BAB II

LANDASAN TEORI

II.1. Pengertian Sistem

Suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Dari defenisi ini dapat dirinci lebih lanjut pengertian sistem secara umum, yaitu sebagai berikut :

a. Setiapa sistem terdiri dari berbagai unsur. Sistem pernapasan kita terdiri dari suatu kelompok unsur, yaitu hidung, saluran pernapasan, paru-paru dan darah. Unsur-unsur suatu sistem terdiri dari subsistem yang lebih kecil, yang terdiri pula dari kelompok-kelompok unsur yang membentuk suatu subsistem tersebut.
b. Unsur-unsur tersebut merupakan bagian yang tak terpisahkan dari subsistem yang bersangkutan. Unsur-unsur sistem berhubungan erat satu sama lain di mana sifat serta kerja sama antarunsur dalam sistem tersebut mempunyai bentuk tertentu.
c. Unsur-unsur di dalam sistem tersebut bekerja sama untuk mencapai tujuan sistem. Setiap sistem mempunyai tujuan tertentu.
d. Suatu sistem merupakan bagian dari sistem lain yang lebih besar.

Suatu sistem dibuat untuk menangani suatu yang berulang kali atau yang secara rutin terjadi. Pendekatan sistem merupakan suatu filsafat atau presepsi tentang struktur yang mengkoordinasi kegiatan-kegiatan dan operasi-operasi

dalam suatu organisasi dengan cara yang efisien dan yang paling baik. Suatu sistem dapat dirumuskan sebagai setiap kumpulan komponen atau subsistem yang dirancang untuk mencapai tujuan (Tata Sutabri ; 2012 : 6).

II.2. Pengertian Informasi

Informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang. Untuk memperoleh informasi diperlukan data yang akan diolah dan unit pengolah.

Informasi diperlukan oleh para pemakai (manajemen) pada seluruh level manajemen dalam seluruh fungsi *organisatoris.* Informasi tersebut dapat mempunyai fungsi, antara lain :

1. Menambah pengetahuan

   Adanya informasi akan menambah pengetahuan bagi penerimanya yang dapat digunakan sebagai bahan pertimbangan yang mendukung proses pengambilan keputusan.

2. Mengurang ketidakpastian

   Adanya informasi akan mengurangi ketidakpastian karena apa yang akan terjadi dapat diketahui sebelumnya sehingga menghindari keraguan pada saat pengambilan keputusa.

3. Mengurangi resiko kegagalan

   Adanya informasi mengurangi resiko kegagalan karena apa yang akan terjadi dapat diantisipasi dengan baik sehingga kemingkinan terjadinya kegagalan akan dapat dikurangi dengan pengambilan keputusan yang tepat.

4. Mengurangi keanekaragaman yang tidak diperlukan

   Adanya informasi mengurangi yang tidak diperlukan karena keputusan yang diambil lebih terarah.

5. Memberikan standar yang dapat menentukan pencapai sasaran dan tujuan

   Adanya informasi akan Memberikan standar yang dapat menentukan pencapain sasaran dan tujuan yang telah ditetapkan secara lebih baik berdasarkan informasi yang diperoleh (Eddy Sutanta ; 2011 : 13-15).

II.3. Sistem Informasi Geografis

Salah satu jenis informasi yang berhubungan dengan data spasial (keruangan) yang mengenai daerah – daerah yang terdapat di permukaan bumi adalah sistem informasi geografi (SIG). Deskripsi dari SIG adalah suatu sistem informasi khusus yang mengelola data yang memiliki informasi spasial atau dalam arti yang lebih sempit, adalah suatu sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi yang bereferensi geografis, misalnya data yang diidentifikasikan menurut lokasinya, dalam sebuah *database*.Pada kenyataannya SIG merupakan bagian dari ilmu Geografi Teknik (*Technical Geography*) berbasis computer yang digunakan untuk

menyimpan dan memanipulasi data – data spasial (keruangan) untuk kebutuhan atau kepentingan tertentu.

