

BAB III

PEMBAHASAN

3.1. Tinjauan Perusahaan

PT Nasa adalah sebuah badan usaha swasta dalam bidangnya melakukan pendistribusian produk, terdapat *stokist* yang berperan sebagai pusat informasi serta distribusi barang bagi distributor PT Nasa. Distributor dari Nasa yang menghubungkan perusahaan dengan konsumen dengan cara menawarkan produk-produk dimasyarakat.

Salah satu aktivitas yang ada pada PT Nasa Stokist T 2309 Cikampek adalah proses pembelian produk ke PT Nasa Stokist Center, karena produk merupakan salah satu yang paling penting dalam proses perdagangan di dalam menjalankan usahanya. Dengan menyediakan produk di Stokist T 2309 tersebut akan memudahkan para distributor untuk membeli barang dagangnya dan dengan mudah untuk menawarkan kepada konsumen.

Oleh karena itu, produk-produk yang paling penting dalam proses perdagangan, maka perlunya suatu prosedur yang baik dalam mengelola dan membeli produk tersebut agar dapat terkontrol dengan baik. Stokist T 2309 sebagai pusat informasi serta distribusi barang bagi para distributornya, sangatlah membutuhkan manajemen yang baik dalam menjalankan usahanya.

PT Nasa Stokist T 2309 didirikan dengan maksud untuk memudahkan para distributor di daerah untuk membeli produk yang akan dijual tanpa harus membeli produk ke *stokist center* kabupaten ataupun berkunjung langsung ke perusahaan. Hal ini tentu mempermudah masyarakat dalam mendapatkan produk Nasa yang diinginkan

dan bertujuan untuk memajukan kesejahteraan bangsa dan nasional dalam rangka mewujudkan perekonomian bangsa yang lebih maju.

3.1.1. Sejarah Perusahaan

PT Nasa merupakan salah satu perusahaan yang didirikan oleh Darsono Tan. Perusahaan ini terletak di Daerah Istimewa Yogyakarta dan berdiri pada tanggal 1 Oktober 2002. Adapun alamat lengkap dari PT Nasa di Jl. Ring Road Barat no 72 Salakan di Jogja, dengan Nomor Surat Izin Usaha Perdagangan (SIUP) PT Nasa 8120103762753

Sedangkan Stokist T 2309 merupakan stokis di daerah sebagai pusat informasi serta distribusi barang bagi distributor PT Nasa. Berawal dari distributor kecil sampai akhirnya resmi menjadi Stokist dengan kode Stokist T 2309. Usaha ini terletak di Daerah Cikampek dan berdiri pada tanggal 13 Februari 2019 yang beralamat di Kampung Cariu RT.03 RW.05 Kel.Wancimekar Kec.Kota Baru, Karawang-Jawa Barat, 41376.

NASA terdiri dari 2 kata yakni Natural dan Nusantara. Natural memiliki pengertian yakni mengutamakan teknologi alami yang ramah lingkungan, sedangkan Nusantara berarti hasil karya anak bangsa. Nasa mengusung konsep pembuatan produk yang terbuat dari beragam produk alami yang berhasil dikembangkan di bidang Agro kompleks. Berdirinya Nasa, tidak luput dari keprihatinan terhadap kondisi ekosistem yang ada di bumi.

Berawal dari aktivitas budidaya sekaligus mencermati kondisi lingkungan hidup, maka orientasi berkembang ke bidang Penelitian dan pengembangan (R&D) khususnya perbaikan lingkungan hidup/ekosistem Sub. Sektor Pertanian. Hasil-hasil

yang didapat dari kegiatan Penelitian dan pengembangan pada awalnya berupa teknologi pupuk organik cair dan Hormon/Zat Pengatur Tumbuh Organik.

1 Oktober 2002 ditetapkan untuk dikembangkan lebih intensif disertai pola pembinaan SDM (Pola Pikir, Mental, Motivasi, *Skill* dan Modal) menggunakan *brand* Perusahaan PT. NATURAL NUSANTARA (PT. NASA) Group.

Adapun Visi, Misi dan Tujuan PT Nasa dalam menjalankan sebuah usahanya, yaitu:

Visi perusahaan:

“Hidup Bahagia dan Sejahtera Selaras Alam”.

Misi perusahaan:

“Bersama Menuju Masa Depan Lebih Baik”.

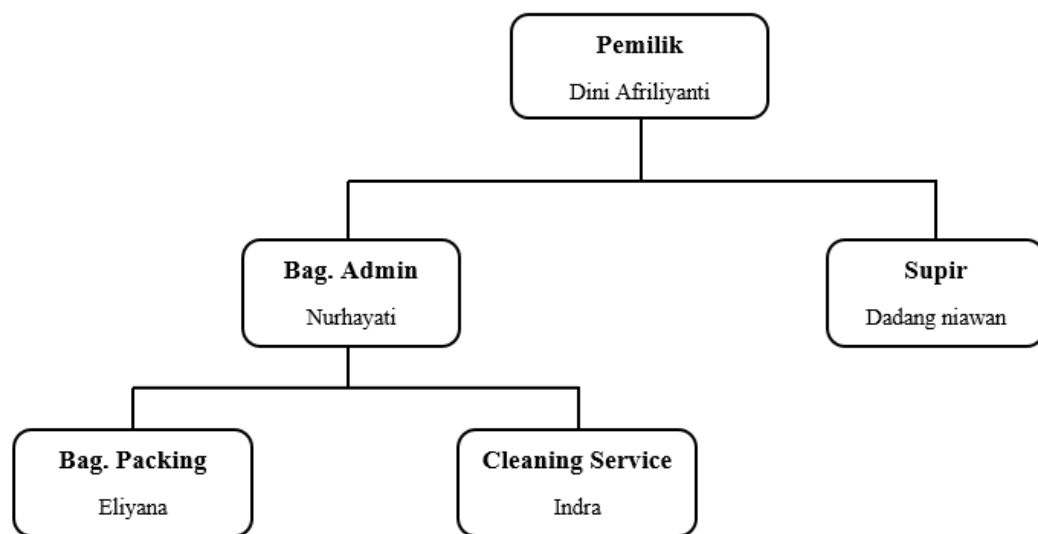
Tujuan perusahaan:

“Pemberdayaan Potensi Sumber Daya Alam (SDA) dan Sumber Daya Manusia (SDM) yang berkelanjutan dengan semangat mencapai Indonesia Makmur Raya Berkeadilan serta memberikan sesuatu yang bermanfaat dan bermakna bagi Dunia.

1. Struktur Organisasi

Struktur organisasi merupakan suatu hal yang penting dalam sebuah perusahaan demi kelancaran berjalannya perusahaan. Agar dapat mencapai tujuan tersebut diperlukan koordinasi dan Kerja sama antara fungsi yang ada di dalam perusahaan untuk mengolah perusahaan dengan baik dan optimal.

Adapun struktur organisasi ini dapat dikatakan sebagai suatu kerangka yang mewujudkan suatu pola dari hubungan antara kedudukan dan peranan dalam suatu kerja sama. Berikut adalah struktur organisasi yang dimiliki PT Nasa Stokist T 2309:



Sumber: (PT Nasa Stokist T 2309,2020)

Gambar III.1.
Struktur Organisasi PT Nasa Stokist T 2309

Sesuai yang tercantum pada bagan organisasi PT Nasa Stokist T 2309 diadakan pembagian wewenang dan tanggung jawab dalam kaitan dengan operasional sehari-hari. Adapun uraian tugas dari masing-masing bagian adalah sebagai berikut:

2. Fungsi Organisasi

1. Pemilik

- a. Bertanggung jawab atas seluruh kegiatan usaha.
- b. Membuat kebijakan dan peraturan yang berhubungan dengan kinerja pegawai.
- c. Mengevaluasi kinerja pegawai.
- d. Menerima dan memeriksa setiap laporan dari tiap-tiap bagian
- e. Mengawasi semua kegiatan pembelian dan penjualan pada Stokist T 2309
- f. Mengatur segala macam pembayaran dan penerimaan uang.

2. Bagian Admin

- a. Melakukan pengecekan dan pencatatan *stock* barang.
- b. Melakukan pembelian berdasarkan order pembelian.

- c. Mengawasi agar semua barang yang dibeli sesuai dengan pesanan.
- d. Bertanggung jawab atas mutu barang yang telah dibeli.
- e. Mengecek dan menyiapkan barang yang akan dikirim.
- f. Mencatat segala transaksi harian.
- g. Membuat laporan keuangan untuk diserahkan kepada pemilik

3. Bagian *Packing*

Melakukan pengemasan barang pesanan

4. *Cleaning Service*

Bertanggung jawab atas segala kebersihan di lingkungan usaha.

5. Sopir

Melakukan pengiriman barang sesuai dengan pesanan.

3.2. Tinjauan Kasus

Pada bab ini penulis akan melakukan pengkajian mengenai proses bisnis sistem berjalan pada pembelian secara tunai pada PT Nasa Stokist T 2309 Cikampek. Dimana proses bisnis sistem berjalan pembelian secara tunai meliputi prosedur permintaan barang dan pemesanan barang, prosedur penerimaan dan pembayaran barang, dan prosedur pembuatan laporan.

3.2.1. Proses Bisnis Sistem Berjalan

Sistem pembelian pada PT Nasa Stokist T 2309 yang berjalan saat ini adalah:

1. Prosedur Permintaan Barang dan Pemesanan Barang

Bagian admin mengecek persediaan barang, setelah itu membuat *form* permintaan pembelian barang (FPPB) untuk persediaan barang yang kurang, lalu bagian admin menyerahkan FPPB ke pemilik untuk di cek barang apa saja yang akan

dibeli. Kemudian jika FPPB di setuju oleh pemilik maka pemilik akan menyimpan FPPB dan menyerahkan FPPB ke *stokist center* untuk menerima pemesanan barang, tetapi jika pemilik tidak menyetujui permintaan pembelian barang maka pemilik menyerahkan kembali kertas catatan permintaan pembelian ke bagian admin.

2. Prosedur Penerimaan dan Pembayaran Barang

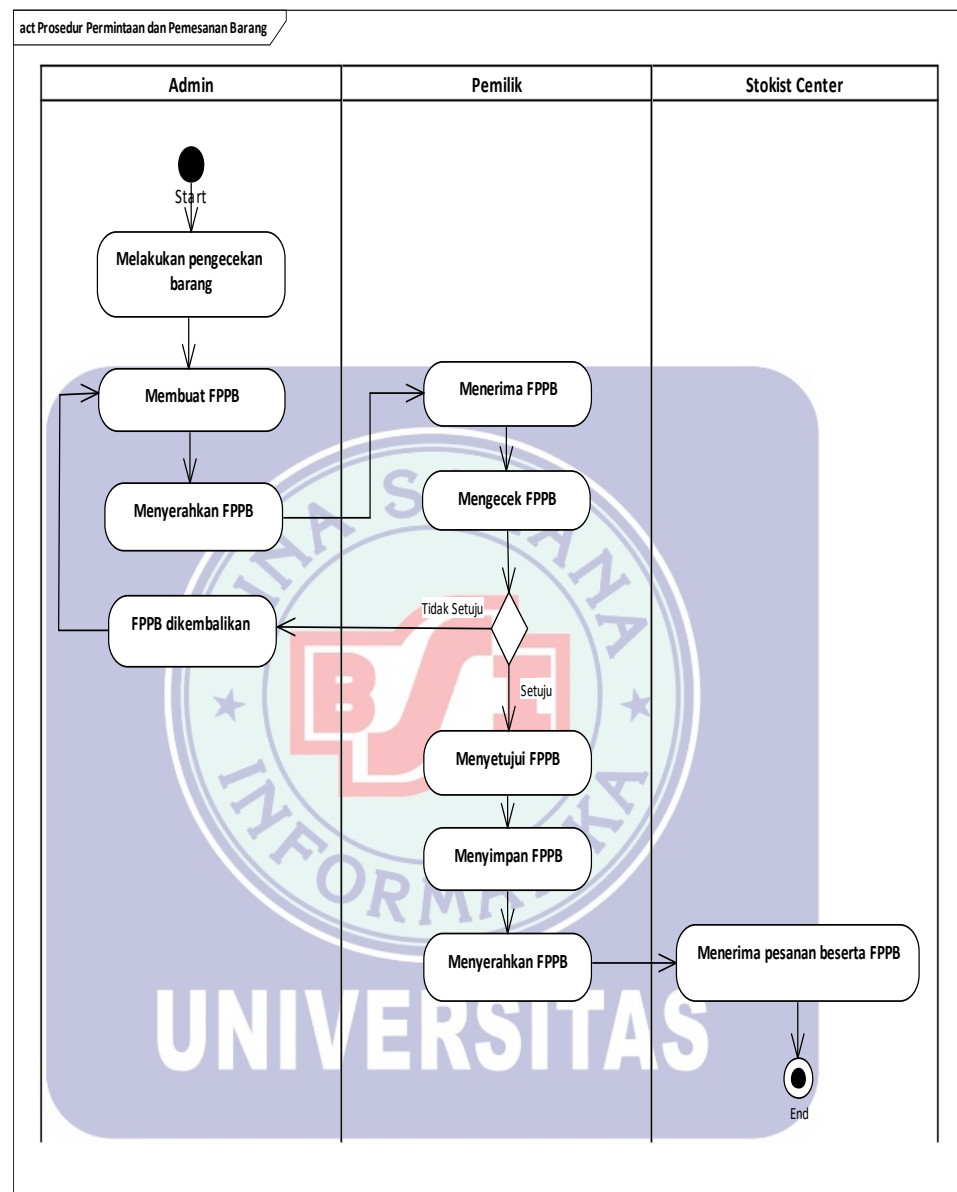
Setelah menerima FPPB *stokist center* menyiapkan barang beserta dengan dokumen surat jalan (SJ), tanda terima barang (TTB), kuitansi. Kemudian barang dan dokumen tersebut diserahkan kepada supir untuk di kirim ke Stokist T 2309 yang nantinya akan diterima oleh admin. Setelah menerima barang beserta dokumen dan memeriksa barang yang diterima sudah sesuai dengan yang dipesan atau tidak. Saat barang yang diterima admin tidak sesuai dengan yang di pesan maka admin mengirimkan pesan kepada admin *stokist center*, sedangkan jika barang yang datang sesuai yang di pesan dan tidak ada kerusakan maka bagian admin menyimpan barang, kemudian semua dokumen di berikan ke pemilik dan selanjutnya pemilik melakukan pembayaran.

3. Prosedur Pembuatan Laporan

Admin membuat laporan keuangan berdasarkan pengumpulan bukti transaksi dan melakukan rekapitulasi dari bukti transaksi tersebut. Admin membuat arsip laporan, Kemudian menyerahkan laporan keuangan kepada pemilik untuk disetujui. Jika pemilik tidak menyetujui maka laporan di kembalikan ke admin untuk di cek Kembali. Setelah disetujui pemilik mengarsipkan laporannya.

3.2.2. Activity Diagram

1. Activity Diagram Permintaan dan Pemesanan Barang

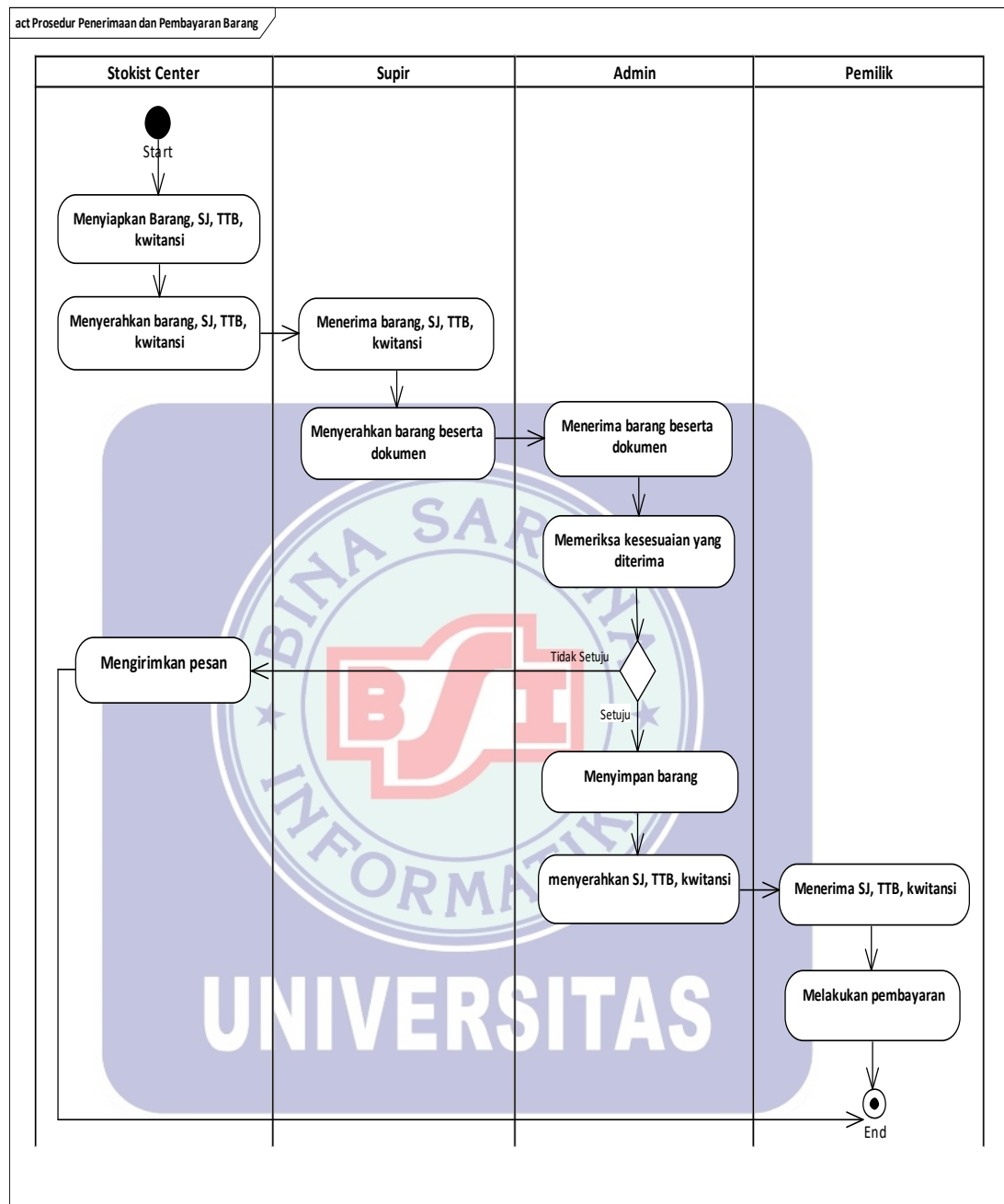


Keterangan :

