BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Perusahaan

Tinjauan perusahaan dimaksudkan untuk mengetahui hal hal yang berkaitan dengan perusahaan, antara lain sejarah berdirinya perusahaan, struktur organisasi perusahaan, serta visi misi perusahaan.

2.1.1 Sejarah PLN

Berawal di akhir abad ke 19, perkembangan ketenagalistrikan di Indonesia mulai ditingkatkan saat beberapa perusahaan asal Belanda yang bergerak di bidang pabrik gula dan pabrik teh mendirikan pembangkit listrik untuk keperluan sendiri.

Antara tahun 1942-1945 terjadi peralihan pengelolaan perusahaan- perusahaan Belanda tersebut oleh Jepang, setelah Belanda menyerah kepada pasukan tentara Jepang di awal Perang Dunia II.

Proses peralihan kekuasaan kembali terjadi di akhir Perang Dunia II pada Agustus 1945, saat Jepang menyerah kepada Sekutu. Kesempatan ini dimanfaatkan oleh para pemuda dan buruh listrik melalui delegasi Buruh/Pegawai Listrik dan Gas yang bersama-sama dengan Pimpinan KNI Pusat berinisiatif menghadap Presiden Soekarno untuk menyerahkan perusahaan-perusahaan tersebut kepada Pemerintah Republik Indonesia. Pada 27 Oktober 1945, Presiden Soekarno membentuk Jawatan Listrik dan Gas di bawah Departemen Pekerjaan Umum dan Tenaga dengan kapasitas pembangkit tenaga listrik sebesar 157,5 MW.

Pada tanggal 1 Januari 1961, Jawatan Listrik dan Gas diubah menjadi BPU-PLN (Badan Pimpinan Umum Perusahaan Listrik Negara) yang bergerak di bidang listrik, gas dan kokas yang dibubarkan pada tanggal 1 Januari 1965. Pada saat yang sama, 2 (dua) perusahaan negara yaitu Perusahaan Listrik Negara (PLN) sebagai pengelola tenaga listrik milik negara dan Perusahaan Gas Negara (PGN) sebagai pengelola gas diresmikan.

Pada tahun 1972, sesuai dengan Peraturan Pemerintah No.17, status Perusahaan Listrik Negara (PLN) ditetapkan sebagai Perusahaan Umum Listrik Negara dan sebagai Pemegang Kuasa Usaha Ketenagalistrikan (PKUK) dengan tugas menyediakan tenaga listrik bagi kepentingan umum.

Maka sejak tahun 1994 status PLN beralih dari Perusahaan Umum menjadi Perusahaan Perseroan (Persero) dan juga sebagai PKUK dalam menyediakan listrik bagi kepentingan umum hingga sekarang.

2.1.2 Visi, Misi dan Moto

Visi

Diakui sebagai Perusahaan Kelas Dunia yang Bertumbuh kembang, Unggul dan Terpercaya dengan bertumpu pada Potensi Insani.

Misi

1. Menjalankan bisnis kelistrikan dan bidang lain yang terkait, berorientasi pada kepuasan pelanggan, anggota perusahaan dan pemegang saham.
2. Menjadikan tenaga listrik sebagai media untuk meningkatkan kualitas kehidupan masyarakat.

3. Mengupayakan agar tenaga listrik menjadi pendorong kegiatan ekonomi.
4. Menjalankan kegiatan usaha yang berwawasan lingkungan.

Moto

Listrik untuk Kehidupan yang Lebih Baik

2.1.3 Struktur Organisasi

Pengenalan struktur organisasi merupakan hal yang penting, karena dengan cara pengenalan tersebut dapat terlihat dengan mudah gambaran tentang seberapa luas jaringan operasi secara menyeluruh dari organisasi mengenai tugas, batasan serta wewenang dan tanggung jawab dari masing masing karyawan yang berada di dalam organisasi tersebut.

Berikut ini adalah struktur organisasi dan uraian tugas di PT PLN (Persero) Distribusi Jawa Barat dan Banten :

1. General Manager

   Bertugas memimpin dan mengelola distribusi sesuai dengan maksud dan tujuan distribusi dengan senantiasa berusaha meningkatkan daya guna dan hasilnya dari distribusi.

2. Manajer Bidang

   Manajer bidang terdiri dari :

   1. Bidang Perencanaan

      Uraian kewajiban dan tanggung jawab pokok :

a. Menyusun Rencana Umum Pengembangan Tenaga Listrik (RUPTL). Rencana Jangka Panjang Perusahaan (RJP), dan Rencana Kerja Anggaran Perusahaan (RKAP)
b. Menyusun Rencana dan Pengembangan sistem ketenagalistrikan
c. Menyusun sistem manajemen kinerja dan unit-unit kerja
d. Menyusun metode evolusi kelayakan investasi dan melakukan penilaian finansialnya
e. Mengembangkan hubungan kerja sama dengan pihak lain dan penyandang dana baik secara bilateral maupun multilateral
f. Menyusun rencana pengembangan aplikasi sistem informasi
g. Mengendalikan aplikasi-aplikasi teknologi informasi
h. Menyusun laporan manajemen
i. Menyiapkan SOP pengelolaan aplikasi sistem informasi
j. Menyusun rencana pengembangan usaha baru serta penerapan perusahaan

2. Bidang Distribusi

Uraian kewajiban dan tanggung jawab pokok :

a. Menyusun rencana pengembangan sistem jaringan distribusi dan membina penerapanya
b. Menyusun strategi pengoperasian dan pemeliharaan jaringan distribusi dan membina penerapanya
c. Menyusun SOP untuk penerapan dan pengujian peraturan distribusi serta SOP untuk operasi dan pemeliharaan jaringan distribusi

d. Menyusun desain standar konstruksi jaringan distribusi dan peralatan kerjanya serta membina penerapanya
e. Mengevaluasi susut energi listrik dan gangguan pada saran pendistribusian tenaga listrik serta sarana perbaikanya
f. Menyusun pengembangan kegiatan konstruksi dan administrasi pekerjaan serta membina penerapanya
g. Menyusun pengembangan serta komunikasi dan otomatisasi operasi jaringan distribusi
h. Menyusun kebijakan manajemen jaringan distribusi serta membuna penerapanya
i. Menyusun regulasi untuk penyempurnaan serta induk jaringan (DU)
j. Memantau dan mengevaluasi serta induk jaringan

