

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Pengertian Sistem**

Secara umum, sistem berarti susunan yang teratur dari pandangan, teori, asas dan sebagainya. Dengan kata lain, sistem adalah suatu kesatuan usaha yang terdiri dari bagian-bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan kompleks. Pengertian tersebut mencerminkan adanya beberapa bagian dan hubungan antara bagian, ini menunjukkan kompleksitas dari sistem yang meliputi kerja sama antara bagian yang interpenden satu sama lain. Selain itu dapat dilihat bahwa sistem berusaha mencapai tujuan. Pencapaian tujuan ini menyebabkan timbulnya dinamika, perubahan-perubahan yang terus menerus perlu dikembangkan dan dikendalikan. Definisi tersebut menunjukkan bahwa sistem sebagai gugus dan elemen-elemen yang saling berinteraksi secara teratur dalam rangka mencapai tujuan atau subtujuan (Marimin ; 2008 : 1).

#### **II.2. Pengertian Informasi**

Informasi diartikan sebagai data yang telah diolah dan bermanfaat bagi pemakai dalam rangka pengambilan keputusan. Sebagai ilustrasi, data jumlah jam kerja karyawan, saat data diproses dapat berubah menjadi informasi. Jika jam kerja dikalikan dengan upah per jam, maka didapat hasil pendapatan kotor. Jika pendapatan kotor ini dijumlahkan, maka penjumlahan ini merupakan total biaya

gaji karyawan harian. Jumlah biaya gaji ini dapat dijadikan informasi bagi manajemen, misalnya dalam alokasi dana. Jadi informasi merupakan data yang telah diolah dan memiliki arti bagi pemakai (Husein Umar ; 2008 : 190).

### **II.3. Pengertian Akuntansi**

Menurut Kusrini (2007:10) dalam bukunya *Tuntunan Praktis Membangun Sistem Informasi Akuntansi Dengan Visual Basic dan Microsoft Sql Server*, Sistem informasi akuntansi merupakan sebuah sistem informasi yang mengubah data transaksi bisnis menjadi informasi keuangan yang berguna bagi pemakainya.

Tujuan dari sistem informasi akuntansi adalah :

1. Mendukung operasi sehari-hari.
2. Mendukung pengambilan keputusan manajemen.
3. Memenuhi kewajiban yang berhubungan dengan pertanggungjawaban.

Komponen-komponen yang terdapat dalam sistem informasi akuntansi adalah sebagai berikut :

1. Orang-orang mengoperasikan sistem tersebut.
2. Prosedur-prosedur, baik manual maupun yang terotomatisasi, yang dilibatkan dalam pengumpulan, pemrosesan dan penyimpanan data aktivitas-aktivitas organisasi.
3. Data tentang proses-proses bisnis.
4. Software yang dipakai untuk memproses data organisasi.
5. Infrastruktur teknologi informasi.

#### **II.4. Bentuk Laporan Keuangan**

Secara umum, ada empat bentuk laporan keuangan pokok yang dihasilkan oleh perusahaan, yaitu :

##### **1. Neraca**

Neraca digunakan untuk menggambarkan kondisi keuangan perusahaan. Neraca bisa digunakan sebagai gambaran potret kondisi keuangan suatu perusahaan pada suatu waktu tertentu (snapshot keuangan perusahaan), yang meliputi aset sumber daya perusahaan dan klaim atas aset tersebut meliputi utang dan saham pribadi. Aset perusahaan menunjukkan keputusan penggunaan dana atau keputusan investasi pada masa lalu, sedangkan klaim perusahaan menunjukkan sumber dana atau keputusan pendanaan di masa lalu. Dengan demikian, neraca adalah menampilkan keseimbangan antara keputusan investasi dengan keputusan pendanaan.

##### **2. Laba Rugi**

Laporan laba rugi merupakan laporan prestasi perusahaan selama jangka waktu tertentu. Berbeda dengan neraca yang merupakan snapshot maka laporan laba rugi hanya mencakup kegiatan operasional perusahaan dalam suatu periode tertentu saja.

