BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem

Sistem *Inventory* adalah sistem yang membahas mengenai persediaan barang pada sebuah perusahaan, yang mana didalamnya mencakup penjualan, pembelian dan transfer barang. Perusahaan memiliki *inventory* (persediaan) dengan tujuan agar dapat memelihara kelancaran bisnis yang dijalankannya. Terkadang perusahaan dilematis ketika memiliki persediaan yang tinggi maka akan memungkinkan perusahaan untuk memenuhi permintaan yang mendadak, akan tetapi pada saat yang sama persediaan yang tinggi menyebabkan perusahaan memerlukan biaya yang semakin besar pula untuk biaya penyimpanannya. Tujuan dari pengelolaan *inventory* adalah perputaran dari *inventory*, yaitu *turnover* secepat mungkin tanpa kehilangan *sales* sebagai akibat kehabisan *inventory*.

Oleh karena itu dibutuhkan sebuah analisis terhadap sistem yang berjalan guna menemukan formulasi rancangan terbaik untuk menyelesaikan berbagai kendala sistematis tersebut. Analisis Sistem merupakan langkah untuk merinci setiap proses yang terjadi di dalam sistem, dimana analisis ini bertujuan untuk menentukankebutuhan informasi dari tiap-tiap bagian organisasi serta menentukan kekurangan dari prosedur yang saat ini berjalan. Analisis ini juga mengharuskan kita untuk menganalisis Dokumen, Prosedur berjalan dan melakukan evaluasi terhadap sistem yang berjalan.

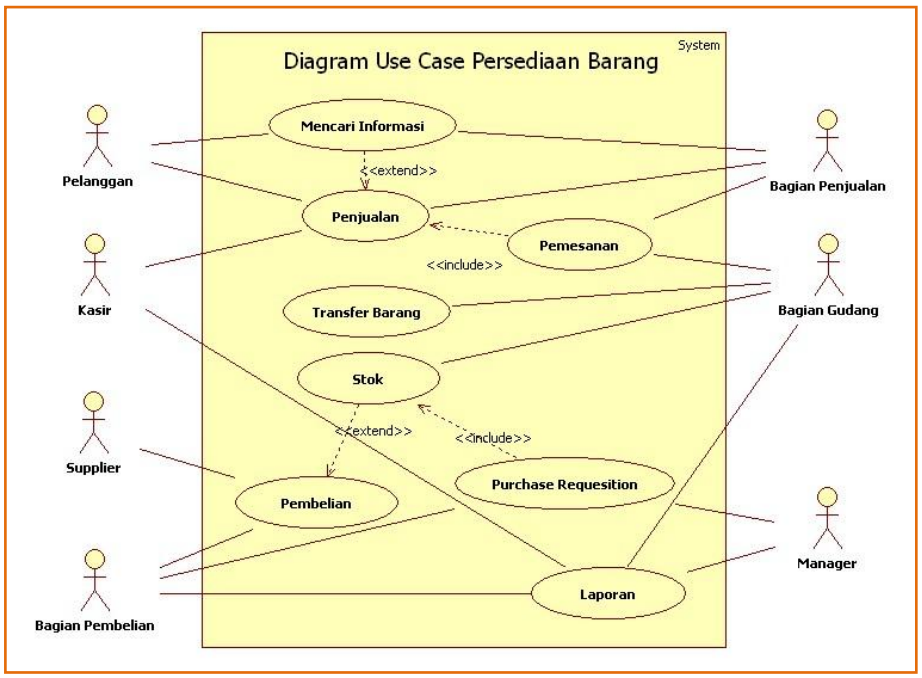
4.1.1 Analisis *Actor* dan *Use Case Diagram*

Actor dan *use case* ditentukan atas dasar fungsi-fungsi dalam sistem. Selanjutnya *use* case menyediakan nilai hasil kepada *actor*. Atas dasar analisis prosedur setidaknya ada tujuh *actor* yaitu Pelanggan, Bagian Pembelian, Bagian Penjualan, Kasir, Bagian Gudang, *Supplier* dan *Manager*.

4.1.1.1 *Use Case Diagram*

Use case diagram adalah diagram yang menyajikan interaksi antara *use case* dan *actor*. Dimana *actor* dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.

Berikut ini adalah gambar model *Use Case Diagram* Persediaan barang yang sedang berjalan, yang digambarkan secara umum:



Gambar 4.1 *Use Case Diagram* Persediaan Barang

4.1.1.2 Dokumentasi Skenario *Use Case*

Setiap *use case* di atas harus dideskripsikan dalam dokumen yang disebut dengan dokumen *flow of event*. Dokumen ini merupakan definisi apa yang harus dilakukan sistem ketika *actor* mengaktifkan *use case*. Berikut adalah dokumentasi *use case* untuk *Use Case Diagram* Penjualan dan Persediaan yang sedang berjalan.

Tabel 4.1 Skenario *Use Case* Mencari Informasi Barang

Use Case	**Mencari Informasi**	
Brief Description	*Use Case* ini memungkinkan *Pelanggan* mengetahui informasi detail mengenai barang yang dijual oleh perusahaan serta mendapatkan solusi penjualan.	
Actor	*Pelanngan, Bagian Penjualan*	
Precondition	*Pelanggan* melihat informasi melalui Katalog, dsb.	
Main Flow	***Actor***	***System***
	1. *Pelanggan* mencari informasi dari katalog, telepon, atau datang langsung ke *outlet*	
		2. *Bagian Penjualan* memeriksa kebutuhan informasi dari *pelanggan*
		3. Memberikan informasi kepada *pelanggan*.
Postcondition	*Pelanggan* mendapatkan informasi barang untuk membantu dalam mengambil keputusan transaksi.	

Tabel 4.2 Skenario *Use Case* Transaksi Penjualan

Use Case	**Penjualan**	
Brief Description	*Use Case* ini memungkinkan *Pelanggan* untuk melakukan transaksi penjualan barang, termasuk dokumen pesanan dan data-data *pelanggan* serta rincian pembayarannya.	
Actor	*Pelanngan, Bagian Penjualan, Kasir*	
Precondition	*Pelanggan* melihat katalog dan memilih barang	
Main Flow	***Actor***	***System***
	1. *Pelanggan*	

	memutuskan memesan barang yang dipilih	
		2. *Bagian Penjualan* membuat Nota Pesanan atas barang yang dipesan oleh *pelanggan*
		3. Nota Pesanan kemudian diserahkan ke *Bagian Gudang* untuk dicek ketersediaan barangnya
		4. Jika barang ada maka Nota Pesanan akan diteruskan pada *Kasir*
		5. Kasir membuat Faktur penjualan
	6. *Pelanggan* melakukan pembayaran di kasir	
Alternative Flow	Jika barang yang dipesan tidak ada, *Bagian Gudang* akan menginformasikan pada *Bagian Penjualan* untuk diteruskan pada *pelanggan*. *Pelanggan* dapat memutuskan apakah akan melakukan pesanan atau pembatalan.	
Postcondition	*Pelanggan* menuju *Kasir* untuk melakukan pembayaran dan mendapatkan Faktur Penjualan.	

Tabel 4.3 Skenario *Use Case* Pemesanan

Use Case	**Nota Pesanan**	
Brief Description	*Use Case* ini memungkinkan *Bagian Penjualan* untuk mengisi pesanan yang akan dibeli oleh *Pelanggan*	
Actor	*Bagian Penjualan*	
Precondition	*Bagian Penjualan* membuat Nota Pesanan	
Main Flow	***Actor***	***System***
	1. *Pelanggan* memilih dan memutuskan pemesanan barang	
		2. *Bagian Penjualan* membuat Nota Pesanan yang berisi, Tanggal, Nama Pelanggan, Nama Barang, Harga

		Barang, Jumlah Barang
Postcondition	Nota Pesanan diserahkan kepada *Bagian Gudang* untuk di cek apakah barang tersedia atau tidak.	

Tabel 4.4 Skenario *Use Case* Stok Barang

Use Case	**Stok**	
Brief Description	*Use Case* ini memungkinkan *Bagian Gudang* mengetahui kondisi persediaan barang	
Actor	*Bagian Gudang*	
Precondition	*Bagian Gudang* melihat Laporan Stok	
Main Flow	***Actor***	***System***
	1. *Bagian Gudang* melakukan cek Stok dan Kondisi Barang dalam Gudang	
		2. Hasil cek kemudian dibuatkan Laporan Stok Barang
Postcondition	Laporan Stok dan Surat Pengajuan Pembelian Barang dari *Bagian Gudang*.	

Tabel 4.5 Skenario *Use Case Purchase Requesition*

Use Case	***Purchase Requesition***	
Brief Description	*Use Case* ini memungkinkan *Bagian Pembelian* untuk mengajukan pembelian Barang pada *Manager*	
Actor	*Bagian Pembelian, Manager*	
Precondition	Laporan dari *Bagian Gudang*	
Main Flow	***Actor***	***System***
	1. *Bagian Gudang* melakukan cek stok barang	
		2. Hasil cek stok dibuatkan Laporan Stok, dan diserahkan pada Bagian Pembelian
		3. Bagian Pembelian membuat

		Surat PR
		4. *Manager* meng-ACC surat pengajuan pembelian barang
Postcondition	Surat Pengajuan Pembelian Barang yang telah di-ACC	

Tabel 4.6 Skenario *Use Case* Transaksi Pembelian

Use Case	**Transaksi Pembelian**	
Brief Description	*Use Case* ini memungkinkan *Bagian Pembelian* untuk melakukan pembelian barang dari *Supplier*	
Actor	*Bagian Pembelian, Supplier*	
Precondition	Berdasarkan Laporan dan Surat Pesanan	
Main Flow	***Actor***	***System***
	1. *Bagian Pembelian* mengirimkan Surat PO kepada *Supplier*	
		2. *Supplier* melakukan cek terhadap surat PO dan melakukan tawar-menawar
		3. Setelah tercapai kesepakatan maka *Supplier* mengirimkan Barang beserta dengan Faktur Pembelian
	4. *Bagian Gudang* dan *Bagian Pembelian* melakukan cek terhadap barang dan Faktur Pembelian	
Postcondition	Mendapatkan barang dan faktur pembelian sesuai dengan surat PO.	

Tabel 4.7 Skenario *Use Case* Laporan

Use Case	**Laporan**
Brief Description	*Use Case* ini memungkinkan *Kasir, Bagian Pembelian* dan *Bagian Gudang* untuk membuat Laporan Penjualan, Stok, Transfer Barang dan Pembelian
Actor	*Kasir, Bagian Gudang, Bagian Pembelian* dan *Manager*

Precondition	Rekap Data Pembelian, Penjualan dan Kartu Stok	
Main Flow	***Actor***	***System***
	1. *Kasir*, *Bagian Pembelian* dan *Bagian Gudang* masing-masing membuat Laporan berkala mengenai Laporan Penjualan Barang, Laporan Pembelian dan Laporan Stok Barang serta Transfer Barang	
		2. Laporan kemudian diserahkan pada *Manager* untuk diperiksa
		3. *Manager* memeriksa Laporan dan menghasilkan keputusan untuk perusahaan
	4. Laporan yang terkoreksi diserahkan duplikatnya sebagai arsip.	
Postcondition	Laporan yang terkoreksi kemudian diarsipkan dan menghasilkan keputusan.	

Tabel 4.8 *Use Case* Transfer Barang

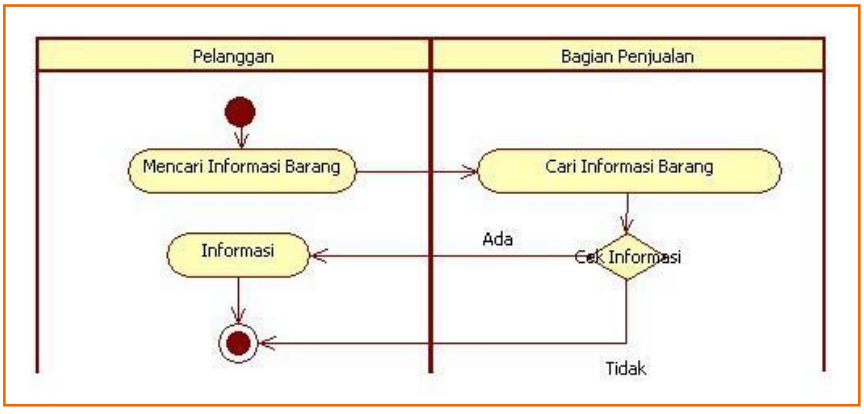
Use Case	**Laporan**	
Brief Description	*Use Case* ini memungkinkan *Bagian Gudang* untuk membuat Laporan Transfer Barang	
Actor	*Bagian Gudang*, *Manager*	
Precondition	Rekap Stok dan Transfer Barang	
Main Flow	***Actor***	***System***
	1. *Bagian Gudang* menerima surat pemindahan barang dari gudang cabang	
		2. Surat kemudian diteruskan

		pada *Manager* untuk di review
		3. *Manager* meng-acc surat
	4. *Bagian Gudang* mengirim barang sesuai dengan permintaan	
Postcondition	Barang dikirim dan dibuat laporan transfer barang	

4.1.2 *Activity Diagram* Berjalan

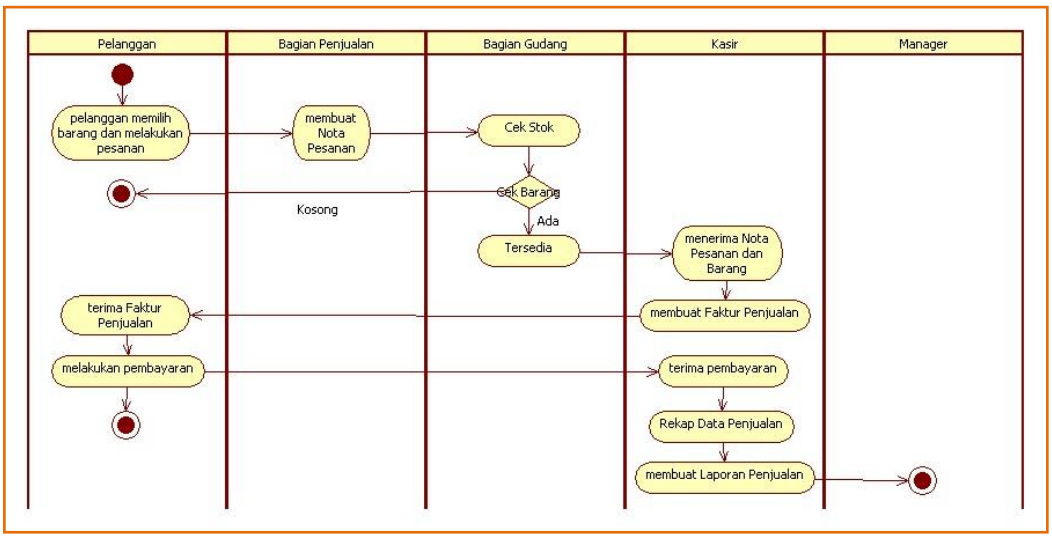
Activity diagram digunakan untuk menggambarkan kegiatan-kegiatan yang ada di dalam suatu sistem, dimana merupakan penggambaran aktivitas dari *case* yang ada pada *Use Cse Diagram*. Agar dapat lebih memahami tentang sistem yang akan dibuat, maka perlu dibuatkan *activity diagram* tentang sistem yang sedang berjalan, yaitu seperti yang ada di bawah ini:

Ini merupakan diagram aktivitas ketika pelanggan dating melakukan pencarian informasi sekaligus memilih barang yang akan di beli, yang mana mendapat bantuan dari bagian penjualan.



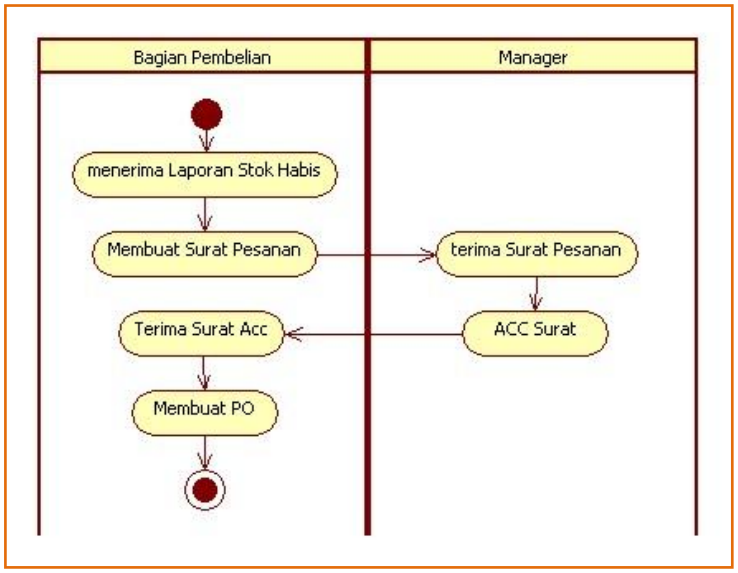
Gambar 4.2 *Activity Diagram* Mencari Informasi Barang

Diagram aktivitas berikut ini menjelaskan mengenai proses aktivitas dalam transaksi penjualan yang berjalan saat ini pada Delta Five.



