DAFTAR ISI

LEMBAR JUDUL ……..…… i

LEMBAR PENGESAHAN ……...ii

KATA PENGANTAR ……..…… iii

DAFTAR ISI ……..…… vi

DAFTAR TABEL ……..…… ix

DAFTAR GAMBAR ……..……x

DAFTAR LAMPIRAN …….....xi

BAB I PENDAHULUAN 6

1.1 Latar Belakang Masalah 6

1.2 Rumusan Masalah 8

1.3 Maksud dan Tujuan 8

1.4 Batasan Masalah 8

1.5 Lokasi dan Waktu Penelitian 9

BAB II PROFIL PERUSAHAAN 10

2.1 Sejarah Singkat Perusahaan 10

2.2 Visi, Misi Perusahaan 10

2.3 Struktur Organisasi 11

2.4 Tugas Pokok Perusahaan 12

2.5 Keadaan Perusahaan 14

2.6 Keadaan Fisik 14

2.7 Bidang usaha 15

BAB III LANDASAN TEORI 16

3.1 Konsep Dasar Sistem 16

3.1.1 Definisi Sistem 16

3.1.2 Karakteristik Sistem 17
3.1.3 Klasifikasi Sistem 19
3.2 Konsep Dasar Informasi 20
3.2.1 Definisi Informasi 20
3.2.2 Kualitas Informasi 23
3.2.3 Nilai Informasi 23
3.3 Konsep Sistem Informasi 24
3.3.1 Definisi Sistem Informasi 24
3.3.2 Komponen Sistem Informasi 25
3.3.3 Kriteria Sistem Informasi 27
3.3.4 Contoh-contoh Sistem Informasi 27
3.4 Metoda Pembangunan Sistem Informasi 29
3.4.1 Metode Rekayasa Perangkat Lunak dengan Model Waterfall [PRE97] 29
3.4.2 Metode Rekayasa Perangkat Lunak dengan Model Prototipe (*Prototyping Model*) [TRI99] 31
3.4.3 Metode Rekayasa Perangkat Lunak dengan Model RAD (*Rapid Application Development*) [TRI99] 32
3.4.4 Metode Rekayasa Perangkat Lunak dengan Model Inkremental (*Incremental Model*) [TRI99] 33
3.4.5 Metode Rekayasa Perangkat Lunak dengan Model Spiral (*Spiral Model*) [TRI99] 34
3.4.6 Metode Rekayasa Perangkat Lunak dengan Model Perakitan Komponen (*Component Assembly Model*) [TRI99] 35
3.4.7 Metode Rekayasa Perangkat Lunak dengan Model Pengembangan Konkuren (*Concurent Development Model*) [TRI99] 35
3.4.8 Metode Rekayasa Perangkat Lunak dengan Model Metode Formal (*Formal Methods Model*) [TRI99] 36
3.5 Basis Data Relasional 36
3.5.1 Abstraksi Data 37
3.5.2 Model Unified Modelling Language 37
BAB IV ANALISIS KERJA PRAKTEK 60
4.1 Identifikasi Masalah 60

4.2 Prinsip Kerja Sistem yang Lama 61
4.2.1 Proses Pembelian dan Penjualan Barang 61
4.2.2 Pembuatan Laporan..... 62
4.2.3 Identifikasi Penyebab Masalah 62
4.2.4 Hasil Analisis Sistem Dahulu 63
4.2.5 Hasil Analisis Sistem Yang Akan Dikembangkan 65
BAB V KESIMPULAN DAN SARAN..... 105
5.1 Kesimpulan..... 105
5.2 Saran..... 105

KATA PENGANTAR

Assalamualaikum Wr. Wb

Puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah memberikn rahmat dan hidayahnya, sehingga penulis dapat menyelesaikan penyusunan laporan Kerja Praktek yang merupakan salah satu syarat dalam pendidikan jenjang S1 pada Fakultas Teknik dan Ilmu Komputer Jurusan Manajemen Informatika di Universitas Komputer Indonesia. Adapun penulis mengambil judul **"ANALISIS SUPPLY CHAIN".**

Penulis menyadari bahwa dalam penulisan laporan kerja praktek ini masih banyak sekali kekurangan dan jauh untuk dikatakan sempurna, baik dari segi isi maupun bahasa serta teknik penulisannya, karena keterbatasan kemampuan dan pengetahuan penulis.

Walaupun demikian penulis sudah berusaha semaksimal mungkin sehingga penulisan laporan kerja praktek ini dapat diselesaikan dengan harapan semoga memenuhi persyaratan yang telah ditentukan. Oleh karena itu penulis sangat mengharapkan sekali saran dan kritik yang membangun demi kesempurnaan isi yang terkandung dalam laporan kerja praktek ini.

Dalam proses penyusunan laporan kerja praktek ini penulis banyak sekali mendapat bimbingan dan bantuan dari berbagai pihak, oleh karenanya dalam kesempatan ini dengan ketulusan hati penulis ingin menyampaikan banyak terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam menyelesaikan pembuatan laporan kerja praktek ini, diantaranya:

1. Bapak, Ibu dan Keluarga Besar yang selalu memberikan doa dan dukungan baik materiil dan spirituil.
2. Bapak Prof. DR. Ir. Ukun Sastra Prawira, Msc., selaku Dekan Fakultas Teknik Universitas Komputer Indonesia.
3. Bapak Dadang Munandar,S.E.,M.Si selaku Ketua Jurusan Manajemen Informatika Universitas Komputer Indonesia.
4. Bapak Iyan Gustiana, selaku Dosen wali kelas MI-8 Universitas Komputer Indonesia.
5. Prima Intrusiano selaku pembimbing kerja praktek di PRIMATAMA COMPUTER.
6. Seluruh staf dan pegawai di PRIMATAMA COMPUTER.
7. Seluruh dosen Universitas Komputer Indonesia yang telah mengajarkan pelajaran-pelajaran yang berharga bagi kami.

Kawan – kawan MI-8 2006 yang telah banyak membantu dalam menyelesaikan laporan ini, dan semua pihak yang telah membantu penulis, baik material Penulis menyadari bahwa Laporak Kerja Praktek ini jauh dari sempurna, untuk itu kritik dan saran senantiasa penulis harapkan demi perbaikan pada masa selanjutnya.

Wassalamualaikum Wr. Wb

Bandung, 8 Oktober 2009

Penulis

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan kemajuan dunia bisnis yang semakin pesat menuntut agar para pelaku bisnis untuk dapat bersaing secara ketat dan melakukan pelayanan yang maksimal. Sehingga terjadi kepuasan para pelaku pasar yang akhirnya menimbulkan kepercayaan konsumen untuk menggunakan suatu produk. Namun sedikit saja kesalahan dalam hal apapun yang berkaitan dengan jalannya usaha dapat menimbulkan toko itu gulung tikar. Dengan berkembangnya perekonomian serta usaha yang semakin luas, maka dituntut adanya pelayanan yang baik, cepat dan tepat sasaran.

Supply chain adalah sebuah sistem yang melibatkan proses produksi, pengiriman, penyimpanan, distribusi dan penjualan produk dalam rangka memenuhi permintaan akan produk tersebut. Supply chain didalamnya termasuk seluruh proses dan kegiatan yang terlibat didalam penyampaian produk tersebut sampai ketangan pemakai (konsumen). Semua itu termasuk proses produksi pada manufaktur, sistem transportasi yang menggerakkan produk dari manufaktur sampai ke outlet retailer, gudang tempat penyimpanan produk tersebut, pusat distribusi tempat dimana pengiriman dalam party besar dibagi kedalam party kecil untuk dikirim kembali ke toko-toko dan akhirnya sampai ke retailer yang menjual produk-produk tersebut.

Primatama Computer, suatu toko yang bergerak dalam bidang penjualan dan pembelian berbagai part dan software komputer. Perusahaan ini tumbuh dan bersaing dengan perusahaan dagang lainnya yang bergerak dalam bidang yang sama. Guna untuk bertahan diperlukan inovasi baru yang mendukung tumbuh kembangnya toko. Salah satunya adalah untuk memastikan sebuah produk berada pada tempat dan waktu yang tepat untuk memenuhi permintaan konsumen tanpa menciptakan stok yang berlebihan atau kekurangan.

Supply chain yang efektif sangat penting dalam pengembangan suatu toko. Karena pada setiap tahap kegiatan perlu dilakukan pengendalian, agar bisa lebih cepat dilakukan koreksi bila terjadi penyimpangan.

Hal ini dilakukan agar tidak ada kesalahan yang merugikan konsumen, maupun toko Primatama Computer. Cara yang ditempuh toko untuk mencapai inovasi tersebut adalah dengan merubah system pengendalian, pembelian, penyimpanan, dan penjualan (cara manual atau konvensional), akan diubah dengan cara baru yaitu komputerisasi, untuk mendukung peningkatan pengendalian, stok barang, pembelian, penyimpanan, dan penjualan di toko Primatama Computer.

Berdasarkan uraian di atas penulis akan mengangkat suatu tema yaitu **”ANALISIS APLIKASI SUPPLY CHAIN”** untuk meningkatkan pelayanan kepada konsumen, dan memudahkan dalam mengendalikan stok barang, pembelian, penyimpanan, dan penjualan di toko Primatama Computer.

1.2 Rumusan Masalah

Beberapa masalah yang sering timbul dari system lama adalah:

1. Informasi barang yang ada tidak diketahui jumlah stok barang yang ada.
2. Masih kurangnya sistem informasi dalam menunjang pembelian, penyimpanan, dan penjualan.
3. Pemeliharaan data barang tidak teratur karena pemeliharaan data barang tidak terorganisir.

1.3 Maksud dan Tujuan

Maksud dari penyusunan tugas akhir ini ini adalah membantu pelaksanaan inovasi Toko, yaitu dengan cara membangun aplikasi stok barang, pembelian, penyimpanan, dan penjualan. Dengan tujuan untuk meningkatkan pelayanan kepada konsumen dan kemudahan dalam mengendalikan stok barang, pembelian, penyimpanan, dan penjualan.

1.4 Batasan Masalah

Karena luasnya bidang yang dihadapi penulis, maka penulis membatasi penyusunan laporan ini berdasarkan ruang lingkup kegiatan, yaitu analisis software aplikasi mengendalikan pembelian, penyimpanan, dan penjualan dengan batasan batasan :

1. Pembuatan sistem informasi hanya meliputi sistem informasi stok barang , pembelian, penyimpanan, dan penjualan.

2. Aplikasi menggunakan Visual Basic 6.0 dan Microsoft Access.

1.5 Lokasi dan Waktu Penelitian

Penelitian dilaksanakan di Toko Primatama Computer yang beralamat di Kandaga Computer Center. Penelitian dilaksanakan pada bulan Juli 2009.

BAB II

PROFIL PERUSAHAAN

2.1 Sejarah Singkat Perusahaan

Pada awalnya Primatama Computer berasal dari usulan dua orang, yaitu Arsymal dan Prima Intrusiano pada sekitar bulan Desember 2007, untuk mendirikan sebuah badan usaha di bidang perdagangan computer, khususnya hardware dan aksesoris Overclocking. Arsyimal dan Prima Intrusiano berusaha untuk mencari sebuah inovasi toko computer seperti apa yang akan dibangun, seperti kebanyakan toko computer lain, biasanya konsumen sudah jenuh dengan barang dan aksesoris itu itu saja. Maka Primatama didesain bagi para Gamers dan juga Overclocker, ataupun customer yang ingin tampilan komputernya tidak seperti biasanya, namun bisa dibilang mewah ataupun Ekstrem. Tapi tidak menutup bagi konsumen perorangan, maupun perusahaan ataupun institusi yang ada untuk bisa mendapatkan harga yang bersaing baik harga eceran maupun grosir. Primatama Computer resmi didirikan pada tanggal 11 Februari 2008 yang berlokasi di Jalan Ahmad yani no 416, Kandaga Computer Centre Blok G no 8B Bandung, mulai beroperasi dari pukul 10.00 – 19.00.

2.2 Visi, Misi Perusahaan

Visi Primatama Computer

“ Menjadi perusahaan yang terpercaya dan dipilih untuk memberikan solusi kebutuhan computer untuk semua kalangan."

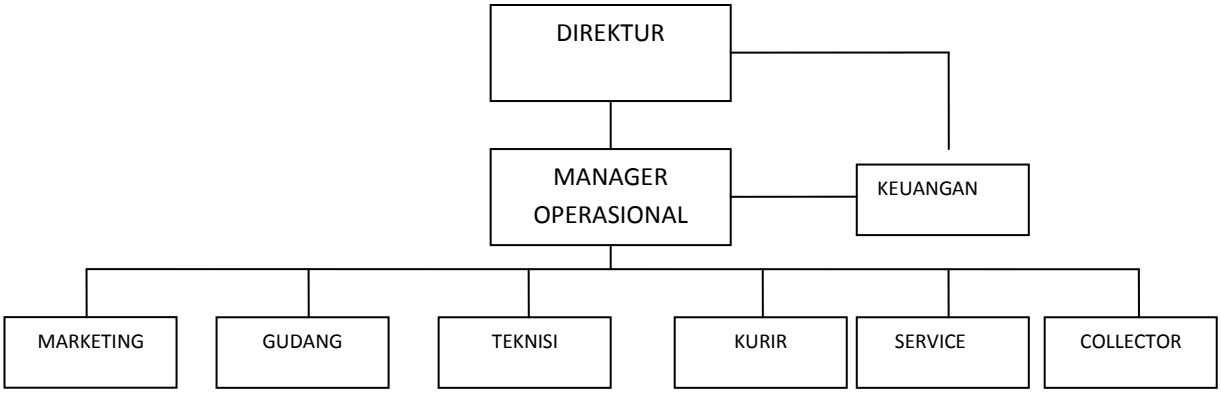
Misi Primatama Computer

1. Mengelola perusahaan secara profesional, untuk dapat memberikan layanan yang terbaik bagi konsumen.
2. Memberikan produk yang bermutu tinggi, dengan tetap memberikan harga yang bersaing kepada konsumen.
3. Memberikan jasa konsultasi yang berkualitas, tepat biaya, tepat mutu dan tepat guna.

2.3 Struktur Organisasi

Struktur organisasi merupakan salah satu komponen dasar, pembentukannya sesuai dengan kebutuhan perusahaan dan diatur sedemikian rupa sehingga mampu mendukung upaya untuk mencapai tujuan perusahaan yang diinginkan. Melalui sistem organisasi tersebut akan diatur mengenai tugas dan kewenangan masing-masing personel perusahaan serta hubungan serta pola kerja antar personil. Kejelasan mengenai hal tersebut akan sangat besar artinya guna menghindari adanya kesalahan penanganan maupun kesalahan dalam menangani suatu pekerjaan. Tanpa adanya struktur organisasi yang baik, tidak mungkin suatu perusahaan akan berjalan dengan baik sesuai yang diharapkan.

Ini berarti bahwa sedikit banyak, sistem organisasi akan berpengaruh terhadap faktor dan kemampuan manajerial. Struktur organisasi Primatama Computer dapat digambarkan sebagai berikut:



Gambar 2.1 struktur organisasi

2.4 Tugas Pokok Perusahaan

1. Direktur.

Direktur merupakan pengurus perusahaan yang bertanggung jawab penuh dalam segala hal yang mengenai perusahaan, yang mewakili perusahaan terhadap pihak luar, yang berhak menandatangani atas nama perusahaan, menghubungkan perusahaan dengan pihak luar dan sebaliknya serta melakukan segala tindakan baik mengenai pengurusan maupun mengenai pemilikan.

2. Manajer Operasional

Manager Operasional bertugas mengkordinasikan tugas harian kepada semua personel dan juga melakukan evaluasi keluar dan masuknya barang, menetapkan harga, dan juga pengambil keputusan bilamana diperlukan.

Bekerjasama dengan pihak luar dalam penyelenggaraan event atau pameran, koordinasi dengan semua manajer, baik itu Marketing, kurir, teknisi, gudang dan juga collector., service, gudang , dan semua komponen dalam Primatama Computer.

3. Keuangan

Bertugas untuk mengatur keuangan dalam Primatama Computer. Baik itu pendapatan dari hasil penjualan, dan juga pengeluaran Primatama computer, baik itu tentang sewa gedung, listrik, kebersihan dan juga perihal tentang beban dan biaya operasional Primatama Computer. Bagian keuangan juga bertugas untuk mengelola utang dan piutang kepada Dealer dan Konsumen, baik itu perorangan ataupun institusi.

4. Marketing

Bertugas untuk memasarkan barang dan juga melakukan pemesanan order barang, bernegosiasi dengan dealer maupun customer, mengetahui tentang spesifikasi produk dan mampu memberikan fitur dan info produk pada customer. Bersentuhan langsung dengan konsumen, bertugas melayani konsumen dan melakukan penjualan

5. Gudang

Bertugas melakukan pengecekan barang masuk dan keluar, melakukan pemeliharaan selama barang ada di gudang

6. Teknisi

Bertugas melakukan perakitan computer dari awal sampai akhir, melakukan pengecekan dan review barang.

7. Kurir

Bertugas melakukan pengiriman barang kepada konsumen, pengambilan barang, melakukan klaim garansi kepada Dealer utama, pembelian dollar ke money changer, setoran uang ke bank.

8. Service

melakukan service terhadap hardware yang rusak secara langsung, baik penggantian komponen sebagian ataupun keseluruhan. Bertugas pula mengelola barang service dari konsumen untuk seterusnya di claim kepada dealer utama bilamana garansi dari hardware tersebut masih berlaku.

9. Collector

Bertugas melakukan penagihan dan menyelesaikan berkas utang dan piutang dengan pihak luar.

2.5 Keadaan Perusahaan

2.6 Keadaan Fisik

Primatama Computer adalah suatu badan usaha di bidang computer, hardware, networking dan juga aksesoris dan peripheral computer yang berorientasi pada kualitas dan juga harga yang kompetitif, baik untuk kebutuhan computer perorangan, keperluan usaha, maupun kebutuhan kebutuhan sebuah

institusi, dengan mengedepankan kepuasan serta service yang *excellent* kepada setiap pelanggan.

2.7 Bidang usaha

Primatama Computer adalah sebuah badan usaha yang bergerak di bidang hardware computer, notebook, Software original, service, networking peripheral, aksesoris , peripheral overclocking maupun gamer peripheral dan juga PC modding.

Primatama computer berusaha dan senantiasa untuk bisa memberikan sebuah produk hardware yang berkualitas dan juga service yang memuaskan para pelanggan. Permintaan computer saat ini semakin meningkat, seiring dengan sebuah computer bukan hanya sebagai alat bantu saja, namun bisa dijadikan sebagai sebuah kebutuhan yang tidak dapat dipisahkan. Saat ini computer sudah masuk ke berbagai bidang dan lapisan masyarakat, baik itu pendidikan, bisnis dan usaha kecil menengah, maupun sebagai alat bantu proses belajar para konsumennya, namun seiring perkembangan zaman, computer juga diminati bukan hanya sebagai alat bantu untuk menyelsaikan pekerjaan, tapi juga sebagai alat untuk menyalurkan hobi dari konsumen itu sendiri, misalnya gaming dan overclocking. Yang tentunya hobi tersebut harus didukung oleh hardware yang baik dan tentu saja berkualitas. Oleh karena itu Primatama computer berusaha hadir untuk mengakomodir kebutuhan setiap pelanggan, baik itu siswa, mahasiswa, pegawai, designer, wiraswasta, perorangan, maupun institusi, baik dalam quantity eceran maupun dengan grosir dengan berusaha untuk memberikan produk dan service yang berkualitas.

BAB III

LANDASAN TEORI

3.1 Konsep Dasar Sistem

3.1.1 Definisi Sistem

Beberapa definisi sistem diantaranya :

a. Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan dikembangkan sesuai dengan suatu skema yang terintegrasi untuk melaksanakan suatu kegiatan utama. [NEU71]

b. Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan berkumpul bersama-sama melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu. [FIT81]

c. Sistem adalah suatu kumpulan komponen-komponen yang berinteraksi membentuk suatu kesatuan dan keutuhan yang komplek di dalam tingkat tertentu untuk mengejar tujuan yang umum. [NAS84]

d. Sistem adalah sekelompok bagian yang disusun dan diatur dengan baik yang bekerjasama untuk melakukan suatu maksud. [BIM]

e. Sistem adalah suatu grup dari elemen-elemen baik yang berbentuk fisik maupun bukan fisik yang menunjukan suatu kumpulan yang saling berhubungan dan berinteraksi bersama-sama menuju satu atau lebih tujuan, sasaran, atau akhir dari system. [ALE74]

Dari beberapa definisi sistem diatas dapat disimpulkan bahwa sistem dikelompokan menjadi dua bagian yang menekankan pada prosedurnya dan ada yang menekankan pada elemennya. Kedua kelompok ini adalah benar dan tidak bertentangan, yang berbeda adalah cara pendekatannya.

3.1.2 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu [JOG99] :

a. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem yang mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

b. Batas Sistem (*Boundary)*

Batas sistem merupakan suatu daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya, yang menunjukkan ruang lingkup (*scope*) sistem tersebut.

c. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar dari suatu sistem adalah segala sesuatu di luar batas sistem yang mempengaruhi operasi sistem, dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar sistem yang

menguntungkan yaitu energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar sistem yang merugikan harus ditahan dan dikendalikan, kalau tidak akan menggangu kelangsungan hidup sistem.

d. Penghubung Sistem (*Interface*)

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Dengan penghubung satu subsistem dapat berinteraksi dengan subsistem lainnya membentuk satu kesatuan. Keluaran dari satu subsistem memungkinkan menjadi masukan untuk subsitem lainnya melalui media penghubung.

e. Masukan Sistem (*Input*)

Masukan adalah energi yang dimasukan ke dalam sistem dan menentukan keluaran sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk mendapatkan keluaran.

f. Keluaran Sistem (*Output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem lain.

g. Pengolah Sistem (*Process*)

Pengolah merupakan bagian yang merubah masukan menjadi keluaran.

h. Sasaran Sistem (*Objectives*) atau Tujuan (*Goal*)

Jika suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan masukan yang akan dibutuhkan oleh sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

3.1.3 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut [JOG99] :

a. Sistem Abstrak (*Abstract System*) dan Sistem Fisik (*Physical System*)

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem fisik merupakan sistem yang ada secara fisik.

b. **Sistem Alamiah (*Natural System*) dan Sistem Buatan Manusia** (*Human Made System*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sistem buatan manusia adalah sistem yang dirancang dan dibuat oleh manusia.

c. **Sistem Tertentu (*Deterministic System*) dan Sistem Tak Tentu** (*Probabilistic System*). Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

d. Sistem Tertutup (*Closed System*) dan Sistem Terbuka (*Open System*)

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur dari pihak luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya.

3.2 Konsep Dasar Informasi

3.2.1 Definisi Informasi

Beberapa definisi informasi diantaranya :

"Informasi adalah pernyataan-pernyataan atau bentuk-bentuk berguna yang dapat digunakan untuk pengambilan keputusan bisnis" [JOG88].

"Informasi adalah kumpulan data yang relevan dan mempunyai arti yang menggambarkan suatu kejadian-kejadian atau kegiatan-kegiatan" [VER82].

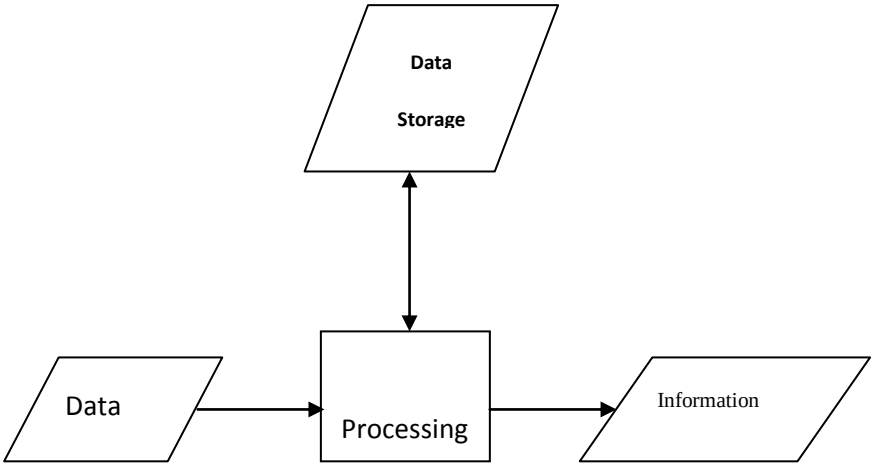
"Informasi adalah data yang diolah ke dalam bentuk yang berguna bagi penerimanya dan nyata atau berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun yang akan datang" [DAV74]

"Informasi adalah data yang telah diletakkan dalam konteks yang lebih berarti dan berguna yang dikomunikasikan kepada penerima untuk digunakan di dalam pembuatan keputusan" [BUR86]

“Informasi merupakan hasil pemrosesan yang berasal dari data dasar menjadi sesuatu yang lebih bernilai informatif atau kesimpulan, argumentasi, atau sebagai dasar untuk peramalan atau pengambilan keputusan” [ANA99].

