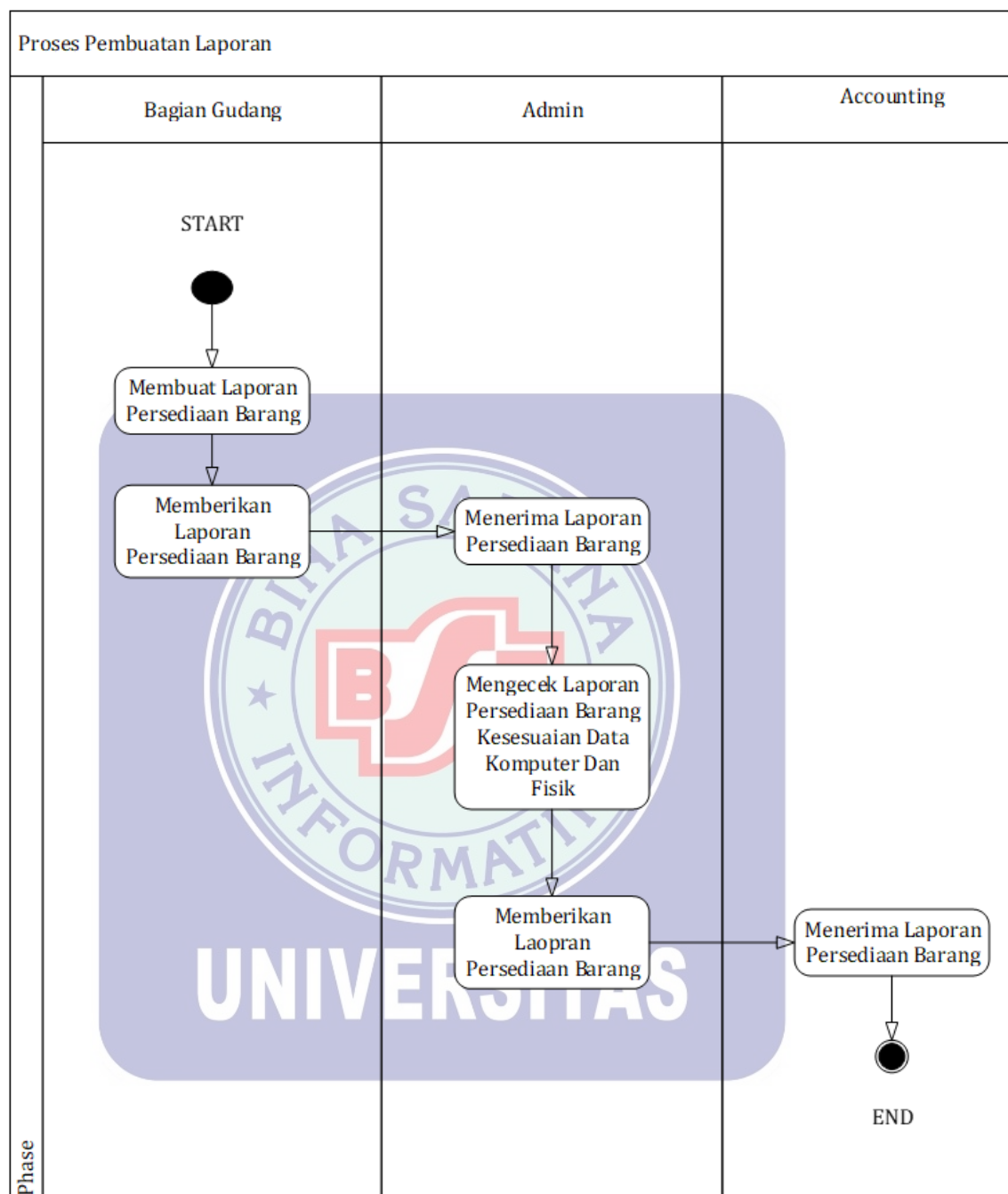
## 2. Proses Pengeluaran Barang



**Gambar III.3.**

**Proses Pengeluaran Barang**

### 3. Proses Pembuatan Laporan



Gambar III.4.

Proses Pembuatan Laporan

### 3.2.3. Dokumen Masukan

Adapun dokumen masukan dari proses sistem berjalan yang ada di PT.

Reykatama Kimasu Indonesia adalah sebagai berikut:

1. Nama Dokumen : *Purchase Order*  
 Fungsi : Untuk Pemesanan Barang  
 Sumber : *Customer*  
 Tujuan : Marketing  
 Media : *Email*  
 Jumlah : 1 Lampiran  
 Frekuensi : Setiap pemesanan Barang  
 Format : Lampiran A1
2. Nama Dokumen : Surat Jalan  
 Fungsi : Sebagai Bukti Pengiriman  
 Sumber : Supplier  
 Tujuan : Bagian Gudang  
 Media : Kertas  
 Jumlah : 1 Rangkap  
 Frekuensi : Setiap Transaksi penerimaan Barang  
 Format : Lampiran A2



### 3.2.4. Dokumen Keluaran

Berikut ini dokumen keluaran yang ada proses sistem berjalan yang ada di PT. Reykatama Kimasu Indonesia:

1. Nama Dokumen : Surat Permintaan Barang  
 Fungsi : Sebagai Permintaan Barang  
 Sumber : Bagian Gudang  
 Tujuan : Admin  
 Media : Kertas  
 Jumlah : 1 Lembar  
 Frekuensi : Setiap Stok Barang Habis  
 Format : Lampiran B1
2. Nama Dokumen : *Purchase Order*  
 Fungsi : Untuk Pemesanan Barang  
 Sumber : Admin  
 Tujuan : Supplier  
 Media : *Email*  
 Jumlah : 1 Lampiran  
 Frekuensi : Setiap Pemesanan Barang  
 Format : Lampiran B2
3. Nama Dokumen : Surat Jalan  
 Fungsi : Sebagai Bukti Pengiriman  
 Sumber : Bagian Gudang  
 Tujuan : *Customer*  
 Media : Kertas Rangkap

|    |              |                                      |
|----|--------------|--------------------------------------|
|    | Jumlah       | : 3 Rangkap                          |
|    | Frekuensi    | : Setiap Transaksi Pengiriman Barang |
|    | Format       | : Lampiran B3                        |
| 4. | Nama Dokumen | : Laporan Persediaan Barang          |
|    | Fungsi       | : Sebagai Bahan Laporan              |
|    | Sumber       | : Bagian Gudang                      |
|    | Tujuan       | : Admin                              |
|    | Media        | : Kertas                             |
|    | Jumlah       | : Sesuai Laporan                     |
|    | Frekuensi    | : Setiap Akhir Bulan                 |
|    | Format       | : Lampiran B4                        |

### 3.2.5. Permasalahan Pokok

Adapun permasalahan pokok persediaan barang dagang pada PT. Reykatama Kimasu

Indonesia antara lain:

1. Sistem pencatatan persediaan barang pada PT. Reykatama Kimasu Indonesia masih kurang efektif yaitu menggunakan microsoft excel sehingga sering menimbulkan keterlambatan dalam pengolahan data barang.
2. Pada saat perhitungan stok barang sering terjadi ketidaksamaan antara data stok barang di komputer dan aktual.
3. Penyimpanan data masih menggunakan arsip sehingga masih ada data yang hilang atau rusak, dalam pencarian data pun masih membutuhkan waktu yang lama dikarenakan banyaknya arsip.

### 3.2.6. Pemecahan Masalah

Adapun Alternatif pemecahan masalah dari permasalahan yang sering timbul tersebut adalah sebagai berikut:

1. Dengan adanya sistem baru diharapkan semua proses pencatatan persediaan barang dagang bisa dilakukan dengan cepat, tepat dan akurat karena proses perhitungan dan pencatatan sudah dilakukan komputer.
2. Perlu diadakan pengecekan stok perminggu guna mengetahui barang masuk dan keluar sesuai dengan aktual dan data stok.
3. Dengan adanya sistem baru diharapkan adanya *database* terdistribusi sehingga dapat di akses dengan cepat.

### 3.3. Analisa Kebutuhan Software

#### 3.3.1. Analisa Kebutuhan

Berikut ini merupakan analisa kebutuhan dari sistem persediaan yang akan dibuat. Analisa kebutuhan ini terdiri dari dua aktor yaitu kebutuhan *user* sebagai pengelola transaksi dan kebutuhan untuk pemilik.

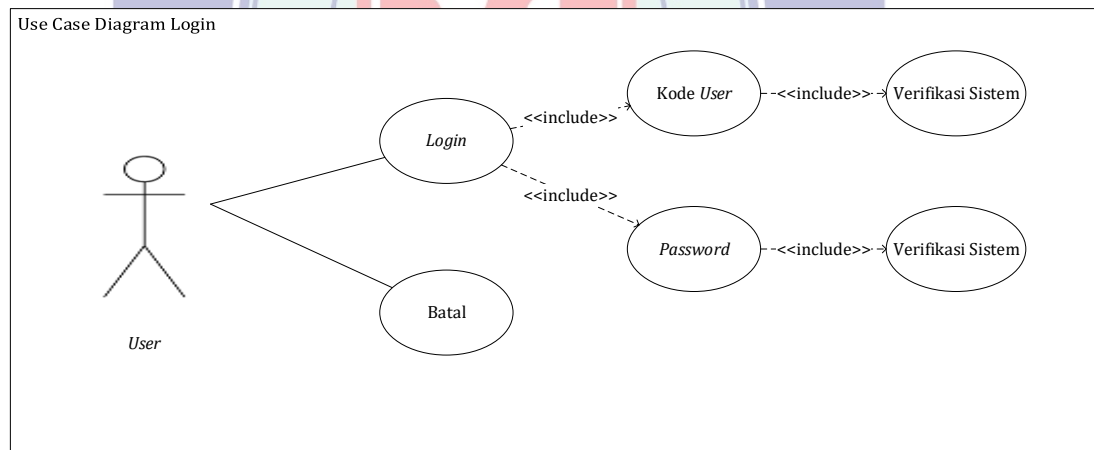
#### A. Kebutuhan User

1. *User Melakukan Login*
2. *User Mengakses Menu Master*
  - 2.1. *User Mengelola Data User*
  - 2.2. *User Mengelola Data Barang*
  - 2.3. *User Mengelola Data Supplier*
  - 2.4. *User Mengelola Data Akun*

3. *User Mengakses Menu Transaksi*
  - 3.1. *User Mengelola Transaksi Penerimaan Barang*
  - 3.2. *User Mengelola Transaksi Pengeluaran Barang*
  - 3.3. *User Mengelola Transaksi Pengeluaran Jurnal*
4. *User Mengakses Menu Laporan*
  - 4.1. *User Mengakses Laporan Persediaan Barang*
  - 4.2. *User Mengakses Laporan Penerimaan Barang*
  - 4.3. *User Mengakses Laporan Pengeluaran Barang*
5. *User Melakukan Logout*

### 3.3.2. *Use Case Diagram*

#### 1. *Use Case Diagram Melakukan Login*



**Gambar III.5**

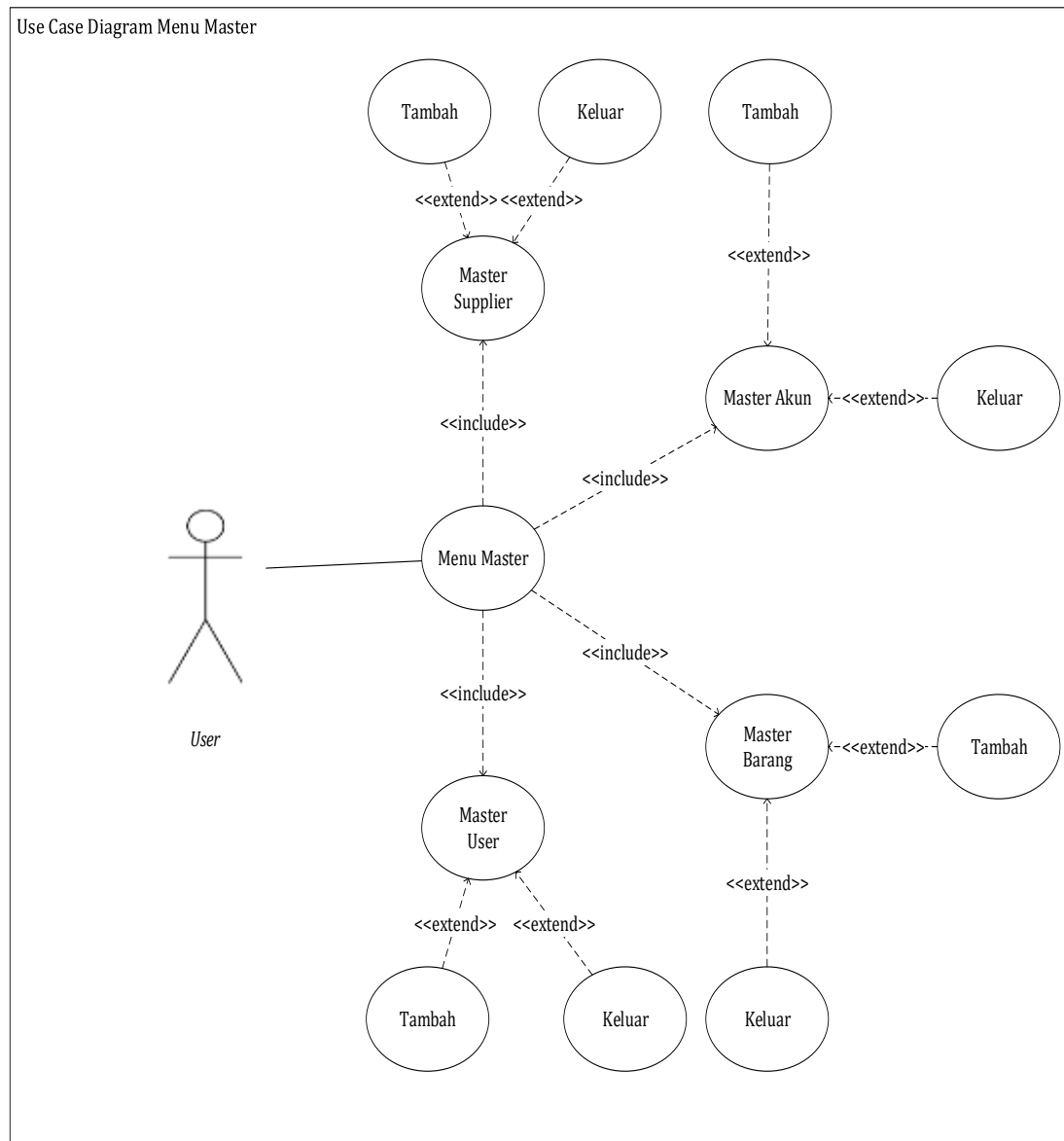
***Use Case Diagram Melakukan Login***

Tabel III.1.