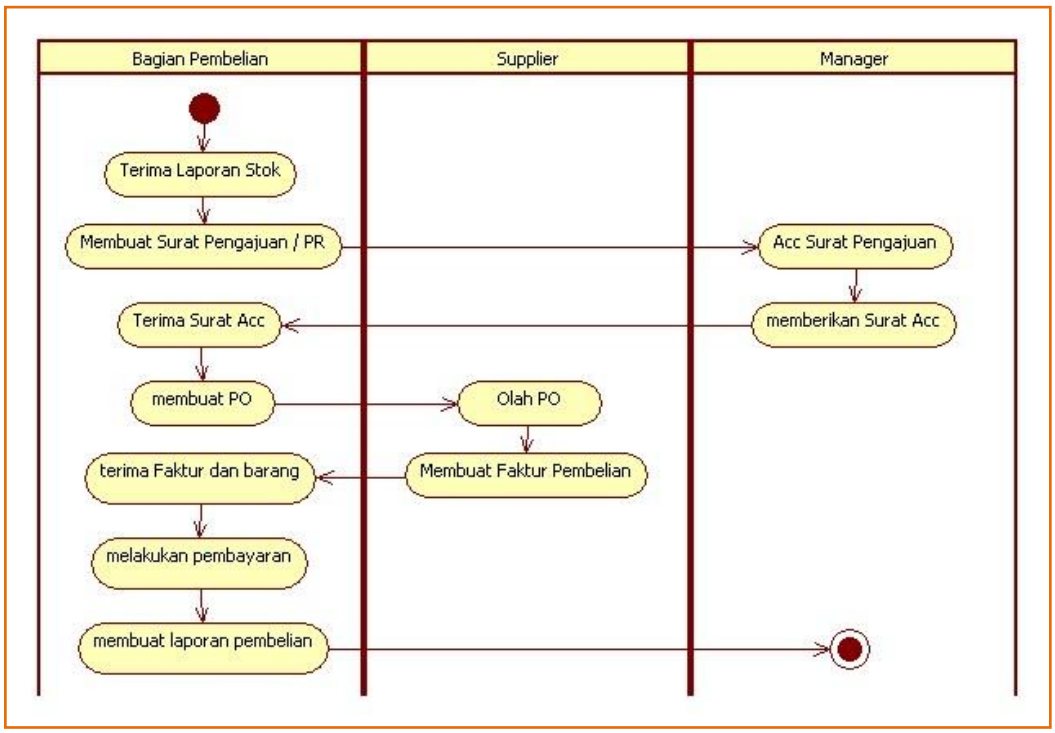
Gambar 4.3 *Activity Diagram* Penjualan Berjalan

Diagram aktivitas dibawah ini merupakan pemaparan kegiatan-kegiatan yang terjadi pada pembuatan surat pengajuan pembelian barang, surat pengajuan dibuat oleh bagian pembelian dan harus mendapat persetujuan *Manager*.



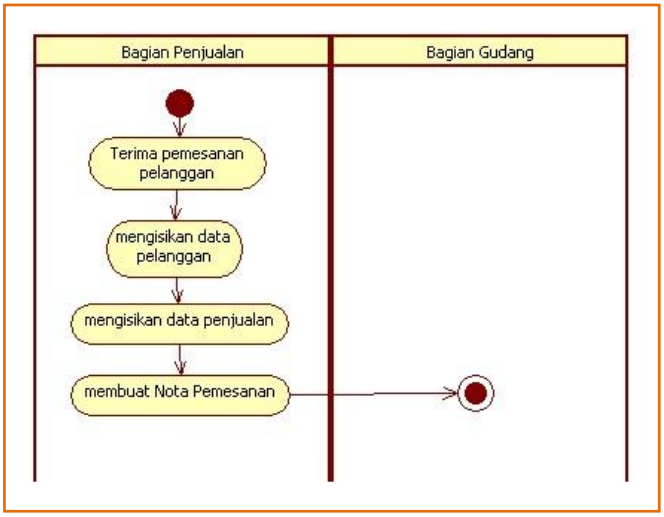
Gambar 4.4 *Activity Diagram* Surat *Purchase Requisition*

Diagram dibawah ini adalah diagram aktivitas yang menunjukkan proses pada transaksi pembelian yang berjalan pada Delta Five.



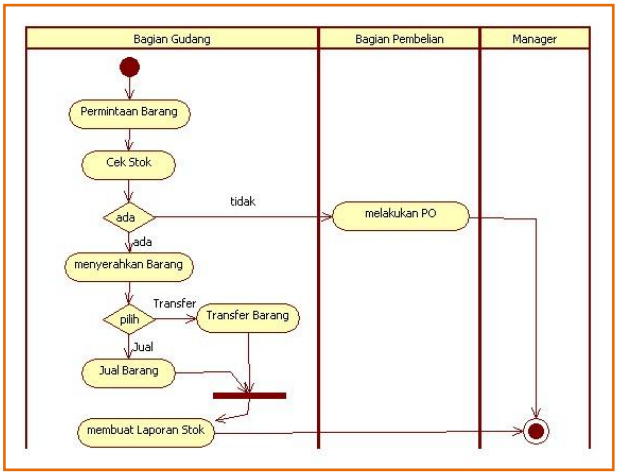
Gambar 4.5 *Activity Diagram* Pembelian Berjalan

Gambar 4.6 di bawah ini merupakan penjelasan mengenai kegiatan pemesanan barang yang dilakukan oleh pelanggan.



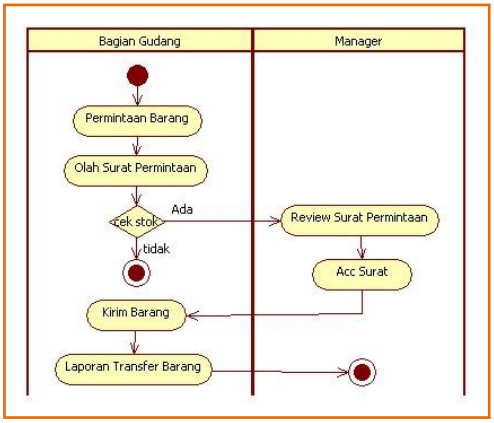
Gambar 4.6 *Activity Diagram* Pemesanan

Diagram aktivitas berikut ini merupakan diagram aktivitas pada kegiatan perhitungan stok barang di dalam gudang.



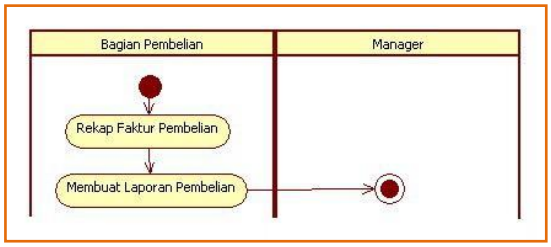
Gambar 4.7 *Actitity Diagram* Stok

Diagram berikut ini merupakan diagram aktivitas yang menyatakan proses dan kegiatan dalam melakukan transfer barang antar gudang yang berjalan.



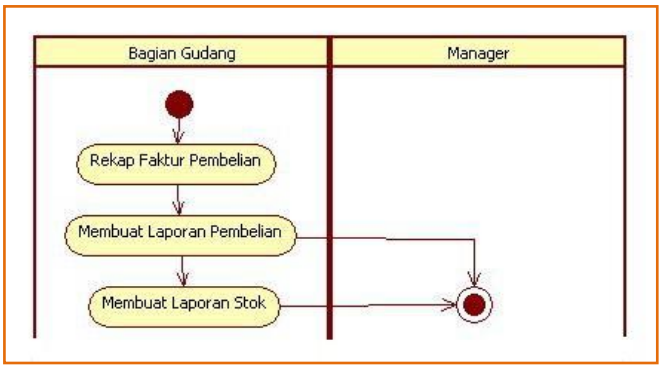
Gambar 4.8 *Activity Diagram* Transfer Barang

Diagram dibawah ini adalah diagram aktivitas untuk kegiatan pembuatan laporan baik laporan pembelian yang berjalan pada Delta Five.



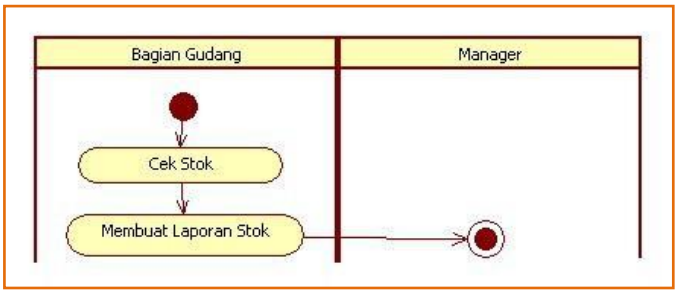
Gambar 4.9 *Activity Diagram* Laporan Pembelian Berjalan

Diagram dibawah ini adalah diagram aktivitas untuk kegiatan pembuatan laporan baik laporan penjualan yang berjalan pada Delta Five.



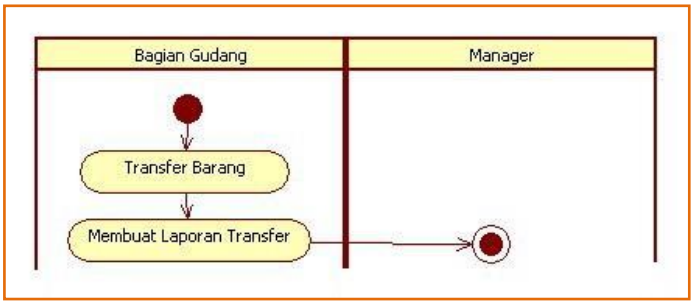
Gambar 4.10 *Activity Diagram* Laporan Penjualan Berjalan

Merupakan tampilan yang berfungsi untuk menampilkan laporan persediaan barang yang berjalan.



Gambar 4.11 *Activity Diagram* Laporan Persediaan Berjalan

Diagram dibawah ini adalah diagram aktivitas untuk menggambarkan kegiatan dalam pembuatan laporan transfer barang.



Gambar 4.12 *Activity Diagram* Laporan Transfer Berjalan

4.1.3 Evaluasi Sitem Yang Berjalan

Setelah penulis mengadakan penelitian, dan mengamati kegiatan yang berhubungan dengan objek penelitian, prosedur serta proses pengolahan data penjualan dan persediaan barang yang meliputi pembuatan dokumen-dokumen, bagian-bagian mana saja yang terlibat, serta pembuatan laporan-laporan, penulis menemukan beberapa kelemahan dalam sistem yang sedang berjalan pada saat ini.

Evaluasi terhadap kelemahan-kelemahan dari sistem penjualan dan persediaan barang yang sedang berjalan terlihat pada tabel 4.9 di bawah ini:

Tabel 4.9 Evaluasi Sistem Yang Berjalan

No.	Permasalahan	Worker	Solusi
1	Tidak tersedianya alat / sistem pengolahan data penjualan dan persediaan barang yang terkomputerisasi secara otomatis	Bagian Penjualan Kasir Bagian Gudang	Membangun sistem informasi *Inventory* yang terkomputerisasi sebagai alternatif baru dalam melakukan proses penjualan dan persediaan barang, sehingga dapat meng-efisien-kan transaksi dan kerja karyawan
2	Pimpinan kesulitan dalam mendapatkan informasi mengenai penjualan dan persediaan barang, karena harus mengecek langsung ke Bagian Penjualan dan Bagian Gudang	*Manager*	Membangun media penyimpanan informasi yang terkomputerisasi agar dapat memberikan informasi kepada *manager* tentang laporan penjualan dan persediaan. Hal ini diaplikasikan dengan membangun *Database* dengan sistem *Client-Server*.
3	Data transfer barang antar gudang tidak terdokumentasi dengan baik	Bagian Gudang	Membangun modul dalam aplikasi yang mampu mendokumentasikan alur transfer barang dari satu

			gudang kegudang lainnya.
4	Perusahaan sulit untuk menentukan waktu yang tepat dalam pemesanan (ROP) dan menentukan stok aman untuk tiap produk yang ada	*Manager* Bagian Gudang	Membangun aplikasi yang mampu melakukan perhitungan ROP (*Reorder Point*) serta *Safety Stock*.
5	Kesulitan dalam menentukan jumlah ROP yang harus dipesan	Bagian Gudang	Mengintegrasikan metode perhitungan EOQ (*Economic Order Quantity*) ke dalam aplikasi untuk membantu menghitung jumlah pemesanan minimum dengan tingkat harga yang paling minimal.
6	Faktur Penjualan masih berupa kertas dan menggunakan Kalkulator untuk proses perhitungannya	Kasir	Sistem terkomputerisasi dan otomatis serta mudah dicetak
7	Dokumen dan data sering hilang karena masih bersifat *Paper based*	Semua Bagian	Disimpan dalam *Database*

4.2 Perancangan Sistem

Analisis dan perancangan adalah serangkaian kegiatan yang selalu beriringan dalam setiap pengembangan *software*, sebagai sebuah hubungan sebab dan akibat yang memunculkan sebuah siklus hidup sistem. Walaupun dalam kenyataannya pengembangan sistem yang sederhana, aktivitas ini tidak tampak.

4.2.1 Tujuan Perancangan Sistem

Perancangan sistem merupakan suatu kegiatan pengembangan prosedur dan proses yang sedang berjalan untuk menghasilkan sesuatu yang baru atau memperbaharui sistem yang ada untuk meningkatkan efektifitas kerja, agar dapat memenuhi hasil yang diinginkan. Rancangan sistem yang baru, akan diterapkan

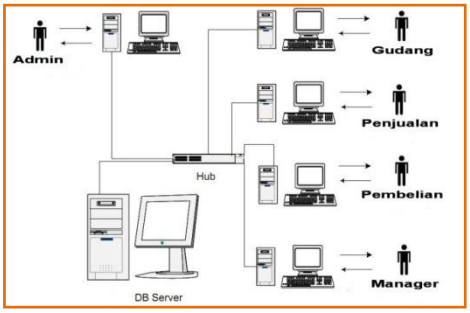
suatu kegiatan untuk menemukan dan mengembangkan metoda, prosedur dan proses suatu data agar tujuan dari suatu organisasi dapat tercapai.

Adapun tujuan dari tahap perancangan sistem ini adalah untuk menghasilkan perancangan sistem berupa pemodelan menggunakan pendekatan *object oriented*, pengolahan manajemen persediaan barang *multi warehouse* terutama untuk program aplikasi yang berbasis *client-server* sehingga dapat memperbaiki atau meningkatkan efisiensi kerja sistem yang sedang berjalan.

4.2.2 Gambaran Umum Perangkat Lunak

Aplikasi ini sendiri dibangun atas dasar kebutuhan akan pengelolaan *inventory* dengan kondisi memiliki lebih dari satu gudang penyimpanan barang, dan mampu menghasilkan perhitungan nilai EOQ. Sistem yang dikembangkan adalah pembuatan aplikasi untuk menghitung pembelian yang ekonomis agar persediaan yang tersedia tidak kurang dan tidak pula berlebihan. Selain itu juga dapat menentukan periode pemesanan, sehingga tidak sampai kehabisan stok.

Adapun pengguna dari sistem terbagi atas lima bagian yaitu, Admin Master (*system control*), Bagian Gudang, Bagian Penjualan, Bagian Pembelian dan *Manager*. Sedangkan perancangannya adalah menggunakan UML sebagai deskripsi dari sistem yang akan dibangun.



Gambar 4.13 Pengguna Sistem

4.2.2.1 Modul Perangkat Lunak

Modularitas adalah sebuah teori pemecahan suatu sistem menjadi sub sistem (*break down*) menjadi yang lebih kecil, yang biasa dikenal dengan modul. Untuk memudahkan dalam pengelolaan aplikasi ini, maka kelas-kelas akan dipecah kedalam empat modul tersendiri, yaitu:

Tabel 4.10 Modul Perangkat Lunak

No.	Modul	Keterangan
1	*InManagement*	Modul antaramuka utama yang mengontrol pemanggilan proses-proses pada sub modul-modulnya
2	*Inventory*	Modul *inventory* meliputi manipulasi data gudang, kategori barang, dan spesifikasi barang
3	*Transaksi*	Modul transaksi meliputi manipulasi data pembelian, penjualan, pemasok, dan pelanggan
4	*Management*	Modul *management* meliputi penyesuaian stok, transfer barang dan melihat isi gudang, laporan *inventory*, laporan transaksi, laporan *management*

4.2.2.2 Fitur Utama Perangkat Lunak

Perangkat lunak yang dibangun memiliki beberapa fitur-fitur pada umumnya yang berfungsi untuk memanipulasi keberadaan data di dalam *database*. Berikut ini penjelasan fitur-fitur yang terdapat dalam setiap modul:

Tabel 4.11 Fitur Utama Perangkat Lunak

No.	Modul	Fitur
1	InManagement	Halaman Utama perangkat lunak dan *setting* DB
2	*Inventory*	Gudang : *input, delete, update* Kategori Barang : *input, delete, update* Spesifikasi Barang : *input, delete, update*
3	Transaksi	Pembelian : *input, delete, update* Penjualan : *input, delete, update* Pemasok, Pelanggan: *input, delete, update* Pencarian : berdasarkan transaksi

4	*Management*	Penyesuaian Stok, Isi Gudang Transfer Barang : *input, delete, update* Pencarian : berdasarkan penyesuaian dan transfer barang
5	*Report*	Cetak laporan *inventory*, laporan transaksi, *management*

4.2.3 Kebutuhan Perangkat Lunak

Deskripsi kebutuhan dari perangkat lunak menjelaskan mengenai kebutuhan-kebutuhan baik Kebutuhan Antarmuka Eksternal, Fungsional maupun Non-Fungsional. Berikut ini kebutuhan perangkat lunak yang akan dibangun:

4.2.3.1 Kebutuhan Antarmuka Eksternal

Perangkat lunak yang dibangun membutuhkan perangkat lunak lain sebagai penunjang agar dapat berjalan sesuai dengan fungsinya. Kebutuhan tersebut yaitu:

a) Antarmuka Pemakai

Sebagai penunjang antarmuka pemakai dari perangkat lunak, diperlukan JAVA SDK (JRE dan JDK) untuk dapat menjalankan program.

b) Antarmuka Komunikasi

Untuk komunikasi antara *server* dan *client* akan menggunakan protokol TCP/IP atau *Wireless*. Oleh karena itu dibutuhkan perangkat keras RJ45 LAN *Card* pada setiap komputer yang masuk ke dalam sistem.

c) Aplikasi *Server*

Dibutuhkan *server* untuk memusatkan proses dari perangkat lunak, yaitu *database server*. Kebutuhan *database server* pada aplikasi ini dapat menggunakan MySQL. Aplikasi server ini akan mengatur *request* ke *server* dan juga respon terhadap *request* dari *server* ke *client*.