##### **3. Arus Kas**

Laporan arus kas menyajikan informasi mengenai aliran kas yang masuk maupun keluar bersih pada suatu periode waktu yang merupakan hasil dari tiga kegiatan pokok perusahaan yaitu operasional, investasi, dan pendanaan.

#### 4. Buku Besar

*Buku besar* / Ledger adalah kumpulan akun-akun yang digunakan untuk meringkas transaksi yang telah dicatat dalam jurnal. Buku besar juga dapat diartikan tahapan catatan terakhir dalam akuntansi book of final entry yang menampung ringkasan data yang sudah dikelompokan atau diklasifikasikan yang berasal dari jurnal (Rosita Ayu Saraswati ; 2012 : 5).

### II.5. Pengertian Jasa

Jasa adalah sesuatu yang diproduksi dan dikonsumsi secara simultan. Jadi, jasa tidak pernah ada dan hasilnya dapat dilihat setelah terjadi. Misal : bila anda potong rambut, jasa dikonsumsi ketika diproduksi, tetapi hasil jasa tampak dan akan berakhir beberapa waktu. Kesenjangan produksi dan konsumsi merupakan perbedaan yang penting. Jasa tidak dapat diproduksi di satu tempat dan dikirim ke tempat lain seperti barang, juga tidak dapat disimpan. Semua karakteristik ini dapat dihubungkan dengan keserentakan produksi dan konsumsi. Jasa terdiri dari tindakan dan interaksi yang merupakan kontak sosial. Jasa lebih dari sekedar hasil sesuatu yang tak terhalang, dan jasa merupakan interaksi sosial antara produsen dan konsumen.

Studi tentang jasa mulai dengan penyamarataan tentang apa yang harus dipertimbangkan suatu topik yang luas :

1. Semua orang adalah seorang tenaga ahli pada jasa.
2. Jasa adalah *idiosyncratic* dan bergantung situasi.
3. Kualitas produk bukanlah jasa kualitas.

4. Kebanyakan jasa berisi suatu campuran atribut tak terukur dan terukur yang mendasari suatu paket jasa.
5. Jasa kontak tinggi adalah yang berpengalaman, sedangkan barang adalah yang dikonsumsi.
6. Manajemen yang efektif dari jasa membutuhkan pemahaman dari marketing dan personel, seperti halnya operasi.
7. Jasa sering mengambil bentuk pertemuan antara pelanggan dan proses pengiriman jasa, termasuk *face to face*, telepon, hubungan melalui surat (Hery Prasetya ; 2009 : 72).

## II.6. Pengertian Java

Menurut Hariyanto (2011:1) Bahasa pemrograman Java merupakan karja Sun Microsystem Inc. Rilis resmi level beta dilakukan pada November 1995. Dua bulan berikutnya, Netscape menjadi perusahaan pertama yang memperoleh lisensi bahasa Java dari Sun.

Pada tahun 1996, Sun mengeluarkan JSDK (*Java Software Development Kit*) yang kemudian secara berturut-turut:

1. Versi 1.02 yang mendukung konektifitas basis data dan objek-objek tersebar.
2. Versi 1.1 pada tahun 1997 ditambahkan model kejadian (*event model*) yang andal, *internationalization* dan model komponen JavaBeans.
3. Versi 1.2 pada tahun 1998 mempunyai banyak peningkatan diantaranya User Interface Toolkit Swing yang memungkinkan pemrogram membuat aplikasi berbasis GUI yang sepenuhnya portable. Sejak ini, disebut Java 2.
4. Versi 1.3 dirilis tahun 200 dengan banyak peningkatan.