Secara umum, kegunaan dari sistem informasi geografis adalah memberikan informasi mengenai yang terdiri atas basis data sesuai peruntukannya yang berhubungan dengan data geografi suatu wilayah. Untuk lebih spesifiknya berikut ini adalah manfaat dari SIG: inventaris sumber daya alam, disater management, dan penataan ruang dan pembangunan sarana – prasarana (JURNAL TEKNIK ITS Vol. 4, No. 1 ; 2015 : A65-66).

II.4. Metode Haversine

Metode Haversine digunakan untuk menghitung jarak antara titik di permukaan bumi menggunakan garis lintang (*longitude*) dan garis bujur (*lattitude*) sebagai variabel inputan. Haversine formula adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari R 6.367, 45 km, dan lokasi dari 2 titik di koordinant bola (lintang dan bujur) masing-masing adalah lon1, lat1, dan lon2, lat2, maka rumus Haversine dapat ditulis dengan persamaan sebagai berikut :

x = (lon2-lon1)*cos((lat1+lat2)/2);

y = (lat2-lat1);

d = sqrt(x*x+y*y)*R

Keterangan :

x = Longitude (Lintang)

y= Lattitude (Bujur)

d= Jarak

R= Radius Bumi =6371 km

1 derajat= 0.0174532925 radian [6].

II.5. Quantum GIS

Quantum GIS (QGIS) adalah cross-platform perangkat lunak bebas (*ope source*) desktop pada sistem informasi geografis (SIG). Aplikasi ini dapat menyediakan data, melihat, mengedit, dan kemampuan analisis. Quantum GIS berjalan pada sistem operasi yang berbeda termasuk Mac OS X, Linux, UNIX, dan Microsoft Windows. Dalam perizinan, QGIS sebagai perangkat lunak bebas aplikasi dibawah GPL (*General public License*), dapat secara bebas dimodifikasi untuk melakukan tugas yang berbeda atau lebih khusus (Adam Suseno ; 2012 : 15).

II.6. Database

dalam beberapa leteratur, basis data telah didefinisikan dengan cara yang berbeda-beda. Salah satu definisi yang cukup lengkap dan cukup baik tentang istilah basis data adalah definisi yang diberikan oleh James Martin sebagai berikut.

> *“A database may be defined as a collection of interrelated data stored together without harmful unnecessary redudancy to server one or more applications in an optimal fashion; the data are store so that they are*

independent of program with use the data; a common and controlled approach its used in adding new data in modifying and retrieving exiting data within the database".

Dengan memahami definisi diatas maka istilah basisdata dapat dipahami sebagai suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-samapada suatu media, tanpa perlu suatu kerangkapan data (kalau pun ada maka kerangkapan data tersebut harus seminimal mungkin dan terkontrol [*controlled redundancy]*), data disimpan dengan cara-cara tertentu sehingga mudah digunakan atau ditampilkan kembali, data dapat digunakan oleh satu atau program-program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan dengan program yang akan menggunakannya,data disimpan sedemikan rupa sehingga proses penambahan, pengambilan, dan modifikasi dapat dilakukan dengan mudah dan terkontrol (Eddy Sutanta ; 2011 : 29-30).

II.7. Pengertian PHP

PHP (PHP: *Hypertext Preprocessor*) adalah bahasa *server-side scripting* yang menyatu degan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya dikirimkan ke browser dalam format HTML. Dengan demikian kode program yang ditulis akan dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih teramin. PHP dirancang untuk membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data kke halaman web.

PHP termasuk kedalam *Oven Source Product*, sehingga *Source Code* PHP dapat diubah dan didistribusikan secara bebas. PHP juga dapat berjalan pada berbagai web server seperti IIS (*Internet Information Server*), PWS (*Personal Web Server*), Apache, Xitami. PHP juga mampu lintas *platform*. Artinya PHP dapat berjlan di banyak sistem operasi yang beredar sekarang ini, diantaranya: Sistem Operasi Microsoft Windows (semua versi), Linux, Mac OS, Solaris. PHP dapat dibangun sebagai modul pada *web server* Apache dan sebagai binary yang dapat berjalan sebagai CGI (*Icommon Gateway Interface*). PHP dapat mengirim HTML header, dapat mengatur *cookies*, mengatur *authentication* dan *redirect users*.