3. Bidang Niaga

Uraian kewajiban dan tanggung jawab pokok :

a. Menyusun ketentuan dan strategi perusahaan
b. Menyusun rencana penjualan energi dan rencana pendapatan
c. Mengevaluasi harga jual energi listrik
d. Menghitung biaya penyediaan listrik
e. Menyusun strategi dan pengembangan pelayanan pelanggan
f. Menyusun standar dan produk pelayanan
g. Menyusun ketentuan Data Induk Pelanggan (DIP) dan Data Induk Skala (DIS) serta kontrak jual beli tenaga listrik

h. Mengkaji pengelolaan pencatatan meter dan menyusun penyempurnaanya
i. Mengkoordinasikan pelaksanaan dan penagihan kepada pelanggan tertentu, antara lain TNI/POLRI dan internal vertikal
j. Melaksanakan pengendalian DIS dan *opname* saldo piutan
k. Menyusun konsep kebijakan sistem informasi pelayanan pelanggan
l. Menyusun interaksi antar urut pelaksanaan
m. Menyusun rencana pengembangan usaha baru serta peraturanya
n. Menyusun laporan manajemen dibidangnya

4. Bidang Keuangan

   Uraian kewajiban dan tanggung jawab pokok :

   a. Mengendalikan aliran kas pendapatan dan membuat laporan rekonsiliasi keuangan
   b. Mengendalikan anggaran dan evaluasi laporan keuangan unit-unit serta rencana aliran kas pembiayaan
   c. Melakukan analisis dan evaluasi laporan keuangan unit-unit menyusun laporan keuangan konsolidasi
   d. Menyusun dan menganalisa kebijakan risiko dan penghapusan aset
   e. Melakukan pengelolaan keuangan
   f. Menyusun laporan manajemen dibidangnya

5. Bidang SDM dan Organisasi

   Uraian kewajiban dan tanggung jawab pokok :

a. Menyusun kebijakan pengembangan organisasi dan mengelola pelaksanaanya
b. Menyusun kebijakan manajemen SDM dan mengelola pelaksanaanya
c. Menyusun kebijakan pengembangan organisasi dan pengembangan SDM
d. Menyusun laporan manajemen dibidangnya

6. Bidang Komunikasi, Hukum dan Administrasi

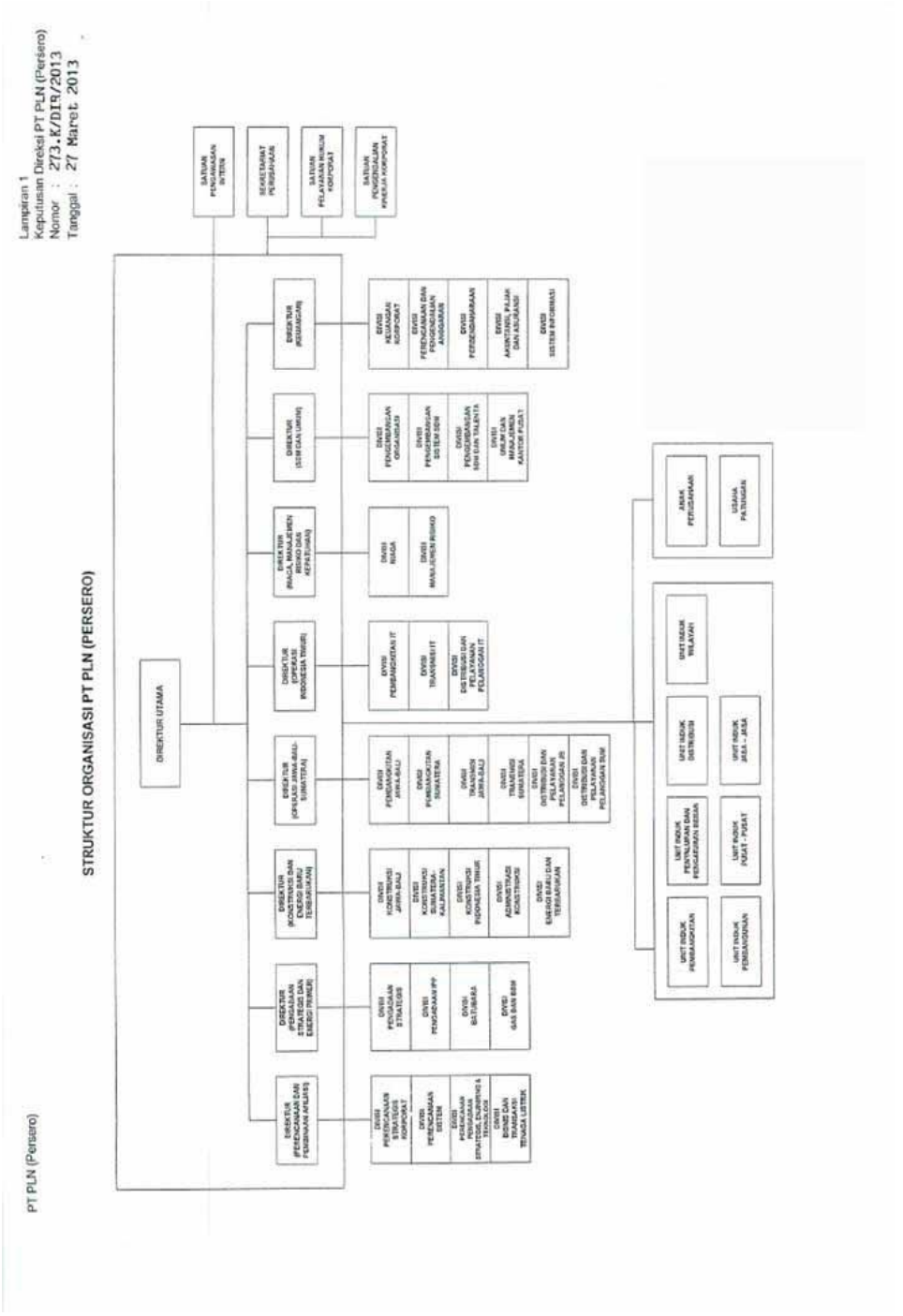
   Uraian kewajiban dan tanggung jawab pokok :

