

BAB III

PEMBAHASAN

3.1. Tinjauan Perusahaan

Dalam tinjauan perusahaan ini, penulis menuliskan beberapa hal seperti sejarah perusahaan, struktur organisasi perusahaan dengan tugas dan fungsi dari masing-masing bagian yang ada di dalam Apotek Cito Karawang.

3.1.1. Sejarah Perusahaan

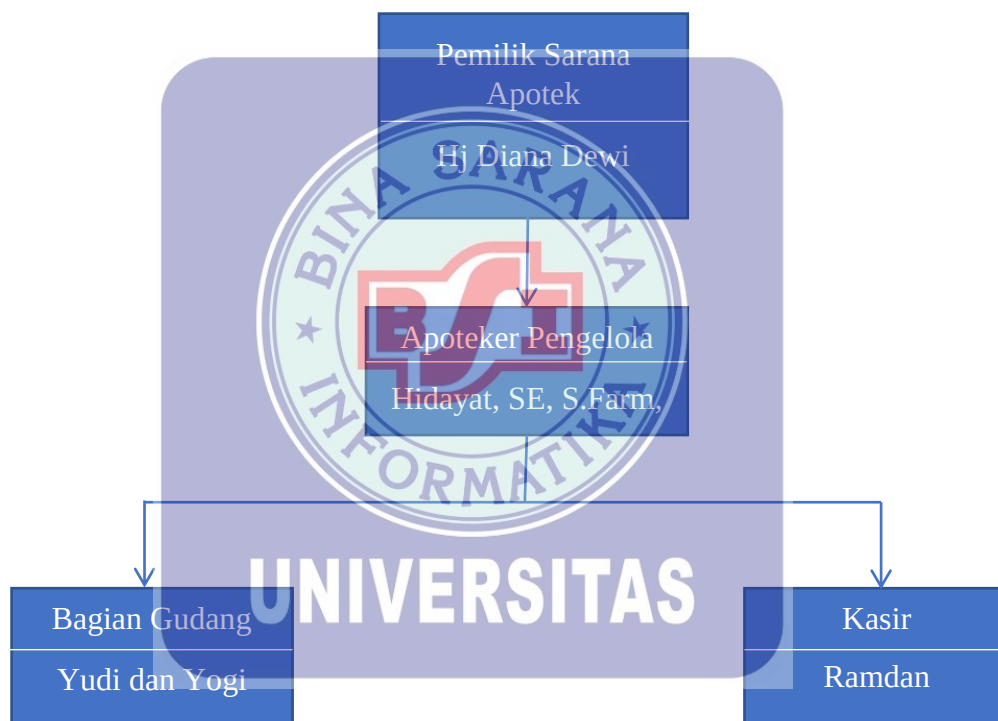
Apotek Cito Karawang didirikan pada tahun 2016 beralamat di Jl. Kertabumi No.27 Karawang, Kabupaten Karawang Provinsi Jawa Barat. Apotik ini memiliki 1 orang Asisten Apoteker dan Penanggung jawab. Toko obat ini telah mendapat ijin dari Departemen Kesehatan RI dengan Surat Ijin Kerja: KP.01.01.1.3.0587.

Pada tahun 2005 Apotek ini berkembang dengan pesat. Sistem penjualan Apotek langsung ke konsumen yang menyediakan obat dengan resep, obat tanpa resep (obat bebas dan obat bebas terbatas). Apotek ini melakukan kegiatannya mulai pukul 16.00 sampai dengan pukul 22.00, buka setiap hari Senin-Jum'at.

3.1.2. Struktur Organisasi dan Fungsi

Suatu perusahaan harus mempunyai suatu struktur organisasi, karena struktur organisasi merupakan suatu kerangka yang dapat menunjukkan pola hubungan diantara fungsi-fungsi, tugas, wewenang, dan tanggung jawab masingmasing anggota organisasi menjadi jelas, sehingga tidak terjadi tumpang tindih dalam pelaksanaan tugasnya masing-masing.

Setiap anggota organisasi dapat dengan mudah memahami tugas yang harus dilakukannya dan kepada siapa ia harus bertanggung jawab. Dengan demikian kegiatan perusahaan dapat berjalan dengan lancar. Selain itu, dengan adanya struktur organisasi yang baik, maka akan tercipta kerja sama dan hubungan kerja yang serasi antar anggota organisasi dan antar bagian dalam organisasi. Adapun struktur organisasi yang terdapat pada Apotek adalah sebagai berikut :



Sumber: Apotek Cito Karawang

Gambar III.1.
Struktur Organisasi

Untuk melengkapi struktur organisasi suatu perusahaan diperlukan uraian tugas yang akan menjelaskan wewenang dan tanggung jawab masing-masing fungsi dalam perusahaan. Uraian jabatan pada Apotek Cito adalah sebagai berikut :

1. Pemilik sarana Apotek

Sebagai sumber dana pendirian Apotek.

2. Tugas Apoteker Pengelola :

- a. Memimpin seluruh kegiatan apotek, baik kegiatan teknis maupun non teknis kefarmasian sesuai dengan ketentuan maupun perundangan yang berlaku.
- b. Mengatur, melaksanakan dan mengawasi administrasi.
- c. Mengusahakan agar apotek yang dipimpinnya dapat memberikan hasil yang optimal sesuai dengan rencana kerja dengan cara meningkatkan omset, mengadakan pembelian yang sah dan penekanan biaya serendah mungkin.
- d. Melakukan pengembangan usaha apotek.

3. Tugas Kasir

- a. Melayani konsumen atas permintaan obat resep maupun non resep.
- b. Menyetorkan hasil penjualan obat

4. Tugas Bagian Gudang

- a. Mengecek stok barang di gudang
- b. Menerima barang dari supplier
- c. Bertanggung jawab dengan barang yang ada di gudang

3.2. Tinjauan Kasus

Dalam tinjauan kasus ini, penulis menuliskan beberapa hal seperti proses bisnis sistem berjalan, *activity diagram*, spesifikasi dokumen masukan, spesifikasi dokumen keluaran, permasalahan pokok dan pemecahan masalah.

3.2.1. Proses Bisnis Sistem Berjalan

Berikut ini adalah tahapan-tahapan dari sistem yang sedang berjalan pada Apotek Cito Karawang:

1. Proses Pemeriksaan Obat dan Pemesanan Obat

Bagian gudang mengecek ketersediaan obat pada apotek, jika stok obat kurang dari stok minimum persediaan gudang, maka bagian gudang membuat Surat Pesanan (SP) dan diserahkan kepada Apoteker untuk keperluan ACC dan pemberian tanda tangan. Setelah disetujui oleh Apoteker, Apoteker menyerahkan SP kepada *Supplier* untuk keperluan pemesanan obat.

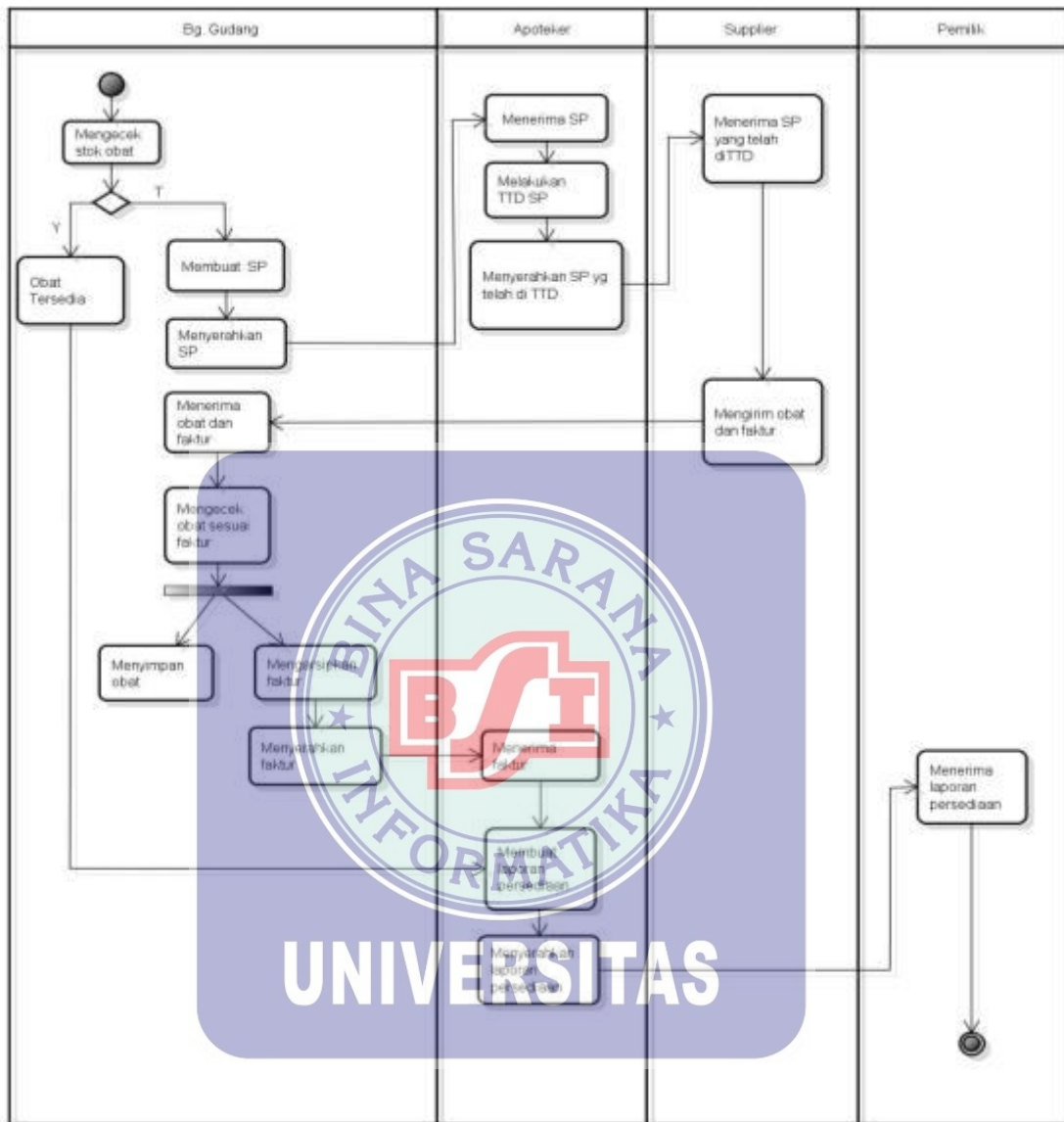
2. Proses Penerimaan Obat dan Pengecekan Obat

Setelah melakukan pemesanan obat, *supplier* memberikan obat dan faktur kepada bagian gudang. Bagian gudang mengecek kesesuaian antara obat dengan faktur.

3. Proses Pembuatan Laporan Persediaan

Kemudian bagian gudang menyimpan obat dan mengarsipkan faktur. Faktur yang telah diarsipkan kemudian diserahkan kepada apoteker. Apoteker membuat laporan persediaan perbulan berdasarkan faktur dan transaksi penjualan obat yang kemudian diserahkan kepada Pemilik.

3.2.2. Activity Diagram



Gambar III.2.

Activity Diagram Proses Bisnis Sistem Berjalan

3.2.3. Spesifikasi Dokumen Masukan

Nama Dokumen : Faktur Penjualan

Fungsi : Sebagai bukti pembelian obat.

Sumber : Supplier

Tujuan : Bagian Gudang

Frekuensi : Saat pengiriman obat ke Apotek
Media : kertas
Bentuk : Lampiran A.1

3.2.4. Spesifikasi Dokumen Keluaran

Nama Dokumen : Surat Pesanan
Fungsi : Sebagai bukti pemesanan obat
Sumber : Bagian Gudang
Tujuan : Supplier
Frekuensi : Saat pemesanan obat ke Supplier
Media : kertas
Bentuk : Lampiran B-1

Nama Dokumen : Laporan Stok Obat
Fungsi : Sebagai laporan stok barang perbulan
Sumber : Apoteker
Tujuan : Pemilik Apotek
Frekuensi : Saat laporan persediaan perbulan
Media : kertas
Bentuk : Lampiran B-2

3.2.5. Permasalahan Pokok

Dalam proses persediaan yang terjadi pada Apotek Cito Karawang terdapat beberapa permasalahan yang timbul diantaranya, yaitu:

1. Masih menggunakan Ms.Excel yang termasuk semi-terkomputerisasi dalam proses pengelolaan data persediaan obat, sehingga kurang efektif dan efisien.
2. Belum adanya sistem untuk melakukan pengecekan stok obat dan tanggal kadaluarsa pada setiap obat.
3. Belum adanya sistem yang memudahkan dalam pembuatan laporan persediaan setiap bulan.

3.2.6. Pemecahan Masalah

Untuk mengatasi masalah diatas penulis memberikan beberapa alternative pemecahan masalah pada sistem berjalan, guna mengatasi dan melakukan pembaharuan sistem. Maka penulis mengajukan pemecahan masalah berupa perancangan sistem informasi persediaan pada Apotek Cito Karawang. Adapun alternative untuk permasalahan diatas yaitu:

1. Membuat sistem yang telah terkomputerisasi sehingga memudahkan dalam proses pengelolaan data persediaan.
2. Membuat sistem yang dapat memudahkan dalam proses pengecekan stok maupun tanggal kadaluarsa obat dengan menggunakan program aplikasi berbasis Java Netbeans versi 8.2.
3. Membuat sistem yang dapat memudahkan dalam pembuatan laporan persediaan obat secara cepat dan tepat.

3.3. Analisis Kebutuhan Software

Berdasarkan prosedur sistem berjalan pada apotek maka akan dilanjutkan dengan tahapan selanjutnya yaitu berupa analisa kebutuhan. Berikut ini merupakan analisa kebutuhan software dari sistem pesediaan obat yang ada pada Apotek Cito Karawang.

3.3.1. Analisis Kebutuhan

Berdasarkan proses bisnis yang ada pada proses transaksi dan persediaan obat pada Apotek Cito, maka tahapan berikutnya adalah analisa kebutuhan.

Berikut ini spesifikasi kebutuhan (*system requirement*) usulan penulis:

1. Halaman Admin
 - A.1 Admin *Login*
 - A.2 Admin dapat mengelola Kategori
 - A.2.1 Admin dapat mengelola *Supplier*
 - A.2.2 Admin dapat mengelola Kategori Barang
 - A.2.3 Admin dapat mengelola Satuan
 - A.3 Admin dapat mengelola Barang
 - A.4 Admin dapat mengelola Pengguna
 - A.5 Admin dapat mengelola Akun
 - A.6 Admin dapat mengelola Pembelian
 - A.7 Admin dapat mengelola Penjualan
 - A.8 Admin dapat mengelola Surat Pesanan
 - A.9 Admin dapat mengelola Jurnal
 - A.10 Admin dapat mengelola Laporan Persediaan Barang

Tabel III.1.

Deskripsi Use Case Diagram Login

Use Case Name	Login
Requirement	A.1
Goal	Bagian admin dapat <i>login</i> dan masuk ke halaman menu utama
Pre-Condition	Bagian admin telah mengisi username dan password
Post-Condition	Bagian admin berhasil <i>login</i>
Failed end Condition	Bagian admin gagal <i>login</i>
Primary Actor	Apoteker
Main Flow/Basic Path	1. Admin dapat login 2. Admin dapat masuk ke halaman menu utama
Alternate Flow/invariant 1	

Tabel III.2.

Deskripsi Use Case Diagram mengelola Supplier

Use Case Name	mengelola <i>supplier</i>
Requirement	A.2.1
Goal	Admin dapat mengelola data <i>supplier</i>
Pre-Condition	Admin telah login sebagai Admin dan mengakses menu master serta memilih <i>form supplier</i>
Post-Condition	Data <i>supplier</i> berhasil tersimpan
Failed end Condition	Gagal menyimpan data <i>supplier</i>
Primary Actor	Admin
Main Flow/Basic Path	1. Admin dapat memilih tombol “Tambah” 2. Admin dapat menginput data <i>supplier</i> 3. Admin dapat mencari data <i>supplier</i> 4. Admin dapat menghapus data <i>supplier</i> 5. Sistem menampilkan data <i>supplier</i>
Alternate Flow/invariant 1	1. Admin menginput data <i>supplier</i> 2. Admin memilih tombol “Simpan” 3. Sistem menyimpan data <i>supplier</i>
	1. Sistem menampilkan data <i>supplier</i> 2. Admin memilih tombol “Ubah” 3. Sistem berhasil mengubah <i>supplier</i>

Tabel III.3.

Deskripsi Use Case Diagram mengelola Kategori Barang

Use Case Name	mengelola kategori barang
<i>Requirement</i>	A.2.2
<i>Goal</i>	Admin dapat mengelola data kategori barang
<i>Pre-Condition</i>	Admin telah login sebagai Admin dan mengakses menu master serta memilih <i>form</i> kategori barang
<i>Post-Condition</i>	Data kategori barang berhasil tersimpan
<i>Failed end Condition</i>	Gagal menyimpan data kategori barang
<i>Primary Actor</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin dapat memilih tombol “Tambah” 2. Admin dapat menginput data kategori barang 3. Admin dapat mencari data kategori barang 4. Admin dapat menghapus data kategori barang 5. Sistem menampilkan data kategori barang
<i>Alternate Flow/invariant 1</i>	1. Admin menginput data kategori barang 2. Admin memilih tombol “Simpan” 3. Sistem menyimpan data kategori barang
	1. Sistem menampilkan data kategori barang 2. Admin memilih tombol “Ubah” 3. Sistem berhasil mengubah kategori barang

Tabel III.4.

Deskripsi Use Case Diagram mengelola Satuan

Use Case Name	mengelola satuan
<i>Requirement</i>	A.2.3
<i>Goal</i>	Admin dapat mengelola data satuan
<i>Pre-Condition</i>	Admin telah login sebagai Admin dan mengakses menu master serta memilih <i>form</i> satuan
<i>Post-Condition</i>	Data kategori barang berhasil tersimpan
<i>Failed end Condition</i>	Gagal menyimpan data satuan
<i>Primary Actor</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin dapat memilih tombol “Tambah” 2. Admin dapat menginput data satuan 3. Admin dapat mencari data satuan 4. Admin dapat menghapus data satuan 5. Sistem menampilkan data satuan
<i>Alternate Flow/invariant 1</i>	1. Admin menginput data satuan 2. Admin memilih tombol “Simpan” 3. Sistem menyimpan data satuan

	1. Sistem menampilkan data satuan 2. Admin memilih tombol “Ubah” 3. Sistem berhasil mengubah satuan
--	---

Tabel III.5.

Deskripsi Use Case Diagram mengelola Barang

Use Case Name	mengelola Barang
<i>Requirement</i>	A.3
<i>Goal</i>	Admin dapat mengelola data barang
<i>Pre-Condition</i>	Admin telah login sebagai Admin dan mengakses menu master serta memilih <i>form</i> data barang
<i>Post-Condition</i>	Data barang berhasil tersimpan
<i>Failed end Condition</i>	Gagal menyimpan data barang
<i>Primary Actor</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin dapat memilih tombol “Tambah” 2. Admin dapat menginput data barang 3. Admin dapat mencari data barang 4. Admin dapat menghapus data barang 5. Sistem menampilkan data barang
<i>Alternate Flow/invariant 1</i>	1 Admin menginput data barang 2 Admin memilih tombol “Simpan” 3 Sistem menyimpan data barang
<i>Alternate Flow/invariant 2</i>	1. Sistem menampilkan data barang 2. Admin memilih tombol “Ubah” 3. Data barang berhasil diubah

Tabel III.6.

Deskripsi Use Case Diagram mengelola Pengguna

Use Case Name	mengelola pengguna
<i>Requirement</i>	A.4
<i>Goal</i>	Admin dapat mengelola data pengguna
<i>Pre-Condition</i>	Admin telah login sebagai Admin dan mengakses menu master serta memilih <i>form</i> data pengguna
<i>Post-Condition</i>	Data pengguna berhasil tersimpan
<i>Failed end Condition</i>	Gagal menyimpan data pengguna
<i>Primary Actor</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin dapat memilih tombol “Tambah” 2. Admin dapat menginput data pengguna 3. Admin dapat mencari data pengguna 4. Admin dapat menghapus data pengguna 5. Sistem menampilkan data pengguna
<i>Alternate Flow/invariant 1</i>	1 Admin menginput data pengguna 2 Admin memilih tombol “Simpan”

	3 Sistem menyimpan data pengguna
<i>Alternate Flow/invariant 2</i>	1 Sistem menampilkan data pengguna 2 Admin memilih tombol “Ubah” 3 Sistem berhasil mengubah data pengguna

Tabel III.7.

