

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Sistem merupakan kumpulan *elemen* yang saling berkaitan yang bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*). (Kusrini; 2009 : 11).

##### **II.1.1. Data**

Data merupakan representasi dari fakta atau gambaran mengenai suatu objek atau kejadian, ambil contoh fakta mengenai biodata Mahasiswa yang meliputi nama, alamat, jenis kelamin, agama yang dianut, dan lain-lain. Contoh lain dari fakta mengenai kejadian / transaksi dalam sebuah perusahaan dagang adalah seperti transaksi penjualan yang meliputi waktu transaksi, pelaku transaksinya (pelanggan, kasir), barang yang ditransaksikan, serta jumlah dan harganya. Data dinyatakan dengan nilai yang berbentuk angka, deretan karakter, atau simbol. (Kusrini; 2009 : 3).

##### **II.1.2. Informasi**

Informasi merupakan hasil olahan data, di mana data tersebut sudah diproses dan diinterpretasikan menjadi sesuatu yang bermakna untuk pengambilan keputusan. Informasi juga diartikan sebagai himpunan dari data yang relevan dengan satu atau beberapa orang dalam suatu waktu.

Suatu informasi berguna bagi Pembuat keputusan karena informasi bisa menurunkan ketidakpastian (meningkatkan pengetahuan) tentang hal yang sedang

dipikirkan. Makna dari sebuah informasi tentu berbeda-beda antara seorang dengan lainnya, tergantung pada tingkat kepentingannya, misalnya informasi daftar pelanggan yang potensial akan sangat dibutuhkan oleh bagian marketing di suatu perusahaan guna meningkatkan penjualan produk, tetapi barangkali tidak akan menjadi perhatian dibagian personalia.

Kegunaan informasi bagi seseorang juga sangat tergantung pada waktu. Pada suatu waktu tertentu informasi tersebut mungkin sangat diperlukan dilain hari, mungkin saja hal tersebut sudah tidak berguna sama sekali. Contohnya, informasi perbandingan harga barang akan sangat dibutuhkan oleh seseorang yang akan membeli barang tersebut. Namun saat ini dia sedang tidak mempertimbangkan untuk membeli barang tersebut, informasi tersebut menjadi kurang bermakna. (Kusrini; 2009 : 4).

### **II.1.3. Kualitas Informasi**

Agar bisa menyediakan keluaran yang berguna untuk membantu *manager* atau para pengambil keputusan, sebuah sistem informasi harus mampu mengumpulkan data dan mentransformasikan data tersebut kedalam informasi yang memiliki kualitas-kualitas tersebut. (Kusrini; 2009 : 4).

Berikut karakteristik informasi yang berkualitas :

- 1. Relevan.** Informasi yang disajikan sebaiknya terkait dengan keputusan yang akan diambil oleh pengguna informasi tersebut. Misalnya, seorang *manager* yang akan memberikan kredit kepada pelanggan bisa melihat laporan keuangan pelanggan tersebut karena laporan tersebut terkait

dengan keputusan yang akan dibuat, yaitu memberikan atau tidak memberikan kredit kepada pelanggan tersebut.

2. **Akurat.** Kecocok antara informasi dengan kejadian-kejadian atau objek-objek yang diwakilinya. Misalnya, laporan inventaris yang tidak akurat menyebutkan bahwa terdapat 15 unit barang yang tersisadi gudang. Kenyataanya, masih ada 51unit barang di dalam gudang.
3. **Lengkap.** Merupakan derajat sampai seberapa jauh informasi menyertakan kejadian-kejadian atau objek-objek yang berhubungan. Misalnya, penjualan selama satu hari yang seharusnya ada 150 transaksi di laporan hanya tercatat sebanyak 145 transaksi.
4. **Tepat waktu.** Informasi yang tidak tepat waktu akan menjadi informasi yang tidak berguna atau tidak dapat di gunakan untuk membantu pengambilan keputusan. Misalnya, informasi jadwal ujian seorang mahasiswa disampaikan setelah kegiatan ujian diselenggarakan. Informasi ini menjadi tidak berguna lagi.
5. **Dapat dipahami.** Hal tersebut terkait dengan bahasa dan cara penyajian informasi agar pengguna lebih mudah mengambil keputusan.
6. **Dapat dibandingkan.** Sebuah informasi yang memungkinkan seorang pemakai untuk mengidentifikasi persamaan dan perbedaan antara dua objek atau kejadian yang mirip. Misalnya, membandingkan laporan pendapatan antara tahun 2006 dan 2007.

#### **II.1.4. Sistem Informasi**

Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan strategi dari suatu organisasi, serta menyediakan laporan-laporan yang diperlukan untuk pihak luar.

Berdasarkan dukungan kepada pemakainya, sistem informasi dibagi menjadi beberapa :

1. Sistem Pemrosesan Transaksi (*Transaction Processing System*) atau *TPS*.
2. Sistem Informasi Manajemen (*Management Information System*) atau *MIS*.
3. Sistem Otomasi Perkantoran (*Office Automation System / OAS*).
4. Sistem Pendukung Keputusan (*Decision Support System*) atau *DSS*.
5. Sistem Informasi Eksekutif (*Executive Information System*) atau *EIS*.
6. Sistem Pendukung Kelompok (*Group Support System*) atau *GSS*.
7. Sistem Pendukung Cerdas (*Inteleigent Support System*) atau *ISS*.