   a. Menyusun kebijakan mengelola komunikasi kemasyarakatan dan pelanggan baik internal maupun eksternal
   b. Menyusun kebijakan K3, linggungan dan *community development*
   c. Menyusun kebijakan dan mengelola fasilitas kerja, sistem pengamanan dan manajemen kantor
   d. Menyusun kebijakan administrasi
   e. Menyusun dan mengkaji produk-produk hukum dan peraturan-peraturan perusahaan
   f. Memberikan advokasi dalam bisnis energi listrik dan ketenagalistrikan
   g. Menyusun standar fasilitas kantor
   h. Mengelola aset tanah dan bangunan serta sarana kerja
   i. Mengelola kesekretan rumah tangga kantor induk
   j. Menyusun laporan manajemen dan bidangnya

7. Audit Internal

Uraian kewajiban dan tanggung jawab pokok :

a. Menyusun program kerja pemeriksaan tahunan, sesuai program kerja perusahaan
b. Melaksanakan Audit Internal, meliputi keuangan, teknik, manajemen dan sumber daya manusia
c. Memberikan masukan dan rekomendasi yang menyangkut proses manajemen operasional
d. Memonitori tindak lanjut temuan hasil Audit Internal
e. Menyusun laporann manajemen dibidangnya



Gambar 2.1.3.1 Struktur Organisasi

2.2 Landasan Teori

2.2.1 Web Profile

Web profile sebuah website yang membuat profile perusahaan, diantaranya mengenai tentang perusahaan atau bisnis, visi dan misi, our service atau product, our client and principal dan lain sebagainya. Saat ini website jenis ini banyak digunakan sebagai pengganti company profile konvensional bentuk cetakan. Bisa juga untuk bahan presentasi kepada principal atau klien prospek. Contoh web profile : situs pemerintah, badan atau perusahaan. [1]

2.2.2 Intranet

Intranet adalah jaringan komputer di suatu perusahaan atau organisasi untuk mamakai resource secara bersamaan dan memungkinkan untuk saling bertukar informasi, dengan menerapkan teknologi dan arsitektur internet. Intranet menggunakan TCP/IP yang memungkinkan satu komputer mengirim dan memberi alamat data ke komputer lain sekalius memastikan pengiriman data sampai ke tujuan dengan baik. Pada dasarnya perbedaan internet dan intranet dapat dilihat dalam cakupan aksesnya. Pada internet cakupannya bersifat global, komunikasi melalui saluran telekomunikasi publik dan pemakainya tidak dibatasi atau dibedakan posisinya terkait kepentingan dengan informasi yang diaksesnya. Sedangkan pada intranet cakupannya terbatas hanya pada suatu perusahaan atau organisasi itu saja, sehingga pemakainya bisa dibatasi sesuai kebutuhan dan ketentuan perusahaan atau oranisasi tersebut. [2]

2.2.3 Intranet WWW (World-Wide Web)

World Wide Web yang sering disingkat dengan www atau web adalah suatu metode untuk menampilkan informasi di internet, baik berupa teks, gambar, suara maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (link) satu dokumen (hypertext) dengan dokumen lainnya yang dapat diakses melalui sebuah browser. Browser adalah perangkat lunak untuk mengakses halaman-halaman web, seperti Internet Explorer, Google Crome, Mozilla Firefox, Opera dan lain-lain.

Informasi yang disajikan melalui browser dibangun dengan bahasa semi pemprograman HyperText Markup Language atau yang lebih akrab disebut HTML, dan memungkinkan untuk bisa ditinkatkan fungsinya dengan menyisipkan kode-kode bahasa pemprograman web, seperti PHP, ASP, JSP dan lain-lain, sehinga mampu menampilkan informasi yan lebih interaktif dan dinamis serta terhubung dengan database. [3]

2.2.4 Apache HTTP Server

Server HTTP Apache atau Server Web/WWW Apache adalah server web yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah. Apache

merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan Apache Software Foundation.[4]

2.2.5 HTML

HyperText Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegerasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman web dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (Standard Generalized Markup Language), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa). [5]

2.2.6 CSS

Cascading Style Sheet (CSS) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemograman.

Sama halnya styles dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa style, misalnya heading, subbab, bodytext, footer, images, dan style lainnya

untuk dapat digunakan bersama-sama dalam beberapa berkas (file).Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya.CSS adalah bahasa style sheet yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda. [6]

2.2.7 PHP

PHP dibuat oleh Rasmus Lerdorf pada tahun 1994, pada awalnya tidak untuk di distribusikan dan hanya digunakan pada home page pribadinya. Pada tahun 1995 dikeluarkan versi pertama yang dapat digunakan oleh umum dengan nama personal home page tools. Ditulis kembali pada pertengahan 1995 dan diberi nama sebagai PHP/F1 version 2. F1 berasal dari paket Rasmus yang mana merupakan html interpreter untuk data form. Pada hasil kombinasi tersebut juga ditambah dukungan SQL. PHP/F1 terus berkembang dan banyak orang mulai memberikan kontribusi dalam pengembangan. Pada tahun 1996 PHP/F1 diperkirakan telah digunakan 15.000 situs web dunia, dan pada pertengahan 1997 jumlah ini berkembang melebihi 50.000. pada pertengahan 1997 juga terjadi perubahan pada PHP dimana berubah menjadi proyek yang didukung oleh team yang lebih terorganisasi. Parser-nya ditulis ulang oleh Zeev Suraski dan Andi Gutmans dan parser baru inilah yang membentuk basis untuk PHP versi 3. Banyak kode utility telah selesai ditulis kembali. PHP seperti kita ketahui ialah bahasa pemrograman berbasis web.