5. Versi 1.4 ditambahkan fasilitas asersi untuk dukungan design-by-contract.
6. Versi 1.5 ditambahkan fitur-fitur baru di level bahasa diantaranya generics (parameterized type), enumeration, dan metadata. versi ini disebut JDK.
7. Versi 1.5 ditambahkan lightweight database system yaitu Derby. Derby merupakan hasil pengembangan dari proyek basis data Apache. Derby mulanya merupakan CloudScape dari IBM.

## **II.7. Pengertian NetBeans**

NetBeans merupakan salah satu proyek *open source* yang disponsori oleh *Sun Microsystem*. Proyek ini berdiri pada tahun 2000 dan telah menghasilkan 2 produk, yaitu NetBeanss IDE dan NetBeans Platform. NetBeans IDE merupakan produk yang digunakan untuk melakukan pemrograman baik menulis kode, meng-*compile*, mencari kesalahan dan mendistribusikan program. Sedangkan NetBeans Platform adalah sebuah modul yang merupakan kerangka awal / pondasi dalam bangun aplikasi desktop yang besar.

NetBeans juga menyediakan paket yang lengkap dalam pemrograman dari pemrograman standar (aplikasi desktop), pemrograman *enterprise*, dan pemrograman perangkat mobile. Saat ini NetBeans telah mencapai versi 6.8 (Wahana Komputer ; 2010 : 15).

## **II.8. Pengertian Database**

*Database* merupakan kumpulan data yang saling berhubungan, hubungan antar data dapat ditunjukkan dengan adanya *field* kunci dari setiap tabel yang beda.

Dalam satu *file* atau tabel terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* terdiri dari *field* yang saling berhubungan menunjukkan bahwa *field* tersebut satu pengertian yang lengkap dan disimpan dalam satu *record*. Basis data mempunyai beberapa kriteria penting yaitu :

1. Bersifat data oriented dan bukan program oriented.
2. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya.
3. Dapat dikembangkan dengan mudah, baik *volume* maupun strukturnya.
4. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah.
5. Dapat digunakan dengan cara-cara yang berbeda.

Prinsip utama *database* adalah pengaturan data dengan tujuan utama fleksibel dan kecepatan pada saat pengambilan data kembali. Adapun ciri-ciri basis data di antaranya adalah sebagai berikut :

1. Efisiensi meliputi kecepatan, ukuran dan ketepatan.
2. Data dalam jumlah besar.
3. Berbagi pakai (dipakai bersama-sama atau *sharebility*).

Mengurangi bahkan menghilangkan terjadinya duplikasi dan data yang tidak konsisten (Windu Gata ; 2013 : 19).

## **II.9. Pengertian MySQL**

Menurut Supardi (2007 : 97), perangkat lunak MySQL adalah perangkat lunak basis data server yang terkenal dan bersifat open-source dengan dukungan driver yang luas dari berbagai vendor. MySQL adalah seakuntansi implementasi

dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya. SQL (*Structured Query Language*). SQL adalah seakuntansi konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

#### **II.10. Entity Relationship Diagram (ERD)**

*Entity relationship* (ER) data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antarobjek. Entitas adalah sesuatu atau objek dalam dunia nyata yang dapat dibedakan dari objek lain. Misal: mahasiswa, dan matakuliah. Entitas digambarkan dalam basis data dengan kumpulan atribut. Misalnya: nim, nama, alamat, dan kota. Relasi adalah hubungan antara beberapa entitas. Misalnya: relasi menghubungkan mahasiswa dengan mata kuliah yang diambilnya (D. Tri Octafian ; 2011 : 150).

#### **II.11. Kamus Data**

Kamus data (*data dictionary*) mencakup definisi-definisi dari data yang disimpan di dalam basis data dan dikendalikan oleh sistem manajemen basis data. Figur 6.5 menunjukkan hanya satu tabel dalam basis data jadwal. Struktur basis

data yang dimuat dalam kamus data adalah kumpulan dari seluruh definisi *field*, definisi tabel, relasi tabel, dan hal-hal lainnya. Nama *field* data, jenis data (seperti teks atau angka atau tanggal), nilai-nilai yang valid untuk data, dan karakteristik-karakteristik lainnya akan disimpan dalam kamus data. Perubahan-perubahan pada struktur data hanya dilakukan satu kali di dalam kamus data, program-program aplikasi yang mempergunakan data tidak akan ikut terpengaruh (Raymond McLeod ; 2008 : 171).