**Deskripsi Use Case Diagram Melakukan Login**

| <b>Use Case Narrative Login</b>      |                                                                                                                                                                                                              |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tujuan                               | User dapat <i>login</i> ke aplikasi persediaan barang                                                                                                                                                        |
| Deskripsi                            | Sistem ini memungkinkan aktor untuk <i>login</i> ke dalam aplikasi persediaan barang, mengakses menu utama, menu master, menu transaksi dan menu laporan serta <i>logout</i> dari aplikasi persediaan barang |
| <b>Skenario Utama</b>                |                                                                                                                                                                                                              |
| Aktor                                | User                                                                                                                                                                                                         |
| Kondisi Awal                         | Aktor melakukan <i>login</i> dan mengakses aplikasi sistem persediaan barang                                                                                                                                 |
| <b>Aksi Aktor</b>                    | <b>Reaksi Sistem</b>                                                                                                                                                                                         |
| 1. Aktor memilih tombol <i>Login</i> | Sistem menampilkan pesan “Selamat Bekerja”                                                                                                                                                                   |
| 2. Aktor memilih tombol Batal        | Sistem membatalkan proses <i>Login</i>                                                                                                                                                                       |
| Kondisi Akhir                        | Jika perintah sesuai maka sistem akan menampilkan sesuai keinginan aktor                                                                                                                                     |

## 2. Use Case Diagram Menu Master



**Gambar III.6**

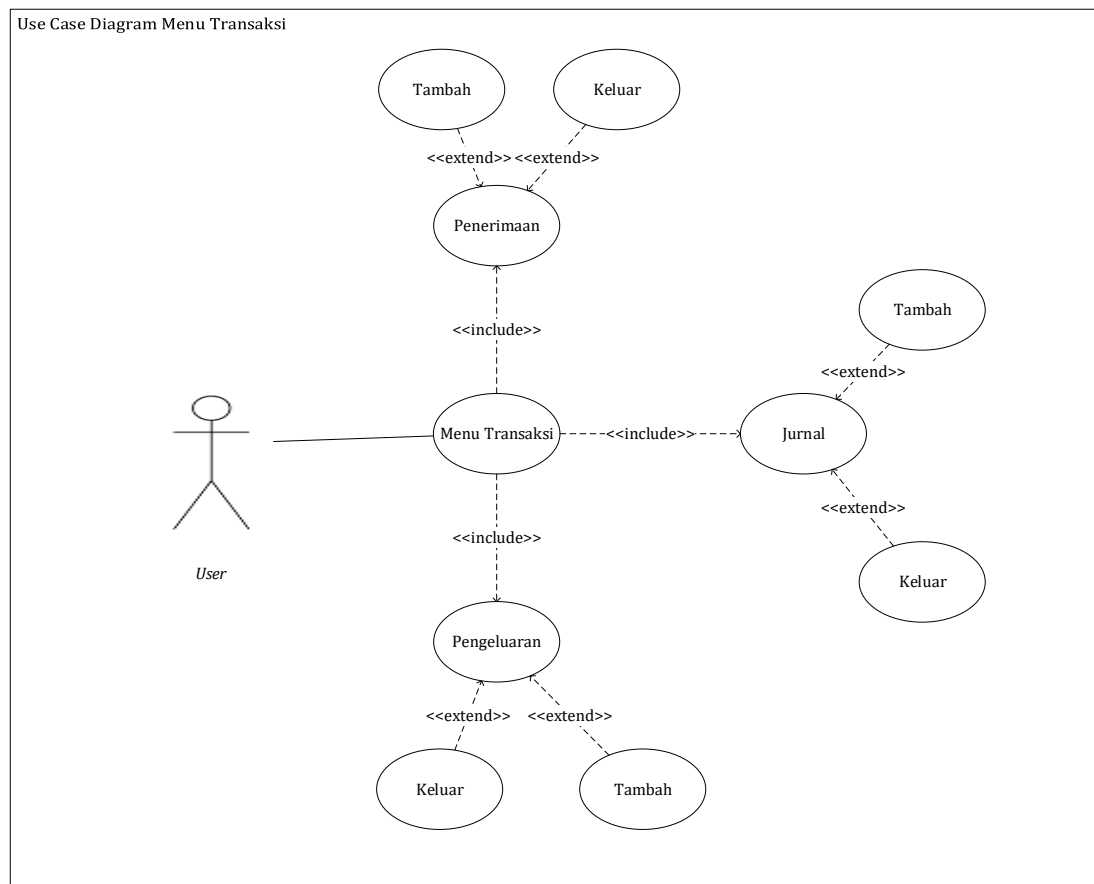
*Use Case Diagram Menu Master*

Tabel III.2.

**Deskripsi Use Case Diagram Menu Master**

| <b>Use Case Narrative Menu Master</b> |                                                                                                                                                                      |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tujuan                                | User dapat melakukan pengolahan data yang ada didalam aplikasi persediaan barang                                                                                     |
| Deskripsi                             | Sistem ini memungkinkan aktor untuk mengolah sistem persediaan barang pada menu master yang terdiri dari master user, master barang, master Supplier dan master akun |
| <b>Skenario Utama</b>                 |                                                                                                                                                                      |
| Aktor                                 | User                                                                                                                                                                 |
| Kondisi Awal                          | Aktor melakukan <i>login</i> dan mengakses menu master                                                                                                               |
| <b>Aksi Aktor</b>                     | <b>Reaksi Sistem</b>                                                                                                                                                 |
| 1) Aktor memilih Master User          | Sistem menampilkan opsi tambah dan keluar                                                                                                                            |
| 2) Aktor memilih Master Barang        | Sistem menampilkan opsi tambah dan keluar                                                                                                                            |
| 3) Aktor memilih Master Supplier      | Sistem menampilkan opsi tambah dan keluar                                                                                                                            |
| 4) Aktor memilih Master Akun          | Sistem menampilkan opsi tambah dan keluar                                                                                                                            |
| Kondisi Akhir                         | Jika perintah sesuai maka sistem akan menampilkan sesuai keinginan aktor                                                                                             |

### 3. Use Case Diagram Menu Transaksi



Gambar III.7

Use Case Diagram Menu Transaksi

UNIVERSITAS

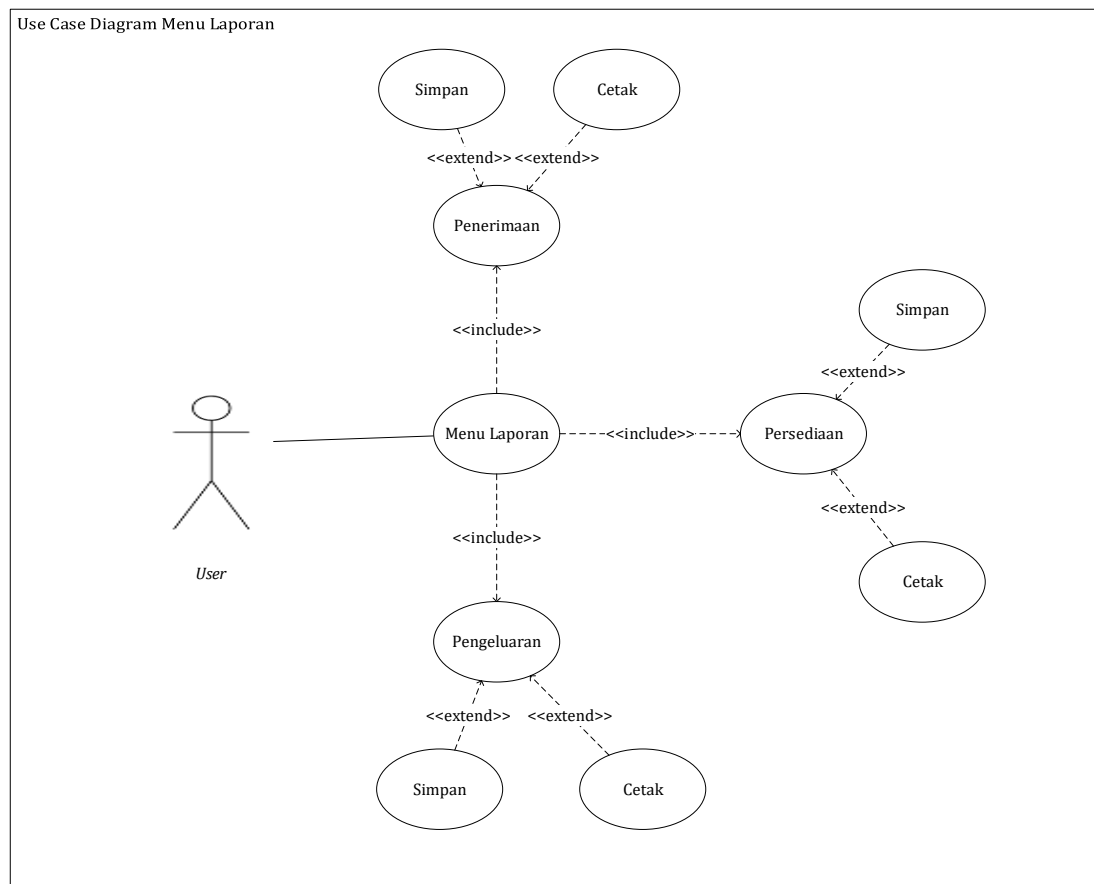


Tabel III.3.

**Deskripsi Use Case Diagram Menu Transaksi**

| <b>Use Case Narrative Transaksi</b>    |                                                                                                                                                                         |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tujuan                                 | User dapat melakukan pengolahan data yang ada didalam aplikasi persediaan barang                                                                                        |
| Deskripsi                              | Sistem ini memungkinkan aktor untuk mengolah sistem persediaan barang pada menu transaksi terdiri dari transaksi penerimaan, transaksi pengeluaran dan transaksi jurnal |
| <b>Skenario Utama</b>                  |                                                                                                                                                                         |
| Aktor                                  | User                                                                                                                                                                    |
| Kondisi Awal                           | Aktor melakukan <i>login</i> dan mengakses menu transaksi                                                                                                               |
| <b>Aksi Aktor</b>                      | <b>Reaksi Sistem</b>                                                                                                                                                    |
| 1) Aktor memilih Transaksi Penerimaan  | Sistem menampilkan opsi tambah dan keluar                                                                                                                               |
| 2) Aktor memilih Transaksi Pengeluaran | Sistem menampilkan opsi tambah dan keluar                                                                                                                               |
| 3) Aktor memilih Transaksi Jurnal      | Sistem menampilkan opsi tambah dan keluar                                                                                                                               |
| Kondisi Akhir                          | Jika perintah sesuai maka sistem akan menampilkan sesuai keinginan aktor                                                                                                |

#### 4. Use Case Diagram Menu Laporan



**Gambar III.8**

***Use Case Diagram Menu Laporan***

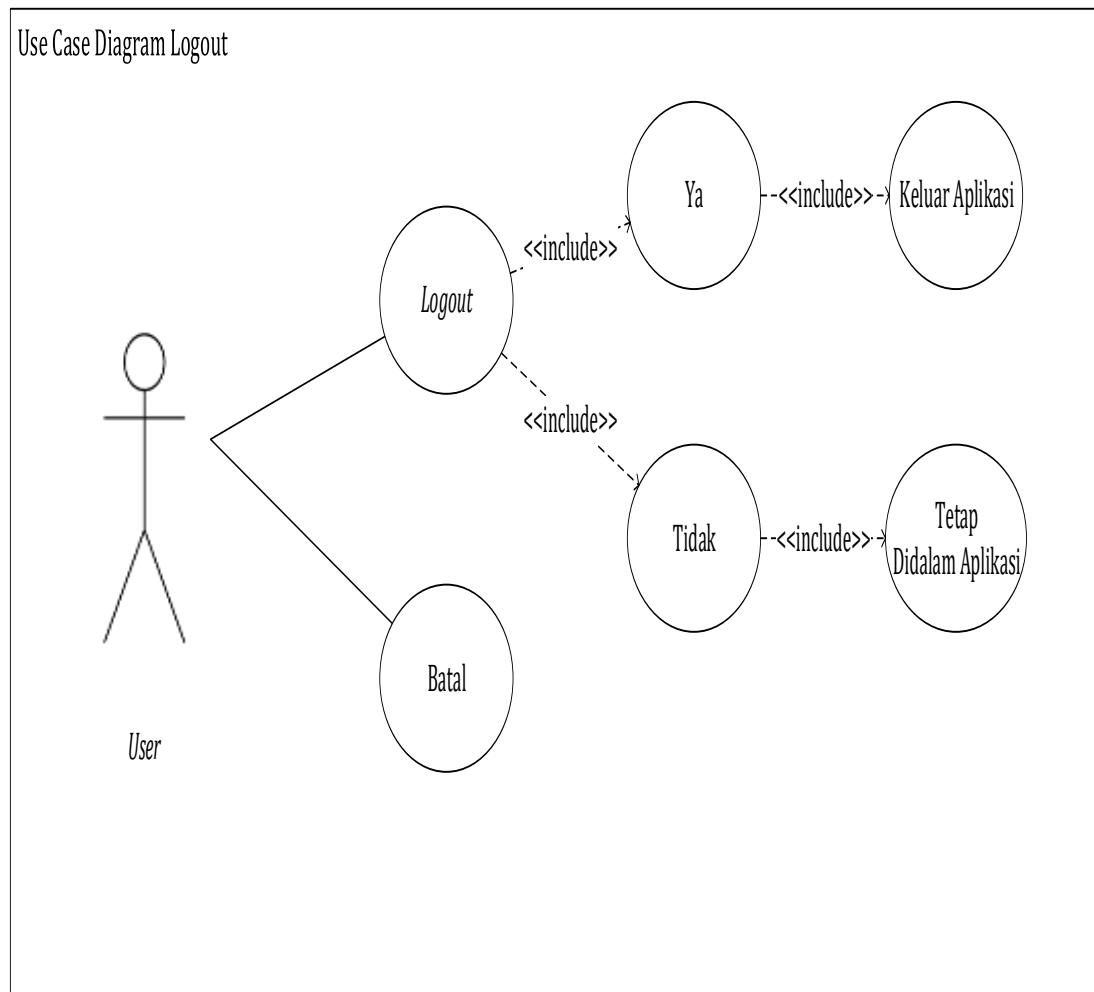
**UNIVERSITAS**

Tabel III.4.

**Deskripsi Use Case Diagram Menu Laporan**

| <b>Use Case Narrative Menu Laporan</b>     |                                                                                                                                                                           |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tujuan                                     | User dapat melakukan pengolahan data yang ada didalam aplikasi persediaan barang                                                                                          |
| Deskripsi                                  | Sistem ini memungkinkan aktor untuk mengolah sistem persediaan barang pada menu laporan yang terdiri dari laporan persediaan, laporan penerimaan dan laporan pengeluaran. |
| <b>Skenario Utama</b>                      |                                                                                                                                                                           |
| Aktor                                      | User                                                                                                                                                                      |
| Kondisi Awal                               | Aktor melakukan <i>login</i> dan mengakses menu laporan                                                                                                                   |
| <b>Aksi Aktor</b>                          | <b>Reaksi Sistem</b>                                                                                                                                                      |
| 1) Aktor memilih Laporan Persediaan Barang | Sistem menampilkan opsi simpan dan cetak                                                                                                                                  |
| 2) Aktor memilih Laporan Penerimaan        | Sistem menampilkan opsi simpan dan cetak                                                                                                                                  |
| 3) Aktor memilih Laporan Pengeluaran       | Sistem menampilkan opsi simpan dan cetak                                                                                                                                  |
| Kondisi Akhir                              | Jika perintah sesuai maka sistem akan menampilkan sesuai keinginan aktor                                                                                                  |

## 5. Use Case Diagram Melakukan Logout



**Gambar III.9.**  
**Use Case Diagram Melakukan Logout**

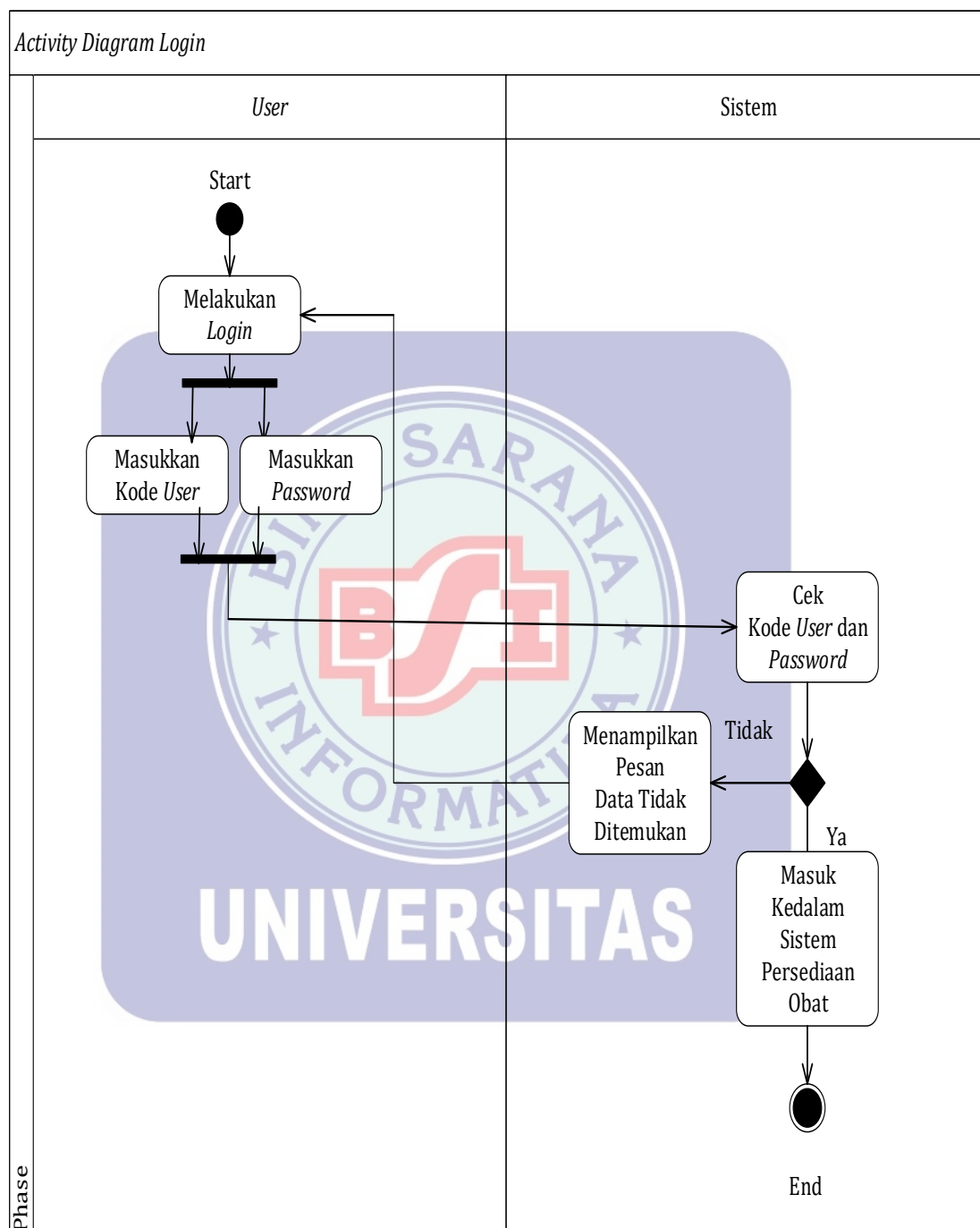
Tabel III.5.