Mengingat bahwa *EIS*, *DSS*, dan *MIS* digunakan untuk pendukung manajemen, maka ketiga sistem tersebut sering disebut Sistem Pendukung Manajemen (*management support system*) atau *MSS*. (Kusrini; 2009 : 11).

#### **II.2. Sistem Informasi Geografis**

Sistem informasi yang memiliki kepaduan antara teknologi informasi dan aktifitas dari orang yang menggunakan teknologi itu untuk mengembangkan dan mengaplikasikan dalam mendukung sebuah operasi atau manajemen di bidang

geografis, merupakan bagian dari perkembangan di ilmu sistem informasi geografis.

Dalam pengembangannya, sistem informasi ini dibuat dengan tujuan memanfaatkan teknologi informasi. Hal ini tidak terlepas dari semakin banyaknya *software* yang dibuat untuk membantu dalam pengerjaannya khususnya dalam sistem informasi geografis. Sesuatu yang berhubungan dengan sistem informasi tentunya tidak terlepas dari hubungan dengan sistem data dan aktivitas lain dalam penggunaan *software* nya. Oleh karena itu dibutuhkan sebuah *software* yang mendukung dalam sistem informasi geografis. (Adam Suseno & Ricky Agus; 2012 : 1).

Sistem Informasi Geografis atau disingkat SIG dalam bahasa Inggris *Geographic Information System* (disingkat GIS) merupakan sistem informasi khusus yang mengelola data yang memiliki informasi *spasial* (bereferensi keruangan). Atau dalam arti yang lebih sempit adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis atau data geospasial untuk mendukung pengambilan keputusan dalam perencanaan dan pengelolaan suatu wilayah, misalnya data yang diidentifikasi menurut lokasinya, dalam sebuah *database*. Para praktisi juga memasukkan orang yang membangun dan mengoperasikannya dan data sebagai bagian dari sistem ini.

Teknologi Sistem Informasi Geografis dapat digunakan untuk investigasi ilmiah, pengelolaan sumber daya, perencanaan pembangunan, kartografi dan perencanaan rute. Misalnya, SIG bisa membantu perencana untuk secara cepat

menghitung waktu tanggap darurat saat terjadi bencana alam, atau SIG dapat digunakan untuk mencari lahan basah (*wetlands*) yang membutuhkan perlindungan dari polusi atau dapat digunakan mencari informasi sebuah tempat khusus dan banyak manfaat lain yang dapat dikembangkan dalam sistem informasi geografis ini. (Adam Suseno & Ricky Agus; 2012 : 2).

### II.2.1. Komponen Sistem Informasi Geografis

Komponen-komponen pendukung SIG terdiri dari lima komponen yang bekerja secara terintegrasi yaitu perangkat keras (*hardware*), perangkat lunak (*software*), data, manusia, dan *metode* yang dapat di lihat pada gambar II.1 sebagai berikut :



**Gambar II.1 : Komponen Sistem Informasi Geografis**

Sumber : Adam Suseno & Ricky Agus (2012 : 6)

#### 1. Perangkat Keras (*Hardware*)

Perangkat keras SIG adalah perangkat-perangkat fisik yang merupakan bagian dari sistem komputer yang mendukung analisis geografi dan pemetaan. Perangkat keras SIG mempunyai kemampuan untuk menyajikan citra dengan resolusi dan kecepatan yang tinggi serta

mendukung operasi-operasi basis data dengan *volume* data yang besar secara cepat. Perangkat keras SIG terdiri dari beberapa bagian untuk *menginput* data, mengolah data, dan mencetak hasil proses. Berikut ini pembagian berdasarkan proses :

- a. *Input Data : Mouse, Digitizier, Scanner*
- b. *Olah Data : Harddisk, Processor, RAM, VGA Card*
- c. *Output Data : Plotter, Printer, Screening*

## **2. Perangkat Lunak (Software)**

Perangkat lunak digunakan untuk melakukan proses menyimpan, menganalisa, memvisualkan data-data baik data spasial maupun non-spasial. Perangkat lunak yang harus terdapat dalam komponen *software* SIG adalah :

- a. Alat untuk memasukkan dan memanipulasi data SIG.
- b. *Data Base Management System (DBMS)*.
- c. Alat untuk menganalisa data-data.
- d. Alat untuk menampilkan data dan hasil analisa.

## **3. Data**

Pada prinsipnya terdapat dua jenis data untuk mendukung SIG yaitu :

- a. *Data Spasial*

Data spasial adalah gambaran nyata suatu wilayah yang terdapat di permukaan bumi. Umumnya direpresentasikan berupa grafik, peta, gambar dengan *format digital* dan disimpan dalam bentuk koordinat

$x,y$  (*vektor*) atau dalam bentuk image (*raster*) yang memiliki nilai tertentu.

b. *Data Non Spasial (Atribut)*

*Data non spasial* adalah data berbentuk tabel dimana tabel tersebut berisi informasi- informasi yang dimiliki oleh objek dalam data *spasial*. Data tersebut berbentuk data tabular yang saling terintegrasi dengan data *spasial* yang ada.

#### **4. Manusia**

Manusia merupakan inti elemen dari SIG karena manusia adalah perencana dan pengguna dari SIG. Pengguna SIG mempunyai tingkatan seperti pada sistem informasi lainnya, dari tingkat spesialis teknis yang mendesain dan mengelola sistem sampai pada pengguna yang menggunakan SIG untuk membantu pekerjaannya sehari-hari.