FPPB : Form Permintaan Pembelian Barang

Gambar III.2.
Activity Diagram Permintaan dan Pemesanan Barang

2. Activity Diagram Penerimaan dan Pembayaran Barang



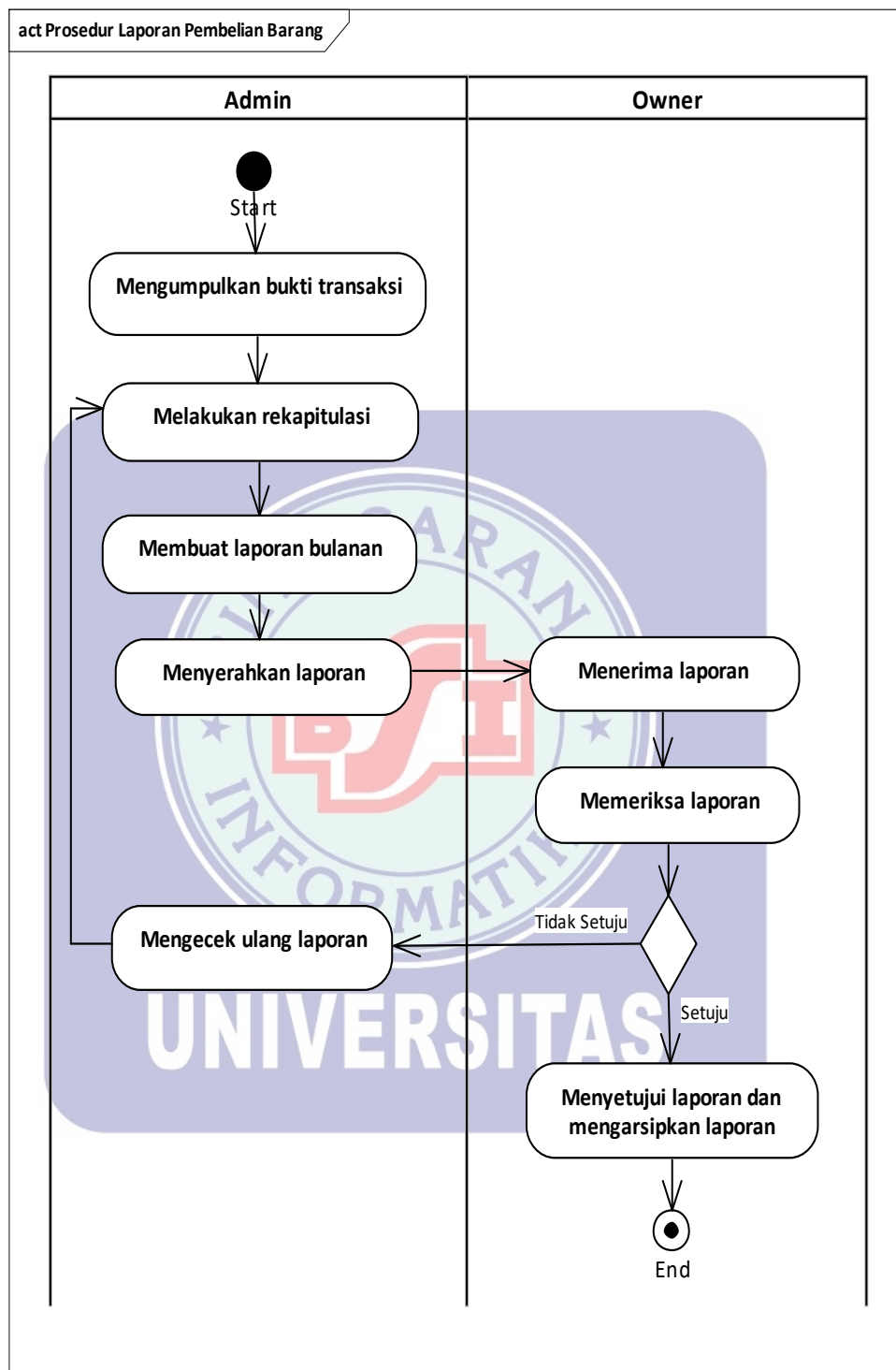
Keterangan :

SJ : Surat Jalan

TTB :Tanda Terima Barang

Gambar III.3.
Activity Diagram Penerimaan dan Pembayaran Barang

3. Activity Diagram Pembuatan Laporan



Gambar III.4.
Activity Diagram Pembuatan Laporan

3.2.3. Dokumen Masukan

Bentuk dokumen masukan (*input*) yang dibutuhkan dalam spesifikasi sistem akuntansi berjalan adalah:

1. Nama Dokumen : Surat Jalan
 - Fungsi : Sebagai bukti barang telah dikirim dari *stockist*
 - Sumber : *Stockist*
 - Tujuan : Bagian Admin
 - Frekuensi : Setiap ada pembelian barang
 - Media : Kertas
 - Jumlah : 2 Lembar
 - Format : Lampiran A1
2. Nama Dokumen : Kuitansi
 - Fungsi : Sebagai bukti pembayaran barang
 - Sumber : *Stockist*
 - Tujuan : Bagian Admin
 - Frekuensi : Setiap ada pembayaran
 - Media : Kertas
 - Jumlah : 2 Lembar
 - Bentuk : Lampiran A2
3. Nama Dokumen : Tanda Terima Barang
 - Fungsi : Sebagai bukti barang telah diterima dari *stockist*
 - Sumber : *Stockist*
 - Tujuan : Bagian Admin
 - Frekuensi : Setiap ada pembelian barang
 - Media : Kertas

Jumlah : 2 Lembar
 Format : Lampiran A3

3.2.4. Dokumen Keluaran

Bentuk dokumen keluaran (*output*) yang dibutuhkan dalam spesifikasi sistem akuntansi berjalan adalah:

1. Nama Dokumen : *Form Permintaan Pembelian Barang*
 - Fungsi : Untuk permintaan barang
 - Sumber : Bagian Admin
 - Tujuan : Pemilik
 - Frekuensi : Setiap ada permintaan barang
 - Media : Kertas
 - Jumlah : 2 Lembar
 - Format : Lampiran B1
2. Nama Dokumen : *Laporan Pembelian*
 - Fungsi : Sebagai bukti telah terjadi transaksi pembelian
 - Sumber : Bagian Admin
 - Tujuan : Pemilik
 - Frekuensi : Setiap bulan
 - Media : Kertas
 - Jumlah : 2 Lembar
 - Bentuk : Lampiran B2

3.2.5. Permasalahan Pokok

Setelah melihat dan mempelajari dari sistem berjalan pada PT Nasa Stokist T 2309, penulis mencoba memberikan suatu analisa tentang sistem tersebut. Dalam hal ini penulis menemukan beberapa permasalahan yang timbul dan harus segera diadakan pemecahan, agar tidak menghambat kegiatan dalam hal menyediakan informasi yang lebih akurat khususnya dalam penanganan masalah pembelian tunai. Karena dalam penanganan pembelian tunai masih dilakukan dengan cara yang manual atau belum terkomputerisasi.

Permasalahan tersebut antara lain :

1. Sering terjadinya selisih pembayaran pada setiap pembelian barang, karena barang yang dibeli tidak sesuai dengan data yang dipesan. Sehingga dapat mengakibatkan kerugian pengeluaran.
2. Penggabungan pencatatan data-data dan laporan pembelian data dalam satu buku menyebabkan data tersebut tidak beraturan.
3. Keterlambatan pengolahan data stok barang yang mengakibatkan pembelian barang dilakukan secara berlebihan.
4. Penyimpanan data yang lebih banyak dalam bentuk arsip, sehingga memerlukan banyak tempat dan mengakibatkan keterlambatan pencarian data yang kemungkinan akan rusak ataupun hilang.

3.2.6. Pemecahan Masalah

Dalam meningkatkan pelayanan serta efisiensi pada PT Nasa Stokist T2309, diperlukan adanya sistem yang mendukung dan mampu untuk mengatasi masalah-masalah tersebut di atas. Maka penulis mengusulkan dengan menggunakan sistem komputerisasi, mengingat keuntungan komputerisasi adalah sebagai berikut :

1. Sistem pembayaran sebaiknya dilaksanakan dengan sistem berbasis komputer, dengan pembuatan program atau sistem.
2. Penanganan dan penyimpanan data akan lebih baik dalam bentuk *database* sehingga tidak merugikan seperti yang tidak diinginkan.
3. Pembuatan *database* yang tepat guna untuk meminimalisasi terjadinya selisih.
4. Penyimpanan data lebih baik menggunakan *database* sehingga memudahkan pencarian data dan menghemat ruang penyimpanan karena hasil pengolahan data tersimpan dalam komputer.

3.3. Analisis Kebutuhan Software

Dalam pembuatan sebuah program desktop dibutuhkannya beberapa kebutuhan software untuk menciptakan sebuah program desktop yang baik dan dapat dimanfaatkan secara optimal, yaitu sebagai berikut:

3.3.1. Analisis Kebutuhan

Analisa kebutuhan *software* untuk program pembelian tunai pada PT Nasa Stokist T 2309 diusulkan dengan beberapa prosedur diantaranya:

A. Bagian Admin

1. *Login*.
2. Mengelola barang.
3. Mengelola *stockist*.
4. Mengelola pembelian.
5. Mencetak bukti pembelian.
6. Mengelola retur pembelian
7. Mencetak bukti retur pembelian

B. Skenario Use Case

1) Skenario Use Case Diagram Menambah Stockist Center

Tabel III.1.
Skenario Use Case Diagram Menambah Stockist Center

Tujuan	Admin dapat menambah data <i>stockist</i>
Deskripsi	Sistem ini memungkinkan aktor menambah <i>stockist</i> .
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i> dan sudah membuka <i>form stockist</i>
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan data <i>stockist</i>	2. Melakukan validasi masukan 3. Menyimpan data <i>stockist</i> ke <i>database</i> 4. Menampilkan pesan berhasil
Skenario Alternatif	
1. Memasukkan data <i>stockist</i>	2. Melakukan validasi masukan 3. Menampilkan pesan peringatan <i>error</i>
4. Memperbaiki masukan yang salah.	5. Melakukan validasi masukan 6. Menyimpan data <i>stockist</i> ke <i>database</i> 7. Menampilkan pesan berhasil
Kondisi Akhir	Sistem dapat menyimpan data <i>stockist</i> baru

2) Skenario Use Case Diagram Mengubah Stockist Center

Tabel III.2.
Skenario Use Case Diagram Mengubah Stockist Center

Tujuan	Admin dapat mengubah data <i>stockist</i>
Deskripsi	Sistem ini memungkinkan aktor mengubah <i>stockist</i> .
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i> dan sudah membuka <i>form stockist</i>

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan nama <i>stockist</i> yang dicari	2. Mencari data <i>stockist</i> yang akan diubah.
4. Melakukan perubahan pada data <i>stockist</i>	3. Menampilkan data <i>stockist</i>
	5. Melakukan validasi masukan
	6. Menyimpan data <i>stockist</i> ke <i>database</i>
	7. Menampilkan pesan berhasil
Skenario Alternatif	
1. Memasukkan nama <i>stockist</i> yang dicari	2. Mencari data <i>stockist</i> yang akan diubah.
4. Melakukan perubahan pada data <i>stockist</i>	3. Menampilkan data <i>stockist</i>
7. Memperbaiki masukan yang salah	5. Melakukan validasi masukan
	6. Menampilkan pesan peringatan bahwa data yang dimasukkan tidak valid
	8. Melakukan validasi masukan
	9. Menyimpan data <i>stockist</i> ke <i>database</i>
	10. Menampilkan pesan berhasil
Kondisi Akhir	Sistem dapat menyimpan data <i>stockist</i> yang telah diubah

3) Skenario *Use Case Diagram* Mencari *Stockist*

Tabel III.3.
Skenario *Use Case Diagram* Mencari *Stockist Center*

Tujuan	Admin dapat mencari data <i>stockist</i>
Deskripsi	Sistem ini memungkinkan aktor mencari <i>stockist</i> .
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i> dan sudah membuka <i>form stockist</i>
Aksi Aktor	Reaksi Sistem
Skenario Normal	

1. Memasukkan nama <i>stockist</i> yang ingin dicari	2. Mencari data <i>stockist</i> di <i>database</i> . 3. Menampilkan daftar <i>stockist</i> yang sesuai.
Kondisi Akhir	Sistem dapat mencari data <i>stockist</i>

4) Skenario *Use Case Diagram* Menghapus *Stockist*

Tabel III.4.
Skenario *Use Case Diagram* Menghapus *Stockist*

Tujuan	Admin dapat menghapus data <i>stockist</i>
Deskripsi	Sistem ini memungkinkan aktor menghapus <i>stockist</i> .
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i> dan sudah membuka <i>form stockist</i>
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan nama <i>stockist</i> yang ingin dicari 3. Memilih data <i>stockist</i> 5. Memilih jawaban YA	2. Mencari data <i>stockist</i> yang akan diubah. 4. Menampilkan konfirmasi hapus 6. Menghapus data <i>stockist</i> dari <i>database</i> 7. Menampilkan pesan berhasil dihapus
Kondisi Akhir	Sistem dapat menghapus data <i>stockist</i>

5) Skenario *Use Case Diagram* Menambah Barang

Tabel III.5.
Skenario *Use Case Diagram* Menambah Barang

Tujuan	Admin dapat menambah data barang
Deskripsi	Sistem ini memungkinkan aktor menambah barang.
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i> dan sudah membuka <i>form</i> barang
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan data barang	2. Melakukan validasi masukan 3. Menyimpan data barang ke <i>database</i> 4. Menampilkan pesan berhasil
Skenario Alternatif	
1. Memasukkan data barang	2. Melakukan validasi masukan 3. Menampilkan pesan peringatan <i>error</i>
4. Memperbaiki masukan yang salah.	5. Melakukan validasi masukan 6. Menyimpan data barang ke <i>database</i> 7. Menampilkan pesan berhasil
Kondisi Akhir	Sistem dapat menyimpan data barang baru

6) Skenario *Use Case Diagram* Mengubah Barang

Tabel III.6.
Skenario *Use Case Diagram* Mengubah Barang

Tujuan	Admin dapat mengubah data barang
Deskripsi	Sistem ini memungkinkan aktor mengubah barang.
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i> dan sudah membuka <i>form</i> barang

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan nama barang yang dicari	2. Mencari data barang yang akan diubah. 3. Menampilkan data barang
4. Melakukan perubahan pada data barang	5. Melakukan validasi masukan 6. Menyimpan data barang ke <i>database</i> 7. Menampilkan pesan berhasil
Skenario Alternatif	
1. Memasukkan nama barang yang dicari	2. Mencari data barang yang akan diubah. 3. Menampilkan data barang
4. Melakukan perubahan pada data barang	5. Melakukan validasi masukan 6. Menampilkan pesan peringatan bahwa data yang dimasukkan tidak valid
7. Memperbaiki masukan yang salah	8. Melakukan validasi masukan 9. Menyimpan data barang ke <i>database</i> 10. Menampilkan pesan berhasil
Kondisi Akhir	Sistem dapat menyimpan data barang yang telah diubah

7) Skenario *Use Case Diagram* Mencari Barang

Tabel III.7.
Skenario *Use Case Diagram* Mencari Barang

Tujuan	Admin dapat mencari data barang
Deskripsi	Sistem ini memungkinkan aktor mencari barang.
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i> dan sudah membuka <i>form</i> barang

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan nama barang yang ingin dicari	2. Mencari data barang di <i>database</i> . 3. Menampilkan daftar barang yang sesuai.
Kondisi Akhir	Sistem dapat mencari data barang

8) Skenario *Use Case Diagram* Menghapus Barang

Tabel III.8.
Skenario *Use Case Diagram* Menghapus Barang

Tujuan	Admin dapat menghapus data barang
Deskripsi	Sistem ini memungkinkan aktor menghapus barang.
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i> dan sudah membuka <i>form</i> barang
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan nama barang yang ingin dicari 3. Memilih data barang 5. Memilih jawaban YA	2. Mencari data barang yang akan diubah. 4. Menampilkan konfirmasi hapus 6. Menghapus data barang dari <i>database</i> 7. Menampilkan pesan berhasil dihapus
Kondisi Akhir	Sistem dapat menghapus data barang

9) Skenario *Use Case Diagram* Menambah Pembelian

Tabel III.9.
Skenario *Use Case Diagram* Menambah Pembelian

Tujuan	Admin dapat menambah pembelian
Deskripsi	Sistem ini memungkinkan kasir menambah pembelian
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i>
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan data pembelian 2. Memasukkan data barang yang dibeli	3. Melakukan validasi masukan 4. Menyimpan data barang ke <i>database</i> 5. Menampilkan pesan berhasil
Skenario Alternatif	
1. Memasukkan data pembelian 2. Memasukkan data barang yang dibeli 5. Memperbaiki masukan yang salah	3. Melakukan validasi masukan 4. Menampilkan pesan peringatan <i>error</i> 6. Melakukan validasi masukan 7. Menyimpan data barang ke <i>database</i> 8. Menampilkan pesan berhasil

10) Skenario *Use Case Diagram* Mencetak Bukti Pembelian

Tabel III.10.
Skenario *Use Case Diagram* Mencetak Bukti Pembelian

Tujuan	Admin dapat mencetak bukti pembelian
Deskripsi	Sistem ini memungkinkan kasir mencetak bukti pembelian
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i>

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih data pembelian 2. Klik Cetak	3. Menampilkan data pembelian yang akan dicetak

11) Skenario *Use Case Diagram* Menambah Retur Pembelian

Tabel III.11.
Skenario *Use Case Diagram* Menambah Retur Pembelian

Tujuan	Admin dapat menambah retur pembelian
Deskripsi	Sistem ini memungkinkan kasir menambah retur pembelian
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i>
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan data retur pembelian 2. Memasukkan data barang yang dibeli	3. Melakukan validasi masukan 4. Menyimpan data barang ke <i>database</i> 5. Menampilkan pesan berhasil
Skenario Alternatif	
1. Memasukkan data retur pembelian 2. Memasukkan data barang yang dibeli 5. Memperbaiki masukan yang salah	3. Melakukan validasi masukan 4. Menampilkan pesan peringatan <i>error</i> 6. Melakukan validasi masukan 7. Menyimpan data barang ke <i>database</i> 8. Menampilkan pesan berhasil

12) Skenario *Use Case Diagram* Mencetak Bukti Retur Pembelian

Tabel III.12.
Skenario *Use Case Diagram* Mencetak Bukti Retur Pembelian

Tujuan	Admin dapat mencetak bukti retur pembelian
Deskripsi	Sistem ini memungkinkan kasir mencetak bukti retur pembelian
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i>
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pilih data retur pembelian 2. Klik Cetak	3. Menampilkan data retur pembelian yang akan dicetak

13) Skenario *Use Case Diagram* Menambah Akun

Tabel III.13.
Skenario *Use Case Diagram* Menambah Akun

Tujuan	Admin dapat menambah data akun
Deskripsi	Sistem ini memungkinkan aktor menambah akun.
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i> dan sudah membuka <i>form</i> akun
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan data akun	2. Melakukan validasi masukan 3. Menyimpan data akun ke <i>database</i> 4. Menampilkan pesan berhasil
Skenario Alternatif	
1. Memasukkan data akun	2. Melakukan validasi masukan 3. Menampilkan pesan peringatan <i>error</i>

4. Memperbaiki masukan yang salah.	5. Melakukan validasi masukan 6. Menyimpan data akun ke <i>database</i> 7. Menampilkan pesan berhasil
Kondisi Akhir	Sistem dapat menyimpan data akun baru

14) Skenario *Use Case Diagram* Mengubah Akun

Tabel III.14.
Skenario *Use Case Diagram* Mengubah Akun

Tujuan	Admin dapat mengubah data akun
Deskripsi	Sistem ini memungkinkan aktor mengubah akun.
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i> dan sudah membuka <i>form</i> akun
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan nama akun yang dicari	2. Mencari data akun yang akan diubah. 3. Menampilkan data akun
4. Melakukan perubahan pada data akun	5. Melakukan validasi masukan 6. Menyimpan data akun ke <i>database</i> 7. Menampilkan pesan berhasil
Skenario Alternatif	
1. Memasukkan nama akun yang dicari	2. Mencari data akun yang akan diubah. 3. Menampilkan data akun
4. Melakukan perubahan pada data akun	5. Melakukan validasi masukan 6. Menampilkan pesan peringatan bahwa data yang dimasukkan tidak valid

7. Memperbaiki masukan yang salah	8. Melakukan validasi masukan 9. Menyimpan data akun ke <i>database</i> 10. Menampilkan pesan berhasil
Kondisi Akhir	Sistem dapat menyimpan data akun yang telah diubah

15) Skenario *Use Case Diagram* Mencari Akun

Tabel III.15.
Skenario *Use Case Diagram* Mencari Akun

Tujuan	Admin dapat mencari data akun
Deskripsi	Sistem ini memungkinkan aktor mencari akun.
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i> dan sudah membuka <i>form</i> akun
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan nama akun yang ingin dicari	2. Mencari data akun di <i>database</i> . 3. Menampilkan daftar akun yang sesuai.
Kondisi Akhir	Sistem dapat mencari data akun