**Deskripsi Use Case Diagram Melakukan Logout**

| <b>Use Case Narrative Logout</b> |                                                                                    |
|----------------------------------|------------------------------------------------------------------------------------|
| Tujuan                           | User dapat <i>logout</i> dari aplikasi persediaan barang                           |
| Deskripsi                        | Sistem ini memungkinkan aktor untuk <i>logout</i> dari aplikasi persediaan barang. |
| <b>Skenario Utama</b>            |                                                                                    |
| Aktor                            | User                                                                               |
| Kondisi Awal                     | Aktor melakukan <i>logout</i> dari aplikasi sistem persediaan barang               |
| <b>Aksi Aktor</b>                | <b>Reaksi Sistem</b>                                                               |
| 1. Aktor memilih Ya              | Sistem keluar dari aplikasi persediaan barang                                      |
| 2. Aktor memilih Batal           | Sistem membatalkan proses <i>Logout</i> dan tetap di aplikasi persediaan barang    |
| Kondisi Akhir                    | Jika perintah sesuai maka sistem akan menampilkan sesuai keinginan aktor           |

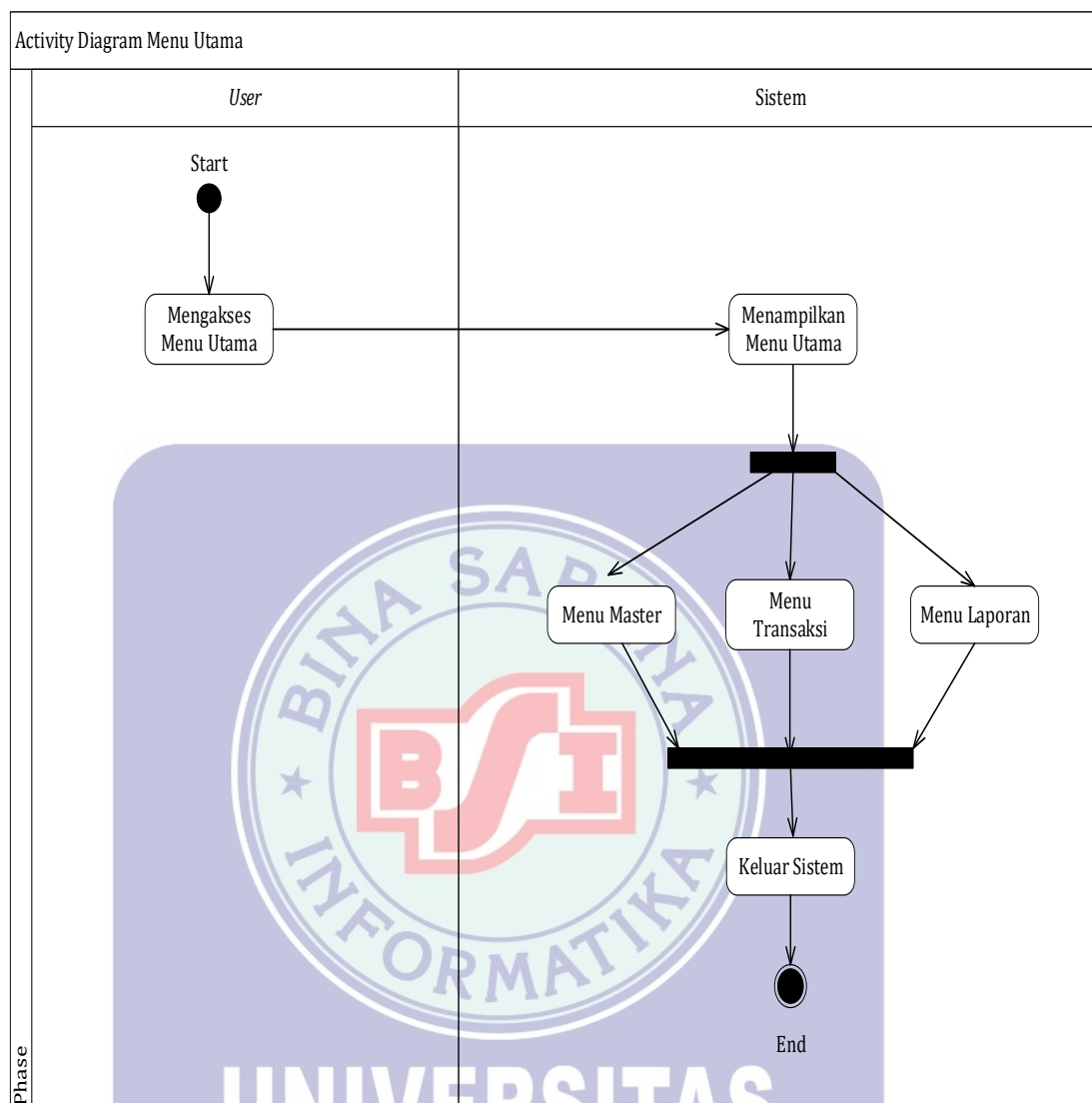
### 3.3.3. Activity Diagram

#### 1. Activity Diagram Menu Login



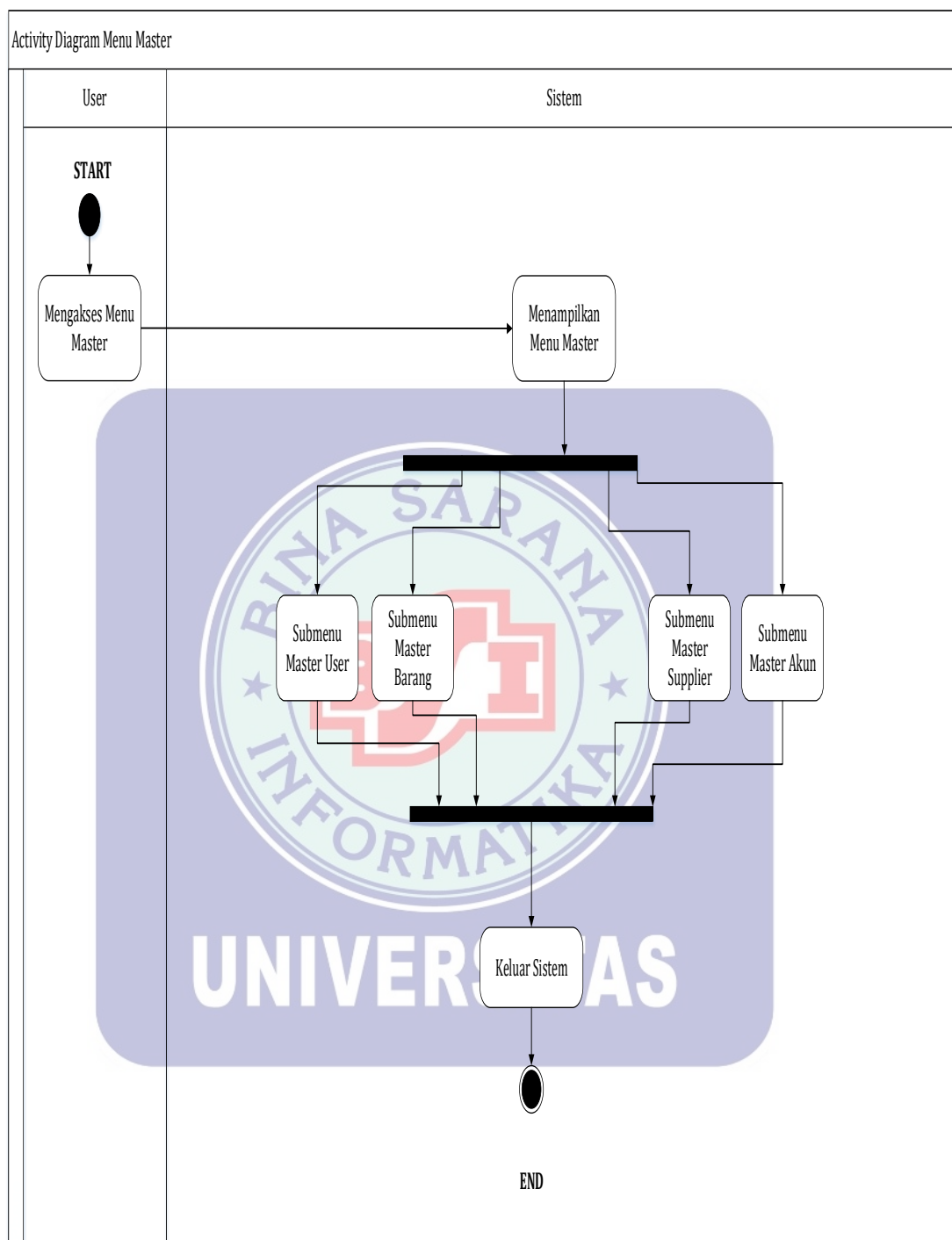
**Gambar III.10.**  
**Activity Diagram Menu Login**