Deskripsi Use Case Diagram mengelola Akun

Use Case Name	mengelola akun
<i>Requirement</i>	A.5
<i>Goal</i>	Admin dapat mengelola data akun
<i>Pre-Condition</i>	Admin telah login sebagai Admin dan mengakses menu master serta memilih <i>form</i> data akun
<i>Post-Condition</i>	Data akun berhasil tersimpan
<i>Failed end Condition</i>	Gagal menyimpan data akun
<i>Primary Actor</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin dapat memilih tombol “Tambah” 2. Admin dapat menginput data akun 3. Admin dapat mencari data akun 4. Admin dapat menghapus data akun 5. Sistem menampilkan data akun
<i>Alternate Flow/invariant 1</i>	1. Admin menginput data akun 2. Admin memilih tombol “Simpan” 3. Sistem menyimpan data akun
	1. Sistem menampilkan data akun 2. Admin memilih tombol “Ubah” 3. Sistem berhasil mengubah data akun

Tabel III.8.

Deskripsi Use Case Diagram mengelola Penjualan

Use Case Name	Mengelola penjualan
<i>Requirement</i>	A.6
<i>Goal</i>	Admin dapat mengelola data transaksi penjualan
<i>Pre-Condition</i>	Admin telah login sebagai Admin dan mengakses menu transaksi serta memilih <i>form</i> penjualan
<i>Post-Condition</i>	Data transaksi penjualan berhasil tersimpan
<i>Failed end Condition</i>	Gagal menyimpan data transaksi penjualan
<i>Primary Actor</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin dapat memilih tombol “Tambah” 2. Admin dapat menginput data transaksi penjualan 3. Sistem menampilkan data transaksi penjualan
<i>Alternate Flow/invariant 1</i>	1. Admin menginput data transaksi penjualan 2. Admin memilih tombol “Simpan” 3. Sistem menyimpan data transaksi penjualan

	1. Sistem menampilkan data penjualan 2. Admin memilih tombol “Hapus” 3. Sistem berhasil menghapus data penjualan
--	--

Tabel III.9.

Deskripsi Use Case Diagram mengelola Pembelian

Use Case Name	Mengelola pembelian
<i>Requirement</i>	A.7
<i>Goal</i>	Admin dapat mengelola data transaksi pembelian
<i>Pre-Condition</i>	Admin telah login sebagai Admin dan mengakses menu transaksi serta memilih <i>form</i> pembelian
<i>Post-Condition</i>	Data transaksi pembelian berhasil tersimpan
<i>Failed end Condition</i>	Gagal menyimpan data transaksi pembelian
<i>Primary Actor</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin dapat memilih tombol “Tambah” 2. Admin dapat menginput data transaksi pembelian 3. Sistem menampilkan data transaksi pembelian
<i>Alternate Flow/invariant 1</i>	1. Admin menginput data transaksi pembelian 2. Admin memilih tombol “Simpan” 3. Sistem menyimpan data transaksi pembelian
	1. Sistem menampilkan data pembelian 2. Admin memilih tombol “Hapus” 3. Sistem berhasil menghapus data transaksi pembelian

Tabel III.10.

Deskripsi Use Case Diagram mengelola Surat Pesanan

Use Case Name	Mengelola surat pesanan
<i>Requirement</i>	A.8
<i>Goal</i>	Admin dapat mengelola data surat pesanan
<i>Pre-Condition</i>	Admin telah login sebagai Admin dan mengakses menu transaksi serta memilih <i>form</i> surat pesanan
<i>Post-Condition</i>	Data surat pesanan berhasil dicetak
<i>Failed end Condition</i>	Gagal mencetak surat pesanan
<i>Primary Actor</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin dapat memilih tombol “Tambah” 2. Admin dapat menginput data surat pesanan 3. Sistem menampilkan data surat pesanan
<i>Alternate Flow/invariant 1</i>	1. Admin memilih data surat pesanan 2. Admin memilih tombol “Cetak” 3. Sistem mencetak data surat pesanan

Tabel III.11.

Deskripsi Use Case Diagram mengakses Jurnal

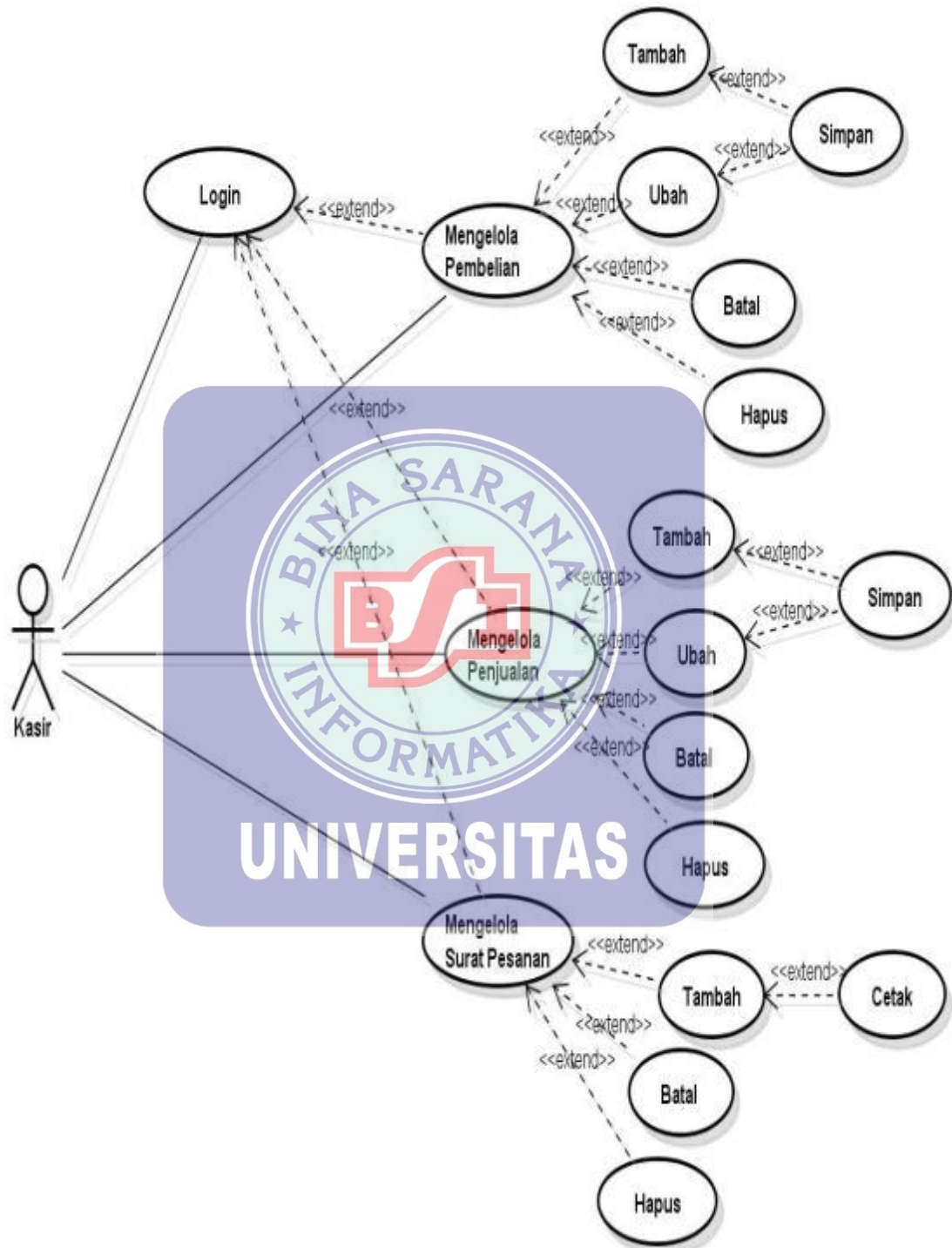
Use Case Name	Mengelola jurnal
<i>Requirement</i>	
<i>Goal</i>	Admin dapat mengakses data jurnal
<i>Pre-Condition</i>	Admin telah login sebagai admin dan mengakses menu transaksi kemudian memilih jurnal
<i>Post-Condition</i>	Admin dapat melihat data jurnal
<i>Failed end Condition</i>	Admin tidak dapat melihat data jurnal
<i>Primary Actor</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin dapat memilih periode jurnal 2. Admin dapat melihat jurnal
<i>Alternate Flow/invariant 1</i>	

Tabel III.12.

Deskripsi Use Case Diagram mengakses Laporan Persediaan Barang

Use Case Name	Mengelola laporan persediaan barang
<i>Requirement</i>	A.9
<i>Goal</i>	Admin dapat mencetak laporan persediaan barang
<i>Pre-Condition</i>	Admin telah melakukan login
<i>Post-Condition</i>	Laporan persediaan barang telah tercetak
<i>Failed end Condition</i>	Gagal mencetak laporan persediaan barang
<i>Primary Actor</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin memilih periode laporan 2. Admin mencetak laporan persediaan barang
<i>Alternate Flow/invariant 1</i>	

2. Use Case Diagram Halaman Kasir



Gambar III.4.
Use Case Diagram Halaman Kasir

Tabel III.13.

Deskripsi Use Case Diagram mengelola Pembelian

Use Case Name	Mengelola pembelian
<i>Requirement</i>	B.2
<i>Goal</i>	Kasir dapat mengelola data transaksi pembelian
<i>Pre-Condition</i>	Kasir telah login sebagai kasir dan mengakses menu transaksi serta memilih <i>form</i> pembelian
<i>Post-Condition</i>	Data transaksi pembelian berhasil tersimpan
<i>Failed end Condition</i>	Gagal menyimpan data transaksi pembelian
<i>Primary Actor</i>	Kasir
<i>Main Flow/Basic Path</i>	1. Kasir dapat memilih tombol “Tambah” 2. Kasir dapat menginput data transaksi pembelian 3. Sistem menampilkan data transaksi pembelian
<i>Alternate Flow/invariant 1</i>	1. Kasir menginput data transaksi pembelian 2. Kasir memilih tombol “Simpan” 3. Sistem menyimpan data transaksi pembelian
<i>Alternate Flow/invariant 2</i>	1. Sistem menampilkan data pembelian 2. Kasir memilih tombol “Hapus” 3. Sistem berhasil menghapus data transaksi pembelian

Tabel III.14.

Deskripsi Use Case Diagram mengelola Penjualan

Use Case Name	Mengelola penjualan
<i>Requirement</i>	B.3
<i>Goal</i>	Kasir dapat mengelola data transaksi penjualan
<i>Pre-Condition</i>	Kasir telah login sebagai kasir dan mengakses menu transaksi serta memilih <i>form</i> penjualan
<i>Post-Condition</i>	Data transaksi penjualan berhasil tersimpan
<i>Failed end Condition</i>	Gagal menyimpan data transaksi penjualan
<i>Primary Actor</i>	Kasir
<i>Main Flow/Basic Path</i>	1. Kasir dapat memilih tombol “Tambah” 2. Kasir dapat menginput data transaksi penjualan 3. Sistem menampilkan data transaksi penjualan
<i>Alternate Flow/invariant 1</i>	1. Kasir menginput data transaksi penjualan 2. Kasir memilih tombol “Simpan” 3. Sistem menyimpan data transaksi penjualan
<i>Alternate Flow/invariant 2</i>	1. Sistem menampilkan data penjualan 2. Kasir memilih tombol “Hapus” 3. Sistem berhasil menghapus data transaksi penjualan

Tabel III.15.

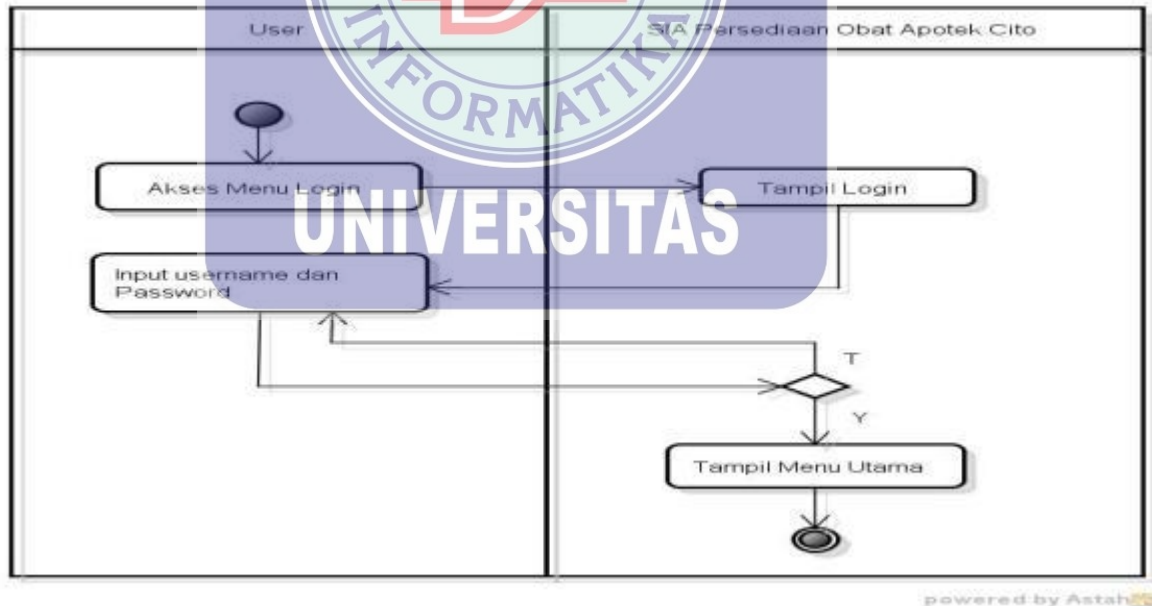
Deskripsi Use Case Diagram mengelola data Surat Pesanan

Use Case Name	Mengelola data surat pesanan
Requirement	B.4
Goal	Kasir dapat mengelola data surat pesanan
Pre-Condition	Kasir telah login sebagai kasir dan mengakses menu transaksi serta memilih <i>form</i> surat pesanan
Post-Condition	Data surat pesanan berhasil dicetak
Failed end Condition	Gagal mencetak surat pesanan
Primary Actor	Kasir
Main Flow/Basic Path	1. Kasir dapat memilih tombol “Tambah” 2. Kasir dapat menginput data surat pesanan 3. Sistem menampilkan data surat pesanan
Alternate Flow/invariant 1	1. Kasir memilih data surat pesanan 2. Kasir memilih tombol “Cetak” 3. Sistem mencetak data surat pesanan

3.3.3. Activity Diagram

1. Halaman Admin

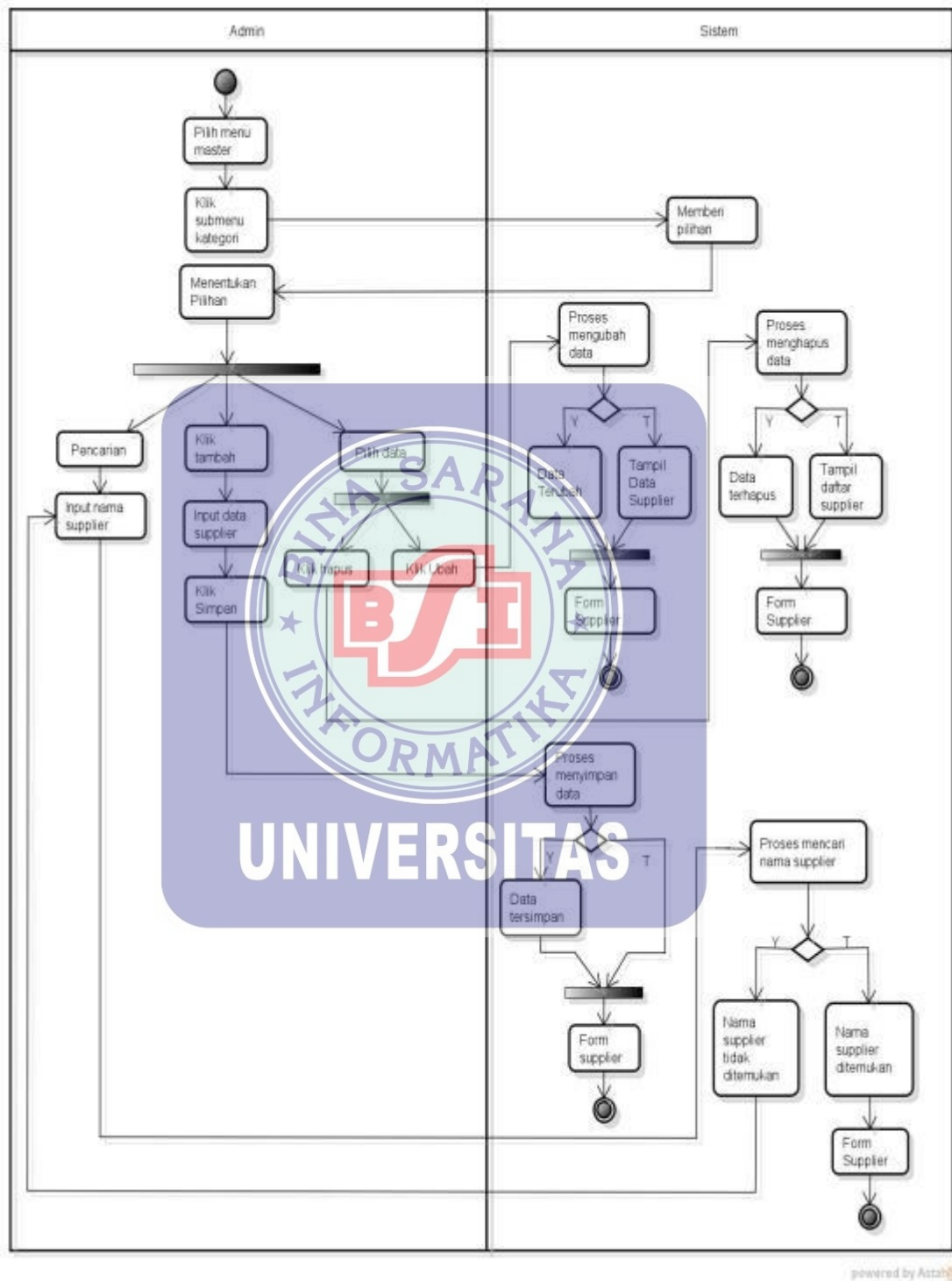
A.1 Admin Login



Gambar III.5.