#### **5. Metode**

Metode yang digunakan dalam SIG akan berbeda untuk setiap permasalahan. SIG yang baik tergantung pada aspek desain dan aspek realnya.

### **II.2.2. Manfaat Sistem Informasi Geografis**

Adapun beberapa manfaat yang dimiliki Sistem informasi geografis memiliki manfaat di berbagai bidang seperti :

#### **1. Manajemen Tata Guna Lahan**

Pemanfaatan dan penggunaan lahan merupakan bagian kajian geografi yang perlu dilakukan dengan penuh pertimbangan dari berbagai segi.



Tujuannya adalah untuk menentukan zonifikasi lahan yang sesuai dengan karakteristik lahan yang ada. Misalnya, wilayah pemanfaatan lahan di kota biasanya dibagi menjadi daerah pemukiman, industri, perdagangan, perkantoran, fasilitas umum, dan jalur hijau. SIG dapat membantu pembuatan perencanaan masing-masing wilayah tersebut dan hasilnya dapat digunakan sebagai acuan untuk pembangunan *utilitas-utilitas* yang diperlukan.

Lokasi dari *utilitas-utilitas* yang akan dibangun di daerah perkotaan (*urban*) perlu dipertimbangkan agar *efektif* dan tidak melanggar kriteria-kriteria tertentu yang bisa menyebabkan ketidak selarasan. Contohnya, pembangunan tempat sampah. Kriteria-kriteria yang bisa dijadikan parameter antara lain : di luar area pemukiman, berada dalam radius 10 meter dari genangan air, berjarak 5 meter dari jalan raya dan sebagainya.

Dengan kemampuan SIG yang bisa memetakan apa yang ada di luar dan di dalam suatu area, kriteria-kriteria ini nanti digabungkan sehingga memunculkan irisan daerah yang tidak sesuai, agak sesuai, dan sangat sesuai dengan seluruh kriteria. Di daerah pedesaan (*rural*) manajemen tata guna lahan lebih banyak mengarah ke sektor pertanian. Dengan terpetakannya curah hujan, iklim, kondisitaneh, ketinggian dan keadaan alam, akan membantu penentuan lokasi tanaman, pupuk yang dipakai, dan bagaimana proses pengolahan lahannya. Pembangunan saluran irigasi agar dapat merata dan minimal biayanya dapat dibantu dengan peta sawah ladang, peta pemukiman penduduk, ketinggian masing-masing tempat dan peta kondisi tanah.

Penentuan lokasi gudang dan pemasaran hasil pertanian dapat terbantu dengan memanfaatkan peta produksi pangan, penyebaran konsumen dan peta jaringan transportasi. Selain untuk manajemen pemanfaatan lahan, SIG juga dapat membantu dalam hal penataan ruang. Tujuannya adalah agar penentuan pola pemanfaatan ruang disesuaikan dengan kondisi fisik dan sosial yang ada, sehingga lebih *efektif* dan *efisien*. Misalnya penataan ruang perkotaan, pedesaan, permukiman, kawasan industri dan lainnya. (Adam Suseno & Ricky Agus; 2012 : 10).

## **2. Inventarisasi Sumber Daya Alam**

Secara sederhana manfaat SIG dalam data kekayaan sumber daya alam ialah sebagai berikut:

- a. Untuk mengetahui persebaran berbagai sumber daya alam, misalnya minyak bumi, batubara, emas, besi dan barang tambang lainnya.
- b. Untuk mengetahui persebaran kawasan lahan, misalnya :
  - 1) Kawasan lahan potensial dan lahan kritis.
  - 2) Kawasan hutan yang masih baik dan hutan rusak.
  - 3) Kawasan lahan pertanian dan perkebunan.
  - 4) Pemanfaatan perubahan penggunaan lahan.
  - 5) Rehabilitasi dan konservasi lahan.

## **3. Pengawasan Daerah Bencana Alam**

Kemampuan SIG untuk pengawasan daerah bencana alam, misalnya :

- a. Memantau luas wilayah bencana alam.
- b. Pencegahan terjadinya bencana alam pada masa datang.

- c. Menyusun rencana-rencana pembangunan kembali daerah bencana.
- d. Penentuan tingkat bahaya erosi.
- e. Prediksi ketinggian banjir.
- f. Prediksi tingkat kekeringan.

#### **4. Perencanaan Wilayah dan Kota**

Kemampuan SIG dalam perencanaan wilayah dan kota seperti :

- a. Untuk bidang sumber daya, seperti kesesuaian lahan pemukiman, pertanian, perkebunan, tata guna lahan, pertambangan dan energi, analisis daerah rawan bencana.
- b. Untuk bidang perencanaan ruang, seperti perencanaan tata ruang wilayah, perencanaan kawasan industri, pasar, kawasan permukiman, penataan sistem dan status pertahanan.
- c. Untuk bidang manajemen atau sarana-prasarana suatu wilayah, seperti manajemen sistem informasi jaringan air bersih, perencanaan dan perluasan jaringan listrik.
- d. Untuk bidang pariwisata, seperti inventarisasi pariwisata dan analisis potensi pariwisata suatu daerah.
- e. Untuk bidang transportasi, seperti inventarisasi jaringan transportasi publik, kesesuaian rute alternatif, perencanaan perluasan sistem jaringan jalan, analisis kawasan rawan kemacetan dan kecelakaan.
- f. Untuk bidang sosial dan budaya, seperti untuk mengetahui luas dan persebaran penduduk suatu wilayah, mengetahui luas dan persebaran lahan pertanian serta kemungkinan pola drainasenya, pendataan dan

pengembangan pusat-pusat pertumbuhan dan pembangunan pada suatu kawasan, pendataan dan pengembangan pemukiman penduduk, kawasan industri, sekolah, rumah sakit, sarana hiburan dan perkantoran. (Adam Suseno & Ricky Agus; 2012 : 13).

