BAB II

LANDASAN TEORI

II.1. Sistem Informasi Geografis

Sistem Informasi Geografis (GIS) merupakan sistem informasi berbasis komputer yang digunakan untuk mengolah dan menyimpan data atau informasi geografis. Keunggulan utama dari sistem informasi geografis adalah memungkinkan kita untuk melihat, memahami, menanyakan, menginterpretasi dan menampilkan data spasial dalam berbagai cara, yang memperlihatkan hubungan, pola dan *trend* secara spasial, dalam bentuk peta, globe, laporan dan grafik. Sistem informasi geografis mampu membantu dalam pemecahan masalah dengan cara menampilkan data menggunakan cara yang mudah dipahami dan hasilnya mudah disebarluaskan (Bramantiyo Marjuki ; 2014 : 1).

II.1.1. Komponen Sistem Informasi Geografis

Sebagai salah satu jenis sistem informasi, sistem informasi geografis mempunyai sub sistem atau komponen yang bekerja sama untuk menghasilkan fungsionalitas sistem informasi geografis. Komponen sistem informasi geografis terdiri dari *hardware, software,* data dan metode.

1. *Hardware* atau perangkat keras merupakan media tempat pelaksanaan proses sistem informasi geografis. *Hardware* yang diperlukan dalam sebuah sistem informasi geografis meliputi perangkat keras untuk masukan data, penyimpanan data, pengolahan data dan pembuatan keluaran.

2. *Software* atau perangkat lunak merupakan alat pelaksana pekerjaan sistem informasi geografis. *Software* standar sistem informasi geografis harus mempunyai kapabilitas data input, penyimpanan, manajemen data, transformasi dan konversi data.
3. Data atau representasi dari sebuah obyek/fenomena adalah bahan yang dianalisa didalam sistem informasi geografis. Sistem informasi geografis memerlukan sebuah jenis data yang spesifik agar dapat memberikan keluaran seperti fungsionalitasnya. Data yang digunakan dalam sistem informasi geografis adalah data geospasial atau data yang berefrensi geografis (mempunyai informasi lokasi).
4. Metode adalah cara bagaimana data diolah untuk menjadi sebuah informasi. Metode meliputi aspek pemasukan data ke dalam sistem, bagaimana data dikelola dan disimpan, bagaimana data dianalisis, dan bagaimana informasi ditampilkan (Bramantiyo Marjuki ; 2014 : 2).

II.1.2. *Data Spasial*

Sebagian besar data yang akan ditangani dalam sistem informasi geografis merupakan data spasial, data yang berorientasi geografis. Data ini memiliki sistem koordinat tertentu sebagai dasar referensinya dan mempunyai dua bagian penting yang berbeda dari data lain, yaitu informasi lokasi (spasial) dan informasi deskriptif (atribut) yang dijelaskan berikut ini :

1. Informasi lokasi (spasial), berkaitan dengan suatu koordinat baik koordinat geografi (lintang dan bujur) dan koordinat XYZ, termasuk diantaranya informasi datum dan proyeksi.
2. Informasi deskriptif (atribut) atau informasi nonspasial, suatu lokasi yang memiliki beberapa keterangan yang berkaitan dengannya. Contoh jenis vegetasi, populasi, luasan, kode pos, dan sebagainya (Mohd. Ichsan ; 2012 : 51).

II.1.3. *Format Data Spasial*

Secara sederhana format dalam bahasa komputer berarti bentuk dan kode penyimpanan data yang berbeda antara file satu dengan lainnya. Dalam sistem informasi geografis, data spasial dapat direpresentasikan dalam dua format, yaitu:

a. Data vektor

   Data vektor merupakan bentuk bumi yang direpresentasikan ke dalam kumpulan garis, area (daerah yang dibatasi oleh garis yang berawal dan berakhir pada titik yang sama), titik dan nodes (titik perpotongan antara dua buah garis).

b. Data raster

   Data raster (disebut juga dengan sel grid) adalah data yang dihasilkan dari sistem penginderaan jauh. Pada data raster, obyek geografis direpresentasikan sebagai struktur sel grid yang disebut dengan *pixe*l (*picture element*) (Mohd. Ichsan ; 2012 : 51).