4.2.3.2 Kebutuhan Fungsional

Merupakan kebutuhan secara fungsional yang harus dipenuhi oleh perangkat lunak yang akan dibangun. Kebutuhan fungsional tersebut akan dideskripsikan dalam bentuk tabel, sebagai berikut:

Tabel 4.12 Deskripsi Kebutuhan Fungsional

Kode	Nama Kebutuhan	Deskripsi
UC-F-001	*Login*	Untuk mengakses terhadap sistem
UC-F-002	*Input Data*	Memasukkan data-data ke dalam *Database*
UC-F-003	*Update Data*	Memperbaharui data dalam *database*
UC-F-004	*Delete Data*	Menghapus data dari *database* sistem
UC-F-005	DB *Setting*	Melakukan konfigurasi dan seting *database*
UC-F-006	Penjualan	Fungsi untuk melakukan penjualan
UC-F-007	Pembelian	Fungsi untuk transaksi pembelian
UC-F-008	Cari Transaksi	Mencari transaksi penjualan dan pembelian
UC-F-009	Transfer Barang	Melakukan proses transfer barang
UC-F-010	Cari Transfer	Mencari data transfer barang
UC-F-011	EOQ	Melakukan peramalan stok
UC-F-012	Grafik	Melihat Grafik Penjualan
UC-F-013	Stok	Melihat stok barang di gudang
UC-F-014	Penyesuaian Stok	Menyesuaikan stok fisik dengan data
UC-F-015	Laporan	Membuat dan menerima laporan

4.2.3.3 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional mencakup fungsi-fungsi yang membantu sistem untuk berjalan dengan baik serta dapat digunakan dengan mudah.

Tabel 4.13 Deskripsi Kebutuhan Non-Fungsional

Kode	Nama Kebutuhan	Deskripsi
NF-001	*User Friendly*	Sistem mudah digunakan
NF-002	*Confirm Alert*	Peringatan sebagai konfirmasi *User*
NF-003	*Data Validation*	Mengecek data yang di input, sesuai atau tidak dengan ketentuan
NF-004	*Desktop Base*	Sistem dibangun berbasis desktop

NF-005	Menggunakan bahasa Indonesia dan Inggris	Penggunaan bahasa Indonesia dan Inggris dalam penulisan menu maupun lainnya.

4.2.4 Kandidat Kelas

Pendefinisian kandidat kelas digunakan untuk menjelaskan objek-objek dalam sistem. Dimana kelas-kelas mendefinisikan model data dan esensi sistem.

Tabel 4.14 Kandidat Kelas

No	Identifikasi Objek	Nama Objek	Ditolak/Diterima (*)	Alasan
1	Objek Fisik	-	-	-
2	Transaksi	Penjualan Detail_Penjualan Pembelian Detail_Pembelian	2 2 2 2	Dalam sistem Dalam sistem Dalam sistem Dalam sistem
3	*Inventory*	Barang Barang_diGudang	2 2	Dalam sistem Dalam sistem
4	Manajemen	Transfer Detai_Transfer Penyesuaian Detail_Penyesuaian	2 2 2 2	Dalam sistem Dalam sistem Dalam sistem Dalam sistem
5	Peranan	Admin Pelanggan, Pemasok *Manager* Bagian Gudang Bagian pembelian Bagian Penjualan	2 1 1 2 2 2	Pengontrol Tidak perlu Ada hubungan PenggunaSistem Pengguna sistem Pengguna sistem Pengguna sistem

Keterangan: (*) 1 – Ditolak 2 – Diterima

4.2.5 *Use Case Diagram*

Actor dan *use case* ditentukan atas dasar fungsi-fungsi dalam sistem. Selanjutnya *use* case menyediakan nilai hasil kepada *actor*. Atas dasar analisis kandidat kelas diatas setidaknya ada lima (5) *actor* yang berhubungan dengan sistem yaitu Bagian Pembelian, Bagian Penjualan, Bagian Gudang, *Admin Master* dan *Manager*.

4.2.5.1 *Use Case Diagram* Usulan

Use Case Diagram menggambarkan fungsionalitas dari sebuah sistem (apa fungsinya), yang merepresentasikan sebuah interaksi antara *actor* dengan sistem (sebuah pekerjaan), misalnya menambah data atau membuat laporan. Elemen-elemennya adalah: *actor*, *use case*, dan hubungan antar objek.

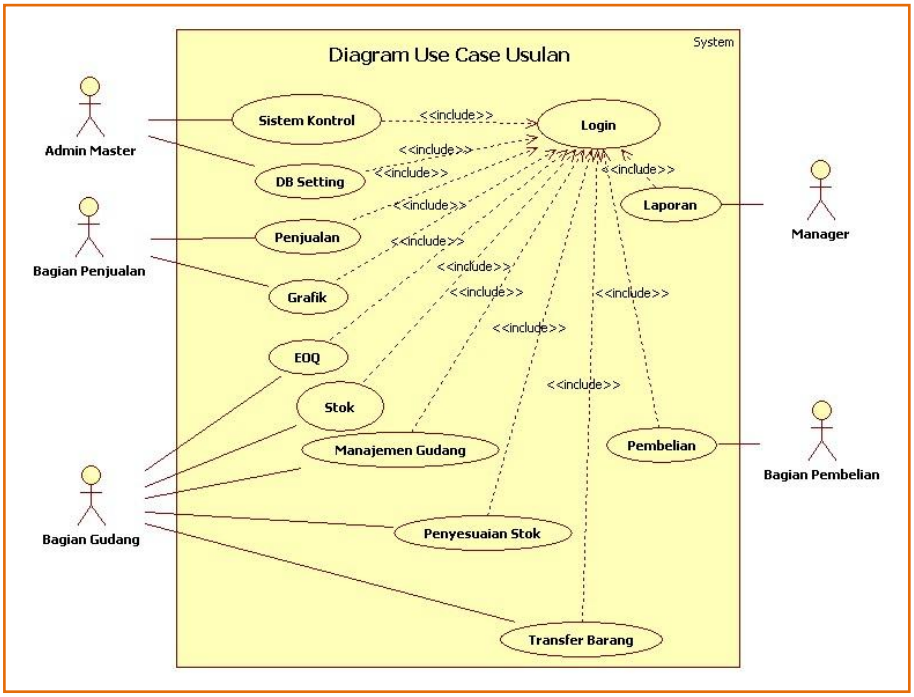
1. *Actor* adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.
2. *Use case* adalah sebuah tidakan atau unit fungsional dari sebuah sistem.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang umum. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behavior*-nya sendiri.

Tabel 4.15 Definisi Aktor

No.	Aktor	Deskripsi
1	*Admin*	Merupakan Admin yang memiliki kewenangan penuh atas seluruh akses terhadap sistem
2	*Manager*	Aktor yang menerima Laporan
3	Gudang	Aktor yang memiliki akses terhadap modul *Inventory*
4	Penjualan	Keterlibatannya adalah dalam proses penjualan barang
5	Pembelian	Aktor yang terlibat di dalam proses pembelian barang

Berikut ini adalah gambar dari model *Use Case Diagram Inventory Multi Warehouse* yang penulis usulkan, yang digambarkan secara umum sebagai berikut:



Gambar 4.14 *Use Case Diagram* Yang Diusulkan

Sementara itu, berikut adalah tabel yang mendeskripsikan *use case* usulan.

Tabel 4.16 Daftar Deskripsi *Use Case* Usulan

Kode Use Case	Nama Use Case	Deskripsi
UC-U-001	*Login*	Untuk mengakses terhadap sistem
UC-U-002	*Input Data*	Memasukkan data-data ke dalam *Database*
UC-U-003	*Update Data*	Memperbaharui data dalam *database*
UC-U-004	*Delete Data*	Menghapus data dari *database* sistem
UC-U-005	DB *Setting*	Melakukan konfigurasi dan seting *database*
UC-U-006	Penjualan	Fungsi untuk melakukan penjualan
UC-U-007	Pembelian	Fungsi untuk transaksi pembelian
UC-U-008	Cari Transaksi	Mencari transaksi penjualan dan pembelian
UC-U-009	Transfer Barang	Melakukan proses transfer barang
UC-U-010	Cari Transfer	Mencari data transfer barang
UC-U-011	EOQ	Melakukan peramalan stok
UC-U-012	Grafik	Melihat Grafik Penjualan
UC-U-013	Stok	Melihat stok barang di gudang
UC-U-014	Penyesuaian Stok	Menyesuaikan stok fisik dengan data
UC-U-015	Laporan	Membuat dan menerima laporan

4.2.5.2 Dokumentasi Skenario *Use Case*

Setiap *use case* di atas harus dideskripsikan dalam dokumen yang disebut dengan dokumen *flow of event*. Dokumen ini merupakan definisi apa yang harus dilakukan oleh sistem ketika *actor* mengaktifkan *use case*. Berikut ini adalah dokumentasi *use case* untuk *Use Case Diagram Inventory Multi Warehouse* yang diusulkan oleh penulis.

Tabel 4.17 Skenario *Use Case Login*

<table>
<tr><th>Use Case</th><th colspan="2">Login</th></tr>
<tr><td>Brief Description</td><td colspan="2">Use Case ini memungkinkan Admin terdaftar melakukan akses terhadap sistem</td></tr>
<tr><td>Actor</td><td colspan="2">Admin / User (pembelian, penjualan, manager, gudang)</td></tr>
<tr><td>Precondition</td><td colspan="2">Admin membuka aplikasi Login</td></tr>
<tr><td rowspan="4">Main Flow</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Admin menginputkan Username dan Password</td><td></td></tr>
<tr><td></td><td>2. Verifikasi username dan password di dalam database</td></tr>
<tr><td></td><td>3. Memberikan informasi login valid atau tidak, jika ya maka otomatis mengakses halaman yang diminta, jika tidak akan keluar pesan gagal login.</td></tr>
<tr><td>Postcondition</td><td colspan="2">Admin mengakses aplikasi yang dibutuhkan</td></tr>
</table>

Tabel 4.18 Skenario *Use Case Input Data*

Use Case	**Input Data**
Brief Description	*Use Case* ini memungkinkan semua proses penginputan data ke dalam *database*.
Actor	*Admin*
Precondition	Menu *Login*

Main Flow	***Actor***	***System***
	1. Admin login	
		2. Cek *Login* Valid atau Tidak
		3. Menampilkan Menu Utama
	4. *Input Data*	
		5. Verifikasi data input sukses
Postcondition	*Database* terupdate dengan penambahan data baru	

Tabel 4.19 Skenario *Use Case Update Data*

Use Case	**Update Data**	
Brief Description	*Use Case* ini memungkinkan user melakukan pengubahan data yang telah tersimpan sebelumnya	
Actor	*Admin*	
Precondition	Menu *Login*	
Main Flow	***Actor***	***System***
	1. *Admin Login*	
		2. Cek *Login*
		3. Menampilkan Menu Utama
		4. Cari Data untuk di *edit*
	5. Update Data	
		6. Komparasi dan Cek kesesuaian data
Postcondition	Data dalam *database* berubah atau terupdate dengan yang baru	

Tabel 4.20 Skenario *Use Case Delete Data*

Use Case	**Delete Data**	
Brief Description	*Use Case* ini memungkinkan user melakukan penghapusan data	
Actor	*Admin*	
Precondition	Menu *Login*	
Main Flow	***Actor***	***System***
	1. *Admin Login*	
		2. Cek *Login*
		3. Menampilkan Menu Utama
		4. Cari Data untuk di hapus

		5. Cek keberadaan data
		6. Verifikasi penghapusan
Postcondition	Data terhapus dari *database*	

Tabel 4.21 Skenario *Use Case DB Setting*

Use Case	**DB Setting**	
Brief Description	*Use Case* ini memungkinkan *Master Admin* melakukan *setting database*	
Actor	*Admin Master*	
Precondition	Menu *Login*	
Main Flow	***Actor***	***System***
	1. *Login*	
		2. Cek *Login*
		3. Menampilkan Menu Utama
	4. Pilih menu DB *Setting*	
	5. Melakukan Konfigurasi DB	
		6. Koneksi dengan sistem
		7. Validasi koneksi sukses
Postcondition	DB terkoneksi dengan sistem, dan mendapatkan file konfigurasi	

Tabel 4.22 Skenario *Use Case* Penjualan

Use Case	**Penjualan**	
Brief Description	*Use Case* ini memungkinkan untuk melakukan transaksi penjualan barang	
Actor	*Penjualan*	
Precondition	*Menu Login*	
Main Flow	***Actor***	***System***
	1. *Login*	
		2. Cek *Login*
		3. Menampilkan Menu Utama
	4. Pilih Menu Transaksi – Sub Menu Penjualan	
		5. Input data penjualan
		6. Klik Tambah

		7. Klik Simpan
	8. Keluar	
Postcondition	Transaksi penjualan tersimpan di dalam *database* dan dipakai untuk proses perhitungan di kasir	

Tabel 4.23 Skenario *Use Case* Pembelian

Use Case	**Penjualan**	
Brief Description	*Use Case* ini memungkinkan untuk melakukan transaksi pembelian barang	
Actor	*Pembelian*	
Precondition	*Menu Login*	
Main Flow	***Actor***	***System***
	1. *Login*	
		2. Cek *Login*
		3. Menampilkan Menu Utama
	4. Pilih Menu Transaksi – Sub Menu Pembelian	
		5. Input data pembelian
		6. Klik Tambah
		7. Klik Simpan
	8. Keluar	
Postcondition	Transaksi pembelian barang tersimpan dalam *database*	

Tabel 4.24 Skenario *Use Case* Cari Transaksi

Use Case	**Cari Transaksi**	
Brief Description	*Use Case* ini memungkinkan mencari *history* transaksi penjualan dan pembelian yang terjadi	
Actor	*Admin (penjualan atau pembelian)*	
Precondition	Menu *Login*	
Main Flow	***Actor***	***System***
	1. *Login*	
		2. Cek *Login*
		3. Menampilkan Menu Utama
	4. Pilih Menu Transaksi – Sub Menu Pencarian	
		5. *Input* tanggal atau no

		transaksi
		6. *Filtering*
		7. Klik OK
		8. *Edit* data jika diperlukan
Postcondition	Mendapatkan data hasil pencarian berdasarkan tanggal atau no transaksi	

Tabel 4.25 Skenario *Use Case* Transfer Barang

Use Case	**Transfer Barang**	
Brief Description	*Use Case* ini memungkinkan perusahaan melakukan manajemen transfer barang antar gudang yang dimiliki	
Actor	*Gudang*	
Precondition	Menu *Login*	
Main Flow	***Actor***	***System***
	1. *Login*	
		2. Cek *Login*
		3. Menampilkan Menu Utama
	4. Pilih Menu Manajemen – Sub Menu Transfer Barang	
		5. *Input* Gudang Asal dan Gudang Tujuan
		6. *Input* Data Barang yang di transfer
		7. Klik Tambah
		8. Klik Simpan
	9. Keluar	
Postcondition	Proses transfer barang tersimpan	

Tabel 4.26 Skenario *Use Case* Cari Transfer

Use Case	**Cari Transfer**	
Brief Description	*Use Case* ini memungkinkan pencarian terhadap *history* transfer barang yang dilakukan	
Actor	*Gudang*	
Precondition	Menu *Login*	
Main Flow	***Actor***	***System***

	1. *Login*	
		2. Cek *Login*
		3. Menampilkan Menu Utama
	4. Pilih Menu Manajemen – Sub Menu Pencarian Transfer	
		5. *Input* tanggal atau no transfer
		6. *Filtering*
		7. Klik OK
		8. *Edit* data jika diperlukan
Postcondition	*Pelanggan* mendapatkan informasi barang untuk membantu dalam mengambil keputusan transaksi.	