Activity Diagram Admin Login

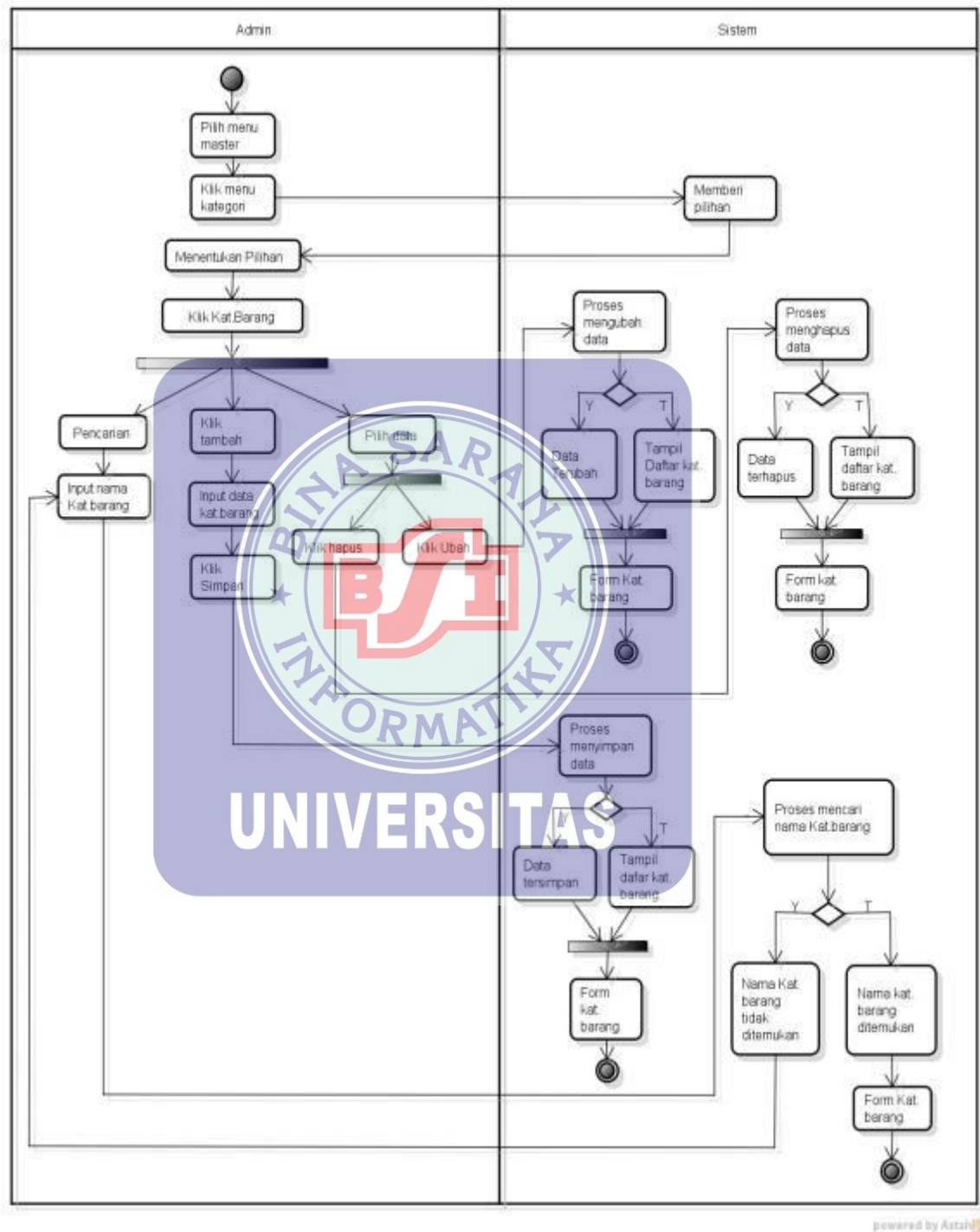
A.2.1 Admin dapat mengelola Supplier



Gambar III.6.

Activity Diagram Admin Mengelola Supplier

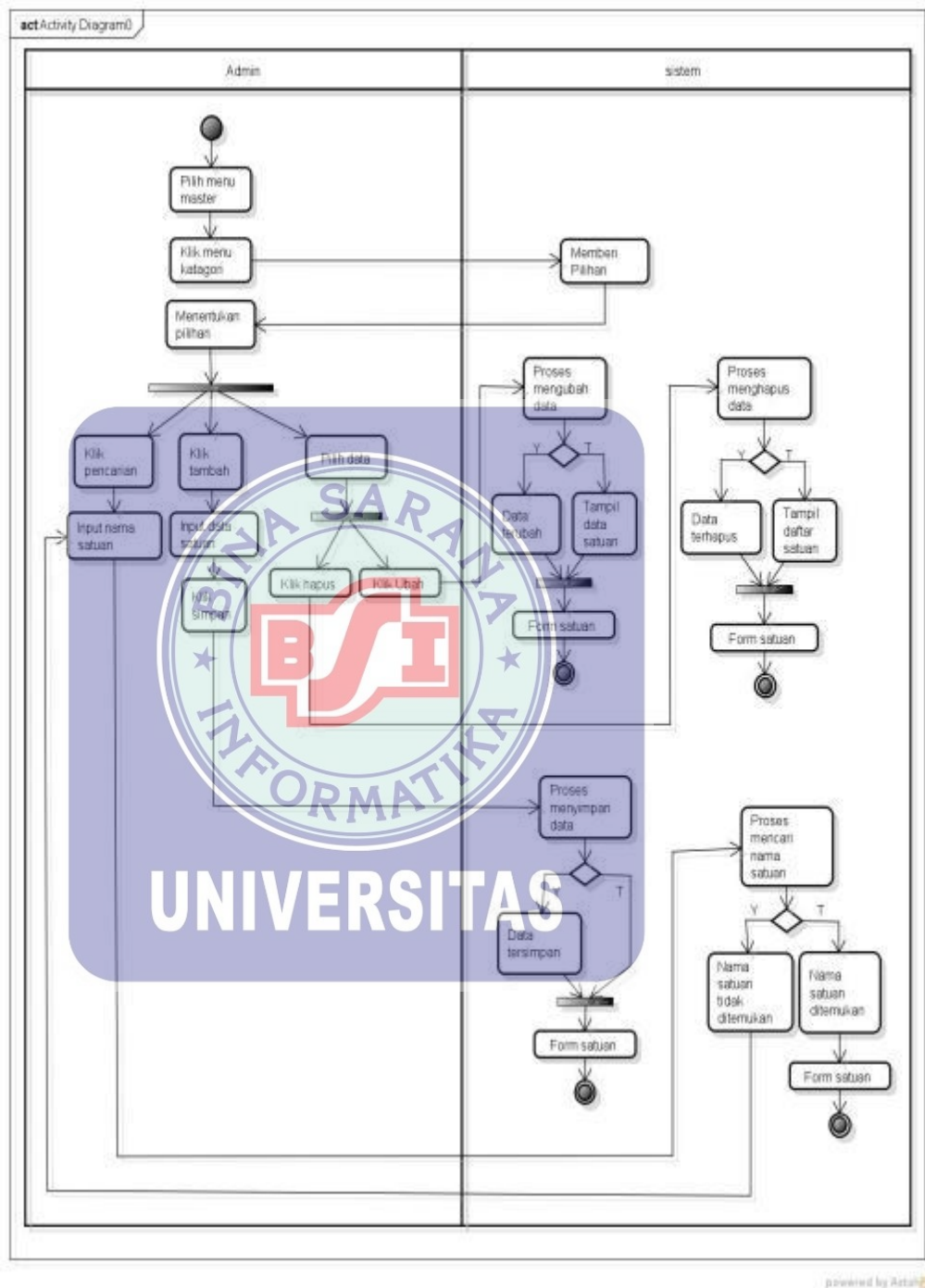
A.2.2 Admin dapat mengelola Kategori Barang



Gambar III.7.

Activity Diagram Admin Kategori Barang

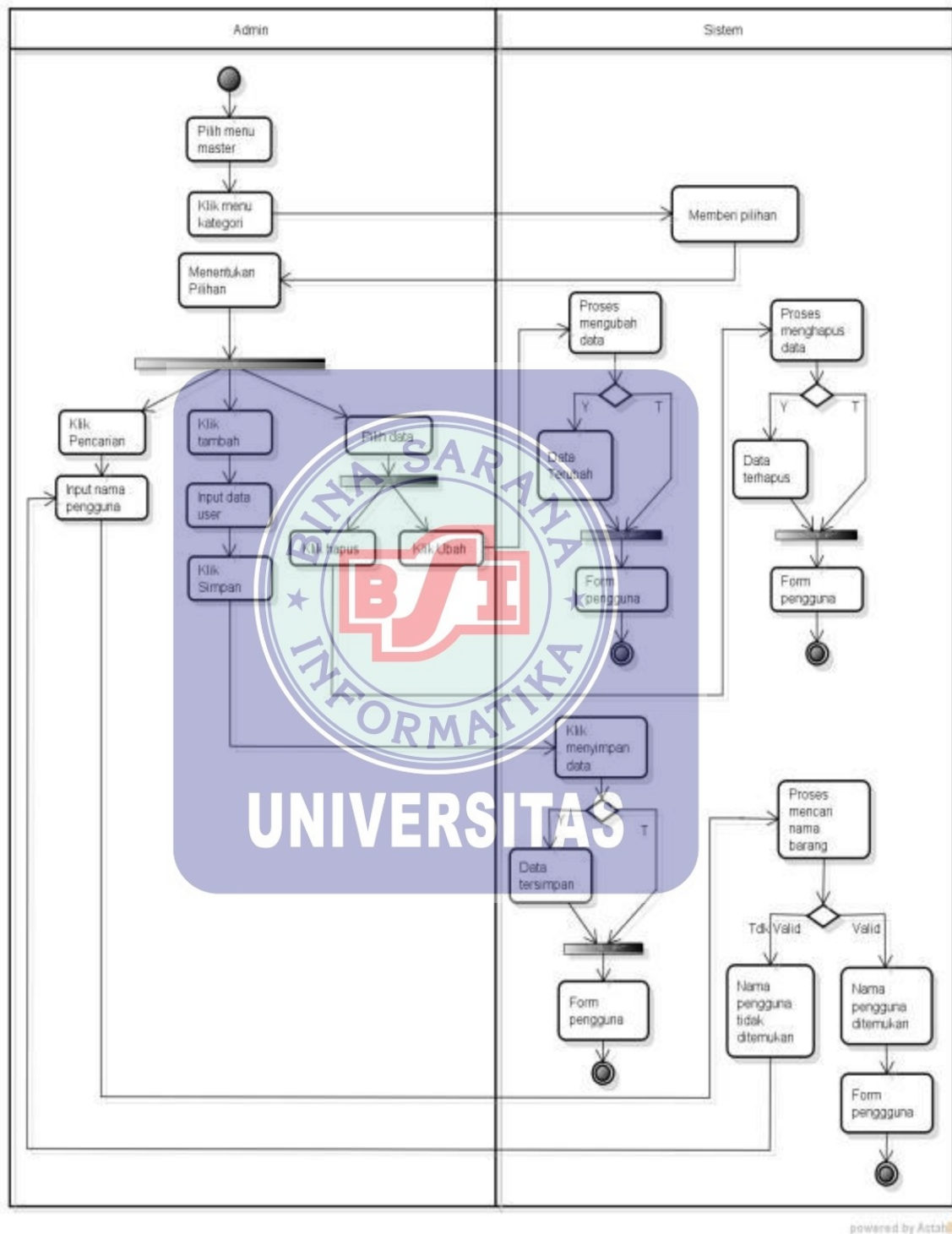
A.2.3 Admin dapat mengelola Satuan



Gambar III.8.

Activity Diagram Admin Mengelola Satuan

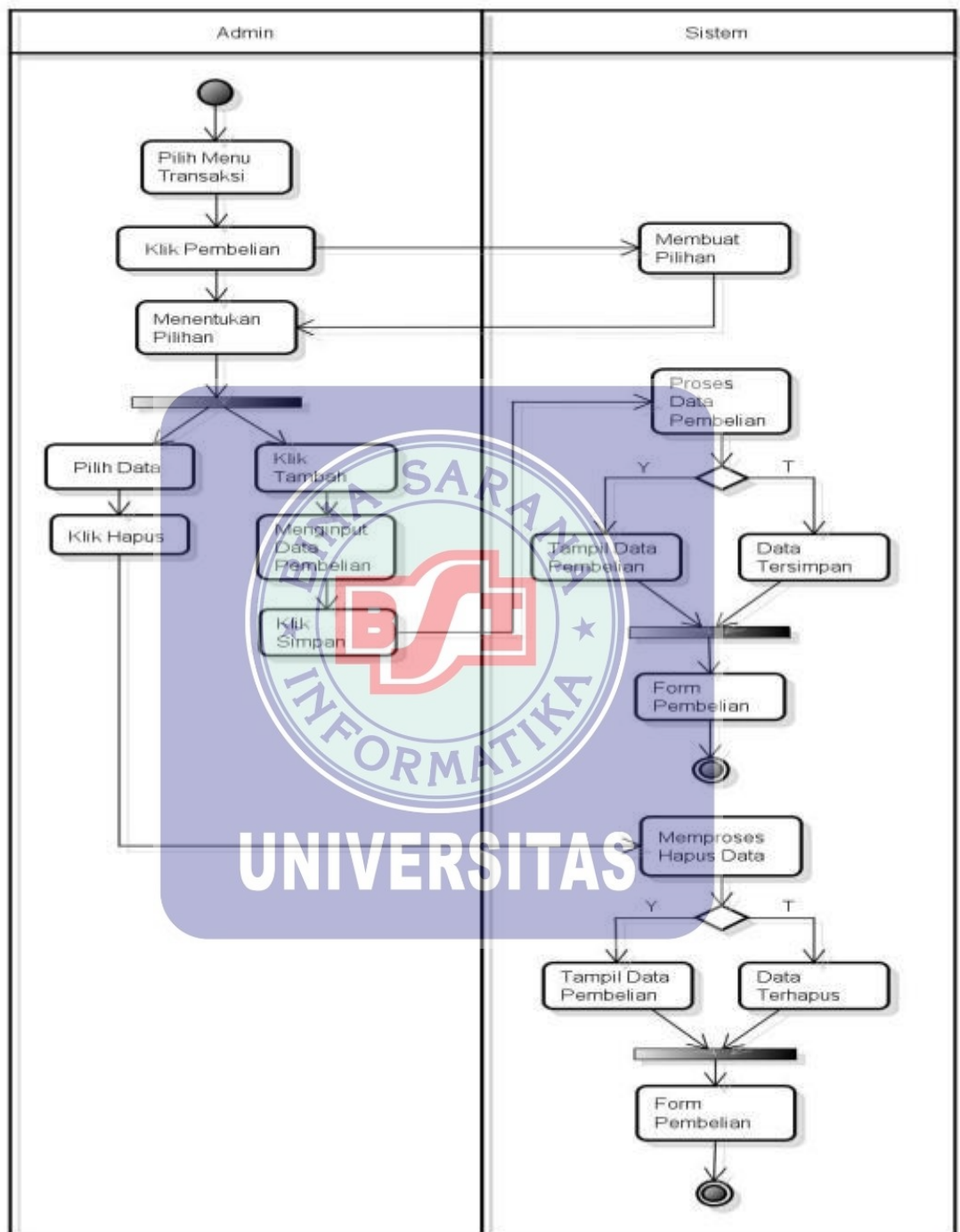
A.4 Admin dapat mengelola Pengguna



Gambar III.9.

Activity Diagram Admin Pengguna

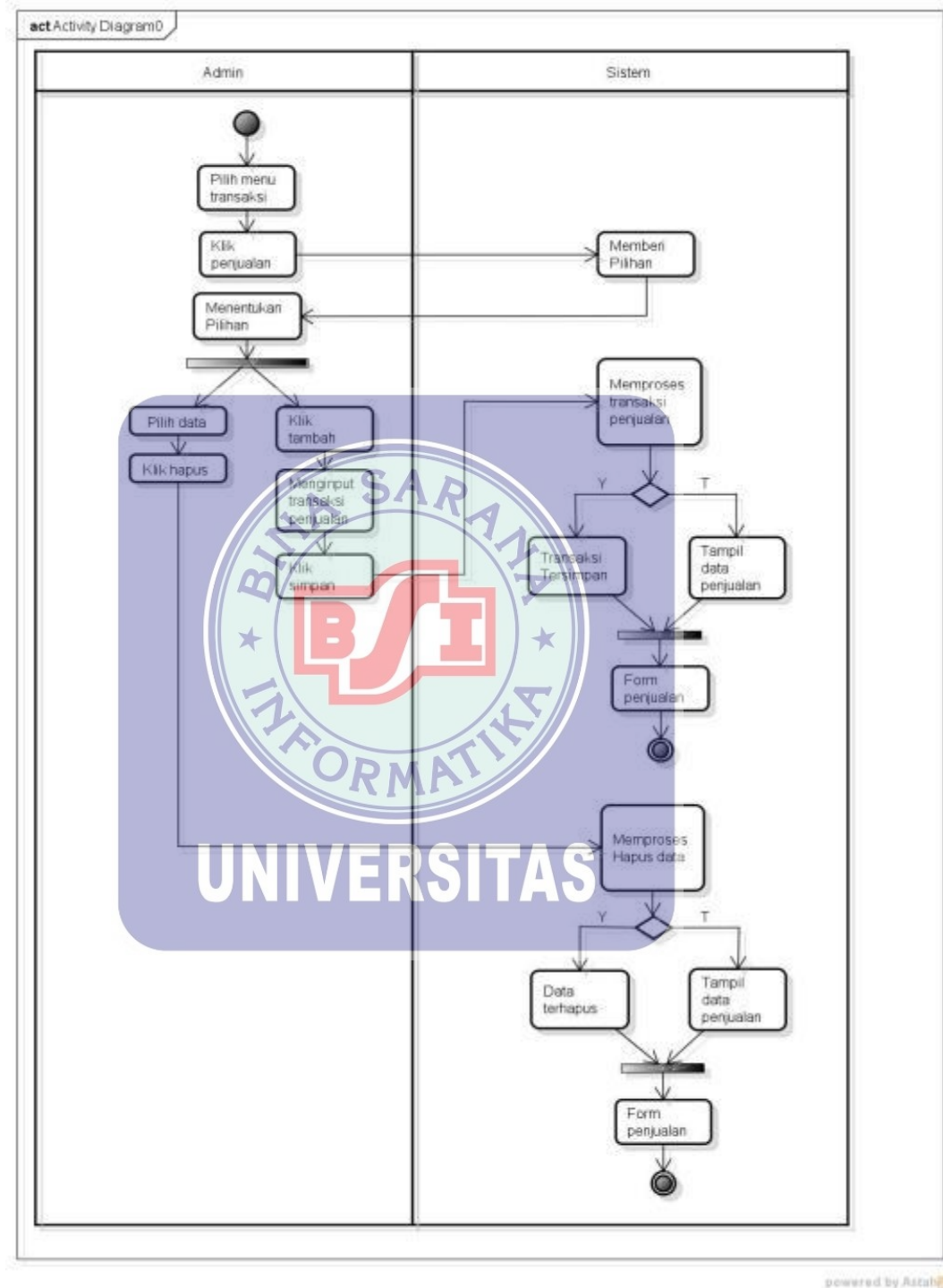
A.6 Admin dapat mengelola Pembelian



Gambar III.10.

Activity Diagram Admin Mengakses Pembelian

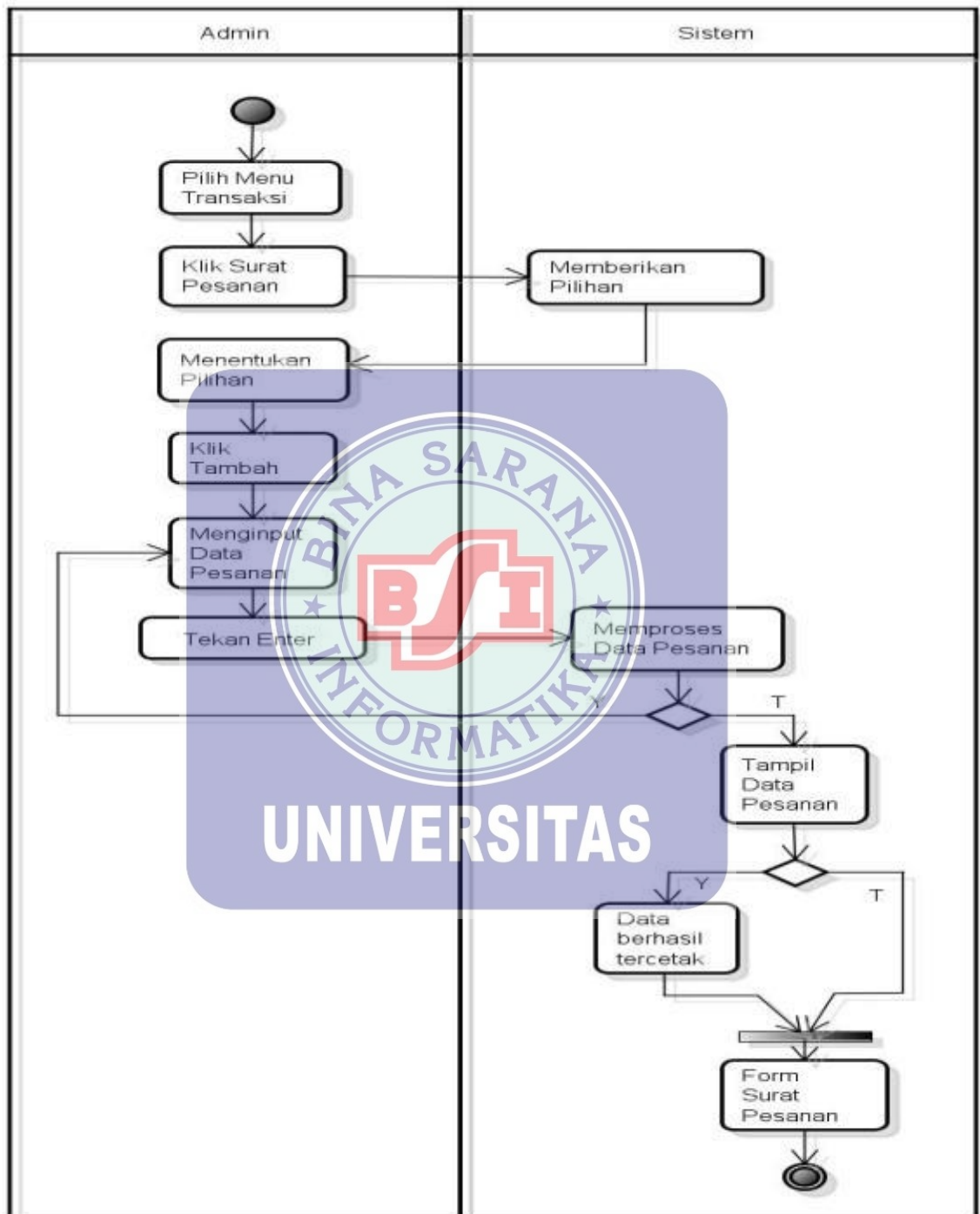
A.7 Admin dapat mengelola Penjualan



Gambar III.11.