II.2. Pengertian Lokasi

Lokasi adalah posisi suatu tempat, benda, peristiwa, atau gejala di permukaan bumi dalam hubungannya dengan tempat, benda dan peristiwa lain. Terdapat dua komponen lokasi, yaitu arah dan jarak. Arah menunjukkan posisi suatu tempat jika dibandingkan dengan tempat dimana orang tersebut berada. Adapun jarak adalah ukuran jauh atau dekatnya dua benda atau tersebut. Contoh, Bandung terletak di sebelah selatan Jakarta, arah tersebut akan berbeda jika orang yang bertanya berada di Semarang, Bandung terletak disebelah barat.

Ada dua macam lokasi yaitu absolut dan lokasi relatif. Lokasi absolut adalah posisi sesuatu berdasarkan koordinat garis lintang dan garis bujur. Misalnya, Indonesia terletak diantara 6° LU-11° LS dan antara 95° BT-141° BT. Lokasi absolut mutlak adanya dan dapat dipercaya karena masa daratan relatif tetap, aspek perubahannya kecil sekali, dan berlaku umum di seluruh dunia. Melalui lokasi absolut, seseorang dapat mengetahui jarak dan arah suatu tempat ke tempat lain di permukaan bumi. Dengan bantuan garis lintang seseorang dapat menggambarkan kondisi iklim suatu daerah, berarti vegetasinya bersifat heterogen, selalu menghijau, kehidupan hewannya beragam, penduduknya termasuk ras mongoloid, sebagian besar penduduknya hidup dalam bidang pertanian, dan ciri-ciri lainnya. Dengan mengetahui lokasi suatu tempat berdasarkan garis lintang, seseorang akan memperoleh gambaran tentang kondisi iklim, kehidupan tumbuhan, hewan dan manusianya (Hartono ; 2008 : 9).

II.3. Quantum GIS

Quantum GIS (QGIS) adalah *cross-platform* perangkat lunak bebas (*open 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, Quantum GIS sebagai perangkat lunak bebas aplikasi dibawah GPL (*General Public License*), dapat secara bebas dimodifikasi untuk melakukan tugas yang berbeda atau lebih khusus. Quantum GIS memungkinkan penggunaan *shapefiles*, pertanggungan, dan Geo*database*s pribadi. MapInfo , PostGIS , dan beberapa format lain yang didukung di Quantum GIS (Adam Suseno ; 2012 ; 15).

II.4. Pengertian PHP

PHP merupakan suatu bahasa pemrograman sisi server yang dapat anda gunakan untuk membuat halaman Web dinamis. Contoh bahasa yang lain adalah *Microsoft Active Server Page (*ASP) dan *Java Server Page*(JSP). Dalam suatu halaman HTML anda dapat menanamkan kode PHP yang akan dieksekusi setiap kali halaman tersebut dikunjungi. Karena kekayaannya akan fitur yang mempermudah perancangan dan pemrograman Web, PHP memiliki pepularitas yang tinggi. Anda dapat mengecek *survey* popularitas yang dilakukan *netcraft* di URL www.php.net/usage.php.

PHP adalah kependekan dari *HyperTextPrepocessor* (suatu akronim rekursif) yang dibangun oleh RasmusLerdorf pada tahun 1994. Dahulu, pada awal

pengembangnnya PHP disebut sebagai kependekan dari *Personal Home Page.* PHP merupakan produk *OpenSource* sehingga anda dapat mengakses *source code*, menggunakan dan mengubahnya tanpa harus membayar sepeser pun. Gratis (Antonius Nugraha Widhi Pratama ; 2010 : 9).