### **II.3. MapServer**

*MapServer* merupakan aplikasi *freeware* dan *open source* yang memungkinkan kita menampilkan data *spasial* (peta) di *web*. Aplikasi ini pertama kali dikembangkan di Universitas Minesotta, Amerika Serikat untuk proyek *ForNet* (sebuah proyek manajemen sumber daya alam) yang disponsori oleh NASA. Pengembangan *MapServer* menggunakan berbagai aplikasi *open source* atau *freeware* seperti *Shapelib* (<http://shapelib.maptools.org>) untuk baca/tulis format data *Shapefile*, *FreeType* untuk membuat karakter, GDAL/OGR untuk baca/tulis berbagai *format* data *vektor* maupun *raster*, dan *Proj4* untuk menangani berbagai proyeksi peta. (Ruslan Nuryadi; 2005 : 3).

Dalam buku Panduan Menggunakan *MapServer* yang ditulis diberitahukan beberapa fitur *MapServer*, yaitu :

- a. Menampilkan data *spasial* dalam *format vektor* seperti *Shapefile*, *Arc View*, *PHP Mapscript* dan berbagai *format* data *vektor* lain dengan menggunakan *Library OGR*.
- b. Menampilkan data *spasial* dalam *format raster* seperti *TIFF/GeoTIFF*, *EPPL7* dan berbagai *format* data *raster* lain dengan menggunakan *library GDAL*.

- c. Menggunakan *quadtree* dalam *indexing* data spasial, sehingga operasi-operasi spasial dapat dilakukan dengan cepat.
- d. Dapat dikembangkan (*customizable*) dengan tampilan keluaran yang dapat diatur menggunakan *file-file template*.
- e. Dapat melakukan seleksi objek berdasar nilai, atau berdasar titik area, atau berdasar sebuah objek *spasial* tertentu.
- f. Mendukung rendering karakter berupa *Font TrueType*.
- g. Mendukung penggunaan data *raster* maupun vektor yang dibagi-bagi (*tiled*) menjadi sub bagian yang lebih kecil sehingga proses untuk mengambil dan menampilkan gambar dapat dipercepat.
- h. Dapat menggambarkan *elemen* peta (skala grafis, peta indeks dan legenda peta) secara otomatis.
- i. Dapat menggambarkan peta tematik yang dibangun menggunakan ekspresi logika maupun ekspresi *reguler*.
- j. Konfigurasi dapat diatur secara langsung melalui parameter yang ditentukan pada *URL*.
- k. Dapat menangani berbagai sistem proyeksi secara langsung.
- l. Saat ini, selain dapat mengakses *MapServer* sebagai program CGI, kita dapat mengakses *MapServer* sebagai modul *MapScript*, melalui berbagai bahasa skrip seperti *PHP*, *Perl*, *Python* atau *Java*. Akses fungsi-fungsi *MapServer* melalui skrip akan memudahkan dalam pengembangan karena pengembang dapat memilih bahasa yang familiar.

#### **II.4. PHP**

*PHP* dibuat pertama kali oleh seorang perancang perangkat lunak (*software engineering*) yang bernama Rasmus Lerdoff. Rasmus Lerdoff membuat halaman *web PHP* pertamanya pada tahun 1994. *PHP4* dengan versi-versi akhir menuju *PHP5* sudah mendukung pemrograman berorientasi objek. *PHP* merupakan bahasa pemrograman yang digunakan untuk pemrograman *web*. *PHP* singkatan dari *hypertext preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *web* yang disisipkan pada dokumen *HTML*. Penggunaan *PHP* memungkinkan *web* dapat dibuat dinamis sehingga *maintenance* situs *web* tersebut lebih mudah dan efisien. *PHP* merupakan *software open-source* yang disebar dan dilisensikan secara gratis dan dapat *download* dari situs resminya yaitu : <http://www.php.net>. (Rosa & M. Shalahuddin; 2011 : 85).

#### **II.5. Database**

Istilah *database* banyak memiliki definisi. Untuk sebagian kalangan sederhana *database* diartikan sebagai kumpulan data (buku, nomor telepon, daftar pegawai, dan lain sebagainya). Ada juga yang menyebut *database* dengan definisi lain yang lebih formal dan tegas. *Database* di definisikan sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil dan dicari secara cepat.

Selain berisi data, *database* juga berisi *metadata*. *Metadata* adalah data yang menjelaskan tentang struktur dari data itu sendiri. Sebagai contoh, Anda

dapat memperoleh informasi tentang nama-nama kolom dan tipe yang ditampilkan tersebut disebut *metadata*. Menurut (Budi Raharjo; 2011 : 3 ).