Sumber dari informasi adalah data. Data merupakan bentuk yang masih mentah, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol seperti huruf-huruf alphabet atau angka-angka, bentuk-bentuk suara, gambar-gambar dan sebagainya[JOG99]. Hubungan data dan informasi didefinisikan sebagai bahan baku dan produk jadi. Data sebagai bahan baku, diolah melalui suatu proses transformasi atau pengolahan data menjadi informasi. Dengan kata lain bahwa informasi merupakan keluaran-keluaran (*outputs*) dari proses transformasi, dimana data berfungsi sebagai masukan-masukannya (*inputs*). Jadi jika ditinjau sebagai suatu sistem, maka sistem informasi akan menerima masukan-masukan yang berupa data dan instruksi, mengolah data sesuai dengan instruksi-instruksi, dan mengeluarkan hasilnya berupa informasi.

Fungsi pengolah data menjadi informasi seringkali memerlukan data yang telah dikumpulkan dan diolah sebelumnya. Oleh karena itu pada model sistem informasi perlu ditambahkan alat penyimpanan yang biasa disebut memori sehingga kegiatan pemrosesan/pengolahan mempunyai data, baik yang baru maupun yang telah disimpan sebelumnya. Model sistem informasi tersebut ditunjukkan dalam gambar 2.1.



Gambar 3.1 Transformasi data menjadi informasi

Sumber : HM Jogiyanto, *Sistem Informasi Berbasis Komputer*.

Dari uraian diatas dapat disimpulkan bahwa informasi adalah data yang sudah diolah, menjadi bentuk yang lebih berguna dan lebih berarti bagi penerima, menggambarkan suatu kegiatan-kegiatan dan kesatuan nyata, dan digunakan untuk pengambilan keputusan.

3.2.2 Kualitas Informasi

Kualitas informasi dapat diukur dari tiga hal, yaitu [JOG99] :

1. Akurat, berarti suatu informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan serta jelas mencerminkan maksudnya.
2. Tepat waktu, berarti informasi yang datang pada penerima tidak boleh terlambat. Suatu informasi harus sesuai dengan keadaan saat itu. Keterlambatan suatu informasi bisa berakibat fatal bagi suatu organisasi atau pemakainya hal ini dikarenakan informasi merupakan landasan dalam pengambilan keputusan.
3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk setiap orang berbeda-beda tergantung pada penggunanya.

3.2.3 Nilai Informasi

Suatu informasi dikatakan bernilai jika manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkannya, ini berarti nilai dari suatu informasi (*value of information*) ditentukan dari dua hal yaitu manfaat dan biaya. Kegunaan informasi adalah untuk mengurangi hal ketidakpastian di dalam proses pengambilan keputusan tentang suatu keadaan. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan didalam suatu informasi umumnya digunakan untuk beberapa kegunaan. Sehingga tidak memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah yang tertentu dengan biaya untuk memperolehnya. Sebagian besar informasi tidak dapat persis ditaksir

keuntungannya dengan suatu nilai uang, tetapi dapat ditaksir dari nilai efektifitasnya.

3.3 Konsep Sistem Informasi

3.3.1 Definisi Sistem Informasi

Definisi sistem informasi diantaranya:

"Sistem Informasi adalah sebuah sistem manusia/mesin yang terpadu untuk menyajikan informasi guna mendukung fungsi operasi, manajemen, dan pengambilan keputusan dalam sebuah organisasi. Sistem ini menggunakan perangkat keras dan perangkat lunak komputer, prosedur pedoman, model manajemen dan keputusan, dan sebuah basis data" [DAV93].

"Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan" [ROB83].

"Sistem informasi adalah suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal

yang penting dan menyediakan suatu dasar informasi untuk pengambila keputusan yang cerdik" [JOG99].

"Sistem informasi adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan laporan yang diperlukan" [JOG88].

Suatu sistem informasi pada dasarnya terbentuk melalui suatu kelompok kegiatan operasional yang tetap yaitu mengumpulkan data, mengelompokkan data, menghitung data, menganalisa data, dan menyajikan laporan.

3.3.2 Komponen Sistem Informasi

Sistem informasi dapat terdiri dari beberapa komponen-komponen yang disebut dengan istilah blok bangunan (*bulding block*). Sebagai sistem blok-blok tersebut saling berinteraksi membentuk satu kesatuan untuk mencapai tujuannya, blok bangunan tersebut terdiri dari enam komponen yaitu [JOG99]:

1. **Blok Masukan (*Input Block*)**

Input mewakili data yang masuk ke dalam sistem informasi. *Input* disini mewakili metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. **Blok Model (*Model Block*)**

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirim keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari tiga bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok Basis Data (*Database Block*)

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data didalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas.

6. Blok Kendali (*Control Block*)

Agar sistem informasi dapat berjalan dengan lancar sesuai dengan yang diinginkan maka perlu diterapkan pengendali-pengendali di dalamnya. Beberapa

pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan dapat segera diatasi.

3.3.3 Kriteria Sistem Informasi

Secara umum kriteria sebuah sistem informasi yaitu:

a. *Debit* yaitu jumlah data dan informasi (*bits*) yang mengalir per satuan waktu. Makin rendah makin baik tanpa mengorbankan fungsinya.
b. *Response Time* yaitu waktu antar kejadian, reaksi terhadap kejadian sampai dengan proses terhadap kejadian selesai dilakukan, semakin cepat semakin baik.
c. *Cost* yaitu biaya, semakin rendah biaya yang dikeluarkan semakin baik.
d. Pemenuhan fungsi, yaitu suatu sistem informasi harus berjalan sesuai dengan fungsinya.

3.3.4 Contoh-contoh Sistem Informasi

Adapun contoh-contoh sistem informasi yang ada disekitar kita diantaranya :

1. Sistem Informasi Poliklinik

Sistem informasi poliklinik ini adalah sistem informasi yang mengelola data pasien dan pengobatannya, data dokter dan jadwal tugasnya, stok dan

penyediaan obat-obatan, laporan penjualan obat, administrasi dan keuangannya, termasuk honorarium dokter.

2. Sistem Informasi Sanggar Tari

Sistem informasi sanggar tari ini adalah sistem informasi yang mengelola data dan informasi mengenai kegiatan tari-tarian (termasuk unsur-unsur biaya yang terlibat di dalamnya), serta membuat laporan bulanan keuangan dan administrasinya.

3. Sistem Informasi Kepegawaian

Sistem informasi kepegawaian adalah sistem informasi kepegawaian untuk kepentingan penggajian, sistem informasi ini mengelola data pegawai dan sistem ini dapat mengeluarkan daftar gaji pegawai dan slip gaji pegawai, serta dapat mencetak data induk pegawai.

4. Sistem Informasi Pengelolaan ATK

Sistem informasi pengelolaan ATK adalah sistem yang mengelola ATK (buku, kertas, formulir, pita printer dan lain-lain), sistem ini menghasilkan laporan mengenai biaya pemakaian ATK selama satu bulan yang ditujukan untukk pengajuan anggaran biaya ATK suatu badan pemerintah atau instansi perusahaan.

5. Sistem Informasi Supermarket

Sistem informasi supermarket adalah sistem informasi yang mengelola data barang-barang yang ada disupermarket mulai dari data barang masuk sampai data barang keluar yang berguna untuk bagian pembelian dan penjualan suatu

supermarket, sistem ini juga menghasilkan laporan harian dan bulanan barang yang terjual dan barang yang dibeli dari distributor.

3.4 Metoda Pembangunan Sistem Informasi

3.4.1 Metode Rekayasa Perangkat Lunak dengan Model Waterfall [PRE97]

Pengembangan sebuah sistem informasi berbasis komputer dapat menggunakan beberapa metode sebagai acuan. Setiap metode akan dibagi menjadi tahapan-tahapan yang akan memudahkan dalam pembangunan sistem informasi. Metode yang sering juga disebut metode "*waterfall*" atau "*classic life cycle*" ini menggunakan pendekatan yang sistematis dan sekuensial dalam membangun perangkat lunak yang dimulai pada level sistem dan pengembangan melalui tahapan analisis, perancangan, pengkodean, pengujian, dan pemeliharaan.

a. Analisis

Dari rumusan sistem yang diperoleh dari tahap pertama, selanjutnya dilakukan analisis yang berkaitan dengan proses dan data yang diperlukan oleh sistem serta keterkaitannya. Tujuan dilakukan tahapan ini adalah untuk memahami sistem yang ada pada saat ini agar dapat mendefinisikan permasalahan sistem sehingga selanjutnya dapat menentukan kebutuhan sistem secara garis besar sebagai persiapan ke tahap perancangan. Analisis di sini dilakukan dengan pemodelan menggunakan dengan *tool Unified Modelling Language (UML)*

b. Perancangan

Pada tahap perancangan ini diberikan gambaran umum yang jelas kepada pengguna dan rancang bangun yang lengkap tentang sistem yang akan dikembangkan kepada pihak-pihak yang terlibat dalam pengembangan sistem. Tahap perancangan ini dilakukan sebagai persiapan untuk tahap implementasi.

c. Implementasi/*Coding*/Pemrograman

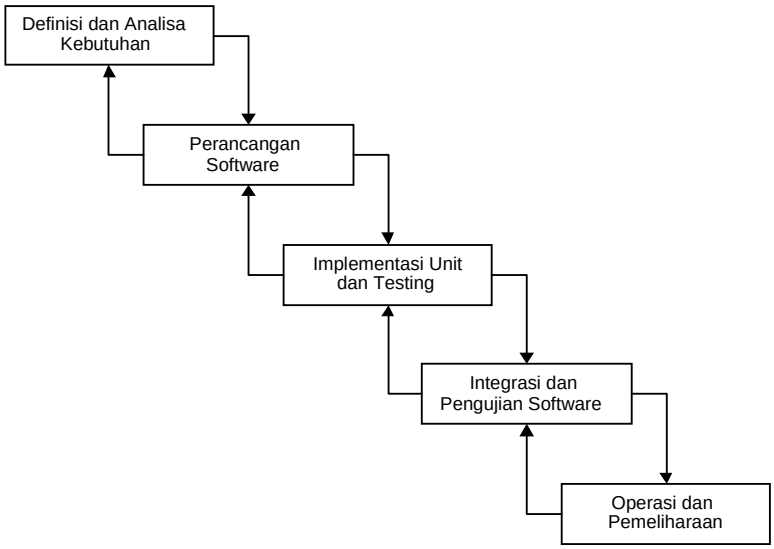
Setelah tahap perancangan sistem, selanjutnya dilakukan konversi rancangan sistem ke dalam kode-kode bahasa pemrograman yang diinginkan. Pada tahap ini dilakukan pembuatan komponen-komponen sistem yang meliputi modul program, antarmuka dan basis data.

d. Pengujian

Tahap pengujian ini dilakukan untuk mendapatkan serta memastikan bahwa perangkat lunak yang dihasilkan adalah valid dan sesuai dengan kebutuhan yang telah dideskripsikan.

e. Pemeliharaan

Pada tahap pemeliharaan ini perangkat lunak sudah diserahkan kepada pengguna. Pada tahap ini dilakukan evaluasi terhadap sistem yang baru untuk mengetahui apakah sistem telah memenuhi tujuan yang ingin dicapai. Dari hasil evaluasi ini dimungkinkan untuk melakukan perubahan-perubahan yang diperlukan terhadap sistem agar sistem senantiasa dapat digunakan dengan baik



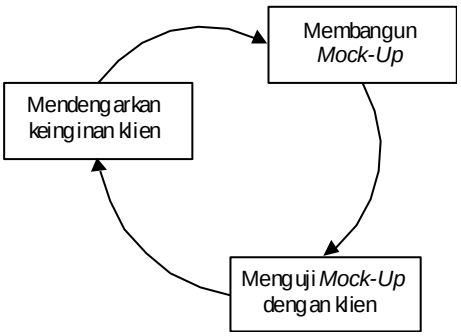
Gambar 3.2 Metode RPL dengan Metode Waterfall

Sumber : Roger S Pressman, *Software Engineering a Practitioner's Approach* Fourth Edition.

3.4.2 Metode Rekayasa Perangkat Lunak dengan Model Prototipe (*Prototyping Model*) [TRI99]

Model ini bertujuan untuk membuat *prototype* dari perangkat lunak yang akan dikembangkan. Kelemahan model ini adalah :

1. *User* cenderung menganggap bahwa *prototype* tersebut adalah produk yang hampir jadi dan hanya memerlukan sedikit perbaikan.
2. Pengembang seringkali melakukan kerja sama untuk membuat *prototype* cepat berfungsi, sehingga implementasi dilakukan dengan kurang mempertimbangkan aspek jangka panjang sistem yang dikembangkan.



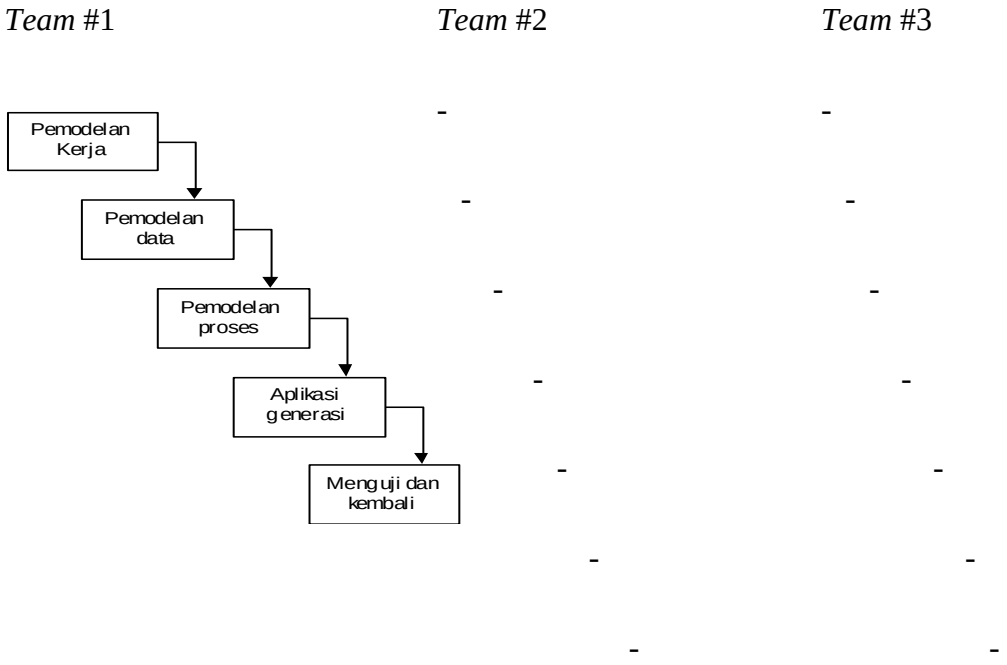
Gambar 3.3 Metode RPL dengan Model Prototipe

Sumber : Trida, *Modul Kuliah Manajemen Perangkat Lunak Semester VIII*

3.4.3 Metode Rekayasa Perangkat Lunak dengan Model RAD (*Rapid Application Development*) [TRI99]

Merupakan model pengembangan perangkat lunak secara sekuensial dengan siklus pengembangan yang sangat singkat. Kelemahan model ini adalah :

1. Untuk proyek dengan skala besar, diperlukan jumlah SDM yang memadai.
2. Pengembang dan *user* harus memiliki komitmen waktu yang tinggi untuk dapat melaksanakan model ini dengan cepat.
3. Aplikasi tertentu tidak cocok dikembangkan dengan model ini, seperti aplikasi-aplikasi untuk sistem yang tidak modular, sistem yang memerlukan kinerja tinggi, dan sistem dengan teknologi baru yang memiliki resiko bisnis lebih besar.



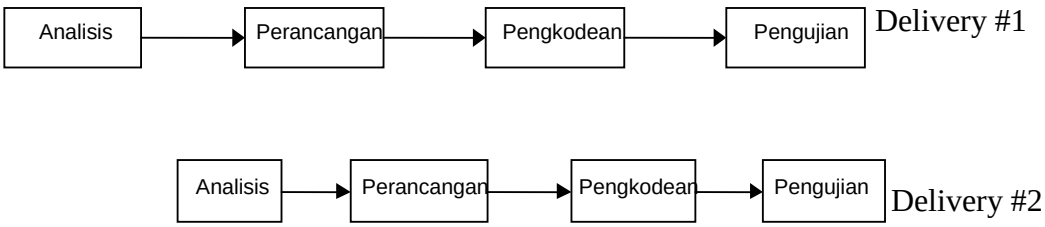
Gambar 3.4 Metode RPL dengan Model RAD

Sumber : Trida, *Modul Kuliah Manajemen Perangkat Lunak Semester VIII*

3.4.4 Metode Rekayasa Perangkat Lunak dengan Model Inkremental (*Incremental Model*) [TRI99]

Model ini menggabungkan elemen dari model linier sekuensial dengan filosofi pengulangan pada model *prototype*. Model ini terdiri dari beberapa linier sekuensial yang menghasilkan inkremen perangkat lunak.

Model inkremental cocok diterapkan pada proyek yang memiliki keterbatasan SDM. Selain itu model ini juga baik untuk meminimalkan resiko teknis yang mungkin terjadi.



……

Gambar 3.5 Metode RPL dengan Model Inkremental

Sumber : Trida, *Modul Kuliah Manajemen Perangkat Lunak Semester VIII*

3.4.5 Metode Rekayasa Perangkat Lunak dengan Model Spiral (*Spiral Model*) [TRI99]

Model ini menggabungkan aspek pengulangan pada model prototipe dan aspek sistematis dari model linier sekuensial. Pada model ini, perangkat lunak dikembangkan melalui serangkaian rilis yang meningkat.

Model spiral dapat digunakan sepanjang hidup *software*, artinya untuk perubahan yang mungkin terjadi setelah *software* diimplementasikan dapat dilakukan dengan menggunakan model ini. Model ini sangat realistis untuk diterapkan dalam pengembangan software skala besar. Disamping itu, model ini memiliki mekanisme untuk meminimalkan resiko. Namun pengurangan resiko ini tentu saja sangat bergantung pada keahlian personil yang melakukan penilaian resiko tersebut.

3.4.6 Metode Rekayasa Perangkat Lunak dengan Model Perakitan Komponen (*Component Assembly Model*) [TRI99]

Model ini dikembangkan dalam rangka teknologi berorientasi objek, dimana kelas yang ada merupakan komponen yang dapat digunakan untuk berbagai aplikasi yang berbeda. Model perakitan komponen ini mirip dengan model spiral, dimana perbedaannya hanya pada tahap konstruksi perangkat lunak.

3.4.7 Metode Rekayasa Perangkat Lunak dengan Model Pengembangan Konkuren (*Concurent Development Model*) [TRI99]

Model pengembangan konkuren memfokuskan pada keadaan dari suatu proyek dimana beberapa aktivitas pada tahap yang berbeda dapat terjadi secara konkuren. Model ini direpresentasikan sebagai serangkaian aktivitas teknis, tugas dan keadaan dari aktifitas tersebut. Maka dengan demikian semua aktivitas ada secara bersamaan/konkuren, tetapi dalam keadaan yang berbeda. Model ini mendefinisikan serangkaian kejadian (*event*) yang dapat merubah suatu aktivitas dari suatu keadaan ke keadaan yang lain.

3.4.8 Metode Rekayasa Perangkat Lunak dengan Model Metode Formal (*Formal Methods Model*) [TRI99]

Model metode formal menekankan kegiatan pengembangan *software* (spesifikasi, pengembangan, dan verifikasi) menggunakan notasi matematis. Penggunaan model ini menjanjikan suatu produk *software* yang bebas dari cacat. Namun demikian, masih terdapat hambatan dalam pembangunan model ini dalam lingkungan bisnis, antara lain karena :

1. Biaya pengembangan model formal sangat lama dan mahal.
2. Diperlukan pelatihan bagi pengembang *software* untuk dapat menggunakan model ini.
3. Sulit digunakan sebagai mekanisme komunikasi dengan *user* yang awam.

3.5 Basis Data Relasional

Model basis data menunjukkan suatu cara yang digunakan untuk mengelola jaringan data secara fisik dalam memori sekunder yang akan berdampak pada bagaimana mengelompokkan dan membentuk keseluruhan data yang terkait dalam sistem yang sedang kita tinjau. Pada sub bab ini akan diuraikan istilah-istilah yang berkaitan dengan teori basis data.

3.5.1 Abstraksi Data

Abstraksi data merupakan tingkatan dalam melihat data sebuah sistem basis data. Tujuannya adalah untuk menyederhanakan interaksi pengguna dengan sistem. Abstraksi data terbagi dalam tiga level [SIL97], yaitu :

1. Level Fisik (*Physical Level*) merupakan level terendah dalam abstraksi data, yang menunjukkan bagaimana sesungguhnya suatu data disimpan. Pada level ini kita berurusan dengan data sebagai teks, angka atau dapat melihatnya sebagai himpunan bit data.
2. Level Lojik/Konseptual (*Conceptual Level*) merupakan level yang lebih tinggi dari level fisik, yang menggambarkan data apa yang sebenarnya (secara fungsional) disimpan dalam basis data dan hubungannya dengan data yang lain.
3. Level Penampakkan (*View Level*) merupakan level tertinggi dari abstraksi data, yang hanya menunjukkan sebagian dari basis data. Data yang diperlihatkan dapat mewakili relasi antar tabel.

3.5.2 Model Unified Modelling Language

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasaberorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering). Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi

Konsepsi dasar UML Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Sebenarnya konsepsi dasar UML bisa kita rangkumkan dalam gambar dibawah.

Major Area	*View*	*Diagrams*	*Main Concepts*
structural	static view	class diagram	class, association, generalization, dependency, realization, interface
	use case view	use case diagram	use case, actor, association, extend, include, use case generalization
	implementation view	component diagram	component, interface, dependency, realization
	deployment view	deployment diagram	node, component, dependency, location
dynamic	state machine view	statechart diagram	state, event, transition, action
	activity view	activity diagram	state, activity, completion transition, fork, join
	interaction view	sequence diagram	interaction, object, message, activation
		collaboration diagram	collaboration, interaction, collaboration role, message
model management	model management view	class diagram	package, subsystem, model
extensibility	all	all	constraint, stereotype, tagged values

Gambar 3.6 Konsep UML

Sumber : www.IlmuKomputer.com

Abstraksi konsep dasar UML yang terdiri dari *structural classification*, *dynamic behavior*, dan *model management*, bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari *Diagrams*. *Main concepts* bisa kita pandang sebagai term yang akan muncul pada saat kita membuat diagram. Dan view adalah kategori dari diagaram tersebut.

Seperti juga tercantum pada gambar diatas UML mendefinisikan diagram-diagram sebagai berikut:

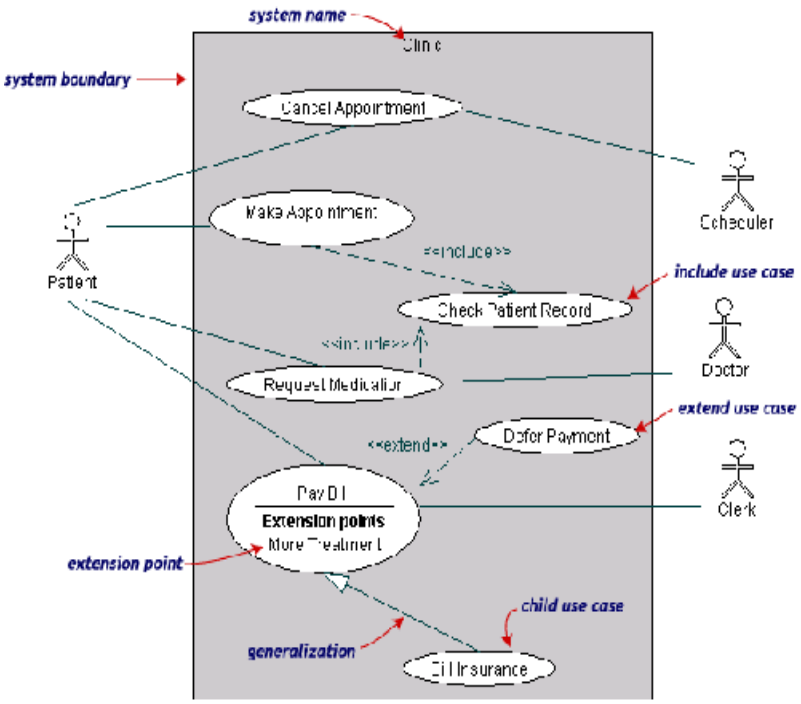
- *use case diagram*
- *class diagram*

- *statechart diagram*
- *activity diagram*
- *sequence diagram*
- *collaboration diagram*
- *component diagram*
- *deployment diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use*

case yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah

use case juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Contoh Use case diagram adalah sebagai berikut :



Gambar 3.7 Use Case Diagram

Sumber : www.IlmuKomputer.com

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus

menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)

2. Atribut

3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

• *Private,* tidak dapat dipanggil dari luar *class* yang bersangkutan

• *Protected,* hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya

• *Public*, dapat dipanggil oleh siapa saja



Gambar 3.8 atribut Class Diagram

Sumber : www.IlmuKomputer.com

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.