16) Skenario *Use Case Diagram* Menghapus Akun

Tabel III.16.
Skenario *Use Case Diagram* Menghapus Akun

Tujuan	Admin dapat menghapus data akun
Deskripsi	Sistem ini memungkinkan aktor menghapus akun.
Aktor	Admin

Kondisi Awal	Admin sudah <i>login</i> dan sudah membuka <i>form</i> akun
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan nama akun yang ingin dicari 3. Memilih data akun 5. Memilih jawaban YA	2. Mencari data akun yang akan diubah. 4. Menampilkan konfirmasi hapus 6. Menghapus data akun dari <i>database</i> 7. Menampilkan pesan berhasil dihapus
Kondisi Akhir	Sistem dapat menghapus data akun

17) Skenario *Use Case Diagram* Menambah Jurnal

Tabel III.17.
Skenario *Use Case Diagram* Menambah Jurnal

Tujuan	Admin dapat menambah jurnal
Deskripsi	Sistem ini memungkinkan kasir menambah jurnal
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i>
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan data jurnal 2. Memasukkan data akun yang dibeli	3. Melakukan validasi masukan 4. Menyimpan data akun ke <i>database</i> 5. Menampilkan pesan berhasil

Skenario Alternatif	
1. Memasukkan data jurnal 2. Memasukkan data akun yang dibeli 5. Memperbaiki masukan yang salah	3. Melakukan validasi masukan 4. Menampilkan pesan peringatan <i>error</i> 6. Melakukan validasi masukan 7. Menyimpan data akun ke <i>database</i> 8. Menampilkan pesan berhasil

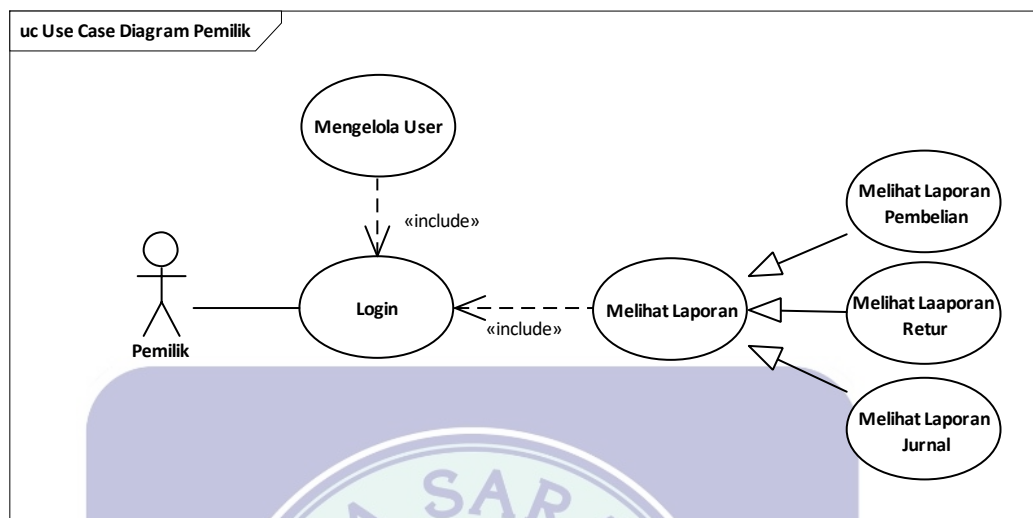
18) Skenario *Use Case Diagram* Melihat Laporan

Tabel III.18.
Deskripsi *Use Case Diagram* Melihat Laporan

Tujuan	Admin dapat melihat laporan
Deskripsi	Sistem ini memungkinkan admin melihat laporan yang dicari
Aktor	Admin
Kondisi Awal	Admin sudah <i>login</i>
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan periode laporan 2. Klik Lihat	3. Menampilkan laporan yang diminta

2. Use Case Diagram Pemilik

A. Gambar Use Case Diagram



Gambar III.6.
Use Case Diagram Pemilik

B. Skenario Use Case

1) Skenario Use Case Diagram Menambah User

Tabel III.19.
Skenario Use Case Diagram Menambah User

Tujuan	Pemilik dapat menambah data <i>user</i>
Deskripsi	Sistem ini memungkinkan aktor menambah <i>user</i> .
Aktor	Pemilik
Kondisi Awal	Pemilik sudah <i>login</i> dan sudah membuka <i>form user</i>
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan data <i>user</i>	2. Melakukan validasi masukan 3. Menyimpan data <i>user</i> ke <i>database</i> 4. Menampilkan pesan berhasil
Skenario Alternatif	
1. Memasukkan data <i>user</i>	2. Melakukan validasi masukan 3. Menampilkan pesan peringatan <i>error</i>

4. Memperbaiki masukan yang salah.	5. Melakukan validasi masukan 6. Menyimpan data <i>user</i> ke <i>database</i> 7. Menampilkan pesan berhasil
Kondisi Akhir	Sistem dapat menyimpan data <i>user</i> baru

2) Skenario *Use Case Diagram* Mengubah *User*

Tabel III.20.
Skenario *Use Case Diagram* Mengubah *User*

Tujuan	Pemilik dapat mengubah data <i>user</i>
Deskripsi	Sistem ini memungkinkan aktor mengubah <i>user</i> .
Aktor	Pemilik
Kondisi Awal	Pemilik sudah <i>login</i> dan sudah membuka <i>form user</i>
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan nama <i>user</i> yang dicari	2. Mencari data <i>user</i> yang akan diubah. 3. Menampilkan data <i>user</i>
4. Melakukan perubahan pada data <i>user</i>	5. Melakukan validasi masukan 6. Menyimpan data <i>user</i> ke <i>database</i> 7. Menampilkan pesan berhasil
Skenario Alternatif	
1. Memasukkan nama <i>user</i> yang dicari	2. Mencari data <i>user</i> yang akan diubah. 3. Menampilkan data <i>user</i>
4. Melakukan perubahan pada data <i>user</i>	5. Melakukan validasi masukan 6. Menampilkan pesan peringatan bahwa data yang dimasukkan tidak valid
7. Memperbaiki masukan yang salah	8. Melakukan validasi masukan 9. Menyimpan data <i>user</i> ke <i>database</i> 10. Menampilkan pesan berhasil

Kondisi Akhir	Sistem dapat menyimpan data <i>user</i> yang telah diubah
---------------	---

3) Skenario *Use Case Diagram* Mencari *User*

Tabel III.21.
Skenario *Use Case Diagram* Mencari *User*

Tujuan	Pemilik dapat mencari data <i>user</i>
Deskripsi	Sistem ini memungkinkan aktor mencari <i>user</i> .
Aktor	Pemilik
Kondisi Awal	Pemilik sudah <i>login</i> dan sudah membuka <i>form user</i>
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan nama <i>user</i> yang ingin dicari	2. Mencari data <i>user</i> di <i>database</i> . 3. Menampilkan daftar <i>user</i> yang sesuai.
Kondisi Akhir	Sistem dapat mencari data <i>user</i>

4) Skenario *Use Case Diagram* Menghapus *User*

Tabel III.22.
Skenario *Use Case Diagram* Menghapus *User*

Tujuan	Pemilik dapat menghapus data <i>user</i>
Deskripsi	Sistem ini memungkinkan aktor menghapus <i>user</i> .
Aktor	Pemilik
Kondisi Awal	Pemilik sudah <i>login</i> dan sudah membuka <i>form user</i>

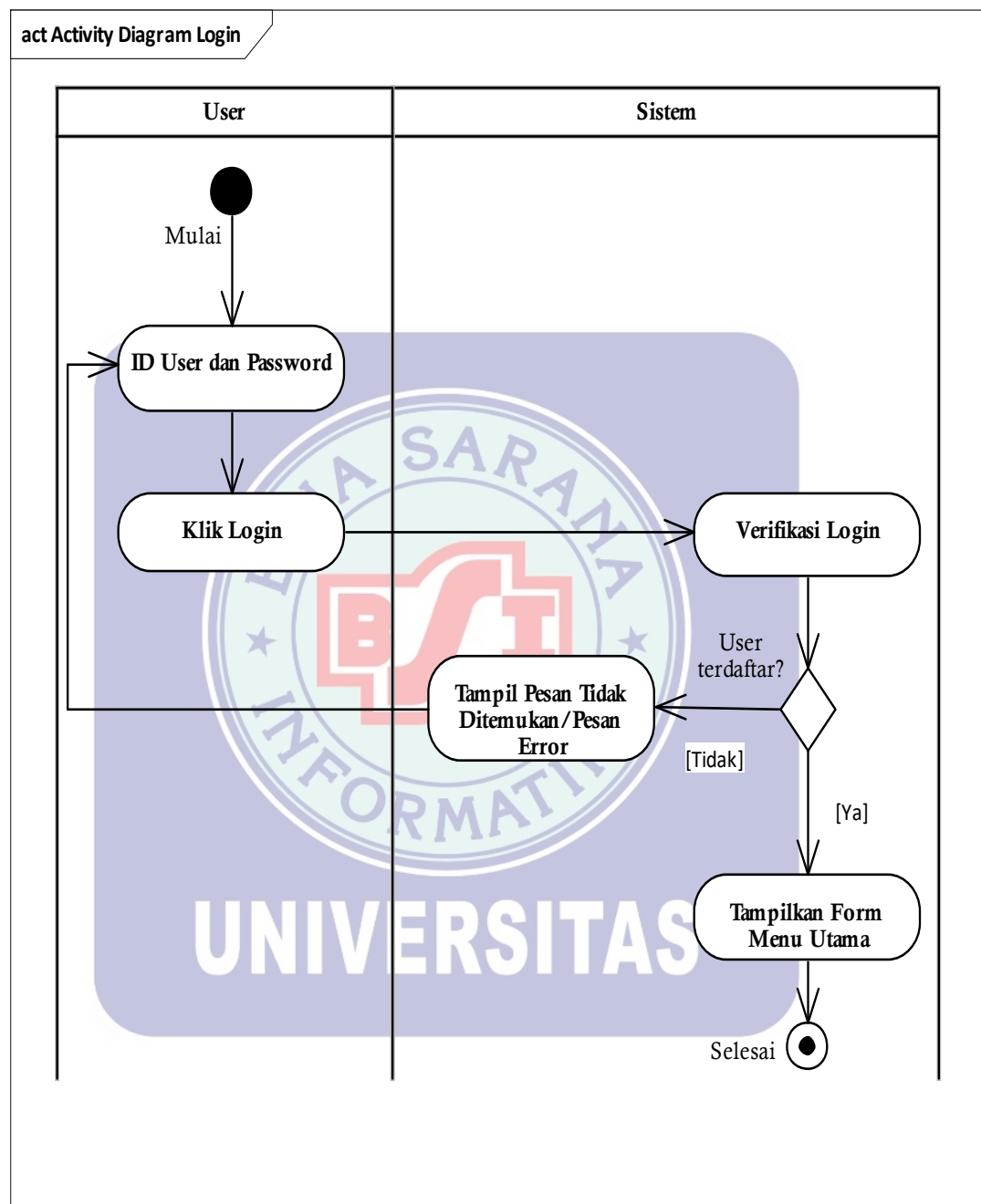
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan nama <i>user</i> yang ingin dicari	2. Mencari data <i>user</i> yang akan diubah.
3. Memilih data <i>user</i>	4. Menampilkan konfirmasi hapus
5. Memilih jawaban YA	6. Menghapus data <i>user</i> dari <i>database</i> 7. Menampilkan pesan berhasil dihapus
Kondisi Akhir	Sistem dapat menghapus data <i>user</i>

Tabel III.23.
Deskripsi Use Case Diagram Melihat Laporan

Tujuan	Pemilik dapat melihat laporan
Deskripsi	Sistem ini memungkinkan pemilik melihat laporan yang dicari
Aktor	Pemilik
Kondisi Awal	Pemilik sudah <i>login</i>
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan periode laporan 2. Klik Lihat	3. Menampilkan laporan yang diminta

