

BAB II

TINJAUAN PUSTAKA

II.1.1 Konsep Dasar Sistem

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai tujuan tertentu (Asbon Hendra ; 2012 : 157)

1. Syarat-Syarat Sistem

- a. Sistem harus dibentuk untuk menyelesaikan tujuan.
- b. Elemen sistem harus mempunyai rencana yang ditetapkan.
- c. Adanya hubungan diantara elemen sistem.
- d. Unsur dasar dari proses (arus informasi, energi, dan material) lebih penting daripada elemen sistem.
- e. Tujuan organisasi lebih penting dari pada tujuan elemen

2. Karakteristik Sistem

a. Komponen (*Component*)

Suatu Sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan, Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli berapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem

secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut supra sistem. Sebagai contoh, suatu perusahaan dapat disebut dengan suatu sistem dan industri yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem. Kalau dipandang industri sebagai suatu sistem, maka perusahaan dapat disebut sebagai subsistem. Demikian juga bila perusahaan dipandang sebagai suatu sistem, maka sistem akuntansi adalah subsistemnya. (Asbon Hendra ; 2012 : 158)

b. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan lainya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut. (Asbon Hendra ; 2012 : 158)

c. Lingkungan Luar Sistem (*Environment*)

Environment merupakan segala sesuatu di luar batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan atau dikendalikan agar tidak mengganggu operasi sistem. (Asbon Hendra ; 2012 : 159)

d. Penghubung Sistem (*Interfase*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain, output dari suatu subsistem akan menjadi input dari subsistem yang lainnya. (Asbon Hendra ; 2012 : 159)

e. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan Sinyal (*Signal Input*) adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh, di dalam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi. (Asbon Hendra ; 2012 : 159)

f. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh sistem, meliputi *output* yang berguna, contohnya informasi yang dikeluarkan oleh komputer. Dan *output* yang tidak berguna dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer. (Asbon Hendra ; 2012 : 160)

g. Pengolahan Sistem (*Process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Contoh CPU pada komputer, bagian produksi yang mengubah

bahan baku menjadi barang jadi, serta bagian akuntansi yang mengolah data transaksi menjadi laporan keuangan (Asbon Hendra ; 2012 : 160)

h. Tujuan Sistem (*Goal*)

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi input yang dibutuhkan dan output yang dihasilkan, dengan kata lain, suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenal sasaran atau tujuannya. Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak akan digunakan. (Asbon Hendra ; 2012 : 160)

II.1.2 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 ditansaksikan, serta jumlah dan harganya. Data dinyatakan dengan nilai yang berbentuk angka, deretan karakter, atau simbol (Kusrini ; 2007:3)

II.1.3. 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 (Kusrini ; 2007 : 4).

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 (Kusrini ; 2007 : 4)

Kegunaan informasi bagi seseorang juga sangat tergantung pada waktu. Pada suatu waktu tertentu informasi tersebut mungkin sangat diperlukan di lain 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 ; 2007:4)

II.1.4. 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 ; 2007 : 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. Kenyataannya, 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 (Kusrini ; 2007 : 5).

II.1.5. 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 (Kusrini ; 2007 : 11)

Berdasarkan dukungan kepada pemakainya, sistem informasi dibagi menjadi

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 (*Intelligent 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 ; 2007 : 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 aktifitas 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 (Adam Suseno & Ricky Agus; 2012 : 2).

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 geografis 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

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

b. Data Raster

Model data ini terdiri dari sekumpulan *grid/sel* seperti peta hasil *scanning* maupun gambar/*image*. Masing-masing *grid/sel* atau *pixel* memiliki nilai tertentu yang bergantung pada bagaimana *image* tersebut digambarkan. Sebagai contoh, pada sebuah *image* hasil penginderaan jarak jauh dari sebuah satelit, masing – masing pixel direpresentasikan sebagai panjang gelombang cahaya yang dipantulkan dari posisi permukaan bumi dan diterima oleh satelit dalam satuan luas tertentu yang disebut pixel. Pada *image* hasil *scanning*, masing – masing *pixel* merepresentasikan keterangan nilai yang berasosiasi dengan point-point tertentu pada *image* hasil *scanning* tersebut

c. Data Non Spasial (Atribut)