Selain itu PHP juga memiliki keunggulan yaitu kemampuannya untuk melakukan koneksi ke berbagai macam software sistem manajemen basis data/ *Database Management System* (DBMS), sehingga dapat menciptakan suatu halam yang dinamis (Arief, M.Rudyanto ; 2011 : 43).

II.8. Pengertian MySQL

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama *MySQL AB* yang pada saat itu bernama *TcX DataKonsult AB* sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak 1979. Awalnya TcX membuat MySQL dengan tujuan mengembangkan aplikasi web untuk klien. TcX merupakan perusahaan pengembang software dan konsultan database. Saat ini MySQL sudah diakusisi oleh Oracel Corp.

MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya. Kepopuleran MySQL antara lain karena MySQL menggunak SQL sebagai bahasa dasar untuk mengakses database-nya sehingga mudah untuk digunakan, kinerja *query* cepat, dan mencukupi untuk kebutuhan database perusahaan-perusahaan skala menengah-kecil.

MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman script untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan software pengembangan aplikasi web yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman script PHP (Arief, M.Rudyanto ; 2011 : 151).

II.9. Normalisasi

Perancangan basisdata menghasilkan sekumpulan relasi yang saling berkerelasian dalam lingkup sebuah sistem. Untuk memenuhi batasan dalam defenisi basis data maka setiap rancangan relasi perlu diuji untuk menetukan apakah relasi tersebut telah optimal. Pengujian dilakukan berdasarkan kriteria tetentu. Jika relasi belum optimal maka perlu dilakukan proses normalisasi. Perwujudan normalisasi adalah dekomposisi relasi menjadi relasi-relasi baru yang sederhana.

Normalisai diartikan sebagai suatu teknik yang menstrukturkan/ mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya

permasalahan pengolahan data dalam basis data. Permasalahn yang dimaksud adala berkaitan dengan penyimpangan-penyimpangan yang terjadi akibat adanya kerangkapan data dalam relasi dan efisiensi pengolahan.

Level normalisasi atau sering disebut sebagai bentuk normal suatu relasi dijelaskan berdasarkan kriteria tertentu pada bentuk normal. Bentuk normal yang dikenal hingga saat ini meliputi bentuk 1NF, 2NF, 3NF, BCNF, 4NF, 5NF, RUNF. Masing-masing level akan dibahas berikut ini:

1. Bentuk Tidak Normal (UNF)

   Relasi-relasi yang dirancang tanpa mengindahkan batasan dalam definisi basis data dan karakteristik RDBM menghasilkan relasi UNF. Bentuk ini harus dihindari dalam perancangan relasi dalam basis data. Relasi UNF memiliki kriteria sebagai berikut :

   a. Jika relasi memiliki bentuk *non flat file* (dapat menjadi akibat data disimpan sesuai dengan kedatangannya, tidak memiliki struktur tertentu, terjadi duplikasi atau tidak lengkap).
   b. Jika relasi memuat set atribut berulang (*no single value*).
   c. Jika relasi memuat atribut *non atomic value.*

2. Bentuk Normal Petama (1NF)

   Definisi dari bentuk normal pertama adalah suatu relasi dikatakan dalam bentuk normal pertama ika dan hanya jika setiap atribut bernilai tunggal untuk setiap baris.

Relasi disebut sebagai 1NF jika memenuhi kriteria sebagai berikut :

a. Jika seluruh atribut dalam relasi bernilai atomik.
b. Jika seluruh atribut dalam relasi bernilai tunggal.
c. Jika relasi tidak memuat set atribut berulang.
d. Jika semua *record* mempunyai sejumlah atribut yang sama.

Permasalahan dalam 1NF dalah sebagai berikut :

a. Tidak dapat menyisipkan informasi parsial.
b. Terhapusnya informasi ketika menghapus sebuah *record*.
c. Pembaruan atribut nonkunci mengakibatkan sejumlah *record* harus diperbaharui.