## 2. Activity Diagram Menu Utama



**Gambar III.11.**  
Activity Diagram Menu Utama

### 3. Activity Diagram Menu Master

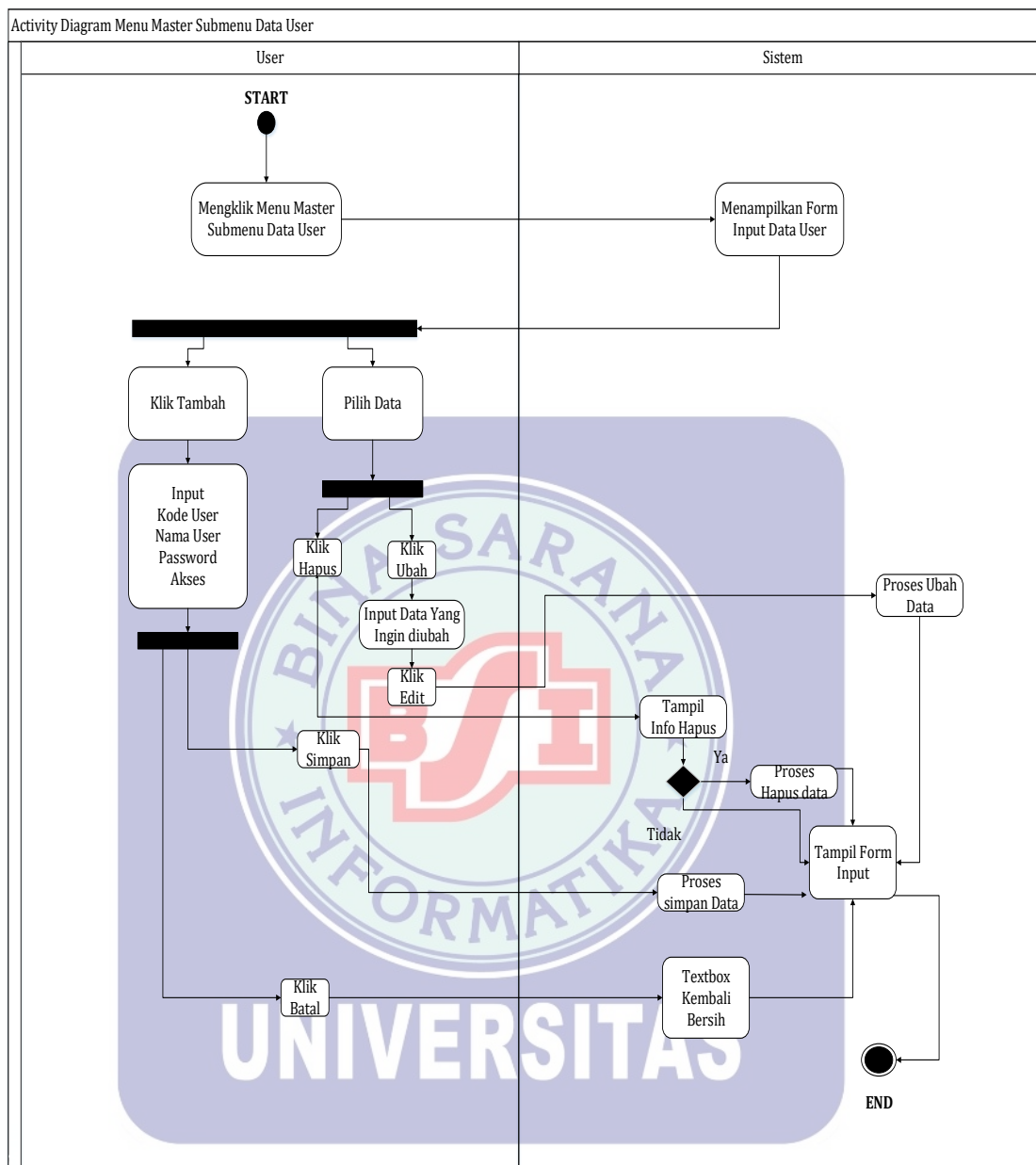


**Gambar III.12.**

**Activity Diagram Menu Master**

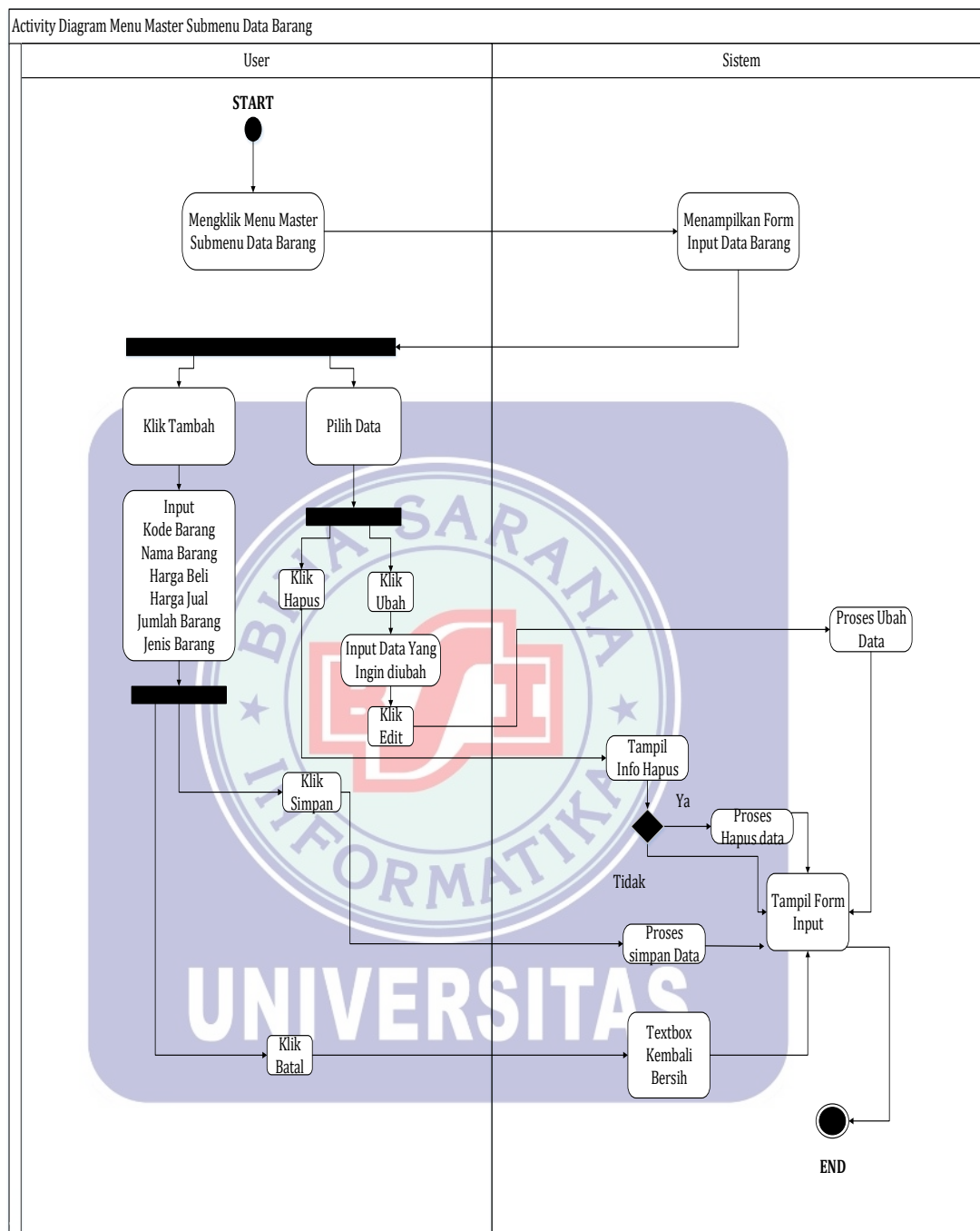


#### 4. Activity Diagram Menu Master Submenu Data User



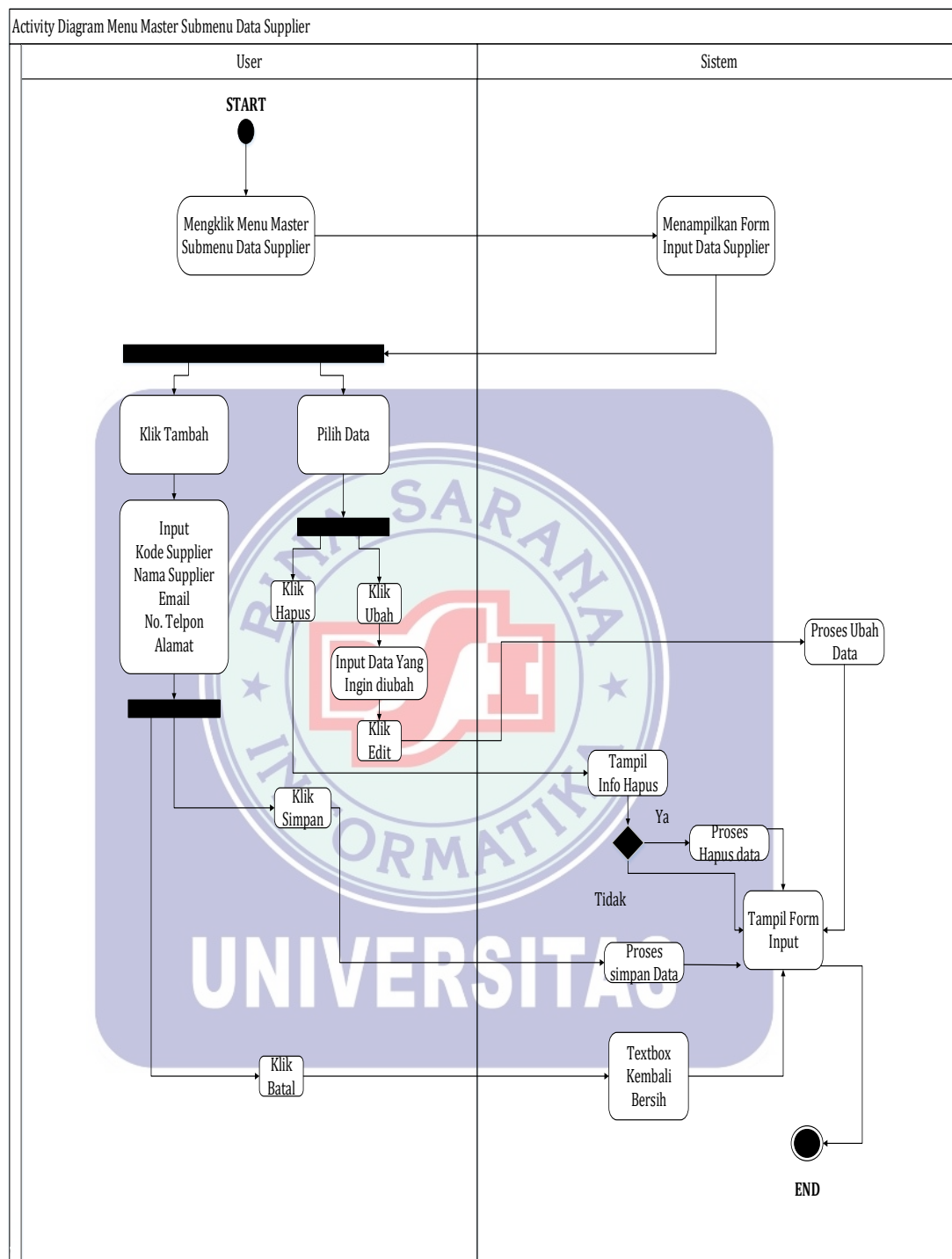
**Gambar III.13.**  
**Activity Diagram Menu Master Submenu Data User**

### 5. Activity Diagram Menu Master Submenu Data Barang



**Gambar III.14.**  
**Activity Diagram Menu Master Submenu Data Barang**

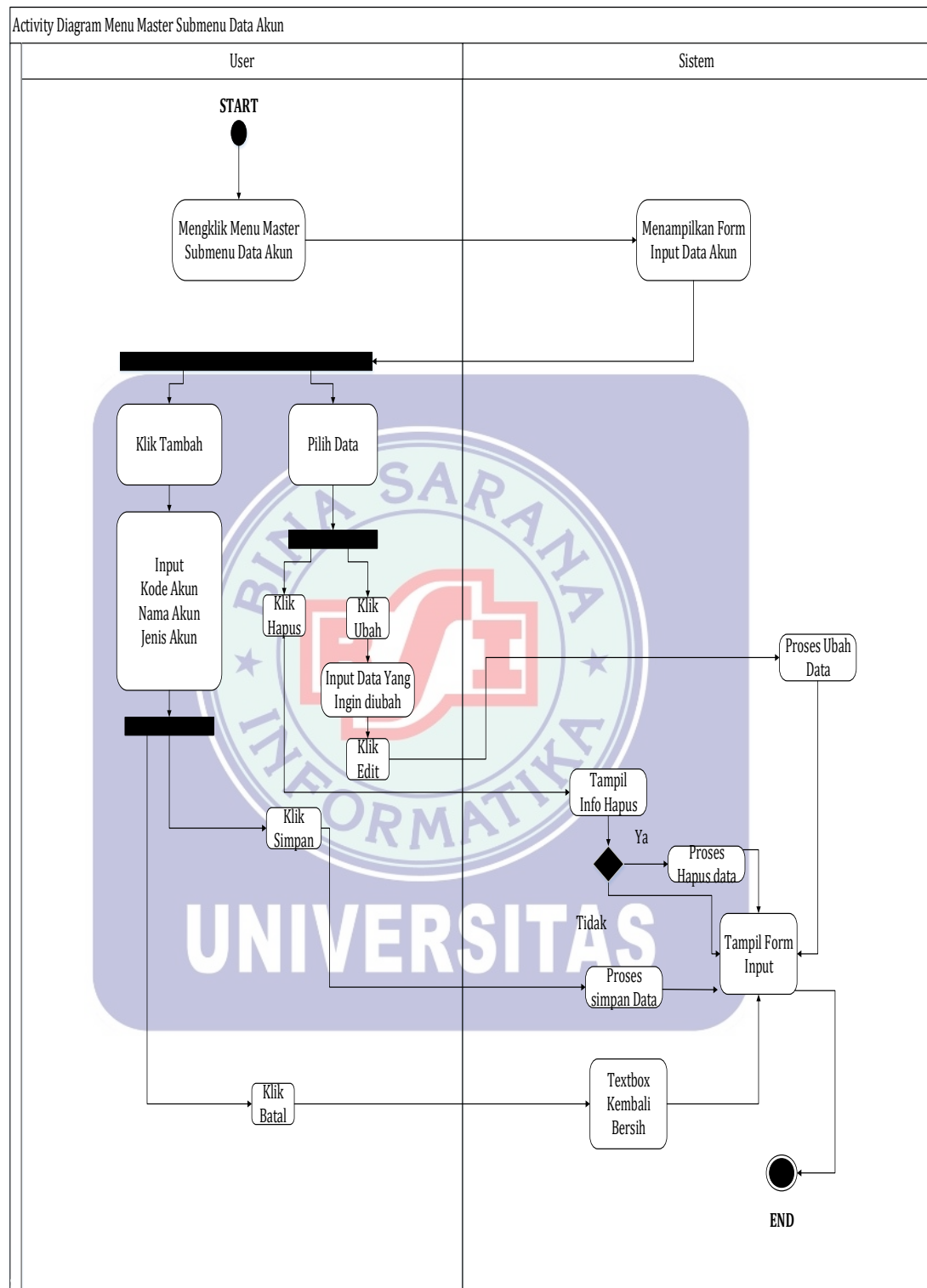
## 6. Activity Diagram Menu Master Submenu Data Supplier



Gambar III.15.

*Activity Diagram Menu Master Submenu Data Supplier*

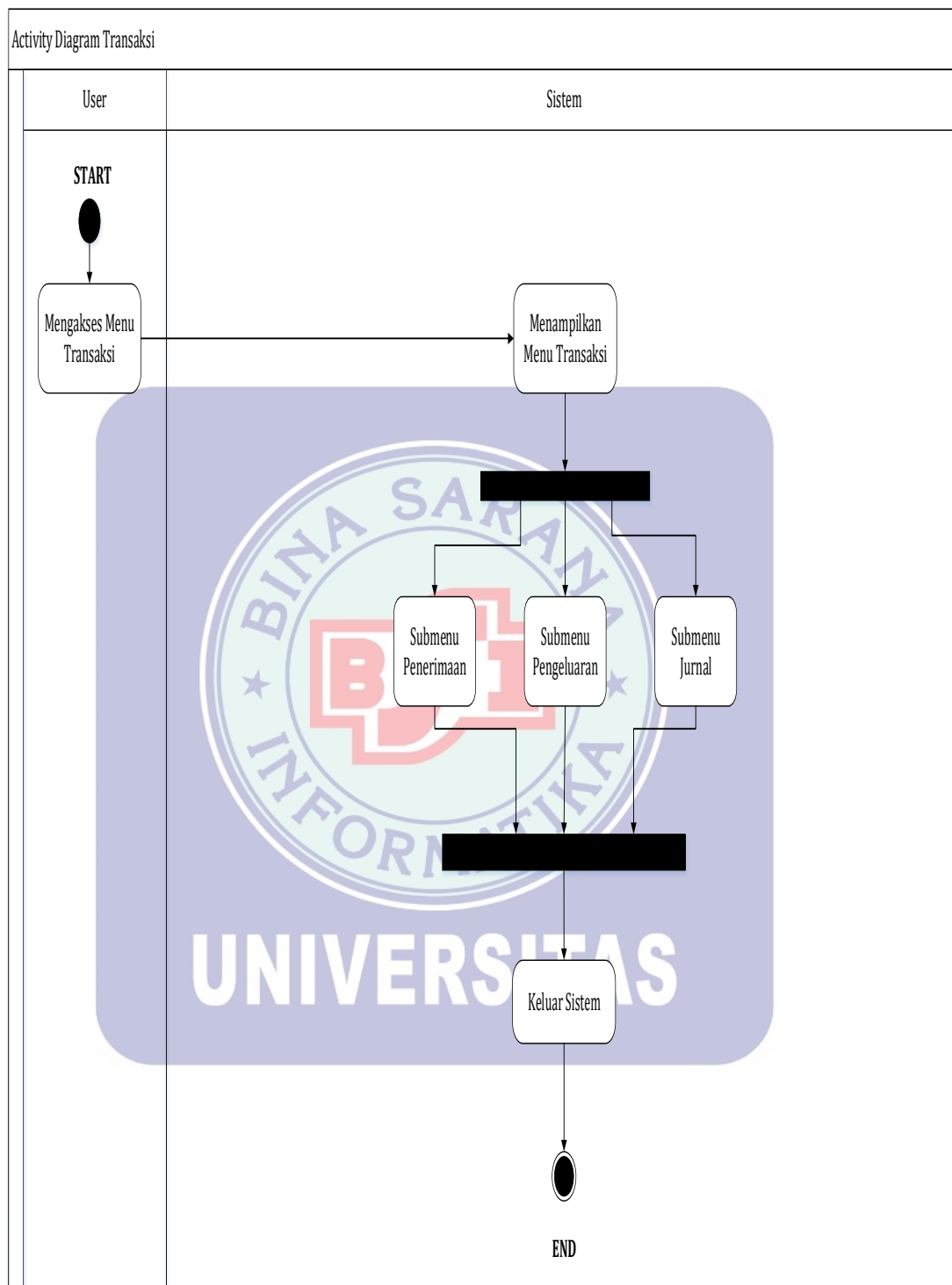
## 7. Activity Diagram Menu Master Submenu Data Akun



Gambar III.16.