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaanya.
2. Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahamanan, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system [7]

2.2.8 Database Management System (DBMS)

Database Management System (DBMS) adalah suatu sistem atau perangkat lunak yang dirancang untuk mengelola suatu basis data dan menjalankan operasi terhadap data yang diminta banyak pengguna. DBMS telah berkembang menjadi bagian standar di bagian pendukung suatu perusahaan. Contoh DBMS adalah Oracle, SQL server, MS Access, MySQL. DBMS merupakan perangkat lunak yang dirancang untuk dapat melakukan utility dan mengelola koleksi data dalam jumah yang besar. DBMS juga dirancang untuk dapat melakukan manipulasi data dengan lebih mudah. [8]

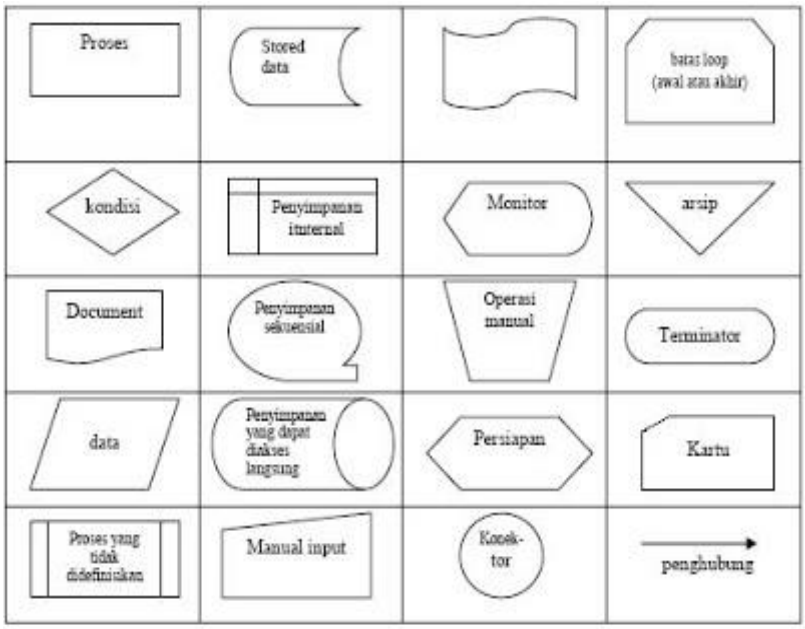
2.2.9 Flow Map

Pengertian Flowmap adalah campuran peta dan flow chart,yang menunjukan pergerakan benda dari satu lokasi ke lokasi lain, seperti jumlah orang dalam migrasi, atau jumlah paket dalam

jaringan. Flowmap menolong analisis dan programmer untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternaitf-alternatif lain dalam pengoprasian.

Bila seorang analis dan programmer akan membuat flowmap , ada beberapa petunjuk yang harus diperhatikan, sepeti :

1. Flowmap digambarkan dari halaman atas ke bawah dan kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat di mengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus di tentukan secara jelas.
4. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
5. Lingkup dan range dari aktifitas yang sedang di gambarkan harus di telusuri dengan hati-hati.
6. Gunakan simbol-simbol flowchart yang standar.



Nama Simbol dan keterangannya :

1. Dokumen : Menunjukan dokume inpt atau output untuk proses manual atau komputer.
2. Proses : Kegiatan proses yang di lakukan dengan komputerisasi.
3. Proses Manual : Kegiatan proses yang di lakukan dengan manual.
4. Garis alir : Menunjukan alir data dari atau ke proses.
5. Data store : Menunjukan penyimpanan arsip atau dokumen non komputer.
6. Database : Tempat penyimpanan data berbasis database
7. Data elektronik : Menunjukan laporan data elektronik [10]

2.2.10 UML (Unified Modeling Language)

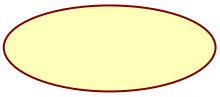
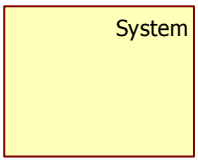
UML (Unified Modeling Language) adalah bahasa pemodelan umum yang digunakan untuk melakukan spesifikasi, visualisasi, konstruksi dan dokumentasi artifak dari software sistem. UML terdiri dari diagram, notasi, konsep dan aturan yang dipakai dalam memodelkan sistem. [9]

Diagram yang digunakan untuk pemodelan dan terdapat pada UML, antara lain :

2.2.10.1 Use Case Diagram

Use Case Diagram merupakan alat komunikasi tingkat tinggi untuk mewakili persyaratan sistem. Diagram yang menunjukkan interaksi antara pengguna dan entitas eksternal lainnya dengan sistem yang sedang dikembangkan.

Simbol yang digunakan dalam use case antara lain :

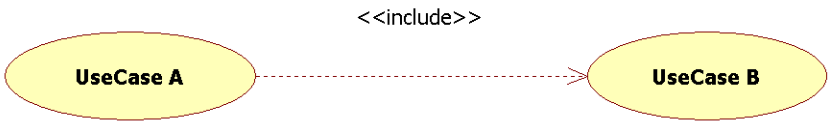
Simbol	Nama Simbol	Fungi
	Aktor	Pihak yang mengakses use case
	Use Case	Mewakili Apa yang Sistem bisa lakukan
	Association	Merelasikan Usecase dangan Aktor
	System Boundary	Menggambarkan batasan sistem terhadap lingkungannya

Tabel 2.2.10.1 Daftar Simbol Use Case

Relasi yang bisa digunakan use case antara lain :

1) Include

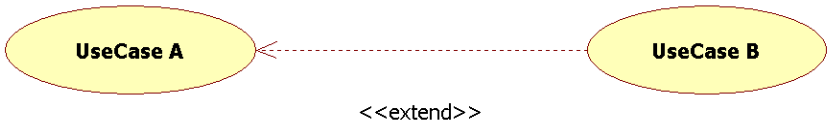
Use case yang direlasikan dengan relasi include berarti usecase yang di-include akan ikut diinplemantasikan setiap kali usecase yang meng-include diimplementasikan. Relasi Include direpresentasikan dengan garis putus-putus bertuliskan <<include>> ke arah use case yang akan di-include.