Activity Diagram Admin Mengakses Penjualan

A.8 Admin dapat mengelola Surat Pesanan

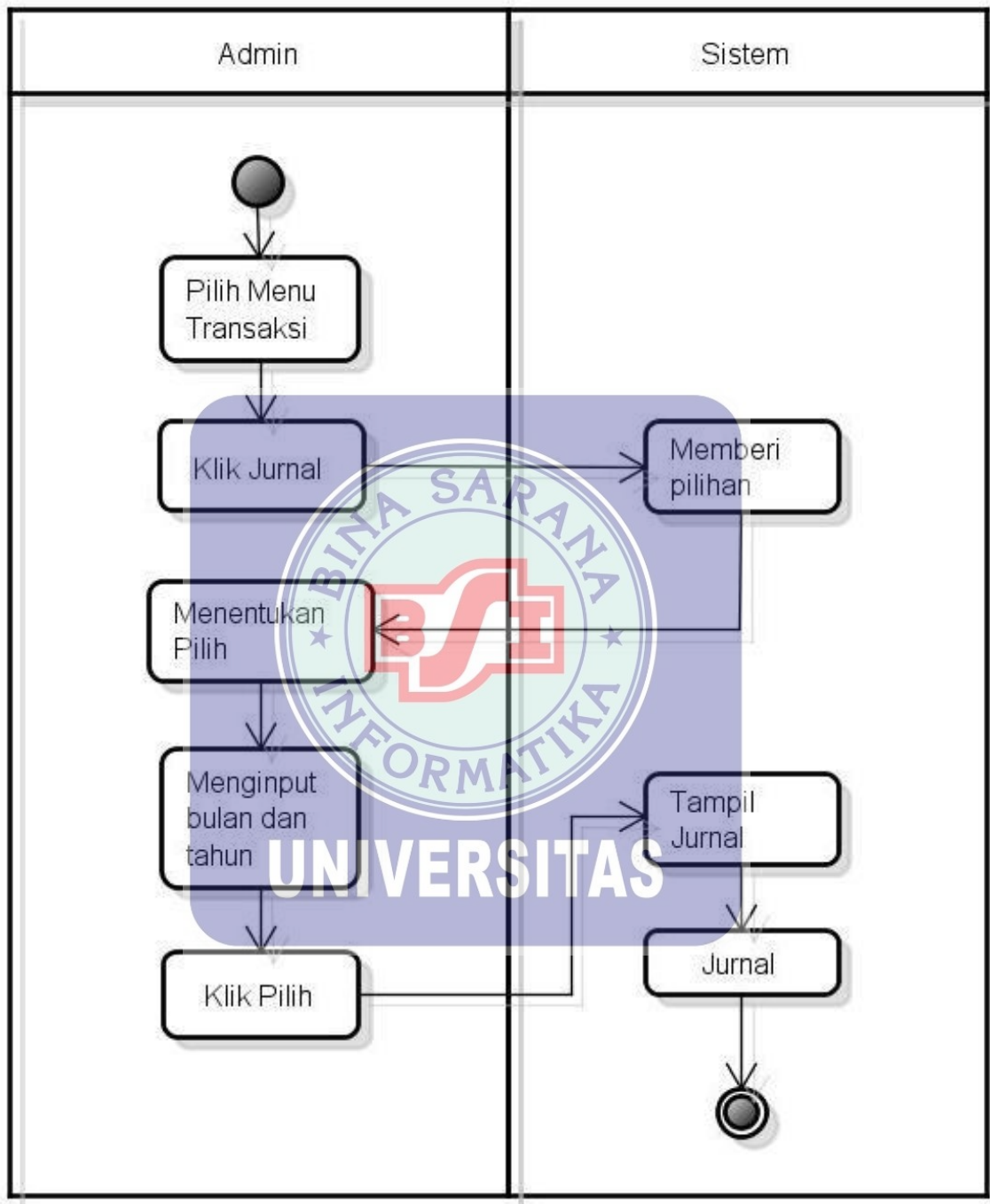


powered by Astah

Gambar III.12.

Activity Diagram Admin Mengelola Surat Pesanan

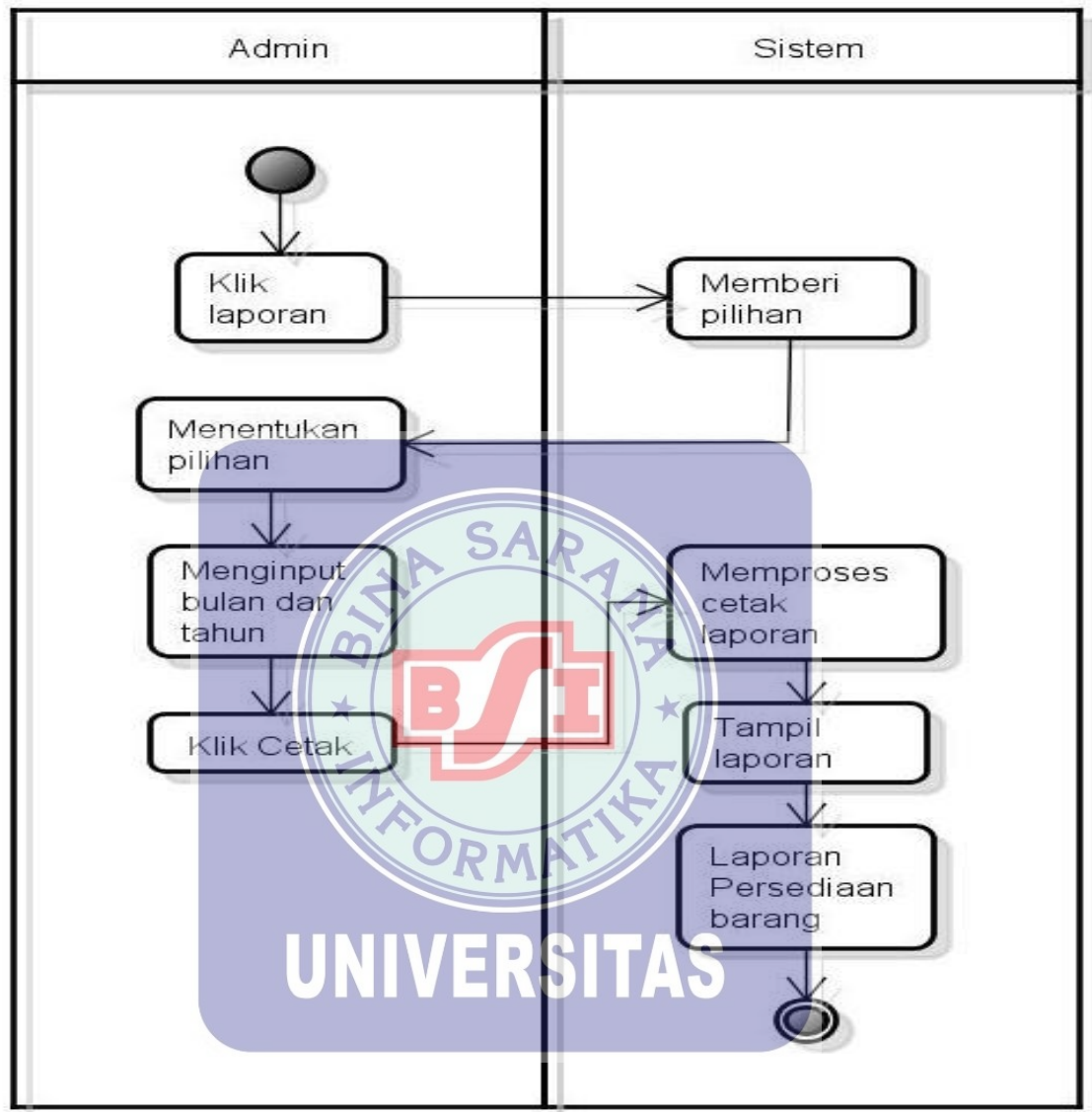
A.9 Admin dapat mengelola Jurnal



Gambar III.13.

Activity Diagram Admin Mengelola Jurnal

A.10 Admin dapat mengelola Laporan Persediaan Barang

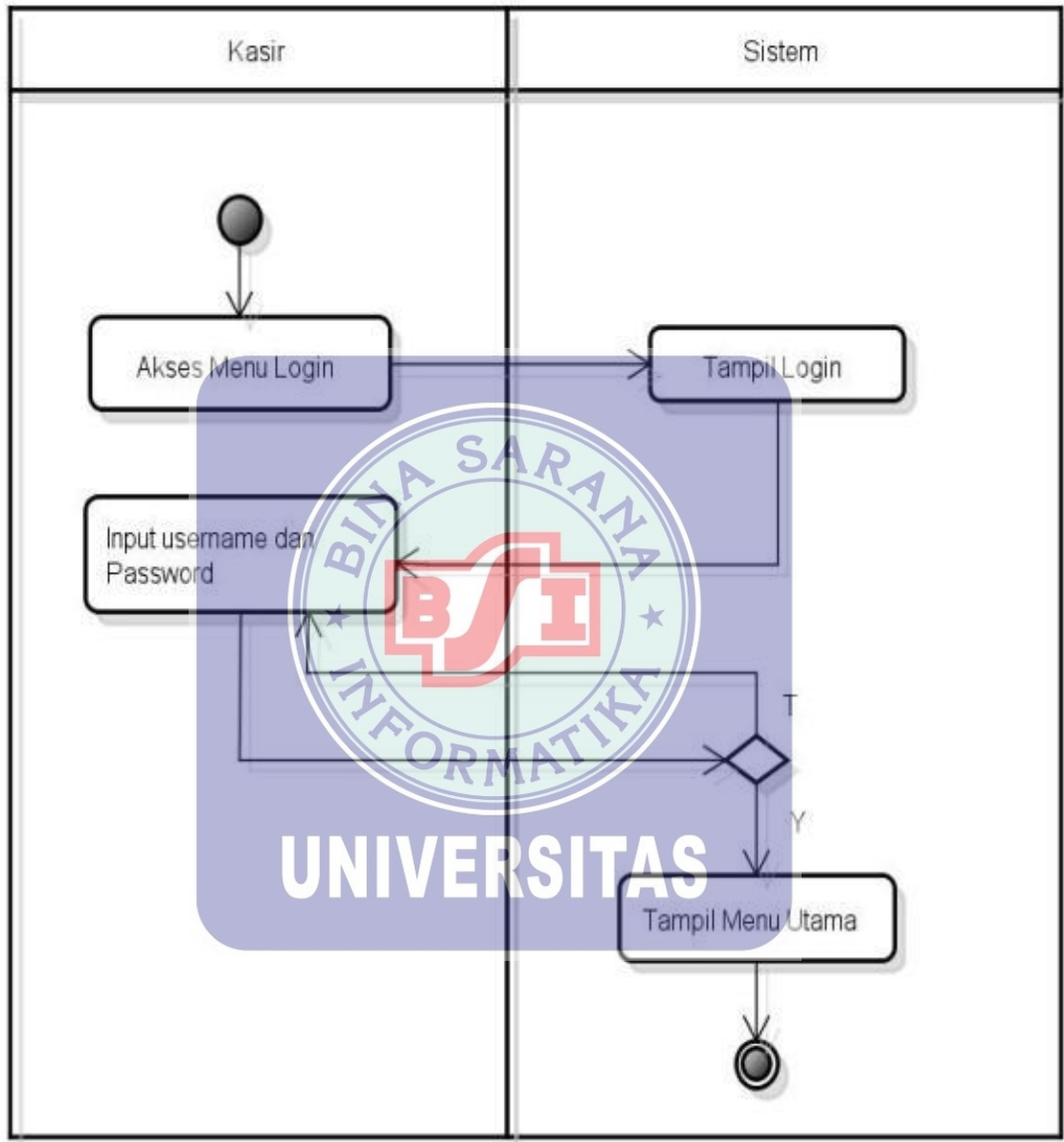


Gambar III.14.

Activity Diagram Admin Mengelola Laporan Stok Obat

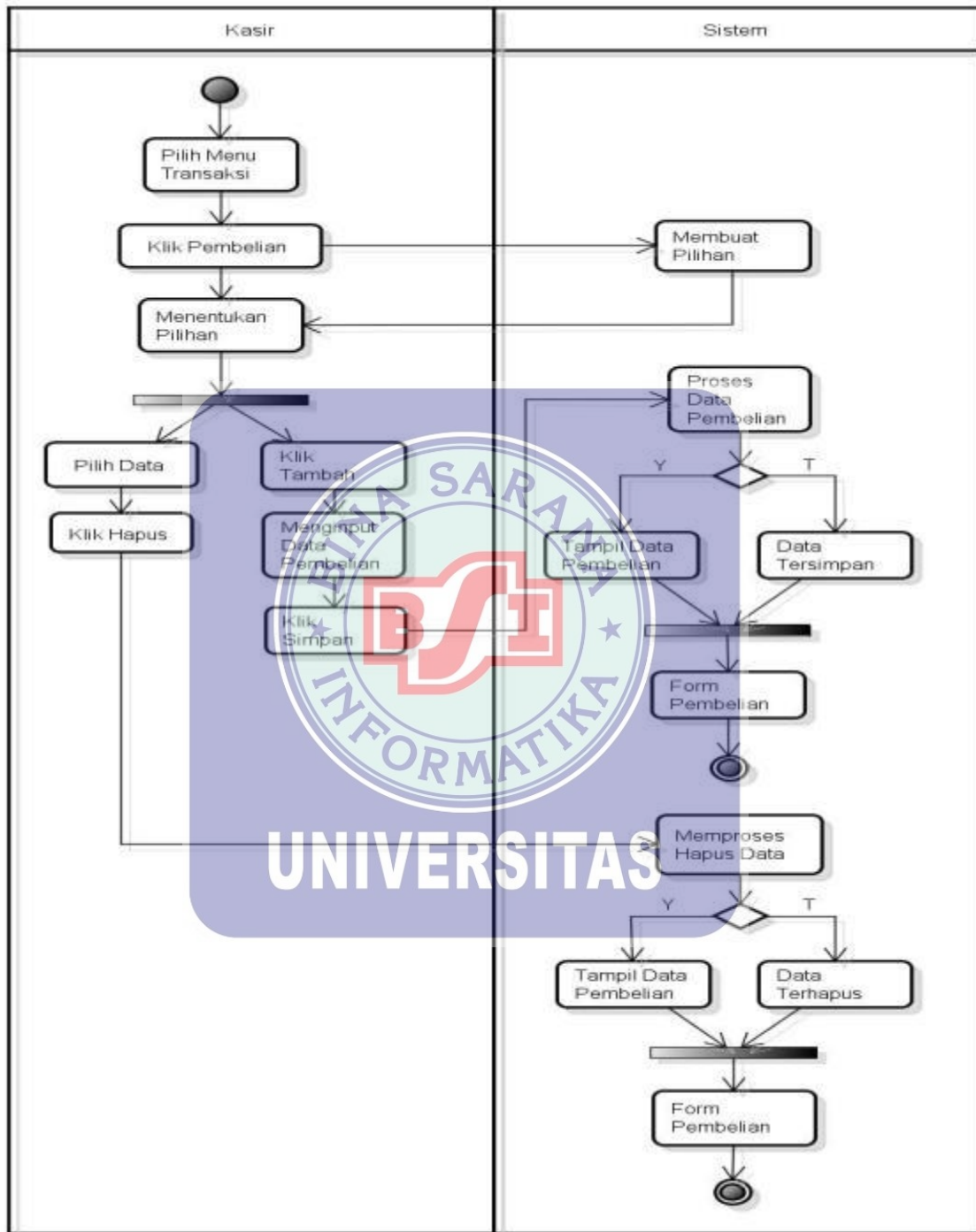
2. Halaman Kasir

B.1 Kasir *Login*



Gambar III.15.
Activity Diagram Kasir Login

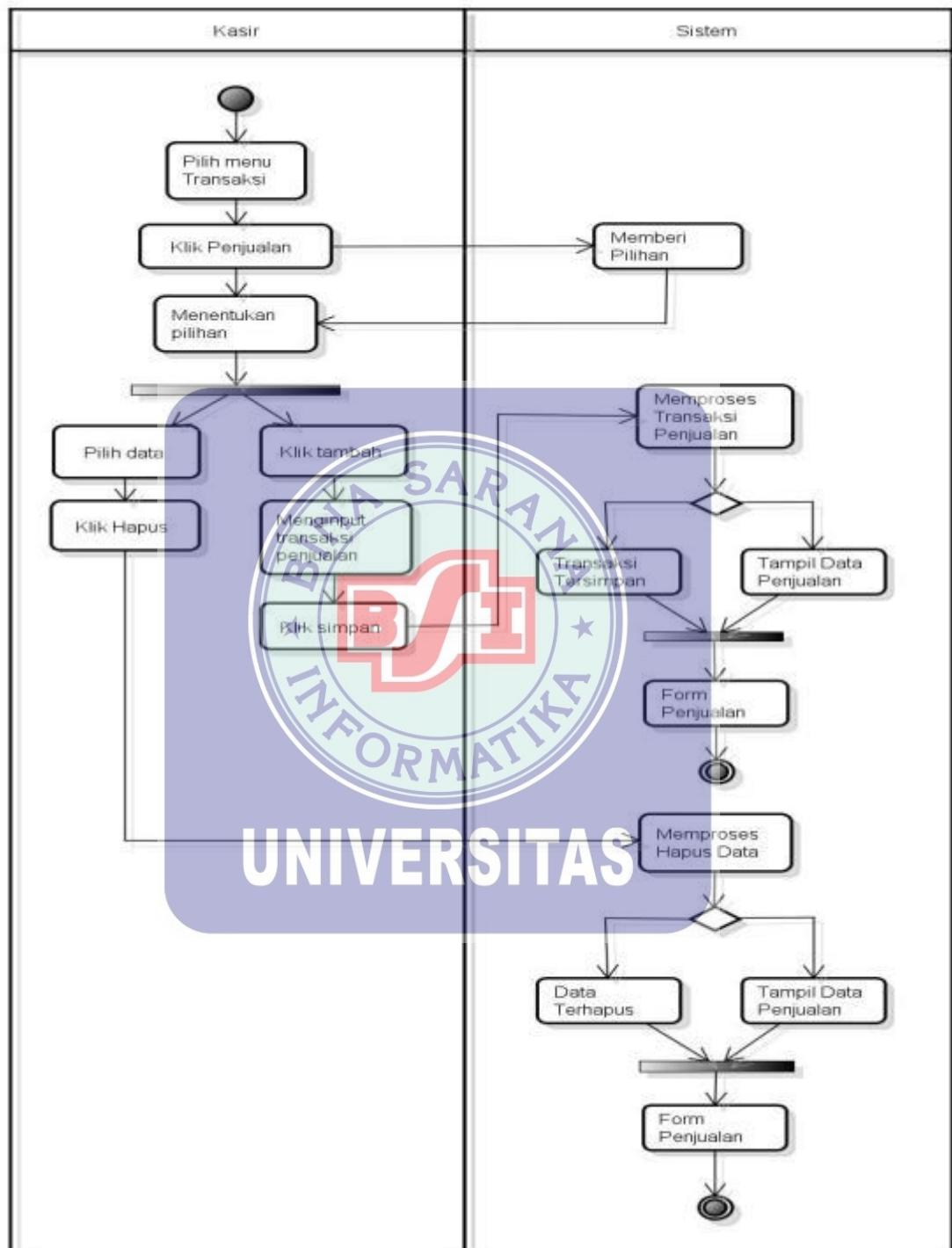
B.2 Kasir dapat mengelola Pembelian



Gambar III.16.