Tabel 4.27 Skenario *Use Case* EOQ

Use Case	**EOQ**	
Brief Description	*Use Case* ini memungkinkan melakukan perhitungan EOQ	
Actor	*Gudang*	
Precondition	Menu *Login*	
Main Flow	***Actor***	***System***
	1. *Login*	
		2. Cek *Login*
		3. Menampilkan Menu Utama
	4. Pilih Menu Persediaan – Sub Menu Barang	
		5. *Input* data Biaya Pesan, *Rate* Biaya Simpan, Kebutuhan/tahun, Kebutuhan Maks/hari, *Lead Time*
		6. Klik Simpan
Postcondition	Mendapatkan Nilai EOQ dan ROP dan *Safety Stock*	

Tabel 4.28 Skenario *Use Case* Grafik

Use Case	**Grafik**	
Brief Description	*Use Case* ini memungkinkan dalam melihat grafik penjualan barang	
Actor	*Penjualan*	
Precondition	Menu *Login*	
Main Flow	***Actor***	***System***
	1. *Login*	
		2. Cek *Login*
		3. Menampilkan Menu Utama
	4. Pilih Menu Laporan – Sub Menu Grafik	
Postcondition	Menampilkan grafik penjualan barang	

Tabel 4.29 Skenario *Use Case* Stok

Use Case	**Stok**	
Brief Description	*Use Case* ini memungkinkan untuk melihat stok barang di dalam masing-masing gudang	
Actor	*Manager; Gudang*	
Precondition	Menu *Utama*	
Main Flow	***Actor***	***System***
	1. *Login*	
		2. Cek *Login*
		3. Menampilkan Menu Utama
	4. Pilih Menu Manajemen – Sub Menu Stok Barang	
	5. Keluar	
Postcondition	Menampilkan informasi stok tiap gudang	

Tabel 4.30 Skenario *Use Case* Penyesuaian Stok

Use Case	**Penyesuaian Stok**
Brief Description	*Use Case* ini memungkinkan melakukan penyesuaian stok yang ada, yaitu dengan melakukan cek pada masing-masing stok gudang
Actor	*Gudang*

Precondition	Menu *Login*	
Main Flow	***Actor***	***System***
	1. *Login*	
		2. Cek *Login*
		3. Menampilkan Menu Utama
	4. Pilih Menu Manajemen – Sub Menu Penyesuaian Stok	
		5. *Input* data tanggal dan status barang
		6. Pilih gudang
		7. Klik Tambah
		8. Klik Simpan
Postcondition	Data stok menjadi sesuai dan akurat	

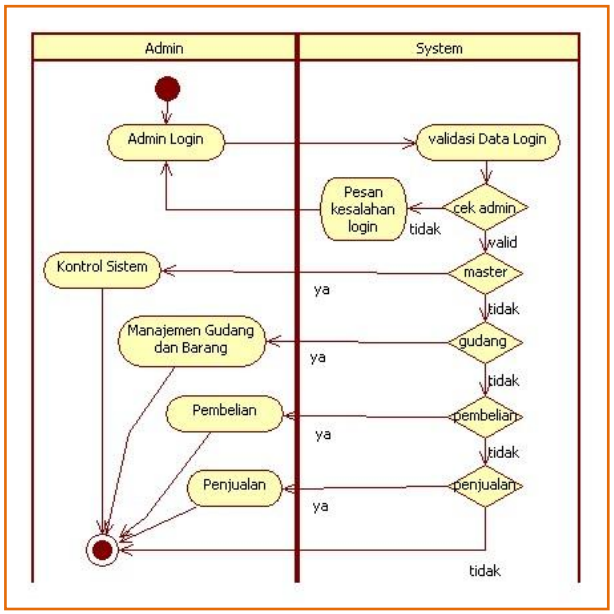
Tabel 4.31 Skenario *Use Case* Laporan

Use Case	**Laporan**	
Brief Description	*Use Case* ini memungkinkan untuk melihat Laporan-laporan	
Actor	*Manager, Pembelian, Penjualan, Gudang, Master*	
Precondition	Menu *Login*	
Main Flow	***Actor***	***System***
	1. *Login*	
		2. Cek *Login*
		3. Menampilkan Menu Utama
	4. Pilih Menu Laporan	
		5. Pilih aksi untuk laporan
	6. Keluar	
Postcondition	Menampilkan Laporan	

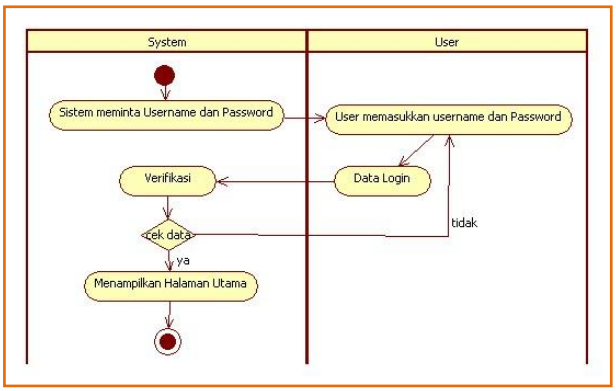
4.2.6. *Activity Diagram*

Activity diagram digunakan untuk menggambarkan kegiatan-kegiatan yang ada di dalam sistem. Agar lebih memahami sistem yang akan dibuat, maka perlu dibuatkan *activity diagram* tentang sistem, yaitu seperti yang ada di bawah ini:

Dua (2) Diagram berikut merupakan diagram aktivitas yang menjelaskan kegiatan *Login* terhadap sistem dalam beberapa tingkatan hak akses, dapat terlihat dari sistem *login* yang dilakukan setiap bagian memiliki modul masing-masing untuk dijalankan. Sementara Gambar 4.16 adalah penjelasan detail dari kegiatan *Login* kedalam sistem yang dilakukan oleh masing-masing admin.

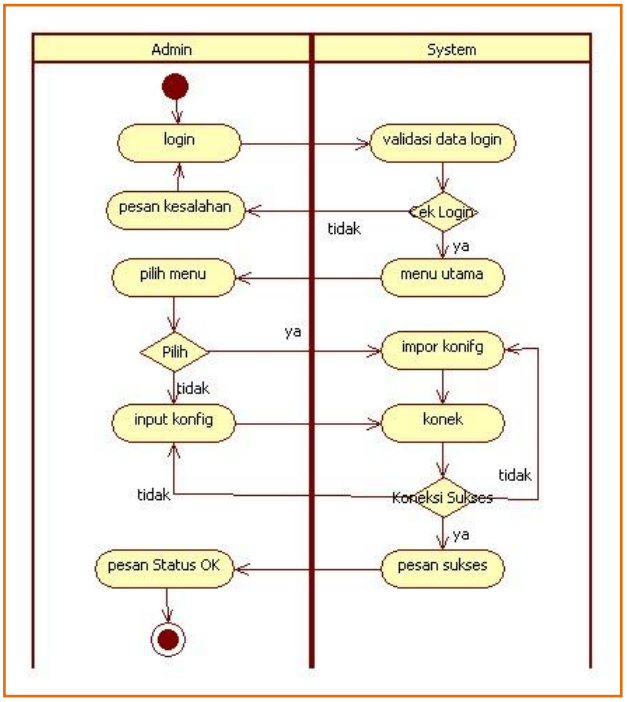


Gambar 4.15 *Activity Diagram* Aktivitas Utama



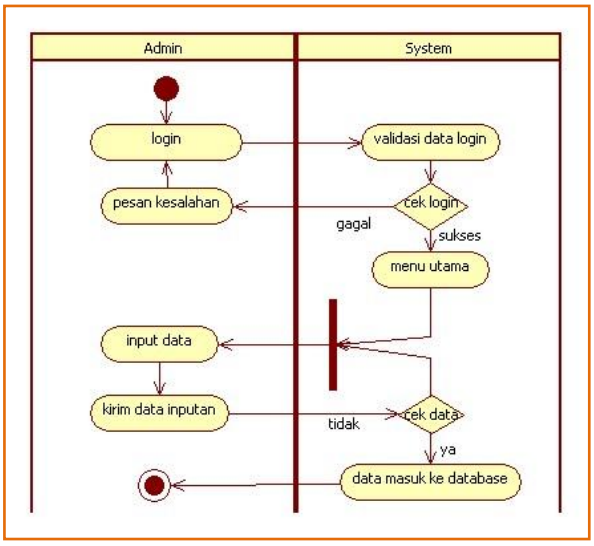
Gambar 4.16 *Activity Diagram Login*

Diagram di bawah ini adalah diagram aktivitas dalam menyeting konfigurasi *database* yang digunakan oleh aplikasi.



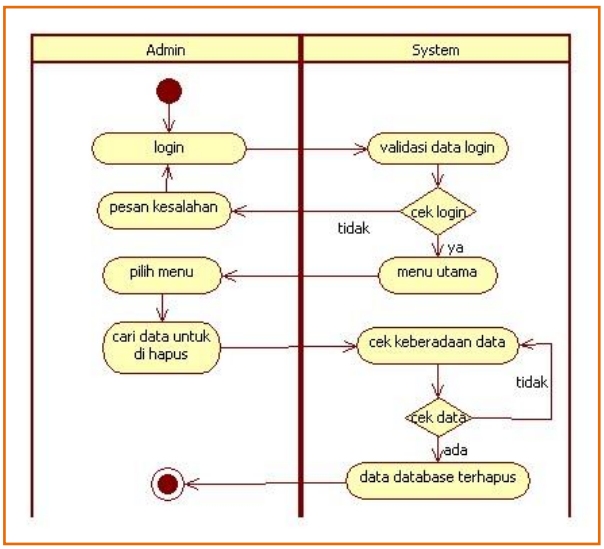
Gambar 4.17 *Activity Diagram DB Setting*

Diagram aktivitas berikut ini adalah penjelasan mengenai kegiatan pemasukan data ke dalam sistem dan *database*.



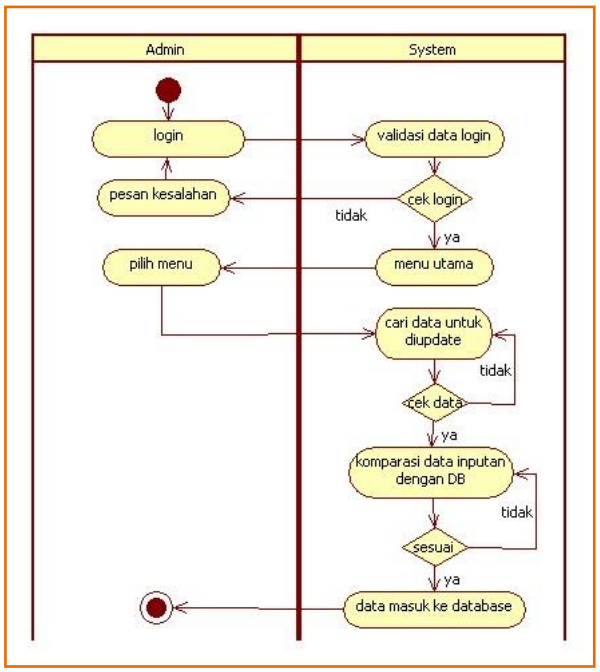
Gambar 4.18 *Activity Diagram Input Data*

Gambar dibawah ini merupakan diagram aktivitas yang menjelaskan mengenai kegiatan menghapus data dari dalam sistem.



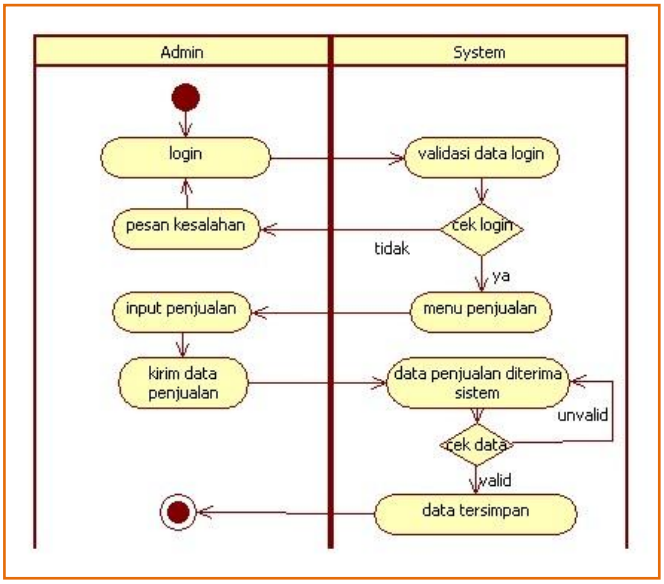
Gambar 4.19 *Activity Diagram Delete Data*

Diagram dibawah ini adalah diagram aktivitas yang menjelaskan kegiatan dalam peng-*updatean data*.



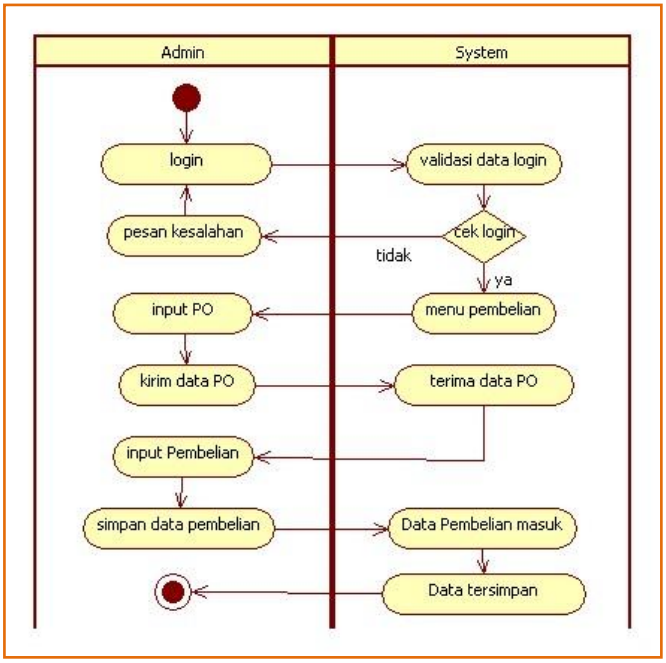
Gambar 4.20 *Activity Diagram Update Data*

Diagram berikut ini merupakan kegiatan pada proses transaksi penjualan barang yang diusulkan.



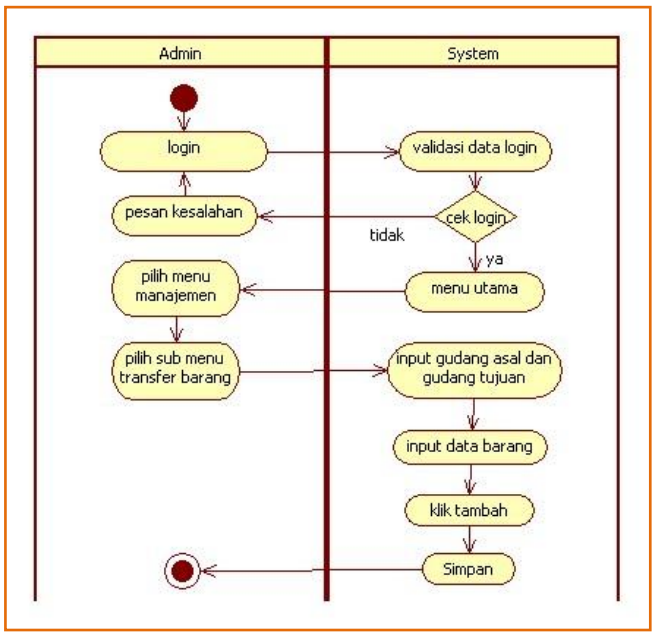
Gambar 4.21 *Activity Diagram* Penjualan

Diagram dibawah ini adalah penjelasan dari kegiatan transaksi pembelian yang diusulkan oleh penulis.



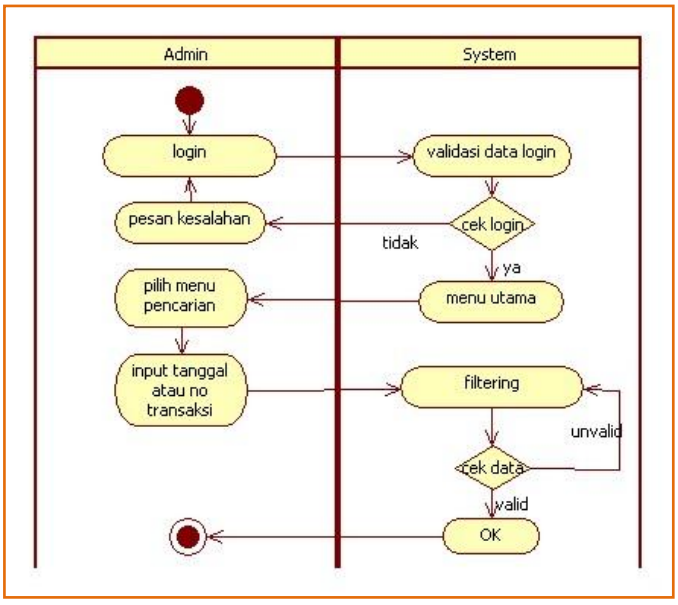
Gambar 4.22 *Activity Diagram* Pembelian

Diagram aktivitas dibawah ini merupakan pemaparan kegiatan untuk aktivitas transfer barang antar gudang yang diusulkan oleh penulis.



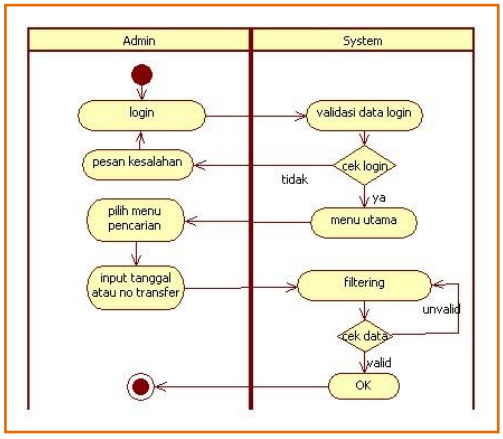
Gambar 4.23 *Activity Diagram* Transfer Barang

Diagram dibawah ini adalah pengambaran kegiatan pencarian data terhadap kegiatan transaksi penjualan maupun pembelian.



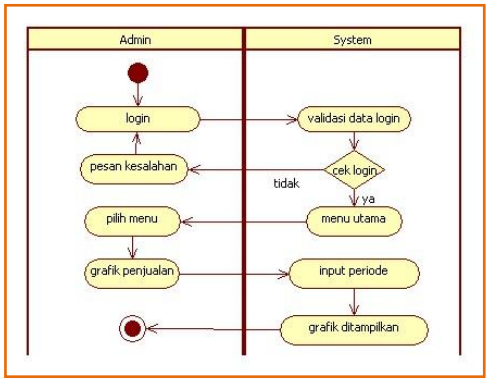
Gambar 4.24 *Activity Diagram* Cari Transaksi

Gambar berikut adalah penggambaran diagram aktivitas untuk kegiatan pencarian proses transfer barang antar gudang.