Hubungan Antar *Class*

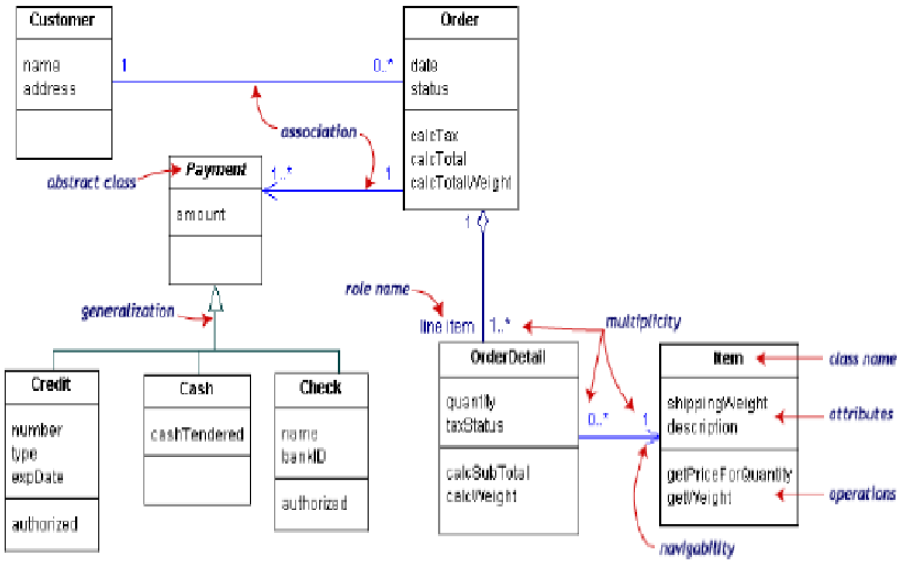
1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.

2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).

3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan

menggunakan *sequence diagram* yang akan dijelaskan kemudian. Contoh *class diagram* :



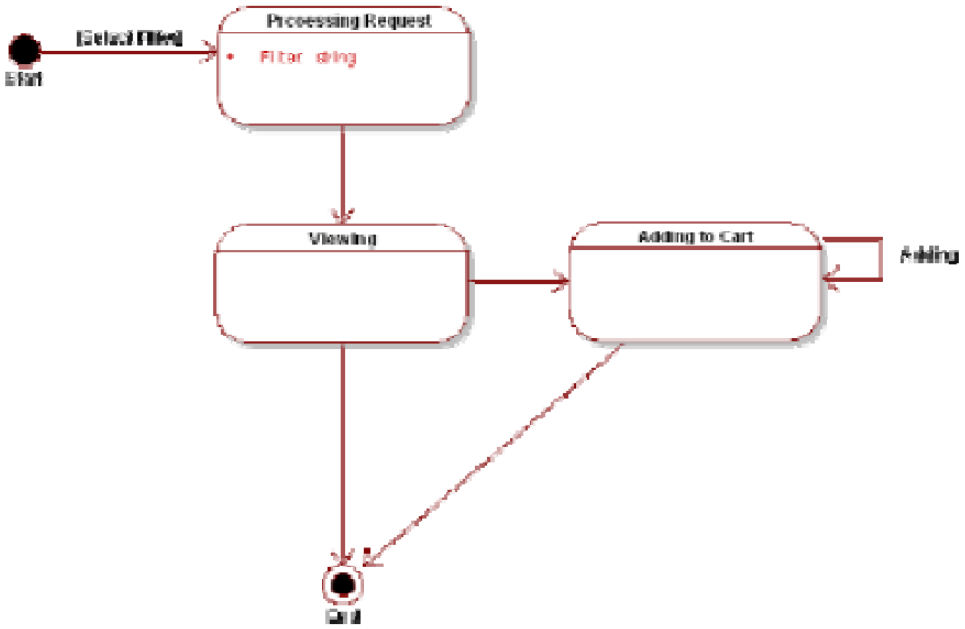
Gambar 3.9 contoh Class Diagram

Sumber : www.IlmuKomputer.com

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*). Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki

namasesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

Contoh *statechart diagram* :

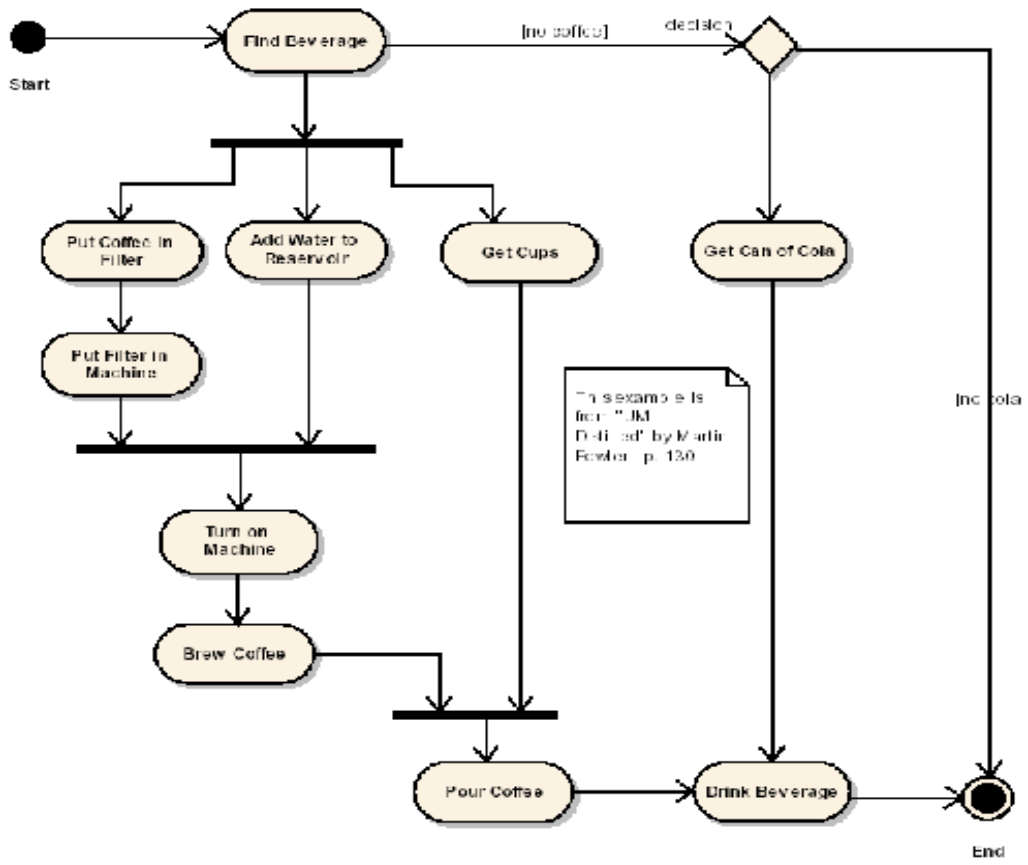


Gambar 3.10 contoh Statechart

Sumber : www.IlmuKomputer.com

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Contoh *activity diagram* tanpa *swimlane*:



Gambar 3.11 contoh Activity Diagram

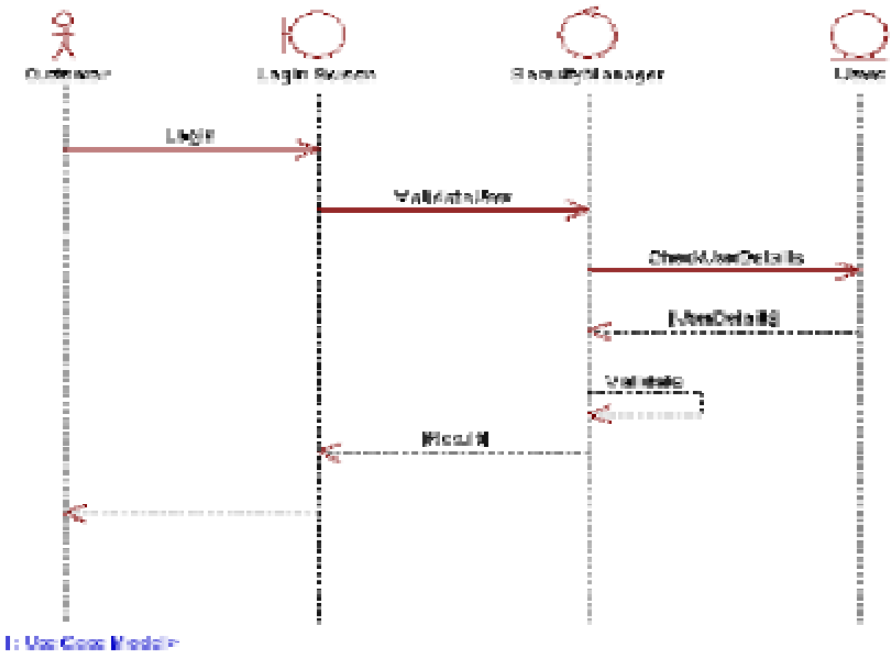
Sumber : www.IlmuKomputer.com

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkahyang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*.

Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.

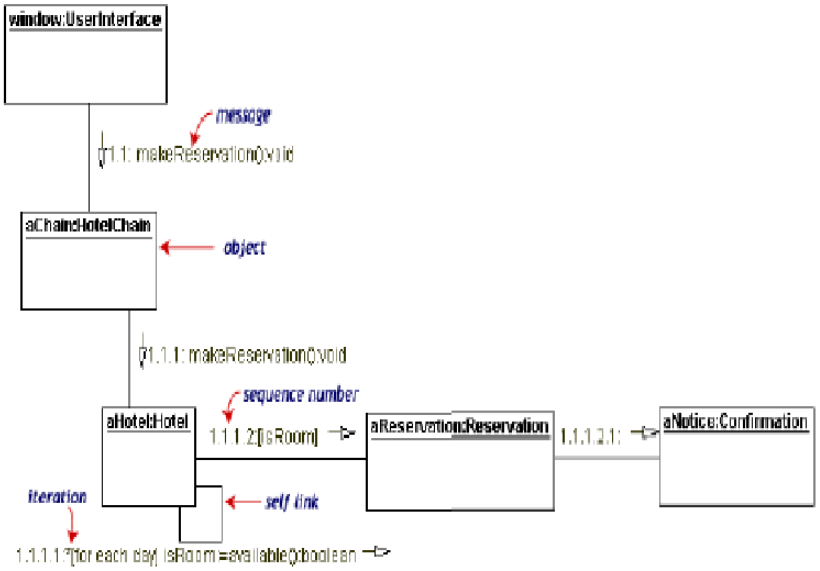
Contoh *sequence diagram* :



Gambar 3.12 contoh Sequence Diagram

Sumber : www.IlmuKomputer.com

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi. Messages dari level yang sama memiliki prefiks yang sama.

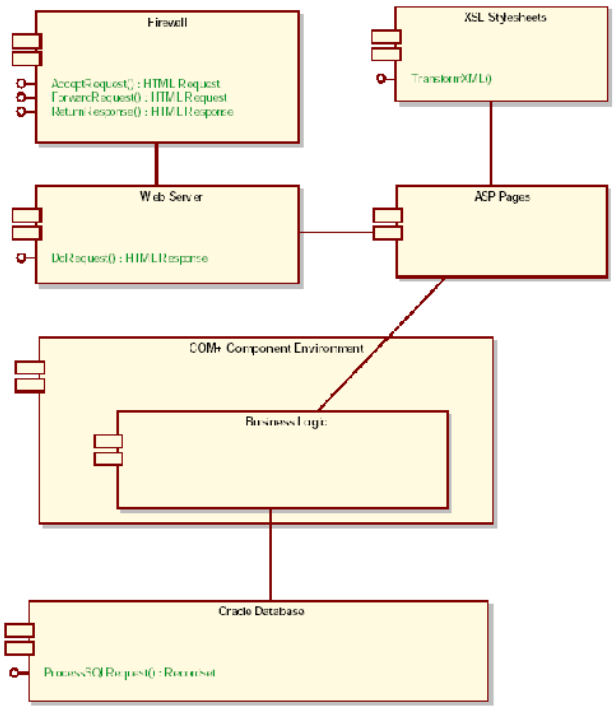


Gambar 3.13 contoh Collaboration Diagram

Sumber : www.IlmuKomputer.com

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya.Komponen piranti lunak adalah modul berisi *code*, baik berisi *source*

code maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain. Contoh *component diagram*:

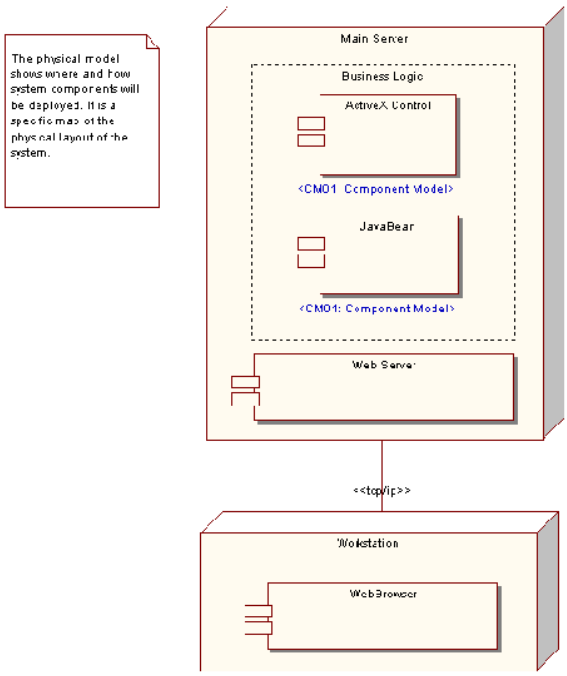


Gambar 3.14 contoh Component Diagram

Sumber : www.IlmuKomputer.com

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak

(pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.



Gambar 3.15 contoh Deployment Diagram

Sumber : www.IlmuKomputer.com

3.5 Perangkat Lunak (*Software*)

Perangkat lunak (*software*) dapat dikategorikan ke dalam 3 bagian, yaitu [JOG88] :

1. Perangkat Lunak Sistem Operasi (*Operating System*), yaitu program yang ditulis untuk mengendalikan dan mengkoordinasikan kegiatan dari sistem komputer.
2. Perangkat Lunak Bahasa (*Language Software*), yaitu program yang digunakan untuk menterjemahkan instruksi-instruksi yang ditulis dalam bahasa pemrograman ke dalam bahasa mesin supaya dapat dimengerti oleh komputer.
3. Perangkat Lunak Aplikasi (*Application Software*), yaitu program yang ditulis dan diterjemahkan oleh *language software* untuk menyelesaikan suatu aplikasi tertentu.

3.5.1 *Software* Sistem

Software sistem merupakan *background* program yang memungkinkan *software* aplikasi dapat berfungsi pada peralatan *hardware* sistem komputer [PAR96].

Software sistem dapat dibedakan menjadi 3 bagian, yaitu [PAR96] :

a. *Operating System*, adalah kumpulan utama dari program yang mengatur aktivitas sistem komputer. Contoh : *Windows '98*, *Macintosh System Software*, *MS-DOS*, *UNIX*, dan *Penpoint.*

Windows ’98 dilengkapi fasilitas-fasilitas yang dapat mempermudah kerja *user*, yaitu [HAR99] :

1. *Built-in Networking*, fasilitas jaringan terintegrasi di sistem operasi.
2. Pengamanan level pemakai *pass-through*
3. *Registry*, semua informasi mengenai sistem disatukan sehingga mempermudah pengelolaan sistem.

Pada *Windows ’98* antarmuka dengan sistem dilakukan dengan cara :

1. Kebanyakan dilakukan dengan menggunakan fasilitas berbasis grafik : *menu, icon*.
2. Tersedia *taskbar* yang mempermudah perpindahan aplikasi yang dijalankan sistem operasi.
3. Terdapat jendela untuk mengetikkan baris perintah secara langsung.

Keluarga *Windows* menyediakan kumpulan *system call* (API) yang dipanggil program aplikasi seperti kita memanggil program atau subrutin. Keluarga *Windows* dikembangkan dengan C, *system call* berbentuk fungsi bahasa C. Meskipun demikian kita dapat memanggil dalam bahasa apapun, kompilator atau interpreter menerjemahkan menjadi panggilan ke sistem operasi *Windows*.

Kumpulan *system call* dikeluarga *Windows* disebut WIN32 API. Bentuk fungsi, struktur, pesan, makro dan antarmuka konsisten dan seragam untuk semua keluarga *Microsoft* berbasis 32 bit. Penciptaan dan pengelolaan antarmuka pemakai, juga fitur-fitur *Windows*, yaitu *Carets*, *Clipboard*, *Cursors*, *Hooks*, *Icons* dan *Menus*. Kendali Umum pada *Windows* terdapat di *command control library*,

COMCTL32.DLL. *Common control* DLL berisi antarmuka pemrograman sehingga aplikasi dapat memanipulasi dan menciptakan *controls*, juga menerima masukan *controls*.

b. *Language Translator*, adalah suatu program sistem yang berfungsi untuk mengkonversikan program aplikasi ke dalam bahasa mesin. Contoh : *Language Translator* pada *COBOL, PASCAL, BASIC, C Language*.
c. *Utility Programs*, adalah suatu program yang menjelaskan atau memperluas kegunaan dari sistem operasi suatu sistem komputer. Contoh: *Backup utilities, Spooling utilities* dan *Data compression utilites*.

3.5.2 *Software* Aplikasi

Software aplikasi merupakan suatu perangkat yang memungkinkan pemakai (*user*) memahami sistem komputer [PAR96]. Berikut beberapa contoh dari *software* aplikasi :

1. *Microsoft Visual Basic* (VB)

Microsoft Visual Basic adalah bahasa pemrograman yang bekerja dalam lingkup *Microsoft Windows. Microsoft Visual Basic* bekerja menggunakan objek-objek sebagai komponen pemrogramannya. Setiap objek digambarkan pada layar dan melakukan pengaturan properti terhadap objek yang digambarkan.

Beberapa komponen dan manfaat *Visual Basic* diantaranya seperti:

a. Membuat program aplikasi berbasis windows
b. Untuk membuat objek-objek pembantu program seperti *Control Active, File Help*, aplikasi internet, dsb.
c. Menguji program (*debugging*) dan menghasilkan program akhir berakhiran EXE yang bersifat *executable* (file yang sudah dikompilasi tidak memerlukan lagi *Dynamic Link Libraries*) atau dapat dijalankan.

Beberapa dukungan yang diberikan oleh *Microsoft Visual Basic* adalah:

a. *Visual Basic* memungkinkan programnya untuk mengembangkan antarmuka pemakai secara mudah dan tepat.
b. *Visual Basic* merupakan produk pengembang dari Bahasa Basic, sebuah bahasa yang sangat populer terutama dalam lingkungan akademis. Meskipun lingkungan pengembangan yang sepenuhnya grafis dan bahasa pemrograman dari *Visual Basic* dengan *intepreter Basic* Awal, keluwesan dan kesederhanaan Basic yang semula tetap, sampai batas-batas tertentu.
c. *Visual Basic* mampu menghasilkan *file EXE* yang berdiri sendiri, kecuali jika aplikasi yang dibuat memerlukan pengaksesan terhadap basis data. Yang dimaksud dengan berdiri sendiri adalah file yang sudah dikompilasi tersebut tidak lagi memerlukan *Dinamic Link Libraries* (DLL).

d. *Visual Basic* menyediakan banyak objek kontrol yang berhubungan dengan pemrograman itu sendiri.

e. *Visual Basic* menyediakan *Database Desktop.*

Ada banyak fasilitas dalam VB yang sangat berguna dalam memngembangkan berbagai program, termasuk didalamnya pengembangan sistem pakar. Kelebihan lain dari VB adalah kemampuan untuk mengkompilasi program dala bentuk *Native Code*, yaitu optimasi pada saat prosesor mengkompilasi dan menjalankan program tersebut. Keuntungan dari *Native Code* adalah kecepatannya dalam mengakses program, dimana hal ini hanya dapat ditemui pada aplikasi-aplikasi yang dikompilasi dengan bahasa pemrograman C++. Selain kemampuan-kemampuan di atas, VB juga menyediakan fasilitas antarmuka penulisan kode program yang lebih mudah dimengerti dan dipakai sehingga berbagai tipe program dapat dikembangkan didalamnya, misalnya EXE, DLL dan OCX, bahkan program-program yang berbasis internet. Berbagai tipe program yang dapat dikembangkan tersebut disimpan dengan tipe-tipe pengenal *(ekstensi)* tertentu, yaitu sebagai berikut:

Form : FRM

Module Umum : BAS

Module Class : CLS

Control : CTL

ActiveX Document : DOB

File-file dengan ekstensi ini akan dioptimasi pada saat dikompilasi dengan *Native Code* untuk mendapatkan kecepatan dan ukuran file EXE (program yang dapat dijalankan di luar lingkungan VB) yang optimal.

2. *Microsoft Access 2000* [SEP00]

Pada *Micorosft Access 2000* terdapat tempat utama yang dipergunakan untuk bekerja yaitu *Database Window*. *Database Window* terdiri dari beberapa bagian yaitu :

a. *Tables*, adalah kumpulan data, merupakan komponen utama dari suatu *database*.
b. *Queries*, digunakan untuk mengatur data mana saja yang terdapat pada suatu *tables* yang akan ditampilkan.
c. *Forms*, digunakan untuk mengatur tampilan data di layar monitor.
d. *Reports*, digunakan untuk mengatur tampilan data yang akan dicetak dengan printer.
e. *Macros*, merupakan fasilitas untuk mengotomasikan sekaligus menghemat waktu yang diperlukan dalam pembuatan aplikasi *database*.
f. Modules, berguna dalam pembangunan aplikasi *database* tingkat lanjut.

Pada *access* telah tersedia berbagai cara yang dapat digunakan untuk pembuatan tabel. Salah satu pilihannya adalah dengan menggunakan *Design View*. Modus *Design View* akan menampilkan nama *field* (*Field Name*), jenis datanya (*Data Type*), serta sebuah kolom keterangan (*Description*) yang merupakan kolom

opsional (boleh tidak diisi). *Access* menyediakan berbagai pilihan jenis data, meliputi *Text, Memo, Number*, *Date/Time*, *Currency*, *AutoNumber*, *Yes/No, OLE Object, Hyperlinks, dan Lookup Wizard*. Bagian bawah dari *Design View* menampilkan properti bagi setiap *field*. Komponen-komponen *properties* hampir sama untuk berbagai jenis data, kecuali :

a. Ukuran field (*Field Size*), hanya berlaku untuk jenis data *Text* dan *Number*, sementara maknanya pun berbeda pada keduanya.
b. Letak Desimal (*Decimal Places*), hanya berlaku untuk *Number* dan *Currency*.
c. Properti *Format* bekerja secara berbeda untuk jenis data yang berlainan.