Activity Diagram Kasir Mengakses Pembelian

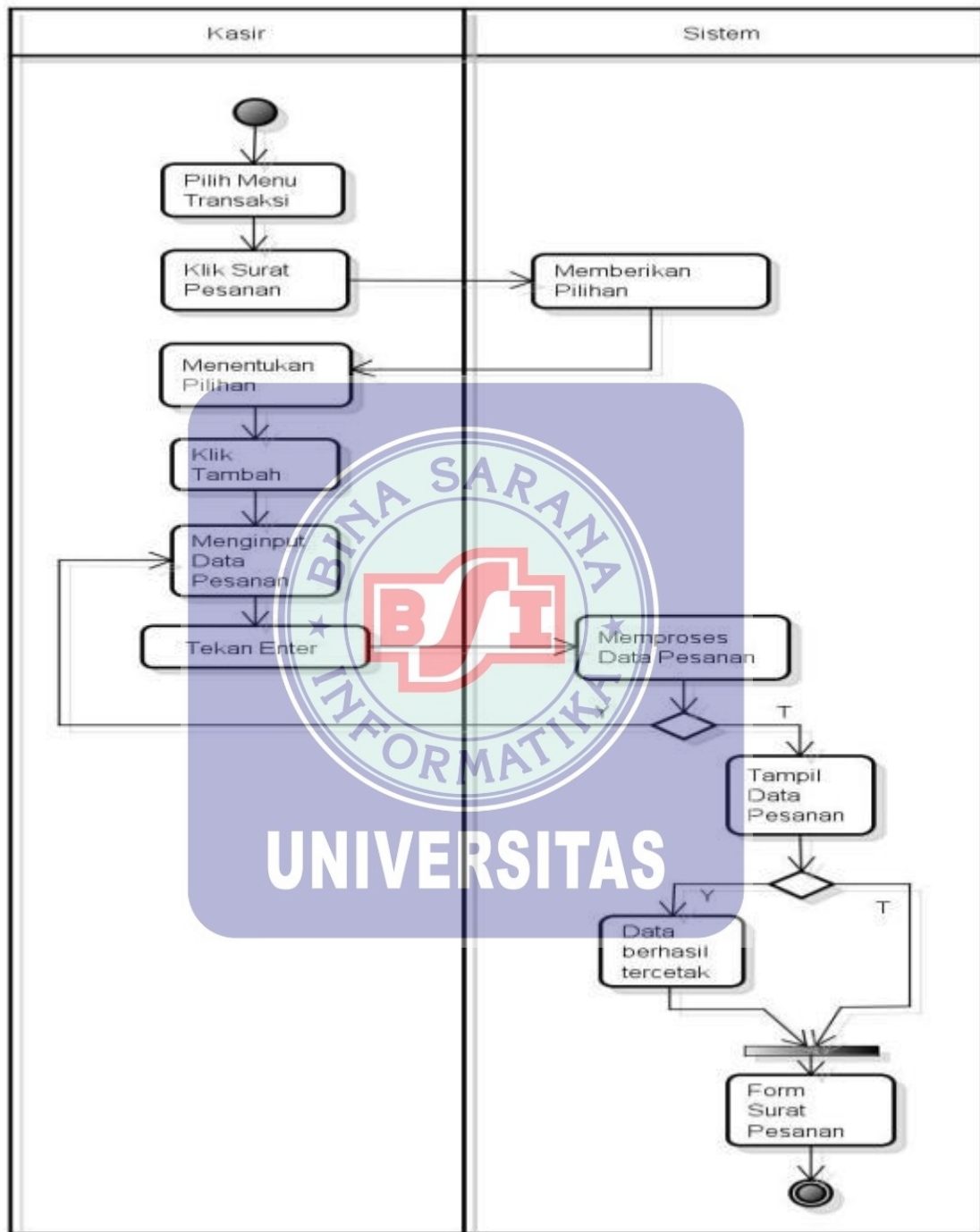
B.3 Kasir dapat mengelola Penjualan



Gambar III.17.

Activity Diagram Kasir Mengakses Penjualan

B.3 Kasir dapat mengelola Surat Pesanan

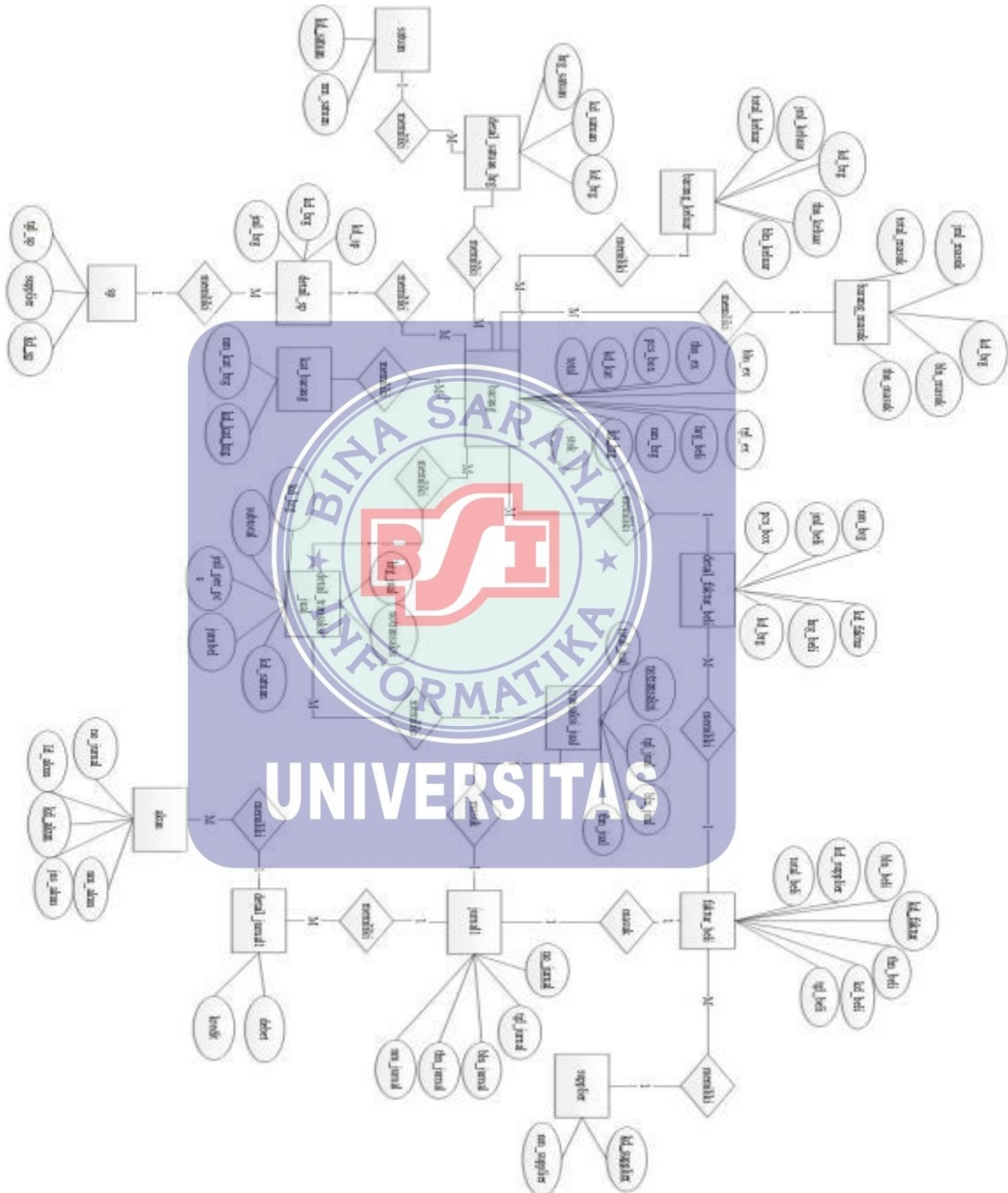


Gambar III.18.

Activity Diagram Kasir Mengakses Surat Pesanan

3.4. Desain

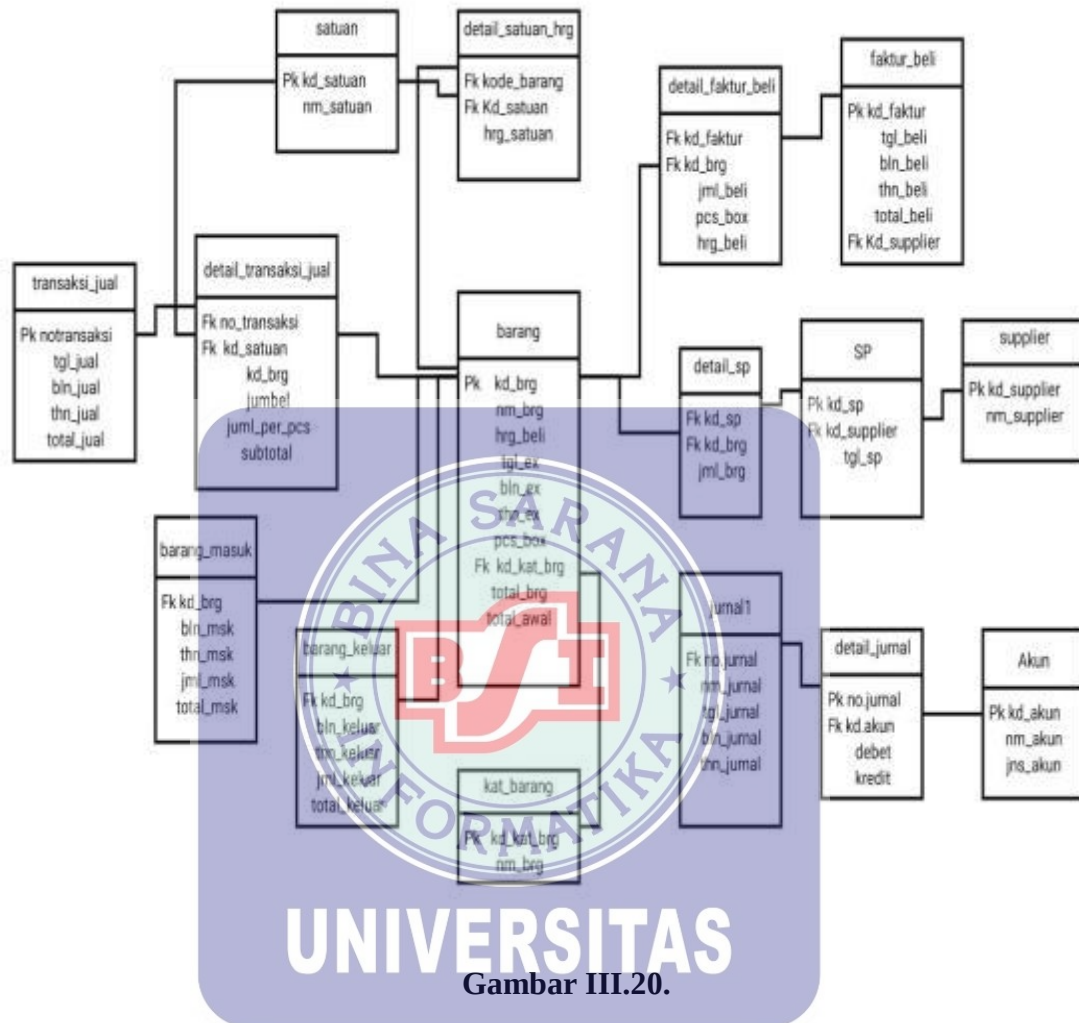
3.4.1. Entity Relationship Diagram (ERD)



Gambar III.19.

Entity Relationship Diagram (ERD)

3.4.2. Logical Relations Structure (LRS)



Gambar III.20.

Logical Relations Structure (LRS)

3.4.3. Spesifikasi File

Spesifikasi *file* merupakan rancangan tabel yang akan digunakan pada sistem usulan sesuai dengan pembahasan diatas.

1. Spesifikasi File User

Nama Database : apotek_ta.sql

Nama File : user

Tipe File : Master

Akses File : Random

Panjang Record : 100 karakter

Field Key : kd_user

Tabel.III.16.
Spesifikasi File User

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	kode user	kd_user	Varchar	25	Primary Key
2	nama user	nm_user	Varchar	25	
3	Password	Password	Varchar	25	
4	Level	Lvl	Varchar	25	

2. Spesifikasi File Barang

Nama Database : apotek_ta.sql

Nama File : barang

Tipe File : Master

Akses File : Random

Panjang Record : 200 karakter

Field Key : kd_brg

Tabel. III.17.
Spesifikasi File Barang

No	Elemen Data	Akronim	Type	Panjang	Keterangan
----	-------------	---------	------	---------	------------

1	Kode barang	kd_brg	Varchar	25	Primary Key
2	Nama barang	nm_brg	Varchar	25	
3	Harga beli	hrng_beli	Int	11	
4	Stok	Stok	Int	11	
5	Tanggal expired	tgl_ex	Varchar	25	
6	Bulan expired	bln_ex	Varchar	25	
7	Tahun expired	thn_ex	Varchar	25	
8	Pcs/box	pcs_box	Int	11	
9	Kode kategori barang	kd_kat_brg	Int	11	Foreign Key
10	Total	Total	Int	20	
11	Subtotal	total_awal	Int	11	

3. Spesifikasi File Barang Keluar

Nama Database : apotek_ta.sql

Nama File : barang keluar

Tipe File : Transaksi

Akses File : Random

Panjang Record : 106 karakter

Field Key : -

Tabel. III.18.
Spesifikasi File Barang Keluar

No	Elemen Data	Akronim	Type	Panjang	Keterangan
----	-------------	---------	------	---------	------------

1	Kode barang	kd_brg	<i>varchar</i>	25	<i>Foreign key</i>
2	Bulan keluar	bln_keluar	<i>Varchar</i>	25	
3	Tahun keluar	thn_keluar	<i>Varchar</i>	25	
4	Jumlah keluar	jml_keluar	<i>Int</i>	11	
5	Total	Total	<i>Int</i>	20	

4. Spesifikasi *File* Barang Masuk

Nama *Database* : apotek_ta.sql

Nama *File* : barang_masuk

Tipe *File* : transaksi

Akses *File* : *random*

Panjang *Record* : 106 karakter

Field Key : -

Tabel. III.19.
Spesifikasi *File* Barang Masuk

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Kode barang	kd_brg	<i>Varchar</i>	25	<i>Foreign key</i>
2	Bulan masuk	bln_masuk	<i>Varchar</i>	25	
3	Tahun masuk	thn_masuk	<i>Varchar</i>	25	
4	Jumlah masuk	Jml	<i>Int</i>	11	
5	Total	Total	<i>Int</i>	20	

5. Spesifikasi *File* Faktur Beli

Nama *Database* : apotek_ta.sql

Nama *File* : faktur_beli

Tipe *File* : *Transaksi*

Akses *File* : *Random*

Panjang *Record* : 122 karakter

Field Key : kd_faktur

Tabel.III.20.
Spesifikasi *File* Faktur Beli

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Kode faktur	kd_faktur	Varchar	25	Primary Key
2	Tanggal beli	tgl_beli	Varchar	25	
3	Bulan beli	bln_beli	Varchar	25	
4	Tahun beli	thn_beli	Varchar	25	
5	Total beli	Total	Int	11	
6	Kode supplier	kd_supplier	Int	11	Foreign Key

6. Spesifikasi *File Detail* Faktur Beli

Nama *Database* : apotek_ta.sql

Nama *File* : detail_faktur_beli

Tipe *File* : *Transaksi*

Akses *File* : *Random*

Panjang *Record* : karakter

Field Key : -

Tabel.III.21.

Spesifikasi *File Detail Faktur Beli*

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Kode faktur	kd_faktur	Varchar	25	Foreign key
2	Nama barang	nm_brg	Varchar	25	
3	Jumlah beli	jml_beli	Int	11	
4	Pcs per box	pcs_box	Int	11	
5	Harga beli	hrg_beli	Int	11	
6	Kode barang	kd_brg	Varchar	25	Foreign key

7. Spesifikasi *File Transaksi Jual*

Nama Database : apotek_ta.sql

Nama File : transaksi_jual

Tipe File : Transaksi

Akses File : Random

Panjang Record : 111 karakter

Field Key : no_transaksi

Tabel.III.22.

Spesifikasi *File Transaksi Jual*

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Nomor transaksi	no_transaksi	Varchar	25	Primary key
2	Tanggal jual	tgl_jual	Varchar	25	

3	Bulan jual	bln_jual	<i>Varchar</i>	25	
4	Tahun jual	thn_jual	<i>Varchar</i>	25	
5	Total	Total	<i>Int</i>	11	

8. Spesifikasi File Detail Transaksi Jual

Nama Database : apotek_ta.sql

Nama File : detail_transaksi_jual

Tipe File : Transaksi

Akses File : *Random*

Panjang Record : 194 karakter

Field Key : -

Tabel.III.23.

Spesifikasi File Detail Transaksi Jual

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Nomor transaksi	notransaksi	<i>Varchar</i>	25	<i>Foreign key</i>
2	Harga	Hrg	<i>Int</i>	11	
3	Nama satuan	nm_satuan	<i>Varchar</i>	25	
4	Jumlah beli	Jmlbel	<i>Int</i>	11	
5	Jumlah per pcs	jml_per_pcs	<i>Int</i>	11	
6	Subtotal	Subtotal	<i>Int</i>	11	
7	Kode barang	kd_brg	<i>Varchar</i>	25	<i>Foreign key</i>

9. Spesifikasi File Jurnal

Nama Database : apotek_ta.sql

Nama File : jurnal1

Tipe File : Transaksi

Akses File : Random

Panjang Record : karakter

Field Key : no_jurnal

Tabel.III.24.

Spesifikasi File Jurnal

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Nomor jurnal	no_jurnal	Varchar	25	Primary Key
2	Nama jurnal	nm_jurnal	Varchar	25	
3	Tanggal jurnal	tgl_jurnal	Varchar	25	
4	Bulan jurnal	bln_jurnal	Varchar	25	
5	Tahun jurnal	thn_jurnal	Varchar	25	

10. Spesifikasi File Detail Jurnal

Nama Database : apotek_ta.sql

Nama File : detail_jurnal1

Tipe File : Transaksi

Akses File : Random

Panjang Record : 72 karakter

Field Key : -

Tabel.III.25.

Spesifikasi *File Detail Jurnal*

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Nomor jurnal	no_jurnal	Varchar	25	Foreign key
2	Kode akun	kd_akun	Varchar	25	
3	Debit	Debet	Int	11	
4	Kredit	Kredit	Int	11	

11. Spesifikasi *File Supplier*

Nama Database : apotek_ta.sql

Nama File : supplier

Tipe File : Master

Akses File : Random

Panjang Record : 36 karakter

Field Key : kd_supplier

Tabel.III.26.