Activity Diagram Menu Master Submenu Data Akun

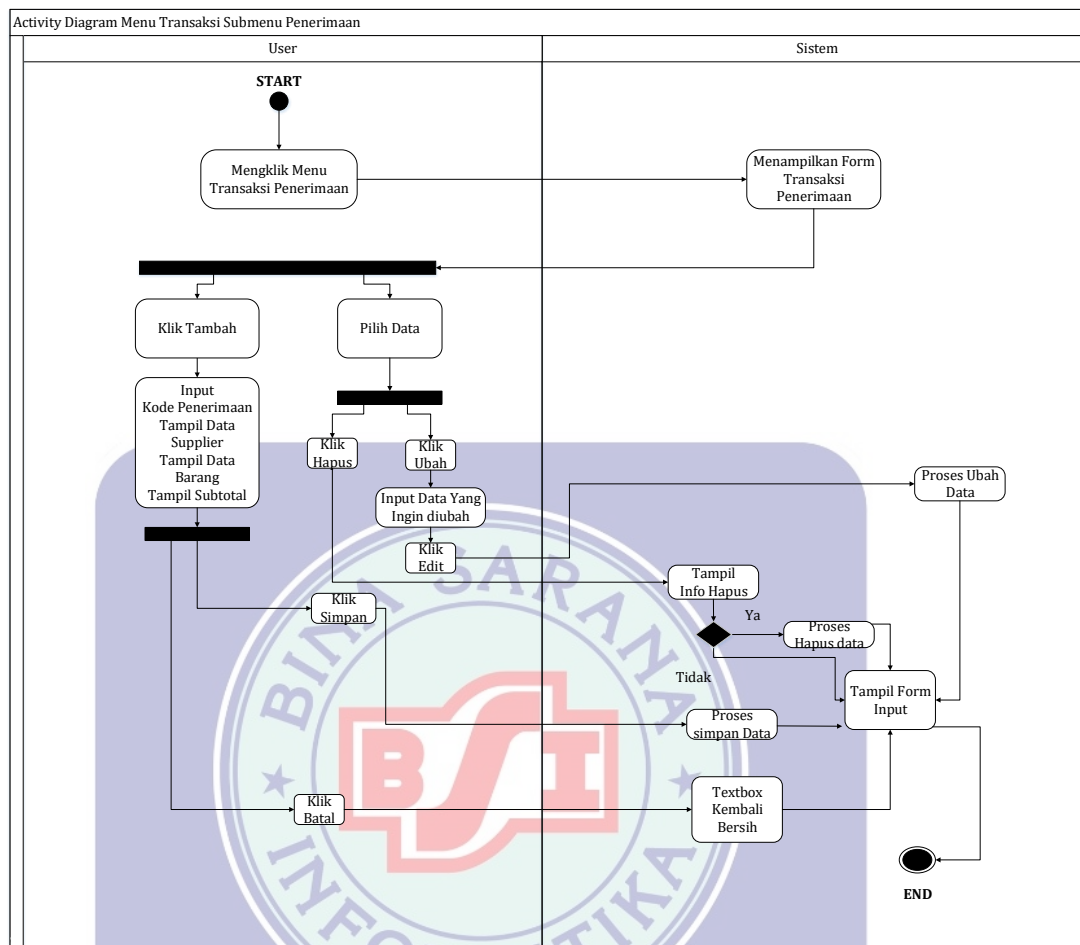
## 8. Activity Diagram Menu Transaksi



**Gambar III.17.**

**Activity Diagram Menu Transaksi**

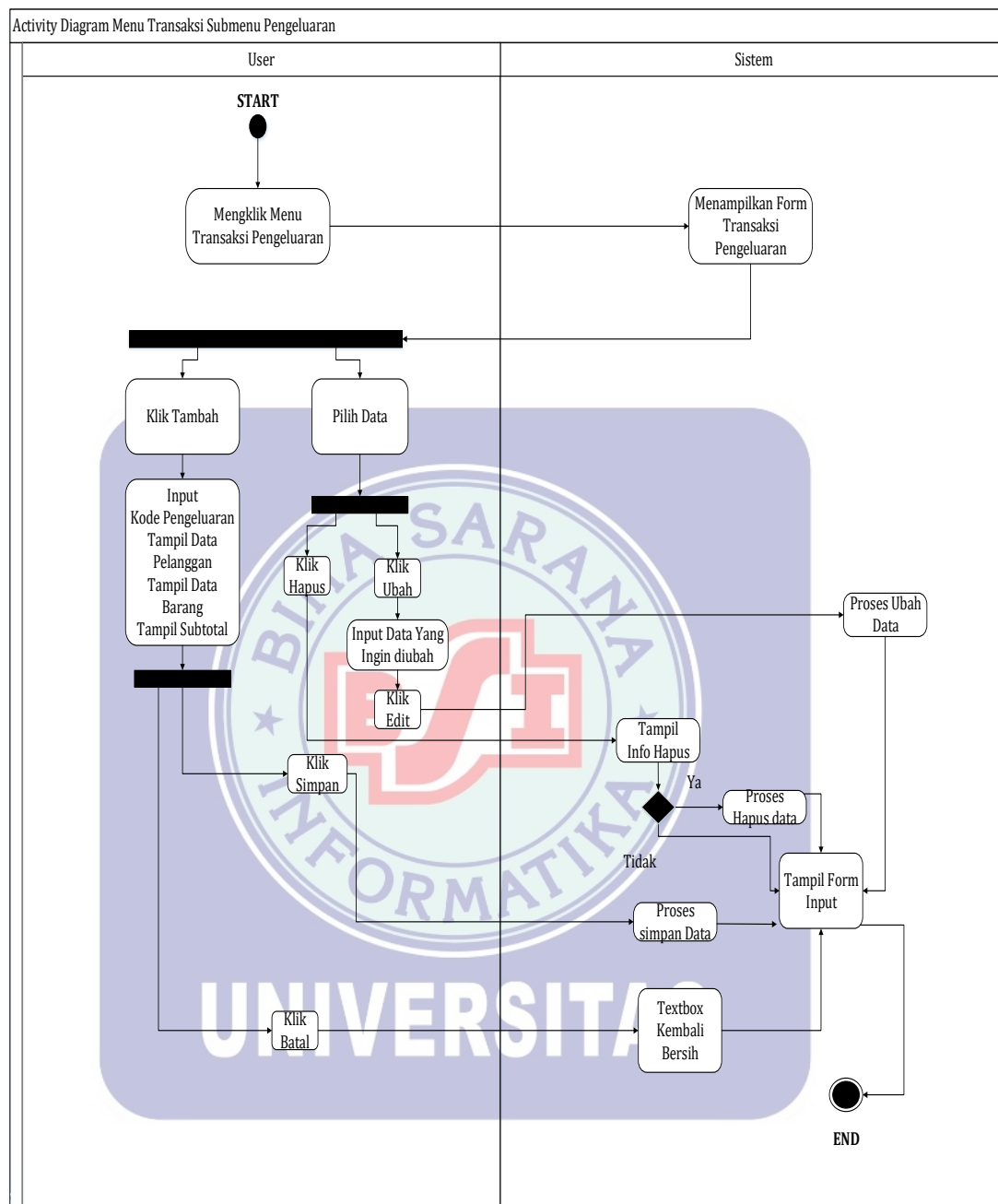
## 9. Activity Diagram Menu Transaksi Submenu Penerimaan Barang



Gambar III.18.

Activity Diagram Menu Transaksi Submenu Penerimaan Barang

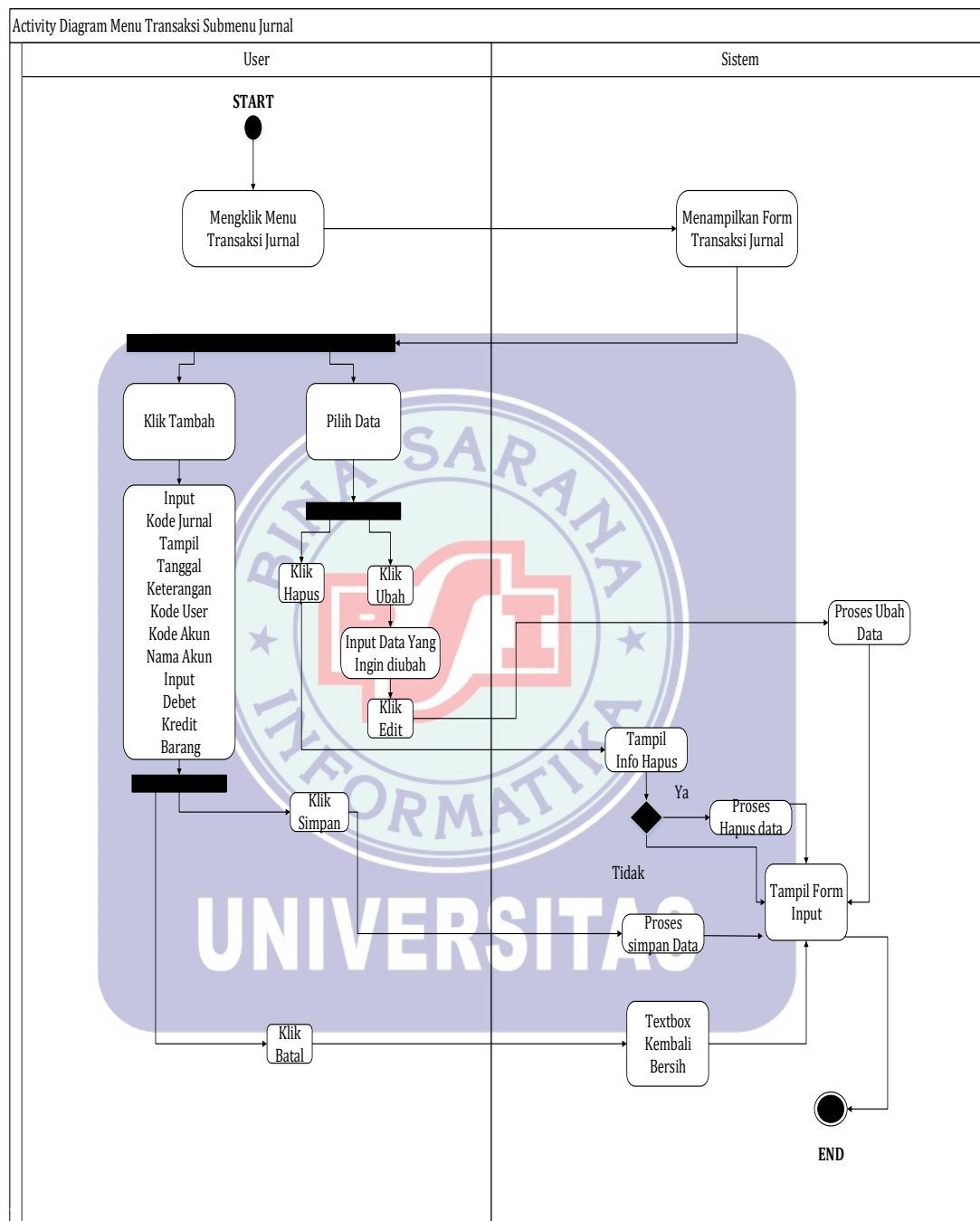
# 10. Activity Diagram Menu Transaksi Submenu Pengeluaran Barang



Gambar III.19.

Activity Diagram Menu Transaksi Submenu Pengeluaran Barang

# 11. Activity Diagram Menu Transaksi Submenu Jurnal

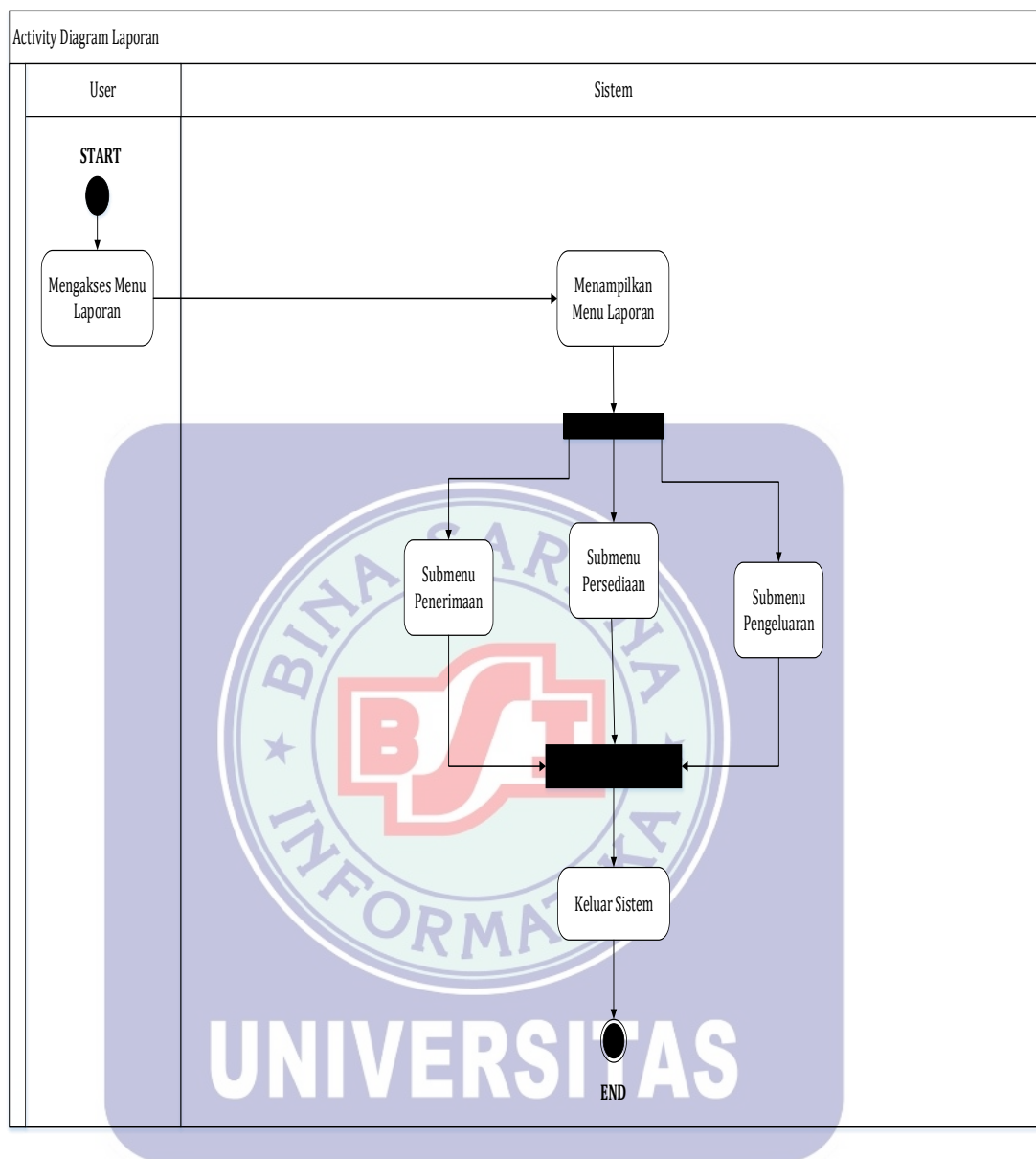


Gambar III.20.