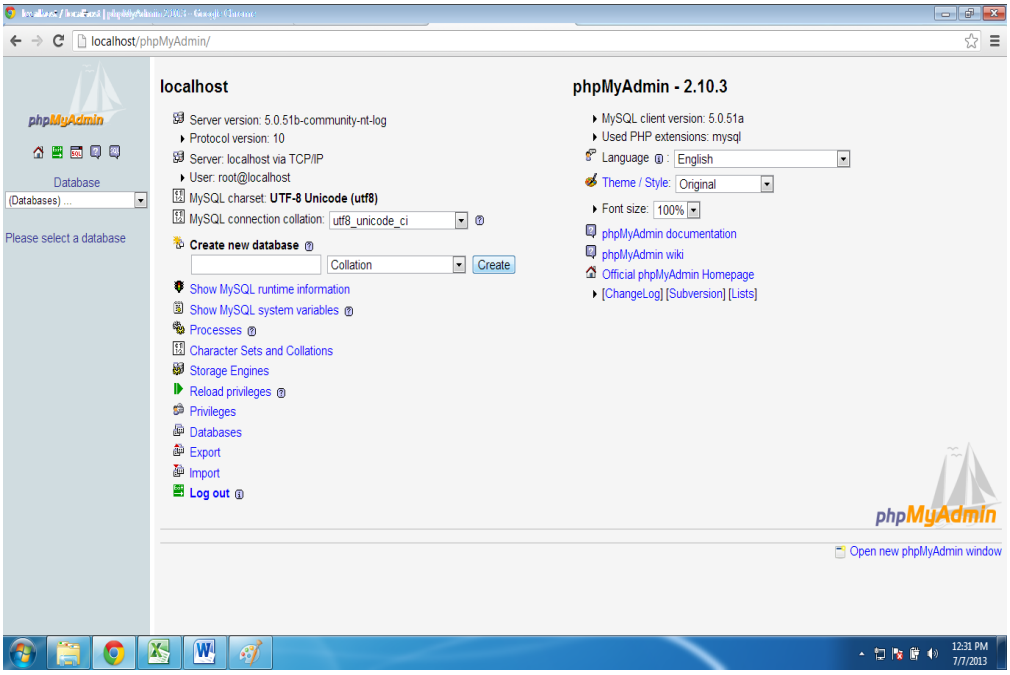
II.5. Pengertian *Database*

Database atau basis data merupakan sebuah media penyimpanan yang mutlak digunakan dalam membangun sebuah sistem. *Database* memudahkan para *user* / pengguna aplikasi dalam mengolah sebuah data yang sangat banyak. Pengolahan *database* dapat berupa penambahan data, pengeditan data dan penghapusan data. Pengolahan *database* atau suatu organisasi informasi yang sangat penting bagi instansi atau suatu organisasi berupa laporan, sehingga dapat digunakan sebagai pertimbangan dalam mengambil keputusan (Wahana Komputer ; 2010 : 5).

II.6. Pengertian MySQL

MySQL adalah suatu sistem manajemen basis data relasional (RDBMS-*Relational Database Management System*) yang mampu bekerja dengan cepat, kokoh, dan mudah digunakan. Contoh RDBMS lain adalah *Oracle, Sybase*. Basis data memungkinkan anda untuk menyimpan, menelusuri, menurutkan dan mengambil data secara efesien. *Server* MySQL yang akan membantu melakukan fungsionaliitas tersebut. Bahasa yang digunakan oleh MySQL tentu saja adalah SQL-*standar* bahasa basis data relasional di seluruh dunia saat ini.

MySQL dikembangkan, dipasarkan oleh sebuah perusahaan Swedia bernama MySQL AB. RDBMS ini berada dibawah bendera GNU GPL sehingga termasuk produk *Open Source* dan sekaligus memiliki lisensi komersial. Apabila menggunakan MySQL sebagai basis data dalam suatu situs Web. (Antonius Nugraha Widhi Pratama ; 2010 : 10).



Gambar II.1. Tampilan MySQL
(Sumber : Antonius Nugraha Widhi Pratama ; 2010 : 10)

II.7. Teknik Normalisasi

Salah satu topik yang cukup kompleks dalam dunia manajemen *database* adalah proses untuk menormalisasi tabel-tabel dalam *database relasional.* Dengan normalisasi kita ingin mendesain *database relasional* yang terdiri dari tabel-tabel berikut :

1. Berisi data yang diperlukan.
2. Memiliki sesedikit mungkin redudansi.
3. Mengakomodasi banyak nilai untuk tipe data yang diperlukan.
4. Mengefisienkan update.
5. Menghindari kemungkinan kehilangan data secara tidak disengaja/tidak diketahui.

Alasan utama dari normalisasi *database* minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya "*insertion anomalies"*, "*deletion anomalies",* dan "*update anomalies".* Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada *database* yang tidak normal.

1. Bentuk Tidak Normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

2. Bentuk Normal Tahap Pertama (1" *Normal Form*)

Sebuah table disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut.
- Masing-masing *cell* bernilai tunggal

3. Bentuk Normal Tahap Kedua (2nd *normal form*)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada *primary key* secara utuh.