## **II.12. Teknik Normalisasi**

Proses normalisasi menyediakan cara sistematis untuk meminimalkan terjadinya kerangkapan data di antara relasi dalam perancangan logikal basis data.

Format normalisasi terdiri dari lima bentuk, yaitu:

### **II.12.1. Bentuk-bentuk Normalisasi**

#### **a. Bentuk normal tahap pertama (1<sup>st</sup> Normal Form)**

Suatu tabel dikatakan sudah 1NF jika telah memenuhi ketentuan sebagai berikut:

- Tidak ada atribut mempunyai nilai berulang atau nilai array
- Tidak mempunyai baris yang rangkap

Bentuk unnormal mengijinkan nilai-nilai pada suatu atribut dapat berulang.

#### **b. Bentuk normal tahap kedua (2<sup>nd</sup> normal form)**

Relasi dapat dikatakan format normal kedua jika sudah dalam format normal pertama dan diikuti kondisi sebagai berikut:

- Key terdiri dari atribut tunggal

- Setiap atribut nonkey ketergantungan fungsional pada semua key atau tidak terjadinya ketergantungan pada key *composite*.

Misalnya tabel UNIV berada dalam normal kedua dengan mengasumsikan DNO sebagai key, kecuali CRSE. Jika ditentukan CNO dan SECNO sebagai key composite, atribut nonkey CNAME tergantung hanya pada CNO, bukan pada SECNO, sehingga CNAME tidak secara ketergantungan fungsional penuh terhadap key (CNO, SECNO).

**c. Bentuk normal tahap ketiga (3<sup>rd</sup> normal form)**

Relasi dikatakan format normal ketiga jika sudah dalam format normal kedua dan tidak ada ketergantungan transitif di antara atribut. Misalnya tabel STUDNT mempunyai atribut SSNO sebagai *key* (2NF). Ketergantungan transitif terjadi di antara DNO dan COLREG. Saat DNO determinan COLREG tanpa melibatkan *key* SSNO. Contohnya, DNO='CS' termasuk COLREG='Arts/Sc.' tidak tergantung oleh atribut SSNO, sehingga STUDNT belum termasuk 3NF. Yang menjadi catatan, ketergantungan transitif tidak akan terjadi jika ada ketergantungan fungsional di antara atribut-atribut *non-key* yang melibatkan *key*.

**d. Boyce Code Normal Form (BCNF)**

BCNF menentukan setiap determinan adalah kunci kandidat (*candidate key*). Misalnya UNIV mempunyai dua determinan yaitu DNO dan DNAME yang merupakan *kunci kandidat* sehingga termasuk

ke dalam BCNF. Di lain pihak CRSLST dalam 3NF tetapi tidak dalam BCNF. Atribut komposisinya (CNO, SECNO, SID, OFRNG) sebagai kunci-kunci kandidat dan tidak ada ketergantungan transitif, sehingga CRSLST termasuk ke dalam 3NF. Namun atribut CNO adalah determinan saat SECNO tergantung penuh secara fungsional terhadap CNO, walaupun CNO bukan kunci kandidat, sehingga CRSLST belum termasuk BCNF.

**e. Bentuk Normal Tahap Keempat dan Kelima**

Bentuk ini adalah bentuk normal ketiga atau BCNF dengan nilai atribut tidak tergantung pada nilai banyak (*multivalued dependency*). Konsep pada bentuk ini adalah ketergantungan pada gabungan beberapa atribut (*join dependency*) (Haidar Dzacko ; 2007 : 12).