BAB IV

ANALISIS SISTEM

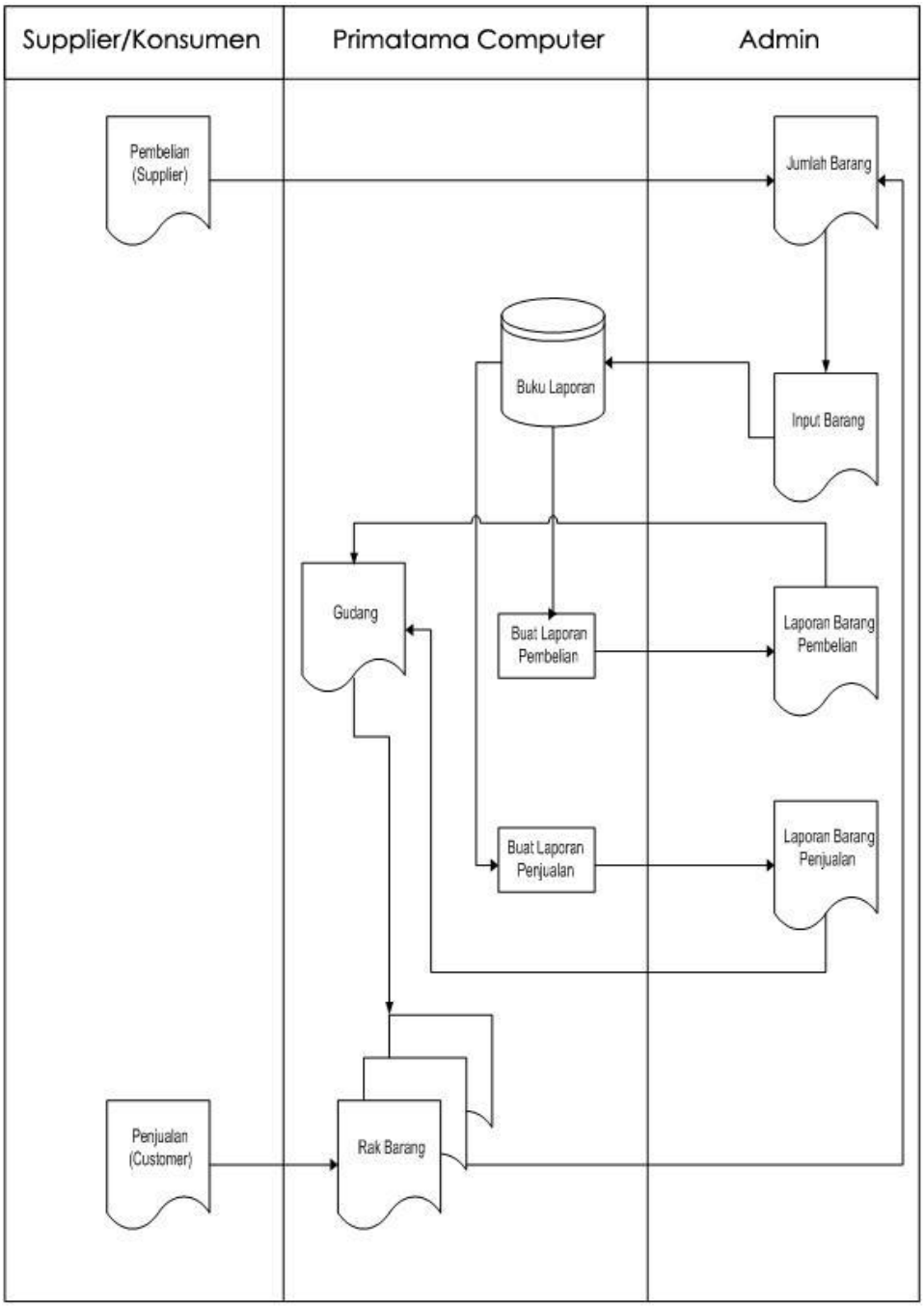
4.1 Identifikasi Masalah

Toko Primatama computer adalah salah satu usaha yang ingin menerapkan kenyamanan dalam pembelian atau penjualan bagi supplier maupun customer di toko tersebut. Informasi data yang ada saat ini dikelola melalui pencatatan secara manual. Aplikasi yang akan dibuat ini untuk membantu toko Primatama computer dalam mengolah sistem informasi penjualan dan pembelian yang meliputi :

1. Informasi data persediaan barang.
2. Informasi data penjualan dan pembelian barang.
3. Informasi data penjualan pembelian barang perhari, perbulan.

4.2 Prinsip Kerja Sistem yang Lama

4.2.1 Proses Pembelian dan Penjualan Barang



Gambar 4.1 Prosedur Sistem Dahulu

Data persediaan barang dilakukan dengan mendata barang secara manual ditulis dalam buku. Untuk saat ini informasi barang yang tersedia disimpan pada buku pembukuan pembukuan tanpa di komputerisasi. Informasi yang disimpan hanya nama barang, banyak barang dan harganya. Sedangkan informasi persediaan barang yang tersedia hanya diketahui oleh karyawan.

4.2.2 Pembuatan Laporan

Pembuatan laporan pembelian dan penjualan dilakukan perhari oleh staf administrasi.Laporan dibuat berdasarkan Transaksi pembelian dan penjualan barang. Laporan pembelian dan penjualan ditulis pada sebuah buku laporan.

4.2.3 Identifikasi Penyebab Masalah

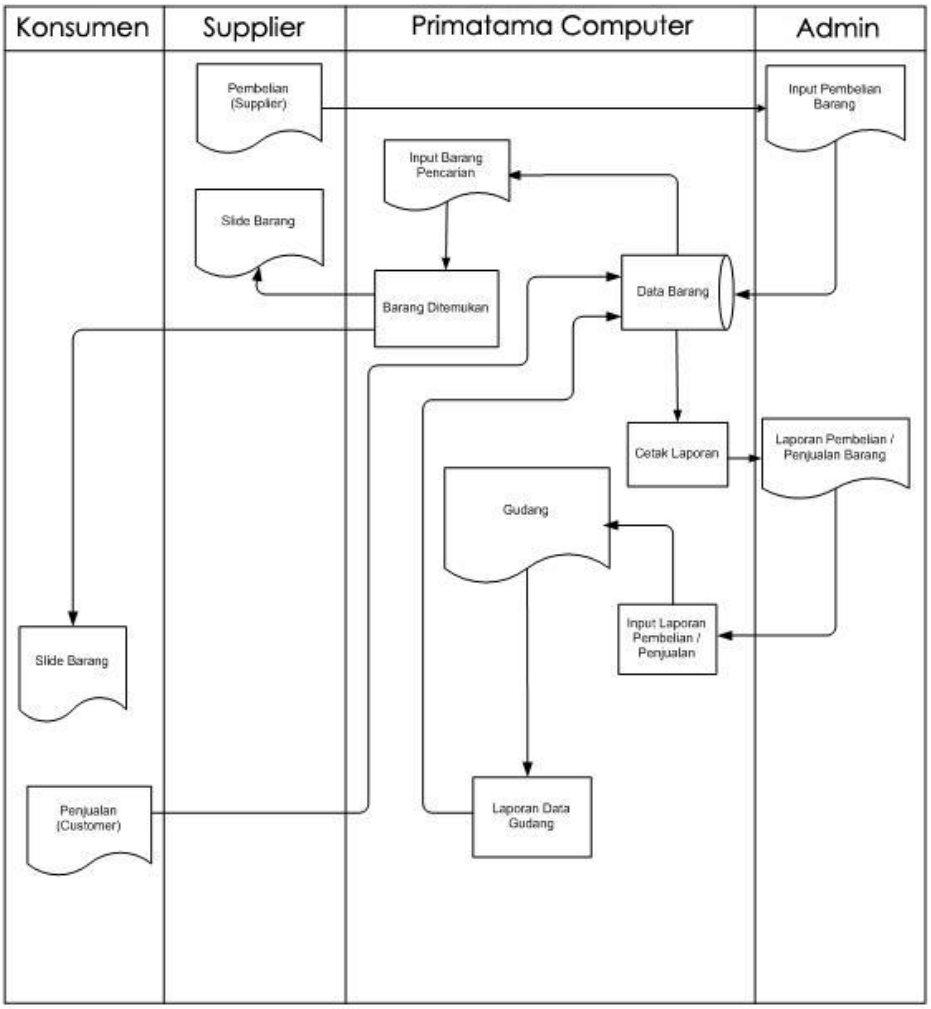
Sistem yang akan dibangun seharusnya dapat bekerja lebih baik atau dapat menunjukkan peningkatan untuk menutupi kelemahan sistem yang telah ada serta mengurangi kesalahan yang sering terjadi. Segala permasalahan efisiensi, efektifitas, kemudahan dan kecepatan adalah hal yang mendasar mengapa perlunya peningkatan mutu sebuah sistem.

4.2.4 Hasil Analisis Sistem Dahulu

Hasil analisis sistem di toko Primatama computer pada sistem dahulu yaitu :

1. Laporan pembelian dan penjualan yang dibuat perhari pada sistem yang lama ditulis pada sebuah buku. Laporan pembelian dan penjualan dibuat berdasarkan data barang yang terjual. Cara pembuatan laporan pembelian dan penjualan dengan cara ini bisa berakibat tidak akuratnya data karena bisa saja ada data yang hilang.
2. Informasi barang pada sistem yang lama disimpan pada buku secara manual. Pada buku tersebut hanya tertulis nama barang, banyak barang dan harga barang. Sedangkan informasi persediaan barang hanya diketahui karyawan pada kertas. Hal ini pasti beresiko hilangnya barang pada gudang penyimpanan. Bila ada pengambilan barang tanpa diketahui oleh pegawai atau tidak dicatat pada data persediaan barang maka informasi persediaan barang tidak lagi akurat, sehingga bisa berakibat keluarnya barang yang tidak terdata.
3. Laporan pembayaran terkadang mengalami kesalahan dalam memberi harga dan salah melihat nama barang, mengakibatkan data penjualan barang tidak konsisten.

4.1 Deskripsi Sistem yang akan dikembangkan



Gambar 4.2 Prosedur Sistem yang sedang dikembangkan

4.2.5 Hasil Analisis Sistem Yang Akan Dikembangkan

Berdasarkan identifikasi masalah dan penyebab masalah yang telah dijelaskan diatas dapat disimpulkan bahwa untuk memperbaiki sistem yang lama diperlukan langkah-langkah perbaikan sebagai berikut:

1. Pembuatan aplikasi yang mendukung kegiatan operasional perusahaan untuk mempermudah pembuatan laporan penbelian dan penjualan perusahaan.
2. Data persediaan barang akan disimpan pada database beserta kode, nama , dan harga barang untuk mempermudah perolehan informasi data barang dan pembuatan laporan pembelian dan penjualan barang.
3. Pembayaran barang yang telah diterima akan dicatat dan disimpan datanya di database dengan bantuan aplikasi yang akan dibuat agar informasi barang yang telah dibayar dapat dengan mudah diketahui. Selain itu nantinya akan digunakan untuk pembuatan laporan penbelian dan penjualan.
4. Laporan pembelian dan penjualan barang akan dibuat secara otomatis oleh aplikasi yang akan dibuat berdasarkan data persediaan barang yang terjual yang telah tersimpan di database, agar laporan yang dibuat memiliki informasi yang akurat.

4.2.6 Perancangan Umum

Langkah pertama dalam merancang suatu sistem adalah perancangan sistem tersebut secara logika yang akhirnya dapat dikembangkan menjadi suatu aplikasi. Pada tahap ini disusun model fungsi awal dengan menggunakan :

1. *Use Case Diagram*
2. *Activity Diagram*
3. *Sequence Diagram*
4. *Class Diagram*
5. *Colaboration Diagram*

4.2.7 Penjelasan Diagram

Interaksi diagram di atas dapat dilihat dengan jela pada pemodelan dibawah ini :

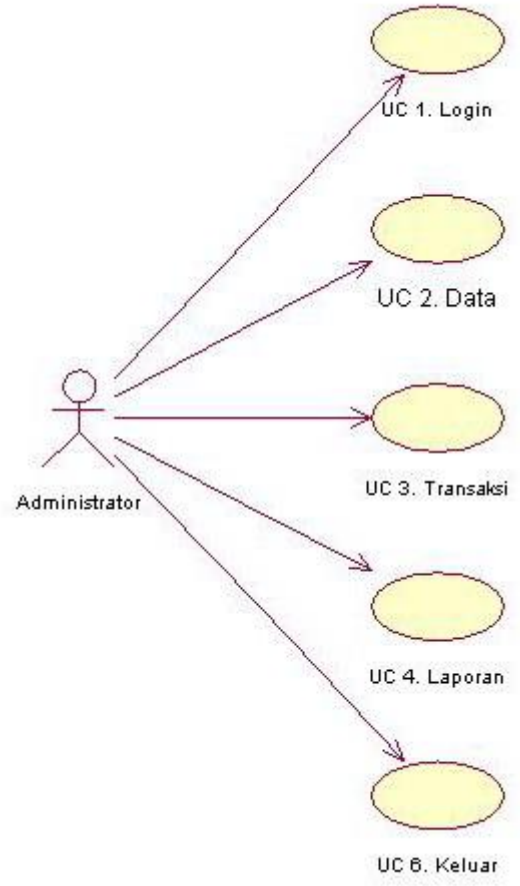
4.2.8 *Use Case* Diagram

Perancangan ini melibatkan beberapa aktor, diantaranya :

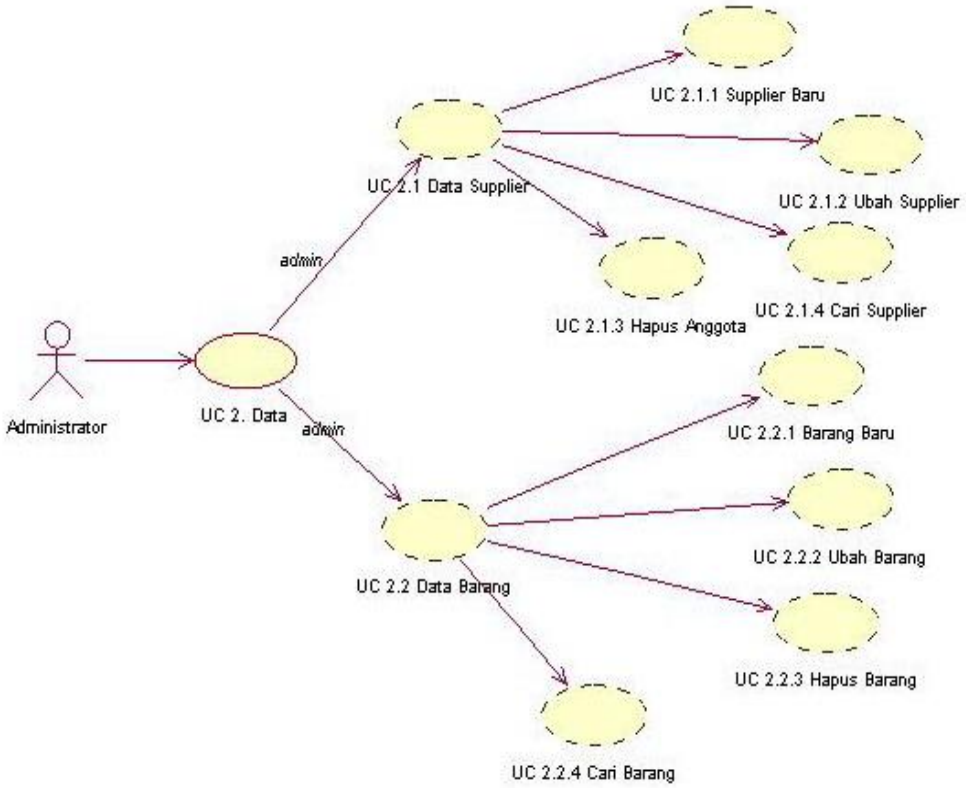
a. Operator sebagai yang mengoperasikan hasil akhir dari aplikasi.
b. Administrator sebagai yang menjalankan aplikasi secara keseluruhan.

Interaksi aktor diatas dapat dilihat dengan jelas pada pemodelan berikut :

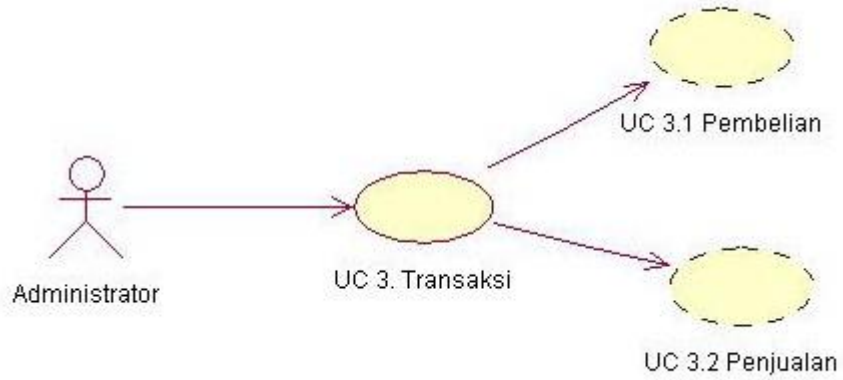
USE CASE DIAGRAM



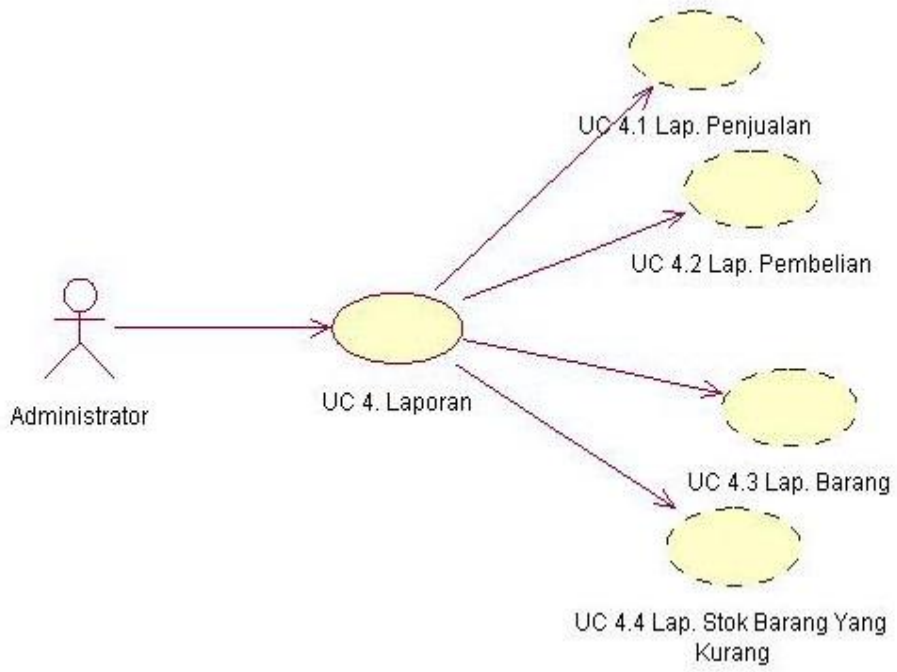
Gambar 5.1 *Use Case* Diagram Global Toko



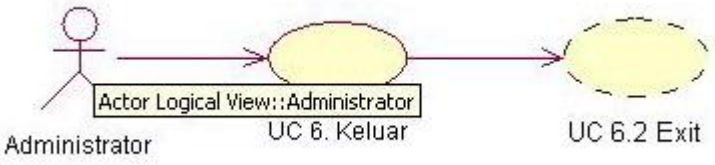
Gambar 5.2 *Use Case* Diagram Realisasi Data



Gambar 5.3 *Use Case* Diagram Realisasi Transaksi



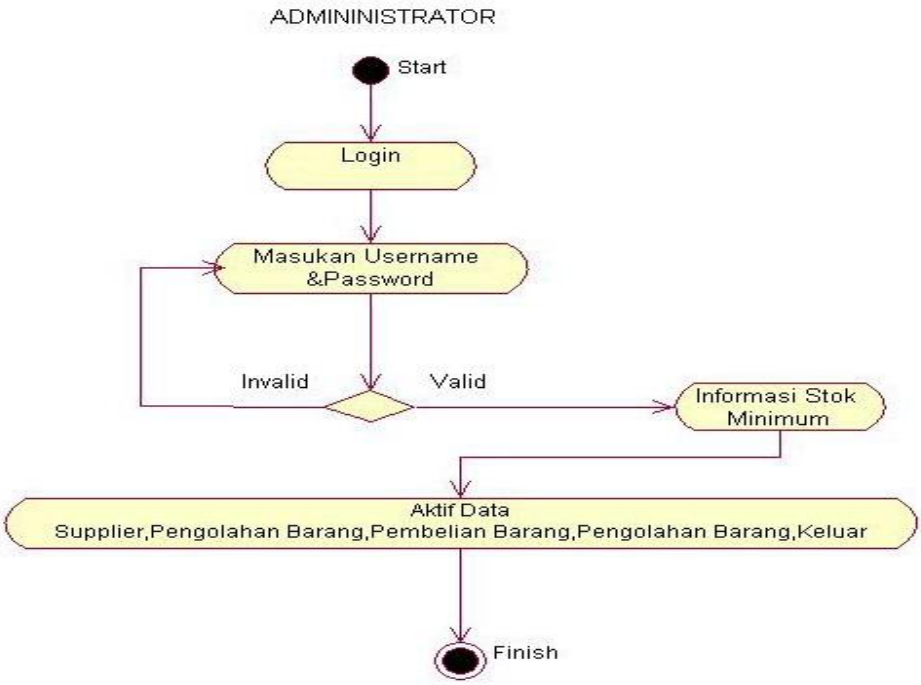
Gambar 5.4 *Use Case* Diagram Realisasi Global



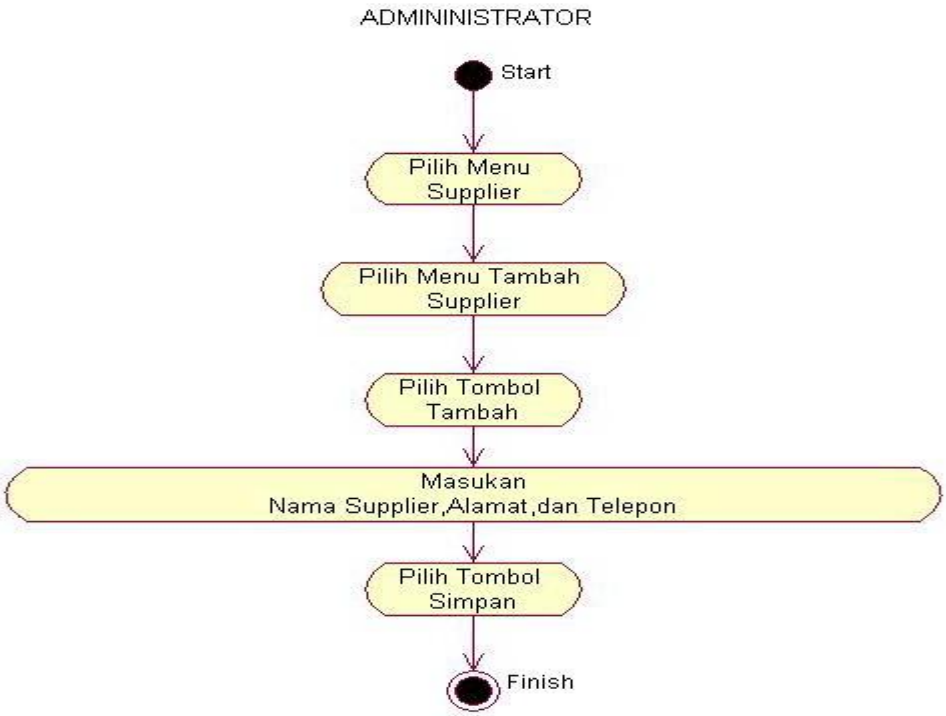
Gambar 5.5 *Use Case* Diagram Realisasi keluar

4.2.9 Activity Diagram

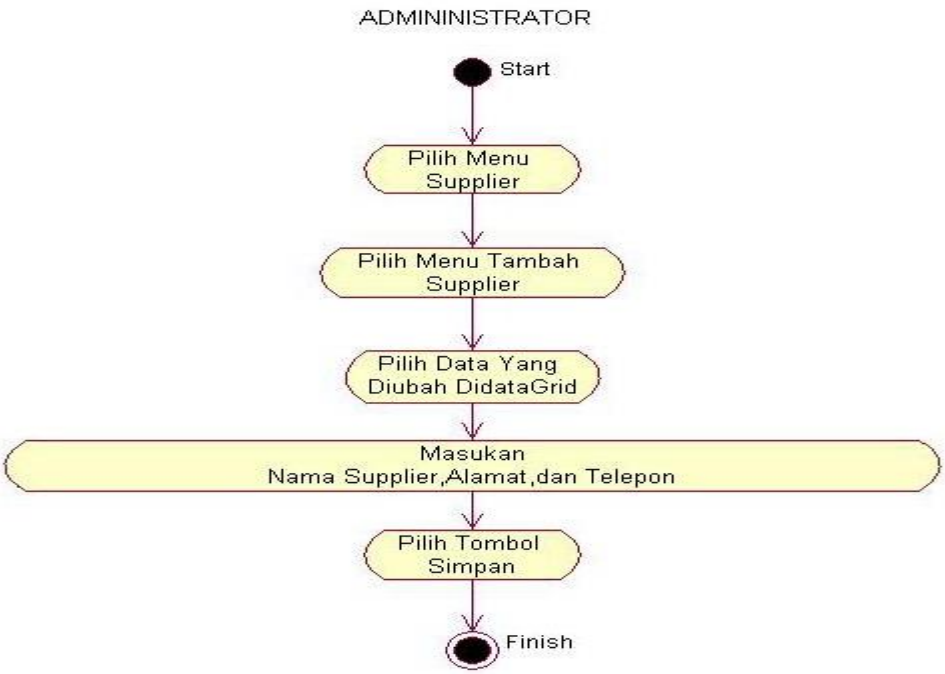
Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas laiinnya seperti *use case* atau interaksi.