Activity Diagram Menu Transaksi Submenu Jurnal



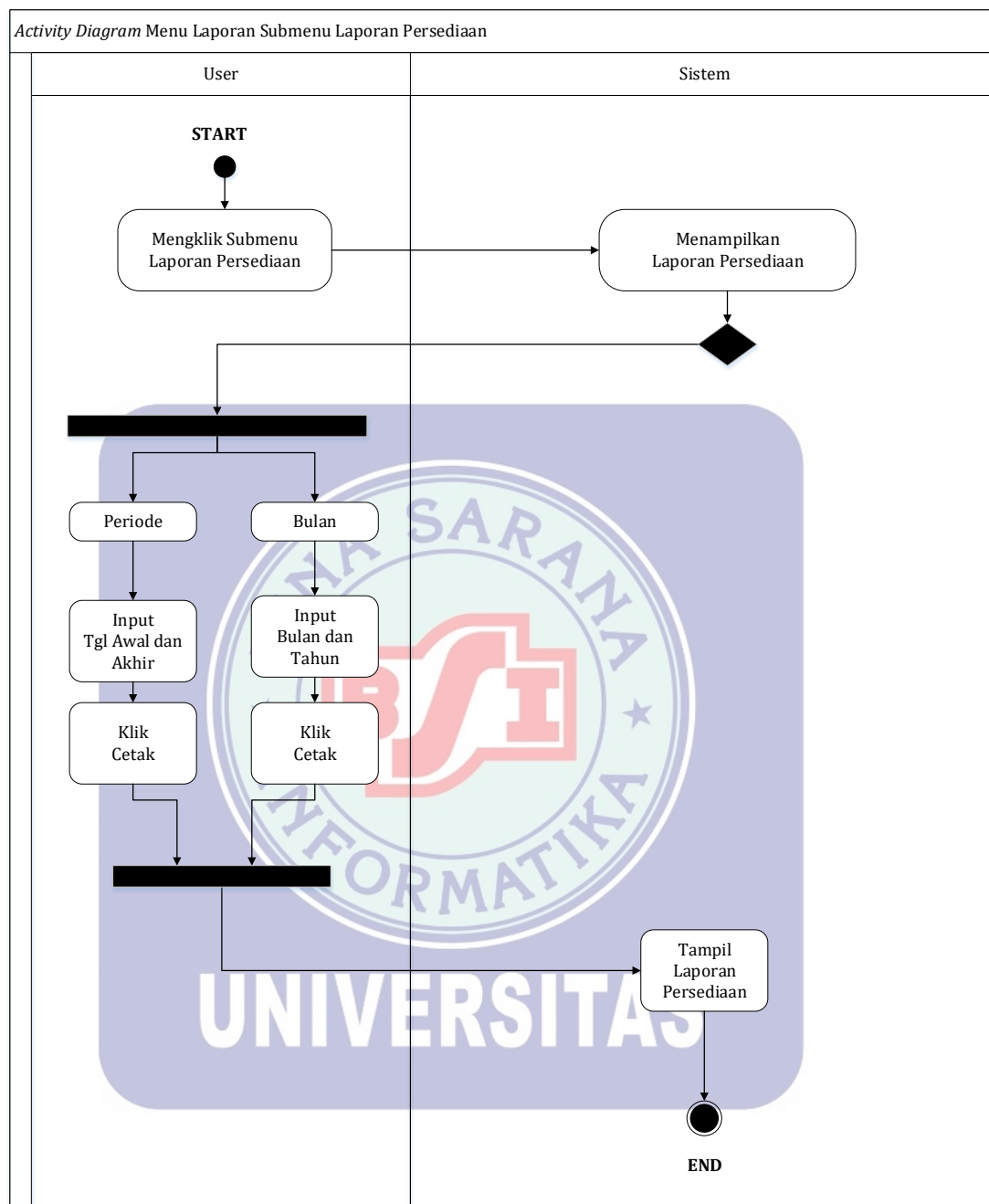
## 12. Activity Diagram Menu Laporan



**Gambar III.21.**

**Activity Diagram Menu Laporan**

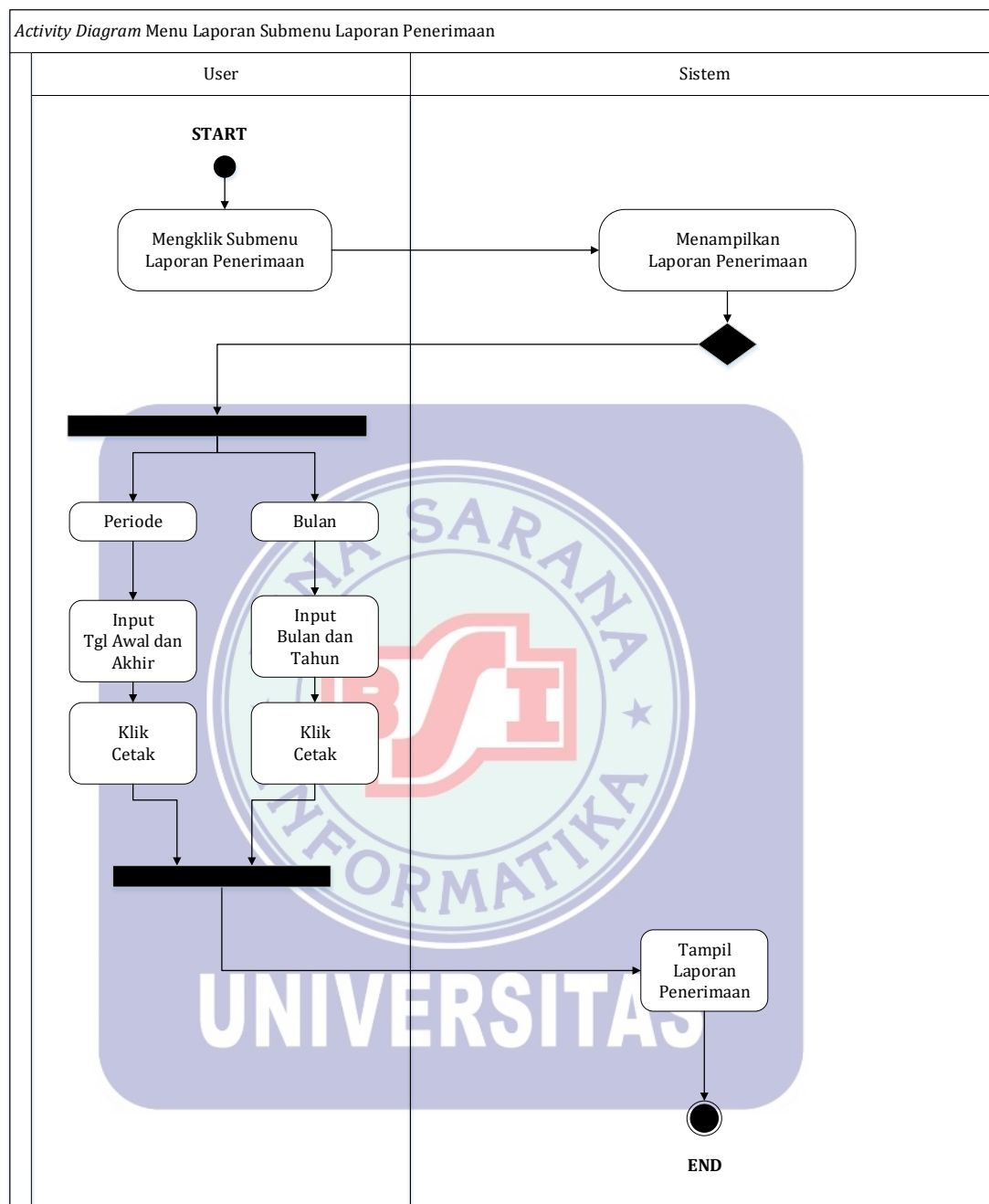
### 13. Activity Diagram Menu Laporan Persediaan



Gambar III.22.

Activity Diagram Menu Laporan Persediaan

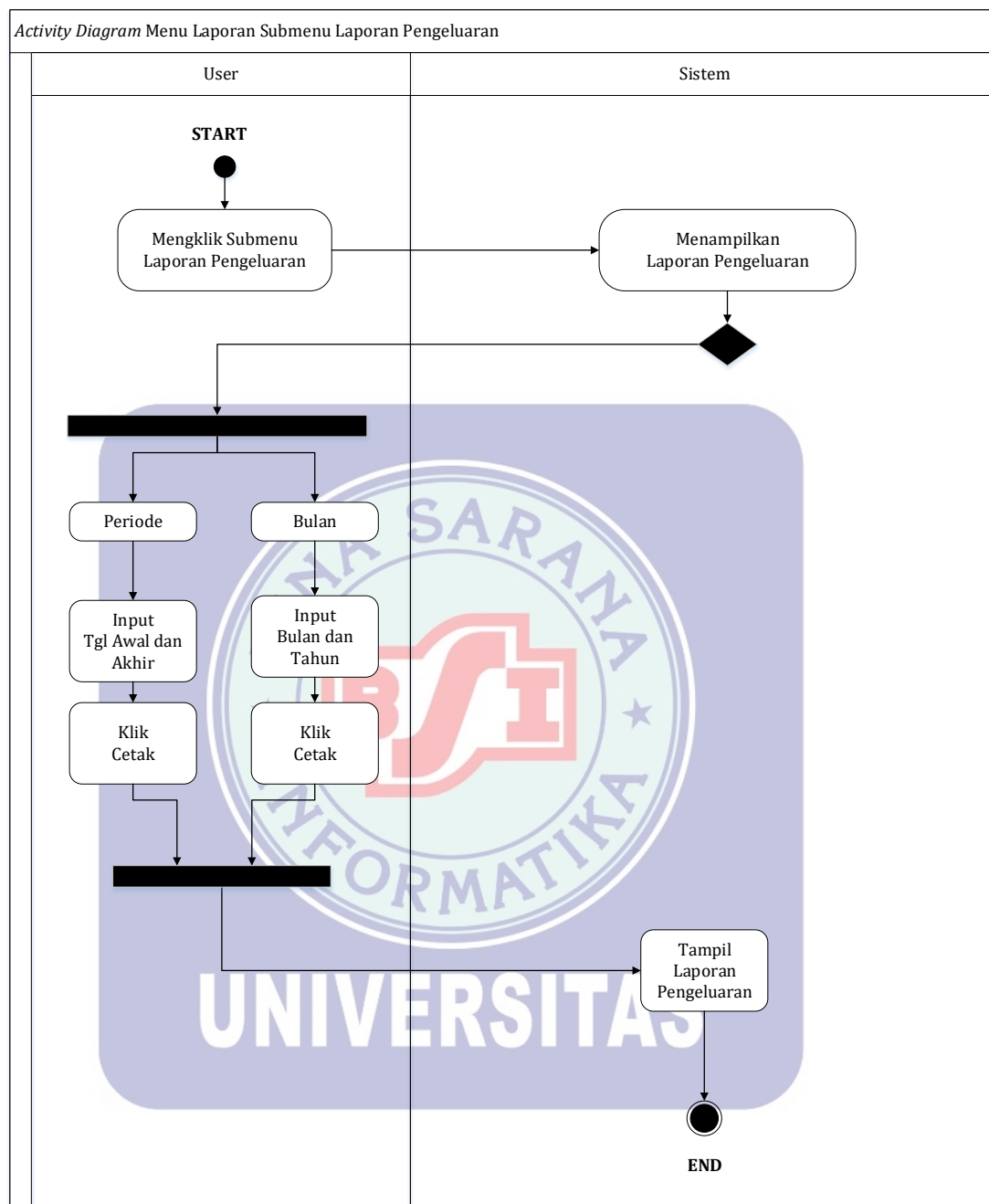
#### 14. Activity Diagram Menu Laporan Penerimaan Barang



Gambar III.23.

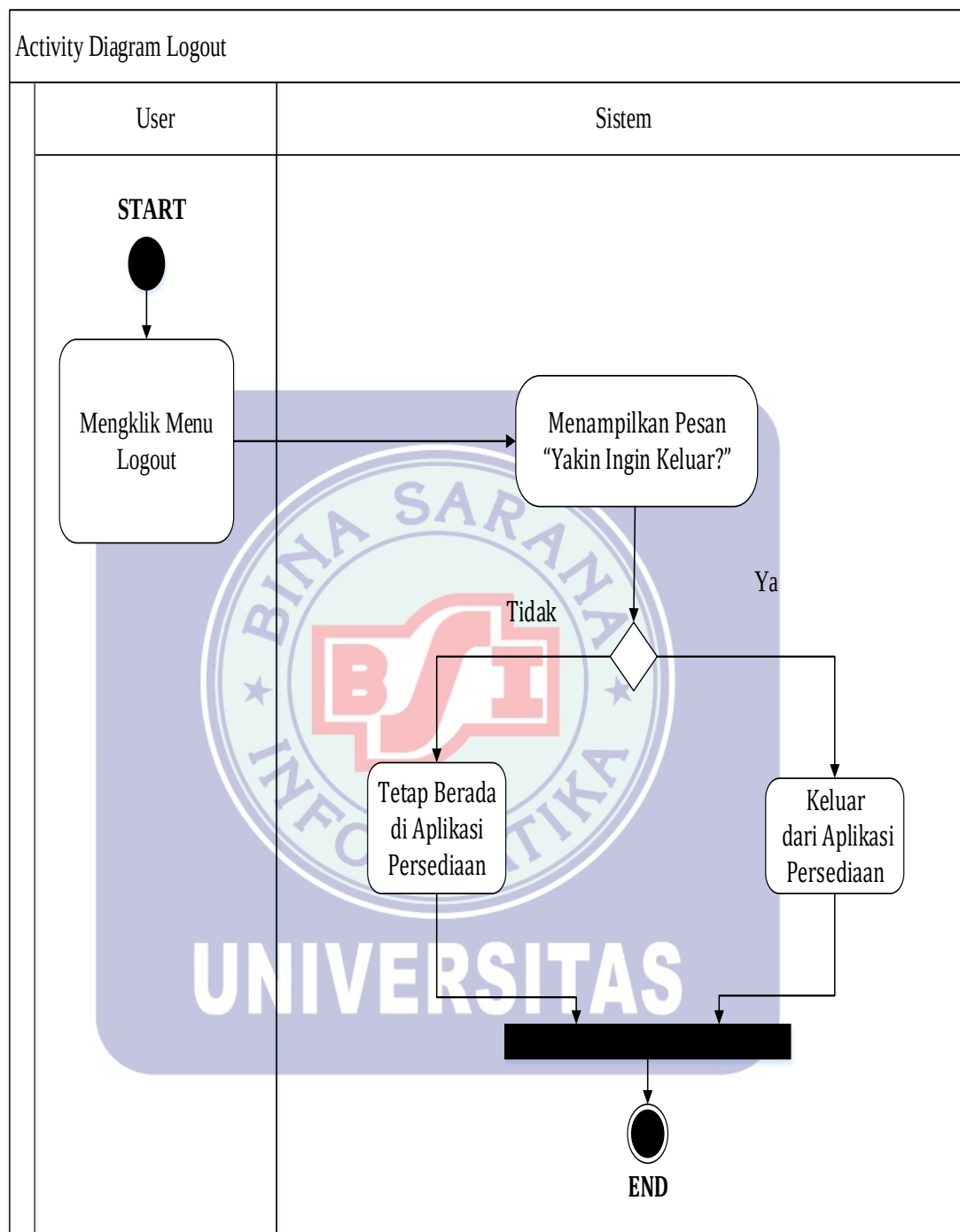
Activity Diagram Menu Laporan Penerimaan Barang

### 15. Activity Diagram Menu Laporan Pengeluaran Barang



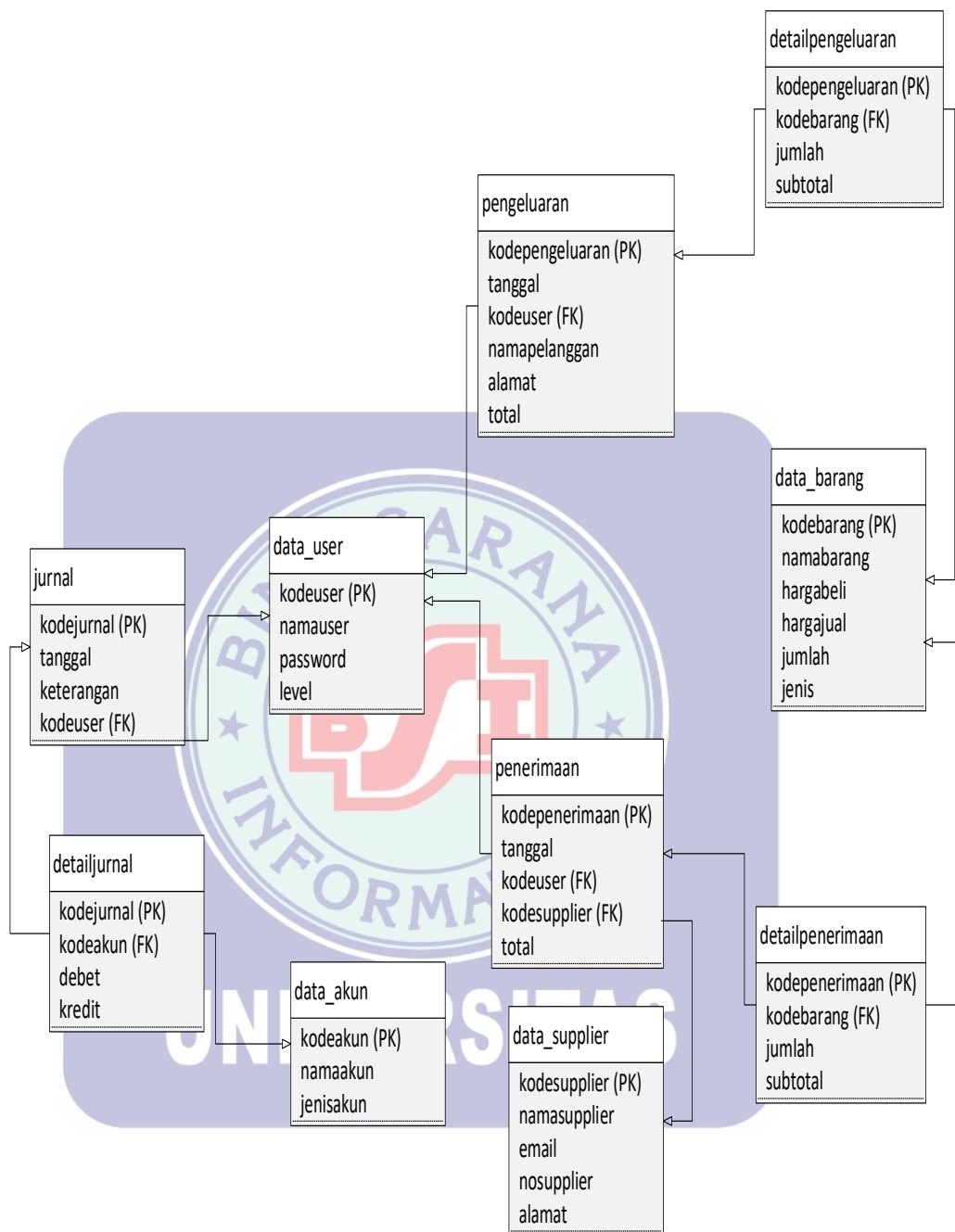
Gambar III.24..