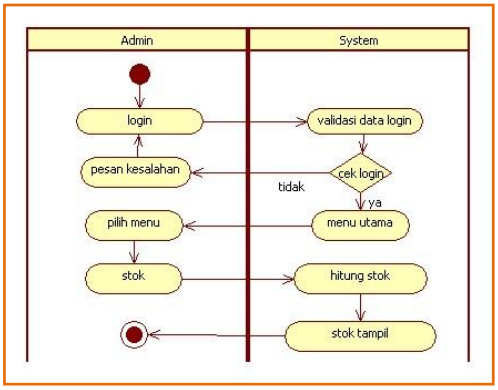
Gambar 4.25 *Activity Diagram* Cari Transfer

Diagram aktivitas berikut ini adalah untuk menampilkan grafik penjualan.



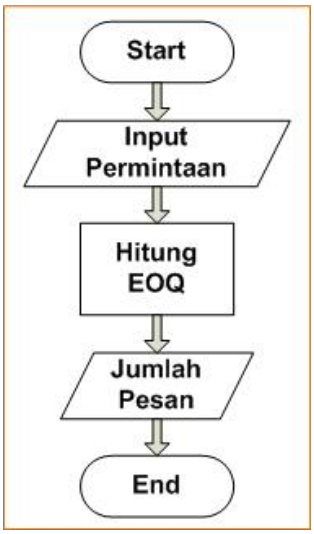
Gambar 4.26 *Activity Diagram* Grafik

Gambar dibawah ini adalah penjelasan mengenai kegiatan untuk melihat dan menampilkan daftar stok barang.



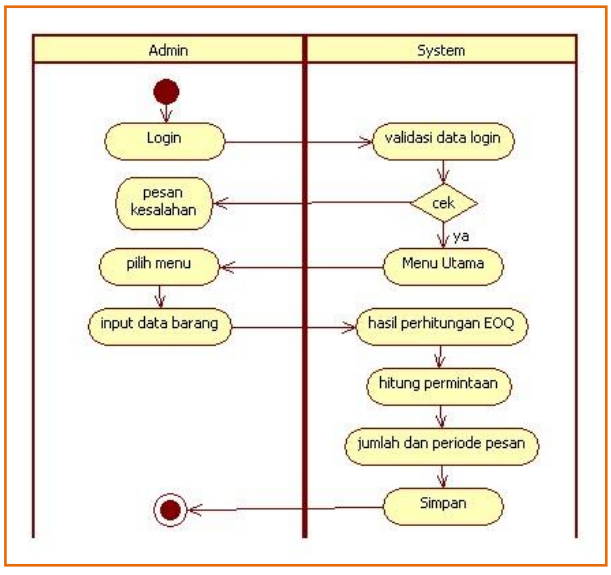
Gambar 4.27 *Activity Diagram* Stok

Untuk menghitung pemesanan barang ini, penulis menggunakan metode EOQ. Berikut adalah gambar *flowchart* untuk perhitungan pemesanan barang menggunakan metode EOQ.



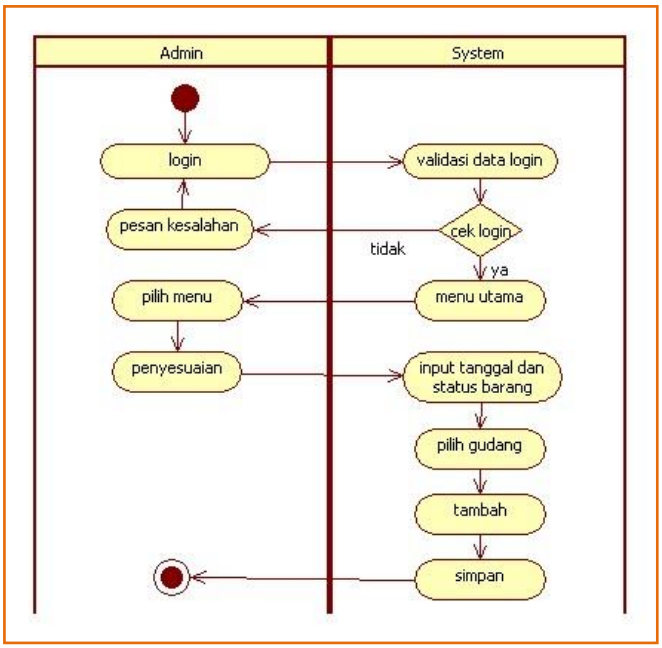
Gambar 4.28 *Flowchart* Metode EOQ

Gambar dibawah ini merupakan gambar proses pada kegiatan perhitungan Metode EOQ, berikut adalah diagram aktivitas dari Metode EOQ tersebut.



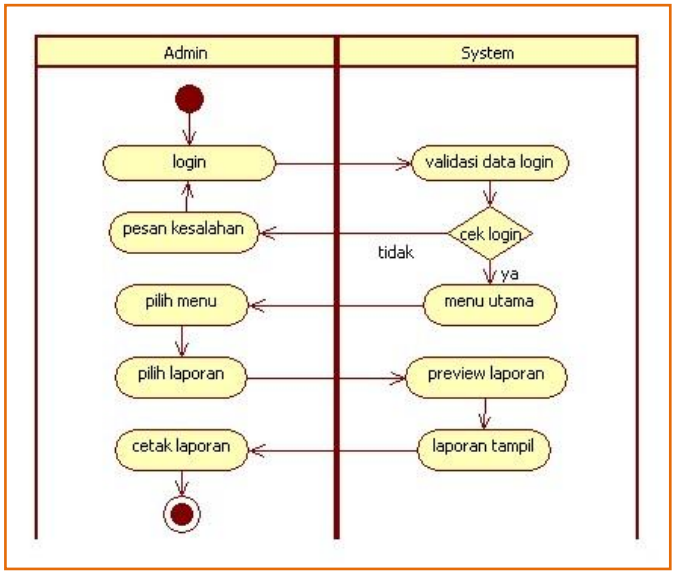
Gambar 4.29 *Activity Diagram* EOQ

Diagram di bawah ini merupakan diagram aktivitas untuk proses penyesuaian stok yang terdapat di dalam gudang.



Gambar 4.30 *Activity Diagram* Penyesuaian Stok

Gambar dibawah ini merupakan langkah-langkah kegiatan dalam diagram aktivitas membuat Laporan.

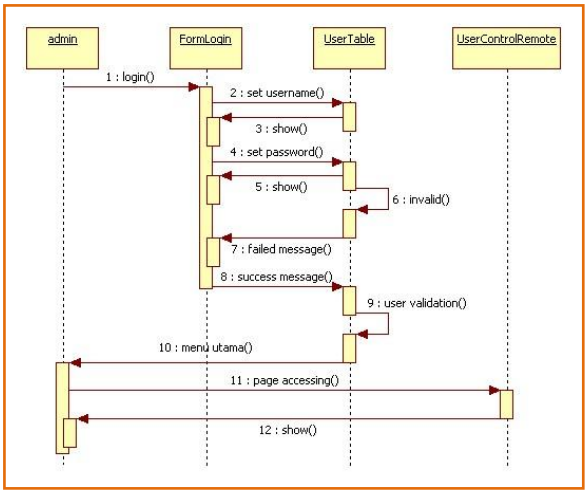


Gambar 4.31 *Activity Diagram* Laporan

4.2.7 *Sequence Diagram*

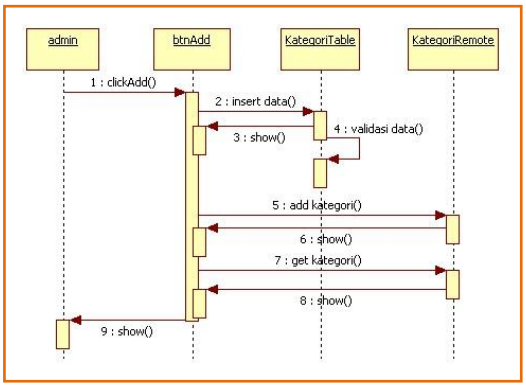
Adi Nugroho (2005:92) *sequence diagram* adalah *interaction diagram* yang memperlihatkan event-event yang berurutan sepanjang berjalannya waktu. Masing-masing *sequence diagram* akan menggambarkan aliran-aliran pada suatu *use case*.

Berikut ini adalah penggambaran diagram *sequence* untuk proses *Login* terhadap sistem.



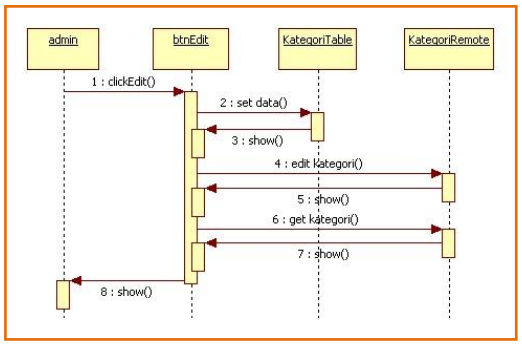
Gambar 4.32 *Sequence Diagram Login*

Diagram di bawah ini merupakan diagram *sequence* untuk proses peng-inputan data kedalam sistem.



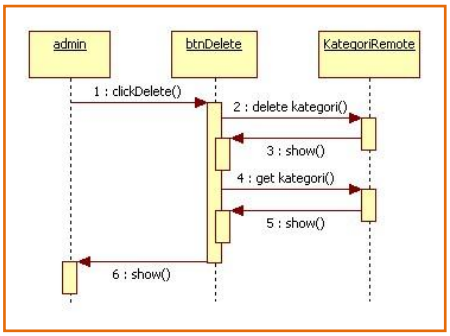
Gambar 4.33 *Sequence Diagram Input Data*

Diagram untuk proses *update data*, dapat dilihat seperti pada gambar dibawah ini.



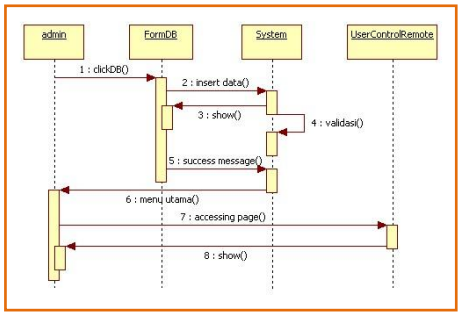
Gambar 4.34 *Sequence Diagram Update Data*

Untuk proses penghapusan data dari *database*, maka berikut ini merupakan langkah-langkahnya yang digambarkan dalam *sequence diagram.*



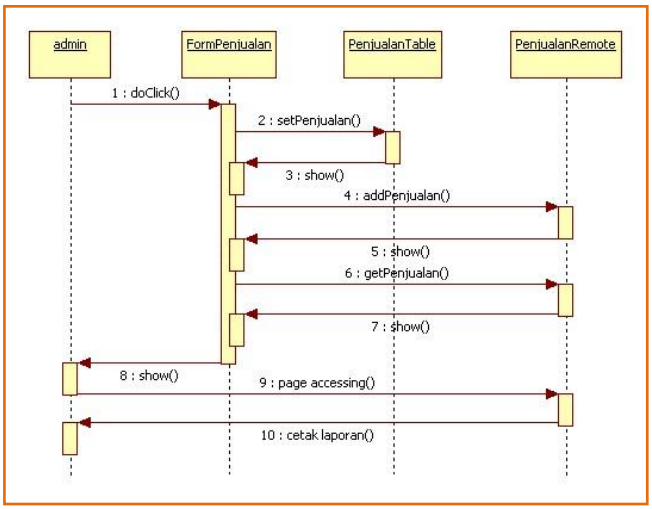
Gambar 4.35 *Sequence Diagram Delete Data*

Sequence diagram berikut ini merupakan penggambaran terhadap proses penyetingan *Database*.



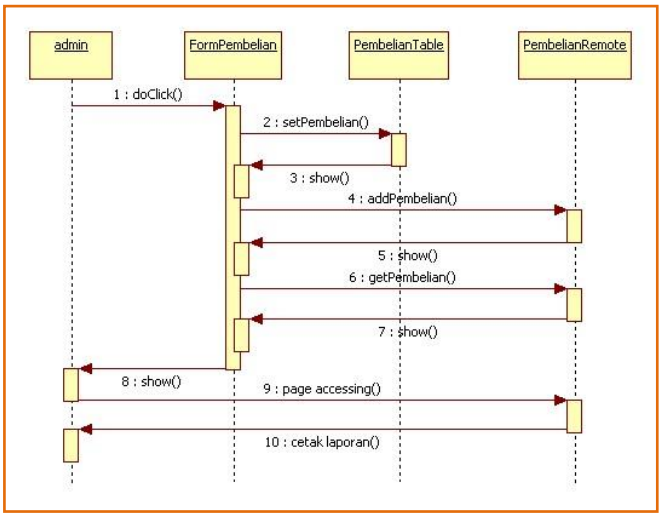
Gambar 4.36 *Sequence Diagram DB Setting*

Gambar dibawah ini merupakan *sequence diagram* untuk proses transaksi penjualan barang yang diusulkan oleh penulis.



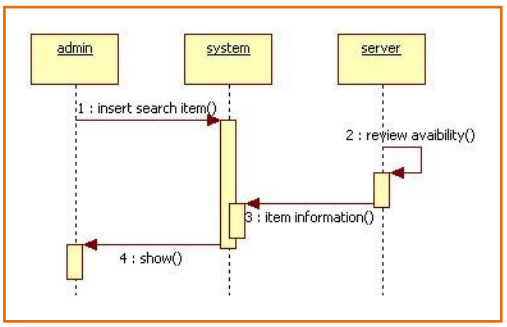
Gambar 4.37 *Sequence Diagram* Penjualan

Gambar diagram ini merupakan penggambaran diagram *sequence* untuk melakukan proses transaksi pembelian barang.



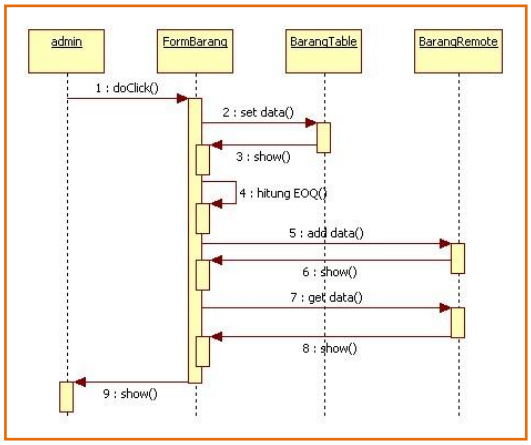
Gambar 4.38 *Sequence Diagram* Pembelian

Dalam setiap aplikasi akan dibuat sebuah fasilitas pencarian yang berfungsi untuk mencari *history* data, berikut ini merupakan *sequence diagram* untuk proses pencarian data.



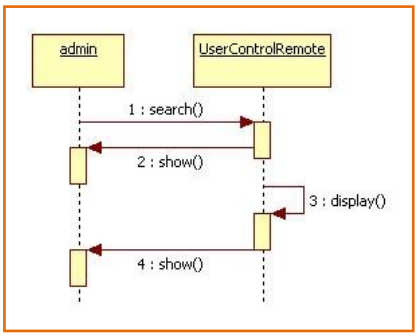
Gambar 4.39 *Sequence Diagram* Pencarian

Gambar dibawah ini adalah *sequence diagram* untuk menggambarkan proses perhitungan menggunakan metode EOQ.



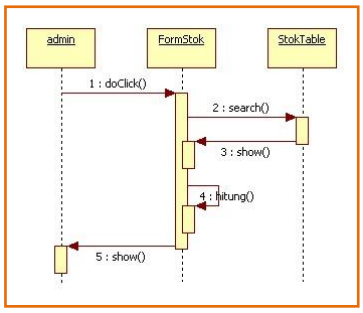
Gambar 4.40 *Sequence Diagram* EOQ

Gambar dibawah ini adalah penjelasan *sequence diagram* untuk proses menampilkan grafik.



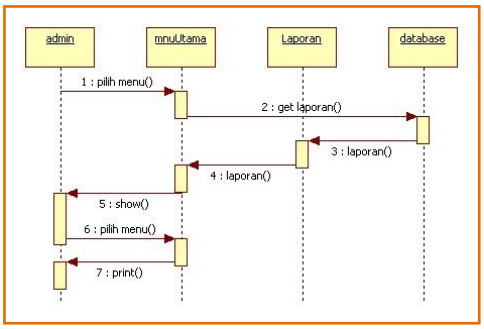
Gambar 4.41 *Sequence Diagram* Grafik

Gambar dibawah adalah *sequence diagram* untuk menampilkan stok barang.



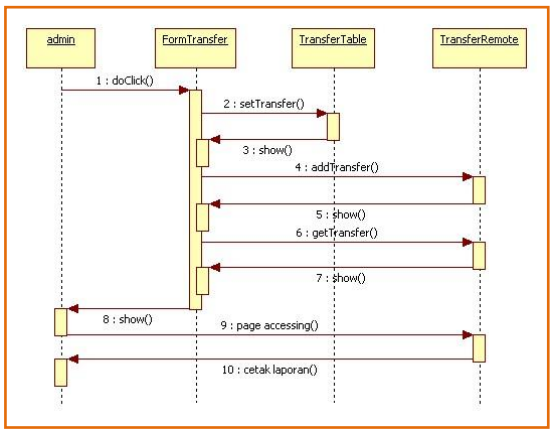
Gambar 4.42 *Sequence Diagram* Stok

Gambar dibawah ini adalah sequence diagram untuk menampilkan dan mencetak laporan.