4. Bentuk Normal Tahap Ketiga (3rd *normal form*)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi X -> A, dimana A mewakili semua atribut tunggal didalam tabel yang tidak ada didalam X, maka :

- X haruslah *superkey* pada tabel tersebut.
- A merupakan bagian dari *primary key* pada tabel tersebut.

5. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

6. Boyce Code Normal Form (BCNF)
   a. Memenuhi 1st NF
   b. Relasi harus bergantung fungsi pada atribut superkey (Kusrini ; 2010 : 39-43).

II.8. UML (*Unified Modeling Language*)

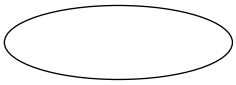
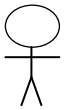
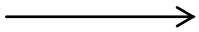
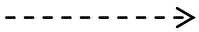
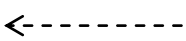
Menurut Evi Triandini (2012) UML menyediakan beberapa notasi dan artifak standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. Artifak didalam UML didefinisikan sebagai

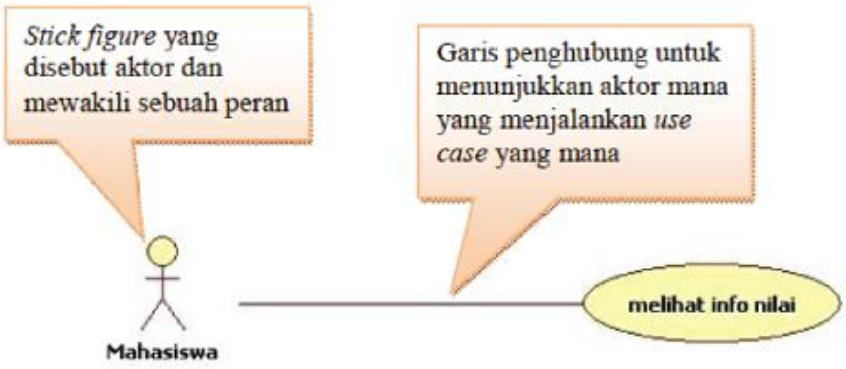
informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak. Contohnya adalah *source code* yang dihasilkan oleh proses pemrograman. Yang perlu diperhatikan untuk menjaga konsistensi antar artifak selama proses analisis dan desain adalah bahwa setiap perubahan yang terjadi pada satu artifak harus juga dilakukan pada artifak sebelumnya.

II.8.1. *Use Case* Diagram

Use case adalah sebuah kegiatan yang dilakukan oleh sistem, biasanya dalam menanggapi permintaan dari pengguna sistem.

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukan pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki control terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	*Include*, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.



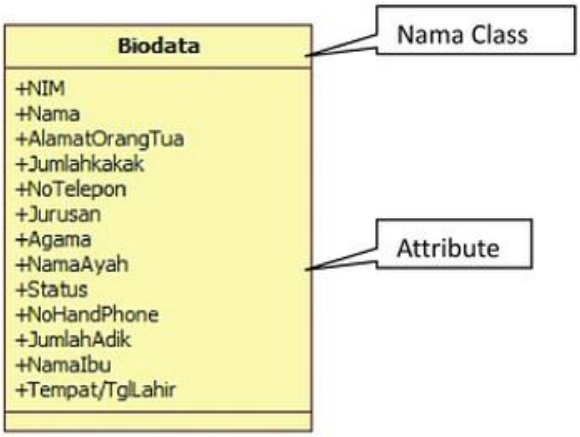
Gambar II.2. Notasi *Use Case* Diagram
(Sumber : Evi triandini ; 2012 : 17)

II.8.2. *Class* Diagram

Class diagram bertujuan untuk mendokumentasikan dan menggambarkan kelas-kelas dalam pemrograman yang nantinya akan dibangun.

Tabel II.2. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

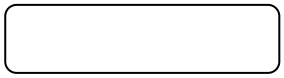
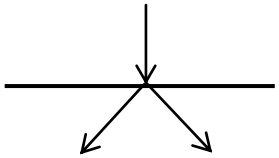


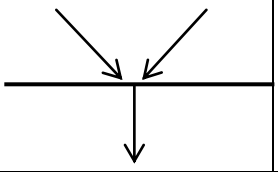
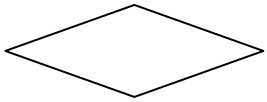
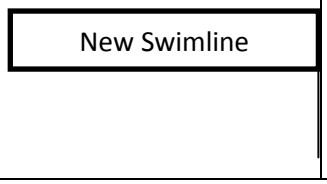
Gambar II.3. Notasi *Class* Diagram
(Sumber : Evi triandini ; 2012 : 49)