Data non *spasial* adalah data berbentuk tabel dimana tabel tersebut berisi informasi- informasi yang dimiliki oleh obyek 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.3. Sistem Informasi Berbasis Arcview

Kemampuan *Arcview GIS* pada berbagai serinya tidaklah diragukan lagi. *Arcview GIS* adalah *software* yang di keluarkan oleh ESRI (*Environmental Sistem Research Instituet*). Perangkat lunak ini memberikan fasilitas teknis yang berkaitan dengan pengelola data spasial. Kemampuan grafis yang baik dan kemampuan teknis dalam mengelola data spasial tersebut memberikan kekuatan secara nyata pada *Arcview* untuk melakukan analisis spasial. Kekuatan analisis inilah yang pada akhirnya menjadikan *Arcview* banyak ditetapkan dalam berbagai pekerjaan, seperti analisis pemasaran, perencanaan wilayah dan tata ruang, sistem informasi persil, pengendalian dampak lingkungan, bahkan untuk keperluan militer (Eko budyanto ; 2010 : 177)

Mengapa *Arcview* dapat memiliki keluwesan yang sedemikian hebat? Hal itu disebabkan oleh adanya dukungan dari skrip *Avenue*. Melalui *avenue* ini dapat dibentuk suatu "kemampuan baru" pada *Arcview*. Tentu saja hal ini membuat *Arcview* menjadi sangat luwes untuk diterapkan pada berbagai permasalahan spasial. *Avenue* dapat di gunakan untuk "merombak" wajah *Arcview* sesuai kebutuhan penggunanya.

Avenue adalah sebuah skrip atau bahasa pemrograman berorientasi objek (*Object Oriented programming*) (Esri, 1996: 5), Dengan *Avenue* ini dapat di bentuk sebuah *interface* baru pada *Arcview*, otomatis pekerjaan-pekerjaan yang bersifat berulang (*Repectitif*), ataupun membuat sebuah alur analisis spasial khusus yang belum terdapat pada *Arcview* tersebut. *Avenue* banyak digunakan untuk membentuk sebuah sistem informasi aplikatif pada suatu lembaga atau instansi dengan berbasis *Arcview GIS*. (Eko budyanto ; 2010 : 177)

II.4. 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 Nuryadin ; 2005 : 3)

Dalam buku Panduan Menggunakan MapServer yang Ruslan Nuryadin 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.

1. 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 (Ruslan Nuryadin ; 2005 : 4)

II.5. 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 *maintanance* situs *web* tersebut lebih mudah dan efisien. PHP merupakan software *open-source* yang disebar dan dilisensikan secara gratis dan dapat didownload dari situs resminya yaitu : <http://www.php.net> (Rosa & M. Shalahuddin; 2011 : 85)

II.6. 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* didefinisikan 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* (Budi Raharjo ; 2011 : 3)

II.6.1. 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.6.2. 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)

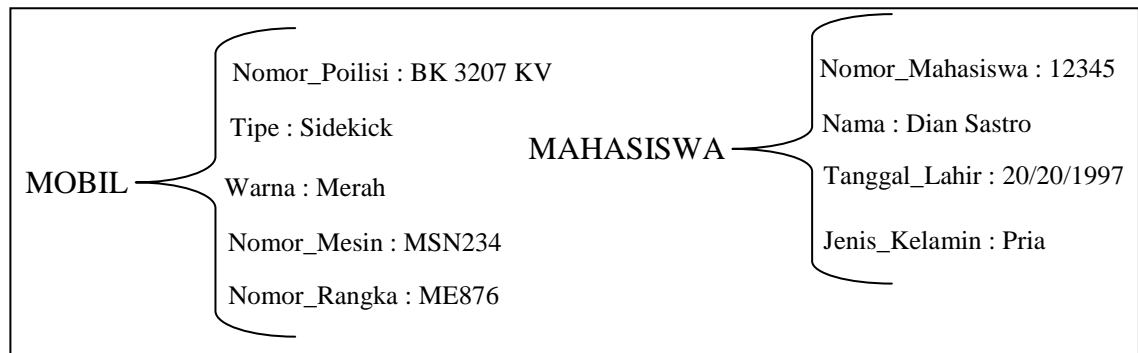
1. Entitas

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 sesuatu 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 (Abdul Kadir ; 2009:31)