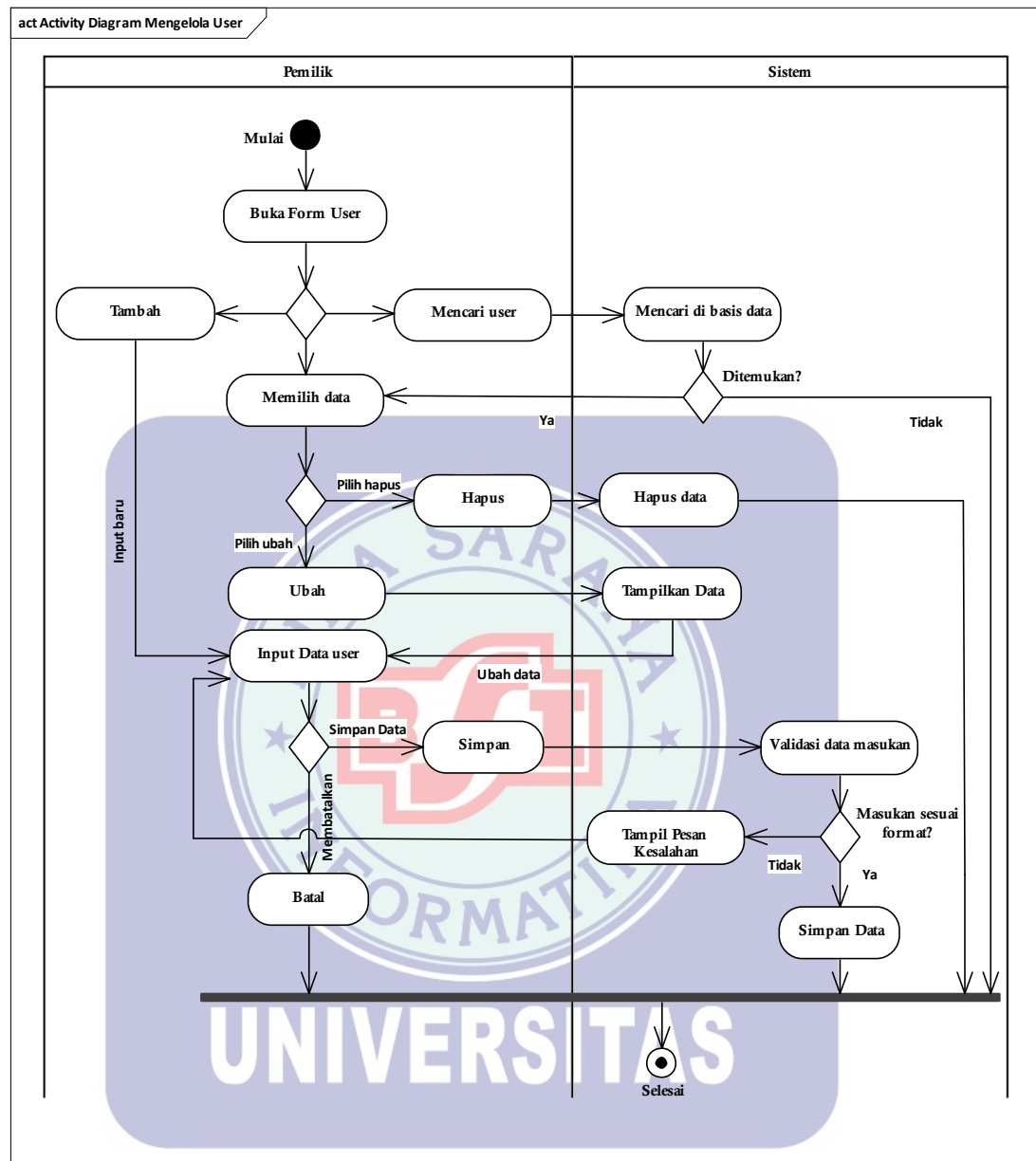
3.3.3. Activity Diagram

1. Activity Diagram Login



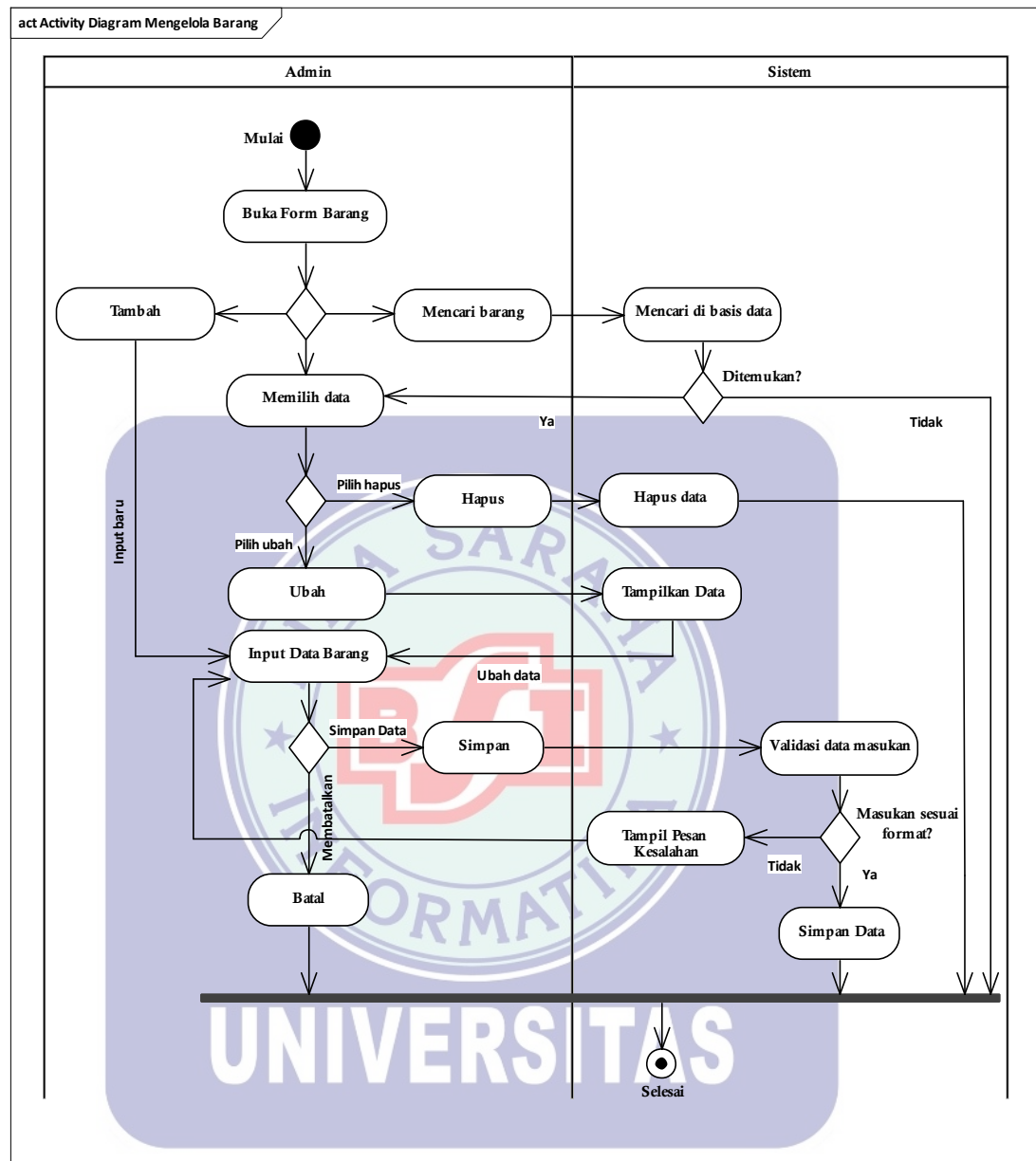
Gambar III.7.
Activity Diagram Login

2. Activity Diagram Mengelola User (Pemilik)



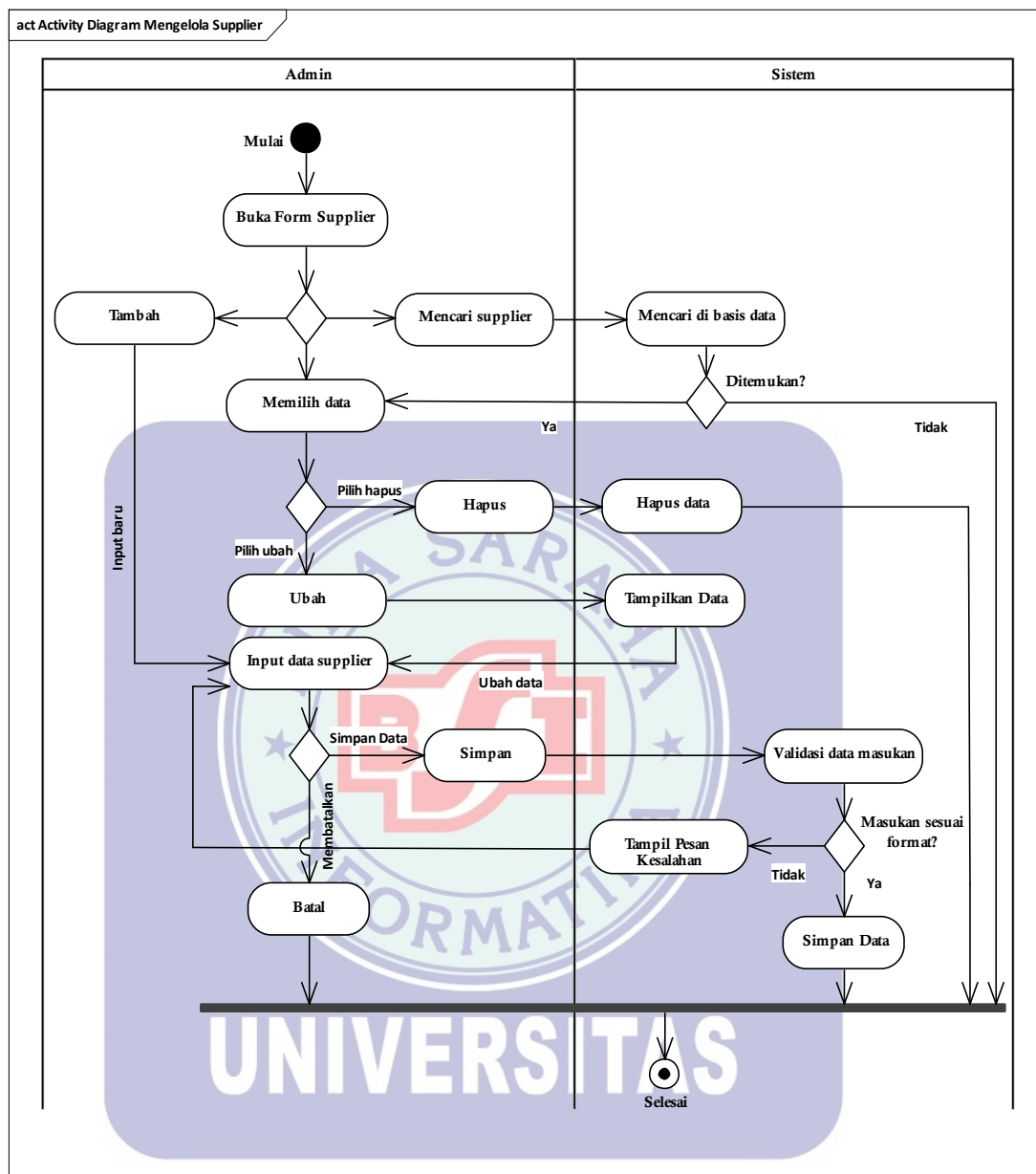
Gambar III.8.
Activity Diagram Mengelola User

3. Activity Diagram Mengelola Barang (Admin)



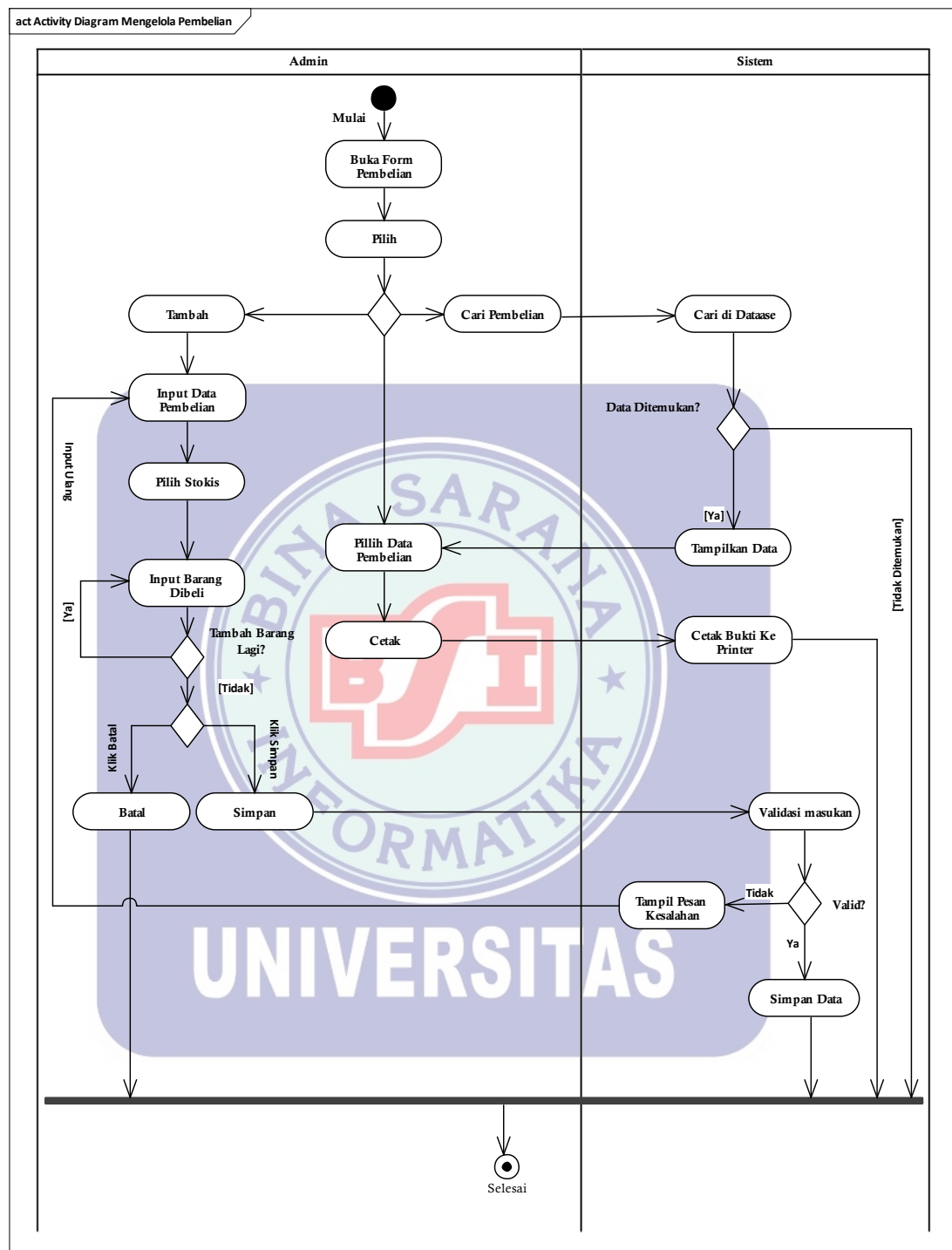
Gambar III.9.
Activity Diagram Mengelola Barang

4. Activity Diagram Mengelola Stockist Center (Admin)



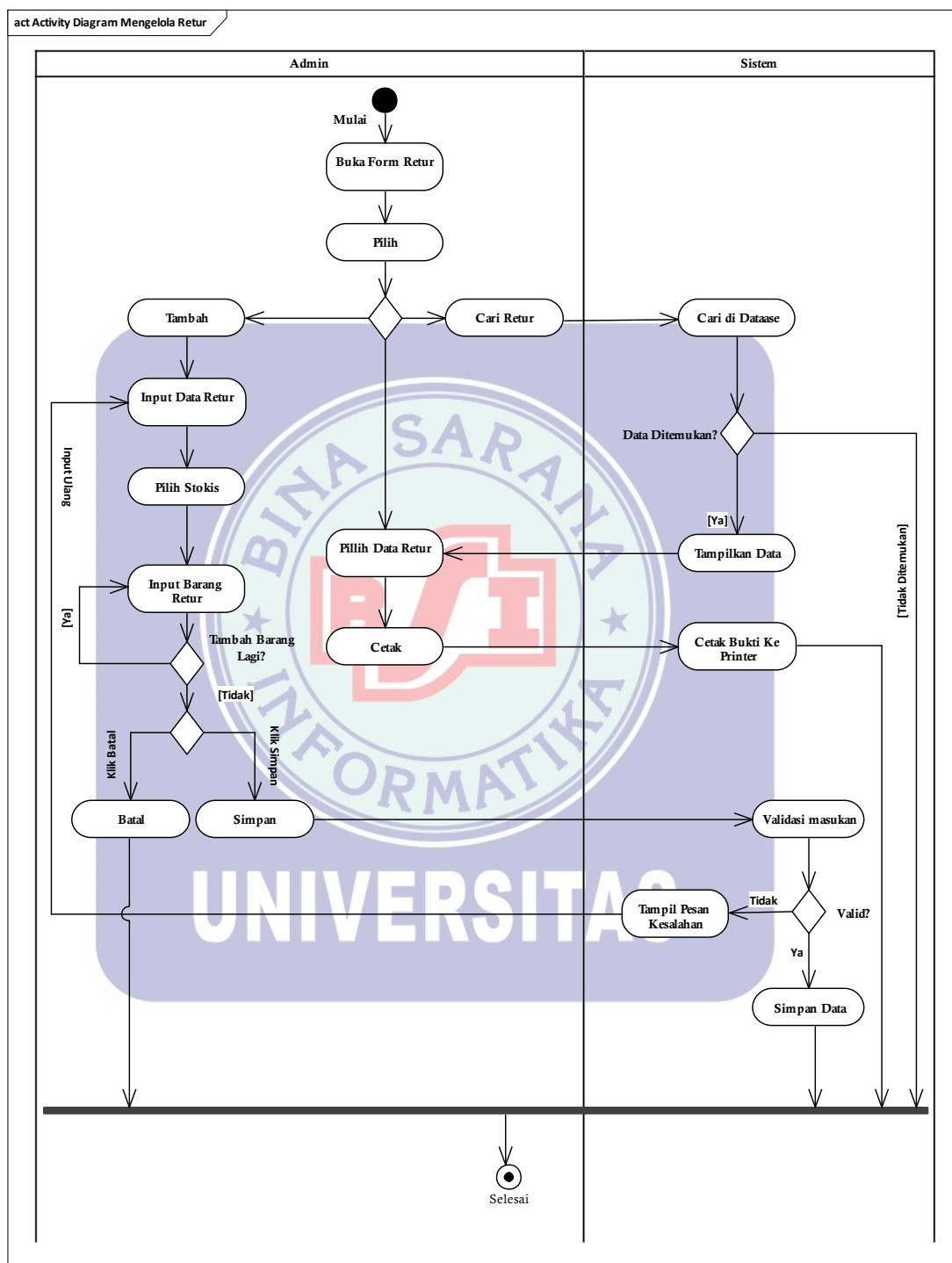
Gambar III.10.
Activity Diagram Mengelola Stockist Center

5. Activity Diagram Mengelola Pembelian dan Mencetak Bukti Pembelian (Admin)



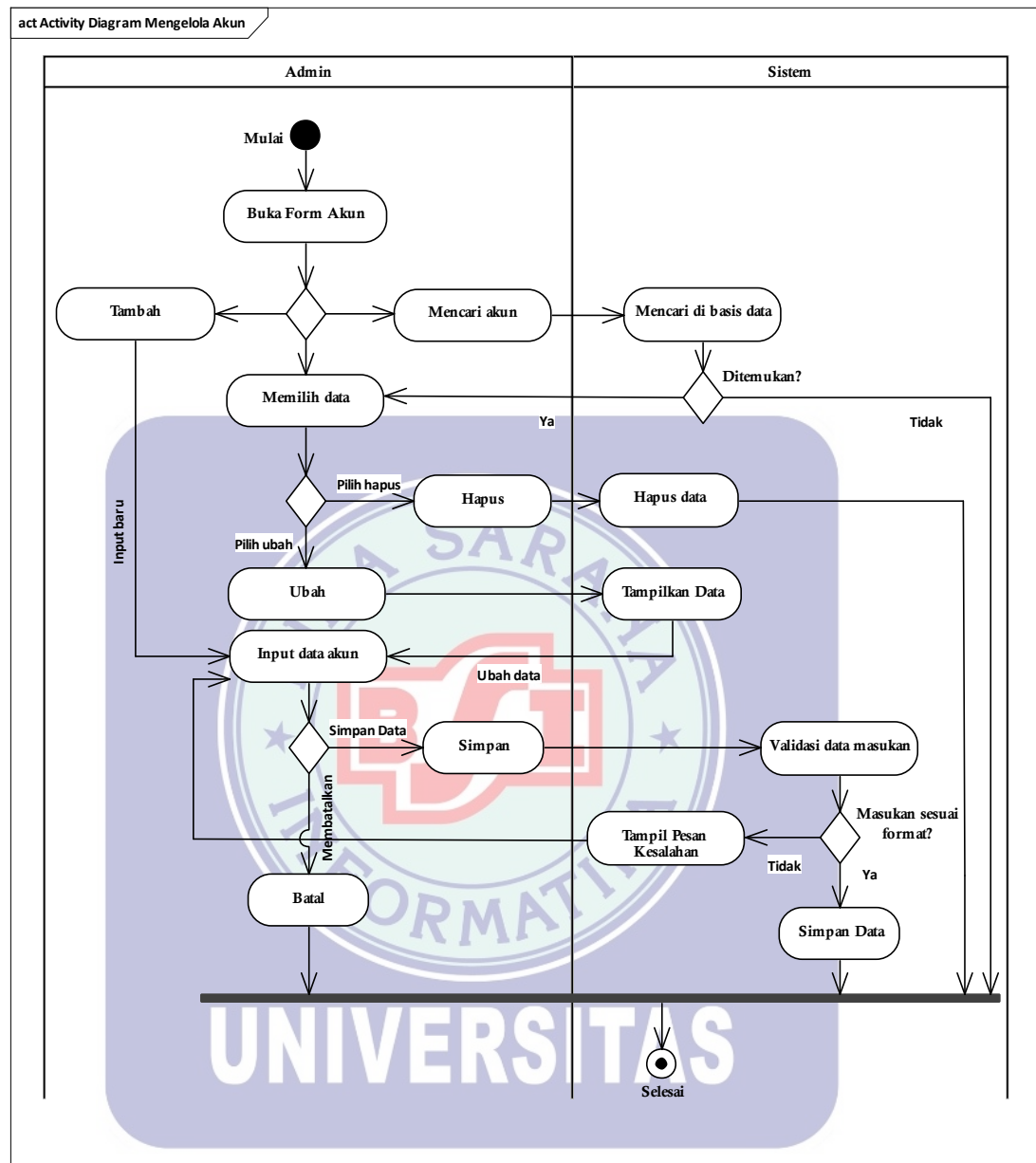
Gambar III.11.
Activity Diagram Mengelola Pembelian

6. *Activity Diagram Mengelola Retur Pembelian dan Cetak Bukti Retur Pembelian*
(Admin)



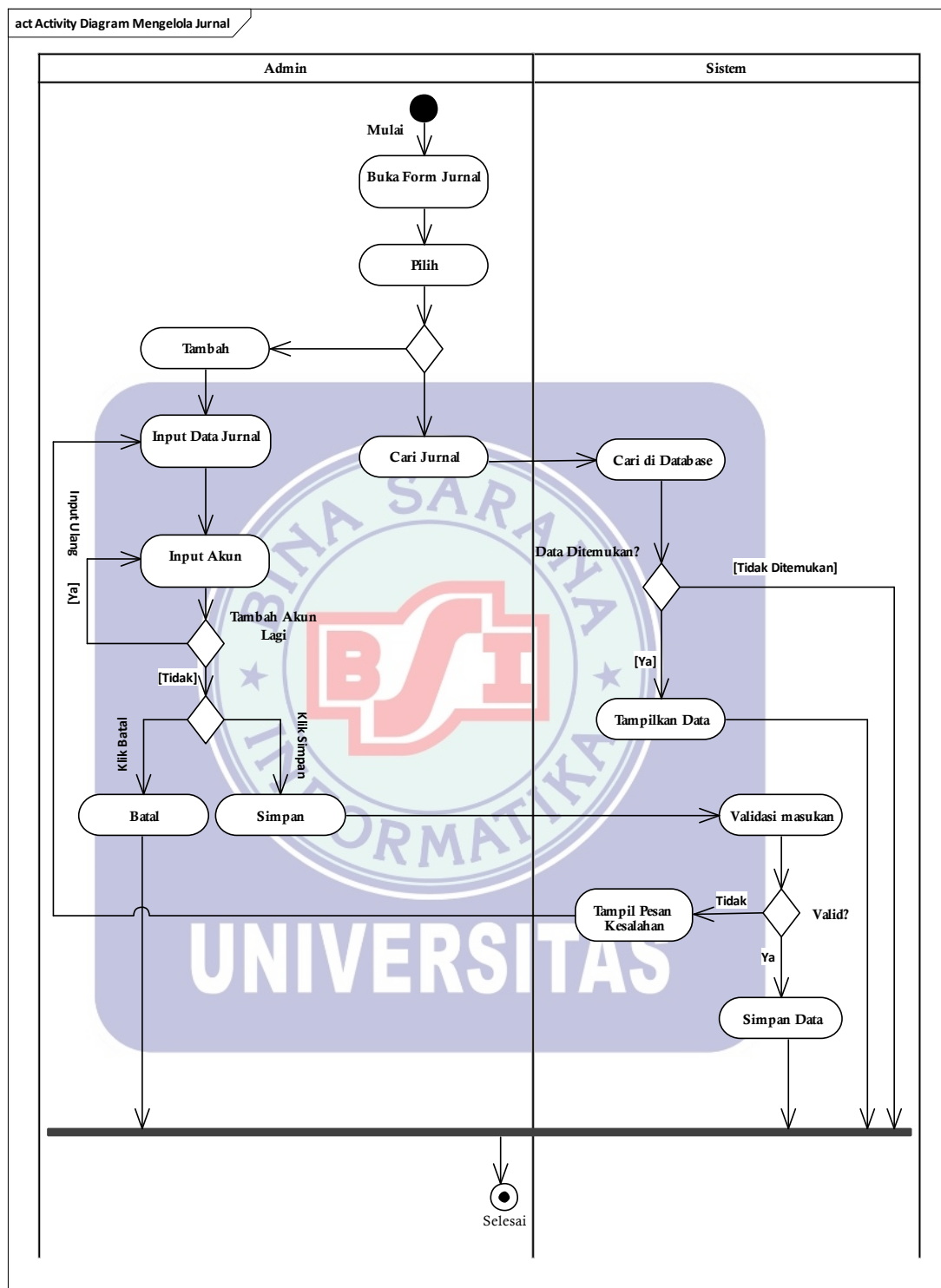
Gambar III.12.
Activity Diagram Mengelola Retur Pembelian