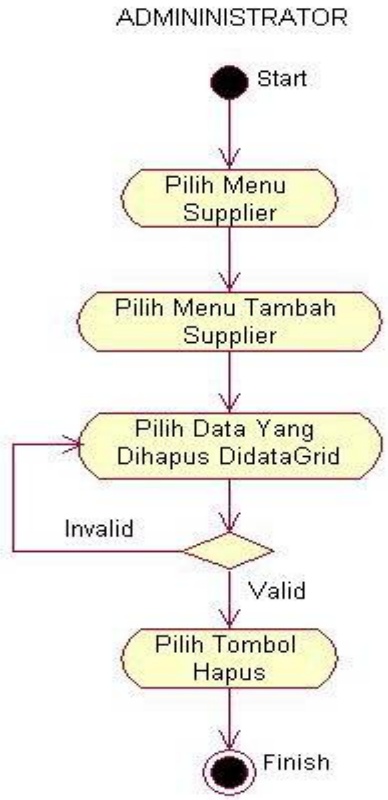
Gambar 5.6 *Activity* Diagram *Login* Admin



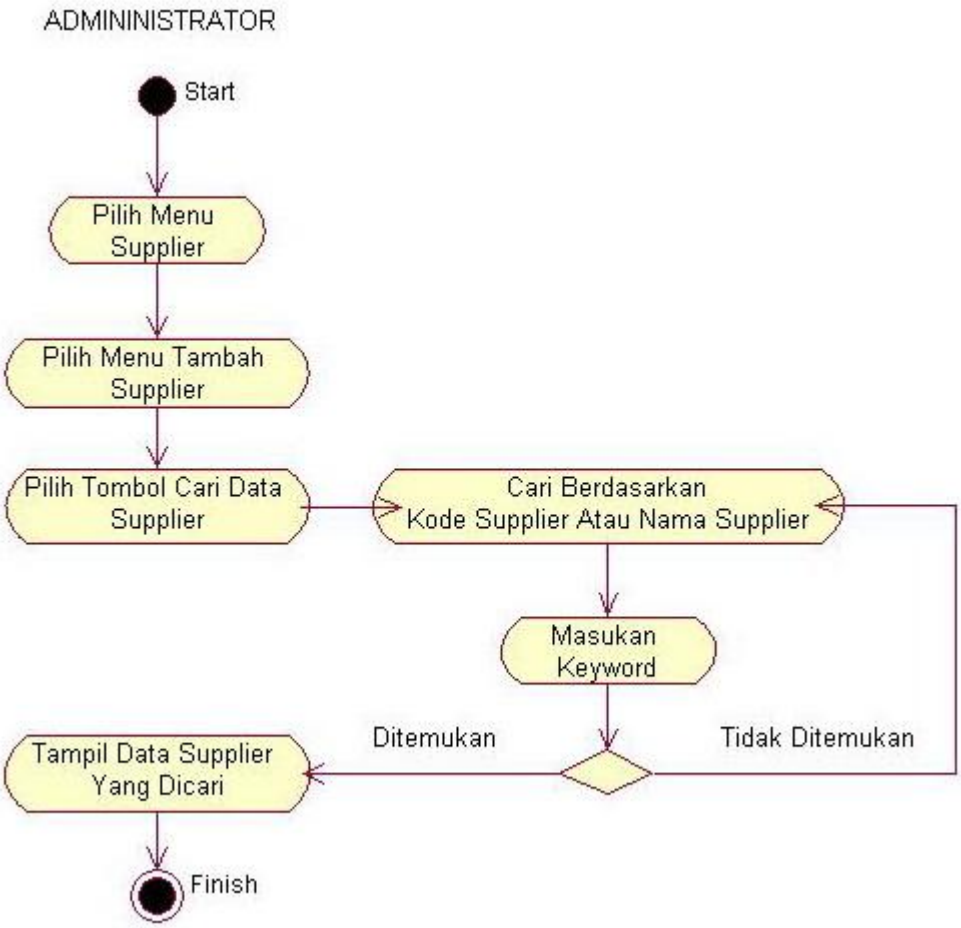
Gambar 5.7 *Activity* Diagram Supplier Baru



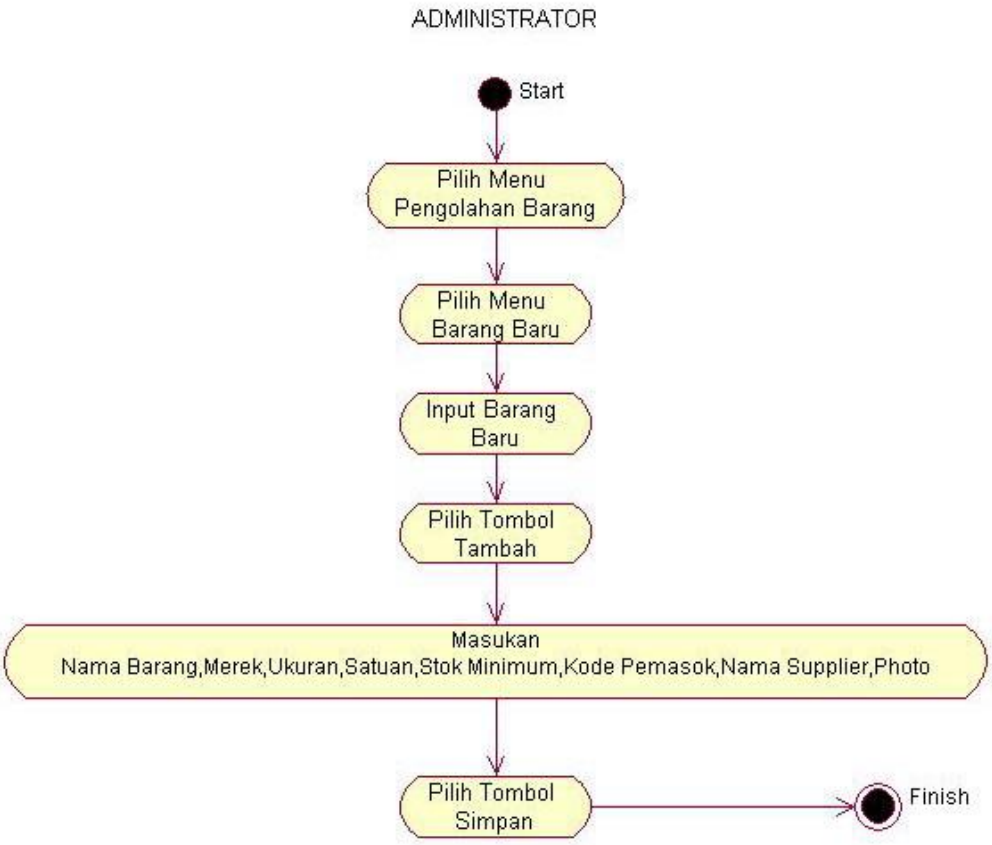
Gambar 5.8 *Activity* Diagram Ubah Supplier



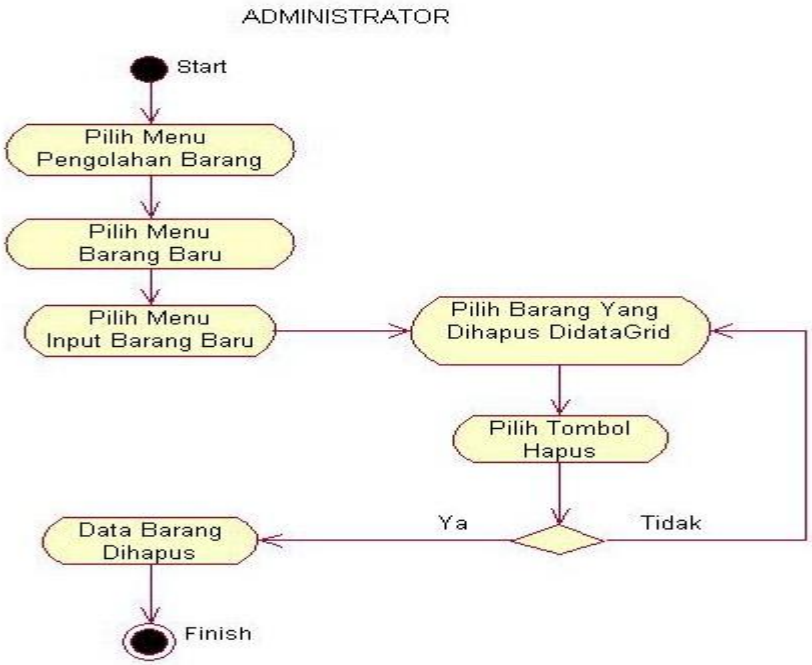
Gambar 5.9 *Activity* Diagram Hapus Supplier



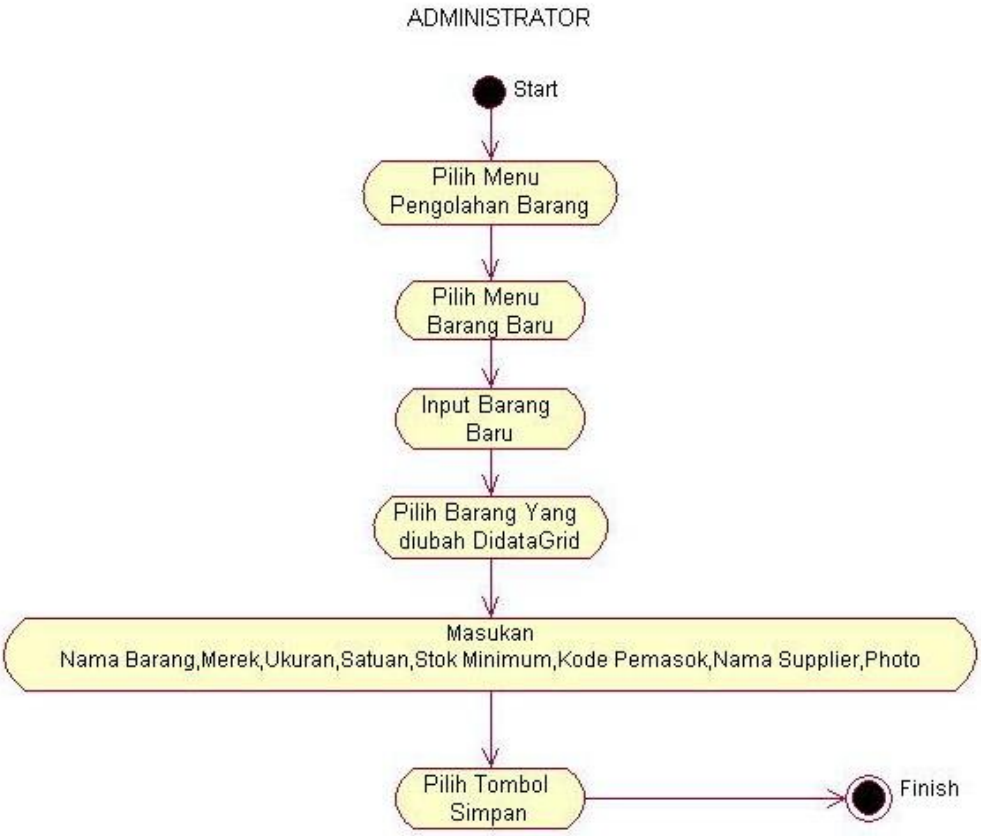
Gambar 5.10 *Activity* Diagram Cari Supplier



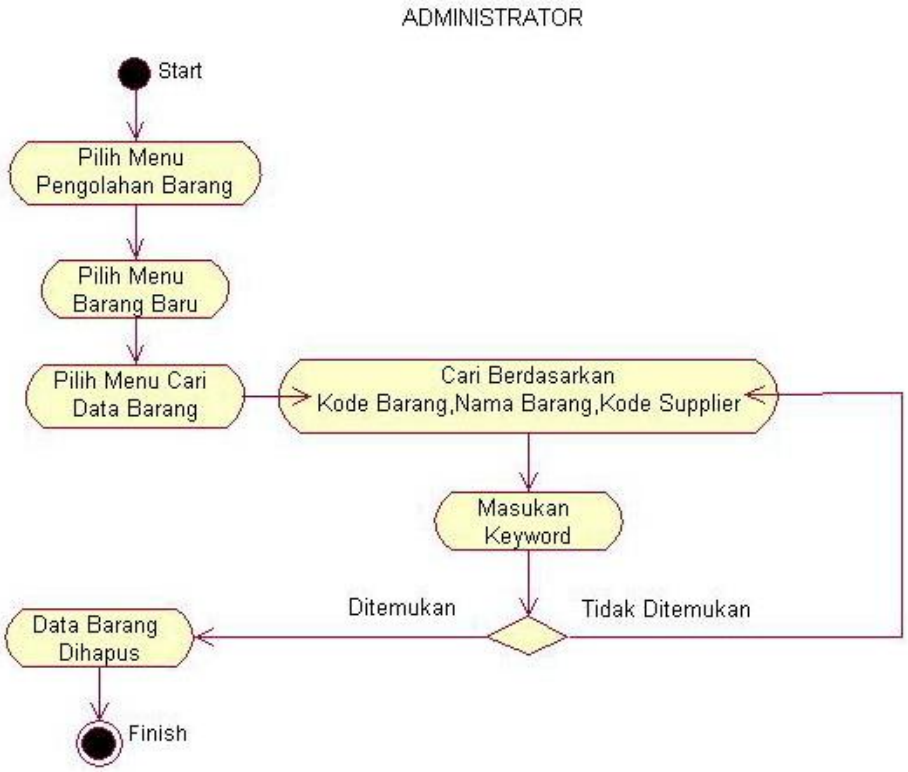
Gambar 5.11 *Activity* Diagram Barang Baru



Gambar 5.12 *Activity* Diagram Hapus Barang



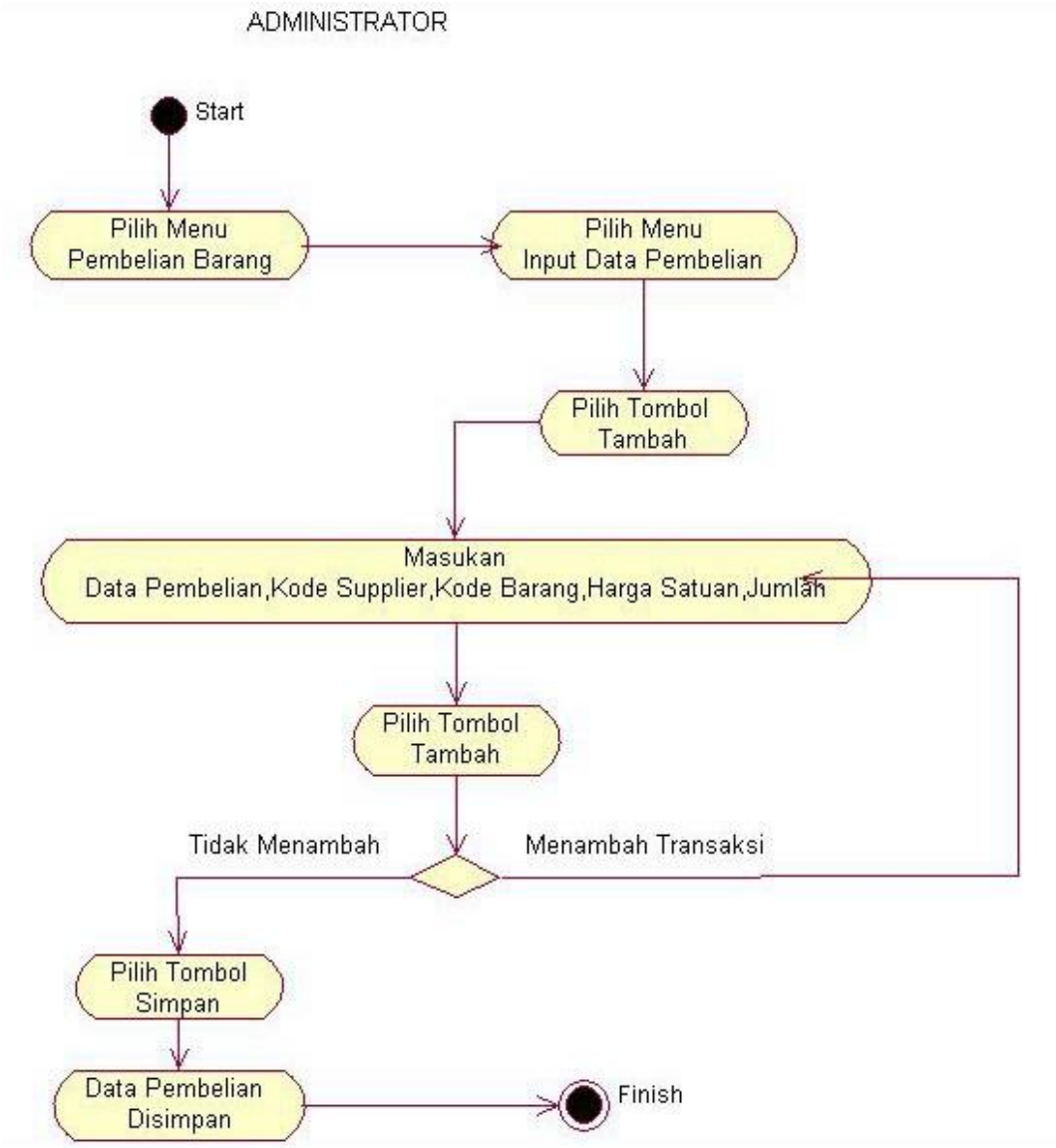
Gambar 5.13 *Activity* Diagram Ubah Barang



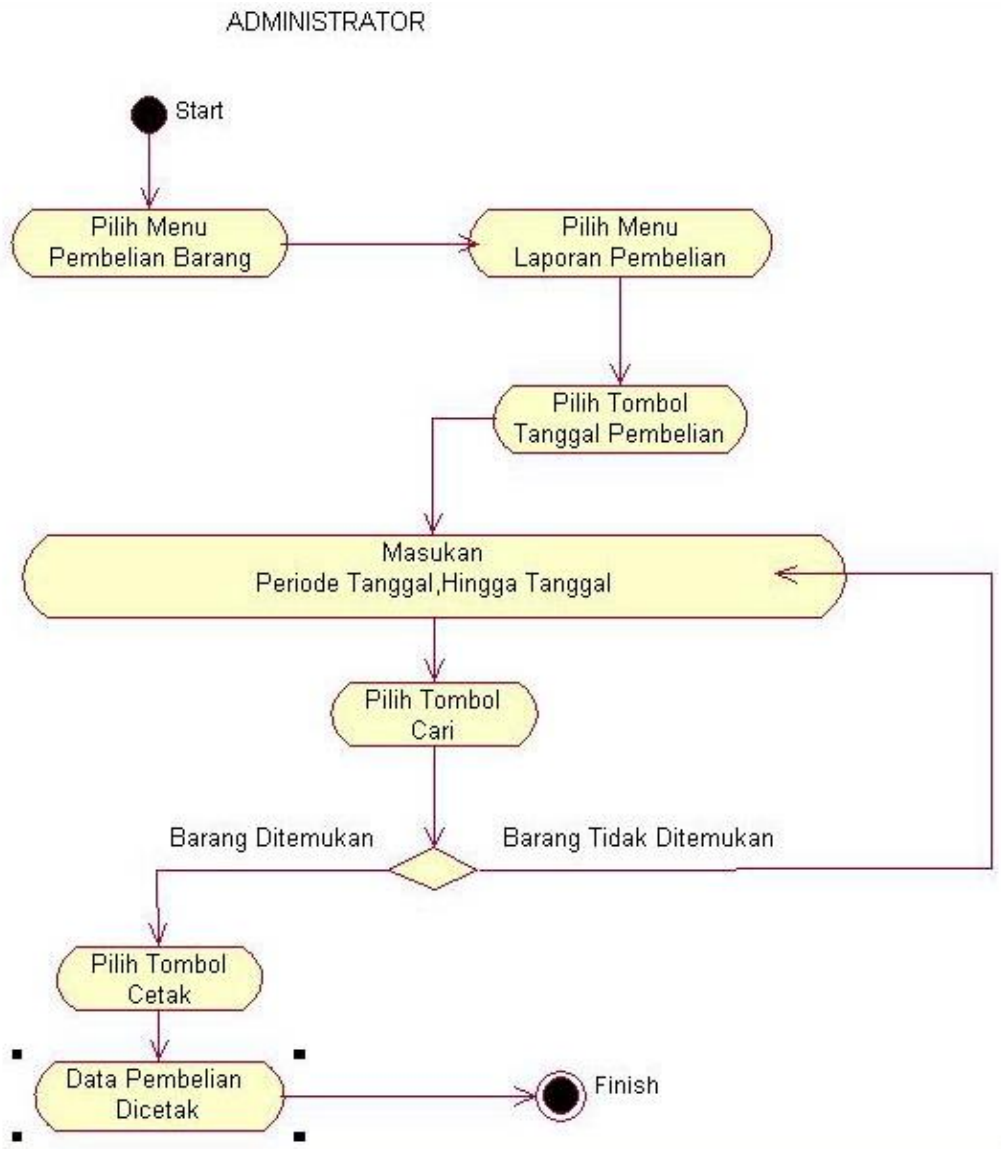
Gambar 5.14 *Activity* Diagram Cari Barang



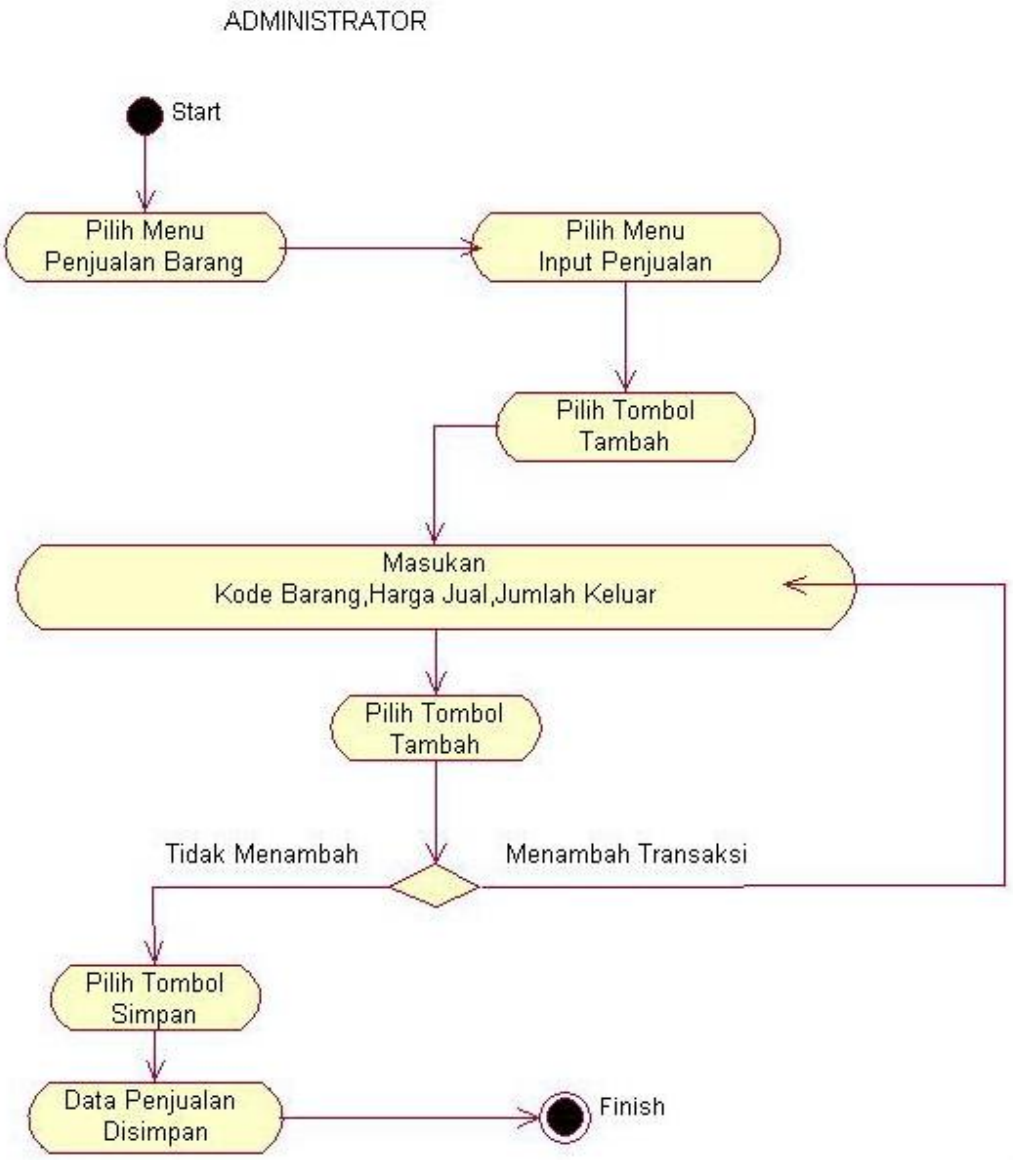
Gambar 5.15 *Activity* Diagram Cetak Laporan Barang