Gambar 2.2.10.1 Relasi Include dalam usecase

2) Extend

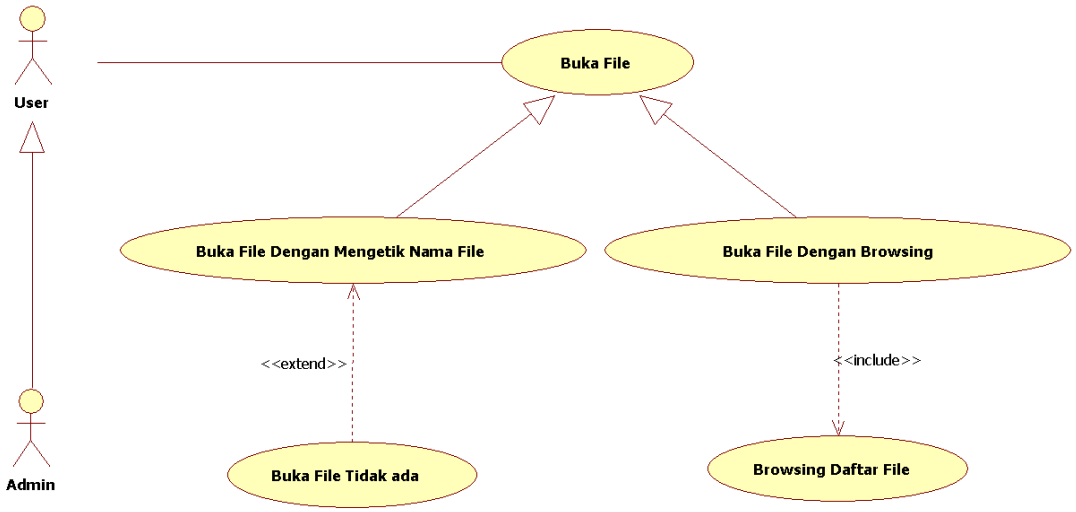
Use case yang direlasikan dengan relasi extend berarti use case yang meng-extend akan bisa saja diimplementasi atau tidak setiap kali use case yang di-extend diimplementasi. Relasi extend direpresentasikan dengan garis putus-putus bertuliskan <<extend>> ke arah use case yang akan di-extend.



Gambar 2.2.10.2 Relasi extend dalam usecase

3) Generalization

Generalization digunakan untuk membuat aktor dan use case yang lebih spesifik. [9]

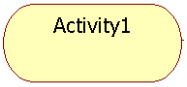


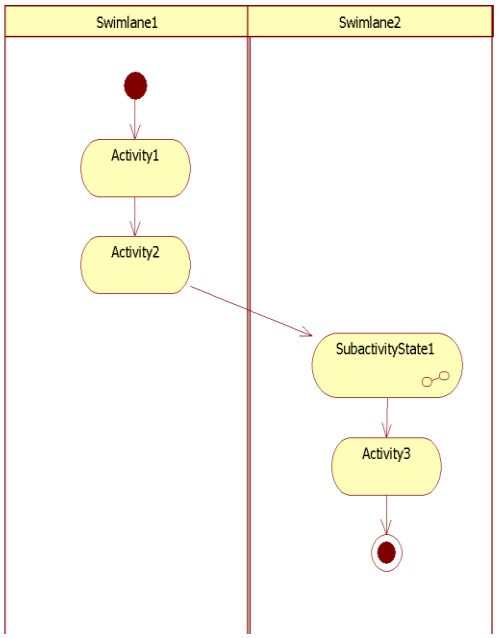
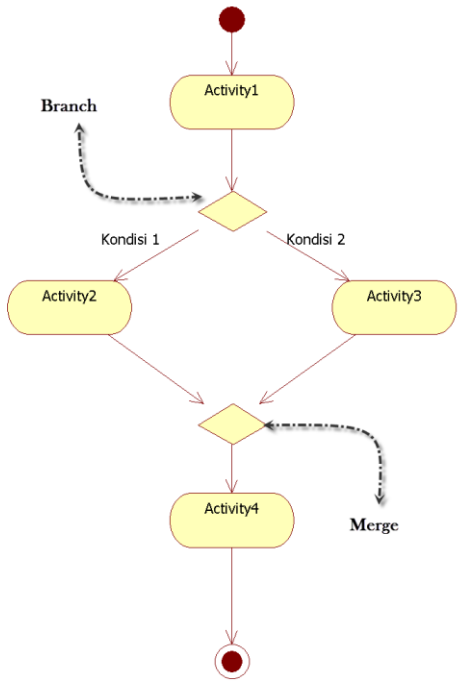
Gambar 2.2.10.3Penggunaan Relasi Dalam Usecase