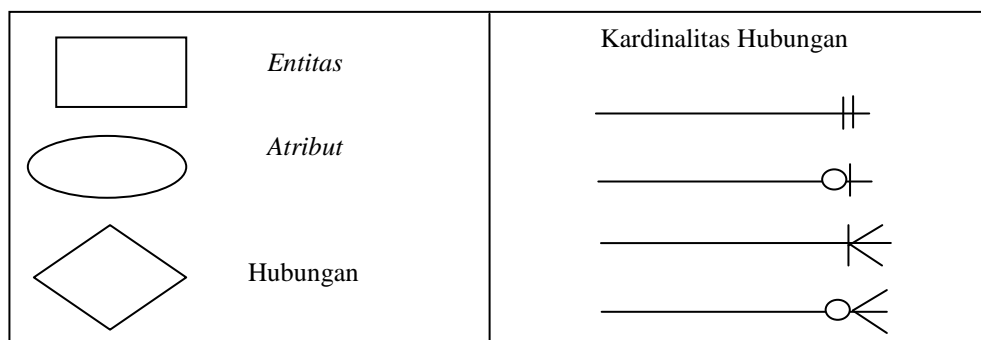
### **II.5.1. Pemodelan Data**

Pada Perancangan konseptual diperlukan suatu pendekatan yang digunakan untuk menggambarkan hubungan antar data. Hubungan tersebut dapat dinyatakan dalam bentuk E-R, karena model E-R adalah dasar penting dalam merancang *database* maka akan dijelaskan tentang gambaran tentang model E-R, penjelasan mengenai komponen-komponen yang menyusun model E-R, hingga cara penyusunan model E-R. (Abdul Kadir; 2009 : 30).

### **II.5.2. Model E-R**

Model E-R adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk *entitas*, *atribut* dan hubungan antar *entitas*. Huruf E sendiri menyatakan *entitas* dan R menyatakan hubungan (dari kata *Relationship*). Model ini dinyatakan dalam bentuk diagram, itulah sebabnya model E-R sering disebut sebagai diagram E-R. (Abdul Kadir; 2009 : 30).

Model E-R melibatkan sejumlah notasi, beberapa notasi dasar dalam model E-R ditunjukkan pada gambar II.2, notasi-notasi tersebut diberikan hanya untuk memberikan suatu pengetahuan dasar.



**Gambar II.2 : Sejumlah Notasi Pada Model E-R**

Sumber : Abdul Kadir (2009 : 31)

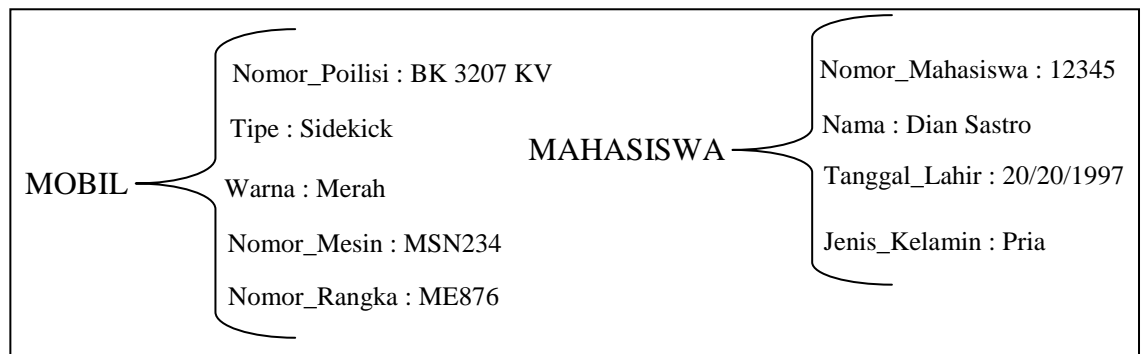
1. *Entitas*

Yang dimaksud dengan *entitas* adalah sesuatu dalam dunia nyata yang keberadaannya tidak bergantung pada yang lain. Sebagai contoh, setiap pegawai dalam sebuah organisasi adalah sebuah *entitas*. *Entitas* dapat berupa sesuat yang nyata ataupun abstrak (berupa suatu konsep). Secara lebih rinci dijelaskan bahwa *entitas* dapat berupa seseorang, sebuah tempat, sebuah objek, sebuah kejadian atau suatu konsep.

2. *Atribut*

Setiap *entitas* dinyatakan dalam sejumlah atribut. *Atribut* adalah properti atau karakteristik yang terdapat pada setiap *entitas*. Sebagai contoh, pada gambar II.3, terdapat *entitas* MOBIL yang mengandung atribut Nomor\_Polisi, Tipe, Warna, Nomor\_Rangka dan Nomor\_Mesin. Selain itu terdapat *entitas* MAHASISWA yang mengandung *atribut* Nomor\_Mahasiswa, Nama, Tanggal\_Lahir, dan Jenis\_Kelamin.