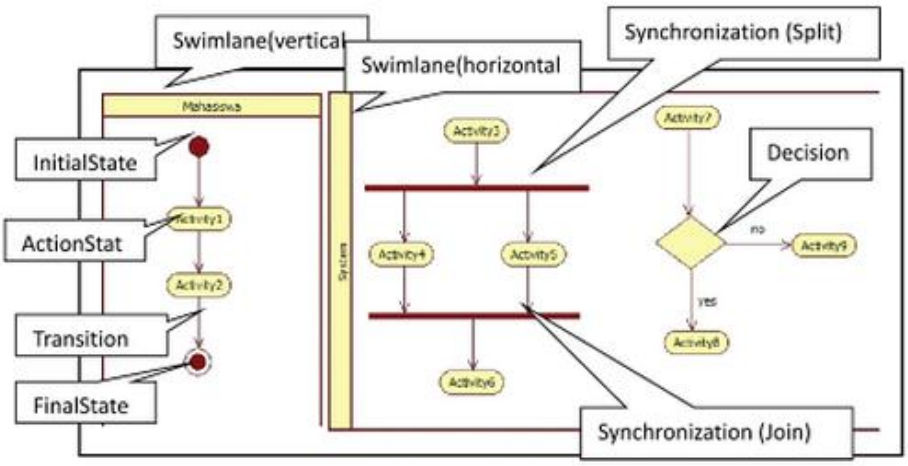
II.8.3. *Activity* Diagram

Activity diagram adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut.

Tabel II.3. Simbol *Activity Diagram*

Gambar	Keterangan
	Start point, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	End point, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true, false.*
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.



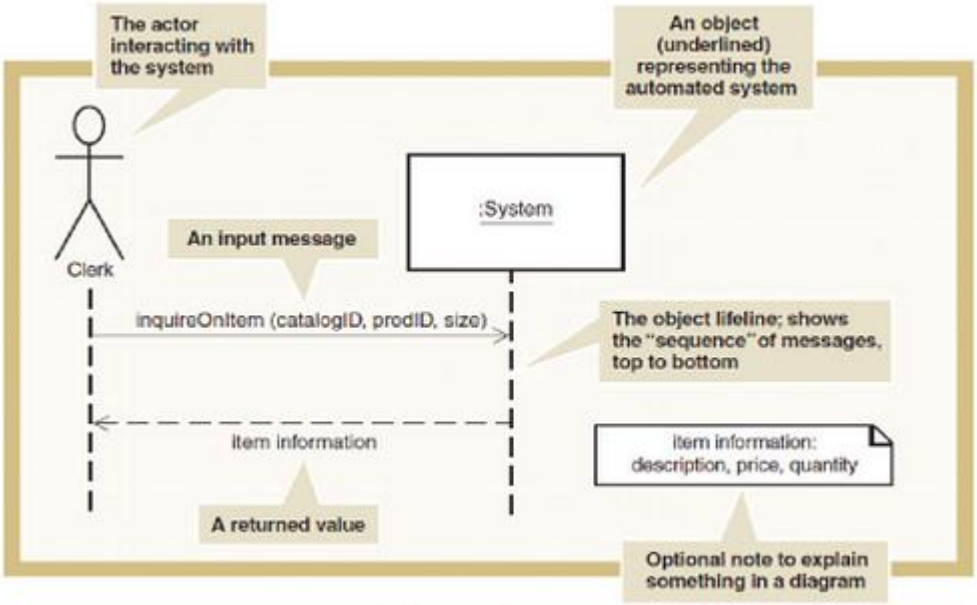
Gambar II.4. Notasi *Activity* Diagram
(Sumber : Evi triandini ; 2012 : 37)

II.8.4. *Sequence* Diagram

Sequence diagram adalah diagram yang digunakan untuk mendefinisikan *input* dan *output* serta urutan interaksi antara pengguna dan sistem untuk sebuah *usecase*.

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	Entity Class, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	Boundary Class, berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan *form* cetak.
	Control class, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	Message, simbol mengirim pesan antar *class*.
	Recursive, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	Activation, *activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.



Gambar II.5. Notasi *Sequence* Diagram
(Sumber : Evi triandini ; 2012 : 71)