*Activity Diagram Menu Laporan Pengeluaran Barang*

**16. Activity Diagram Menu Logout****Gambar III.25.***Activity Diagram Menu Logout*

### Entity Relationship Diagram (ERD)

### 3.4.2. Logical Record Structure (LRS)



**Gambar III.27.**

**Logical Record Structure (LRS)**

### 3.4.3. Spesifikasi File

Spesifikasi *file* menjelaskan tentang *file-file* atau *table* yang telah terbentuk sebagai tempat *File-file* ini tersimpan dalam *database* yaitu sebagai berikut :

#### 1. Spesifikasi File Data User

Nama File : data\_user  
 Akronim : data\_user.sql  
 Fungsi : sebagai penyimpan data user  
 Tipe File : File Master  
 Organisasi File : Indexed Sequential  
 Akses File : Random  
 Media : Harddisk  
 Panjang record : 38 Byte  
 Kunci field : kodeuser  
 Software : MySQL

**Tabel III.6.**  
**Spesifikasi File Data User**

| No | Elemen Data | Nama Field | Type    | Size | Keterangan  |
|----|-------------|------------|---------|------|-------------|
| 1  | Kode User   | Kodeuser   | Varchar | 5    | Primary Key |
| 2  | Nama User   | Namauser   | Varchar | 20   |             |
| 3  | Password    | Password   | Varchar | 8    |             |
| 4  | Level       | Level      | Varchar | 5    |             |



## 2. Spesifikasi *File Data Supplier*

Nama *File* : data\_supplier  
 Akronim : data\_supplier.sql  
 Fungsi : sebagai penyimpan data *supplier*  
 Tipe *File* : *File Master*  
 Organisasi *File* : *Indexed Sequential*  
 Akses *File* : *Random*  
 Media : *Harddisk*  
 Panjang *record* : 60 *Byte*  
 Kunci *field* : kodesupplier  
 Software : MySQL

**Tabel III.7.**  
**Spesifikasi *File Data Supplier***

| No | Elemen Data          | Nama Field   | Type    | Size | Keterangan  |
|----|----------------------|--------------|---------|------|-------------|
| 1  | Kode <i>Supplier</i> | kodesupplier | Varchar | 5    | Primary Key |
| 2  | Nama <i>Supplier</i> | namesupplier | Varchar | 20   |             |
| 3  | Email                | Email        | Varchar | 20   |             |
| 4  | No <i>Supplier</i>   | nosupplier   | Varchar | 15   |             |
| 5  | Alamat               | Alamat       | Text    |      |             |

### 3. Spesifikasi *File* Data Barang

Nama *File* : data\_barang  
 Akronim : data\_barang.sql  
 Fungsi : sebagai penyimpan data barang  
 Tipe *File* : *File Master*  
 Organisasi *File* : *Indexed Sequential*  
 Akses *File* : *Random*  
 Media : *Harddisk*  
 Panjang *record* : 55 *Byte*  
 Kunci *field* : kodebarang  
 Software : *MySQL*

**Tabel III.8.**  
**Spesifikasi *File* Data Barang**

| No | Elemen Data | Nama Field | Type     | Size | Keterangan  |
|----|-------------|------------|----------|------|-------------|
| 1  | Kode Barang | kodebarang | Varchar  | 5    | Primary Key |
| 2  | Nama Barang | namabarang | Varchar  | 15   |             |
| 3  | Harga Beli  | hargabeli  | Int      | 11   |             |
| 4  | Harga Jual  | hargajual  | Int      | 11   |             |
| 5  | Jumlah      | jumlah     | Smallint | 3    |             |
| 6  | Jenis       | Jenis      | Int      | 10   |             |

#### 4. Spesifikasi *File* Data Akun

Nama *File* : data\_akun  
 Akronim : data\_akunt.sql  
 Fungsi : sebagai penyimpan data akun  
 Tipe *File* : *File Master*  
 Organisasi *File* : *Indexed Sequential*  
 Akses *File* : *Random*  
 Media : *Harddisk*  
 Panjang *record* : 50 Byte  
 Kunci *field* : kodebarang  
 Software : MySQL

**Tabel III.9.**  
**Spesifikasi *File* Akun**

| No | Elemen Data | Nama Field | Type    | Size | Keterangan  |
|----|-------------|------------|---------|------|-------------|
| 1  | Kode Akun   | kodeakun   | Varchar | 5    | Primary Key |
| 2  | Nama Akun   | namaakun   | Varchar | 25   |             |
| 3  | Jenis Akun  | jenisakun  | Int     | 20   |             |

5. Spesifikasi *File* Penerimaan

Nama *File* : penerimaan  
 Akronim : *penerimaan.sql*  
 Fungsi : sebagai penyimpan data penerimaan  
 Tipe *File* : *File Transaksi*  
 Organisasi *File* : *Indexed Sequential*  
 Akses *File* : *Random*  
 Media : *Harddisk*  
 Panjang *record* : 25 Byte  
 Kunci *field* : kodepenerimaan  
 Software : MySQL

**Tabel III.10.**  
**Spesifikasi *File* Penerimaan**

| No | Elemen Data        | Nama <i>Field</i> | Type    | Size | Keterangan  |
|----|--------------------|-------------------|---------|------|-------------|
| 1  | Kode Penerimaan    | Kodepenerimaan    | Varchar | 15   | Primary Key |
| 2  | Tanggal Penerimaan | Tanggalpenerimaan | Date    |      |             |
| 3  | Kode User          | Kodeuser          | Varchar | 5    | Foreign Key |
| 4  | Kode Supplier      | kodesupplier      | Varchar | 5    | Foreign Key |
| 5  | Total              | Total             | Double  |      |             |

6. Spesifikasi *File* Surat Penerimaan Detail

Nama *File* : penerimaan\_detail  
 Akronim : penerimaan\_detail.sql  
 Fungsi : sebagai penyimpan data detail penerimaan  
 Tipe *File* : *File Detail*  
 Organisasi *File* : *Indexed Sequential*  
 Akses *File* : *Random*  
 Media : *Harddisk*  
 Panjang *record* : 42 *Byte*  
 Kunci *field* : -  
 Software : *MySQL*

**Tabel III.11.**  
**Spesifikasi *File* Penerimaan Detail**

| No | Elemen Data     | Nama Field     | Type    | Size | Keterangan |
|----|-----------------|----------------|---------|------|------------|
| 1  | Kode Penerimaan | kodepenerimaan | Varchar | 15   |            |
| 2  | Kode Barang     | kodebarang     | Varchar | 5    |            |
| 3  | Jumlah          | jumlah         | Int     | 11   |            |
| 4  | Sub Total       | subtotal       | Int     | 11   |            |

## 7. Spesifikasi File Pengeluaran

Nama *File* : pengeluaran  
 Akronim : *pengeluaran.sql*  
 Fungsi : sebagai penyimpan data pengeluaran  
 Tipe *File* : *File Transaksi*  
 Organisasi *File* : *Indexed Sequential*  
 Akses *File* : *Random*  
 Media : *Harddisk*  
 Panjang *record* : 50 *Byte*  
 Kunci *field* : kodepengeluaran  
 Software : *MySQL*

**Tabel III.12.**  
**Spesifikasi File Pengeluaran**

| No | Elemen Data         | Nama Field         | Type    | Size | Keterangan  |
|----|---------------------|--------------------|---------|------|-------------|
| 1  | Kode Pengeluaran    | kodepengeluaran    | Varchar | 15   | Primary Key |
| 2  | Tanggal Pengeluaran | tanggalpengeluaran | Date    |      |             |
| 3  | Kode User           | kodeuser           | Varchar | 5    | Foreign Key |
| 4  | Nama Pelanggan      | namapelanggan      | Varchar | 30   |             |
| 5  | Alamat              | alamat             | Text    |      |             |
| 6  | Total               | total              | Double  |      |             |

8. Spesifikasi *File* Detail Pengeluaran

Nama *File* : detail\_pengeluaran  
 Akronim : detail\_pengeluaran.sql  
 Fungsi : sebagai penyimpan data detail pengeluaran  
 Tipe *File* : *File Detail*  
 Organisasi *File* : *Indexed Sequential*  
 Akses *File* : *Random*  
 Media : *Harddisk*  
 Panjang *record* : 42 *Byte*  
 Kunci *field* : -  
 Software : *MySQL*

**Tabel III.13.**  
**Spesifikasi *File* Detail Pengeluaran**

| No | Elemen Data      | Nama <i>Field</i> | Type           | Size | Keterangan         |
|----|------------------|-------------------|----------------|------|--------------------|
| 1  | Kode Pengeluaran | kodepengeluaran   | <i>Varchar</i> | 15   | <i>Foreign Key</i> |
| 2  | Kode Barang      | kodebarang        | <i>Varchar</i> | 5    | <i>Foreign Key</i> |
| 3  | Jumlah           | jumlah            | <i>Int</i>     | 11   |                    |
| 4  | Sub Total        | subtotal          | <i>Int</i>     | 11   |                    |

9. Spesifikasi *File* Jurnal

Nama *File* : jurnal  
 Akronim : *jurnal.sql*  
 Fungsi : sebagai penyimpan data jurnal  
 Tipe *File* : *File Transaksi*  
 Organisasi *File* : *Indexed Sequential*  
 Akses *File* : *Random*  
 Media : *Harddisk*  
 Panjang *record* : 85 *Byte*  
 Kunci *field* : kodejurnal  
 Software : *MySQL*

**Tabel III.14.**  
**Spesifikasi *File* Jurnal**

| No | Elemen Data    | Nama Field    | Type    | Size | Keterangan  |
|----|----------------|---------------|---------|------|-------------|
| 1  | Kode Jurnal    | kodejurnal    | Varchar | 15   | Primary Key |
| 2  | Tanggal Jurnal | tanggaljurnal | Varchar | 30   |             |
| 3  | Keterangan     | keterangan    | Int     | 30   |             |
| 4  | Kode User      | kodeuser      | Int     | 10   | Foreign Key |



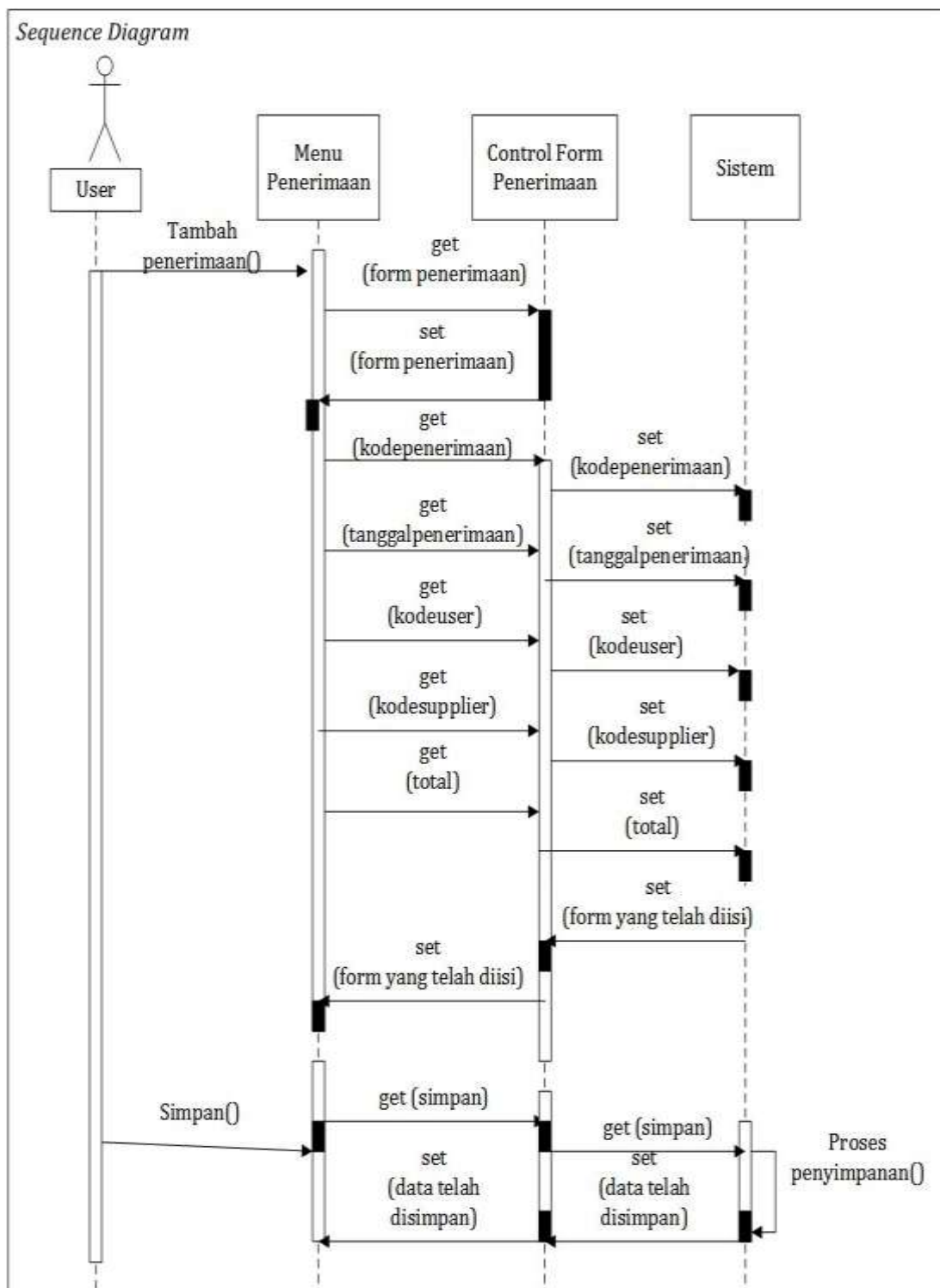
10. Spesifikasi *File* Detail Jurnal

Nama *File* : detail\_jurnal  
 Akronim : detail\_jurnal.sql  
 Fungsi : sebagai penyimpan data detail jurnal  
 Tipe *File* : *File Detail*  
 Organisasi *File* : *Indexed Sequential*  
 Akses *File* : *Random*  
 Media : *Harddisk*  
 Panjang *record* : 47 Byte  
 Kunci *field* : -  
 Software : MySQL

**Tabel III.15.**  
**Spesifikasi *File* Detail Jurnal**

| No | Elemen Data | Nama Field | Type    | Size | Keterangan         |
|----|-------------|------------|---------|------|--------------------|
| 1  | Kode Jurnal | kodejurnal | Varchar | 15   | <i>Foreign Key</i> |
| 2  | Kode Akun   | kodeakun   | Varchar | 10   | <i>Foreign Key</i> |
| 3  | Debet       | Debet      | Int     | 11   |                    |
| 4  | Kredit      | Kredit     | Int     | 11   |                    |

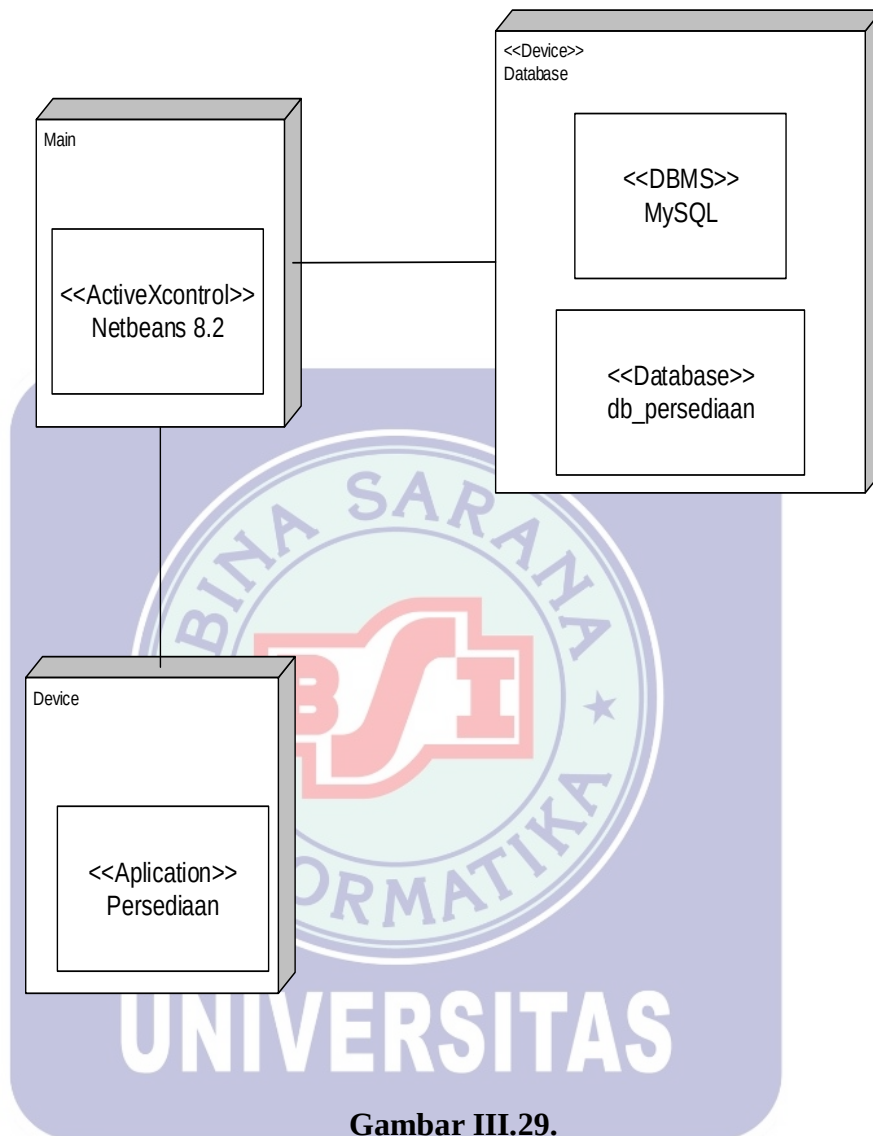
### 3.4.4. Sequence Diagram



Gambar III.28.