### **II.13. UML (*Unified Modeling Language*)**

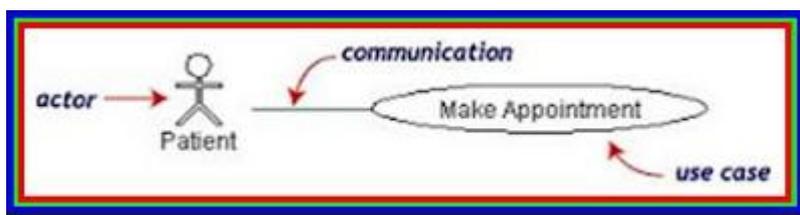
Menurut *Joan Nugroho (2010)* bahwa UML (*Unified Modeling Language*) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. Artifact dapat berupa model, deskripsi atau perangkat lunak) dari system perangkat lunak, seperti pada pemodelan bisnis dan system non perangkat lunak lainnya.

UML merupakan bahasa standar untuk penulisan blueprint software yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak

### II.13.1. Jenis-jenis Diagram UML, yaitu :

#### 1. Use Case Diagram

Use case adalah abstraksi dari interaksi antara system dan actor. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai.



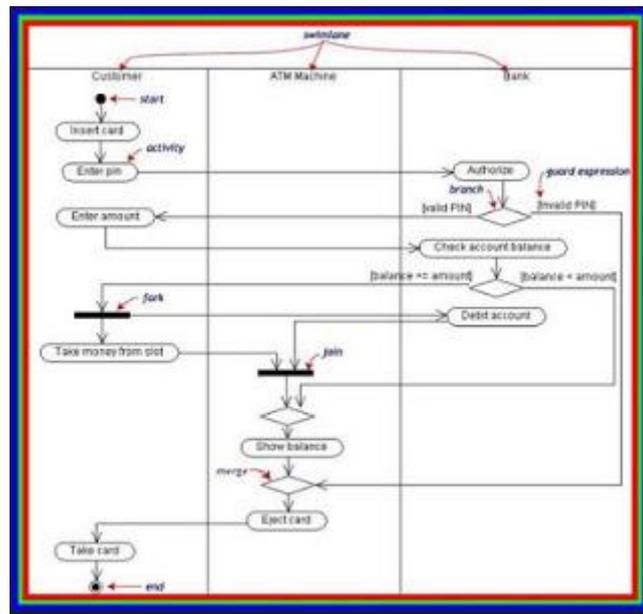
**Gambar II.1. Usecase Diagram**

*Sumber : Joan Nugroho ; 2010 : 56*

#### 2. Activity Diagram

Activity diagram menyediakan analisis dengan kemampuan untuk memodelkan proses dalam suatu sistem informasi. Activity diagram dapat digunakan untuk alur kerja model, use case individual, atau logika keputusan yang terkandung dalam metode individual. Activity diagram juga menyediakan pendekatan untuk proses pemodelan paralel. Activity diagram lebih lanjut .

Pada dasarnya, diagram aktifitas canggih dan merupakan diagram aliran data yang terbaru. Secara teknis, diagram aktivitas menggabungkan ide-ide proses pemodelan dengan teknik yang berbeda termasuk model acara, statecharts, dan Petri Nets.