Mengubah relasi UNF menjadi bentuk 1NF dapat dilakukan dengan cara sebagai berikut :

a. Melengkapi nilai-nilai dalam atribut.
b. Mengubah struktur relasi.

3. Bentuk Normal Kedua (2NF)

Bentuk normal kedua didefinisikan berdasarkan dependensi fungsional. Suatu relasi berada dalam bentuk 2NF jika memenuhi kriteria sebagai berikut :

a. Memenuhi kriteria 1NF.
b. Semua atribut nonkunci ketergantungan fungsional (*functionally dependence*) terhadap kunci primer (*primary key*).

Permasalahan dalam 2NF adalah sebagai berikut :

a. Kerangkapan data (*data redudancy*).

b. Pembaruan yang tidak benar dapat menimbulkan inkonsistensi data (*data inconsistency*).
c. Proses pembaruan data tidak efisien.
d. Penyimpangan pada saat penyisipan, penghapusan, dan pembaruan.

Kriteria tersebut mengindikasikan bahwa di antara atribut dalam 2NF masih mungkin mengalami ketergantungan transitif (*transitive dependency*). Selain itu, relasi 2NF menuntut telah didefinisikan atribut kunci primer dalam relasi. Mengubah relasi 1NF menjadi bentuk 2NF dapat dilakukan dengan mengubah struktur relasi dengan cara :

a. Identifikasikan ketergantungan fungsional (*functionally dependence*) relasi 1NF.
b. Berdasarkan informasi tersebut, dekomposisi relasi 1NF menadi relasi-relasi baru sesuai ketergantungan fungsionalnya (*functionally dependence*).

4. Bentuk Normal Ketiga (3NF)

Suatu relasi disebut sebagai 3NF jika memenuhi keriteria sebagai berikut :

a. Jika memenuhi kriteria 2NF.
b. Jika setiap atribut nonkunci tidak ketergantungan transitif (*non transitive dependency*) terhadap primer key (*primary key*).

Permasalahan dalam 3NF adalah keberadaan penetu yang tidak merupakan bagian dari primer key menghasilkan duplikasi rinci data pada atribut yang berfungsi sebagai ketergantungan fungsional (*functionally dependence*).

Mengubah relasi 2NF menjadi 3NF dapat dilakukan dengan mengubah struktur relasi dengan cara :

a. Identifikasikan ketergantungan transitif (*transitive dependency*) relasi 2NF.
b. Berdasarkan informasi tersebut, dekomposisi relasi 2NF menjadi relasi-relasi baru sesuai ketergantungan transitifnya (*transitive dependency*).

5. Bentuk Normal *Boyce-Codd* (BCNF)

Bentuk normal *Boyce-Codd* adalah suatu relasi yang memenuhi bentuk normal *Boyce-Codd* jika dan hanya jika semua penentu (*determinan*) adalah kunci kandidat (atribut yang bersifat unik). BCNF merupakan bentuk normal sebagai perbaikan terhadap 3NF. Suatu relasi yang memenuhi BCNF selalu memenuhi 3NF, tetapi tidak untuk sebaliknya. Suatu relasi yang memenuhi 3NF belum tentu memenuhi BCNF. Dalam banyak literatur disebutkan bahwa BCNF adalah perbaikan dari 3NF, karena bentuk normal ketiga pun mungkin masih mengandung anomali sehingga masih perlu dinormalisasi lebih lanjut.

Cara mengkonversi relasi yang telah memenuhi bentuk normal ketiga ke BCNF adalah sebagai berikut :

a. Carilah semua penentu.
b. Bila terdapat penentu yang berupa kunci kandidat maka pisahkan relasi tersebut dan buat penentu tersebut sebagai kunci primer.

6. Bentuk Normal Keempat (4NF)

Relasi disebut sebagai 4NF jika memenuhi kriteria sebagai berikut :

a. Jika memenuhi kriteria BCNF.

b. Jika setiap atribut di dalamnya tidak mengalami ketergantungan pada banyak nilai. Atau dengan kalimat lain, bahwa semua atribut yang mengalami ketergantungan pada banyak nilai adalah bergantung secara fungsional (*functionally dependence*).