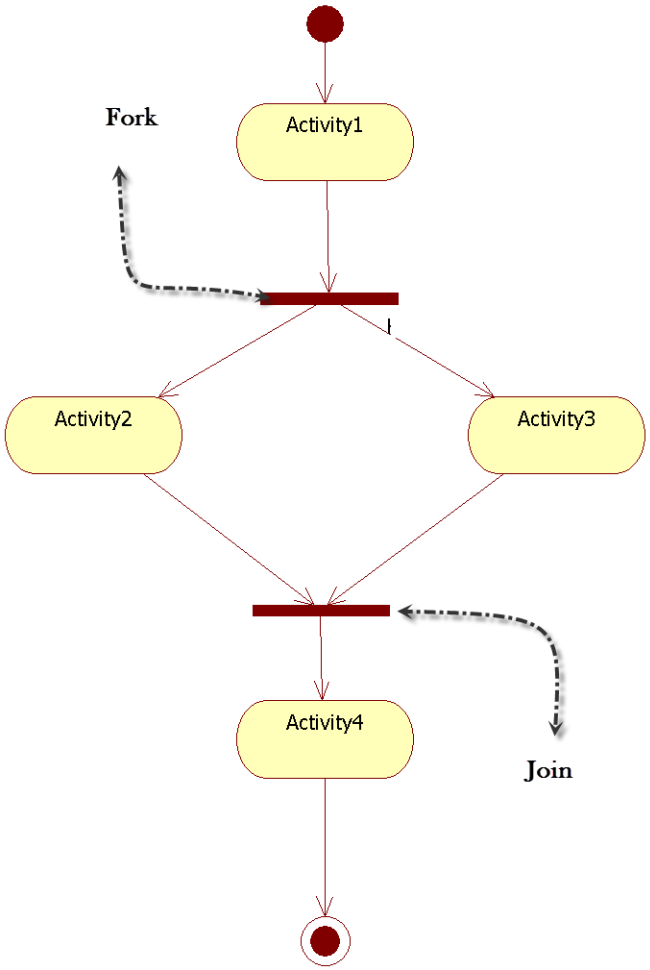
2.2.10.2 Activity Diagram

Activity Diagram menangkap alur dari system, termasuk tindakan utama dan titik keputusan. Diagram ini berguna untuk mendokumentasikan proses bisnis.

Simbol yang digunakan dalam diagram activity antara lain :

Simbol	Nama Simbol	Keterangan
	Initial State	Penanda awal dari aktivitas
	Final State	Penanda akhir dari aktivitas
	Transisi	Menggambarkan alur antar aksi
	Aksi	Menggambarkan aksi yang ada pada satu aktivitas
	Sub Activity	mewakili atau penyederhanaan diagram aktivitas yang lain.

		Swimlane	Mempartisi aksi-aksi pada diagram aktivitas
		Branch	Memisahkan aksi berdasarkan kondisi tertentu
		Merge	Menggabungkan aksi yang dipisahkan oleh branch
		Fork	Memisahkan aksi yang bisa dijalankan secara parallel (independen)

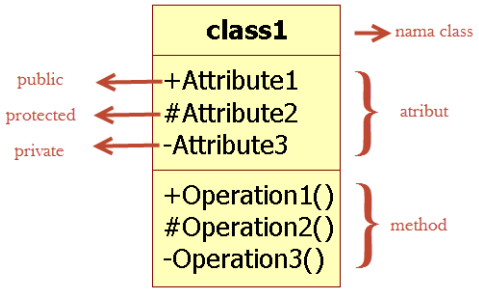
	Join	Menggabungkan aksi yang dipisahkan oleh fork agar sequence kembali

2.2.10.3 Class Diagram

Class Diagram adalah sebuah spesifikasi yang jika diinstansikan akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi objek.

Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). Class diagram menggambarkan struktur dan deskripsi class, package dan object beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Sebuah class memiliki tiga area pokok:

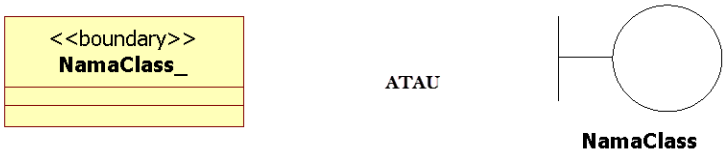
1. Nama , merupakan nama dari sebuah kelas.
2. Atribut, merupakan properti dari sebuah kelas. Atribut melambangkan batas nilai yang mungkin ada pada obyek dari class.
3. Operasi atau Method, adalah sesuatu yang bisa dilakukan oleh sebuah atau yang dapat dilakukan oleh class lain terhadap sebuah class. Atribut dan metoda dapat memiliki salah sifat berikut:
    a. , tidak dapat dipanggil dari luar class yang bersangkutan
    b. *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya.
    c. *Public*, dapat dipanggil oleh siapa saja.



Gambar 2.2.10.4 Class

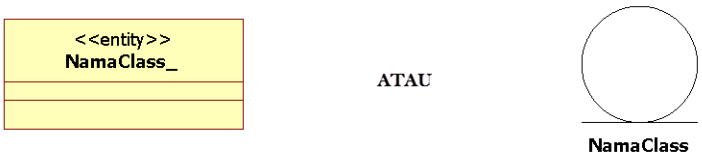
Class juga dapat merupakan implementasi dari sebuah interface, yaitu class abstrak yang hanya memiliki metode. Interface tidak dapat langsung diintansikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat runtime.

Kelas pembatas (Boundary) merupakan kelas yang menyalurkan interaksi antara sistem dengan dunia sekitarnya3. Seperti form, laporan, obyek – obyek pada form. Setiap kelas pembatas biasanya akan mewakili interaksi antara seorang actor dengan use case.



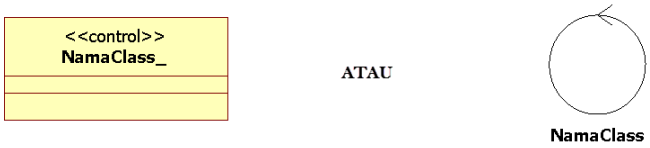
Gambar 2.2.10.5 Notasi Boudary Class

Kelas entitas (Entity) biasanya digunakan untuk menangani informasi yang mungkin akanselalu disimpan dalam proses bisnis. Cara melakukan identifikasi kelas entitas adalah dengan memperhatikan kata benda. Seperti kalimat "petugas mencatat peminjaman buku", buku bisa menjadi Class entitas. Kelas entitas juga dapat digunakan untuk mewakili table – table yang terdapat dalam database.



Gambar 2.2.10.6 Notasi Entity Class

Kelas control (Control) bersifat opsional, apabila kelas ini digunakan maka satu kelas control untuk satu use case yang digunakan mengatur kejadian dalam use case tersebut. Contohnya kelas transaksi yang bertanggung jawab dalam transaksi baik peminjaman maupun pengembalian buku dalam suatu perpustakaan.