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 (Abdul Kadir ; 2009:32)

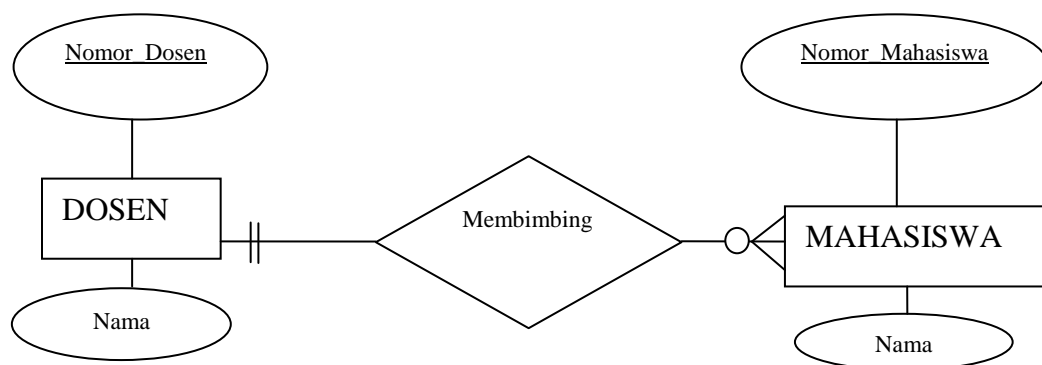


Gambar II.3 : Contoh Entitas dan Atribut

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. (Abdul Kadir ; 2009:45)



Gambar II.4 : Contoh Hubungan antara tipe entitas

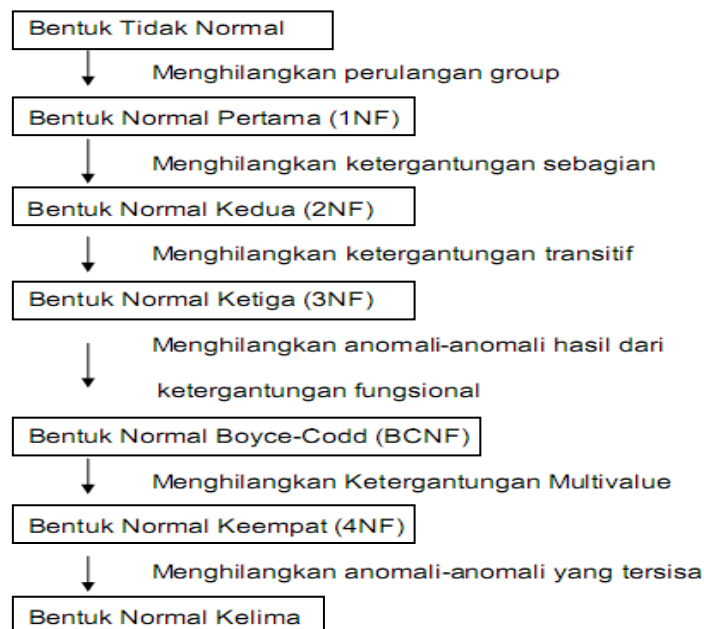
Sumber : Abdul Kadir (2009:45)

II.6.3. 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 disisipkan, dimodifikasi dan dihapus tanpa menimbulkan kesalahan atau ketidakkonsistenan (Abdul Kadir ; 2009 : 116)

Normalisasi sendiri dilakukan melalui sejumlah langkah. Setiap langkah berhubungan dengan bentuk normal (*normal form*) tertentu. 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.7. 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.8. 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)