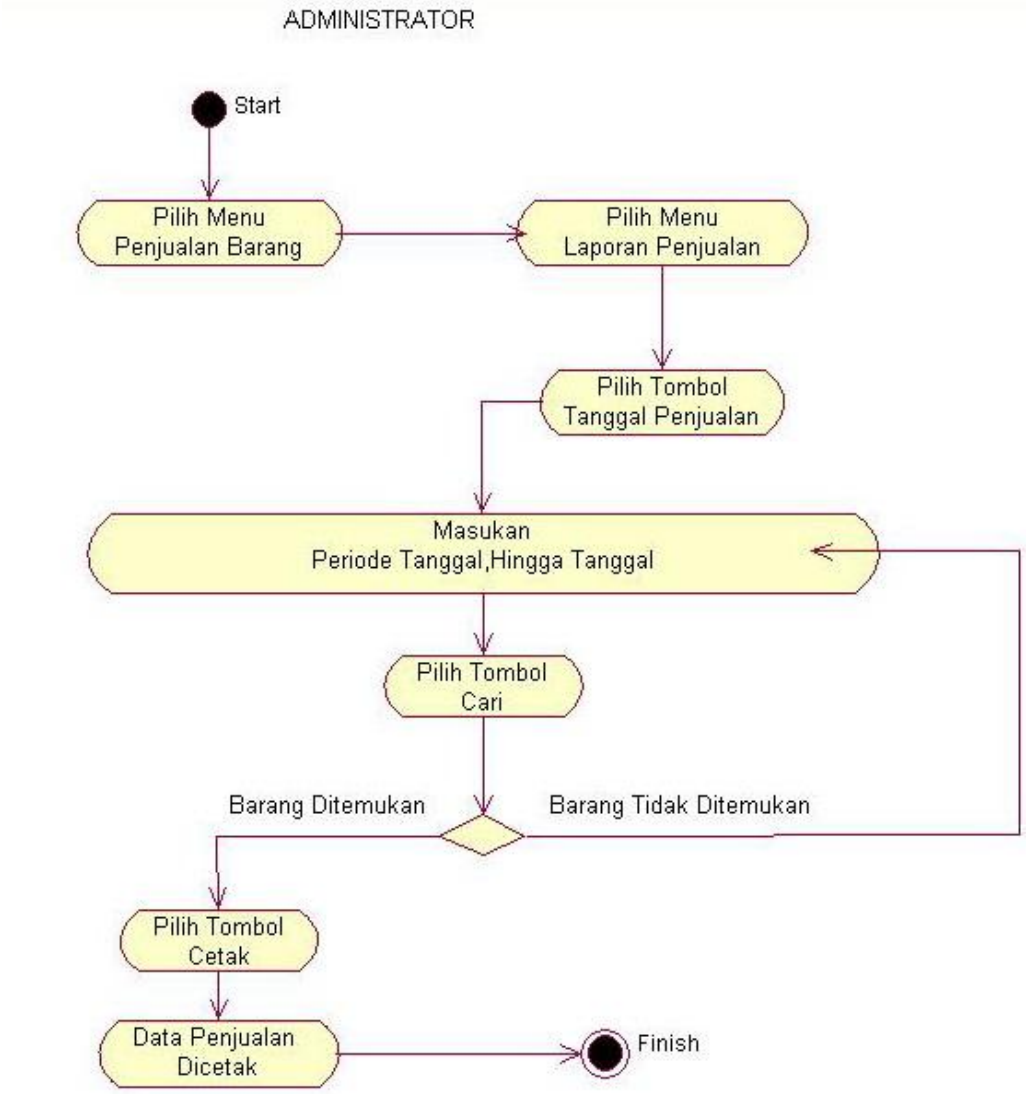
Gambar 5.16 *Activity* Diagram Pembelian



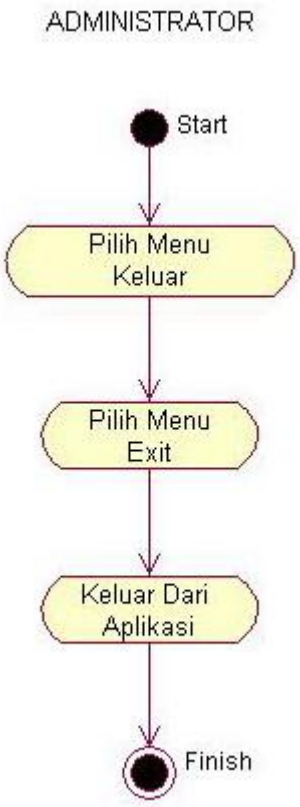
Gambar 5.17 *Activity* Diagram Laporan Pembelian



Gambar 5.18 *Activity* Diagram Penjualan



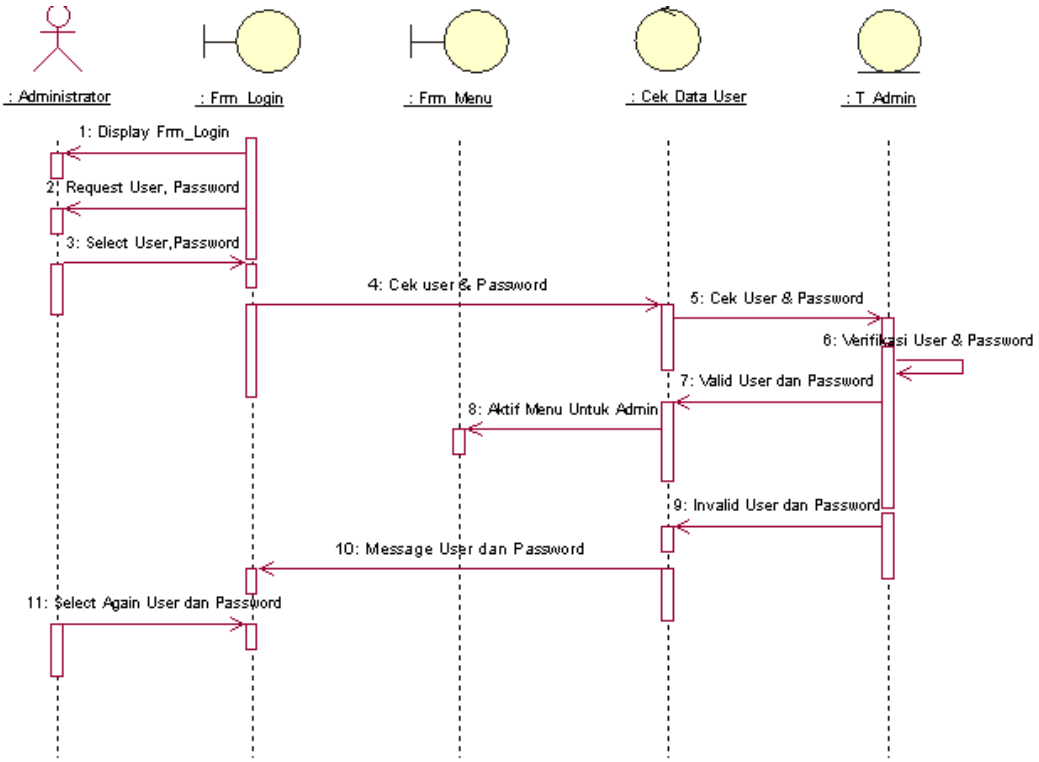
Gambar 5.19 *Activity* Diagram Laporan Penjualan



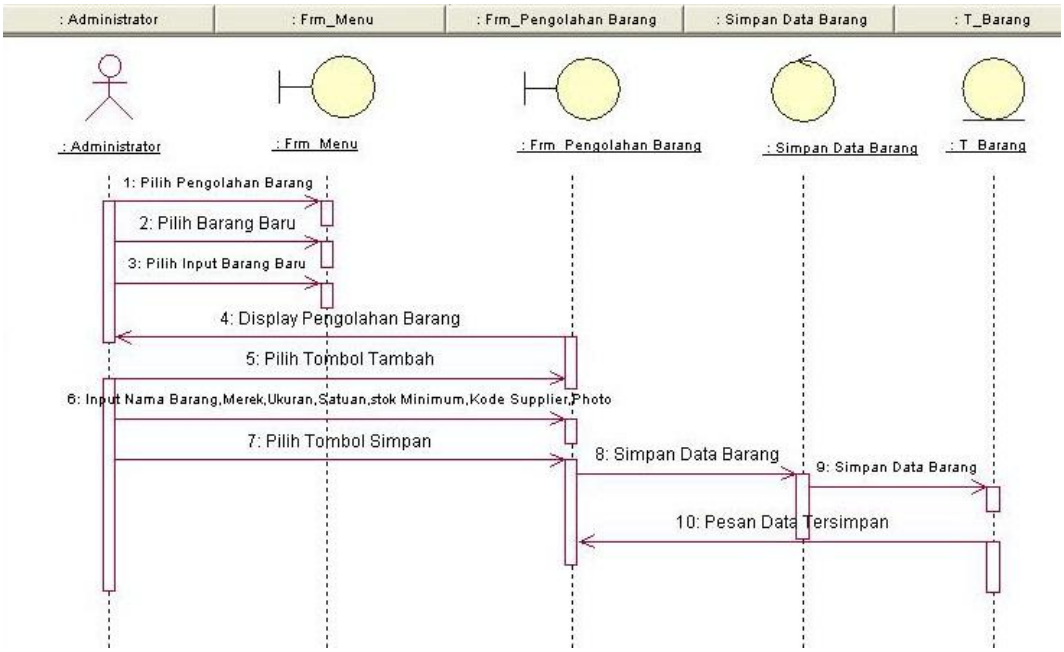
Gambar 5.20 *Activity* Diagram Keluar

4.2.10 Sequence Diagram

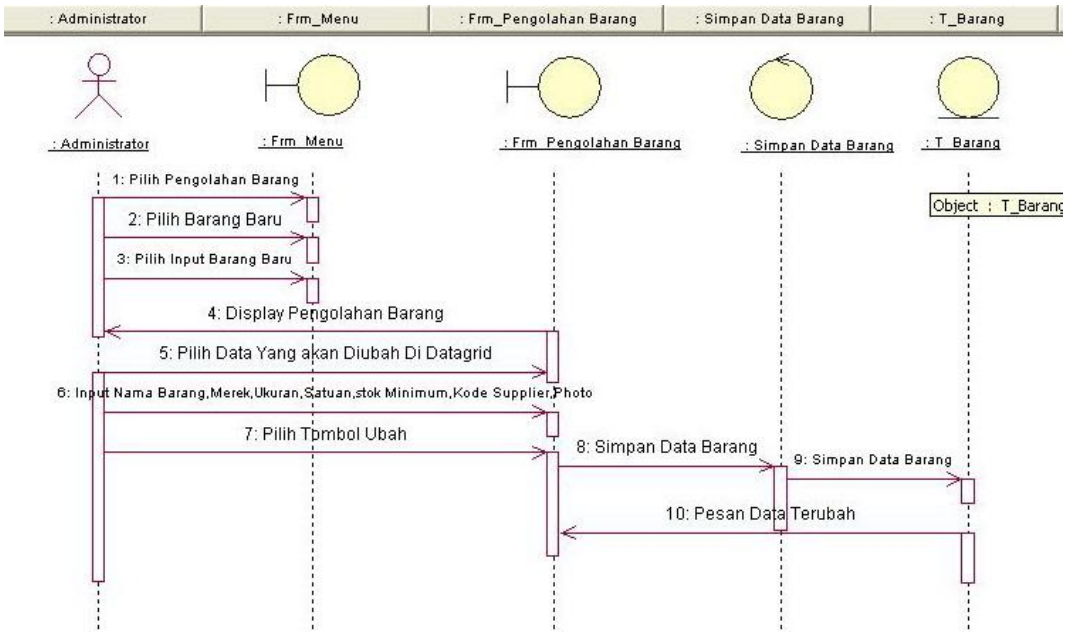
Uraian langkah masing-masing *usecase* diagram secara rinci dimodelkan menggunakan *sequence* diagram dan dapat dilihat pada gambar sebagai berikut :



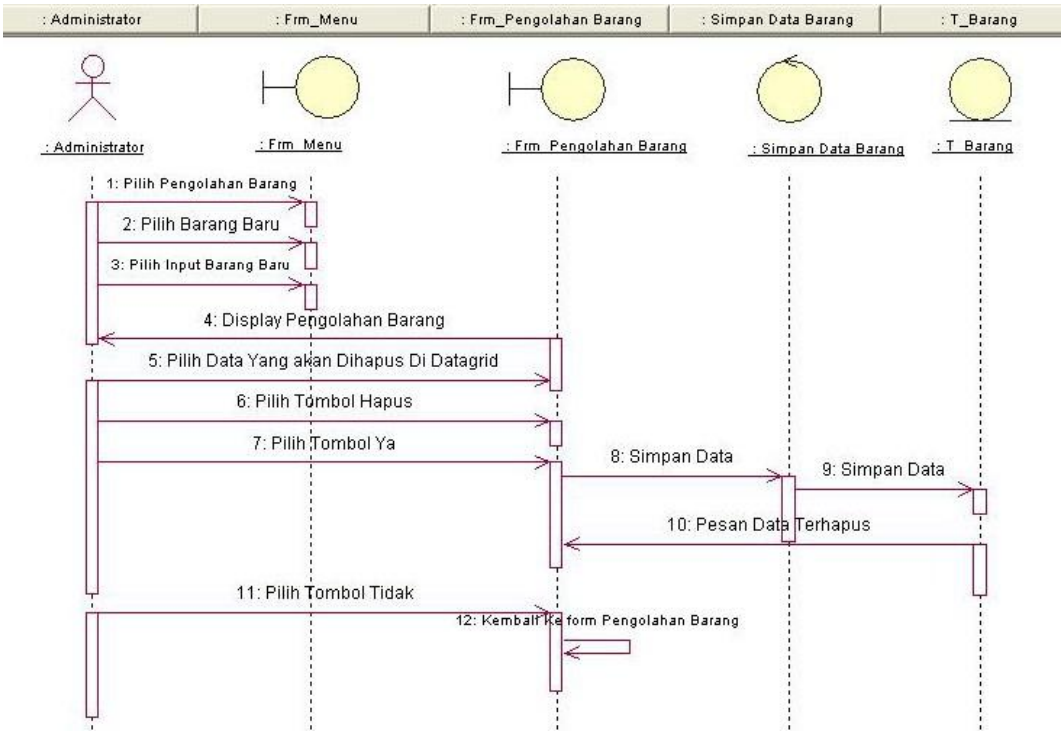
Gambar 5.21 *Sequence* Diagram Login Admin



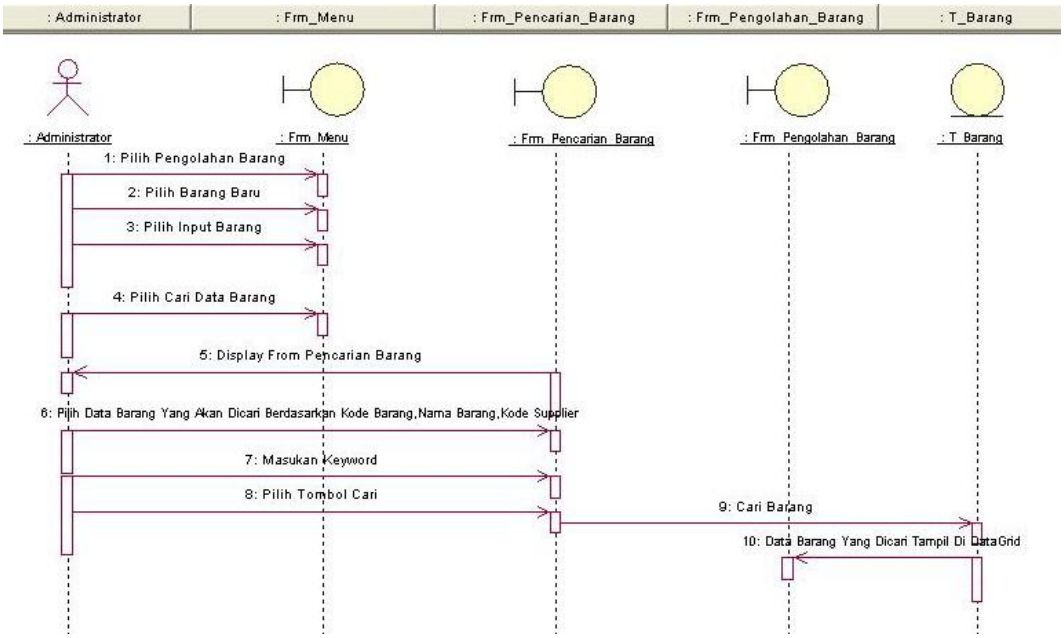
Gambar 5.22 *Sequence* Diagram Barang Baru



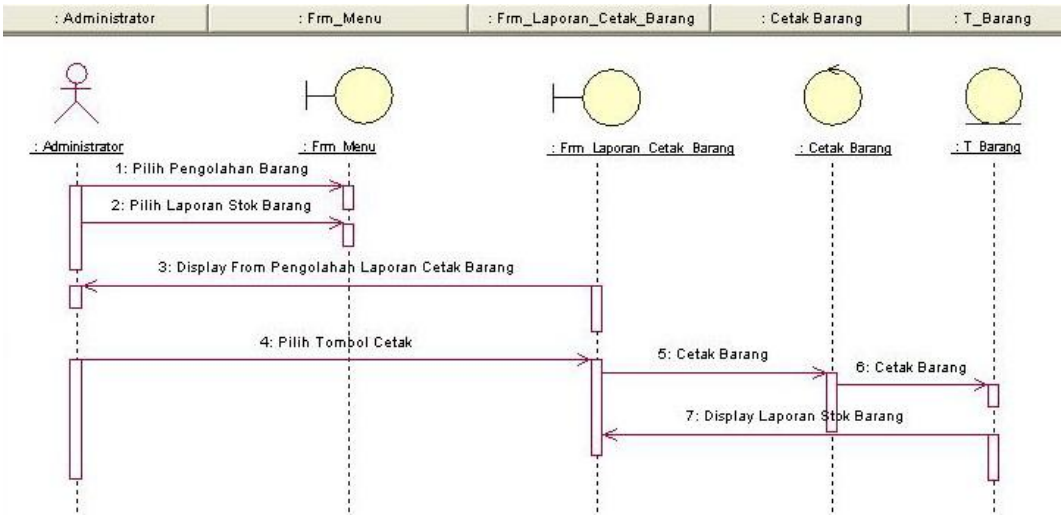
Gambar 5.23 *Sequence* Diagram Ubah Barang



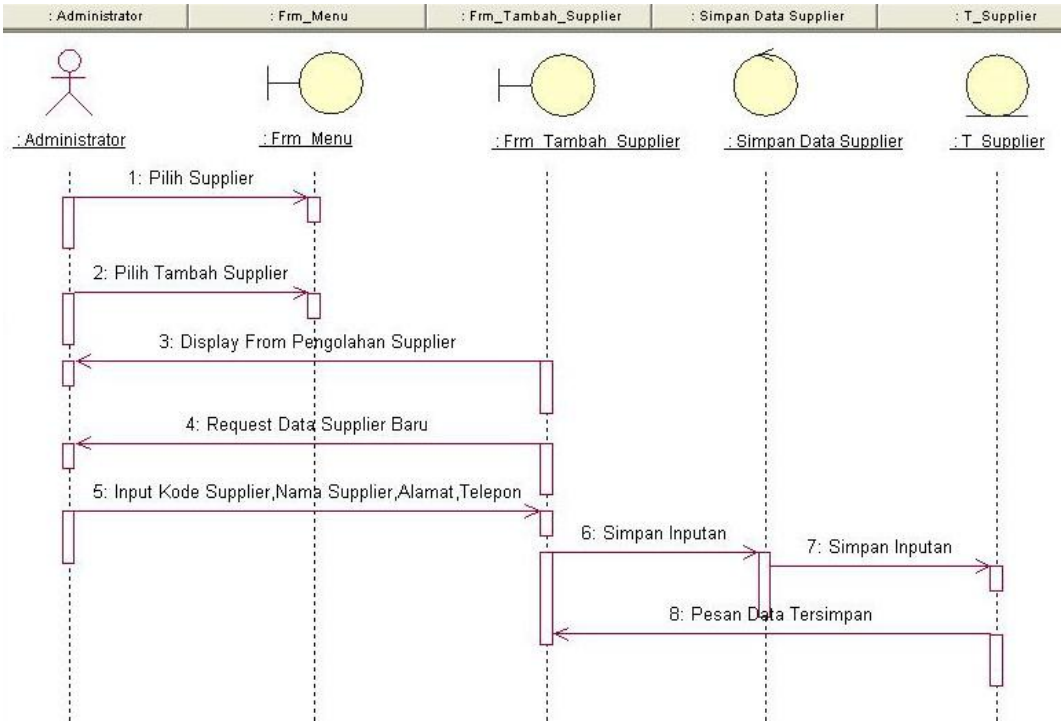
Gambar 5.24 *Sequence* Diagram Hapus Barang



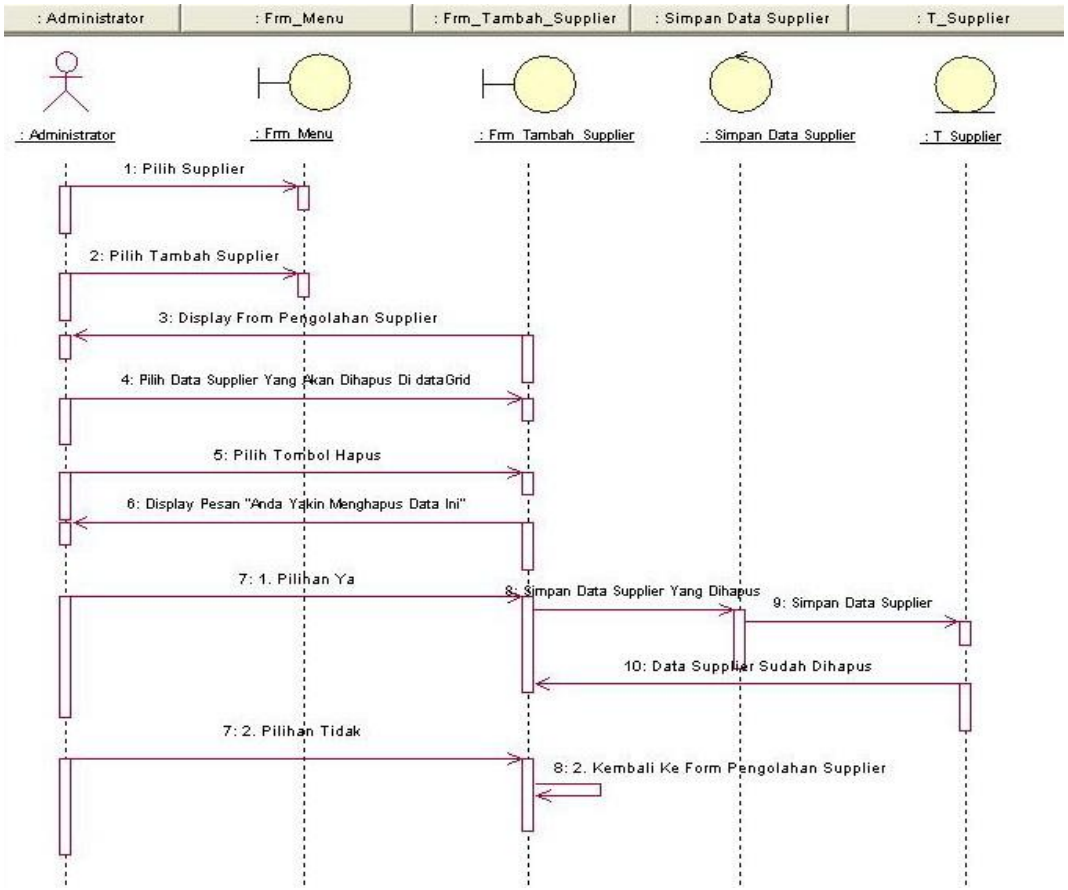
Gambar 5.25 *Sequence* Diagram Cari Barang