7. Activity Diagram Mengelola Akun (Admin)

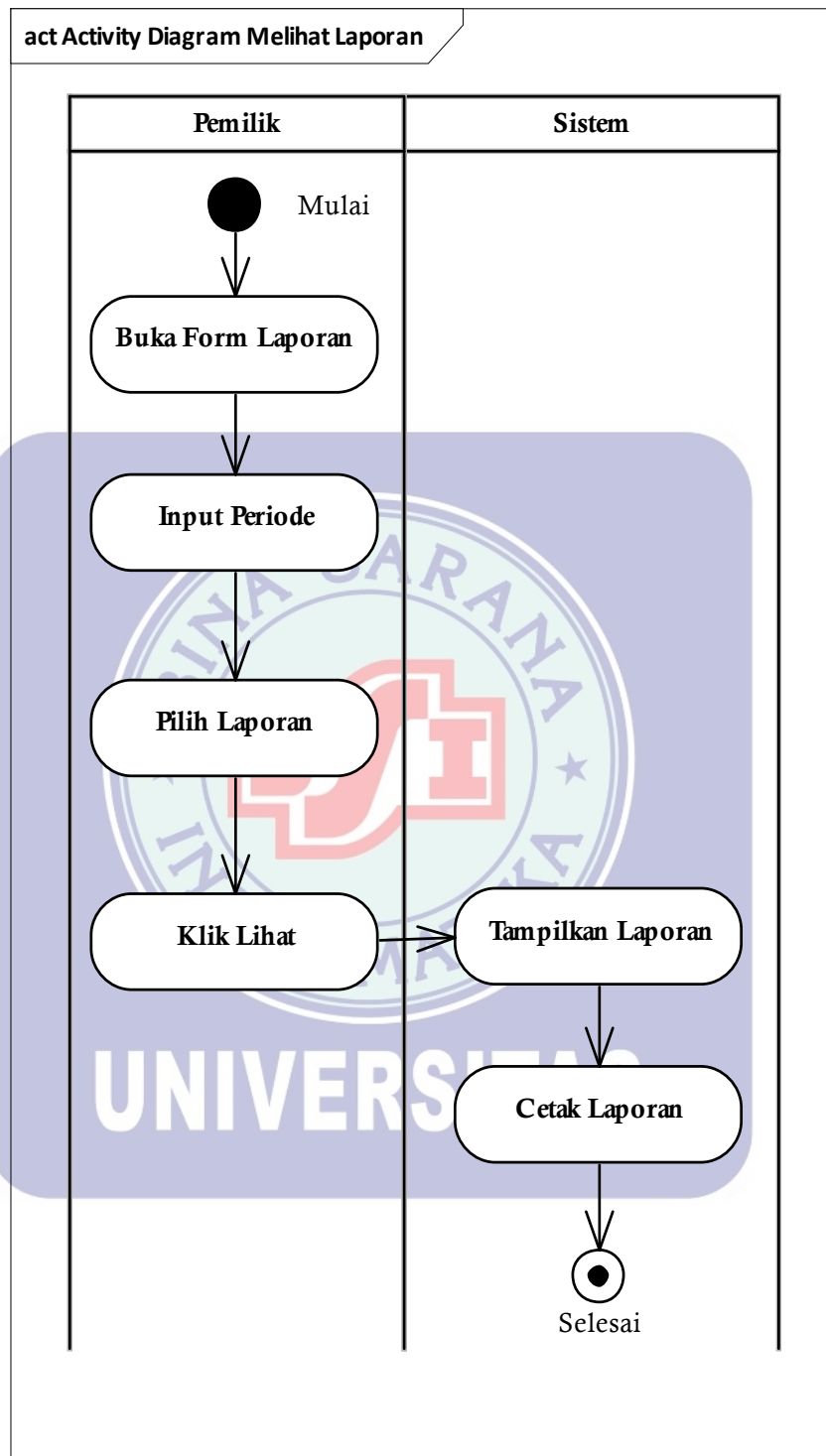


Gambar III.13.
Activity Diagram Mengelola Akun

8. Activity Diagram Mengelola Jurnal (Admin)



Gambar III.14.
Activity Diagram Mengelola Jurnal

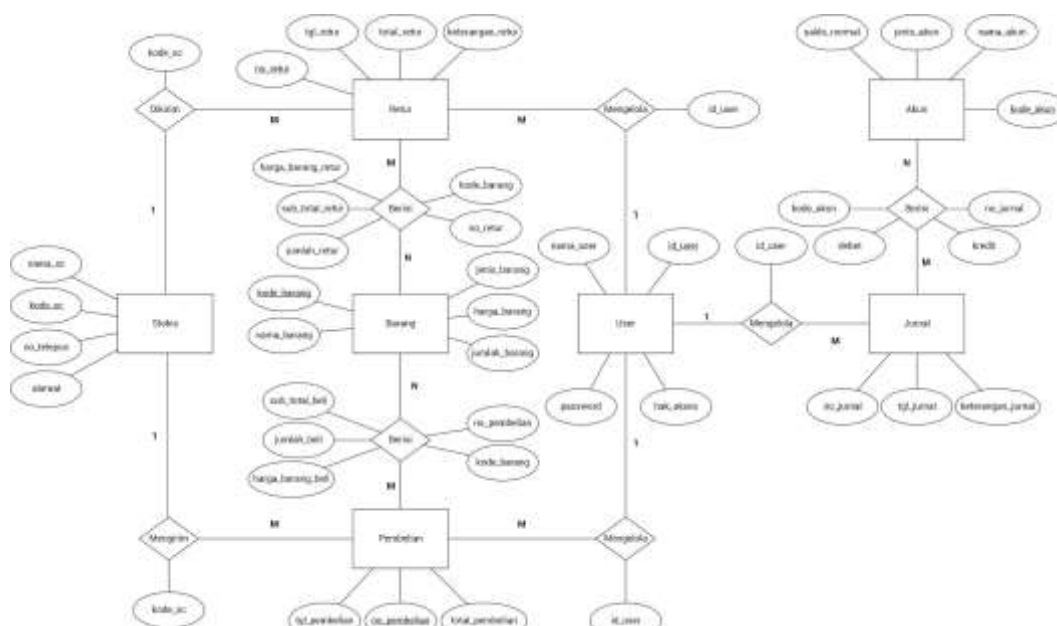
9. *Activity Diagram* Melihat Laporan (Admin dan Pemilik)

Keterangan : Semua laporan berada disatu *form* yang sama

Gambar III.15.
Activity Diagram Melihat Laporan

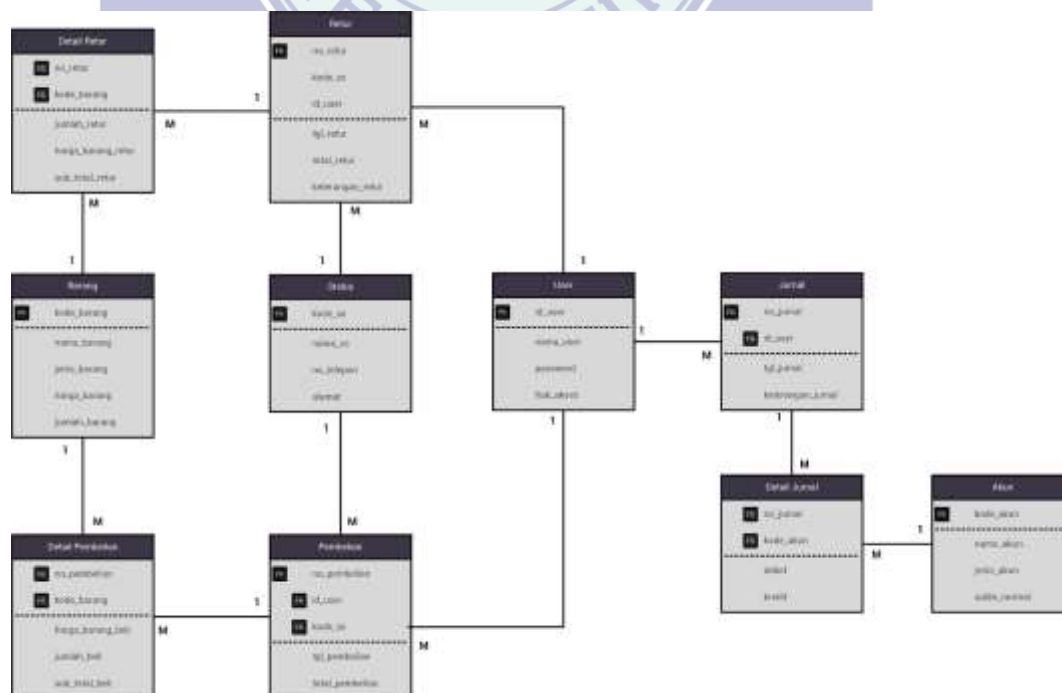
3.4. Desain

3.4.1. Entity Relationship Diagram (ERD)



Gambar III.16.
Entity Relationship Diagram (ERD)

3.4.2. Logical Record Structure (LRS)



Gambar III.17.
Logical Record Structure (LRS)

3.4.3. Spesifikasi File

1. Spesifikasi File User

Nama Database : stokis_nasa
 Nama File : user.frm
 Akronim : user
 Fungsi : Menyimpan data user
 Tipe File : File Master
 Organisasi File : Index Suquential
 Akses File : Random
 Media : Hardisk
 Panjang Record : 87 Karakter
 Kunci Field : id_user
 Software: : MySQL

Tabel III.24.
Spesifikasi File User

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1.	Id User	id_user	CHAR	20	Primary Key
2.	Nama User	nama_user	VARCHAR	25	
3	Password	password	VARCHAR	35	
4	Hak akses	hak_akses	VARCHAR	7	

2. Spesifikasi File Barang

Nama Database : stokis_nasa
 Nama File : barang.frm
 Akronim : barang
 Fungsi : Menyimpan data barang
 Tipe File : File Master
 Organisasi File : Index Suquential

Akses File : Random
 Media : Hardisk
 Panjang Record : 64 Karakter
 Kunci Field : kode_barang
 Software: : MySQL

Tabel III.25.
Spesifikasi File Barang

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1.	Kode Barang	kode_barang	CHAR	5	Primary Key
2.	Nama Barang	nama_barang	VARCHAR	25	
3	Jenis Barang	jenis_barang	VARCHAR	20	
4	Harga Barang	harga_barang	INT	11	
5	Jumlah Barang	jumlah_barang	INT	3	

3. Spesifikasi File Stokis

Nama Database : stokis_nasa
 Nama File : stokis_center.frm
 Akronim : stokis_center
 Fungsi : Menyimpan data stokis
 Tipe File : File Master
 Organisasi File : Index Suquential
 Akses File : Random
 Media : Hardisk
 Panjang Record : 73 Karakter
 Kunci Field : kode_sc
 Software: : MySQL

Tabel III.26.
Spesifikasi File Stokis Center

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1.	Kode SC	kode_sc	CHAR	6	Primary Key
2.	Nama SC	nama_sc	VARCHAR	25	
3	No Telp	no_telp	VARCHAR	12	
4	Alamat	alamat	VARCHAR	30	

4. Spesifikasi File Akun

Nama Database : stokis_nasa

Nama File : akun.frm

Akronim : akun

Fungsi : Menyimpan data akun

Tipe File : File Master

Organisasi File : Index Suquential

Akses File : Random

Media : Hardisk

Panjang Record : 56 Karakter

Kunci Field : kode_akun

Software: : MySQL

Tabel III.27.
Spesifikasi File Akun

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1.	Kode Akun	kode_akun	CHAR	5	Primary Key
2.	Nama Akun	nama_akun	VARCHAR	25	
3	Jenis Akun	jenis_akun	VARCHAR	20	
4	Saldo Normal	saldo_normal	VARCHAR	6	

5. Spesifikasi *File* Pembelian

Nama *Database* : stokis_nasa
 Nama *File* : pembelian.frm
 Akronim : pembelian
 Fungsi : Menyimpan data pembelian
 Tipe *File* : *File* Transaksi
 Organisasi *File* : *Index Suquential*
 Akses *File* : *Random*
 Media : *Hardisk*
 Panjang *Record* : 45 Karakter
 Kunci *Field* : no_pembelian
 Software: : MySQL

Tabel III.28.
Spesifikasi *File* Pembelian

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1.	No Pembelian	no_pembelian	CHAR	7	Primary Key
2.	Id User	id_user	VARCHAR	20	Foreign Key
3	Kode SC	kode_sc	CHAR	7	Foreign Key
4	Tanggal Pembelian	tgl_pembelian	DATE		
5	Total	total_pembelian	INT	11	

6. Spesifikasi *File* Detail Pembelian

Nama *Database* : stokis_nasa
 Nama *File* : detail_pembelian.frm
 Akronim : detail_pembelian
 Fungsi : Menyimpan data detail pembelian
 Tipe *File* : *File* Transaksi
 Organisasi *File* : *Index Suquential*

Akses File : Random
 Media : Hardisk
 Panjang Record : 33 Karakter
 Kunci Field : -
 Software: : MySQL

Tabel III.29.
Spesifikasi File Detail Pembelian

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1.	No Pembelian	no_pembelian	CHAR	7	Foreign Key
2.	Kode Barang	kode_barang	CHAR	5	Foreign Key
3	Harga Barang	harga_barang_beli	INT	7	
4	Jumlah Beli	jumlah_beli	INT	3	
5	Sub Total	sub_total_beli	INT	11	

7. Spesifikasi File Retur

Nama Database : stokis_nasa
 Nama File : retur.frm
 Akronim : retur
 Fungsi : Menyimpan data retur
 Tipe File : File Transaksi
 Organisasi File : Index Suquential
 Akses File : Random
 Media : Hardisk
 Panjang Record : 145 Karakter
 Kunci Field : no_retur
 Software: : MySQL

Tabel III.30.
Spesifikasi *File* Retur

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1.	No Retur	no_retur	CHAR	7	Primary Key
2.	Kode SC	kode_sc	CHAR	7	Foreign Key
3	Id User	id_user	VARCHAR	20	Foreign Key
4	Tanggal Retur	tgl_retur	DATE		
5	Total	total_retur	INT	11	
6	Keterangan Retur	keterangan_retur	VARCHAR	100	

8. Spesifikasi *File* Detail Retur

Nama Database : stokis_nasa

Nama File : detail_retur.frm

Akronim : detail_retur

Fungsi : Menyimpan data detail retur

Tipe File : File Transaksi

Organisasi File : Index Suquential

Akses File : Random

Media : Hardisk

Panjang Record : 33 Karakter

Kunci Field : -

Software: : MySQL

Tabel III.31.
Spesifikasi *File* Detail Retur

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1.	No Retur	no_retur	CHAR	7	Foreign Key
2.	Kode Barang	kode_barang	CHAR	5	Foreign Key
3	Harga Barang	harga_barang_retur	INT	7	
4	Jumlah Retur	jumlah_retur	INT	3	
5	Sub Total	sub_total_retur	INT	11	

9. Spesifikasi *File* Jurnal

Nama *Database* : stokis_nasa
 Nama *File* : jurnal.frm
 Akronim : jurnal
 Fungsi : Menyimpan data jurnal
 Tipe *File* : *File* Transaksi
 Organisasi *File* : *Index Suquential*
 Akses *File* : *Random*
 Media : *Hardisk*
 Panjang *Record* : 127 Karakter
 Kunci *Field* : no_jurnal
 Software: : MySQL

Tabel III.32.
Spesifikasi *File* Jurnal

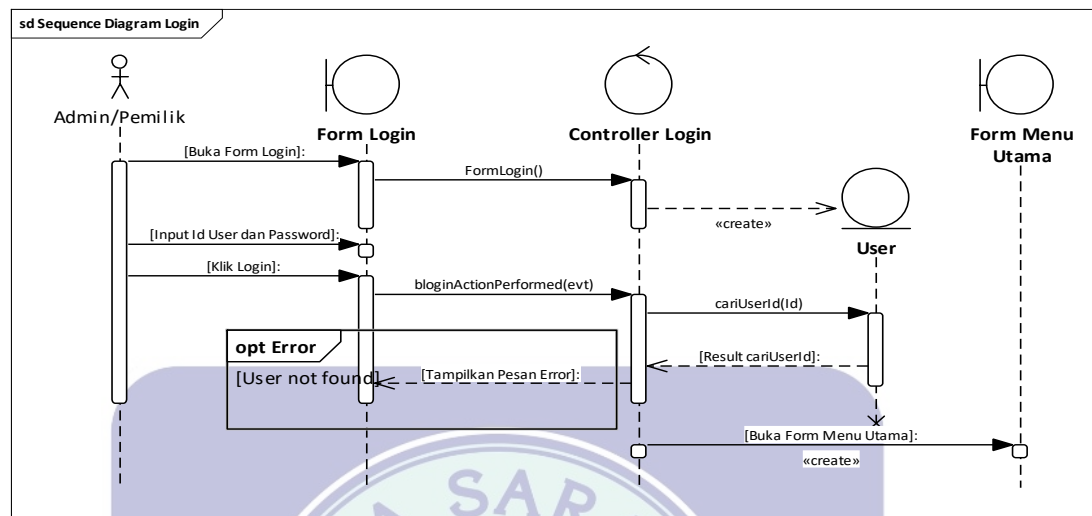
No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1.	No Jurnal	no_jurnal	CHAR	7	Primary Key
2.	Id User	id_user	VARCHAR	20	Foreign Key
3	Tanggal Uurnal	tgl_jurnal	DATE		
4	Keterangan Jurnal	keterangan_jurnal	VARCHAR	100	

10. Spesifikasi *File* Detail Jurnal

Nama *Database* : stokis_nasa
 Nama *File* : detail_jurnal.frm
 Akronim : detail_jurnal
 Fungsi : Menyimpan data detail_jurnal
 Tipe *File* : *File* Transaksi
 Organisasi *File* : *Index Suquential*
 Akses *File* : *Random*

