

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Sistem adalah sebuah tatanan yang terdiri atas kumpulan komponen fungsional (dengan tugas / fungsi khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk mencapai suatu proses / pekerjaan tertentu.

Sebagai contoh, sistem pernafasan terdiri dari : hidung, tenggorokan, paru-paru, pembuluh darah dan darah (Tata Sutabri,S.Kom,MM;2005:6-8).

II.1.1. 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.

b. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem yang dipandang sebagai suatu kesatuan, karena dengan batas sistem ini berfungsi dan tugas dari subsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luar Sistem (*Environment*)

Environment merupakan segala sesuatu diluar batas sistem yang memengaruhi 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.

d. Penghubung Sistem (*Interface*)

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.

e. Masukkan Sistem (*Input*)

Merupakan energi yang dimasukkan kedalam sistem. Masukan dapat berupa Masukan Perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi.

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 berguna dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer.

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.

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 itu mengenai sasaran dan tujuannya. Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. (Asbon Hendra;2012;158-160).

II.2. Pengertian Informasi

Untuk memahami makna sistem informasi, harus dilihat keterkaitan antara data dan informasi sebagai entitas penting pembentuk sistem informasi. Informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung disaat itu juga atau secara tidak langsung pada saat mendatang. Untuk memperoleh informasi, diperlukan data yang akan diolah dan unit pengolah.

Contoh informasi adalah daftar pegawai berdasarkan departemen, daftar pegawai berdasarkan golongan, rekapitulasi transaksi pembelian pada akhir bulan, rekapitulasi transaksi penjualan pada akhir bulan, dan lain-lain (Edhy Sutanta ; 2011 : 13-14).

II.3. Pengertian Sistem Informasi

Sistem informasi disebut sebagai sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen baik manual ataupun berbasis komputer yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi kepada pihak-pihak yang berkepentingan sebagai pemakai informasi tersebut.” (Anastasia Diana & Lilis Setiawati ; 2011 : 4)

II.3.1. Konsep Dasar Sistem Informasi

Komponen-komponen sistem informasi adalah sebagai berikut:

1. Perangkat keras (*hardware*)

Perangkat keras (*hardware*) terdiri dari monitor, CPU, keyboard, mouse dan harddisk.

2. Perangkat lunak (*software*)

Perangkat lunak (*software*) berupa program-program aplikasi yang akan digunakan, yaitu merupakan kumpulan dari perintah atau fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu.

3. Data

Data merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi.

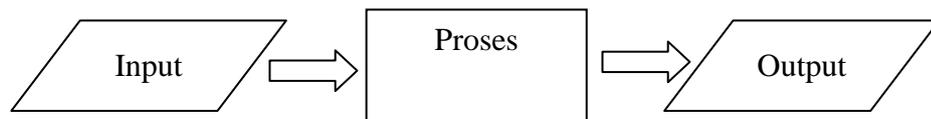
4. Prosedur

Prosedur merupakan dokumentasi prosedur atau proses sistem, tata cara atau penuntun operasional (aplikasi) dan teknis

5. Manusia

Manusia adalah pengguna dari sistem informasi.

Sedangkan komponen utama suatu sistem informasi terdiri dari *input*, proses dan *output*. Adapun komponen utama sistem informasi dapat dilihat pada gambar II.1. dibawah ini :



Gambar II.1. Komponen sistem informasi
Sumber : Anastasia Diana & Lilis Setiawati (2011 : 4)

II.4. Pengertian Produksi

Produksi merupakan salah satu kegiatan yang berhubungan erat dengan kegiatan ekonomi. Melalui proses produksi bisa dihasilkan berbagai macam barang yang dibutuhkan oleh manusia. Tingkat produksi juga dijadikan sebagai patokan penilaian atas tingkat kesejahteraan suatu negara. Jadi tidak heran bila setiap negara berlomba - lomba meningkatkan hasil produksi secara global untuk meningkatkan pendapatan perkapitanya.

II.5. Pengertian Minyak Goreng

Minyak goreng merupakan salah satu kebutuhan pokok manusia sebagai alat pengolahan bahan-bahan makanan. Minyak goreng merupakan media penggorengan yang sangat penting dan kebutuhannya semakin meningkat. Minyak dapat bersumber dari tanaman, misalnya minyak zaitun, minyak jagung, minyak kelapa, dan minyak biji bunga matahari. Minyak juga dapat bersumber dari hewan, misalnya ikan sarden. Ikan paus dan lain-lain.