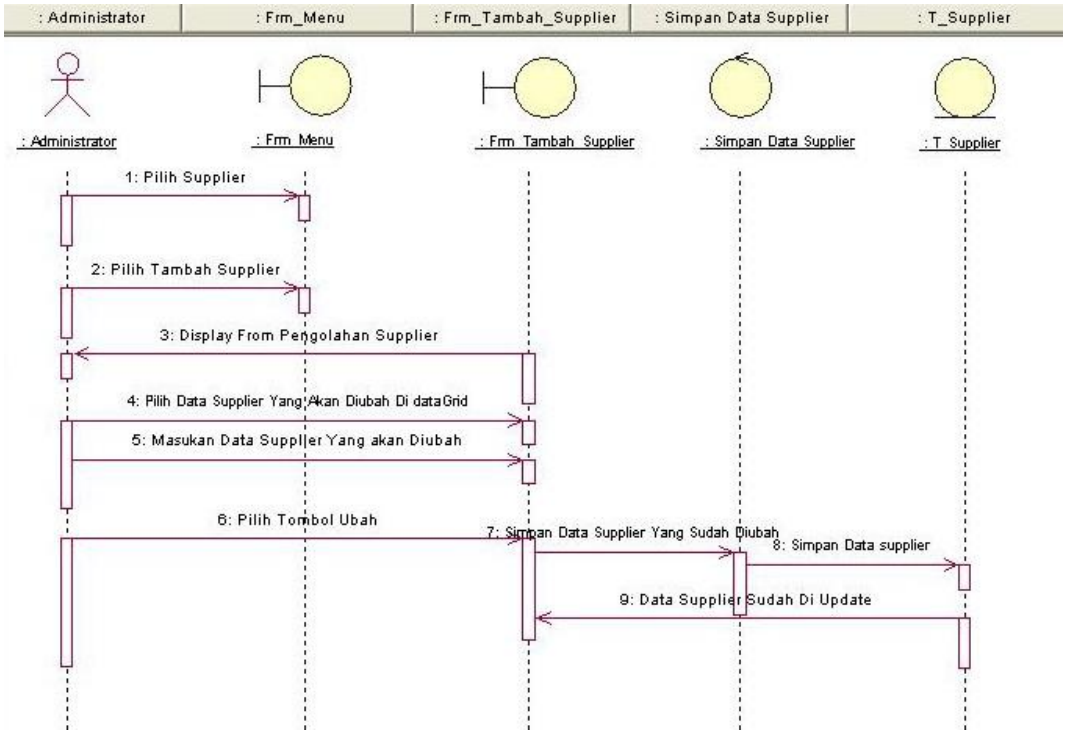
Gambar 5.26 *Sequence* Diagram Laporan Stok Barang



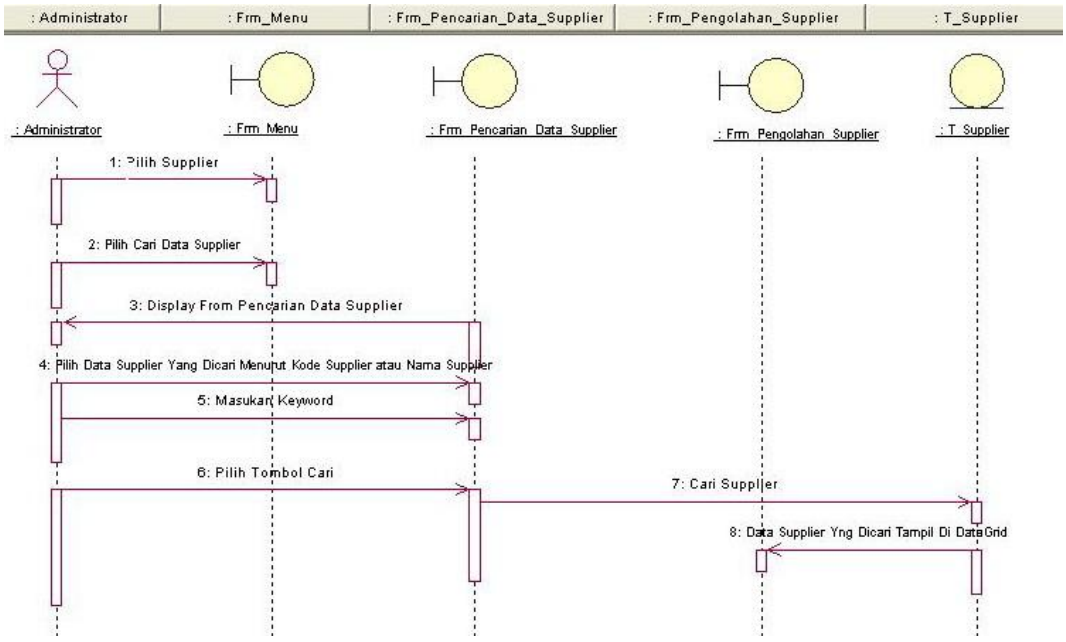
Gambar 5.27 *Sequence* Diagram Supplier Baru



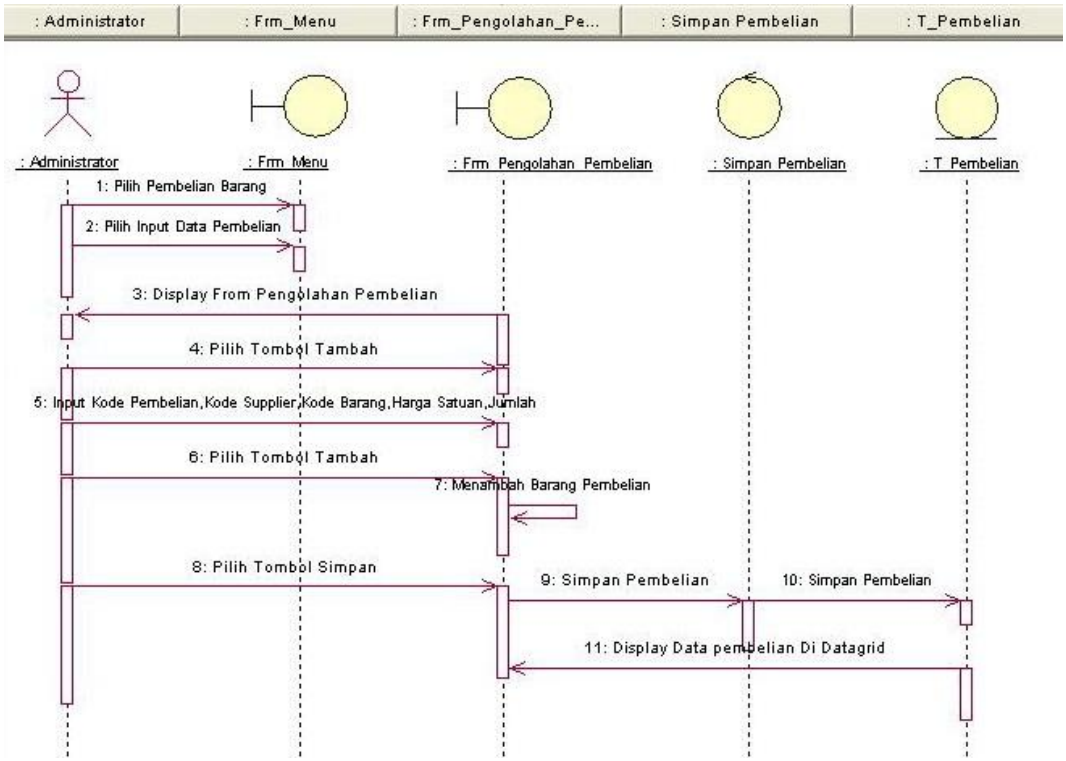
Gambar 5.28 *Sequence* Diagram Hapus Supplier



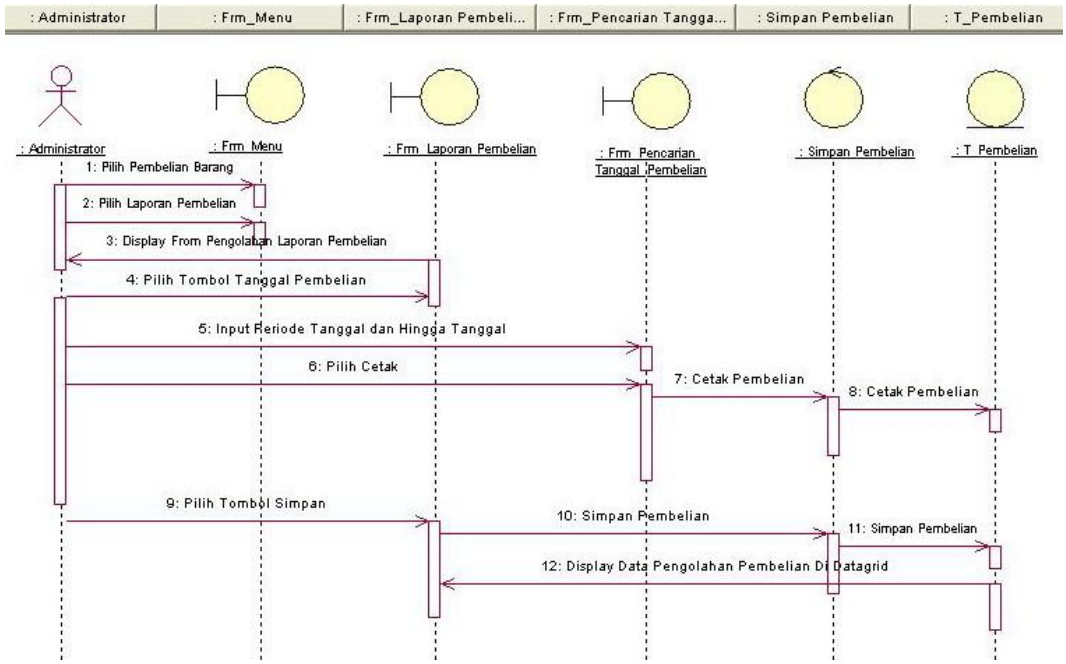
Gambar 5.29 *Sequence* Diagram Ubah Supplier



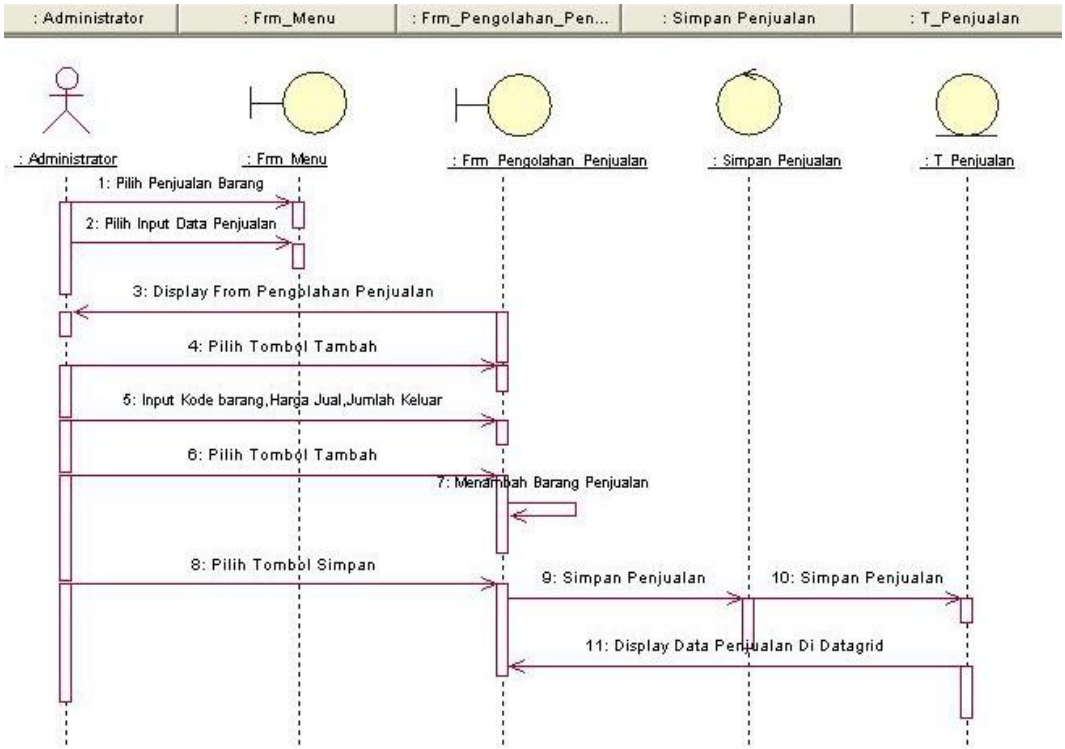
Gambar 5.30 *Sequence* Diagram Cari Supplier



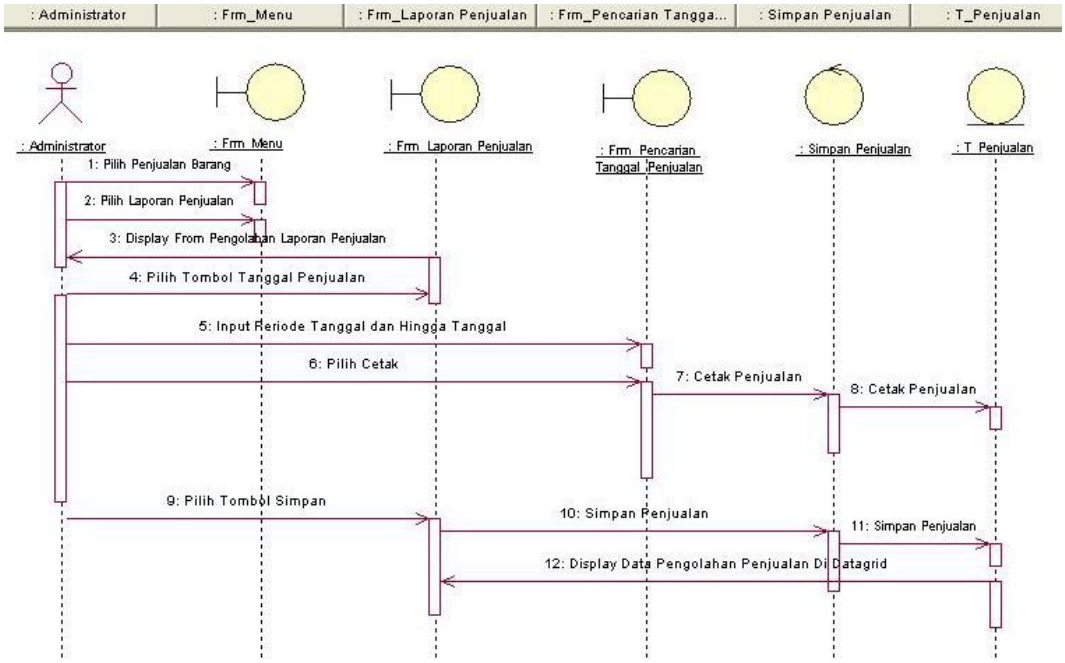
Gambar 5.31 *Sequence* Diagram Pembelian



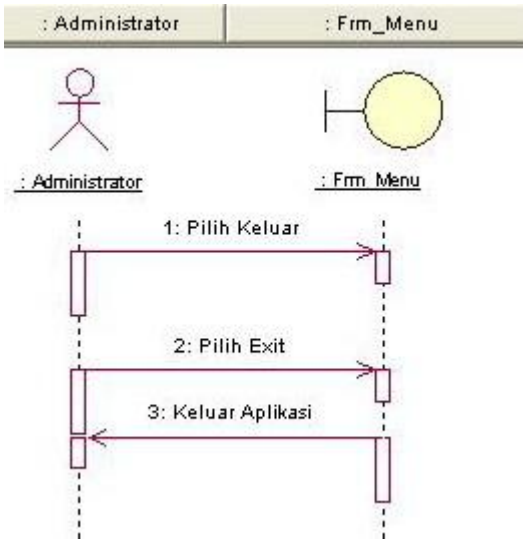
Gambar 5.32 *Sequence* Diagram Laporan Pembelian



Gambar 5.33 *Sequence* Diagram Penjualan



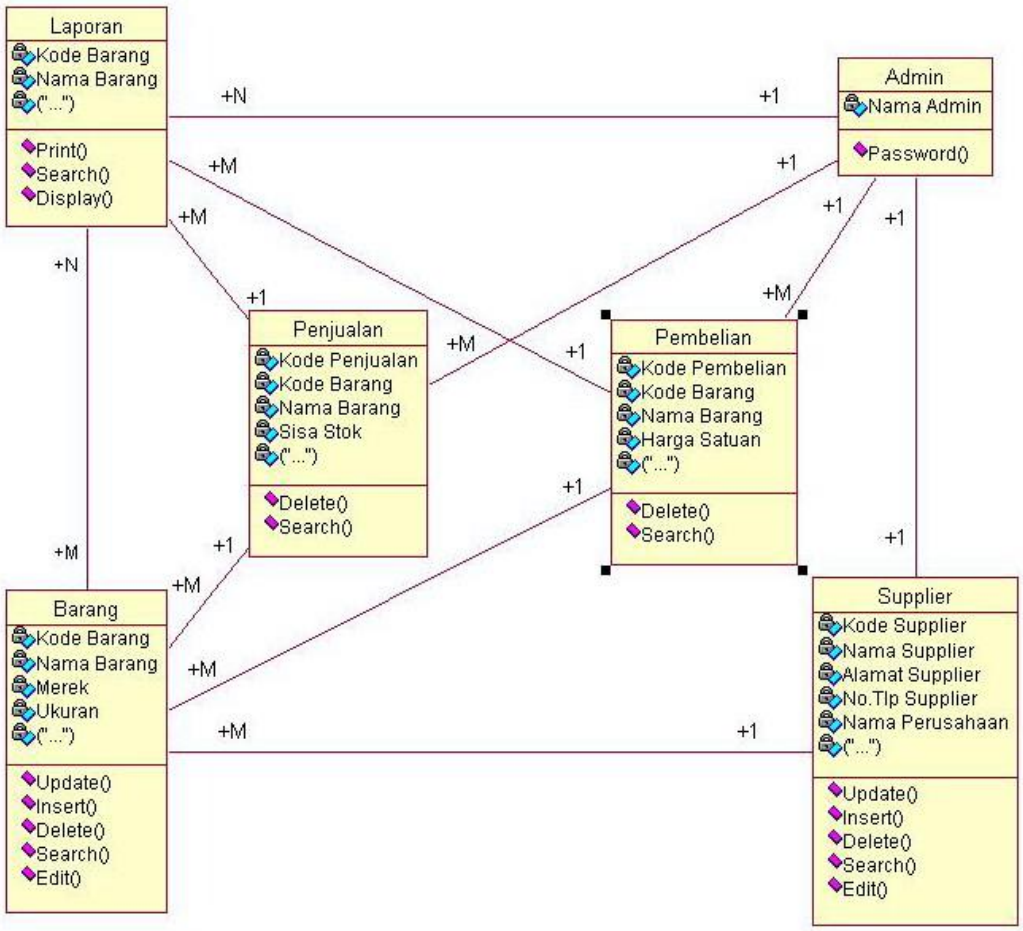
Gambar 5.34 *Sequence* Diagram Laporan_Penjualan



Gambar 5.35 *Sequence Diagram* Keluar

4.2.11 *Class* Diagram

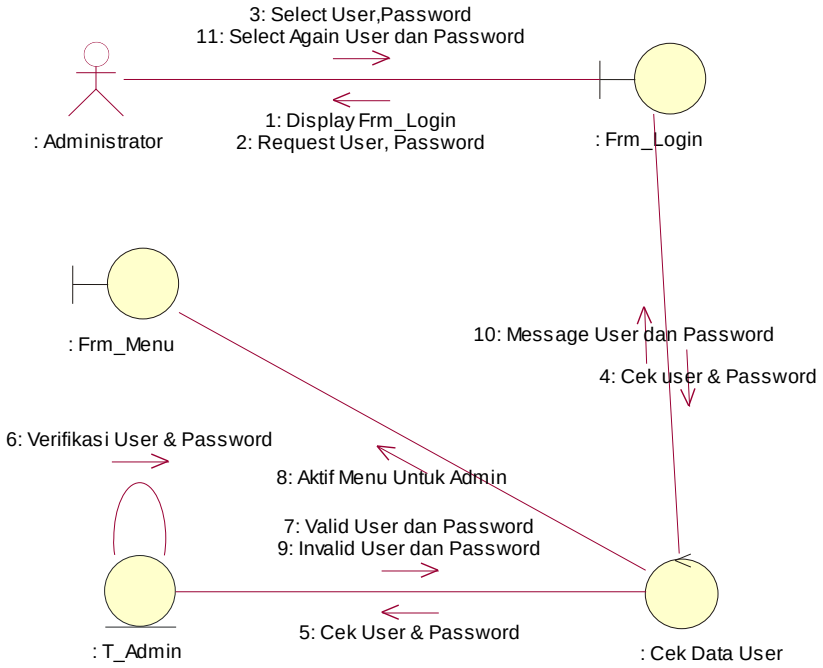
Berdasarkan *use case diagram* serta skenarionya, dirancang beberapa *class* yang digambarkan dalam *class diagram* berikut :



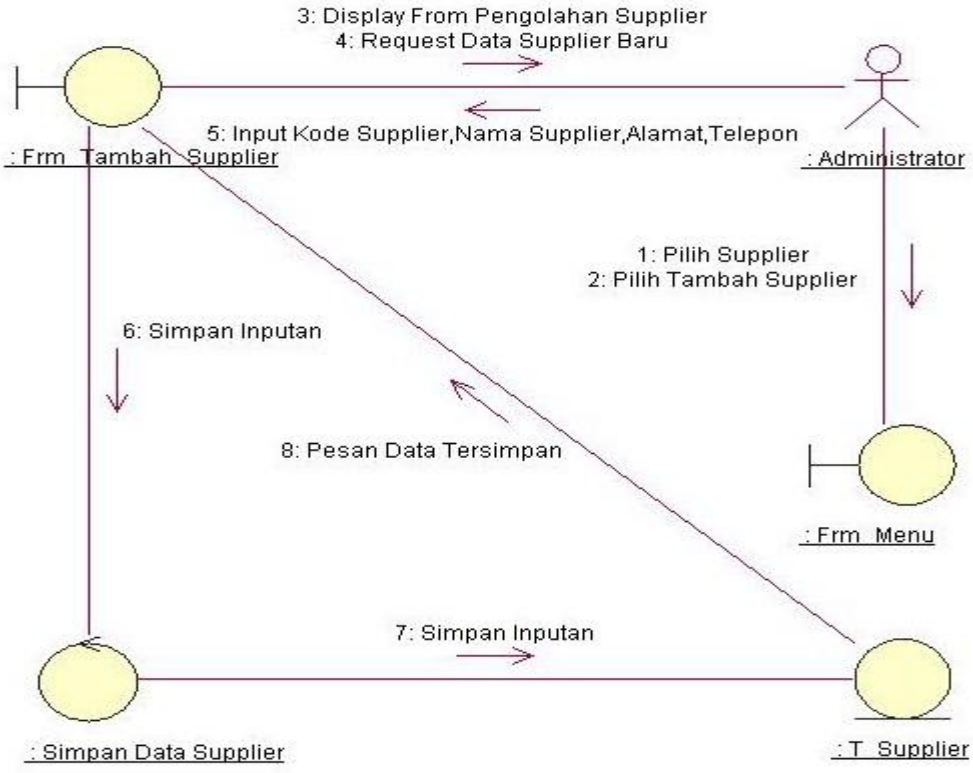
Gambar 5.36 *Class* Diagram Toko PRIMATAMA COMPUTER

4.2.12 Colaboration Diagram

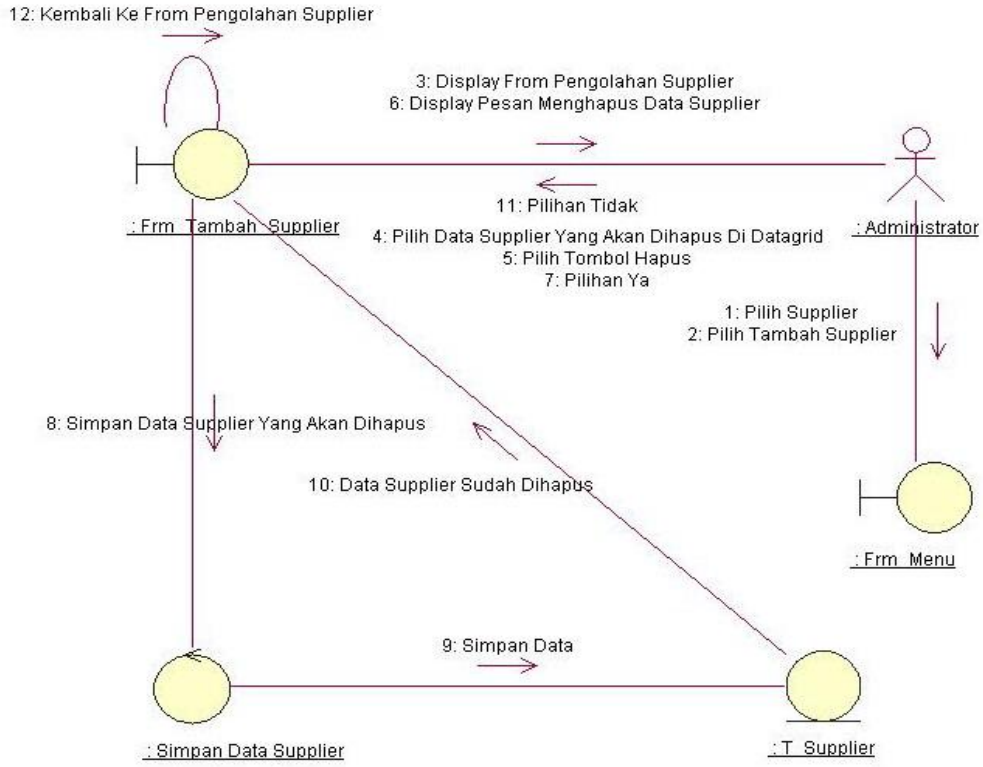
Menggambarkan kolaborasi dinamis seperti *Sequence* Diagram. Dalam menunjukkan pertukaran pesan, *Colaboration* Diagram menggambarkan *object* dan hubungannya (mengacu ke konteks).