3.4.5. Sequence Diagram

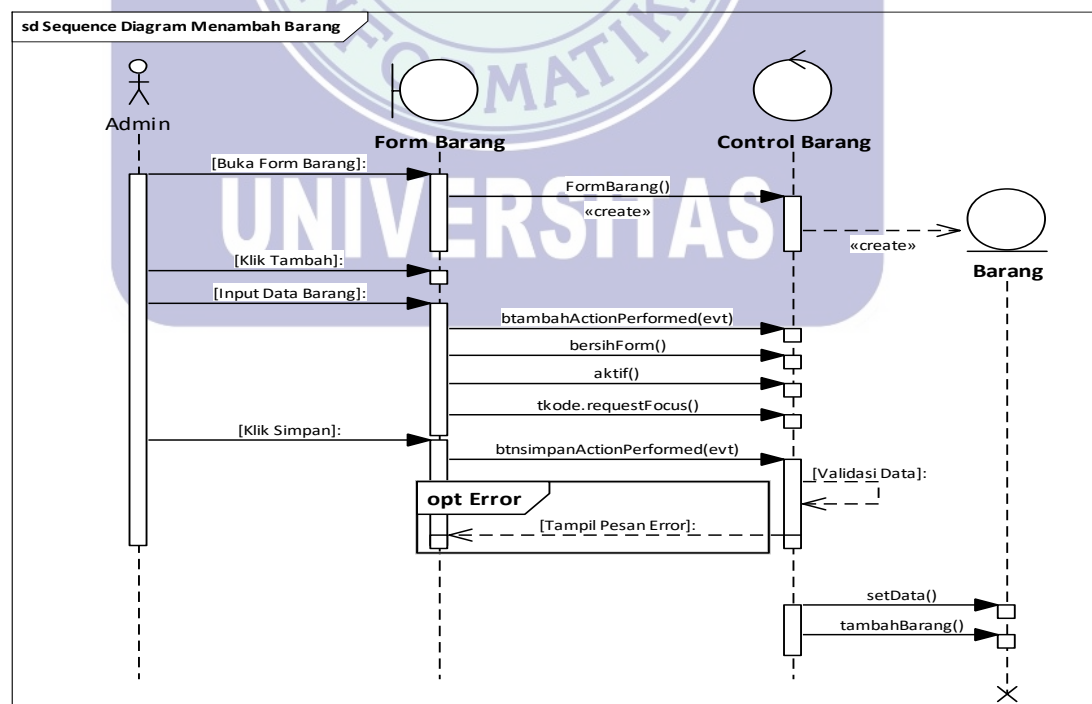
1. Sequence Diagram Login



Gambar III.19.
Class Diagram

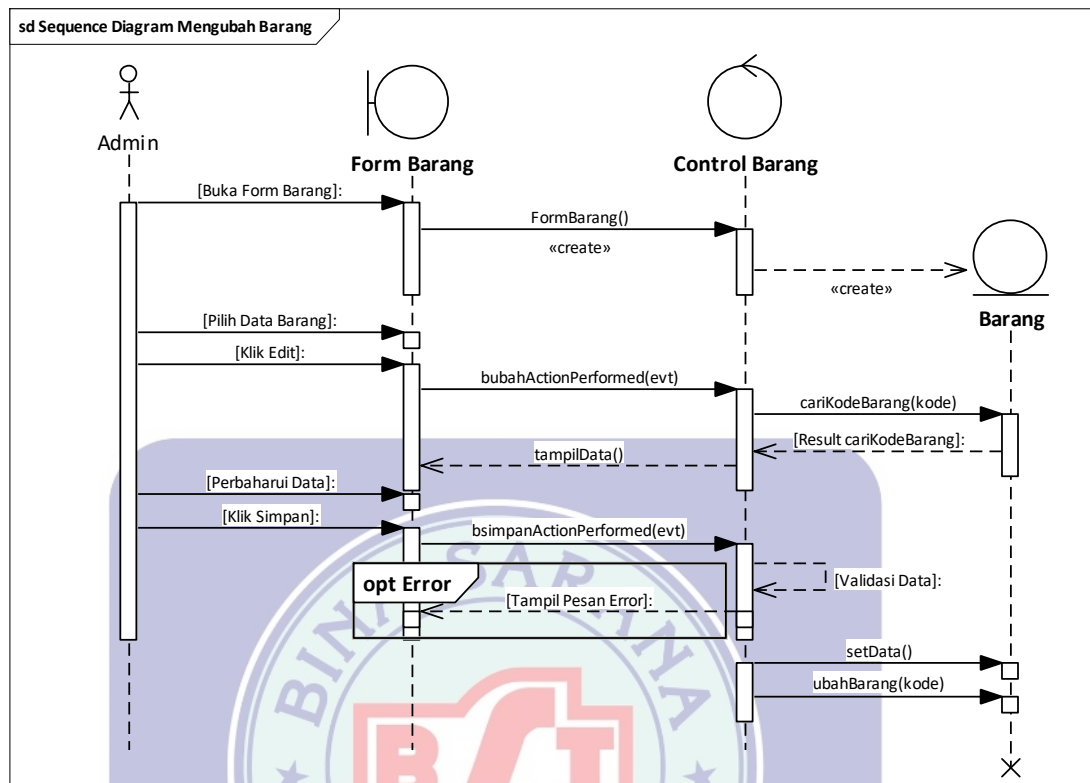
2. Sequence Diagram Mengelola Barang

a. Sequence Diagram Menambah Barang



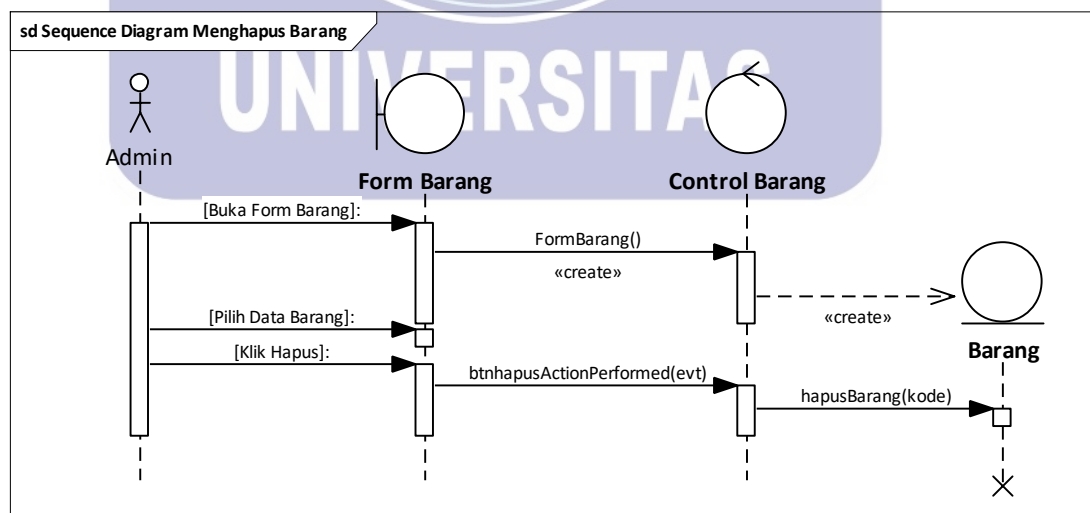
Gambar III.20.
Sequence Diagram Menambah Barang

b. *Sequence Diagram Mengubah Barang*



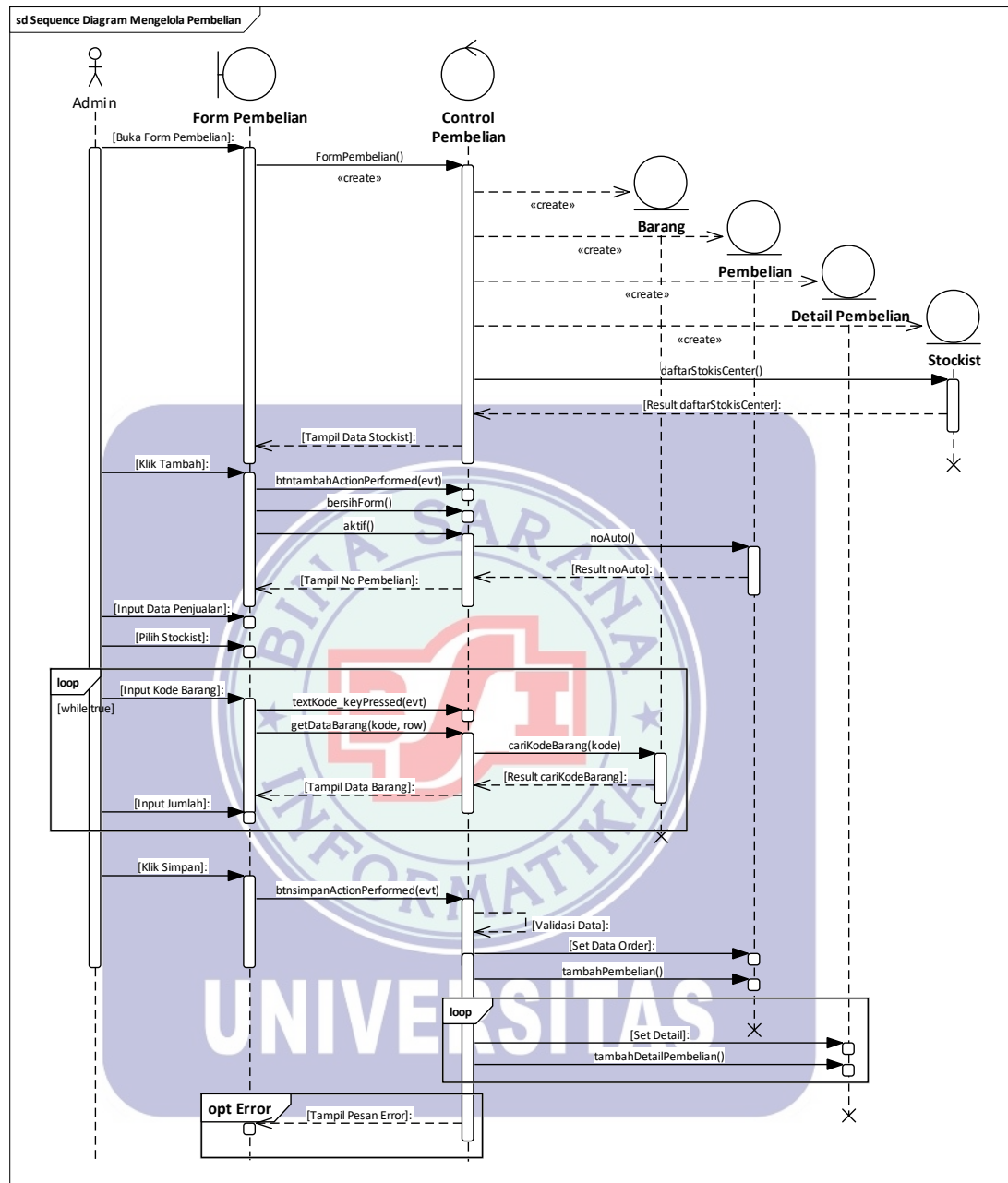
Gambar III.21.
Sequence Diagram Mengubah Barang

c. *Sequence Diagram Menghapus Barang*



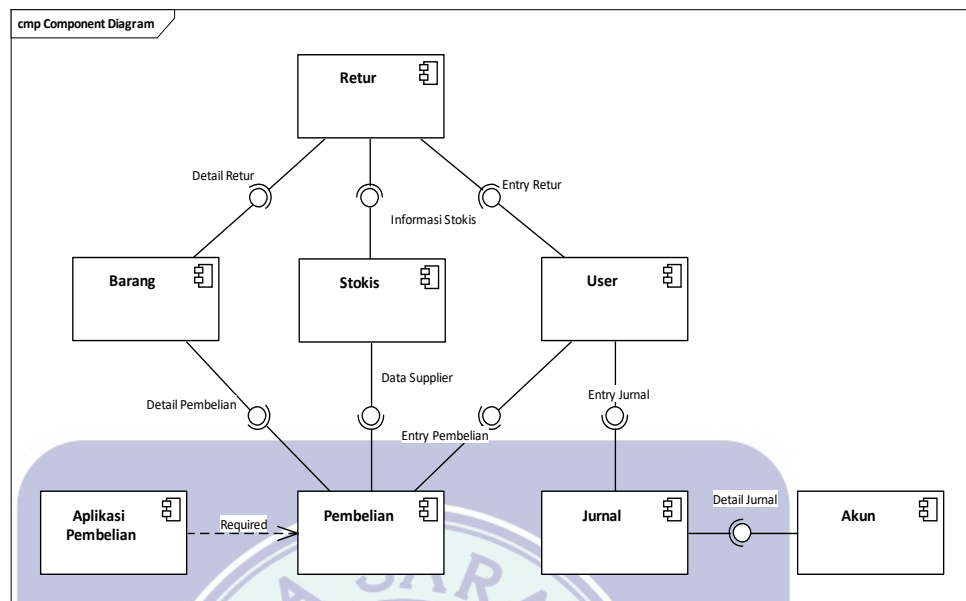
Gambar III.22.
Sequence Diagram Menghapus Barang

3. Sequence Diagram Mengelola Pembelian



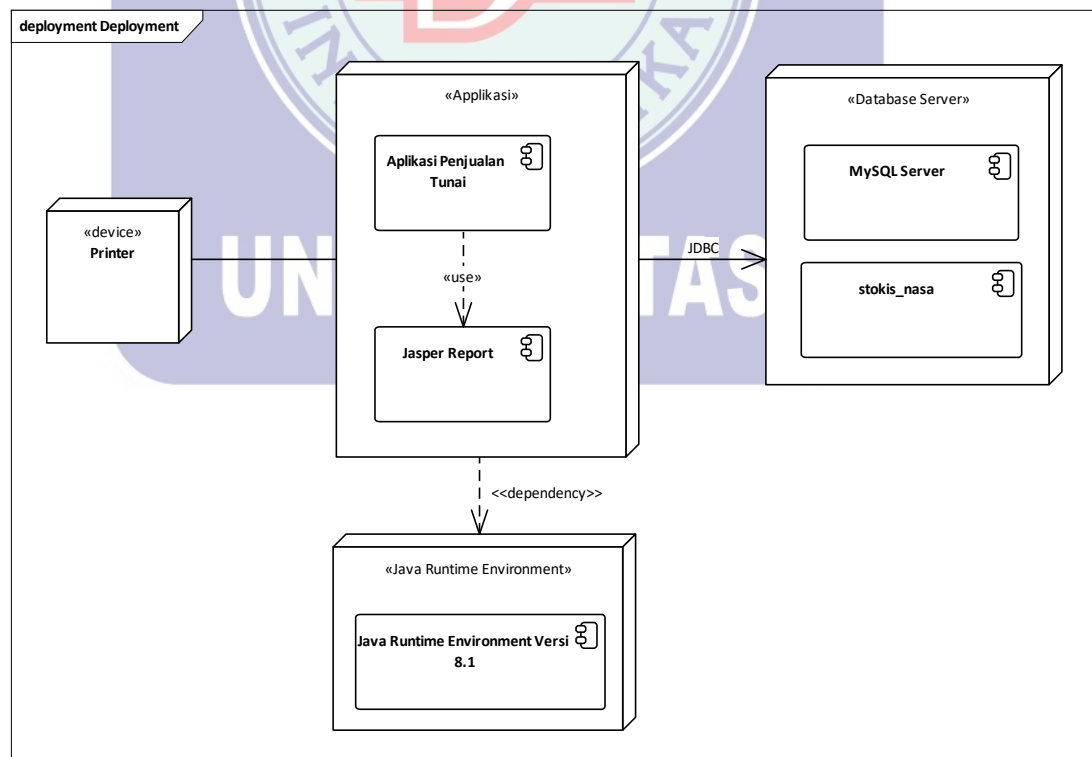
Gambar III.23.
Sequence Diagram Mengelola Pembelian

3.4.6. Component Diagram



Gambar III.24.
Component Diagram

3.4.7. Deployment Diagram



Gambar III.25.
Deployment Diagram

3.4.8. Use Interface

1. User Interface Form Login

Berikut dibawah ini merupakan tampilan dari *form login* pada Sistem Informasi Akuntansi Pembelian Barang Secara Tunai Pada PT Nasa Stokist T 2309 Cikampek:

Gambar III.26.
User Interface Form Login

2. User Interface Form Halaman Utama

Berikut dibawah ini merupakan tampilan dari *form* Halaman Utama pada Sistem Informasi Akuntansi Pembelian Barang Secara Tunai Pada PT Nasa Stokist T 2309 Cikampek:

Gambar III.27.
User Interface Form Halaman Utama

3. User Interface Form Data Master

Berikut dibawah ini merupakan tampilan dari *form* data master pada Sistem Informasi Akuntansi Pembelian Barang Secara Tunai Pada PT Nasa Stokist T 2309 Cikampek:



Gambar III.28.
User Interface Form Data Master

4. User Interface Form Barang

Berikut dibawah ini merupakan tampilan dari *form* barang pada Sistem Informasi Akuntansi Pembelian Barang Secara Tunai Pada PT Nasa Stokist T 2309 Cikampek:

Kode Barang	Nama Barang	Jenis Barang	Harga	Stock/Jumlah
BR001	SHAMPOO	Moreskin	20000	138
BR002	LACOCO	Lacoco & Lacoco	150000	57
BR003	PIATURE	Moreskin	150000	5
BR004	CDL ASALU	skunkstok	300000	5

Gambar III.29.
User Interface Form Barang

5. User Interface Form Stokist Center

Berikut dibawah ini merupakan tampilan dari *form stokist center* pada Sistem Informasi Akuntansi Pembelian Barang Secara Tunai Pada PT Nasa Stokist T 2309 Cikampek:

Kode Stockist	Nama Stockist	No Telp	Alamat
T-2310	YANATI	08789324578	JL. HERANAS

Gambar III.30.
User Interface Form Stockist Center

6. User Interface Form User

Berikut dibawah ini merupakan tampilan dari *form user* pada Sistem Informasi Akuntansi Pembelian Barang Secara Tunai Pada PT Nasa Stokist T 2309 Cikampek:

Id User	Nama User	Password	Hak Akses
saftm1	Saftm	12345678	Pemda
saftmaja	Saftm	12345678	Admin

Gambar III.31.
User Interface Form User

7. User Interface Form Akun

Berikut dibawah ini merupakan tampilan dari *form* akun pada Sistem Informasi Akuntansi Pembelian Barang Secara Tunai Pada PT Nasa Stokist T 2309 Cikampek:



Kode Akun	Nama Akun	Jenis Akun	Saldo Awal
AK001	KAS	Aktiva	DEBIT
AK002	UTANG	Hutang	KREDIT

Gambar III.32
User Interface Form Akun

8. User Interface Form Transaksi

Berikut dibawah ini merupakan tampilan dari *form* transaksi pada Sistem Informasi Akuntansi Pembelian Barang Secara Tunai Pada PT Nasa Stokist T 2309 Cikampek:



Gambar III.33.
User Interface Form Transaksi

9. User Interface Form Pembelian

Berikut dibawah ini merupakan tampilan dari *form* pembelian pada Sistem Informasi Akuntansi Pembelian Barang Secara Tunai Pada PT Nasa Stokist T 2309 Cikampek:

FORM PEMBELIAN

Input Barang Masuk Data Barang Masuk

Tanggal Transaksi

No Pembelian	Tanggal	Stockist	Kode Barang	Nama Barang	Sub Total
PM001	2020-06-02	T-2310	BR001	SHAMPOO	20000
PM002	2020-06-03	T-2310	BR002	LACOCO	1500000
PM003	2020-06-11	T-2310	BR001	SHAMPOO	20000
PM004	2020-06-18	T-2310	BR002	LACOCO	1500000
PM005	2020-07-01	T-2310	BR002	LACOCO	1500000
			BR001	SHAMPOO	20000
PM006	2020-07-01	T-2310	BR002	LACOCO	1050000

FORM PEMBELIAN

Input Barang Masuk Data Barang Masuk

No Pembelian **TOTAL**

Tanggal

Kode Stockist

Kode Barang	Nama Barang	Jumlah Barang	Harga	Sub Total

Gambar III.34.
User Interface Form Pembelian

10. *User Interface Form Retur Pembelian*

Berikut dibawah ini merupakan tampilan dari *form retur* pembelian pada Sistem Informasi Akuntansi Pembelian Barang Secara Tunai Pada PT Nasa Stokist T 2309 Cikampek:

FORM RETUR PEMBELIAN

Input Retur

Data Retur

Tanggal

Cari

Cetak

No Retur	Tanggal	Stokis	Keterangan	Kode Bara...	Nama Bara...	Jumlah Re...	Harga	Sub Total
RT00001	2020-06-25	YANTI	cacat	BR002	LACOCO	5	150000	750000
RT00002	2020-06-26	YANTI	rusak	BR001	SHAMPOO	1	20000	20000
RT00003	2020-06-30	YANTI	cacat	BR001	SHAMPOO	2	20000	20000
RT00004	2020-06-30	YANTI	rusak	BR001	SHAMPOO	3	20000	60000
RT00005	2020-07-01	YANTI	cacat	BR002	LACOCO	5	150000	750000
RT00006	2020-07-01	YANTI	cacat	BR002	LACOCO	8	150000	1200000

Tambah

Home

FORM RETUR PEMBELIAN

Input Retur

Data Retur

No Retur

RT00007

Tanggal Retur

07/08/2020

Kode Stocks

Keterangan

Kode Barang	Nama Barang	Jumlah Retur	Harga	Sub Total
-------------	-------------	--------------	-------	-----------

Total

Simpan

Hapus

Batal

Gambar III.35.
User Interface Form Retur Pembelian

11. User Interface Form Journal

Berikut dibawah ini merupakan tampilan dari *form Journal* pada Sistem Informasi Akuntansi Pembelian Barang Secara Tunai Pada PT Nasa Stokist T 2309 Cikampek:

FORM JURNAL

Input Jurnal Data Jurnal

Tanggal Jurnal

No Jurnal	Tanggal Jurnal	Keterangan	Kode Akun	Nama Akun	Debit	Kredit
JR001	2020-07-01	Hutang	AK002	UTANG	200000	0
JR002	2020-07-01	bayar hutang	AK002	UTANG	200000	0
			AK001	KAS	0	200000
JR003	2020-07-02	kas	AK001	KAS	100000	0

FORM JURNAL

Input Jurnal Data Jurnal

No Jurnal

Tanggal

Keterangan

Kode Akun	Nama Akun	Debit	Kredit

Selisih

Gambar III.36.
User Interface Form Journal

12. User Interface Form Laporan

Berikut dibawah ini merupakan tampilan dari *form* laporan pada Sistem Informasi Akuntansi Pembelian Barang Secara Tunai Pada PT Nasa Stokist T 2309 Cikampek:



The screenshot displays the 'Laporan' (Report) form. The sidebar on the left includes the NASA logo, the name 'Dini Afriliyanti S', the ID 'T-2309', and navigation buttons for 'Data Master', 'Transaksi', 'Laporan', and 'Logout'. The main content area has a title 'Laporan' and a subtitle 'Laporan Pembelian'. Below this, there is a section titled 'LAPORAN PEMBELIAN' with input fields for 'Tanggal' (Date) and 'Laporan' (Report). The 'Tanggal' field has two rows of input boxes. The 'Laporan' field is a dropdown menu with the text '-- Pilih --'. At the bottom, there are two buttons: 'CETAK' (Print) and 'KEMBALI' (Back).