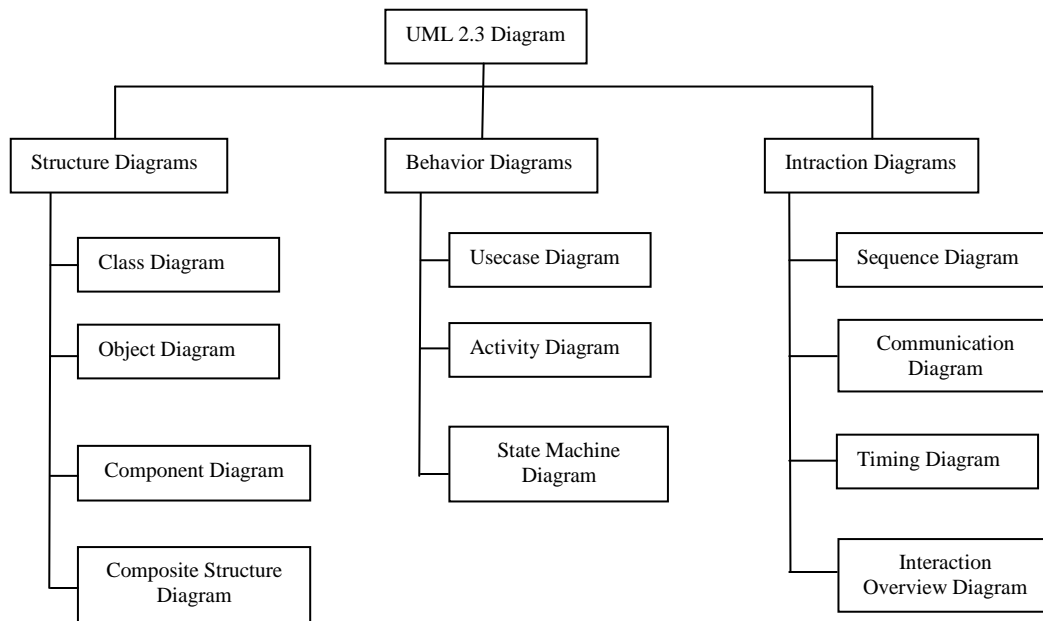
UML diaplikasikan untuk maksud tertentu, biasanya antara lain :

1. Merancang perangkat Lunak.
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.8.1 Diagram-Diagram UML

Menurut Rosa A.S & M. Shalahuddin (2011 : 120) 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.
3. *Interaction Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain.

A. Class Diagram

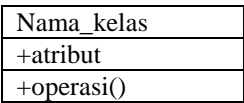



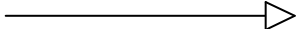
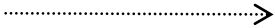
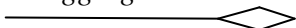
Diagram kelas atau *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

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. Use Case Diagram

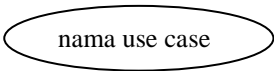

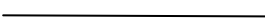

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang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

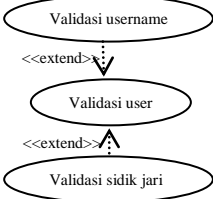
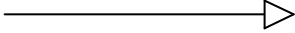
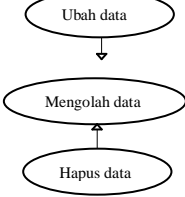
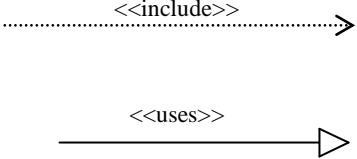
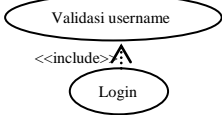
dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