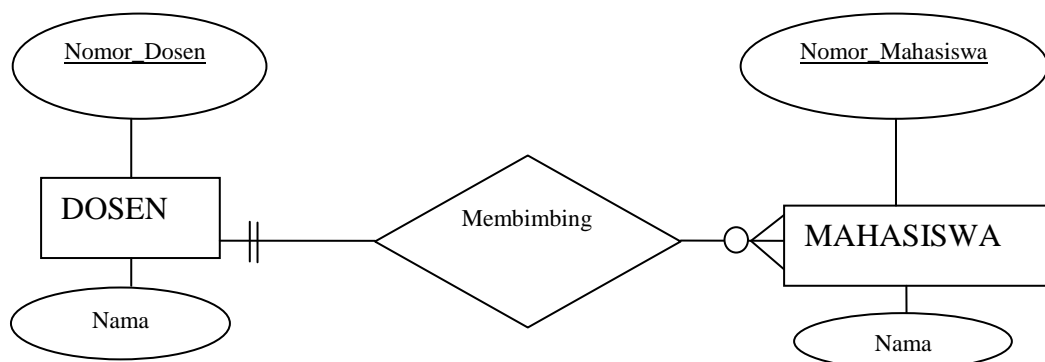


**Gambar II.3 : Contoh *Entitas Dan Atribut***

Sumber : Abdul Kadir (2009 : 32)

### 3. Hubungan (*Relationship*)

Hubungan (*Relationship*) menyatakan ketertarikan antara beberapa tipe *entitas*. Sebagai contoh , tipe *entitas* MAHASISWA dan DOSEN mempunyai hubungan yang mencerminkan bahwa seorang mahasiswa memiliki dosen pembimbing akademis. Gambar II.4 menunjukkan hubungan tersebut.



**Gambar II.4 : Contoh Hubungan Antara Tipe *Entitas***

Sumber : Abdul Kadir (2009 : 45)

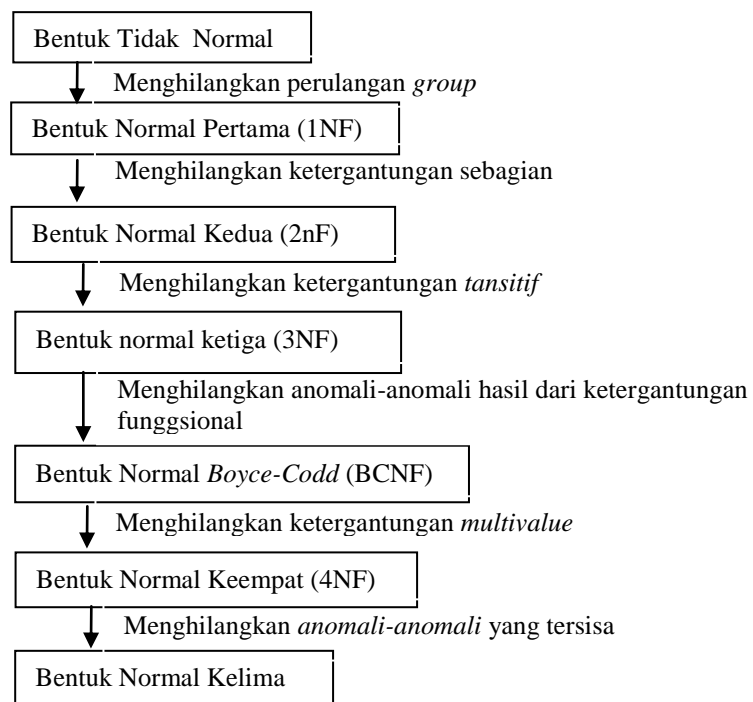
### II.5.3. Normalisasi

*Normalisasi* adalah proses yang digunakan untuk menentukan pengelompokan *atribut-atribut* dalam sebuah *relasi* sehingga diperoleh relasi yang berstruktur baik. Dalam hal ini yang dimaksud dengan *relasi* yang berstruktur baik adalah relasi yang memenuhi dua kondisi berikut :

1. Mengandung *redundansi* sedikit mungkin, dan
2. Memungkinkan baris-baris dalam *relasi* disispkan, dimodifikasi dan dihapus tanpa menimbulkan kesalahan atau ketidak konsistenan.

*Normalisasi* sendiri dilakukan melalui sejumlah langkah. Setiap langkah berhubungan dengan bentuk normal (*normal form*) tertentu. (Abdul Kadir; 2009 : 116).

Gambar II.5 berikut ini akan memperlihatkan hubungan keenam bentuk normal tersebut :



**Gambar II.5 : Langkah-Langkah Dalam Normalisasi**

Sumber : Abdul Kadir (2009 : 118)

## II.6. MySQL

*MySQL* merupakan *software* RDBMS (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*). Saat ini *MySQL* banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai ke industri, baik industri kecil, menengah, maupun besar.

Lisensi *MySQL* terbagi menjadi dua. Anda dapat menggunakan *MySQL* sebagai produk *open source* dibawah GNU *general Public License* (gratis) atau dapat membeli lisensi dari versi komersialnya. *MySQL* versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disertakan pada versi gratis. Pada kenyataannya, untuk keperluan industri menengah kebawah, versi gratis masih dapat digunakan dengan baik. (Budi Raharjo; 2011 : 21).

## II.7. Unified Modeling Language (UML)

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language (UML)*. *UML* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. *UML* merupakan bahasa visual untuk

pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

*UML* hanya berfungsi untuk melakukan pemodelan, jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada *metode* berorientasi objek. (Rosa A.S & M. Shalahuddin; 2011 : 118).