Gambar III.37.
User Interface Form Laporan

3.5. Implementasi

3.5.1. Code Generation

1. Model Pembelian

```

package model;

import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

public class Pembelian {

    private Connection Koneksi;

    private String no_pembelian;
    private Date tgl_pembelian;
    private String kode_sc;
    private String id_user;
    private Integer total;
    private String PesanError;

    public Pembelian() {

        try {
            Class.forName("com.mysql.jdbc.Driver");
            Koneksi
            DriverManager.getConnection("jdbc:mysql://localhost/stokis_nasa", "root", "");
        } catch (ClassNotFoundException | SQLException ex) {
            this.PesanError = ex.getMessage();
            ex.printStackTrace();
        }
    }

    public Connection getKoneksi() {
        return Koneksi;
    }

    public void setKoneksi(Connection Koneksi) {
        this.Koneksi = Koneksi;
    }

    public String getNo_pembelian() {
        return no_pembelian;
    }
}

```

```
public void setNo_pembelian(String no_pembelian) {
    this.no_pembelian = no_pembelian;
}
```

```
public String getId_user() {
    return id_user;
}
```

```
public void setId_user(String id_user) {
    this.id_user = id_user;
}
```

```
public Date getTgl_pembelian() {
    return tgl_pembelian;
}
```

```
public void setTgl_pembelian(Date tgl_pembelian) {
    this.tgl_pembelian = tgl_pembelian;
}
```

```
public String getKode_sc() {
    return kode_sc;
}
```

```
public void setKode_sc(String kode_sc) {
    this.kode_sc = kode_sc;
}
```

```
public Integer getTotal() {
    return total;
}
```

```
public void setTotal(Integer total) {
    this.total = total;
}
```

```
public String getPesanError() {
    return PesanError;
}
```

```
public List<Pembelian> daftarPembelian() {
    try {
        this.PesanError = "";

        String Cmd = "SELECT * FROM pembelian";
        List<Pembelian> lp = new ArrayList();
        Statement st = Koneksi.createStatement();
        ResultSet rs = st.executeQuery(Cmd);
        while (rs.next()) {
            Pembelian br = new Pembelian();
            br.setNo_pembelian(rs.getString("no_pembelian"));
            br.setTgl_pembelian(rs.getDate("tgl_pembelian"));
            br.setKode_sc(rs.getString("kode_sc"));
            br.setTotal(rs.getInt("total"));
        }
    }
}
```

```

        lp.add(br);
    }
    return lp;
} catch (SQLException ex) {
    this.PesanError = ex.getMessage();
    return null;
}
}

public List<Pembelian> cariTglPembelian(Date Tgl1, Date Tgl2) {
    try {
        this.PesanError = "";

        String Cmd = "SELECT * FROM pembelian WHERE tgl_pembelian
        BETWEEN ? AND ?";
        List<Pembelian> lpm = new ArrayList();

        PreparedStatement ps = Koneksi.prepareStatement(Cmd);
        ps.setDate(1, Tgl1);
        ps.setDate(2, Tgl2);
        ResultSet rs = ps.executeQuery();

        while (rs.next()) {
            Pembelian pm = new Pembelian();
            pm.setNo_pembelian(rs.getString("no_pembelian"));
            pm.setTgl_pembelian(rs.getDate("tgl_pembelian"));
            pm.setKode_sc(rs.getString("kode_SC"));
            pm.setId_user(rs.getString("id_user"));
            pm.setTotal(rs.getInt("total"));

            lpm.add(pm);
        }
        return lpm;
    } catch (SQLException ex) {
        this.PesanError = ex.getMessage();
        return null;
    }
}

public int tambahPembelian() {
    try {
        this.PesanError = "";

        String Cmd = "INSERT INTO pembelian (no_pembelian, tgl_pembelian,
        kode_sc, id_user, total) VALUES (?, ?, ?, ?, ?)";
        PreparedStatement ps = Koneksi.prepareStatement(Cmd);

        ps.setString(1, this.no_pembelian);
        ps.setDate(2, this.tgl_pembelian);
        ps.setString(3, this.kode_sc);
        ps.setString(4, this.id_user);
        ps.setInt(5, this.total);
        return ps.executeUpdate();
    }
}

```

```

    } catch (SQLException ex) {
        this.PesanError = ex.getMessage();
        System.err.println("Got Error : "+ex.getMessage());
        return -1;
    }
}

public int ubahPembelian(String No) {
    try {
        this.PesanError = "";

        String Cmd = "UPDATE pembelian SET tgl_pembelian=?, kode_sc=?,
id_user=?, total=? WHERE no_pembelian=?";
        PreparedStatement ps = Koneksi.prepareStatement(Cmd);

        ps.setString(1, this.no_pembelian);
        ps.setDate(2, this.tgl_pembelian);
        ps.setString(3, this.kode_sc);
        ps.setString(4, this.id_user);
        ps.setInt(5, this.total);
        return ps.executeUpdate();
    } catch (SQLException ex) {
        this.PesanError = ex.getMessage();
        return -1;
    }
}

public int hapusPembelian(String No) {
    try {
        this.PesanError = "";
        String Cmd = "DELETE FROM pembelian WHERE no_pembelian=?";
        PreparedStatement ps = Koneksi.prepareStatement(Cmd);
        ps.setString(1, No);
        return ps.executeUpdate();
    } catch (SQLException ex) {
        this.PesanError = ex.getMessage();
        return -1;
    }
}

public Pembelian cariNoPembelian(String No) {
    this.PesanError = "";
    try {
        String Cmd = "SELECT * FROM pembelian WHERE no_pembelian=?";
        PreparedStatement ps = Koneksi.prepareStatement(Cmd);
        ps.setString(1, No);

        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            this.setNo_pembelian(rs.getString("no_pembelian"));
            this.setTgl_pembelian(rs.getDate("tgl_pembelian"));
            this.setKode_sc(rs.getString("kode_sc"));
            this.setId_user(rs.getString("id_user"));
            this.setTotal(rs.getInt("total"));
        }
    }
}

```

```

    } else {
        this.setNo_pembelian(null);
        this.setTgl_pembelian(null);
        this.setKode_sc(null);
        this.setId_user(null);
        this.setTotal(0);
    }
    return this;
} catch (SQLException ex) {
    System.out.println(ex.getMessage());
    this.PesanError = ex.getMessage();
    return null;
}
}

public String noAuto() {
    try {
        String Cmd = "SELECT * FROM pembelian ORDER BY
SUBSTR(no_pembelian,3,5)* 1 DESC LIMIT 1";
        Statement st = Koneksi.createStatement();
        ResultSet rs = st.executeQuery(Cmd);
        if (rs.next()) {
            String No = rs.getString("no_pembelian");
            No = No.substring(2, 5);
            int urut = Integer.parseInt(No) + 1;
            No = String.valueOf(urut);
            No = "PM" + String.format("%03d", Integer.parseInt(No));
            return No;
        } else {
            return "PM001";
        }
    } catch (SQLException ex) {
        this.PesanError = ex.getMessage();
        return "";
    }
}
}

```

2. Form Pembelian

```

package form;

import pembeliannasa.PembelianNasa;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.Toolkit;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.io.File;
import java.text.DateFormat;
import java.text.DecimalFormat;
import java.text.NumberFormat;
import java.text.SimpleDateFormat;

```

```

import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.ListIterator;
import javax.swing.DefaultCellEditor;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumn;
import model.Barang;
import model.Pembelian;
import model.DetailPembelian;
import model.StokisCenter;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.util.JRLoader;
import net.sf.jasperreports.view.JasperViewer;

public class FormPembelian extends javax.swing.JFrame {

    String NoBarangKeluar = "";
    String Status = "";

    int row = -1;
    int total_all = 0, total = 0;

    Barang barang = new Barang();
    Pembelian beli = new Pembelian();
    DetailPembelian detail_beli = new DetailPembelian();
    StokisCenter stokis = new StokisCenter();

    DefaultComboBoxModel tm1 = null;
    DefaultTableModel model, modelData;

    TableColumn cKode, cjumlah;
    JTextField textKode = new JTextField();
    JTextField textJumlah = new JTextField();

    NumberFormat nf = new DecimalFormat("#");

    SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");

    public FormPembelian() {
        initComponents();
        Toolkit tk = Toolkit.getDefaultToolkit();
        Dimension d = tk.getScreenSize();
        int x, y;
        x = (int) ((d.getWidth() - getSize().width) / 2);
        y = (int) ((d.getHeight() - getSize().height) / 2);

        tm1 = (DefaultComboBoxModel) cstokist.getModel();
        setLocation(x, y);
    }

```

```
model = (DefaultTableModel) tblDetail.getModel();
modelData = (DefaultTableModel) tblData.getModel();
```

```
textKode.setFont(new Font("BellGothic BT", 0, 16));
textJumlah.setFont(new Font("BellGothic BT", 0, 16));
```

```
tblDetail.getTableHeader().setFont(new Font("BellGothic BT", 1, 16));
tblDetail.setRowHeight(30);
tblData.getTableHeader().setFont(new Font("BellGothic BT", 1, 16));
tblData.setRowHeight(30);
```

```
cKode = (TableColumn) tblDetail.getColumnModel().getColumn(0);
cJumlah = (TableColumn) tblDetail.getColumnModel().getColumn(2);
```

```
cKode.setCellEditor(new DefaultCellEditor(textKode));
cJumlah.setCellEditor(new DefaultCellEditor(textJumlah));
```

```
textKode.addKeyListener(new KeyAdapter() {
    public void keyPressed(KeyEvent e) {
        if (e.getKeyCode() == KeyEvent.VK_ENTER) {
            row = tblDetail.getSelectedRow();
            textKode.setText(textKode.getText().toUpperCase());

            if (!"".equals(textKode.getText())) {
                getDataBarang(textKode.getText(), row);

                for (int i = 0; i < model.getRowCount() - 1; i++) {
                    String kode = model.getValueAt(i, 0).toString();
                    if (kode.equalsIgnoreCase(textKode.getText()) && row != i) {
                        model.setValueAt(null, row, 0);
                        model.setValueAt(null, row, 1);
                        model.setValueAt(null, row, 2);
                        model.setValueAt(null, row, 3);
                        model.setValueAt(null, row, 4);
                        textKode.setText(null);
                        JOptionPane.showMessageDialog(rootPane, "Data Sudah ada !!!", "Info", JOptionPane.ERROR_MESSAGE);
                    }
                }
            }
        }
    }
});
```

```
        break;
    }
}

if (e.getKeyCode() == KeyEvent.VK_F1) {
    textKode.setText(new FormCariBarang(null, true).cariBarang());
    row = tblDetail.getSelectedRow();
    textKode.setText(textKode.getText().toUpperCase());

    if (!"".equals(textKode.getText())) {
        getDataBarang(textKode.getText(), row);

        for (int i = 0; i < model.getRowCount() - 1; i++) {
```

```

String kode = model.getValueAt(i, 0).toString();
if (kode.equalsIgnoreCase(textKode.getText()) && row != i) {
    model.setValueAt(null, row, 0);
    model.setValueAt(null, row, 1);
    model.setValueAt(null, row, 2);
    model.setValueAt(null, row, 3);
    model.setValueAt(null, row, 4);
    textKode.setText(null);
    JOptionPane.showMessageDialog(rootPane, "Data sudah ada
!!!", "Info", JOptionPane.ERROR_MESSAGE);

    break;
}
}
}
}
});
textJumlah.addKeyListener(new KeyAdapter() {
    public void keyTyped(KeyEvent e) {
        if (!Character.isDigit(e.getKeyChar()) && e.getKeyChar() !=
KeyEvent.VK_BACK_SPACE && e.getKeyChar() != KeyEvent.VK_DELETE) {
            textJumlah.setEditable(false);
            JOptionPane.showMessageDialog(rootPane, "Inputan hanya boleh
berupa angka", "Error", JOptionPane.ERROR_MESSAGE);
        } else {
            textJumlah.setEditable(true);
            try {
                textJumlah.setText(String.format("%d",
Integer.parseInt(textJumlah.getText())));
            } catch (Exception ex) {
            }
            if (textJumlah.getText().equals("")) {
                textJumlah.setText("");
            }
        }
    }
});

public void keyPressed(KeyEvent e) {
    if (e.getKeyCode() == KeyEvent.VK_ENTER || e.getKeyCode() ==
KeyEvent.VK_TAB) {
        try {
            row = tblDetail.getSelectedRow();

            if (textJumlah.getText().equals("")) {
                textJumlah.setText("1");
            }
            int jumlah = Integer.parseInt(textJumlah.getText());
            int harga = Integer.parseInt(model.getValueAt(row, 3).toString());
            int subtotal = jumlah * harga;

            model.setValueAt(subtotal, row, 4);

            hitungTotal();

```

```

        } catch (Exception ex) {

        }
    }
});
aktif();

DateFormat df = new SimpleDateFormat("dd/mm/yy");
java.util.Date dt = new java.util.Date();
tanggal.setText(df.format(dt));

daftarStokisCenter();
tabs.setSelectedIndex(1);

daftarPembelian();
this.NoBarangKeluar = "";
aktif();
}

private void aktif() {
    btnbatal.setEnabled(!"".equals(Status));
    btnsimpan.setEnabled(!"".equals(Status));
    btnhapus.setEnabled(!"".equals(Status));
    //btncari.setEnabled(!"".equals(Status));
    //btn.setEnabled(!"".equals(Status));
    // btn cetak.setEnabled(!"".equals(Status) &&
    !"".equals(this.NoBarangKeluar));
    tabs.setEnabledAt(0, !"".equals(Status));
    tabs.setEnabledAt(1, !"".equals(Status));
}

private void bersihForm() {
    cstokist.setSelectedItem("");
    nopembelian.setText("");
    tanggal.setText("");
    tttotal.setText("");
    model.setRowCount(0);
    tblDetail.repaint();
    textKode.setText("");
    textJumlah.setText("");
}

private void daftarStokisCenter() {
    List<StokisCenter> liststokist = stokis.daftarStokisCenter();
    ListIterator li = liststokist.listIterator();
    tm1.removeAllElements();
    cstokist.addItem("-- Pilih --");
    while (li.hasNext()) {
        StokisCenter sc = (StokisCenter) li.next();
        cstokist.addItem(sc.getKode_sc() + " - " + sc.getNama_sc());
    }
}
}

```

```

private void getDataBarang(String kode, int row) {
    barang = barang.cariKodeBarang(kode);
    if (barang.getNama_barang() == null) {
        JOptionPane.showMessageDialog(rootPane, "Data Barang yg dicari tidak
ada", "Error", JOptionPane.ERROR_MESSAGE);
        model.setValueAt(null, row, 0);
        model.setValueAt(null, row, 1);
        model.setValueAt(null, row, 2);
        model.setValueAt(null, row, 3);
        model.setValueAt(null, row, 4);
        model.setValueAt(null, row, 5);
    } else {
        model.setValueAt(kode.toUpperCase(), row, 0);
        model.setValueAt(barang.getNama_barang(), row, 1);
        model.setValueAt(1, row, 2);
        model.setValueAt(barang.getHarga_barang(), row, 3);
        model.setValueAt(barang.getHarga_barang(), row, 4);
    }

    hitungTotal();
}

private void daftarPembelian() {
    List<Pembelian> lp = beli.daftarPembelian();
    ListIterator li = lp.listIterator();
    modelData.setRowCount(0);
    tbldata.repaint();
    String No = "";
    Object[] data = null;
    while (li.hasNext()) {
        beli = (Pembelian) li.next();
        stokis = stokis.cariKodeStokisCenter(beli.getKode_sc());

        List<DetailPembelian> dp =
detail_beli.daftarDetailPembelian(beli.getNo_pembelian());
        ListIterator<DetailPembelian> it = dp.listIterator();
        while (it.hasNext()) {
            detail_beli = it.next();
            barang = barang.cariKodeBarang(detail_beli.getKode_barang());
            if (!No.equalsIgnoreCase(beli.getNo_pembelian())) {
                Object[] jdata = {
                    beli.getNo_pembelian(),
                    beli.getTgl_pembelian(),
                    stokis.getKode_sc(),
                    detail_beli.getKode_barang(),
                    barang.getNama_barang(),
                    detail_beli.getSub_total()
                };
                data = jdata;
            } else {
                Object[] jdata = {
                    "",
                    "",
                    "",
                    ""
                };
            }
        }
    }
}

```

```

        """,
        detail_beli.getKode_barang(),
        barang.getNama_barang(),
        detail_beli.getSub_total()
    };
    data = jdata;
}
modelData.addRow(data);
No = beli.getNo_pembelian();
}
}
}

private void daftarPembelian(Date tgl1, Date tgl2) {
    List<Pembelian> lp = beli.cariTglPembelian(new
java.sql.Date(tgl1.getTime()), new java.sql.Date(tgl2.getTime()));
    ListIterator li = lp.listIterator();
    modelData.setRowCount(0);
    tbldata.repaint();
    String No = "";
    Object[] data = null;
    while (li.hasNext()) {
        beli = (Pembelian) li.next();
        stokis = stokis.cariKodeStokisCenter(beli.getKode_sc());

        List<DetailPembelian> dp =
detail_beli.daftarDetailPembelian(beli.getNo_pembelian());
        ListIterator<DetailPembelian> it = dp.listIterator();
        while (it.hasNext()) {
            detail_beli = it.next();
            barang = barang.cariKodeBarang(detail_beli.getKode_barang());
            if (!No.equalsIgnoreCase(beli.getNo_pembelian())) {
                Object[] jdata = {
                    beli.getNo_pembelian(),
                    beli.getTgl_pembelian(),
                    stokis.getKode_sc(),
                    detail_beli.getKode_barang(),
                    barang.getNama_barang(),
                    detail_beli.getSub_total()
                };
                data = jdata;
            } else {
                Object[] jdata = {
                    "",
                    "",
                    "",
                    "",
                    detail_beli.getKode_barang(),
                    barang.getNama_barang(),
                    detail_beli.getSub_total()
                };
                data = jdata;
            }
        }
        modelData.addRow(data);
        No = beli.getNo_pembelian();
    }
}