Gambar 2.2.10.7

Relasi atau relationship merupakan keterhubungan antar kelas yang muncul pada saat sebuah kelas berinteraksi dengan kelas-kelas lainnya. Didalam class diagram, setiap kelas pasti akan berinteraksi dengan baik satu maupun lebih kelas. Relasi yang muncul pada setiap keterhubungan antar kelas juga memiliki atribut-atribut yang akan lebih menjelaskan defenisi dari sebuah relasi yang terjadi.

Multiplicity atau kardinalitas adalah jumlah banyaknya objek sebuah class yang berelasi dengan sebuah objek lain pada class lain yang berasosiasi pada class tersebut. Untuk menyatakan

multiplisitas perancang dapat meletakkannya diatas garis asosiasi berdekatan dengan class yang sesuai.

Multiplicity	Keterangan
0..1	0 atau 1
1	Tepat 1
0..*	0 atau lebih
1..*	1 atau lebih
*	tidak tertentu jumlahnya

Tabel 2.2.10.2 Relasi Class - Multiplicity

1) Asosiasi (Association)

Relasi ini terjadi pada class-class dimana salah satu instance dari class tersebut menjadikan instance dari class lainnya sebagai atribut dirinya. Bentuk Association bisa diartikan uni-directional (satu arah) atau bidirectional (dua arah).



Gambar 2.2.10.8 Asosiasi Bidirectional(dua arah)



Gambar 2.2.10.9 Asosiasi Uni-Directional (satu arah)

2) Composition

Relasi ini terjadi pada class-class digunakan untuk menyusun object-object sederhana menjadi suatu object yang lebih kompleks. Bentuk Relasi Composition dapat di argikan "… is part of …" ("… bagian dari …"). Terdapat class sebagai part class (class bagian) yang merupakan

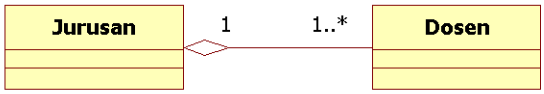
bagian dari class lain (whole class). Namun jika whole class tidak ada , part class tidak dapat berdiri sendiri.



Gambar 2.2.10.10 Relasi Class Coposition

3) Aggregation

Relasi ini seperti hal nya betuk relasi composition akan tetapi jika whole class tidak ada part class masih bisa berdiri sendiri. Bentuk Relasi Aggregation dapat di artikan "… owns a …" ("… terdiri dari …").



Gambar 2.2.10.11Relasi Class Aggregation

4) Dependency

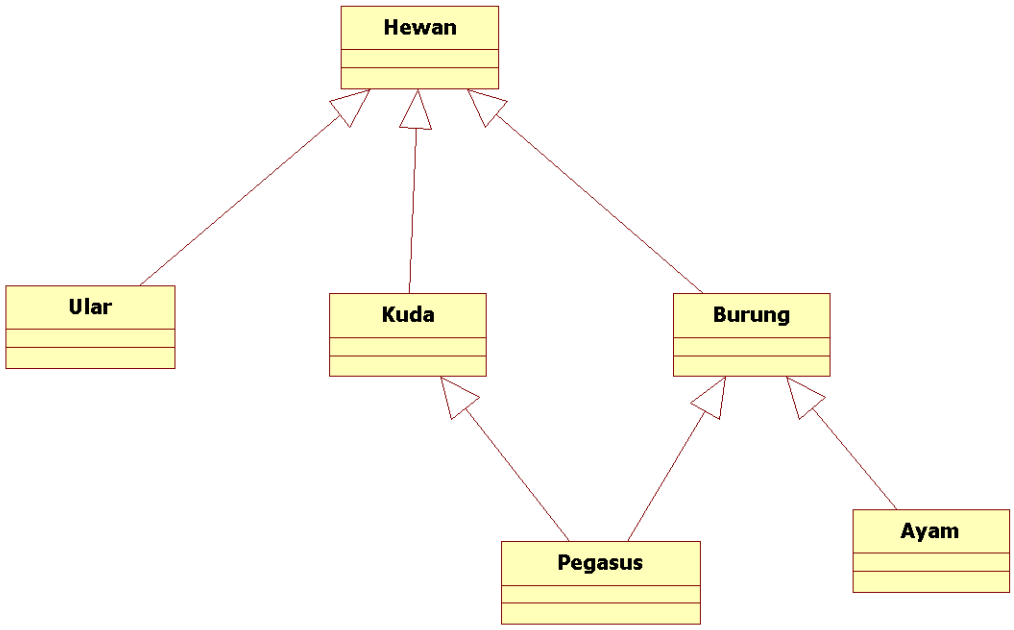
Merupakan hubungan ketergantungan antara class. Suatu class memiliki ketergantungan terhadap class lain, tetapi tidak berlaku sebaliknya. Perubahan pada salah satu elemen class (pemasok) berdampak pada class lain (klien). Penggunaan class berelasi dependency terjadi jika class (klien) berada pada parameter method, return value atau sebagai variable local di dalam class (pemasok).



Gambar 2.2.10.12Relasi Class Dependency

5) Generalization

Dapat diartikan sebagai relasi "… is a …", penggunaan relasi ini untuk merepresentasikan pewarisan. Suatu class (child class) dapat diturunkan dari kelas lain dan mewarisi semua atribut dan method induknya (parent class) dan dapat menambah method atau atribut baru. [9]



Gambar 2.2.10.13 Relasi Class Generalization

2.2.10.4 Sequence Diagram

Sequence diagram menjelaskan bagaimana objek-objek berkolaborasi dalam suatu urutan tingkah laku. Biasanya sequence diagram menangkap tingkah laku dari suatu scenario. Diagram ini menunjukkan sejumlah objek dan pesan-pesan yang dilewati diantara objek-objek ini dalam suatu usecase.

1) Life Line

Lifeline merepresentasikan individual objek yang berpartisipasi di dalam sequence diagram. Lifeline biasanya berbentuk kotak dengan nama object di dalamnya, Lifeline dapat juga menggunakan Aktor terlibat atau stereotype seperti: boundary, control, dan entity.