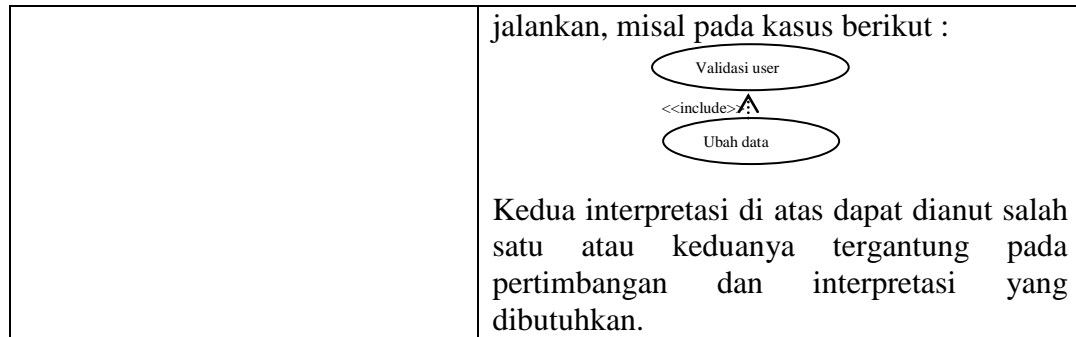
- 1) Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- 2) *Use case* merupakan fungsionalitas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

Berikut tabel II.2 menerangkan simbol-simbol pada diagram *use case*

Tabel II.2 : Diagram Use case

Simbol	Deskripsi
Use case 	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>
Aktor / <i>actor</i>  nama aktor	Orang, proses, atau sistem yang lain berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat itu sendiri
Asosiasi / <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> , atau <i>usecase</i> memiliki interaksi dengan aktor
Ekstensi / <i>extend</i> <<extend>> 	Relasi usecase 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 inheritance pada pemrograman berorientasi

	<p>objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan misal</p>  <pre> graph TD A(Validasi username) -.-> <<extend>> B(Validasi user) B -.-> <<extend>> 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) --> B(Mengolah data) C(Hapus data) --> B </pre> <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan / <i>include / uses</i></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 <i>include</i> di <i>usecase</i></p> <ol style="list-style-type: none"> 1. <i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> dijalankan misal pada kasus berikut :  <pre> graph TD A(Validasi username) -.-> <<include>> B(Login) </pre> <ol style="list-style-type: none"> 2. <i>include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah di jalankan sebelum <i>use case</i> tambahan di



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


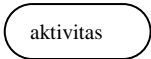



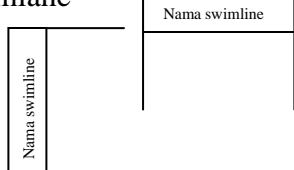
C. *Activity Diagram*

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 antarmuka tampilan
- 3) Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Berikut adalah tabel II.3 yang menggambarkan simbol-simbol yang ada pada diagram aktivitas :

Tabel II.3 : Diagram Aktivitas


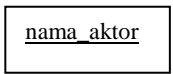

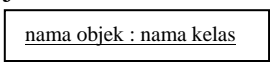

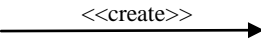
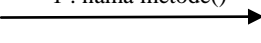
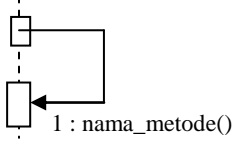
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / decesion 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan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

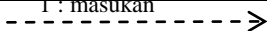
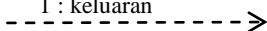
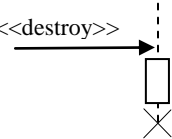
Sumber : Rosa A.S & M. Shalahuddin (2011 : 134)

D. 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 didefenisikan interaksi jalanya pesan sudah dicakup dapa diagram sekuen sehingga semakin banyak use case yang didefenisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut adalah tabel II.4 yang menerangkan simbol-sombol yang ada pada diagram sekuen :

Tabel II.4 : Diagram Squence

Simbol	Deskripsi
<p>Aktor</p>  <p>atau</p>  <p>tampa waktu aktif</p>	<p>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 frase nama aktor</p>
<p>Garis hidup / lifeline</p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi pesan</p>
<p>Pesan tipe create</p> 	<p>Objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> 	<p>Menyatakan suatu objek memanggil operasi / metode 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 / metode yang di panggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Pesan tipe send</p>	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya,</p>

<p>1 : masukan </p>	<p>arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe return 1 : keluaran </p>	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p>Pesan tipe destroy </p>	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy</p>

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