```

```

    }
}

private void hitungTotal() {
    this.total = 0;
    for (int i = 0; i < model.getRowCount(); i++) {
        try {
            this.total += Integer.parseInt(model.getValueAt(i, 4).toString());
        } catch (Exception ex) {
        }
    }
    tttotal.setText(nf.format(this.total));
}

/**
 * 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() {

    tabs = new javax.swing.JTabbedPane();
    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    nopembelian = new javax.swing.JTextField();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblDetail = new javax.swing.JTable();
    jLabel6 = new javax.swing.JLabel();
    tttotal = new javax.swing.JTextField();
    cstokist = new javax.swing.JComboBox<>();
    ttanggal = new javax.swing.JTextField();
    btnsimpan = new javax.swing.JButton();
    btnhapus = new javax.swing.JButton();
    btnbatal = new javax.swing.JButton();
    jPanel2 = new javax.swing.JPanel();
    jScrollPane2 = new javax.swing.JScrollPane();
    tblData = new javax.swing.JTable();
    jLabel5 = new javax.swing.JLabel();
    btncari = new javax.swing.JButton();
    tgl1 = new com.toedter.calendar.JDateChooser();
    tgl2 = new com.toedter.calendar.JDateChooser();
    btncetak = new javax.swing.JButton();
    btntambah = new javax.swing.JButton();
    btnhome = new javax.swing.JButton();
    jPanel3 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

```

```

tabs.setBackground(new java.awt.Color(51, 153, 0));

jPanel1.setBackground(new java.awt.Color(102, 255, 102));

jLabel2.setText("No Pembelian");

jLabel3.setText("Tanggal");

jLabel4.setText("Kode Stockist");

tblDetail.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null}
    },
    new String [] {
        "Kode Barang", "Nama Barang", "Jumlah Barang", "Harga", "Sub Total"
    }
){
    boolean[] canEdit = new boolean [] {
        true, false, true, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
tblDetail.setSelectionBackground(new java.awt.Color(102, 255, 102));
tblDetail.setSelectionForeground(new java.awt.Color(0, 0, 0));
tblDetail.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblDetailMouseClicked(evt);
    }
});
tblDetail.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        tblDetailKeyPressed(evt);
    }
});
jScrollPane1.setViewportViewView(tblDetail);

jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
jLabel6.setText("TOTAL");

tTotal.setEditable(false);
tTotal.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tTotalActionPerformed(evt);
    }
});

```

```

        cstokist.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"-- Pilih --" }));
        cstokist.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                cstokistActionPerformed(evt);
            }
        });

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

        btnsimpan.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
        btnsimpan.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/save.png"))); // NOI18N
        btnsimpan.setText("Simpan");
        btnsimpan.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnsimpanActionPerformed(evt);
            }
        });

        btnhapus.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
        btnhapus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/hapus.jpg"))); // NOI18N
        btnhapus.setText("Hapus");

        btnbatal.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
        btnbatal.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/batal.png"))); // NOI18N
        btnbatal.setText("Batal");
        btnbatal.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnbatalActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(btnsimpan, javax.swing.GroupLayout.DEFAULT_SIZE, 100, true)
                    .addComponent(btnhapus, javax.swing.GroupLayout.DEFAULT_SIZE, 100, true)
                    .addComponent(btnbatal, javax.swing.GroupLayout.DEFAULT_SIZE, 100, true)
                )
            )
        );
    }

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

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

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

```

```

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel2)
    .addComponent(jLabel3)
    .addComponent(jLabel4))
.addGap(33, 33, 33)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addComponent(nopembelian,
            javax.swing.GroupLayout.PREFERRED_SIZE, 71,
            javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 556,
    Short.MAX_VALUE)
        .addComponent(jLabel6)
        .addGap(44, 44, 44)
        .addComponent(tttotal,
            javax.swing.GroupLayout.PREFERRED_SIZE, 180,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(25, 25, 25))
        .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(ttanggal,
        javax.swing.GroupLayout.PREFERRED_SIZE, 100,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(cstokist,
            javax.swing.GroupLayout.PREFERRED_SIZE, 140,
            javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE))))
        .addComponent(jScrollPane1,
            javax.swing.GroupLayout.Alignment.TRAILING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(btnsimpan,
                javax.swing.GroupLayout.PREFERRED_SIZE, 103,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(btnhapus,
                javax.swing.GroupLayout.PREFERRED_SIZE, 95,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(btnbatal,
                javax.swing.GroupLayout.PREFERRED_SIZE, 93,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))))
    );
jPanel1Layout.setVerticalGroup(

```

```

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(4, 4, 4)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
    .addComponent(tttotal,
javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel6)))
    .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
    .addComponent(jLabel2)
    .addComponent(nopembelian,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(21, 21, 21)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
    .addComponent(jLabel3)
    .addComponent(ttanggal,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
    .addComponent(cstokist,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel4)))
    .addGap(18, 18, 18)
    .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 168,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 83,
Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)

```

```

        .addComponent(btnsimpan)
        .addComponent(btnhapus)
        .addComponent(btnbatal))
        .addContainerGap()
    );

    tabs.addTab("Input Barang Masuk", jPanel1);

    jPanel2.setBackground(new java.awt.Color(102, 255, 102));

    tbldata.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null, null, null, null},
            {null, null, null, null, null, null},
            {null, null, null, null, null, null},
            {null, null, null, null, null, null}
        },
        new String [] {
            "No Pembelian", "Tanggal", "Stockist", "Kode Barang", "Nama Barang",
            "Sub Total"
        }
    ){
        boolean[] canEdit = new boolean [] {
            false, false, false, false, false, false
        };

        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    });
    tbldata.setSelectionBackground(new java.awt.Color(102, 255, 102));
    tbldata.setSelectionForeground(new java.awt.Color(0, 0, 0));
    tbldata.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            tbldataMouseClicked(evt);
        }
    });
    jScrollPane2.setViewportViewView(tbldata);

    jLabel5.setText("Tanggal Transaksi");

    btncari.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
    btncari.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/icon/cari1.jpg"))); // NOI18N
    btncari.setText("Cari");
    btncari.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btncariActionPerformed(evt);
        }
    });

    btncetak.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
    btncetak.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/icon/cetak.png"))); // NOI18N

```

```

btncetak.setText("Cetak");
btncetak.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btncetakActionPerformed(evt);
    }
});

btntambah.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
btntambah.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/tambah.jpg"))); // NOI18N
btntambah.setText("Tambah");
btntambah.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btntambahActionPerformed(evt);
    }
});

btnhome.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
btnhome.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/backhome.png"))); // NOI18N
btnhome.setText("Home");
btnhome.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnhomeActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(10, 10, 10)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel5)
            .addComponent(tgl1, javax.swing.GroupLayout.PREFERRED_SIZE,
142, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(tgl2, javax.swing.GroupLayout.PREFERRED_SIZE,
137, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(btncari,
javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 467,
Short.MAX_VALUE)
    .addComponent(btncetak)
    .addContainerGap()
    .addComponent(jScrollPane2,
javax.swing.GroupLayout.DEFAULT_SIZE, 1034, Short.MAX_VALUE)
    .addGroup(jPanel2Layout.createSequentialGroup())
    .addComponent(btntambah)
    .addGap(18, 18, 18)
    .addComponent(btnhome)
    .addGap(0, 0, Short.MAX_VALUE))))
);
jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
    .addGroup(jPanel2Layout.createSequentialGroup())
    .addContainerGap()

    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)

    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.TRAILING)
        .addComponent(tgl1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel5)
        .addComponent(tgl2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btncari))
        .addComponent(btncetak))
        .addGap(18, 18, 18)
        .addComponent(jScrollPane2,
javax.swing.GroupLayout.DEFAULT_SIZE, 305, Short.MAX_VALUE)
        .addGap(18, 18, 18)

    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING, false)
        .addComponent(btntambah)
        .addGroup(jPanel2Layout.createSequentialGroup())
        .addComponent(btnhome,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGap(2, 2, 2)))
        .addContainerGap()
);

tabs.addTab("Data Barang Masuk", jPanel2);

jPanel3.setBackground(new java.awt.Color(51, 153, 0));

```

```

jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 36)); // NOI18N
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("FORM PEMBELIAN");

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

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
    .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
        .addGap(17, Short.MAX_VALUE)
        .addComponent(jLabel1)
        .addGap())
    );

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(tabs)
        .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(tabs)
            .addGap())
        );

pack();
} // </editor-fold>

private void btntambahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.Status = "Tambah";

    bersihForm();

```

```

nopembelian.setText(beli.noAuto());

ttanggal.setText(sdf.format(new java.util.Date()));
aktif();
tabs.setSelectedIndex(0);
model.setRowCount(1);
this.NoBarangKeluar = "";
}

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

    try{
        String FileReport = "src/report/Pembelian.jasper";
        HashMap parameter = new HashMap();
        parameter.put("No", this.NoBarangKeluar);

        File file = new File(FileReport);
        JasperReport jr = (JasperReport) JRLoader.loadObject(file);
        JasperPrint jrp = JasperFillManager.fillReport(jr, parameter,
beli.getKoneksi());
        JasperViewer.viewReport(jrp, false);
    }catch (Exception ex){
        ex.printStackTrace();
    }
}

private void btncariActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (tgl1.getDate() == null || tgl2.getDate() == null) {
        daftarPembelian();
    } else {
        daftarPembelian(tgl1.getDate(), tgl2.getDate());
    }
}

private void tbldataMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int row = this.tbldata.getSelectedRow();
    this.NoBarangKeluar = modelData.getValueAt(row, 0).toString();
    aktif();
}

private void btnbatalActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Status = "";
    nopembelian.setText("");

    DateFormat df = new SimpleDateFormat("dd/mm/yyyy");
    java.util.Date dt = new java.util.Date();
    ttanggal.setText(df.format(dt));
    tabs.setSelectedIndex(1);
    bersihForm();
}

```

```

    aktif();
}

private void btsimpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if ("".equals(cstokist.getSelectedItem().toString())) {
        JOptionPane.showMessageDialog(rootPane, "Stokis belum di pilih",
            "Galat", JOptionPane.ERROR_MESSAGE);
        return;
    }

    if ("".equals(nopembelian.getText())) {
        JOptionPane.showMessageDialog(rootPane, "No Pembelian belum diisi",
            "Galat", JOptionPane.ERROR_MESSAGE);
        return;
    }

    if ("".equals(ttanggal.getText())) {
        JOptionPane.showMessageDialog(rootPane, "Tanggal Beli belum diisi",
            "Galat", JOptionPane.ERROR_MESSAGE);
        return;
    }

    if ("".equals(model.getRowCount() == 0)) {
        JOptionPane.showMessageDialog(rootPane, "Belum ada Barang yang
            dipilih", "Galat", JOptionPane.ERROR_MESSAGE);
        return;
    }

    beli.setNo_pembelian(nopembelian.getText());
    java.util.Date dt = new java.util.Date();
    java.sql.Date sdate = new java.sql.Date(dt.getTime());
    beli.setNo_pembelian(nopembelian.getText());
    beli.setTgl_pembelian(sdate);
    String[] Stok = cstokist.getSelectedItem().toString().split("-");
    beli.setKode_sc(Stok[0] + "-" + Stok[1]);
    beli.setNo_pembelian(nopembelian.getText());
    beli.setTotal(total);
    beli.setId_user(PembelianNasa.id_user);

    System.err.println(beli.getKode_sc());

    beli.tambahPembelian();
    for (int i = 0; i < model.getRowCount(); i++) {
        //try {
        String Kode = (String) model.getValueAt(i, 0);
        int sub_total = Integer.parseInt(model.getValueAt(i, 4).toString());
        int Jumlah = Integer.parseInt(model.getValueAt(i, 2).toString());
        int Harga = Integer.parseInt(model.getValueAt(i, 3).toString());
        detail_beli.setNo_pembelian(nopembelian.getText());
        detail_beli.setKode_barang(Kode);
        detail_beli.setJumlah_beli(Jumlah);
        detail_beli.setHarga_barang(Harga);
        }
    }
}

```

```

        detail_beli.setSub_total(sub_total);

        detail_beli.tambahDetailPembelian();

        barang = barang.cariKodeBarang(Kode);
        barang.setJumlah_barang(barang.getJumlah_barang() + Jumlah);
        barang.ubahBarang(Kode);

        // } catch (Exception ex) {
        // }
    }
    nopembelian.setText("");
    cstokist.setSelectedItem("");
    model.getDataVector().removeAllElements();
    tblDetail.repaint();
    JOptionPane.showMessageDialog(rootPane, "Penerimaan Barang Berhasil",
    "Info", JOptionPane.INFORMATION_MESSAGE);
    tabs.setSelectedIndex(1);

    Status = "";
    daftarPembelian();
    aktif();
}

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

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

private void tblDetailKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if (evt.getKeyCode() == KeyEvent.VK_DELETE) {
        if (!"".equals(NoBarangKeluar)) {
            row = tblDetail.getSelectedRow();
            model.removeRow(row);
            tblDetail.repaint();
            if (model.getRowCount() == 0) {
                model.setRowCount(1);
            }
        }
    }
}

if (evt.getKeyCode() == KeyEvent.VK_TAB) {
    try {
        double subtotal =
        Double.parseDouble(model.getValueAt(model.getRowCount() - 1, 4).toString());
        if (subtotal > 0) {
            model.setRowCount(model.getRowCount() + 1);
            tblDetail.repaint();
        }
    } catch (Exception ex) {
    }
}

```

```

    }
}

private void tblDetailMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    try {
        row = tblDetail.getSelectedRow();
        NoBarangKeluar = (String) model.getValueAt(row, 0);
    } catch (Exception ex) {
    }
}

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

private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //MenuUtama1 mu = new MenuUtama1();
    //mu.setVisible(true);
    dispose();
}

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

java.util.logging.Logger.getLogger(FormPembelian.class.getName()).log(java.util.l
ogging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(FormPembelian.class.getName()).log(java.util.l
ogging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

```

```

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

```

```

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

```

```

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

```

```

// Variables declaration - do not modify
private javax.swing.JButton btnbatal;
private javax.swing.JButton btncari;
private javax.swing.JButton btncetak;
private javax.swing.JButton btnhapus;
private javax.swing.JButton btnhome;
private javax.swing.JButton btnsimpan;
private javax.swing.JButton btntambah;
private javax.swing.JComboBox<String> cstokist;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTextField nopembelian;
private javax.swing.JTabbedPane tabs;
private javax.swing.JTable tbldata;
private javax.swing.JTable tbldetail;
private com.toedter.calendar.JDateChooser tgl1;
private com.toedter.calendar.JDateChooser tgl2;
private javax.swing.JTextField ttanggal;
private javax.swing.JTextField tttotal;
// End of variables declaration

```

```

}

```

3.5.2. Blackbox Testing

1. Blackbox Testing Form Login

Tabel III.34.
Blackbox Testing Form Login

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Id User dan password kosong kemudian klik Login	Id User: (Kosong) Password: (Kosong)	Sistem akan menolak akses dan menampilkan pesan " Id User atau Password tidak boleh kosong!"	Sesuai Harapan	Valid
2	Id User diisi sedangkan Password tidak diisi kemudian klik Login	Id User: safitri.aja Password: (Kosong)	Sistem akan menolak akses dan menampilkan pesan "Id User atau Password tidak boleh kosong! "	Sesuai Harapan	Valid
3	Id User tidak di kosong dan Password diisi kemudian klik Login	Id User: (Kosong) Password: 12345678	Sistem akan menolak akses dan menampilkan pesan "Id User dan Password tidak boleh kosong!"	Sesuai Harapan	Valid
4	Memasukkan salah satu kondisi salah pada Id User atau password kemudian klik Login	Id User: safitri.aja (Benar) Password: 123456(Salah)	Sistem akan menolak akses dan menampilkan pesan "Id User atau Password Anda keliru"	Sesuai Harapan	Valid
5	Mengetikkan Id User dan password	Id User: safitri.aja (Benar)	Sistem akan menerima akses Login	Sesuai Harapan	Valid

	dengan data yang benar kemudian klik <i>Login</i>	<i>Password:</i> 12345678 (Benar)	dan menampilkan <i>Form Menu Utama</i>		
--	---	---	--	--	--

2. Blackbox Testing Form Barang

Tabel III.35.
Blackbox Testing Form Barang

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Semua <i>field</i> tidak diisi kemudian klik [Simpan]	Kode Barang: (auto) Nama Barang (Kosong) Jenis Barang (Kosong) Jumlah Barang (Kosong) Harga Barang (Kosong)	Sistem akan menolak dan menampilkan pesan "Mohon isi dulu semua data!"	Sesuai Harapan	<i>Valid</i>
2	Semua <i>field</i> diisi, kecuali <i>field</i> [Jenis Barang], kemudian klik [Simpan]	Kode Barang: (auto) Nama Barang Lacoco & Looke Jenis Barang (Kosong) Harga Barang 150000 Jumlah Barang 50	Sistem akan menolak dan menampilkan pesan "Mohon isi dulu semua data!"	Sesuai Harapan	<i>Valid</i>
3	Semua <i>field</i> diisi, tetapi <i>field</i> [Harga Barang] tidak diisi dengan angka, kemudian klik [Simpan]	Kode Barang: (auto) Nama Barang Lacoco & Looke Jenis Barang Lacoco Harga Barang AAA123 Jumlah Barang 50	Sistem akan menolak dan menampilkan pesan "Fomat [Harga Barang] tidak valid!"	Sesuai Harapan	<i>Valid</i>
4	Semua <i>field</i> diisi, tetapi <i>field</i> [Jumlah Barang] tidak diisi dengan	Kode Barang: (auto) Nama Barang Lacoco & Looke	Sistem akan menolak dan menampilkan pesan "Fomat [Jumlah	Sesuai Harapan	<i>Valid</i>

	angka, kemudian klik [Simpan]	Jenis Barang Lacoco Harga Barang 150000 Jumlah Barang XYZ	Barang] tidak valid!"		
5	Semua <i>field</i> diisi dengan data yang diinginkan, kemudian klik [Simpan]	Kode Barang: (auto) Nama Barang Lacoco & Looke Jenis Barang Lacoco Harga Barang 150000 Jumlah Barang 50	Sistem akan menerima dan menyimpan data kemudian menampilkan pesan "Data berhasil disimpan"	Sesuai Harapan	<i>Valid</i>

3. Blackbox Testing Form Pembelian

Tabel III.36.
Blackbox Testing Form Pembelian

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Semua <i>field</i> tidak diisi kemudian klik tombol [Simpan]	No Pembelian: (otomatis) Tanggal Pembelian: (otomatis) Stokis Center: (kosong) Daftar Barang: (kosong)	Sistem akan menolak dan menampilkan pesan " Mohon isi dulu semua data!"	Sesuai Harapan	<i>Valid</i>
2	Semua <i>field</i> diisi, tetapi belum ada barang yang dipilih dari daftar, kemudian klik tombol [Simpan]	No Pembelian: (otomatis) Tanggal Pembelian: (otomatis) Stokis Center: T-2310 Yanti Daftar Barang: (kosong)	Sistem akan menolak dan menampilkan pesan "Stokis belum di pilih"	Sesuai Harapan	<i>Valid</i>
3	Semua <i>field</i> diisi dan barang sudah dipilih dan	No Pembelian: (otomatis) Tanggal Pembelian:	Sistem akan menerima dan menyimpan data kemudian	Sesuai Harapan	<i>Valid</i>

	dimasukkan ke dalam daftar, kemudian klik tombol [Simpan]	(otomatis) Stokis Center: T-2310 Yanti Daftar Barang: (kosong)	menampilkan pesan “Data berhasil disimpan”		
--	---	--	--	--	--

3.5.3. Spesifikasi *Hardware* dan *Software*

Tabel III.40
Spesifikasi *Hardware* dan *Software*

Kebutuhan	Keterangan
<i>Sistem</i>	<i>Windows 7</i> atau di atasnya
<i>Processor</i>	Intel Core 2 Duo
<i>RAM</i>	2 GB (minimal) disarankan 4GB
<i>Harddisk</i>	1 GB untuk instalasi Java Runtime Environment dan <i>library</i>
<i>Monitor</i>	14” LCD
<i>Keyboard</i>	102 Keys
<i>Printer</i>	Dot Matrix
<i>Mouse</i>	Standar
<i>Software</i>	Kebutuhan menjalankan aplikasi secara lokal: Bahasa Pemrograman Java 8.1 DBMS MySQL Server versi 5 atau MariaDB Report Jasper Report dengan <i>library</i> IReportx