Spesifikasi *File Supplier*

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Kode supplier	kd_supplier	Int	11	Primary Key
2	Nama supplier	nm_supplier	Varchar	25	

12. Spesifikasi *File Surat Pesanan*

Nama Database : apotek_ta.sql

Nama File : sp

Tipe File : Transaksi

Akses File : Random

Panjang Record : 50 karakter

Field Key : kd_sp

Tabel.III.27.
Spesifikasi File Surat Pesanan

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Kode surat pesanan	kd_sp	Varchar	25	Primary Key
2	Kode supplier	kd_supplier	Varchar	25	Foreign Key
3	Tanggal	Tgl	Date		

13. Spesifikasi File Detail Surat Pesanan

Nama Database : apotek_ta.sql

Nama File : detail_sp

Tipe File : Transaksi

Akses File : Random

Panjang Record : 61 karakter

Field Key : -

Tabel.III.28.
Spesifikasi File Detail Surat Pesanan

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Kode supplier	kd_supplier	Varchar	25	Foreign key
2	Kode barang	kd_brg	Varchar	25	
3	Jumlah	Jml	Int	11	

14. Spesifikasi File Satuan

Nama Database : apotek_ta.sql

Nama File : satuan

Tipe File : Master

Akses File : Random

Panjang Record : 36 karakter

Field Key : -

Tabel.III.29.
Spesifikasi File Satuan

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Kode satuan	kd_satuan	Int	11	Primary Key
2	Nama satuan	nm_satuan	Varchar	25	

15. Spesifikasi File Detail Satuan Harga

Nama Database : apotek_ta.sql

Nama File : detail_satuan_hrg

Tipe File : Master

Akses File : Random

Panjang Record : 47 karakter

Field Key : -

Tabel.III.30.

Spesifikasi *File Detail Satuan Harga*

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Kode barang	kd_brg	<i>Varchar</i>	25	
2	Kode satuan	kd_satuan	<i>Int</i>	11	
3	Harga	Hrg	<i>Int</i>	11	

16. Spesifikasi *File Akun*

Nama *Database* : apotek_ta.sql

Nama *File* : akun1

Tipe *File* : *Master*

Akses *File* : *Random*

Panjang *Record* : karakter

Field Key : kd_akun

Tabel.III.31.

Spesifikasi *File Akun*

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Kode akun	kd_akun	<i>Varchar</i>	25	<i>Primary Key</i>
2	Nama akun	nm_akun	<i>Varchar</i>	25	
3	Jenis akun	jns_akun	<i>Varchar</i>	25	

17. Spesifikasi *File* Kategori Barang

Nama Database : apotek_ta.sql

Nama *File* : kat_barang

Tipe *File* : *Master*

Akses *File* : *Random*

Panjang *Record* : 36 karakter

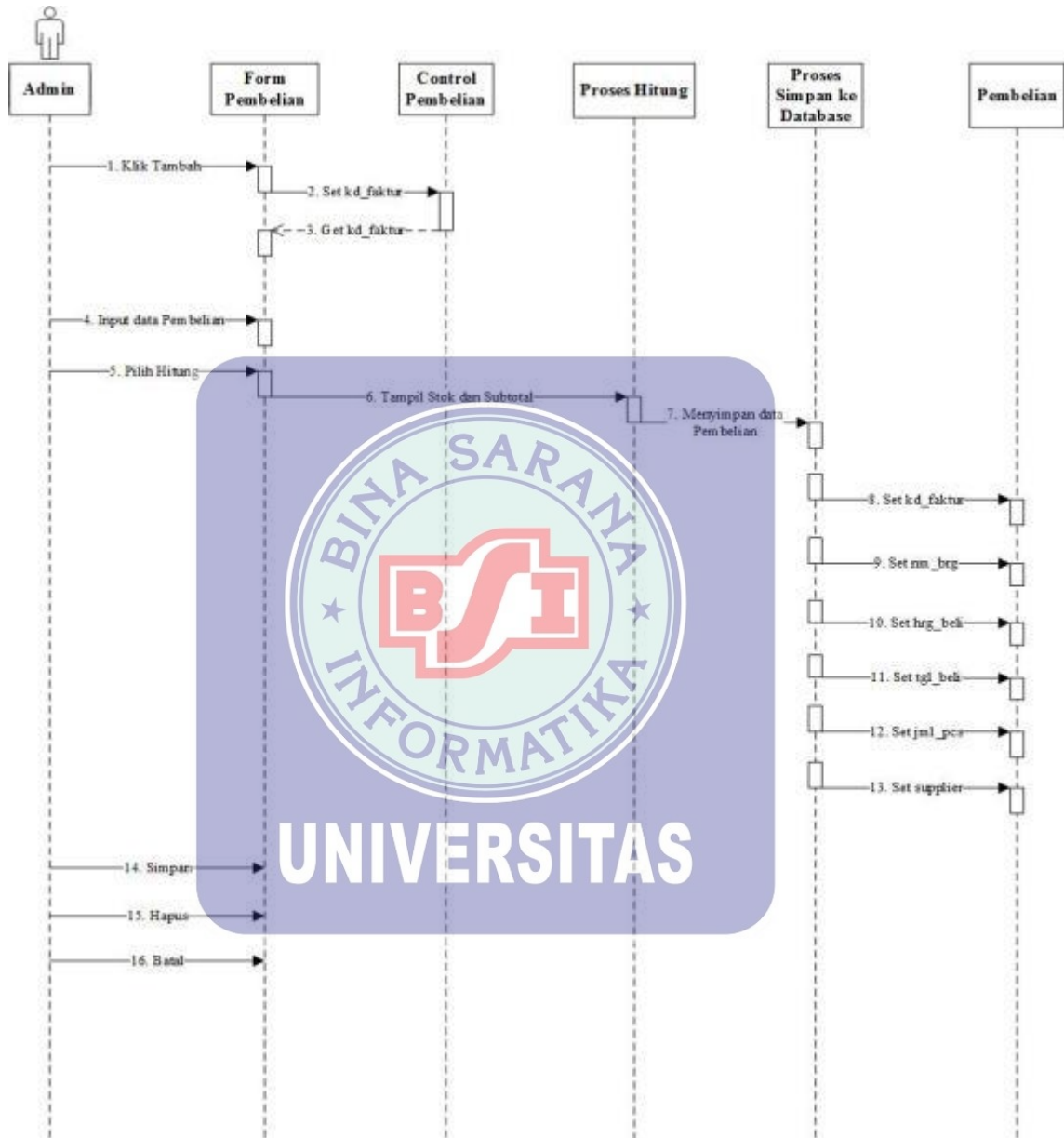
Field Key : -

Tabel.III.32.
Spesifikasi *File* Kategori Barang

No	Elemen Data	Akronim	Type	Panjang	Keterangan
1	Kode kategori barang	kd_kat_brg	Int	11	<i>Primary Key</i>
2	Nama kategori barang	nm_kat_brg	Varchar	25	

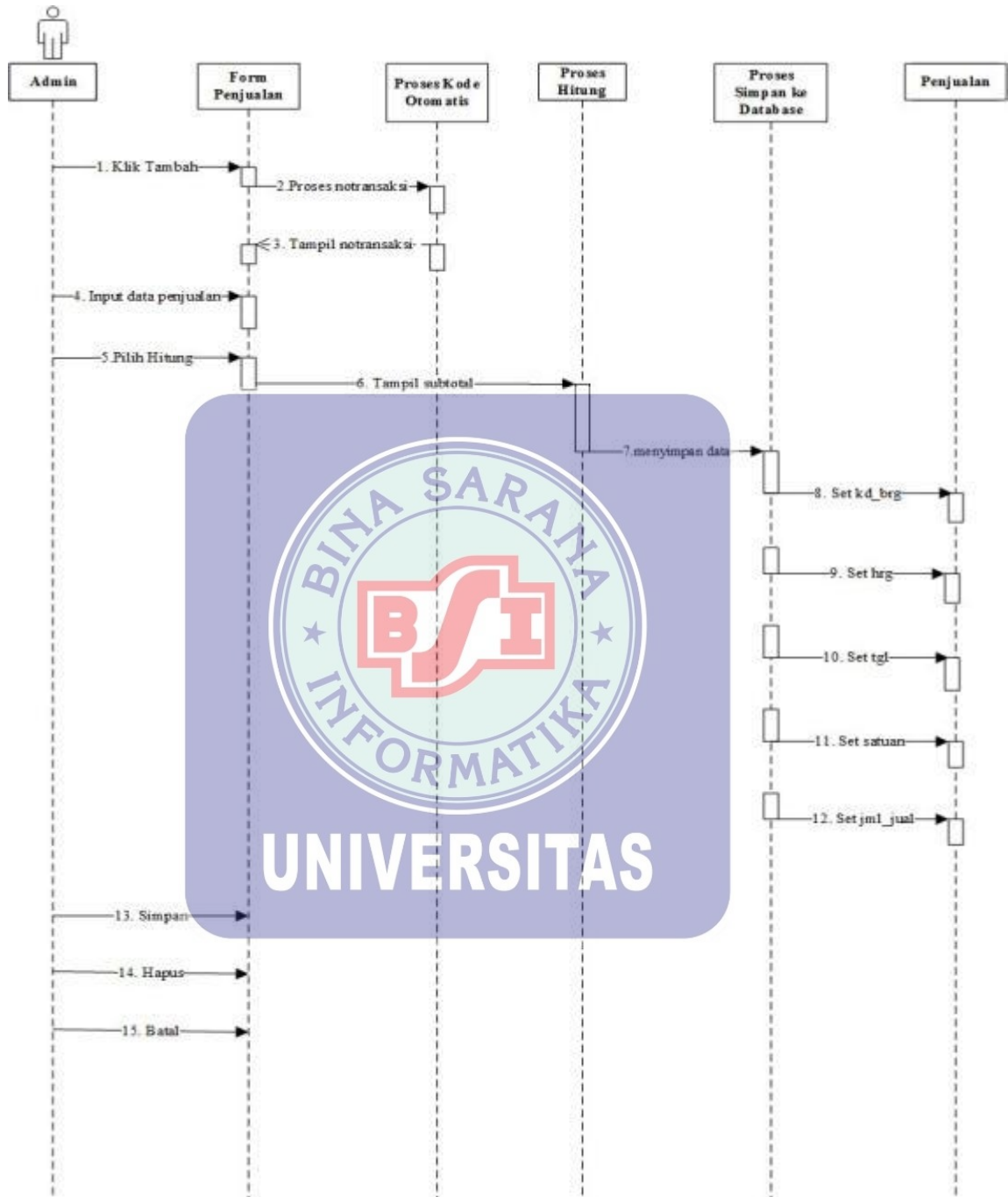
3.4.4. Sequence Diagram

1. Sequence Diagram Pembelian



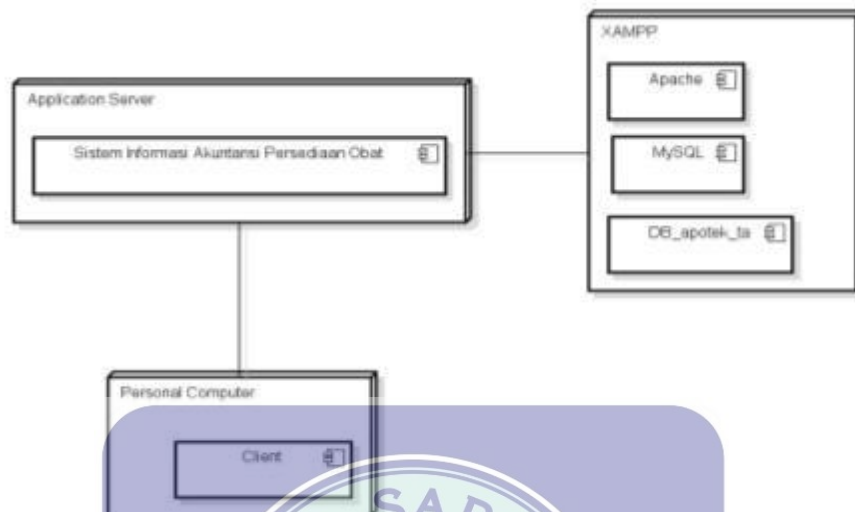
Gambar III.20.
Sequence Diagram Pembelian

2. Sequence Diagram Penjualan



Gambar III.21.
Sequence Diagram Penjualan

3.4.5. Deployment Diagram



Gambar III.22.
Deployment Diagram

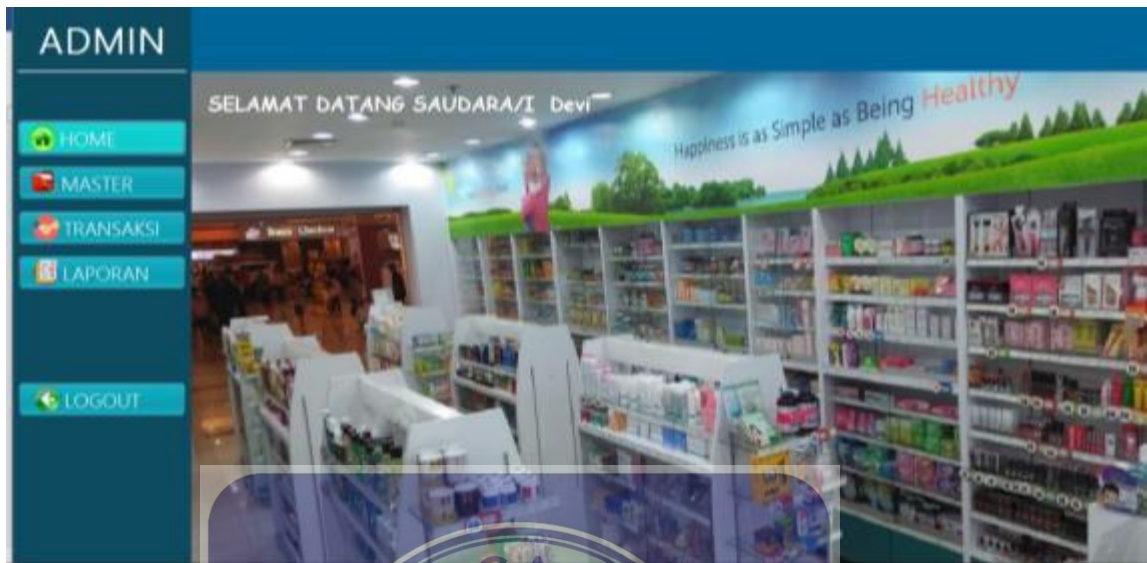
3.4.6. User Interface

1. *User Interface Form Login*



Gambar III.23.
User Interface Form Login

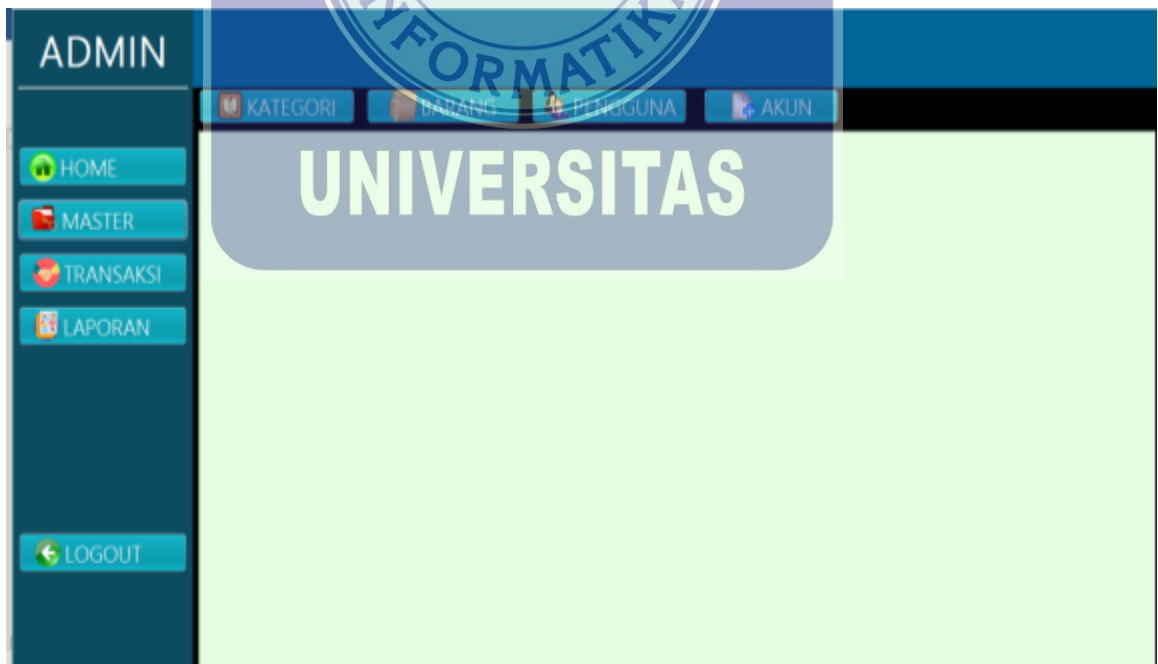
2. *User Interface Menu Home*



Gambar III.24.

User Interface Menu Home

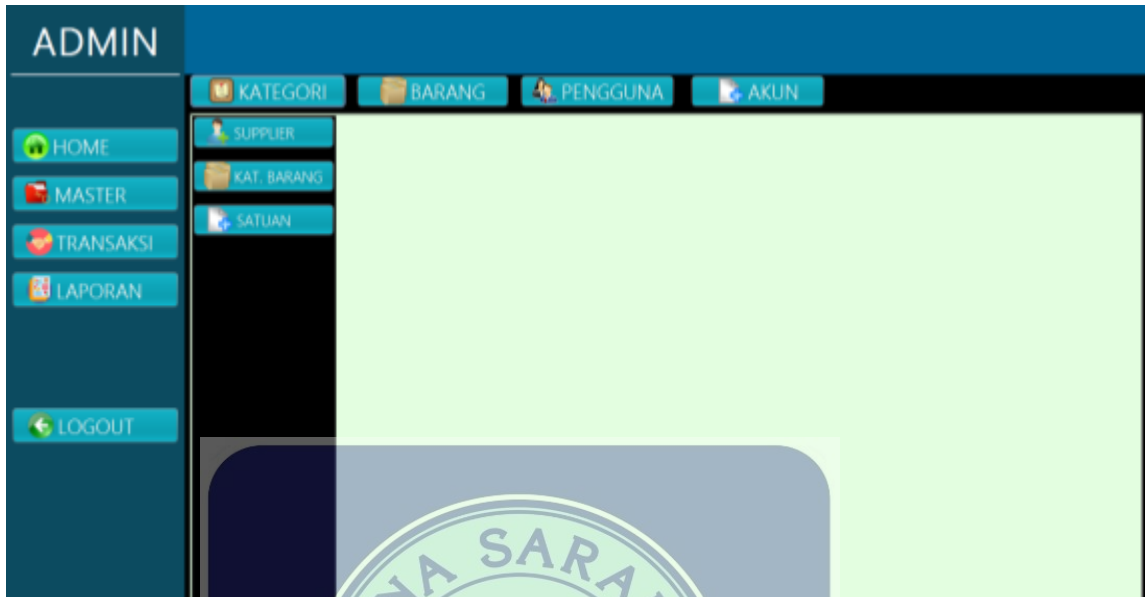
3. *User Interface Menu Master*



Gambar III.25.