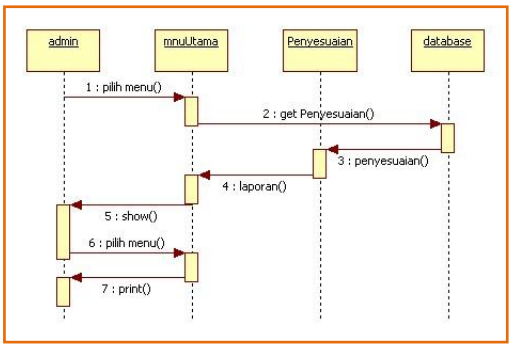
Gambar 4.43 *Sequence Diagram* Laporan

Gambar berikut ini merupakan *sequence diagram* untuk proses transfer barang antar gudang yang diusulkan.



Gambar 4.44 *Sequence Diagram* Transfer Barang

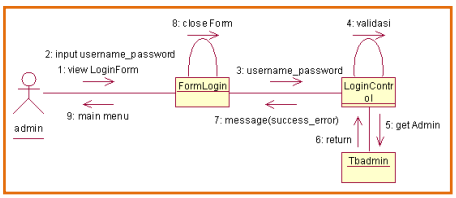
Sequence diagram untuk proses penyesuaian stok barang yaitu:.



Gambar 4.45 *Sequence Diagram* Penyesuaian Stok

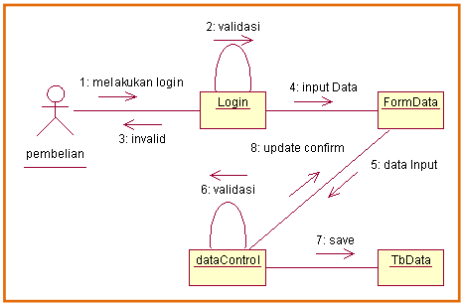
4.2.8 *Collaboration Diagram*

Seperti *sequence diagram*, *collaboration diagram* juga digunakan untuk memperlihatkan aliran-aliran pada *use case*. Sementara *sequence diagram* berurutan menurut waktu, *collaboration diagram* berfokus pada relasi-relasi yang terjadi antara objek yang satu dengan objek-objek yang lainnya.



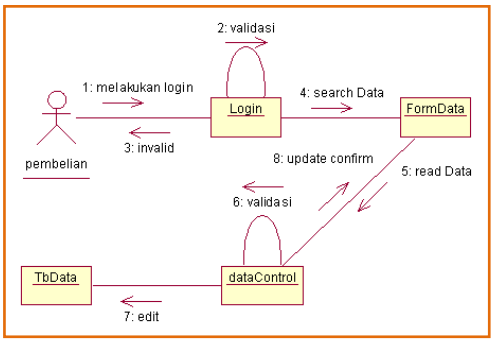
Gambar 4.46 *Collaboration Diagram Login*

Berikut ini merupakan collaboration diagram untuk proses menginputkan data kedalam *database*.



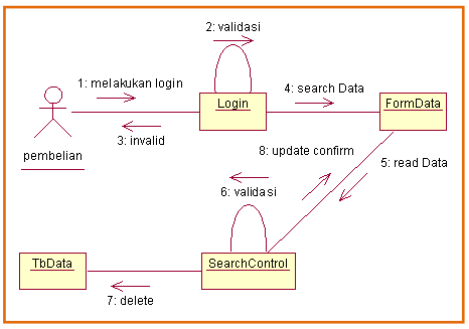
Gambar 4.47 *Collaboration Diagram Input Data*

Collaboration diagram untuk proses *update data* digambarkan seperti tampak pada gambar berikut ini.



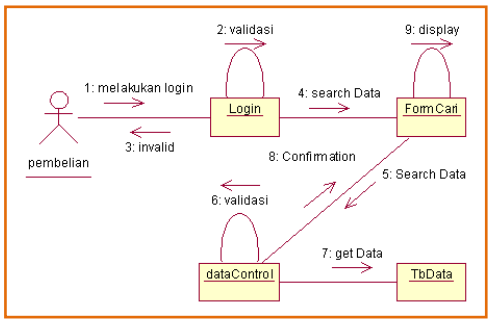
Gambar 4.48 *Collaboration Diagram Update Data*

Sementara itu untuk proses penghapusan data dari dalam sistem, dapat digambarkan sebagai brikut.



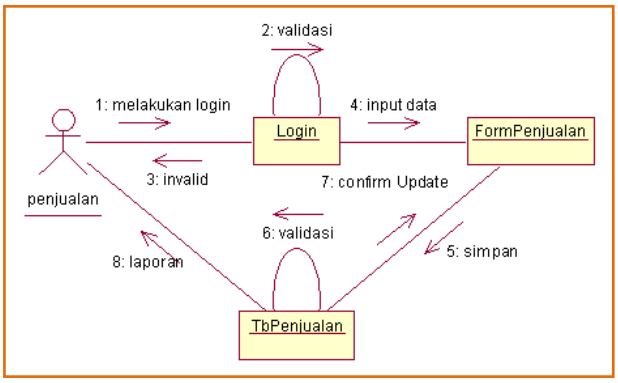
Gambar 4.49 *Collaboration Diagram Delete Data*

Gambar di bawah ini merupakan *collaboration diagram* untuk proses pencarian data.



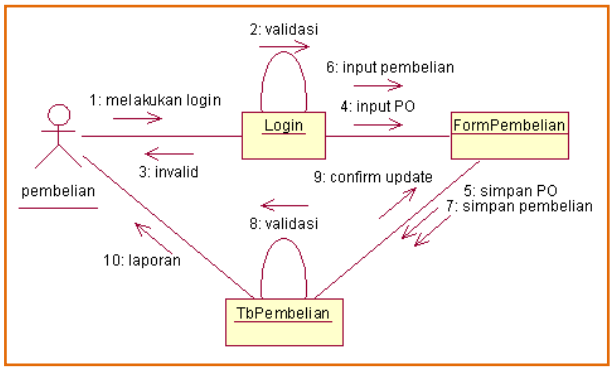
Gambar 4.50 *Collaboration Diagram Searching Data*

Gambar berikutnya adalah *collaboration diagram* yang menjelaskan proses transaksi penjualan.



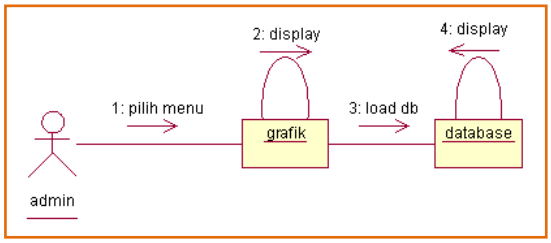
Gambar 4.51 *Collaboration Diagram* Penjualan

Gambar di bawah ini merupakan *collaboration diagram* untuk proses transaksi pembelian barang.



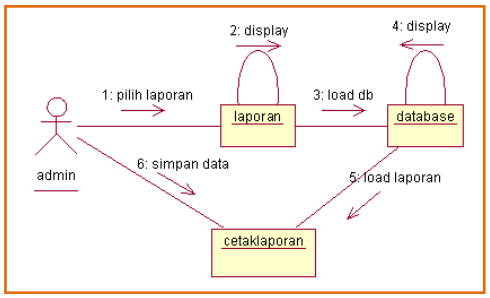
Gambar 4.52 *Collaboration Diagram* Pembelian

Untuk menampilkan grafik penjualan, maka prosesnya dapat digambarkan menjadi *collaboration diagram* seperti berikut ini:



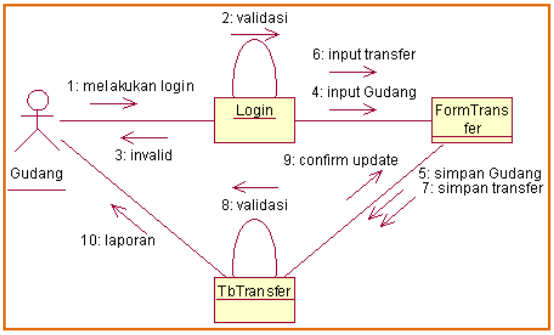
Gambar 4.53 *Collaboration Diagram* Grafik

Gambar di bawah ini merupakan *collaboration diagram* untuk proses menampilkan dan mencetak laporan-laporan.



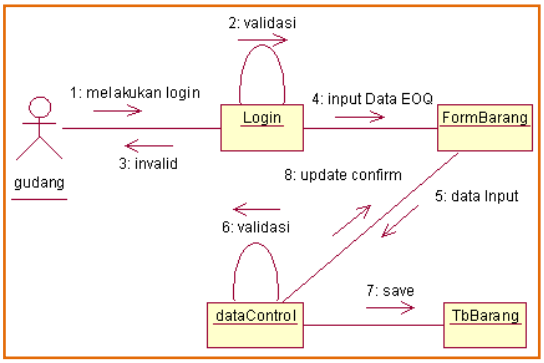
Gambar 4.54 *Collaboration Diagram* Laporan

Gambar berikutnya adalah *collaboration diagram* untuk proses transfer barang antar gudang.



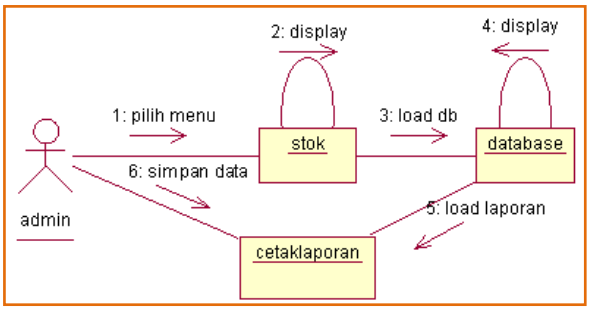
Gambar 4.55 *Collaboration Diagram* Transfer Barang

Sementara untuk perhitungan menggunakan metode EOQ, *collaboration* diagramnya digambarkan sebagai berikut.



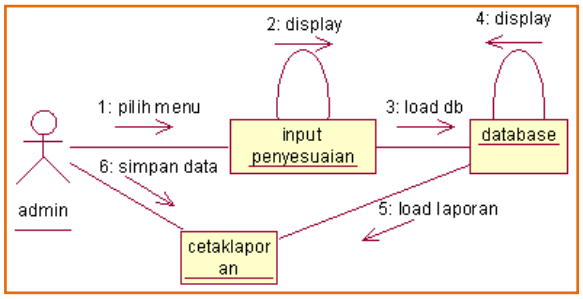
Gambar 4.56 *Collaboration Diagram* EOQ

Di bawah ini adalah *collaboration diagram* untuk proses pengecekan stok barang dalam gudang.



Gambar 4.57 *Collaboration Diagram* Stok

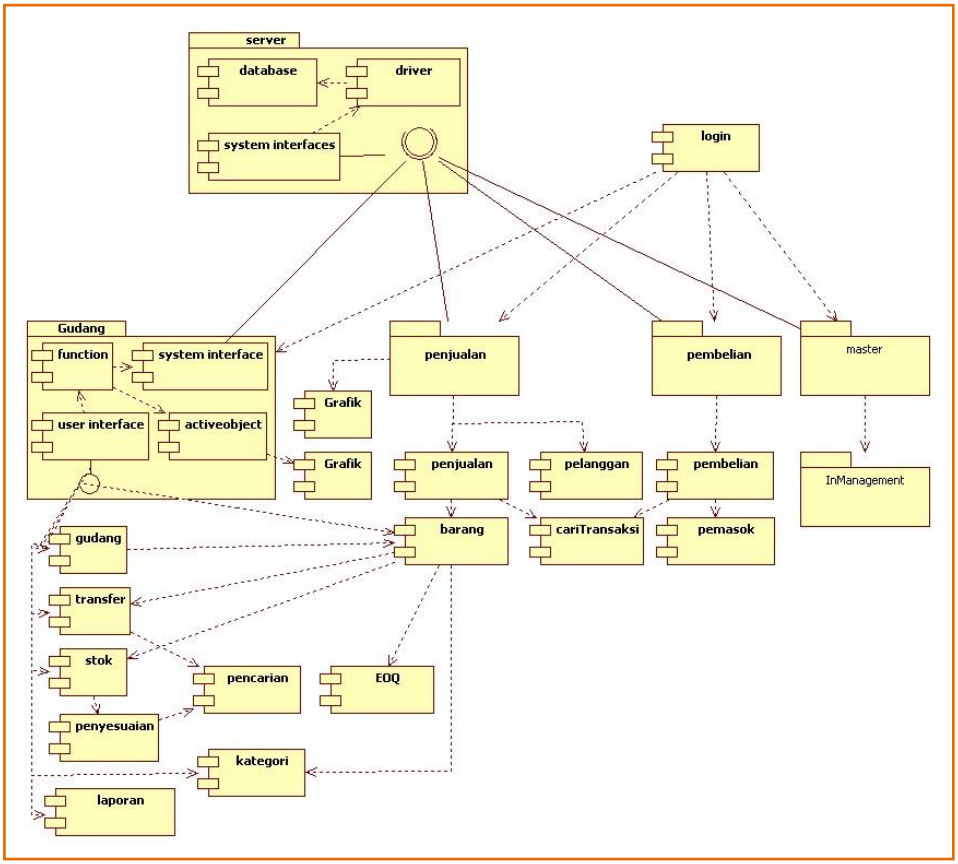
Sementara itu dalam proses melakukan penyesuaian data fisik dan aktual, maka digambarkan dengan *collaboration diagram* sebagai berikut:



Gambar 4.58 *Collaboration Diagram* Penyesuaian Stok

4.2.9 *Component Diagram*

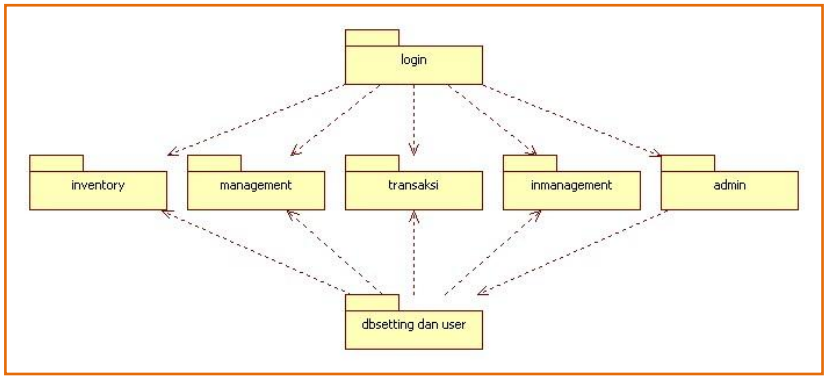
Adi Nugroho (2005:200) Diagram Komponen (*component diagram*) adalah diagram yang menggambarkan komponen-komponen dalam sistem serta *dependency* antar komponen. Dengan diagram komponen, orang-orang yang bertanggungjawab untuk mengkompilasi dan menyebarkan komponen antar mesin (*deploying*) akan diketahui pustaka kode mana yang sudah ada dan berkas tereksekusi mana yang akan diciptakan saat kode dikompilasi. Adapun Diagram Komponen yang terbentuk adalah sebagai berikut:



Gambar 4.59 *Component Diagram*

4.2.10 *Package Diagram*

Package adalah sebuah bentuk pengelompokan yang memungkinkan untuk mengambil setiap bentuk di UML dan mengelompokan elemen-elemennya dalam tingkatan unit yang lebih tinggi. Kegunaannya untuk mengelompokan *class*.