2) Life Create dan Delete

Suatu lifeline atau object dapat diciptakan dan dihancurkan dalam suatu rentang waktu.

3) Massage

Message mendefinisikan suatu jenis komunikasi atara Life Line. Message disini bisa berupa pemanggilan saja atau pemanggilan dan penerimaan mesage. Return message bersifat optional dalam sequence diagram. Penggunaan return message tergantung tingkat ke-detail-an dari sequence diagram. Return message sangat berguna jika ingin menggambarkan sequence secara detail.

4) Self Message

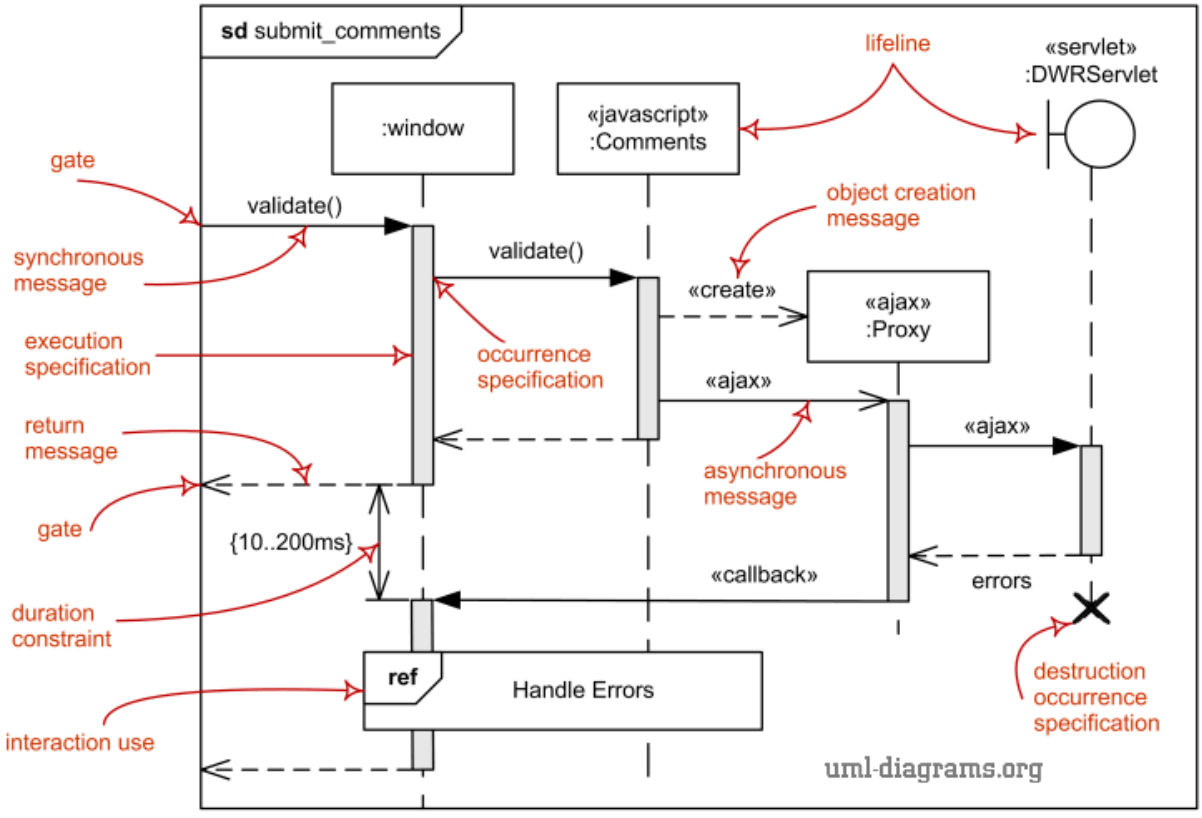
Sebuah self message merepresentasikan pemanggilan operasi yang recursif atau pemanggilan metode yang dimiliki oleh object itu sendiri.

5) Fragments

Pada dasarnya sequence diagram tidak dibuat untuk menampilkan logis procedural yang kompleks akan tetapi adakalanya penggambaran procedural logis ini diperlukan untuk memperlihatkan alur logis yang lebih kompleks. Fragment berguna untuk menggambarkan sequence dalam suaujjtu kondisi tertentu seperti; looping, alternate, parallel, break, negative, critical, ignore, consider, weak sequence, dan strict.

6) Gate

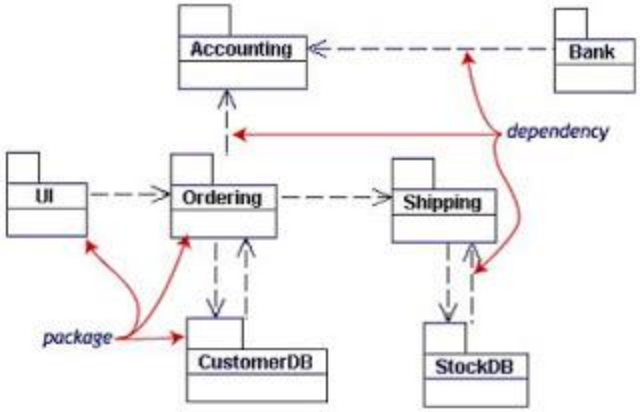
Gate adalah titik koneksi untuk mengkoneksikan message dari dalam fragment dengan message di luar fragment. [9]



Gambar 2.2.10.114 Sequence diagram

2.2.10.5 Package Diagram

Diagram objek melengkapi notasi grafik untuk pemodelan objek, kelas dan relasinya dengan yang lain. Diagram objek bermanfaat untuk pemodelan abstrak dan membuat perancangan program. Untuk mengatur pengorganisasian diagram Class yang kompleks, dapat dilakukan pengelompokan kelas-kelas berupa package (paket-paket). Package adalah kumpulan elemen-elemen logika UML. Gambar di bawah ini mengenai model bisnis dengan pengelompokan kelas-kelas dalam bentuk paket-paket. [9]



Gambar 2.2.10.115Package Diagram