User Interface Menu Master

4. *User Interface Halaman Kategori*



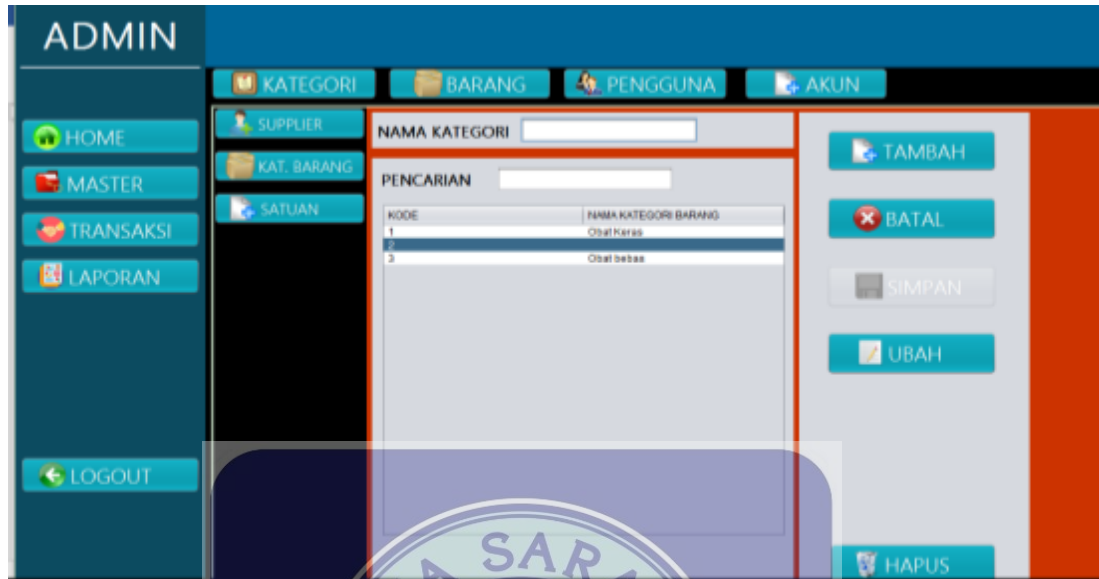
Gambar III.26.
User Interface Halaman Kategori

5. *User Interface Halaman Supplier*



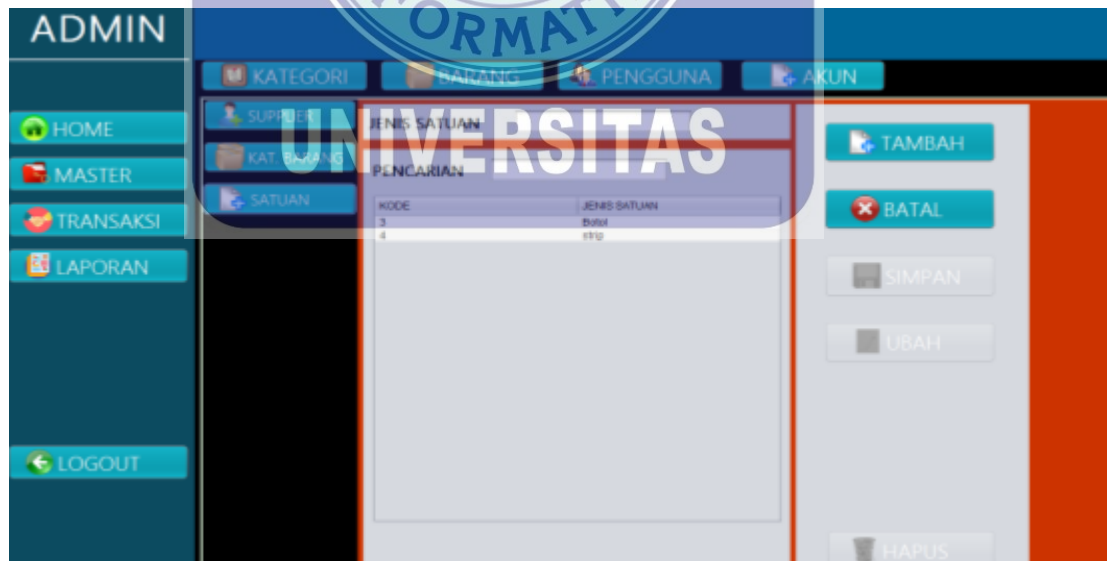
Gambar III.27.
User Interface Halaman Supplier

6. *User Interface Halaman Kategori Barang*



Gambar III.28.
User Interface Halaman Kategori Barang

1. *User Interface Halaman Satuan*



Gambar III.29.
User Interface Halaman Satuan

8. *User Interface Halaman Barang*

KODE	NAMA BARANG	STOK	JML. POS/BOX	KATEGORI	HARGA POKOK	TOL. EX	BLN. EX	THN. EX
B-20011	serabi	10	10	Cand. manis	1125	22	July	2021
B-2726172	neutrogen	55	20	Cand. manis	3165	19	July	2021

Gambar III.30.
User Interface Halaman Barang

1. *User Interface Halaman Pengguna*

KODE	NAMA PENGGUNA	PASSWORD	LEVEL
1110000	Dev	1110000	ADMIN
11101010	Mega	11101010	KASIR

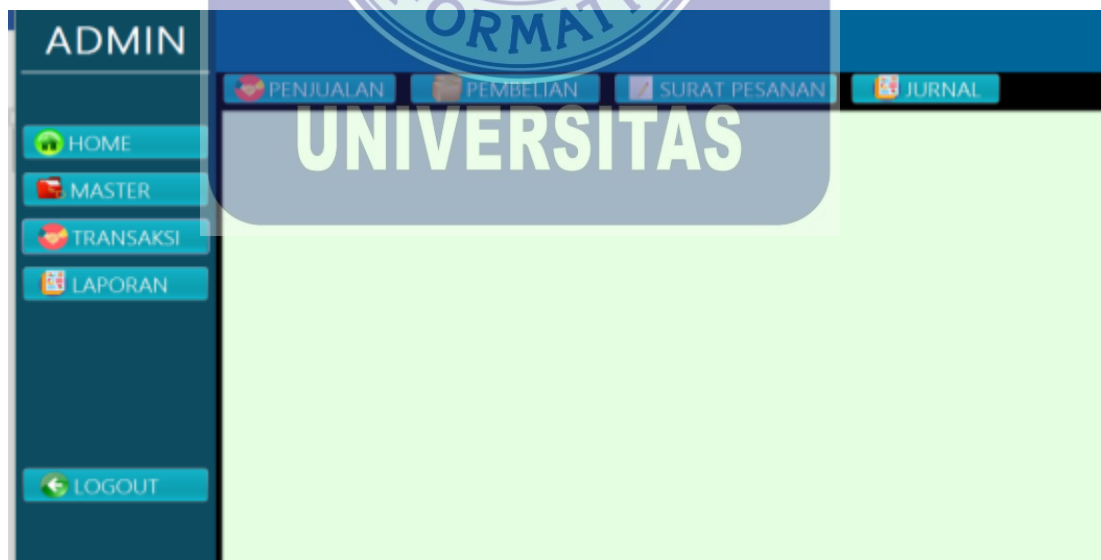
Gambar III.31.
User Interface Halaman Pengguna

11. *User Interface Halaman Akun*



Gambar III.32.
User Interface Halaman Akun

12. *User Interface Menu Transaksi*



Gambar III.33.
User Interface Menu Transaksi

13. *User Interface Submenu Penjualan*

ADMIN

[PENJUALAN](#)
[PEMBELIAN](#)
[SURAT PESANAN](#)
[JURNAL](#)

NO. TRANSAKSI:
 TANGGAL: 18
 BULAN: August
 TAHUN: 2019

KODE BARANG:
 SATUAN:

NAMA BARANG:
 HARGA Rp.

STOK:
 JUMLAH BELI: ENTER

EXPIRED:
 SUBTOTAL:

KATEGORI:
 UANG BAYAR: ENTER

UANG KEMBALI:

TOTAL BAYAR

KODE	NAMA BARANG	JENIS SATUAN	HARGA	JUMLAH BELI	SUBTOTAL

[TAMBAH](#)
[BATAL](#)
[SIMPAN](#)
[HAPUS](#)

Gambar III.34.
User Interface Submenu Penjualan

14. *User Interface Submenu Pembelian*

ADMIN

[PENJUALAN](#)
[PEMBELIAN](#)
[SURAT PESANAN](#)
[JURNAL](#)

NO. FAKTUR:
 TANGGAL: 18
 BULAN: August
 TAHUN: 2019

NAMA BARANG:
 SUPPLIER:

HARGA BELI Rp.
 DISKON: %

STOK (PCS):
 JUMLAH BOX:

SUBTOTAL Rp.
 JUMLAH PCS (1 BOX) :

HITUNG

TOTAL

SUPPLIER	NAMA BARANG	STOK BARANG (PCS / BOTOL)	SUBTOTAL

[TAMBAH](#)
[BATAL](#)
[SIMPAN](#)
[HAPUS](#)

Gambar III.35.
User Interface Submenu Pembelian

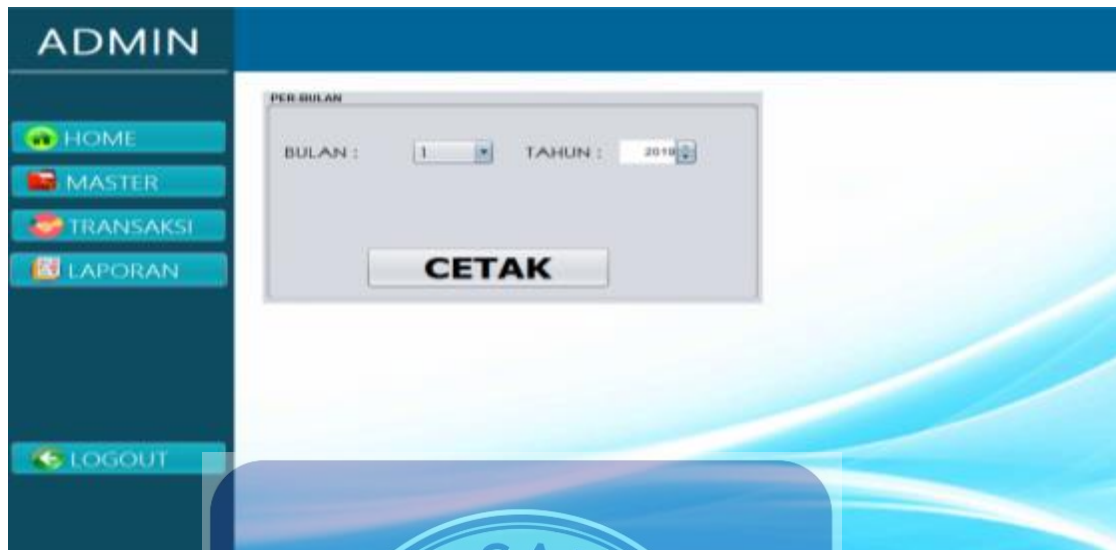
15. *User Interface Submenu Surat Pesanan*

Gambar III.36.
User Interface Submenu Surat Pesanan

16. *User Interface Submenu Jurnal*

Gambar III.37.
User Interface Submenu Jurnal

17. *User Interface Menu Laporan*



Gambar III.38.
User Interface Menu Laporan

3.5. Implementasi

3.5.1. *Code Generation*

1. Code Laporan

```
if(cmmbulan1.getSelectedItem().equals("1"))
{
    ambil_blnbeli="Januari";
    ambil_blnbeli1="January";
}else if(cmmbulan1.getSelectedItem().equals("2"))
{
    ambil_blnbeli="Februari";
    ambil_blnbeli1="February";
}else if(cmmbulan1.getSelectedItem().equals("3"))
```

```

{
    ambil_blnbeli="Maret";

    ambil_blnbeli1="March";
}else if(cmbbulan1.getSelectedItem().equals("4"))
{
    ambil_blnbeli="April";

    ambil_blnbeli1="April";
}else if(cmbbulan1.getSelectedItem().equals("5"))
{
    ambil_blnbeli="Mei";

    ambil_blnbeli1="May";
}else if(cmbbulan1.getSelectedItem().equals("6"))
{
    ambil_blnbeli="Juni";

    ambil_blnbeli1="June";
}else if(cmbbulan1.getSelectedItem().equals("7"))
{
    ambil_blnbeli="Juli";

    ambil_blnbeli1="July";
}else if(cmbbulan1.getSelectedItem().equals("8"))
{
    ambil_blnbeli="Agustus";

    ambil_blnbeli1="August";
}

```

```

}else if(cmmbulan1.getSelectedItem().equals("9"))
{
    ambil_blnbeli="September";
    ambil_blnbeli1="September";
}else if(cmmbulan1.getSelectedItem().equals("10"))
{
    ambil_blnbeli="Oktober";
    ambil_blnbeli1="October";
}else if(cmmbulan1.getSelectedItem().equals("11"))
{
    ambil_blnbeli="November";
    ambil_blnbeli1="November";
}else
{
    ambil_blnbeli="Desember";
    ambil_blnbeli1="December";
}

int a = yctahun1.getYear();

String b = Integer.toString(a);

try {
    File file = new File("src/laporan/persediaan_bulanan.jrxml");
    jasperDesign = JRXmlLoader.load(file);

```

```

HashMap param = new HashMap();

param.put("bln", ambil_blnbeli1);

param.put("bln1", ambil_blnbeli1);

param.put("thn", b);

param.put("thn1", b);

jasperReport = JasperCompileManager.compileReport(jasperDesign);

jasperPrint = JasperFillManager.fillReport(jasperReport, param,

kon.setKoneksi());

JasperViewer viewer = new JasperViewer(jasperPrint, false);

viewer.setFitPageZoomRatio();

viewer.setVisible(true);

viewer.setExtendedState(MAXIMIZED_BOTH);

} catch (JRException e) {

JOptionPane.showMessageDialog(null, e.getMessage());

}

```

2. Code embelian

```

//untuk bagian pembelian

private String nama_barang_p;

String kode_supplier;

String nama_supplier;

public String getkode_supplier(){

return kode_supplier;
}

```

```

}

public String getnama_supplier(){

    return nama_supplier;

}

public void nonaktif_p(){

    no_fak_p.setEditable(false);

    kd_mB.setEditable(false);

    hrg_beli_p.setEditable(false);

    stok_p.setEditable(false);

    sbttl_p.setEditable(false);

    nm_sup_p.setEditable(false);

    diskon_p.setEditable(false);

    jml_box_p.setEditable(false);

    jml_brg_p.setEditable(false);

    ttl_p.setEditable(false);

    ubay_p.setEditable(false);

    ukem_p.setEditable(false);

    jButton48.setEnabled(false);

    jButton39.setEnabled(false);

}

public void aktif_p(){

    hrg_beli_p.setEditable(true);

    diskon_p.setEditable(true);

```



```

jml_box_p.setEditable(true);

jml_brg_p.setEditable(true);

ubay_p.setEditable(true);

jButton48.setEnabled(true);
}

public void bersih_p(){

    kd_mB.setText("");

    nm_brg_p.setText("");

    hrg_beli_p.setText("");

    stok_p.setText("");

    sbttl_p.setText("");

    diskon_p.setText("");

    jml_box_p.setText("");

    jml_brg_p.setText("");

    ttl_p.setText("");

    ubay_p.setText("");

    ukem_p.setText("");

}

// ambil id akun

private String ambil_id1, ambil_id2,ambil_id3,ambil_id4;

public String getambil_id1(){

    return ambil_id1;

}

```




```

public String getambil_id2(){
    return ambil_id2;
}

private void ambil_id_akun1_pembelian()
{
    try {
        String p="Pembelian",q="PEMBELIAN",r="pembelian";

        String sql = "Select * from akun1 "
            + "where nm_akun='"+p+"' OR nm_akun='"+q+"' OR nm_akun='"+r+"'";
        kon.rs = kon.st.executeQuery(sql);
        while (kon.rs.next()) {
            ambil_id1=kon.rs.getString("kd_akun");
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

private void ambil_id_akun2_pembelian()
{
    try {
        String p="Kas",q="KAS",r="kas";

        String sql = "Select * from akun1 "
            + "where nm_akun='"+p+"' OR nm_akun='"+q+"' OR nm_akun='"+r+"'";

```

```

        kon.rs = kon.st.executeQuery(sql);

        while (kon.rs.next()) {

            ambil_id2=kon.rs.getString("kd_akun");

        }

    } catch (SQLException e) {

        JOptionPane.showMessageDialog(null, e);

    }

}

//untuk debet
private void simpan_detail_jurnal1_p()
{
    try{

        int nol=0;

        String sql="insert into detail_jurnal1 values('"+no_fak_p.getText()+"',"

            +"""+ambil_id1+"","

            +"""+ttl_p.getText()+""," //debet

            +"""+nol+"')";        //kredit

        kon.st.executeUpdate(sql);

    }

    catch(SQLException e){

        JOptionPane.showMessageDialog(null, e);

    }

}

```

```

//untuk kredit

private void simpan_detail_jurnal2_p()

{

    try{

        int nol=0;

        String sql="insert into detail_jurnal1 values('"+no_fak_p.getText()+"',"

            +"""+ambil_id2+"',"

            +"""+nol+"'," //debet

            +"""+ttl_p.getText()+"')"; //kredit

        kon.st.executeUpdate(sql);

    }

    catch(SQLException e){

        JOptionPane.showMessageDialog(null, e);

    }

}

//untuk simpan jurnal1

private void simpan_jurnal_p()

{

    try{

        String nm="Pembelian";

        int nol=0;

        String sql="insert into jurnal1 values('"+no_fak_p.getText()+"',"

            +"""+nm+"',"

```

```

        +"""+tgl_p.getText()+""","
        +"""+bulan_p.getText()+""","
        +"""+tahun_p.getText()+""");

    kon.st.executeUpdate(sql);

}

catch(SQLException e){

    JOptionPane.showMessageDialog(null, e);

}

}

private void simpan_detail_faktur_p()
{
    try{

        String kd_brg=""; //sementara kosong di isi saat input barang

        String sql="insert into detail_faktur_beli values('"+no_fak_p.getText()+""","
        +"""+nm_brg_p.getText()+""","
        +"""+stok_p.getText()+""","
        +"""+jml_brg_p.getText()+""","
        +"""+sbttl_p.getText()+""","
        +"""+kd_brg+"')";

        kon.st.executeUpdate(sql);

    }

    catch(SQLException e){

        JOptionPane.showMessageDialog(null, e);
    }
}

```

```

    }

    }

    private void simpan_faktur_p()
    {
        try{
            String sql="insert into faktur_beli values('"+no_fak_p.getText()+"",
                +""+tgl_p.getText()+"",
                +""+bulan_p.getText()+"",
                +""+tahun_p.getText()+"",
                +""+ttl_p.getText()+"",
                +""+kode_supplier+"')";
            kon.st.executeUpdate(sql);
        }
        catch(SQLException e){
            JOptionPane.showMessageDialog(null, e);
        }
    }
}

```

```

//SimpleDateFormat noformat_p=new SimpleDateFormat("yyyyMMdd");

```

```

public String no_fak_p()

```

```

{
    String urutan=null;

    try{

        kon.rs=kon.st.executeQuery("select right(kd_faktur,3)+1"
        + " from faktur_beli as Nomor order by kd_faktur desc");

        if(kon.rs.next())
        {
            urutan=kon.rs.getString(1);
            while(urutan.length()<3)
                urutan="0"+urutan;
            urutan="F-"+getWaktu()+urutan;
        }else
        {
            urutan="F-"+getWaktu()+"1";
        }
    }

    catch (Exception e){

        JOptionPane.showMessageDialog(null, e);

    }

    return urutan;
}

void setTanggal(){

    java.util.Date skrg=new java.util.Date();

```

```

        java.text.SimpleDateFormat kal=new java.text.SimpleDateFormat("dd");

        tgl_p.setText(kal.format(skrng));

        tgl_t.setText(kal.format(skrng));

    }

    void setBulan(){

        java.util.Date skrng=new java.util.Date();

        java.text.SimpleDateFormat kal=new
java.text.SimpleDateFormat("MMMMMMMMMM");

        bulan_p.setText(kal.format(skrng));

        bln_t.setText(kal.format(skrng));

    }

    void setTahun(){

        java.util.Date skrng=new java.util.Date();

        java.text.SimpleDateFormat kal=new java.text.SimpleDateFormat("YYYY");

        tahun_p.setText(kal.format(skrng));

        thn_t.setText(kal.format(skrng));

    }

    private void Bacatbl_sm_p()

    {

        try{

            String sql="Select stb.*, s.nm_supplier From sementara_transaksi_beli stb LEFT
JOIN supplier s ON stb.kd_supplier=s.kd_supplier";

            kon.rs=kon.st.executeQuery(sql);

```



```

ResultSetMetaData m=kon.rs.getMetaData();

int kolom=m.getColumnCount();

int baris=0;

while (kon.rs.next()){

    baris=kon.rs.getRow();

}

datasementara=new Object[baris][kolom];

int x=0;
kon.rs.beforeFirst();
while(kon.rs.next()){

    datasementara[x][0]=kon.rs.getString("s.nm_supplier");
    datasementara[x][1]=kon.rs.getString("stb.nm_brg");
    datasementara[x][2]=kon.rs.getString("stb.jml_beli");
    datasementara[x][3]=kon.rs.getString("stb.subtotal");
    x++;

}

tbl_sm_p.setModel(new DefaultTableModel(datasementara,labelsementara));

}

catch(SQLException e){

    JOptionPane.showMessageDialog(null, e);

}

}

private void simpan_sementara_transaksi_beli()