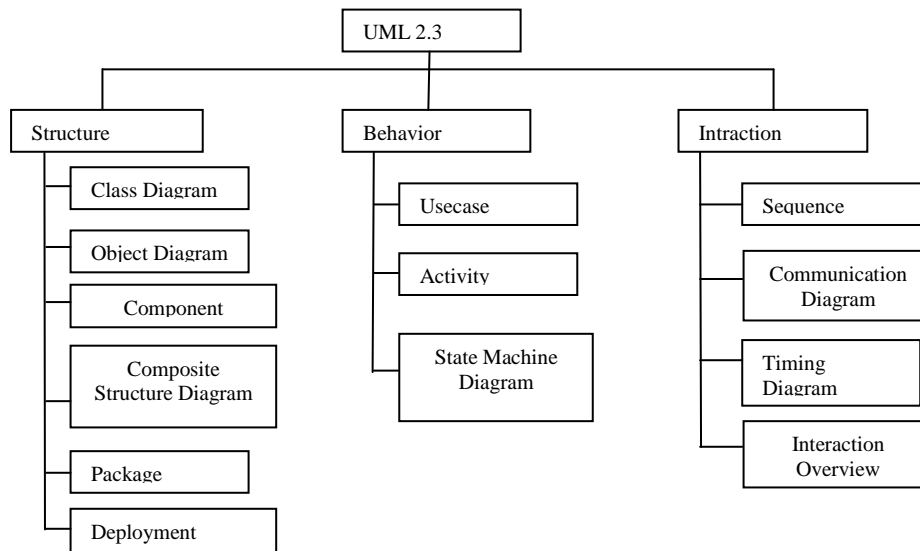
*UML* diaplikasikan untuk maksud tertentu, biasanya antara lain :

1. Merancang perangkat Lunak (*Software*).
2. Sarana Komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dengan mengaplikasikan beragam sistem. Intinya *UML* merupakan alat komunikasi yang konsisten dalam mendukung para pengembang sistem saat ini. (Prabowo Pudjo Widodo & Herlawati; 2011 : 6).

### II.7.1. Diagram-Diagram UML

Pada *UML 2.3* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar II.6 di bawah ini :



**Gambar II.6 : Diagram UML**

Sumber : Rosa A.S & M. Shalahuddin (2011 : 121)

Berikut ini penjelasan singkat dari pembagian kategori tersebut :

1. *Structure Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

### A. Class Diagram

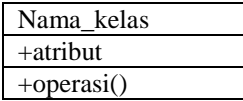



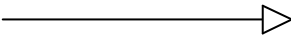
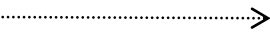
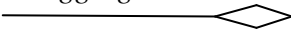
Diagram kelas atau *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Kelas memiliki apa yang disebut *atribut* dan *metode* atau operasi.

- 1) *Atribut* merupakan *variabel-variabel* yang dimiliki oleh suatu kelas.
- 2) Operasi atau *metode* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut Tabel II.1 menerangkan simbol-simbol pada diagram kelas :

**Tabel II.1 : Diagram Kelas**

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i>   Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>  Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum/khusus)
Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Semua bagian ( <i>whole part</i> )

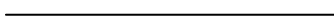
Sumber : Rosa A.S & M. Shalahuddin (2011 : 124)

## B. *Object Diagram*

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan. Untuk apa mendefinisikan sebuah kelas sedangkan pada jalannya sistem, objeknya tidak pernah dipakai.

Berikut adalah Tabel II.2 menerangkan simbol-simbol diagram objek :

**Tabel II.2 : Diagram Objek**

Simbol	Deskripsi
Objek <div style="border: 1px solid black; padding: 2px; margin: 5px 0;">           Nama_objek : nama_kelas            Atribut = nilai         </div>	Objek dari kelas yang berjalansaat sistem dijalankan
<i>Link</i> 	<i>Relasi</i> antar objek

Sumber : Rosa A.S & M. Shalahuddin (2011 : 124)

## C. *Component Diagram*

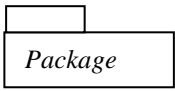
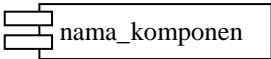
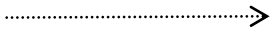

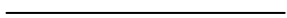
Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. Komponen dasar yang biasanya ada dalam suatu sistem adalah sebagai berikut :

- 1) Komponen *user interface* yang menangani tampilan.
- 2) Komponen *bussiness procesiing* yang menangani fungsi-fungsi proses bisnis.

- 3) Komponen data yang menangani manipulasi data.
- 4) Komponen *security* yang menangani keamanan sistem.

Komponen lebih terfokus pada penggolongan secara umum fungsi-fungsi yang diperlukan, berikut Tabel II.3 yang menerangkan simbol-simbol yang ada pada diagram komponen.

**Tabel II.3 : Diagram Komponen**

Simbol	Deskripsi
<i>Package</i> 	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen
Komponen 	Komponen Sistem
Kebergantungan / <i>dependency</i> 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
Antar muka / <i>interface</i>  nama_interface	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen
<i>Link</i> 	<i>Relasi</i> antar komponen

Sumber : Rosa A.S & M. Shalahuddin (2011 : 126)

#### D. Use Case Diagram

*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak

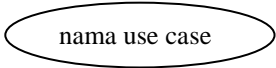


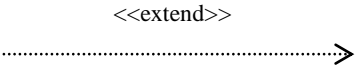


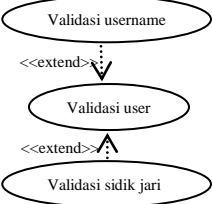
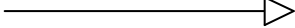
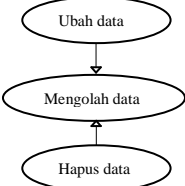
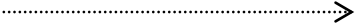
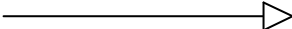
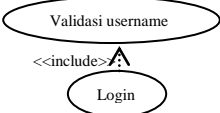
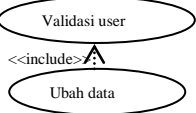
menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

- 1) Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- 2) *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut Tabel II.4 menerangkan simbol-simbol pada diagram *use case*.