Sequence Diagram

### 3.4.5. *Deployment Diagram*



**Gambar III.29.**

***Deployment Diagram***

### 3.4.6. Use Interface

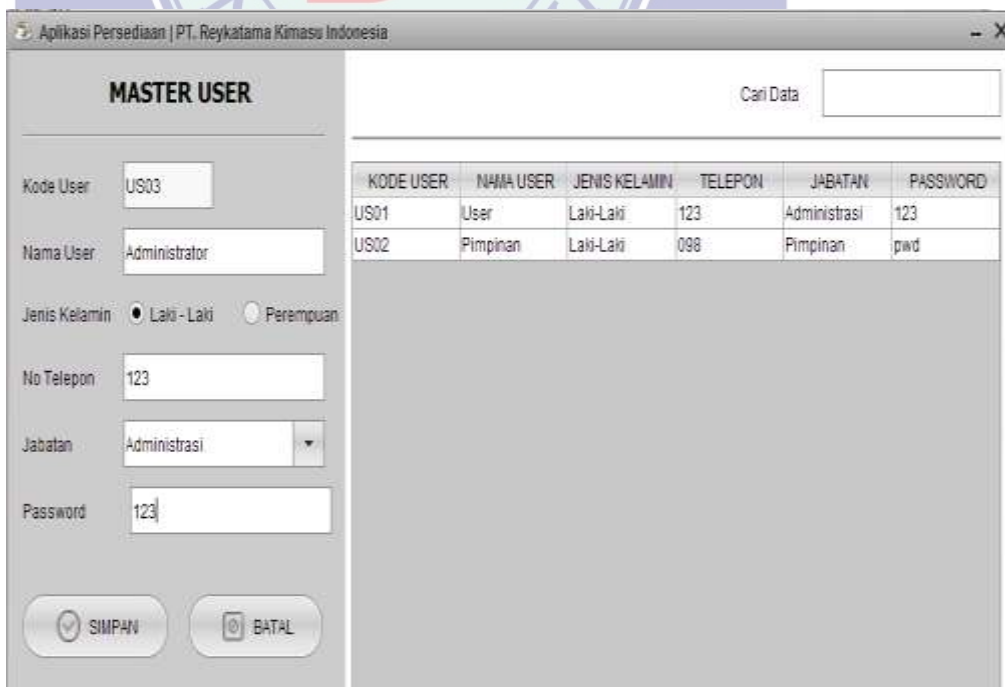
Berdasarkan analisis kebutuhan *software* yang telah dirancang, berikut ini adalah desain tampilan submenu program yang dibuat penulis:

#### 1. Menu *Login*



**Gambar III.30.**  
**Menu *Login***

#### 2. Menu Master Submenu Data *User*



| KODE USER | NAMA USER | JENIS KELAMIN | TELEPON | JABATAN      | PASSWORD |
|-----------|-----------|---------------|---------|--------------|----------|
| US01      | User      | Laki-Laki     | 123     | Administrasi | 123      |
| US02      | Pimpinan  | Laki-Laki     | 098     | Pimpinan     | pwd      |

**Gambar III.31.**

**Menu Master Submenu Data *User***

### 3. Menu Master Submenu Data Barang

**MASTER BARANG**

Kode Barang: BR03

Nama Barang: Tang

Stok: 100

SIMPAN BATAL

Cari Data

| KODE BARANG | NAMA BARANG | STOK |
|-------------|-------------|------|
| BR01        | Barang 1    | 8500 |
| BR02        | Barang 2    | 1000 |

**Gambar III.32.**

### **Menu Master Submenu Data Barang**

### 4. Menu Master Submenu Data Supplier

**MASTER SUPPLIER**

Kode Supplier: SP02

Nama Supplier: PT Fujitsu

Alamat: Karawang

SIMPAN BATAL

Cari Data

| KODE SUPPLIER | NAMA SUPPLIER     | ALAMAT   |
|---------------|-------------------|----------|
| SP01          | Supplier Sekunder | Karawang |

**Gambar III.33.**

### **Menu Master Submenu Data Supplier**

5. Menu Master Submenu Akun

**MASTER AKUN**

Cari Data

| KODE AKUN | NAMA AKUN  | JENIS AKUN |
|-----------|------------|------------|
| AN01      | Kas        | Harta      |
| AN02      | Persediaan | Aktiva     |

Kode Akun: AN03  
 Nama Akun: Bank Mandiri  
 Jenis Akun: Harta

SIMPAN BATAL

Gambar III.34.

Menu Master Submenu Data Akun

6. Menu Transaksi Submenu Permintaan Barang

**TRANSAKSI PERMINTAAN**

2020-07-07 13:49:37 US01

| Kode Barang | Nama Barang | Jumlah Permintaan |
|-------------|-------------|-------------------|
|-------------|-------------|-------------------|

No Permintaan: NO-2007005  
 Kode Barang: BR02 Nama Barang: Barang 2  
 Jumlah Pesan: 200  
 Supplier: Supplier Sekunder

SIMPAN BATAL

Riwayat Transaksi

Gambar III.35.

Menu Transaksi Submenu Permintaan Barang

7. Menu Transaksi Submenu Pengeluaran Barang

Gambar III.36.

Menu Transaksi Submenu Pengeluaran Barang

8. Transaksi Jurnal

| No Jurnal | Kode Akun | Nama Akun  | Debet | Kredit |
|-----------|-----------|------------|-------|--------|
| JU2007004 | AN02      | Persediaan | 1000  | 0      |
| JU2007004 | AN01      | Kas        | 0     | 1000   |

Gambar III.37.

Menu Transaksi Submenu Jurnal

### 3.5. Implementasi

#### 3.5.1. Code Generation

```

/*
 * 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.
 */
package program;

import java.awt.event.KeyEvent;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

public class Login extends javax.swing.JFrame {

    int x, y;
    koneksi kon = new koneksi();

    public Login() {
        initComponents();
        kon.setKoneksi();
    }

    ImageIcon gambar = new ImageIcon("src/program/Home.png");

    /**
     * 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() {

        jPanel2 = new javax.swing.JPanel();
        jLabel5 = new javax.swing.JLabel();
        jPanel3 = new javax.swing.JPanel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        kodeuser = new javax.swing.JTextField();
        pwd = new javax.swing.JPasswordField();
        jPanel4 = new javax.swing.JPanel();
        login = new javax.swing.JButton();
        batal = new javax.swing.JButton();
    }

```



```

jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DO_NOTHING_ON_CLOSE);
setTitle("Login");
setLocationByPlatform(true);
setUndecorated(true);
setResizable(false);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jPanel2.setBackground(java.awt.Color.white);
jPanel2.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
    public void mouseDragged(java.awt.event.MouseEvent evt) {
        jPanel2MouseDragged(evt);
    }
});
jPanel2.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mousePressed(java.awt.event.MouseEvent evt) {
        jPanel2MousePressed(evt);
    }
});
jPanel2.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel5.setBackground(new java.awt.Color(255, 255, 255));
jLabel5.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N
jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel5.setText("LOGIN");
jLabel5.setCursor(new java.awt.Cursor(java.awt.Cursor.MOVE_CURSOR));
jPanel2.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(-2, 0, 260, 40));

jPanel3.setBackground(new java.awt.Color(255, 255, 255));
jPanel3.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel3.setOpaque(false);

jLabel3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel3.setText("KODE USER");

jLabel4.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel4.setText("PASSWORD");

kodeuser.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        kodeuserKeyPressed(evt);
    }
    public void keyTyped(java.awt.event.KeyEvent evt) {
        kodeuserKeyTyped(evt);
    }
});

```

```

    }
    });

    pwd.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyPressed(java.awt.event.KeyEvent evt) {
            pwdKeyPressed(evt);
        }
    });

    javax.swing.GroupLayout jPanel3Layout = new
    javax.swing.GroupLayout(jPanel3);
    jPanel3.setLayout(jPanel3Layout);
    jPanel3Layout.setHorizontalGroup(

    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    )
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(jLabel4)
                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE,
                        95, Short.MAX_VALUE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                )
                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(kodeuser,
                        javax.swing.GroupLayout.PREFERRED_SIZE, 96,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(pwd, javax.swing.GroupLayout.PREFERRED_SIZE,
                        150, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
                        Short.MAX_VALUE)
                )
            );
    jPanel3Layout.setVerticalGroup(

    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    )
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addContainerGap()
                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel3)
                    .addComponent(kodeuser,
                        javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
    .addComponent(jLabel4)
    .addComponent(pwd, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
);

jPanel2.add(jPanel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 46,
290, -1));

jPanel4.setBackground(new java.awt.Color(255, 255, 255));
jPanel4.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
jPanel4.setOpaque(false);

login.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
login.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/program/72.png"))); // NOI18N
login.setText("LOGIN");
login.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        loginActionPerformed(evt);
    }
});

batal.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
batal.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/program/33.png"))); // NOI18N
batal.setText("BATAL");
batal.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        batalActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel4Layout = new
javax.swing.GroupLayout(jPanel4);
jPanel4.setLayout(jPanel4Layout);
jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)

```

```

        .addGroup(jPanel4Layout.createSequentialGroup())
        .addGap(47, 47, 47)
        .addComponent(login)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(batal)
        .addGap(55, 55, 55))
    );
    jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
        .addGroup(jPanel4Layout.createSequentialGroup())
        .addContainerGap()

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
            .addComponent(login)
            .addComponent(batal))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

    jPanel2.add(jPanel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
150, 290, -1));

    jLabel1.setBackground(new java.awt.Color(255, 255, 255));
    jLabel1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
    jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel1.setText("X");

jLabel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.borde
r.BevelBorder.RAISED));
    jLabel1.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
    jLabel1.setOpaque(true);
    jLabel1.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            jLabel1MouseClicked(evt);
        }
    });
    jPanel2.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(260,
0, 27, 24));

    getContentPane().add(jPanel2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 290, 210));

    setSize(new java.awt.Dimension(291, 209));
    setLocationRelativeTo(null);

```

```

} // </editor-fold>

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

    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        pwd.requestFocus();
    }
}

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

    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        masuk();
    }
}

private void loginActionPerformed(java.awt.event.ActionEvent evt) {
    masuk();
}

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

    kodeuser.setText("");
    pwd.setText("");
    kodeuser.requestFocus();
}

private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

private void kodeuserKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    evt.setKeyChar(Character.toUpperCase(evt.getKeyChar()));
}

private void jPanel2MouseDragged(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int a, b;
    a = evt.getXOnScreen();
    b = evt.getYOnScreen();
    this.setLocation(a - x, b - y);
}

private void jPanel2MousePressed(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    x = evt.getX();
    y = evt.getY();
}

```



```

}

private void masuk() {
    try {
        kon.setKoneksi();
        String sql = "select * from data_user where kodeuser='" +
kodeuser.getText() + "' and password='" + pwd.getText() + "'";
        ResultSet rs = kon.st.executeQuery(sql);
        if (rs.next()) {
            String nama = rs.getString("namauser");
            JOptionPane.showMessageDialog(null, "Selamat Datang " + nama + "");
            menuUtama FU = new menuUtama();
            FU.setIconImage(gambar.getImage());
            FU.setLocationRelativeTo(null);
            FU.setVisible(true);
            FU.setExtendedState(JFrame.MAXIMIZED_BOTH);
            FU.ku = rs.getString("kodeuser");
            FU.nu = rs.getString("namauser");
            FU.ak = rs.getString("level");
            dispose();
        } else {
            JOptionPane.showMessageDialog(null, "Data Tidak Ditemukan !!");
            kodeuser.setText("");
            pwd.setText("");
            kodeuser.grabFocus();
        }
    } catch (SQLException e) {
    }
}

/**
 * @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;
            }
        }
    }
}

```



```
//</editor-fold>

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

// Variables declaration - do not modify
private javax.swing.JButton batal;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JTextField kodeuser;
private javax.swing.JButton login;
private javax.swing.JPasswordField pwd;
// End of variables declaration
}
```





3.5.2. *BlackTesting*

**Tabel III.16.**  
**Pengujian Menu Login**

| No | Skenario Pengujian                                                                                                       | Test Case                                                                  | Hasil Yang Diharapkan                                                                  | Hasil Pengujian | Kesimpulan |
|----|--------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|----------------------------------------------------------------------------------------|-----------------|------------|
| 1  | Mengosongkan semua isian data <i>login</i> , kemudian langsung klik tombol ' <i>Login</i> '                              | Kode <i>User</i> :<br>(Kosong)<br><i>Password</i> :<br>(Kosong)            | Sistem akan menolak akses <i>login</i> dan menampilkan pesan "Data tidak ditemukan !!" | Sesuai Harapan  | Valid      |
| 2  | Hanya mengisi data kode <i>user</i> , mengosongkan data <i>password</i> , kemudian langsung klik tombol ' <i>Login</i> ' | Kode <i>User</i> : U-1<br>(Benar)<br><i>Password</i> :<br>(Kosong)         | Sistem akan menolak akses <i>login</i> dan menampilkan pesan "Data tidak ditemukan !!" | Sesuai Harapan  | Valid      |
| 3  | Hanya mengisi data <i>password</i> dan mengosongkan kode <i>user</i> , kemudian langsung klik tombol ' <i>Login</i> '    | Kode <i>User</i> :<br>(Kosong)<br><i>Password</i> :<br>12345<br>(Benar)    | Sistem akan menolak akses <i>login</i> dan menampilkan pesan "Data tidak ditemukan !!" | Sesuai Harapan  | Valid      |
| 4  | Menginputkan dengan kondisi salah satu data benar dan salah satu lagi salah, kemudian klik tombol ' <i>Login</i> '       | Kode <i>User</i> : U-1<br>(Benar)<br><i>Password</i> :<br>11111<br>(Salah) | Sistem akan menolak akses <i>login</i> dan menampilkan pesan "Data tidak ditemukan !!" | Sesuai Harapan  | Valid      |
| 5  | Mengisi kode <i>user</i> dan <i>password</i> atau salah satunya, kemudian klik tombol ' <i>Batal</i> '                   | Kode <i>User</i> : U-1<br>(Benar)<br><i>Password</i> :<br>12345<br>(Benar) | Menampilkan <i>inputan</i> data menjadi kosong                                         | Sesuai Harapan  | Valid      |

|   |                                                                                                      |                                                                   |                                        |                |       |
|---|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|----------------------------------------|----------------|-------|
| 6 | Mengisi kode <i>user</i> dan <i>password</i> atau salah satunya, kemudian <i>klik</i> tombol 'Login' | Kode <i>User</i> : U-1 (Benar)<br><i>Password</i> : 12345 (Benar) | Masuk kedalam sistem persediaan barang | Sesuai Harapan | Valid |
|---|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|----------------------------------------|----------------|-------|

### 3.5.3. Spesifikasi *Hardware* dan *Software*

**Tabel III.17.**  
**Spesifikasi *Hardware* dan *Software***

| Kebutuhan       | Keterangan                                                                                       |
|-----------------|--------------------------------------------------------------------------------------------------|
| Sistem Operasi  | Windows 7 Profesional                                                                            |
| Processor       | Intel (R) Pentium(R) Dual CPU E2160 1.80 Ghz                                                     |
| RAM             | 1.00 GB                                                                                          |
| Harddisk        | 232 GB                                                                                           |
| Monitor         | Generic PnP                                                                                      |
| <i>Keyboard</i> | Standard PS/2                                                                                    |
| <i>Mouse</i>    | Microsoft PS/2                                                                                   |
| <i>Printer</i>  | HP Deskjet 2000 J210 Series                                                                      |
| <i>Software</i> | Bahasa Pemrograman : Java<br>Aplikasi Pendukung : <i>NetBeans IDE 8.2</i><br>DBMS : <i>MySQL</i> |