```

```

{
    try{

        String sql="insert into sementara_transaksi_beli values('"+kode_supplier+"",
            +"""+nm_brg_p.getText()+"",
            +"""+stok_p.getText()+"",
            +"""+sbttl_p.getText()+"");

        kon.st.executeUpdate(sql);

        Bacatbl_sm_p();
    }
    catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);
    }
}

private void hapus_sementara_transaksi_beli(){
    int row=tbl_sm_p.getSelectedRow();
    int x;

    int total=Integer.parseInt(ttl_p.getText());

    x=Integer.parseInt((String)tbl_sm_p.getValueAt(row, 3));

    total=total-x;

    ttl_p.setText(Integer.toString(total));

    try{

        String sql="delete from sementara_transaksi_beli where
nm_brg='"+nama_barang_p+"'";

```

```

        kon.st.executeUpdate(sql);

        Bacatbl_sm_p();
    }

    catch(SQLException e){

        JOptionPane.showMessageDialog(null, e);
    }
}

private void hapus_detail_faktur_beli(){

    int row=tbl_sm_p.getSelectedRow();
    try{

        String sql="delete from detail_faktur_beli where
nm_brg='"+nama_barang_p+"'";

        kon.st.executeUpdate(sql);

        Bacatbl_sm_p();
    }

    catch(SQLException e){

        JOptionPane.showMessageDialog(null, e);
    }
}

private void Hapus_Tabel_sementara_transaksi_beli(){

    try{

        String sql="delete from sementara_transaksi_beli";

        kon.st.executeUpdate(sql);
    }
}

```

```

        Bacatbl_sm_p();
    }

    catch(SQLException e){

        System.out.println("Koneksi Gagal"+e.toString());

    } }

```

3. Code penjualan panel

//untuk penjualan panel

```

private int jml_per_pcs;
public String box_t;
public String getbox_t()
{
    return box_t;
}
private void nonaktif_t()
{
    notrans_t.setEditable(false);

    kd_t.setEditable(false);

    nm_t.setEditable(false);

    kat_t.setEditable(false);

    sat_t.setEditable(false);

    hrg_t.setEditable(false);
}

```



```

jumbel_t.setEditable(false);

sbttl_t.setEditable(false);

ubay_t.setEditable(false);

ukem_t.setEditable(false);

jButton69.setEnabled(false);

jButton68.setEnabled(false);

jButton74.setEnabled(false);

jButton75.setEnabled(false);
}

private void aktif_t()
{
    jumbel_t.setEditable(true);
    ubay_t.setEditable(true);
    jButton74.setEnabled(true);
    jButton75.setEnabled(true);
}

private void bersih_t()
{
    kd_t.setText("");
    nm_t.setText("");
    kat_t.setText("");
    sat_t.setText("");
    hrg_t.setText("");
}

```



```

        jumbel_t.setText("");

        sbttl_t.setText("");

    }

    public String no_t()
{
    String urutan=null;

    try{
        kon.rs=kon.st.executeQuery("select right(notransaksi,3)+1"
        + " from transaksi_jual as Nomor order by notransaksi desc");
        if(kon.rs.next())
        {
            urutan=kon.rs.getString(1);
            while(urutan.length()<3)
                urutan="0"+urutan;
            urutan="T-"+getWaktu()+urutan;
        }else
        {
            urutan="T-"+getWaktu();
        }
    }

    catch (Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
}

```

```

        return urutan;
    }

    private void Bacatbl_jual()
    {
        try{

            String sql="Select *From sementara_transaksi_jual";

            kon.rs=kon.st.executeQuery(sql);

            ResultSetMetaData m=kon.rs.getMetaData();

            int kolom=m.getColumnCount();

            int baris=0;

            while (kon.rs.next()){

                baris=kon.rs.getRow();

            }

            datajual=new Object[baris][kolom];

            int x=0;

            kon.rs.beforeFirst();

            while(kon.rs.next()){

                datajual[x][0]=kon.rs.getString("kd_brg");

                datajual[x][1]=kon.rs.getString("nm_brg");

                datajual[x][2]=kon.rs.getString("satuan");

                datajual[x][3]=kon.rs.getString("hrg");

                datajual[x][4]=kon.rs.getString("jumbel");

                datajual[x][5]=kon.rs.getString("subtotal");

```



```

        x++;

    }

    tbl_jual.setModel(new DefaultTableModel(datajual,labeljual));

}

catch(SQLException e){

    JOptionPane.showMessageDialog(null, e);

}

}

private String total_seluruh_hpp;

private void Bacatbl_sementara_jurnal()
{
    try{

        String sql="Select *From sementara_jurnal";

        kon.rs=kon.st.executeQuery(sql);

        ResultSetMetaData m=kon.rs.getMetaData();

        int kolom=m.getColumnCount();

        int baris=0;

        while (kon.rs.next()){

            baris=kon.rs.getRow();

        }

        datajual=new Object[baris][kolom];

        int x=0;

        kon.rs.beforeFirst();
    
```

```

while(kon.rs.next()){

    datajual[x][0]=kon.rs.getString("total_hpp");

    datajual[x][1]=kon.rs.getString("kd_brg");

    x++;

}

tbl_s_jurnal.setModel(new DefaultTableModel(datajual,labeljual));

}

catch(SQLException e){

    JOptionPane.showMessageDialog(null, e);

}

}

public void simpan_transaksi_jual()

{

    try{

        String sql="insert into transaksi_jual values('"+notrans_t.getText()+"',"

        +"""+tgl_t.getText()+"",

        +"""+bln_t.getText()+"",

        +"""+thn_t.getText()+"",

        +"""+ttl_t.getText()+"",

        +"""+ubay_t.getText()+"",

        +"""+ukem_t.getText()+"")";

        kon.st.executeUpdate(sql);

        Bacatbl_jual();
    
```

```

    }

    catch(SQLException a){

        JOptionPane.showMessageDialog(null, a);

    }

}

//buat masukin total barang keluar

public void  simpan_barang_keluar()

{

    try{

        String sql="insert into barang_keluar values('"+bln_t.getText()+"",
            +"""+thn_t.getText()+"",
            +"""+kd_t.getText()+"",
            +"""+jumbel_t.getText()+"",
            +"""+sbttl_t.getText()+"");

        kon.st.executeUpdate(sql);

    }

    catch(SQLException a){

        JOptionPane.showMessageDialog(null, a);

    }

}

private String

kd_brg_bk="",jml_brg_keluar,hasil_jml,thn_bk,bln_bk,ttl_bk="0",hasil_ttl_bk="0";

private void ambil_data_barang_keluar()

```

```

{
try {
    //ambil data jumlah berdasarkan kd_brg bulan dan tahun

    String sql = "Select * from barang_keluar "

        + "where bln='"+bln_t.getText()+"' AND thn='"+thn_t.getText()+"' "

        + "AND kd_brg='"+kd_t.getText()+"'";

    kon.rs = kon.st.executeQuery(sql);

    while (kon.rs.next()) {

        jml_brg_keluar=kon.rs.getString("jml");
        kd_brg_bk=kon.rs.getString("kd_brg");
        ttl_bk=kon.rs.getString("total");
    }
} catch (SQLException e) {

    JOptionPane.showMessageDialog(null, e);
}
}

public void update_barang_keluar(){

try {

    String sql = "UPDATE barang_keluar SET jml='" + hasil_jml + "',"

        + "total='"+hasil_ttl_bk+"'"

        + " where kd_brg='" + kd_t.getText() + "'";

    kon.st.executeUpdate(sql);

} catch (SQLException e) {

```

```

        JOptionPane.showMessageDialog(null, e);
    }
}

private void simpan_detail_transaksi_jual()
{
    try{
        String sql="insert into detail_transaksi_jual values('"+notrans_t.getText()+"",
            +"""+hrg_t.getText()+"",
            +"""+sat_t.getText()+"",
            +"""+jumbel_t.getText()+"",
            +"""+jml_per_pcs+"",
            +"""+sbtlt_t.getText()+"",
            +"""+kd_t.getText()+"");
        kon.st.executeUpdate(sql);
        Bacatbl_jual();
    }

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

private void simpan_sementara_jurnal()
{
    try{

```

```

String sql="insert into sementara_jurnal values('"+total_hpp+"",
        + ""'+kd_t.getText()+''");
kon.st.executeUpdate(sql);
Bacatbl_sementara_jurnal();
}
catch(SQLException e){
JOptionPane.showMessageDialog(null, e);
}
}
private void simpan_sementara_transaksi_jual()
{
try{
String sql="insert into sementara_transaksi_jual values('"+kd_t.getText()+",
        + ""'+nm_t.getText()+",
        + ""'+sat_t.getText()+",
        + ""'+hrng_t.getText()+",
        + ""'+jumbel_t.getText()+",
        + ""'+sbttl_t.getText()+''");
kon.st.executeUpdate(sql);
Bacatbl_jual();
}
catch(SQLException e){
JOptionPane.showMessageDialog(null, e);

```

```

    }

}

private void hapus_sementara_transaksi_jual(){

    try{

        String sql="delete from sementara_transaksi_jual where
kd_brg='"+kd_t.getText()+"'";

        kon.st.executeUpdate(sql);

        Bacatbl_jual();

    }

    catch(SQLException e){

        JOptionPane.showMessageDialog(null, e);

    }

    int ttl=0;
    for(int t=0; t<tbl_jual.getRowCount();t++)

    {

        int sub=Integer.parseInt((String)tbl_jual.getValueAt(t, 5));

        ttl+=sub;

    }

    ttl_t.setText(Integer.toString(ttl));

}

private void hapus_sementara_jurnal(){

```



```

try{

    String sql="delete from sementara_jurnal where kd_brg='"+kd_t.getText()+"''";

    kon.st.executeUpdate(sql);

    Bacatbl_sementara_jurnal();

}

catch(SQLException e){

    JOptionPane.showMessageDialog(null, e);

}

int ttl=0;
for(int t=0; t<tbl_s_jurnal.getRowCount();t++)
{

    int sub=Integer.parseInt((String)tbl_s_jurnal.getValueAt(t, 0));

    ttl+=sub;

}

total_seluruh_hpp=Integer.toString(ttl);

}

private void hapus_detail_jurnal(){

    int row=tbl_sm_p.getSelectedRow();

    try{

        String sql="delete from detail_transaksi_jual where kd_brg='"+kd_t.getText()+"'

"

        + "AND notransaksi='"+notrans_t.getText()+"''";

```

```

        kon.st.executeUpdate(sql);

        Bacatbl_jual();
    }

    catch(SQLException e){

        JOptionPane.showMessageDialog(null, e);
    }
}

private void Hapus_Tabel_sementara_transaksi_jual(){
    try{
        String sql="delete from sementara_transaksi_jual";
        kon.st.executeUpdate(sql);
        Bacatbl_jual();
    }

    catch(SQLException e){
        System.out.println("Koneksi Gagal"+e.toString());
    }
}

private void Hapus_Tabel_sementara_jurnal(){
    try{
        String sql="delete from sementara_jurnal";
        kon.st.executeUpdate(sql);
        Bacatbl_sementara_jurnal();
    }
}

```

```

    }

    catch(SQLException e){

        System.out.println("Koneksi Gagal"+e.toString());

    }

}

private void c_struk()
{
    try{
        String file = "src/laporan/struk.jasper";
        HashMap param = new HashMap();
        param.put("kd_trans",notrans_t.getText());

        JasperPrint print = JasperFillManager.fillReport(file, param, kon.setKoneksi());
        JasperViewer.viewReport(print, false);
    }

    catch (Exception e){

        JOptionPane.showMessageDialog(null, e.getMessage());

    }

}

//ambil harga pokok barang

private String hpp_brg="0";

private void ambil_HPP()
{

```

```

try {

    String sql = "Select * from barang "

        + "where kd_brg='"+kd_t.getText()+"'";

    kon.rs = kon.st.executeQuery(sql);

    while (kon.rs.next()) {

        hpp_brg=kon.rs.getString("hrg_beli"); //HPP

    }

} catch (SQLException e) {

    JOptionPane.showMessageDialog(null, e);

}

}

private void ambil_id_akun1_penjualan()

{

try {

String p="Penjualan",q="PENJUALAN",r="penjualan";

String sql = "Select * from akun1 "

        + "where nm_akun='"+p+"' OR nm_akun='"+q+"' OR nm_akun='"+r+"'";

    kon.rs = kon.st.executeQuery(sql);

    while (kon.rs.next()) {

        ambil_id1=kon.rs.getString("kd_akun");

    }

} catch (SQLException e) {

    JOptionPane.showMessageDialog(null, e);

```

```

    }
}

private void ambil_id_akun2_penjualan()
{
    try {
        String p="Kas",q="KAS",r="kas";

        String sql = "Select * from akun1 "
            + "where nm_akun='"+p+"' OR nm_akun='"+q+"' OR nm_akun='"+r+"'";

        kon.rs = kon.st.executeQuery(sql);
        while (kon.rs.next()) {
            ambil_id2=kon.rs.getString("kd_akun");
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
}

```

```

private void ambil_id_akun3_penjualan()
{
    try {
        String p="hpp",q="HPP",r="Hpp";

        String sql = "Select * from akun1 "
            + "where nm_akun='"+p+"' OR nm_akun='"+q+"' OR nm_akun='"+r+"'";

        kon.rs = kon.st.executeQuery(sql);
    }
}

```

```

        while (kon.rs.next()) {

            ambil_id3=kon.rs.getString("kd_akun"); //HPP

        }

    } catch (SQLException e) {

        JOptionPane.showMessageDialog(null, e);

    }

}

private void ambil_id_akun4_penjualan()
{
try {

String p="Persediaan",q="PERSEDIAAN",r="persediaan";

String sql = "Select * from akun1 "

        + "where nm_akun='"+p+"' OR nm_akun='"+q+"' OR nm_akun='"+r+"'";

kon.rs = kon.st.executeQuery(sql);

while (kon.rs.next()) {

    ambil_id4=kon.rs.getString("kd_akun"); //persediaan

        }

    } catch (SQLException e) {

        JOptionPane.showMessageDialog(null, e);

    }

}

//jurnal penjualan dan kas

//untuk debet

```

```

private void simpan_detail_jurnal1_j()
{
    try{
        int nol=0;

        String sql="insert into detail_jurnal1 values('"+notrans_t.getText()+"",
            +"""+ambil_id2+"",
            +"""+ttl_t.getText()+"", " //debet
            +"""+nol+""); //kredit
        kon.st.executeUpdate(sql);
    }
    catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);
    }
}

//untuk kredit
private void simpan_detail_jurnal2_j()
{
    try{
        int nol=0;

        String sql="insert into detail_jurnal1 values('"+notrans_t.getText()+"",
            +"""+ambil_id1+"",
            +"""+nol+"", " //debet
            +"""+ttl_t.getText()+""); //kredit
    }
}

```

```

        kon.st.executeUpdate(sql);

    }

    catch(SQLException e){

        JOptionPane.showMessageDialog(null, e);

    }

}

//jurnal penjualan akun HPP & persediaan

//untuk debet

private String total_hpp;

private void simpan_detail_jurnal3_j()

{

    try{

        int nol=0;

        String sql="insert into detail_jurnal1 values('"+notrans_t.getText()+"",

            +"""+ambil_id3+"",

            +"""+total_seluruh_hpp+"", " //debet

            +"""+nol+"")";        //kredit

        kon.st.executeUpdate(sql);

    }

    catch(SQLException e){

        JOptionPane.showMessageDialog(null, e);

    }

}

```



```

//untuk kredit

private void simpan_detail_jurnal4_j()

{
    try{

        int nol=0;

        String sql="insert into detail_jurnal1 values('"+notrans_t.getText()+"",
            +"""+ambil_id4+"",
            +"""+nol+"", //debet
            +"""+total_seluruh_hpp+"")"; //kredit
        kon.st.executeUpdate(sql);
    }
    catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);
    }
}

//untuk simpan jurnal1

private void simpan_jurnal_j()

{
    try{

        String nm="Penjualan";

        int nol=0;

        String sql="insert into jurnal1 values('"+notrans_t.getText()+"",
            +"""+nm+"",

```

```

+"""+tgl_p.getText()+","
+"""+bulan_p.getText()+","
+"""+tahun_p.getText()+""");

kon.st.executeUpdate(sql);

}

catch(SQLException e){

    JOptionPane.showMessageDialog(null, e);

}

}

```



3.5.2. Blackbox Testing

A. Form Login

Tabel.III.33.

Hasil Pengujian Blackbox Testing Login Admin

No.	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1.	<i>Username</i> dan <i>password</i> tidak diisi kemudian klik <i>login</i>	Username: (kosong) Password: (kosong)	Sistem menolak akses dan menampilkan pesan “kode user atau password salah”	Sesuai harapan	Valid
2.	<i>Username</i> diisi dan <i>password</i> tidak diisi kemudian klik <i>login</i>	Username: 2855489 (Benar) Password: (kosong)	Sistem menolak akses dan menampilkan pesan “kode user atau password salah”	Sesuai harapan	Valid
3.	<i>Username</i> tidak diisi dan <i>password</i> diisi kemudian klik <i>login</i>	Username: (kosong) Password: 123456 (benar)	Sistem menolak akses dan menampilkan pesan “kode user atau password salah”	Sesuai harapan	Valid
4.	<i>Username</i> dan <i>password</i> diisi kemudian klik <i>login</i>	Username: 2855489 (benar) Password: 123456 (benar)	Sistem akan menerima akses dan masuk ke menu utama	Sesuai harapan	Valid

B. Form Data Pengguna

Tabel.III.34.

Hasil Pengujian *Blackbox Testing*

No.	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1.	Saat klik Tambah, kode pengguna akan tampil secara otomatis dan acak	Klik Tambah dan kode pengguna secara otomatis tertampil	Kode pengguna tampil otomatis	Sesuai harapan	valid
2.	Input data tidak lengkap, jika Nama Pengguna tidak terisi	Kode Pengguna: (otomatis) Nama Pengguna: (kosong) Password: (123456) Untuk Bagian: (kasir)	Proses penyimpanan ditolak, dan akan tampil pesan “Nama Pengguna tidak boleh kosong”	Sesuai harapan	Valid
3.	Input data tidak lengkap, jika Password tidak terisi	Kode Pengguna: (otomatis) Nama Pengguna: (dinda) Password: (kosong) Untuk Bagian: (kasir)	Proses penyimpanan ditolak, dan akan tampil pesan “Kata Sandi tidak boleh kosong!”	Sesuai harapan	Valid
4.	Input data tidak lengkap, jika pilihan Untuk Bagian tidak dipilih	Kode Pengguna: (otomatis)	Proses penyimpanan tidak berhasil, dan	Tidak sesuai harapan	Tidak valid

		Nama Pengguna: (dinda) Password: (123456) Untuk Bagian: (kosong)	muncul pesan “Untuk Bagian harus dipilih”		
5.	Input data lengkap	Kode Pengguna: (otomatis) Nama Pengguna: (dinda) Password: (123456) Untuk Bagian: (kasir)	Proses penyimpanan berhasil dan data tersimpan pada daftar tabel Pengguna	Sesuai harapan	Valid

C. Form Pembelian

Tabel.III.35.

Hasil Pengujian *Black Box form* Pembelian

No.	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Input data pembelian tidak lengkap	Nama barang, supplier, harga beli, jumlah box, jumlah pcs terisi, namun tidak klik hitung.	Tombol simpan tidak aktif	Sesuai harapan	Valid
2.	input data pembelian lengkap	Nama barang, supplier, stok, harga beli, subtotal, jumlah box, jumlah pcs klik hitung dan total terisi	Tombol simpan aktif	Sesuai harapan	Valid

3.5.3. Spesifikasi *Hardware* dan *Software*

Tabel.III.36.

Spesifikasi *Hardware* dan *Software*

Kebutuhan	Keterangan
Sistem	Windows 10 64-bit
Processor	Intel Dual Core N3050, up to 2.16GHz
RAM	2 GB
Harddisk	500GB
Monitor	LED 15-inch
Keyboard	Standar
Printer	Brother DCP-T310
Mouse	Standar
Software	Kebutuhan menjalankan aplikasi secara lokal: Bahasa script programming : Netbeans IDE 8.2 Web Server : XAMPP Control Panel v3.2.2 DBMS : MySQL