**Tabel II.4 : Diagram Use Case**

Simbol	Deskripsi
<p><i>Use case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
<p>Aktor / <i>actor</i></p> 	Orang, proses, atau sistem yang lain berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat itu sendiri
<p>Asosiasi / <i>association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> , atau <i>use case</i> memiliki interaksi dengan aktor
<p>Ekstensi / <i>extend</i></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan misalnya :

	 <pre> graph TD     A(Validasi username) -.-&gt; &lt;&lt;extend&gt;&gt;  B(Validasi user)     B -.-&gt; &lt;&lt;extend&gt;&gt;  C(Validasi sidik jari)   </pre> <p>arah panah mengarah pada <i>use case</i> yang ditambahkan</p>
<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya misalnya :</p>  <pre> graph TD     A(Ubah data) --&gt; B(Mengolah data)     B --&gt; C(Hapus data)   </pre> <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p>&lt;&lt;include&gt;&gt;</p>  <p>&lt;&lt;uses&gt;&gt;</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p> <p>Ada 2 sudut pandang yang cukup besar mengenai include di <i>use case</i></p> <ol style="list-style-type: none"> <li><i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> dijalankan misal pada kasus berikut :</li> </ol>  <pre> graph TD     A(Validasi username) -.-&gt; &lt;&lt;include&gt;&gt;  B(Login)   </pre> <ol style="list-style-type: none"> <li><i>include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah di jalankan sebelum <i>use case</i> tambahan di jalankan, misal pada kasus berikut :</li> </ol>  <pre> graph TD     A(Validasi user) -.-&gt; &lt;&lt;include&gt;&gt;  B(Ubah data)   </pre>

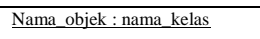

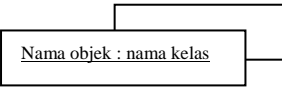


	Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.
--	---

Sumber : Rosa A.S & M. Shalahuddin (2011 : 131)

### E. *Communication Diagram*

Diagram komunikasi mengelompokkan *message* pada kumpulan diagram sekuen menjadi sebuah diagram. Dalam diagram komunikasi yang dituliskan adalah operasi / *metode* yang di jalankan antara objek yang satu dengan objek lainnya secara keseluruhan, oleh karna itu dapat di ambil dari jalanya interaksi pada semua diagram sekuen. Berikut adalah Tabel II.5 yang menerangkan simbol-simbol yang ada pada diagram komunikasi :

**Tabel II.5 : Diagram Komunikasi**

Simbol	Deskripsi
Objek 	Objek yang melakukan interaksi pesan
<i>Link</i> 	<i>Relasi</i> antar objek yang menghubungkan objek satu dengan lainnya atau dengan dirinya sendiri 
Arah pesan / stimulus 	Arah pesan yang terjadi, jika pada suatu <i>link</i> ada dua arah pesan yang berbeda, maka arah juga digambarkan dua arah pada dua sisi <i>link</i> 

Sumber : Rosa A.S & M. Shalahuddin (2011 : 140)


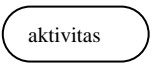



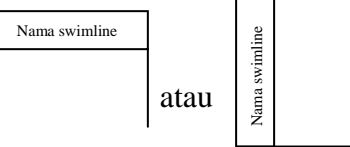
## F. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

- 1) Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- 2) Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
- 3) Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Berikut adalah Tabel II.6 yang menggambarkan simbol-simbol yang ada pada diagram aktivitas :

**Tabel II.6 : Diagram Aktivitas**

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decesion</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi


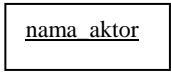

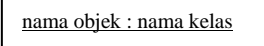

Sumber : Rosa A.S & M. Shalahuddin (2011 : 134)

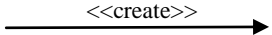
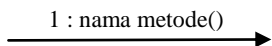
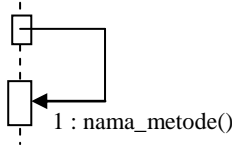
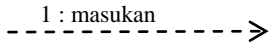
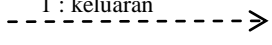
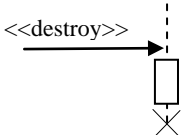
### G. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Banyaknya diagram objek yang digambarkan adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalanya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah Tabel II.7 yang menerangkan simbol-simbol yang ada pada diagram sekuen :

**Tabel II.7 : Diagram Sekuen**

Simbol	Deskripsi
Aktor  nama aktor atau  tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya di nyatakan menggunakan kata benda di awali <i>frase</i> nama aktor
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek
Objek  nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe <i>create</i>	Objek yang lain, arah panah mengarah pada objek yang dibuat

	
<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi / <i>metode</i> yang ada pada objek lain atau dirinya sendiri</p>  <p>Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi/<i>metode</i> yang di panggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe <i>return</i></p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau <i>metode</i> menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i></p>

Sumber : Rosa A.S & M. Shalahuddin (2011 : 138)