7. Bentuk Normal Kelima (5NF)

   Suatu relasi memenuhi kriteria 5NF jika kerelasian antardata dalam relasi tersebut tidak dapat direkonstruksi dan struktur relasi yang sederhana (Eddy Sutanta; 2011: 174).

II.10. Unified Modeling Language (UML)

Unified Modeling Language adalah sebuah bahasa yang diterima dan digunakan oleh software developer dan software analyst sebagai suatu bahasa yang cocok untuk merepresentasikan grafik dari suatu relasi antar entitas-entitas software. UML menyediakan kumpulan alat yang sudah terstandarisasi, yang digunakan untuk mendokumentasi analisis dan perancangan sebuah sistem perangkat lunak. Peralatan utama UML adalah diagram-diagram yang digunakan untuk membantu manusia dalam memivisualisasikan proses pengembangan sebuah sistem perangkat lunak, sama seperti penggunaan denah dalam pembuatan bangunan (Jurnal ULTIMA InfoSys, Vol. IV, No. 1 ; 2013 : 37).

II.10.1. Use Case Diagram

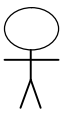
Use Case adalah deskripsi fungsi dari sebuah sistem prespektif pengguna. *Use Case* bekera dengan cara mendeskripsikan tipikal interaktif antara *user*

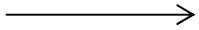
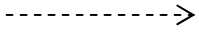
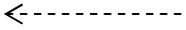
sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut sekenario.

Use Case merupakan awal yang sangat baik untuk setiap fase pengembangan berbasis objek, design, testing, dan dokumentasi yang menggambarkan kebutuhan sistem dari sudut pandang diluar sistem. Dan perlu diingat bahwa *Use Case* hanya menetapkan apa yang harusnya dikerjakan oleh sistem, yaitu kebutuhan fungsional sistem dan tidak untuk menentukan kebutuhan nonfungsional, misalnya: sasaran kinerja, bahasa pemrograman dan lain sebagainya (Jurnal Informatika Mulawarman, Vol.6 No.1 ; 2011 : 6).

Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.1. Simbol *Use Case*

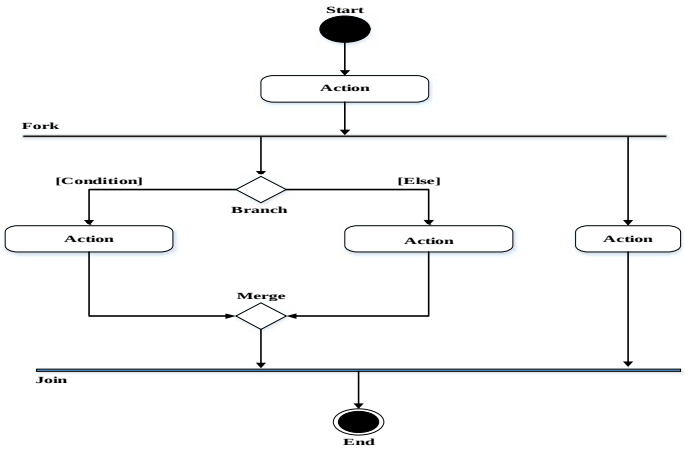
Gambar	Keterangan
	*Use case* untuk mengetahui action atau prosedur apa yang ada didalam sistem.
	Aktor adalah siapa saja yang terlibat dalam action tersebut.
	Asosiasi antara aktor dan *use case*, mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.

	Asosiasi antara aktor dan *use case* yang mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	*Include*, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain,
	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ; 2013 : 4)

II.10.2. Diagram Aktivitas (*Activity Diagram*)

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event.* Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas (Jurnal Informatika Mulawarman, Vol.6 No.1 ; 2011 : 4).



Gambar II.1 Contoh *Activity Diagram*

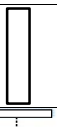
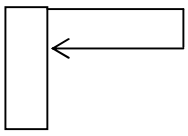
(Sumber : Jurnal ULTIMA InfoSys, Vol. IV, No. 1 ; 2013 : 39)

II.9.3. Diagram Urutan (*Sequence Diagram*)

Sequence Diagram menjelaskan interaksi objek-objek yang saling berkolaborasi, mirip dengan *activity diagram* yaitu menggambarkan alur kejadian sebuah aktivitas tetapi lebih detail dalam menggambarkan aliran data termasuk data atau behaviour yang dikirimkan atau diterima namun kurang mampu menjelaskan detail dari sebuah algoritma.