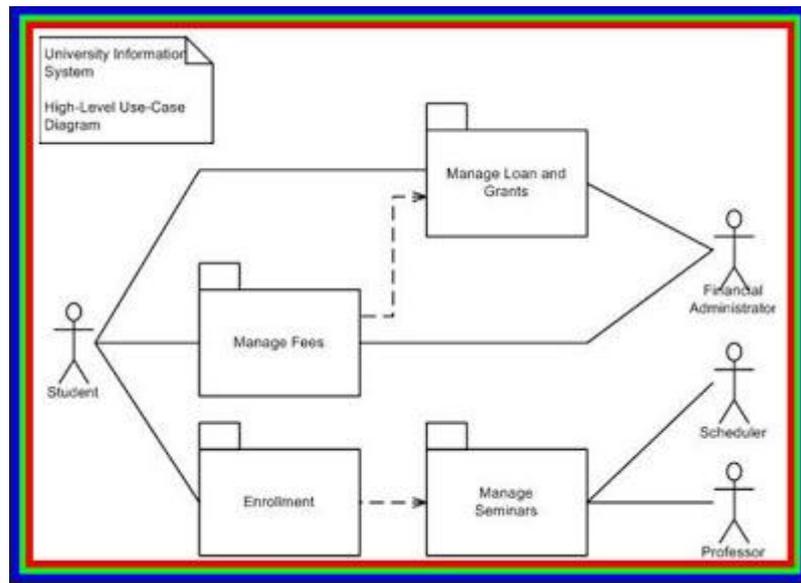


**Gambar II.2. Activity Diagram**  
 Sumber : Joan Nugroho ; 2010 : 56

### 3. Package Diagram

Package diagram utamanya digunakan untuk mengelompokkan elemen diagram UML yang berlainan secara bersama-sama ke dalam tingkat pembangunan yang lebih tinggi yaitu berupa sebuah paket. Diagram paket pada dasarnya adalah diagram kelas yang hanya menampilkan paket, disamping kelas, dan hubungan ketergantungan, disamping hubungan khas yang ditampilkan pada diagram kelas.

Sebagai contoh, jika kita memiliki sistem pendaftaran untuk kantor dokter, mungkin masuk akal untuk kelompok kelas pasien dengan kelas sejarah medis pasien bersama-sama untuk membentuk paket kelas pasien. Selain itu, dapat berguna untuk membuat paket perawatan yang mengandung gejala penyakit, penyakit, dan obat-obatan khas yang diresepkan untuk mereka.

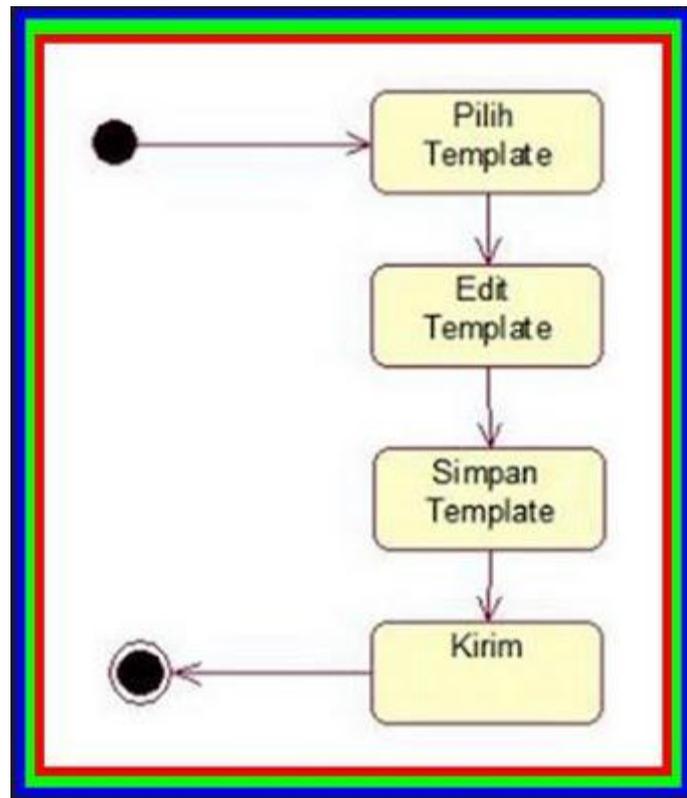


**Gambar II.3. Package Diagram**  
*Sumber : Joan Nugroho ; 2010 : 57*

#### 4. State Machines Diagram

Statechart diagram digunakan untuk memodelkan perilaku dinamis satu kelas atau objek. Statechart diagram memperlihatkan urutan keadaan sesaat (state) yang dilalui sebuah objek, Kejadian yang menyebabkan sebuah transisi dari suatu state atau aktivitas kepada yang lainnya.