Gambar 4.60 *Package Diagram*

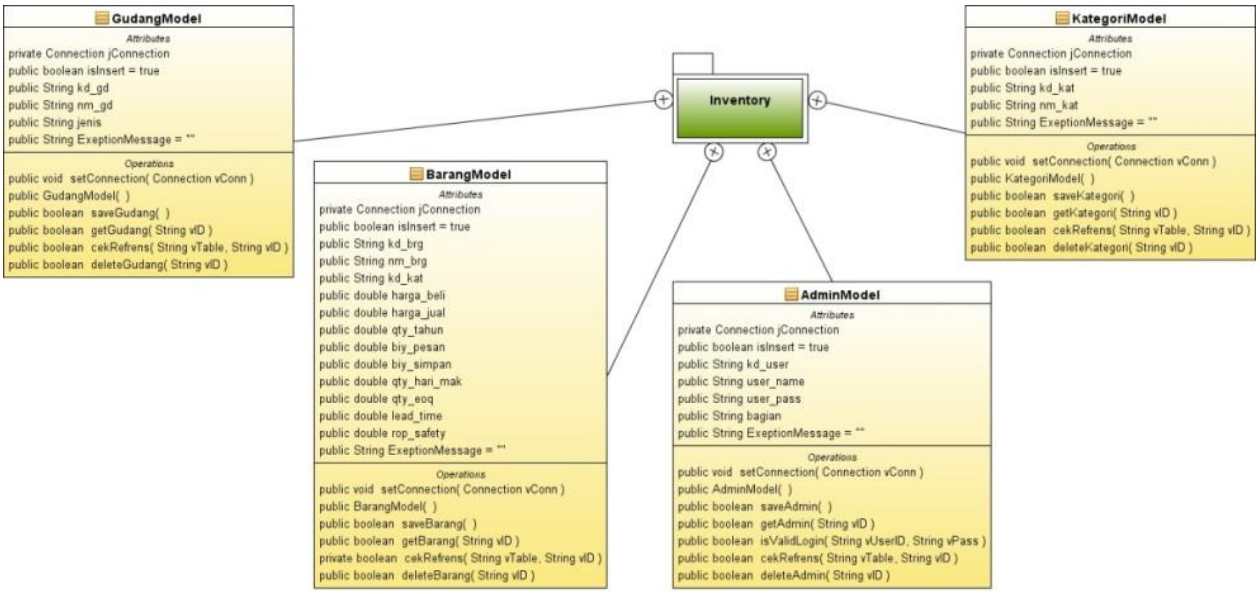
4.2.11 *Class Diagram*

Modularitas adalah sebuah teori pemecahan suatu sistem menjadi sub sistem (*break down*) menjadi yang lebih kecil, yang biasa dikenal dengan modul. Untuk memudahkan dalam pengelolaan aplikasi ini, maka kelas-kelas akan dipecah kedalam empat modul tersendiri, yaitu:

1. Modul *inventory* (manajemen barang dan gudang)
2. Modul transaksi (pembelian dan penjualan)
3. Modul *management* (manajemen: pemindahan dan penyesuaian)
4. Modul *InManagement* (laporan dan penggabungan modul)

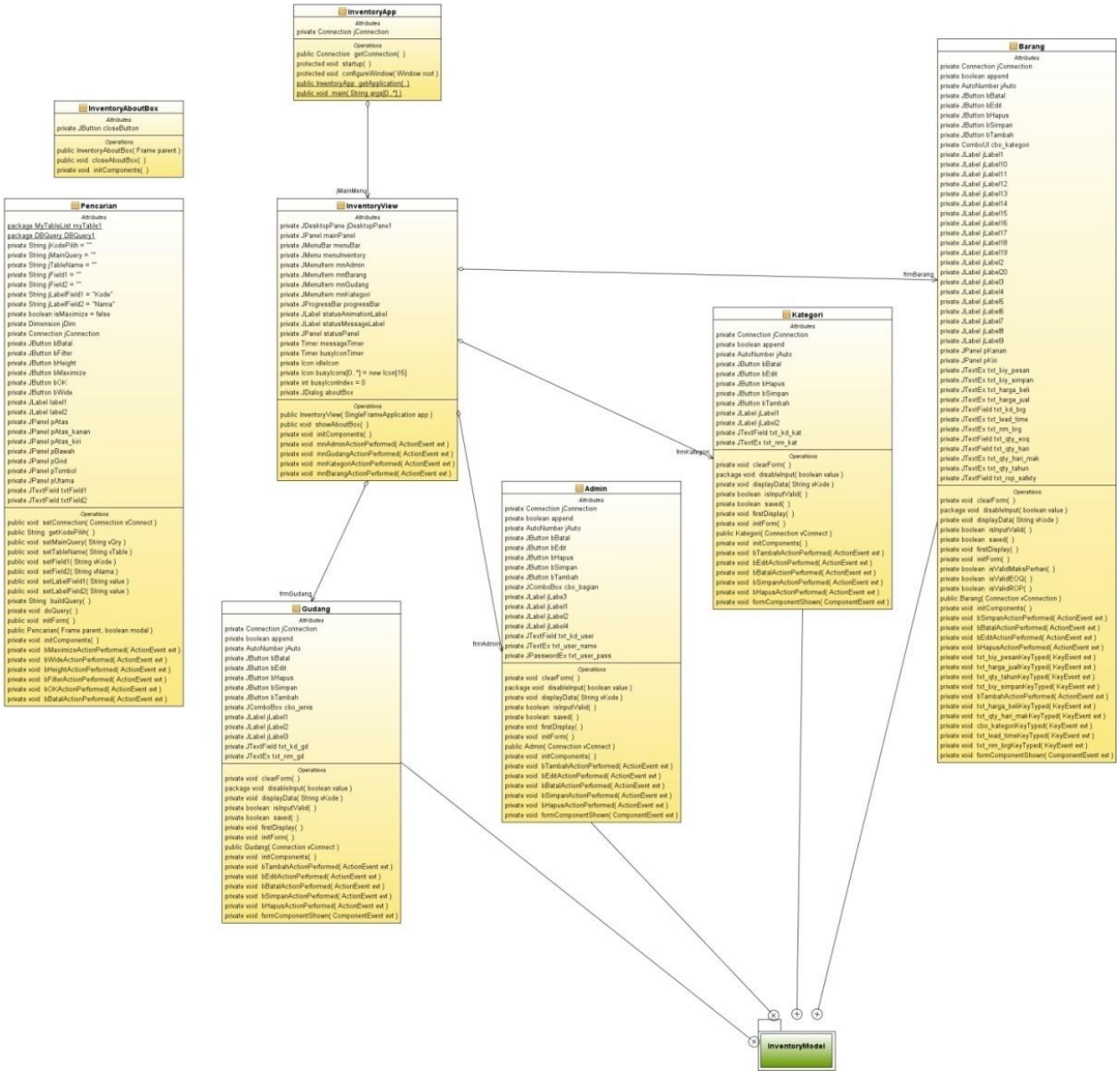
MVC (*model*, *view*, dan *control*) adalah sebuah metode pengkodean yang dikenal dengan slogan *separation of concern* (pembagian fokus masalah), prinsipnya sama dengan *breakdown system* yaitu, memecah persoalan bisnis logik, *controlling* dan tampilan. Dalam aplikasi ini, setiap modul memiliki model dan *control* yang terkumpul dalam paket model, dan memiliki unit *view* dalam paket *view*.

Pada Gambar 4.61 merupakan *Class Diagram* untuk *Model* dan *Control* modul *Inventory*. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



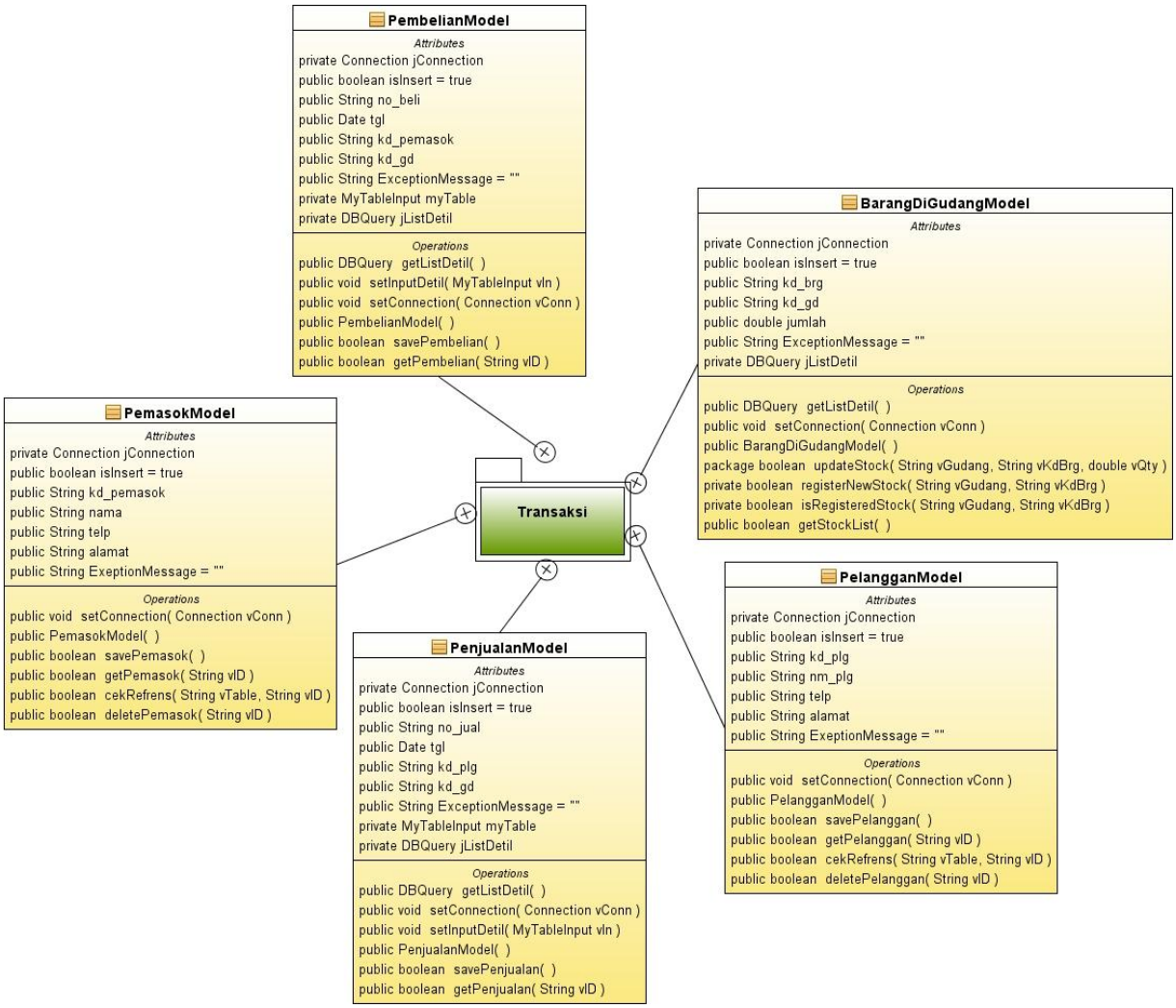
Gambar 4.61 *Class Inventory Model*

Sementara itu untuk unit *view* dari modul *Inventory* dapat dilihat pada *diagram class* di bawah ini.



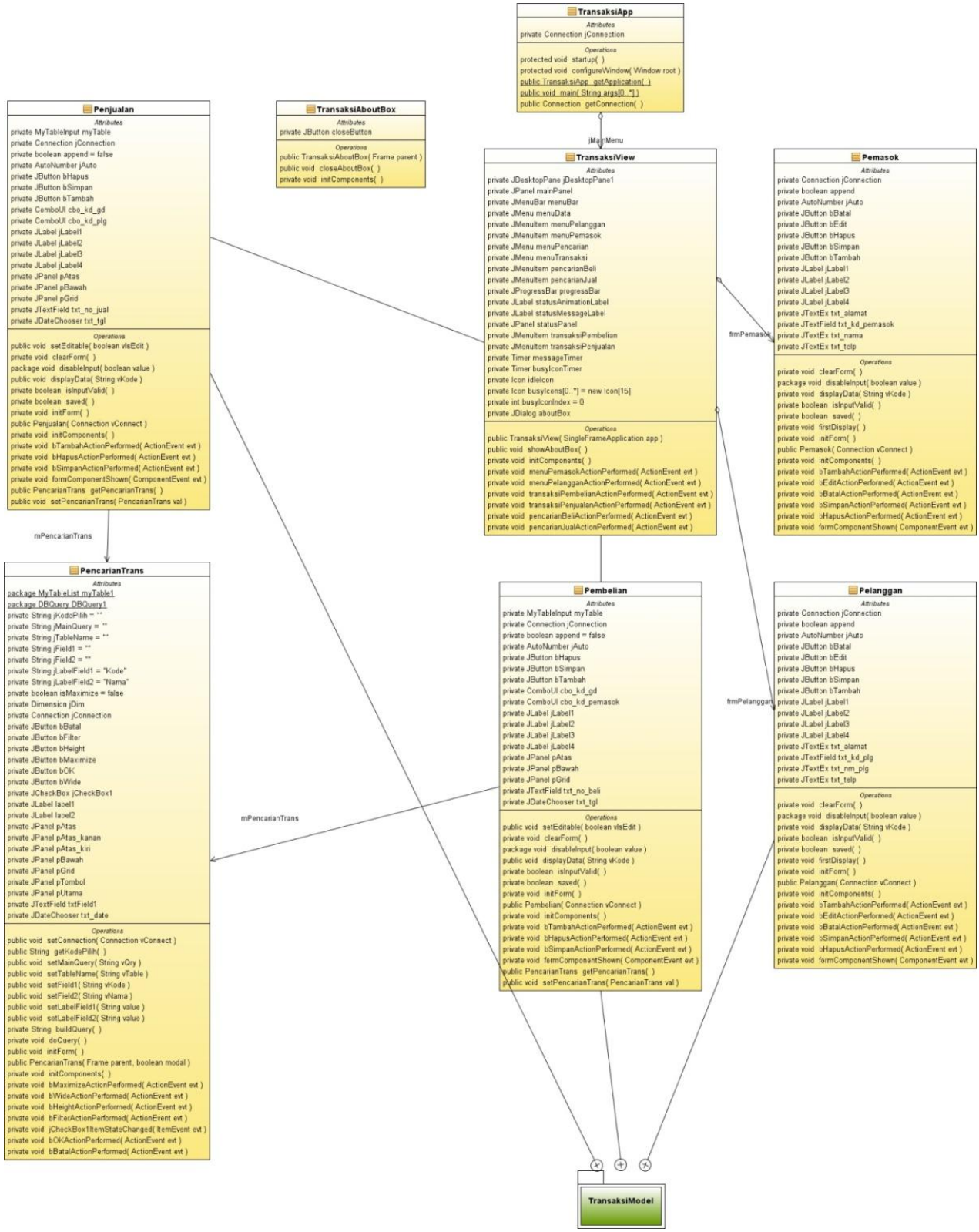
Gambar 4.62 *Class Diagram Inventory*

Pada Gambar 4.63 merupakan *Class Diagram* untuk *Model* dan *Control* modul Transaksi. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



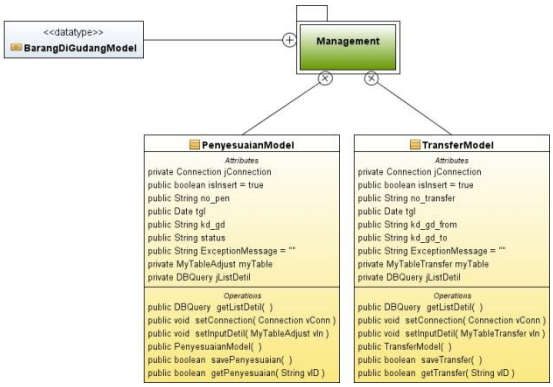
Gambar 4.63 *Class Transaksi Model*

Sementara itu untuk unit *view* dari modul Transaksi dapat dilihat pada *diagram class* di bawah ini.



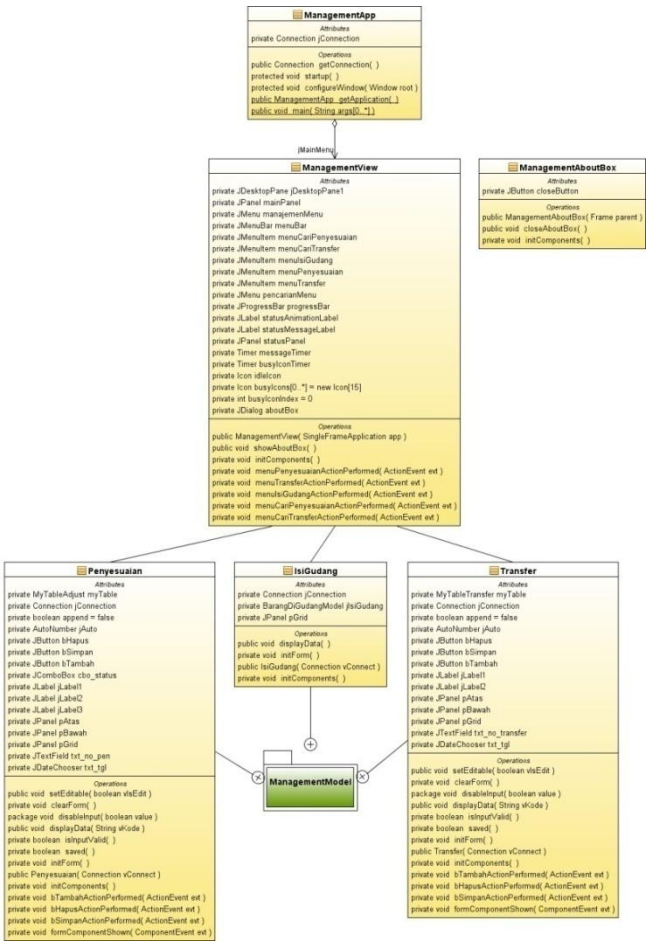
Gambar 4.64 *Class Diagram* Transaksi

Pada Gambar 4.65 merupakan *Class Diagram* untuk *Model* dan *Control* modul *Management*. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



Gambar 4.65 *Class Management Model*

Sementara untuk unit *view* modul *Management* adalah sebagai berikut.



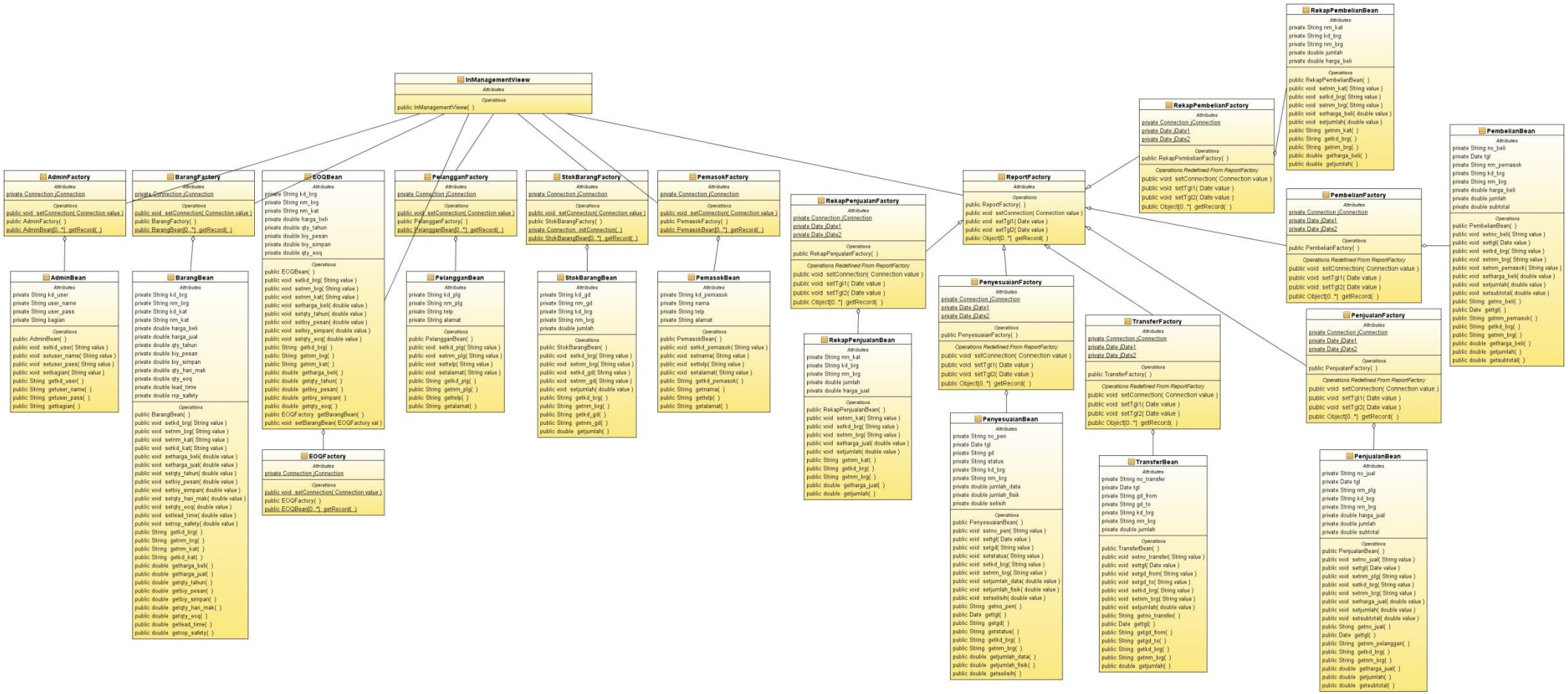
Gambar 4.66 *Class Diagram Management*

Pada Gambar 4.67 merupakan *Class Diagram* untuk *Model* dan *Control* modul *InManagement*. Untuk lebih jelasnya dapat dilihat pada gambar di bawah.