Dalam penggunaannya *Sequence Diagram* tepat untuk memperlihatkan dengan jelas bagaimana urutan kejadian atau proses karena didalamnya terlihat interaksi beberapa objek (Jurnal ULTIMA InfoSys, Vol. IV, No. 1 ; 2013 : 39)

Tabel II.2. Simbol *Sequence Diagram*

Gambar	Keterangan
	*Activation* suatu titik waktu dimana sebuah objek mulai berpartisipasi dalam sebuah *Sequence.*
	*Lifeline* menggambarkan daur hidup sebuah objek.
Participant yaitu objek yang terkait dengan sebuah urutan proses.	
Time elemen paling penting dalam *Sequence Diagram* yang konteksnya adalah urutan bukan durasi.	
Return suatu hasil kembalian sebuah operasi. Operasi mengembalikan hasil tetapi boleh tidak ditulis jika ada perbedaan dengan Getternya.	
	Message, simbol mengirim pesan antar *class*.
	*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

(Sumber : Jurnal ULTIMA InfoSys, Vol. IV, No. 1 ; 2013 : 39)

II.10.4. Diagram Kelas (*Class Diagram*)

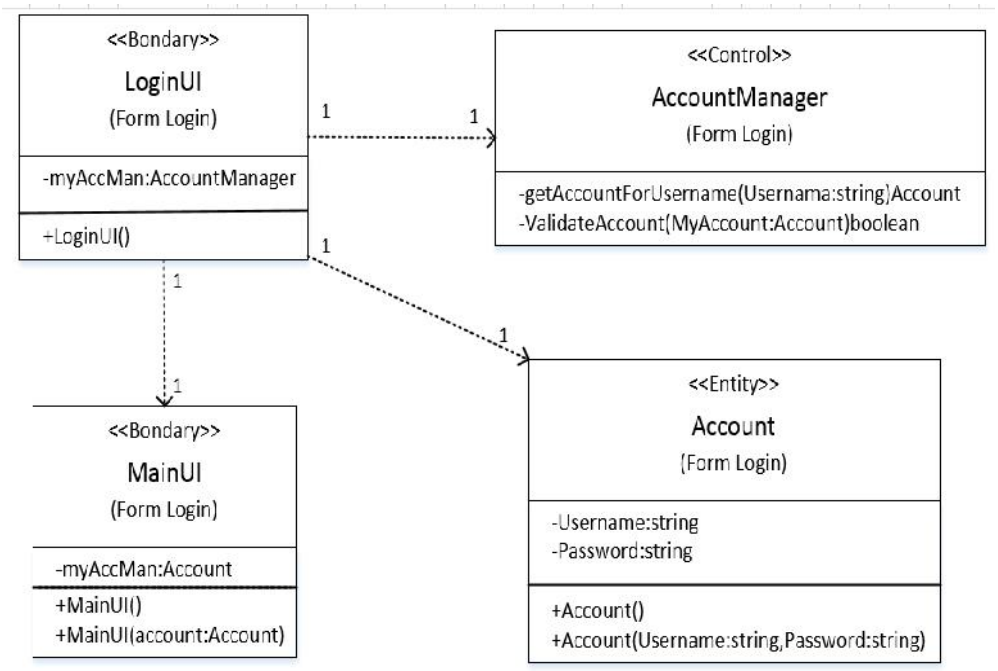
Class Diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat (Jurnal Informatika Mulawarman, Vol.6 No.1 ; 2011 : 3).

Class memiliki tiga area pokok :

a. Nama (*stereosystem*)

b. Atribut

c. Metoda



Gambar II.2 Contoh *Class Diagram*

(Sumber : Jurnal Informatika Mulawarman, Vol.6 No.1 ; 2011 : 3)