Statechart diagram khusus digunakan untuk memodelkan tahap-tahap diskrit dari sebuah siklus hidup objek, sedangkan Activity diagram paling cocok untuk memodelkan urutan aktifitas dalam suatu proses.

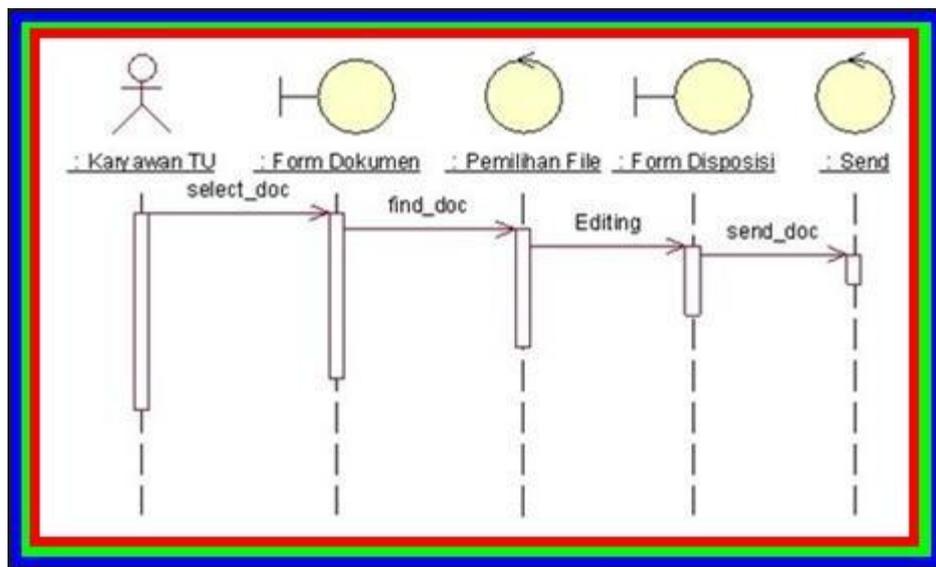


**Gambar II.4. State Machines Diagram**

*Sumber : Joan Nugroho ; 2010 : 57*

## 5. Sequence Diagram

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya sequence diagram adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan use case diagram.