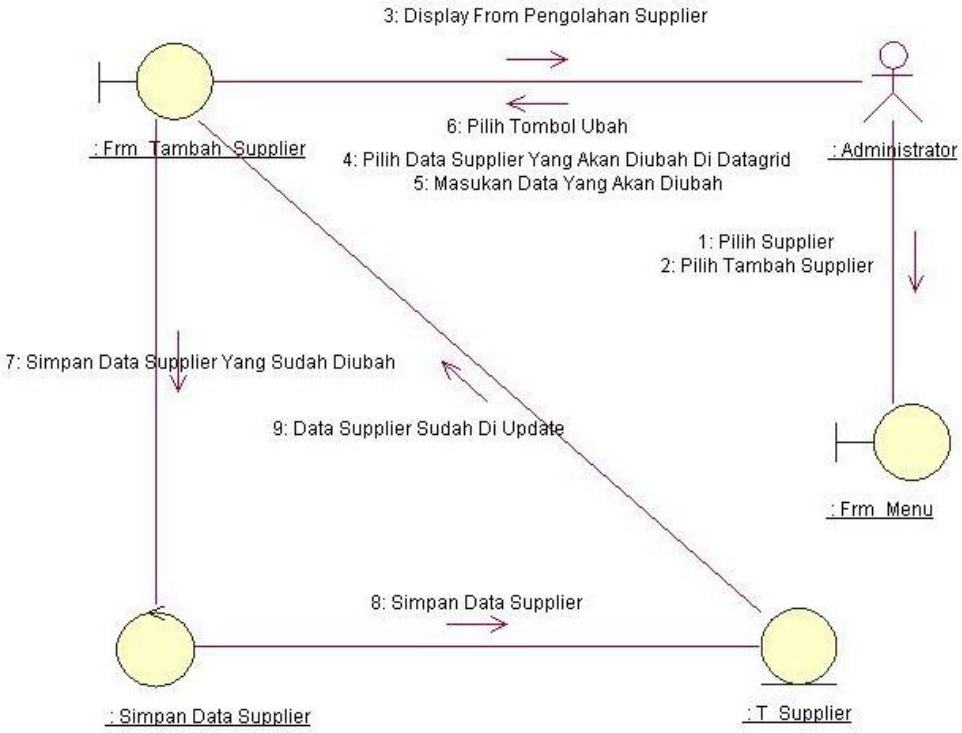
Gambar 5.37 *Colaboration* Diagram *Login* Admin



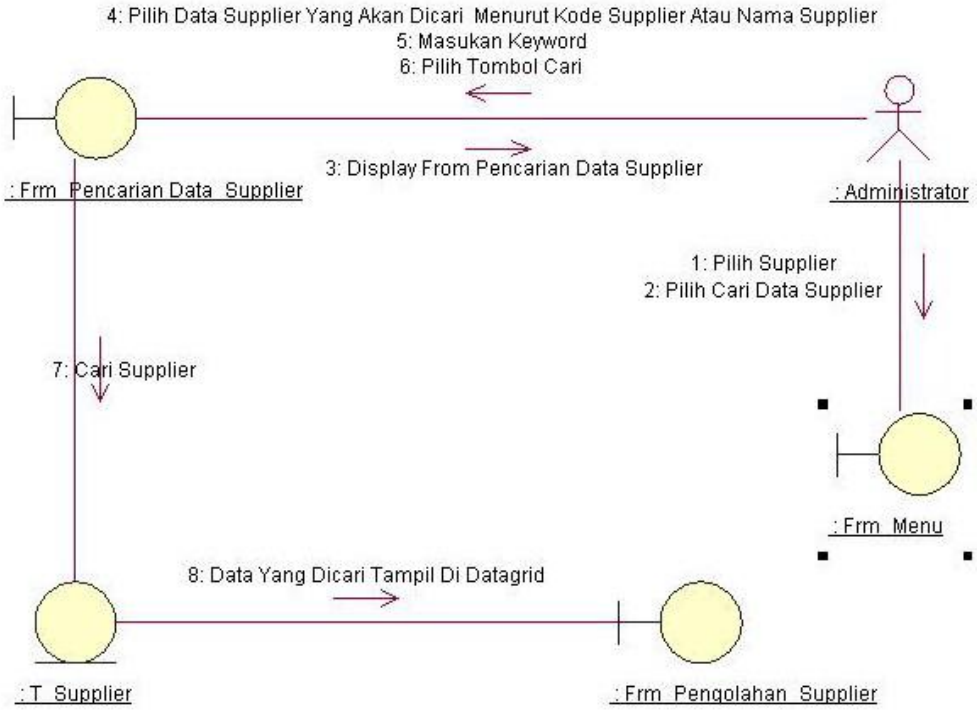
Gambar 5.38 *Colaboration* Diagram Supplier Baru



Gambar 5.39 *Colaboration* Diagram Hapus Supplier



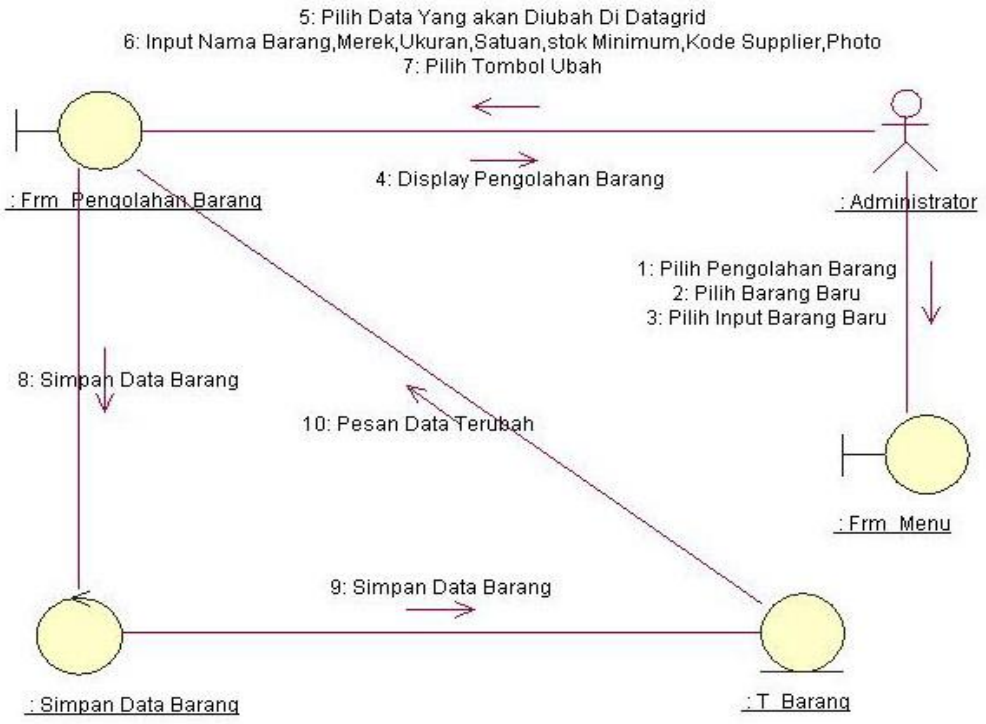
Gambar 5.40 *Colaboration* Diagram Ubah Supplier



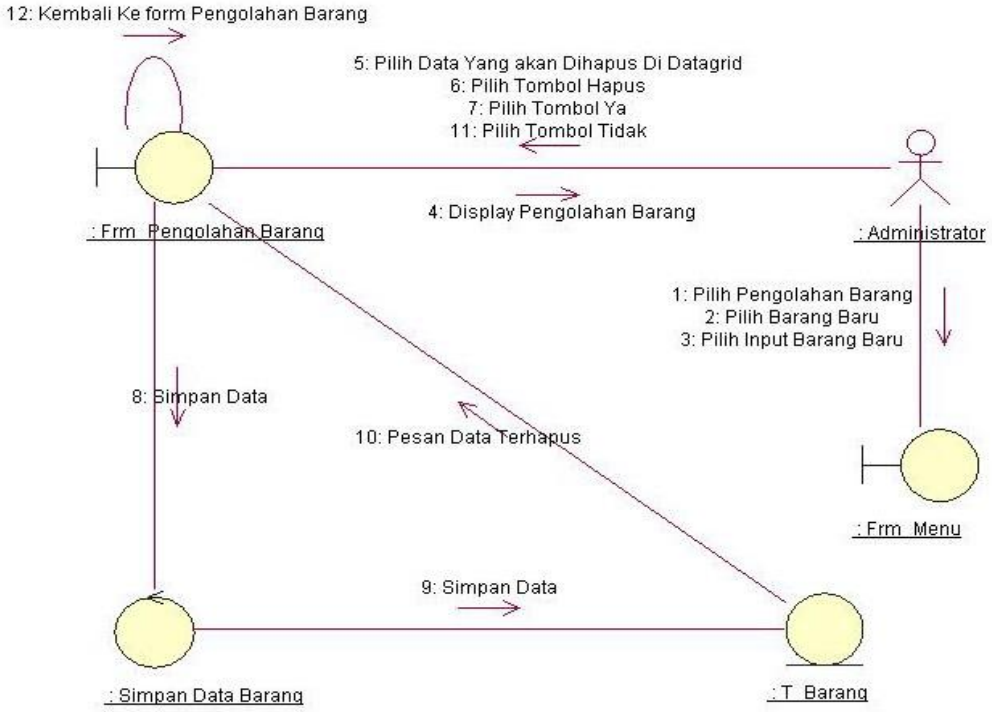
Gambar 5.41 *Colaboration* Diagram Cari Supplier



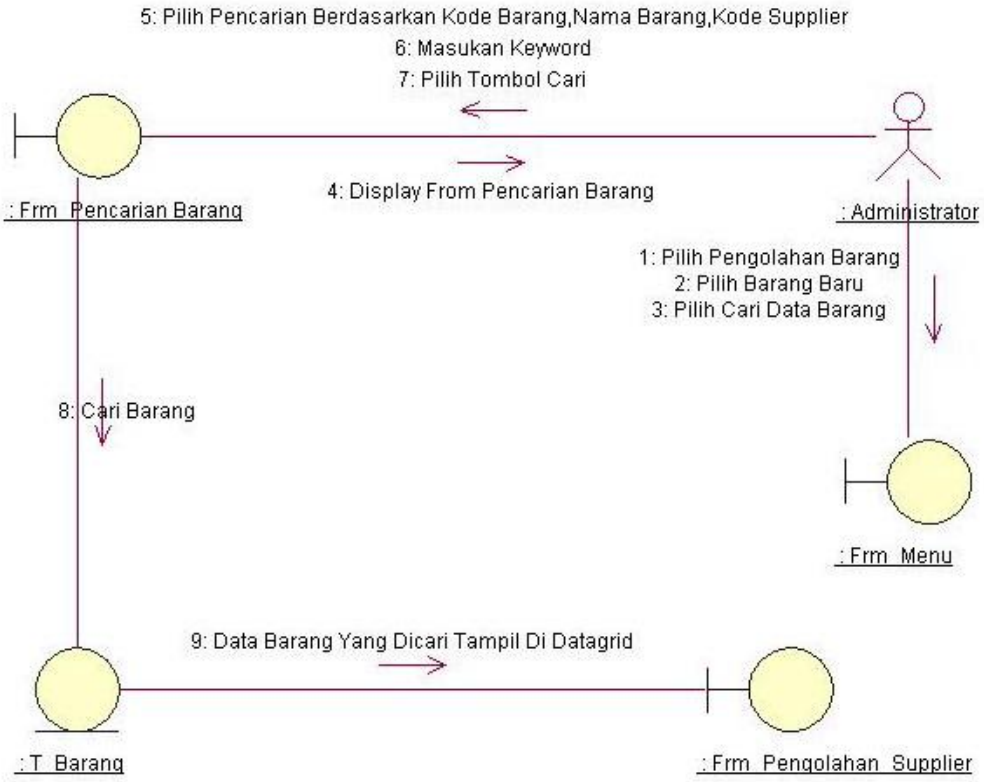
Gambar 5.42 *Colaboration* Diagram Barang Baru



Gambar 5.43 *Colaboration* Diagram Ubah Barang



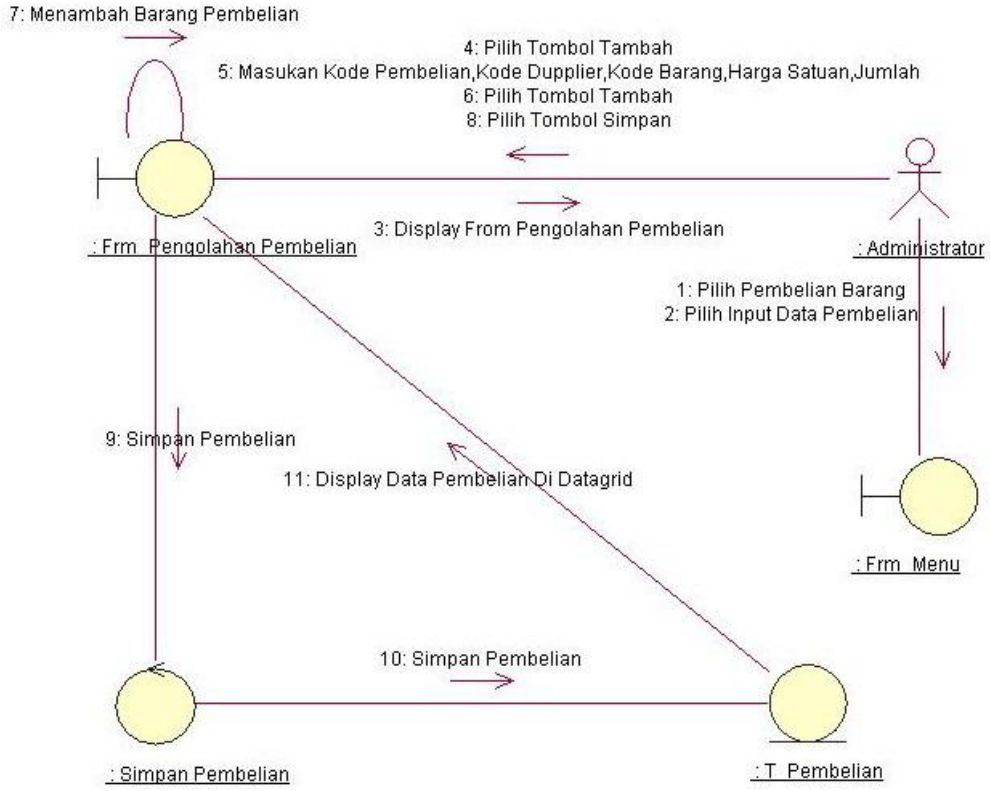
Gambar 5.44 *Colaboration* Diagram Hapus Barang



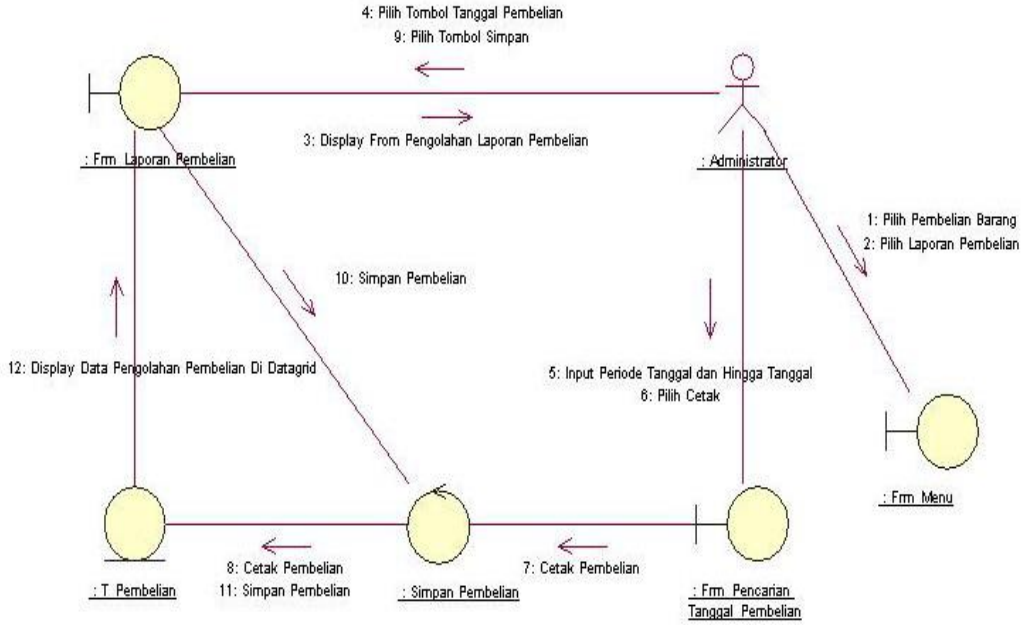
Gambar 5.45 *Colaboration* Diagram Cari Barang



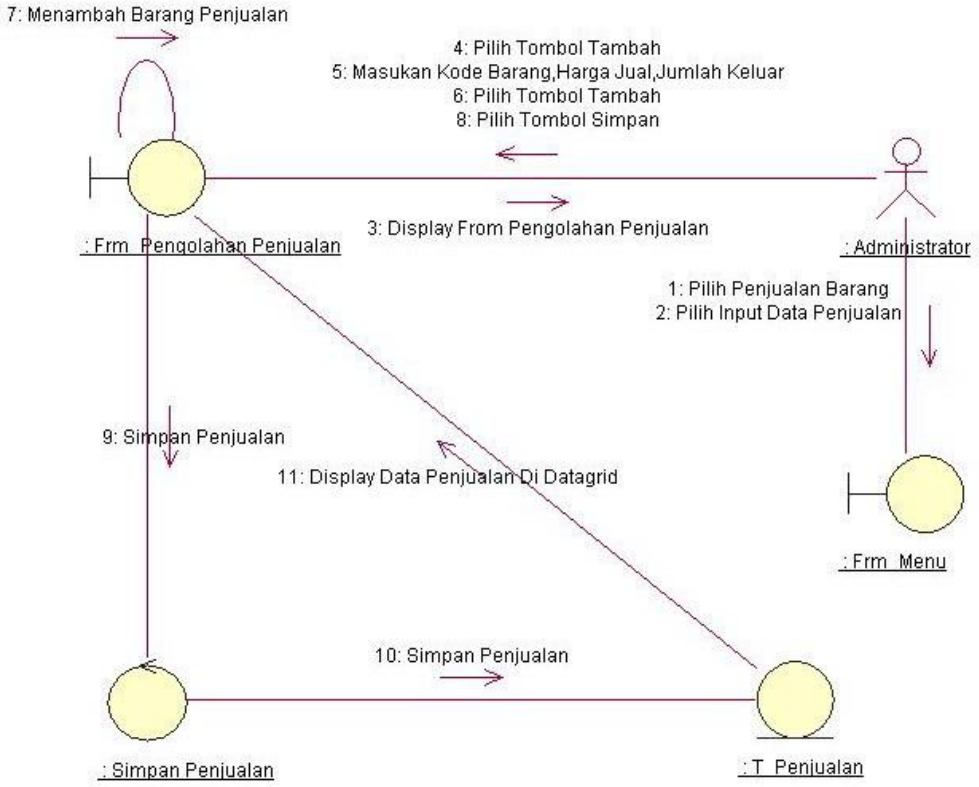
Gambar 5.46 *Colaboration* Diagram Cetak Stok Barang



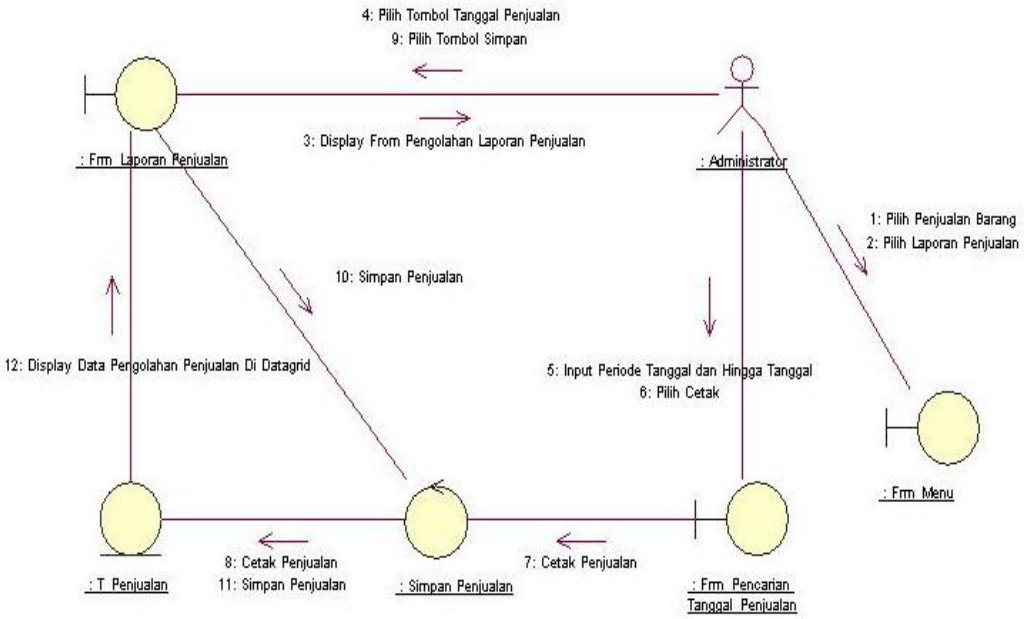
Gambar 5.47 *Colaboration* Diagram Pembelian



Gambar 5.48 *Colaboration* Laporan Pembelian



Gambar 5.49 *Colaboration* Diagram Penjualan



Gambar 5.50 *Colaboration* Diagram Laporan Penjualan



Gambar 5.51 *Colaboration* Diagram Keluar

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari pelaksanaan Kerja Praktek dan proses analisis *software* sistem informasi ini, dapat diambil kesimpulan sebagai berikut:

1. *Software* sistem pengolahan pembelian dan penjualan komputer ini adalah sistem yang membantu pelaksanaan inovasi toko.
2. *Software* sistem sistem pengolahan pembelian dan penjualan computer ini, melakukan pengolahan terhadap data pembelian dan penjualan toko agar penyimpanan arsip data tidak hilang, atau tercecer.

Software sistem informasi pengolahan pembelian dan penjualan computer ini dibangun untuk meningkatkan pelayanan kepada costumer.

5.2 Saran

Software sistem pengolahan pembelian dan penjualan computer yang baru dibuat ini merupakan suatu bentuk sistem informasi komputerisasi. *Software* sistem pengolahan pembelian dan penjualan computer ini belum merupakan sistem yang sempurna, sehingga dibutuhkan penyempurnaan sistem secara berkesinambungan sesuai dengan kebutuhan. Untuk itu disarankan:

1. Setelah diimplementasikan yang meliputi modul program, antarmuka dan basis data, diharapkan dilakukan pengujian terhadap sistem yang dibangun dan dipastikan sesuai dengan kebutuhan.
2. Setelah dilakukan pengujian, dilakukan penyempurnaan terhadap kekurangan-kekurangan sistem.

Diharapkan dilakukan pelatihan terlebih dahulu agar pengguna benar-benar memahami sistem dan cara penggunaannya sekaligus pemeliharaannya