Gambar 4.67 *Class Diagram InManagement*

Sementara itu untuk unit *view* dari modul *InManagement* dapat dilihat pada *diagram class* di bawah ini:



Gambar 4.68 *Class Diagram InManagement View*

4.2.12 Kodifikasi

Pengkodean digunakan untuk tujuan mengklafikasikan data, memasukan data kedalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya. Kode dapat dibentuk dari kumpulan angka, huruf dan karakter-karakter khusus (misalnya %, /, -, $, #, &, ;). Angka merupakan simbol yang banyak digunakan pada sistem pengkodean.

Dalam Sistem Manajemen *Inventory Multi Warehouse* ini terdapat pengkodean yang bertujuan mempermudah dalam memasukan dan pencarian data. Pengkodean dalam aplikasi ini menggunakan *Auto Increment*, sehingga data kode tidak aka nada yang *redudan*.

1. Kode Barang

   XXXX

   Contoh: 0001

   Artinya No. Urut Barang

2. Kode Transfer Barang

   XXXX

   Contoh: 0001

   Artinya No. Urut Transfer

3. Kode Pembelian

   XXXX

   Contoh: 0001

   Artinya No. urut Pembelian

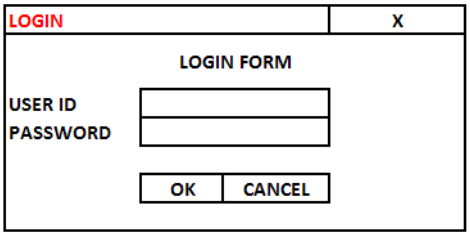
4.2.13 Perancangan Antar Muka

Perancangan antar muka merupakan tahapan untuk membuat tampilan atau *design* dari sistem yang akan dibuat. Rancangan tampilan yang dibuat meliputi rancangan *input* dan rancangan *output* dari sistem yang akan dibuat.

4.2.13.1 Perancangan *Input*

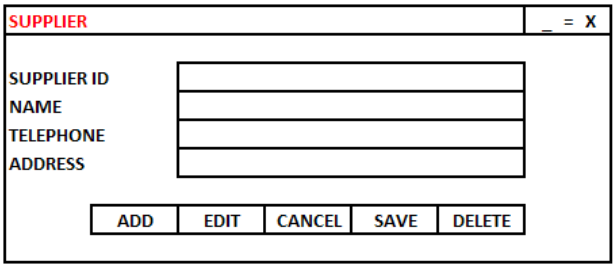
Perancangan *input* diperlukan untuk menentukan tampilan program yang berfungsi sebagai tempat memasukan data.

Form berikut ini digunakan untuk menginputkan *username* dan *password* ketika kita akan *login*.



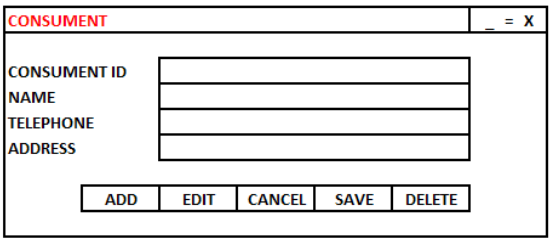
Gambar 4.69 *Form Login*

Form berikut ini digunakan untuk memasukkan data-data pemasok. Di mana didalamnya terdapat *fields* untuk mengisikan nama, alamat dan lain sebagainya.



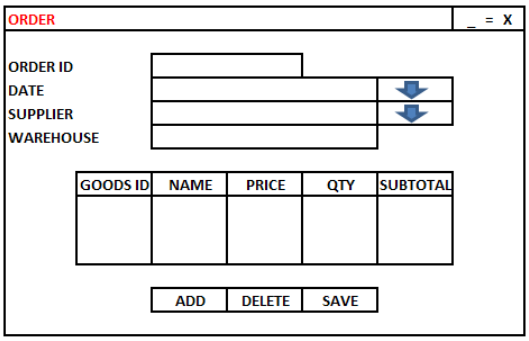
Gambar 4.70 *Form Input* Pemasok

Form di bawah ini merupakan form yang berfungsi untuk memasukkan data konsumen.



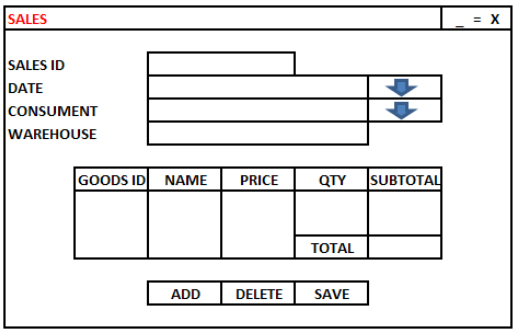
Gambar 4.71 *Form Input* Konsumen

Form berikut ini digunakan untuk memasukkan data-data transaksi pembelian. Di mana didalamnya terdapat *fields* untuk mengisikan nama barang, tanggal dan lain sebagainya.



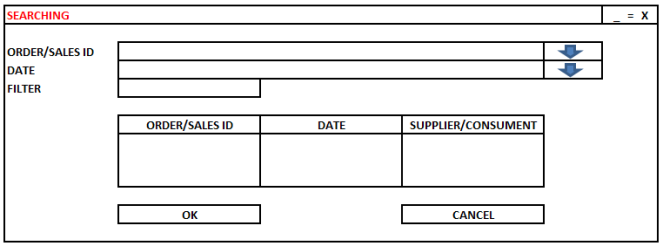
Gambar 4.72 *Form Input* Pembelian

Form di bawah ini merupakan form yang berfungsi untuk memasukkan data penjualan barang.



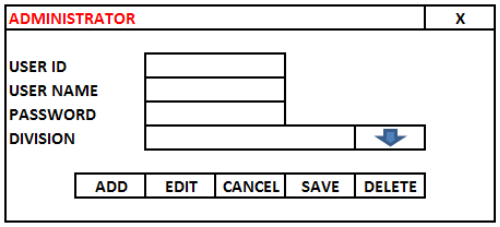
Gambar 4.73 *Form Input* Penjualan

Form di bawah ini merupakan form yang berfungsi untuk mencari data transaksi yang telah terjadi.



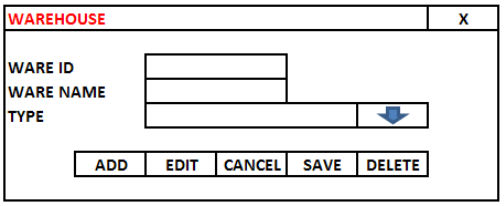
Gambar 4.74 *Form* Pencarian Berdasarkan Transaksi

Form di bawah ini merupakan form yang berfungsi untuk memasukkan data *user* dari sistem atau aplikasi.



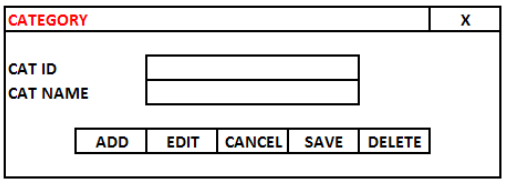
Gambar 4.75 *Form Input User*

Form di bawah ini merupakan form yang berfungsi untuk memasukkan data-data mengenai detail Gudang.



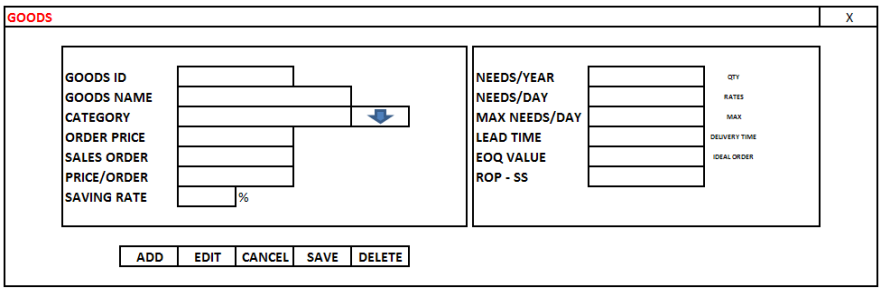
Gambar 4.76 *Form Input* Gudang

Form di bawah ini merupakan form yang berfungsi untuk memasukkan data kategori barang.



Gambar 4.77 *Form Input* Kategori

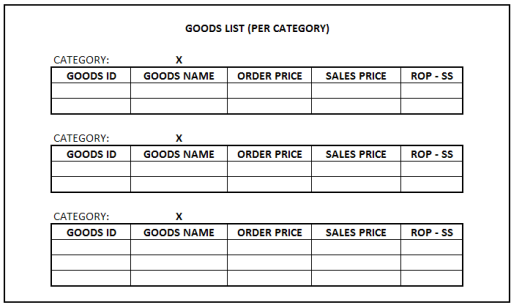
Form di bawah ini merupakan form yang berfungsi untuk memasukkan data Barang-barang di dalam gudang serta perhitungan Metode EOQ.



Gambar 4.78 *Form Input* Barang

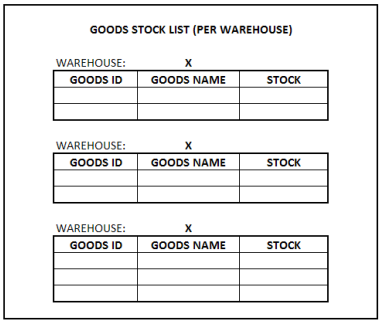
4.2.13.2 Perancangan *Output*

Perancangan *output* diperlukan untuk menentukan tampilan program yang berfungsi sebagai tempat menampilkan informasi dan data yang telah di-*input*-kan.



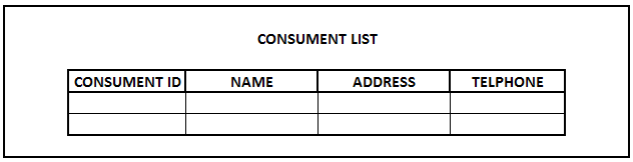
Gambar 4.79 Laporan Daftar Barang

Form di bawah ini merupakan form yang berfungsi untuk menampilkan laporan stok barang per kategori.



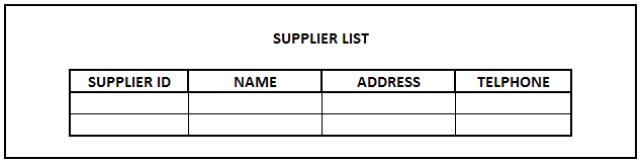
Gambar 4.80 Laporan Stok

Merupakan form yang untuk menampilkan laporan semua pelanggan.



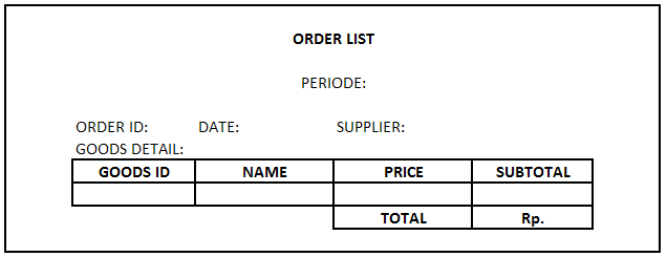
Gambar 4.81 Laporan Pelanggan

Form di bawah ini merupakan form yang berfungsi untuk menampilkan laporan pemasok.



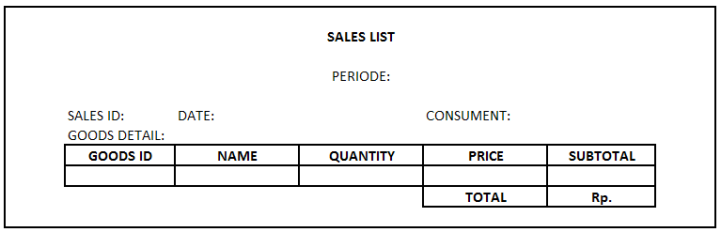
Gambar 4.82 Laporan Pemasok

Form di bawah ini merupakan form yang berfungsi untuk menampilkan laporan pembelian barang.



Gambar 4.83 Laporan Pembelian

Form di bawah ini merupakan form yang berfungsi untuk menampilkan laporan penjualan barang.

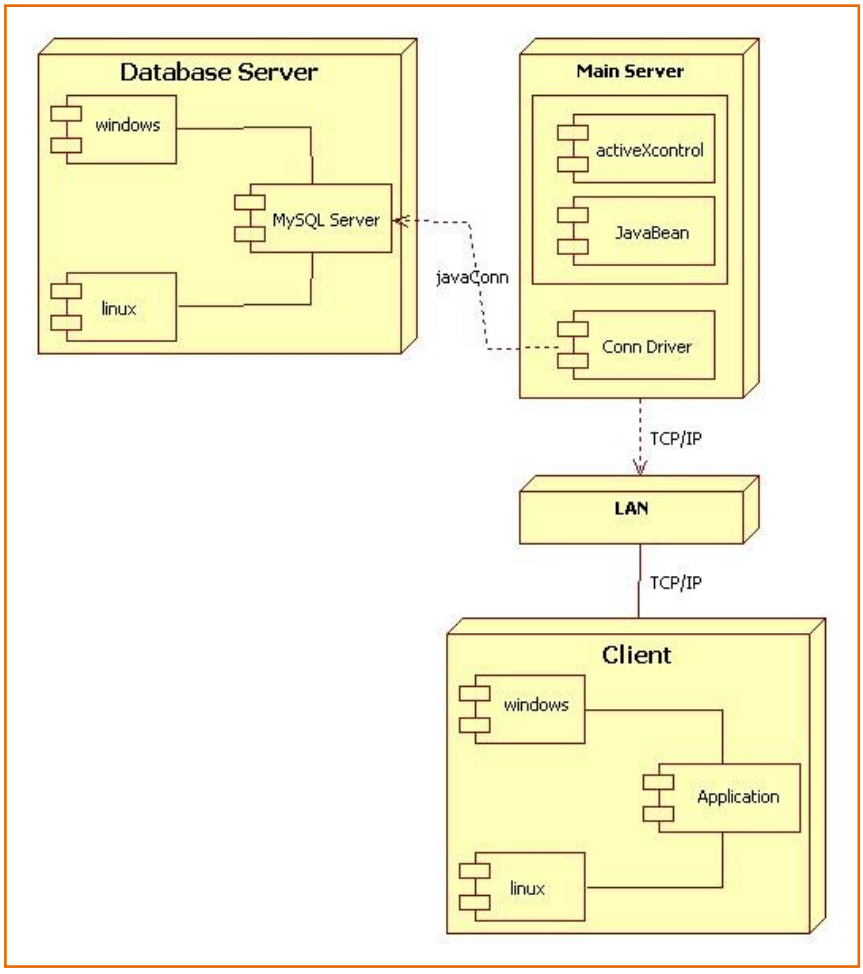


Gambar 4.84 Laporan Penjualan

4.2.14 *Deployment Diagram*

Diagram deployment menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware* yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut.

Jadi penggambaran arsitektur fisik sebuah aplikasi yang melibatkan perangkat, baik perangkat lunak maupun perangkat keras yang disebut dengan *Node* dan menunjukkan bagaimana komponen perangkat lunak dan keras ini bekerja sama akan digambarkan dalam *diagram deployment*.



Gambar 4.85 *Deployment Diagram*

Arsitektur jaringan yang dipakai dalam perancangan ini adalah arsitektur jaringan *client-server*. Pada arsitektur ini ada sebagian yang disebut *client* dan ada yang disebut *server*.

Server adalah sistem atau proses yang menyediakan data atau layanan yang diminta oleh *client*. Secara fisik sebuah *server* dapat berupa komputer (*mainframe,* mini-komputer, *workstation*, ataupun PC) atau piranti lain (misalnya *printer*). *Client* mempunyai kemampuan untuk melakukan proses sendiri.

Ketika sebuah *client* meminta suatu data ke *server*, *server* akan segera menanggapinya dengan memberikan data yang diminta ke *client* bersangkutan. Setelah diterima *client* segera melakukan pemrosesan.

Deployment diagram di atas merupakan konfigurasi *client-server* yang penulis usulkan sebagai arsitektur jaringan sistem.