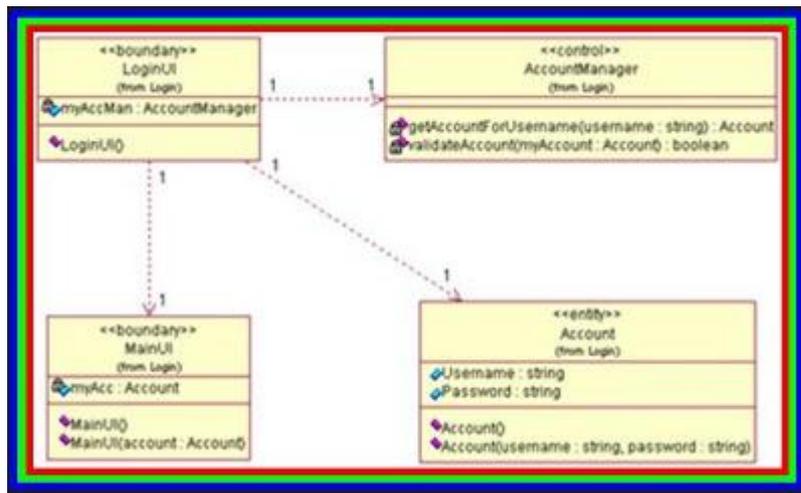


**Gambar II.5. Sequence Diagram**

*Sumber : Joan Nugroho ; 2010 : 58*

## 6. Class Diagram

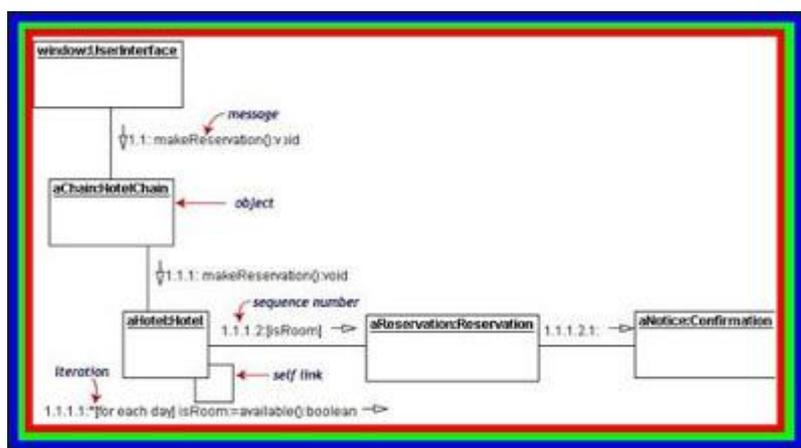
Tujuan utama dari class diagram adalah untuk menciptakan sebuah kosa kata yang digunakan oleh analis dan pengguna. Diagram kelas biasanya merupakan hal-hal, ide-ide atau konsep yang terkandung dalam aplikasi. Misalnya, jika anda sedang membangun sebuah aplikasi penggajian, diagram kelas mungkin akan berisi kelas yang mewakili hal-hal seperti karyawan, cek, dan pendaftaran gaji. Diagram kelas juga akan menggambarkan hubungan antara kelas.



**Gambar II.6. Class Diagram**  
 Sumber : Joan Nugroho ; 2010 : 59

## 7. Communication Diagram

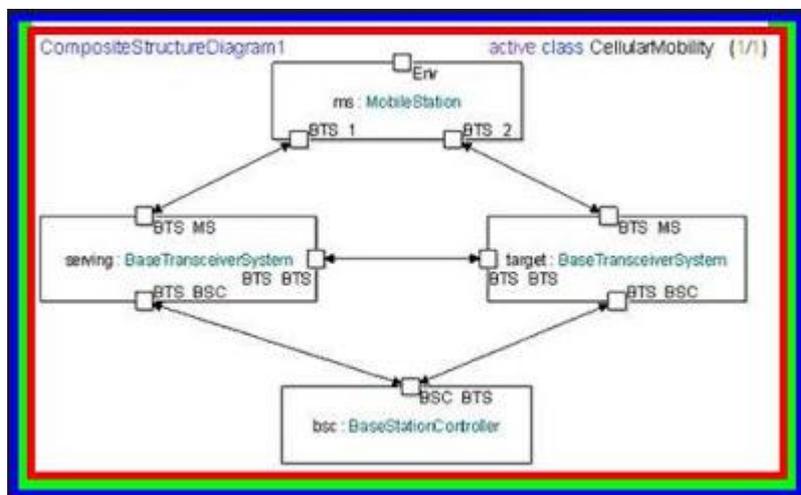
Collaboration diagram menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek. Setiap message memiliki sequence number, dimana message dari level tertinggi memiliki Nomor 1. Diagram membawa informasi yang sama dengan diagram Sequence, tetapi lebih memusatkan atau memfokuskan pada kegiatan obyek dari waktu pesan itu dikirimkan.



**Gambar II.7. Communication Diagram**  
 Sumber : Joan Nugroho ; 2010 : 59

## 8. Composite Structure Diagram

Diagram struktur komposit adalah diagram yang menunjukkan struktur internal classifier, termasuk poin interaksinya ke bagian lain dari system. Hal ini menunjukkan konfigurasi dan hubungan bagian, yang bersama-sama melakukan perilaku classifier. Diagram struktur komposit merupakan jenis diagram struktur yang statis dalam UML, yang menggambarkan struktur internal kelas dan kolaborasi.



**Gambar II.8. Composite Structure Diagram**

*Sumber : Joan Nugroho ; 2010 : 60*