II.6. *Entity Relationship Diagram (ERD)*

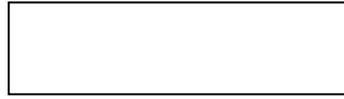
Merupakan suatu model untuk menjelaskan hubungan antar dua dalam basis data berdasarkan suatu persepsi bahwa *real word* terdiri dari *object-object* dasar yang mempunyai hubungan atau antar *object-object* tersebut. Relasi antar *object* dengan menggunakan symbol-simbol grafis tertentu.

Entity Relationship Model merupakan suatu model data yang dikembangkan berdasarkan obyek (Linda Marlinda, S. Kom 2004:16).

II.6.1. **Komponen-komponen yang terdapat didalam *Entity Relationship Model*.**

1. *Entity*
 - a. Adalah sesuatu yang dapat dibedakan dalam dunia nyata dimana informasi yang berkaitan dengannya dikumpulkan.
 - b. *Entity set* adalah kumpulan *entity* yang sejenis.
 - c. Symbol yang digunakan untuk *entity* adalah persegi panjang.
 - d. *Entity set* dapat berupa :
 - *Entity* yang bersifat fisik, yaitu *entity* yang dapat dilihat.
Contohnya : rumah, kendaraan, mahasiswa, dosen, dan lain-lain.
 - *Entity* yang bersifat konsep atau logic, yaitu *entity* yang tidak dapat dilihat.
Contohnya : pekerjaan, perusahaan, rencana, mata kuliah, dan lain-lain.
 - Simbol yang digunakan untuk *entity* adalah persegi panjang.

Untuk melihat gambar *entity* ini, lihat pada gambar II.2. sebagai berikut :



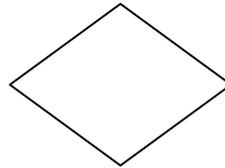
Gambar II.2. Entity

Sumber : Linda Marlinda, S. Kom (2004:17)

2. Relationship

- a. Adalah hubungan yang terjadi antara satu atau lebih *entity*.
- b. *Relationship* tidak mempunyai keberadaan fisik, kecuali yang mewarisi hubungan antara *entity* tersebut.
- c. *Relationship* set adalah kumpulan relationship yang sejenis.
- d. Simbol yang digunakan adalah bentuk belah ketupat, *diamond* atau *rectangle*.

Untuk melihat gambar *relationship* ini, lihat pada gambar II.3. sebagai berikut :



Gambar II.3. Relationship

Sumber : Linda Marlinda, S. Kom (2004:18)

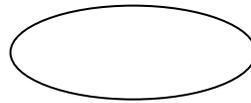
3. Attribute

- a. Adalah karakteristik dari *entity* atau *relationship* yang menyediakan penjelasan detail tentang atau *relationship* tersebut.
- b. Attribute value (nilai atribut) adalah suatu data aktual atau informasi yang disimpan di suatu atribut di dalam suatu *entity* atau *relationship*.

c. Terdapat dua jenis attribute, yaitu :

1. *Indetifer (key)*, untuk menentukan suatu *entity* secara unik.
 2. *Descriptor (nonkey attribute)*, untuk menentukan karakteristik dari suatu *entity* yang tidak unik.
- d. Simbol yang digunakan adalah bentuk oval

Untuk melihat gambar *attribute* ini, lihat pada gambar II.4. sebagai berikut :



Gambar II.4. Attribute

Sumber : Linda Marlinda, S. Kom (2004:18)

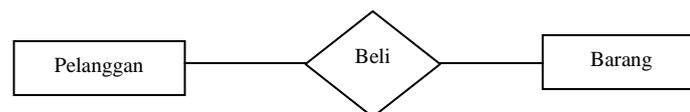
4. Indicator Tipe

a. *Indicator tipe associative object*

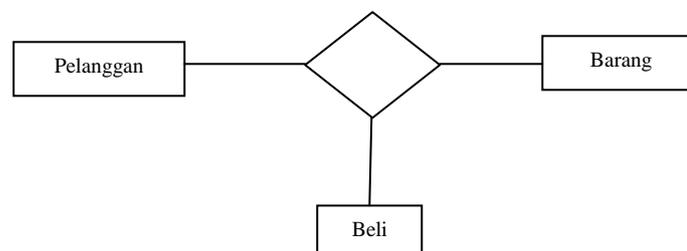
Berfungsi sebagai suatu objek dan suatu *relationship*

Untuk melihat gambar *indicator tipe* ini, lihat pada gambar II.5. sebagai berikut :

Contoh :



Menjadi :



Gambar II.5. Indicator Tipe

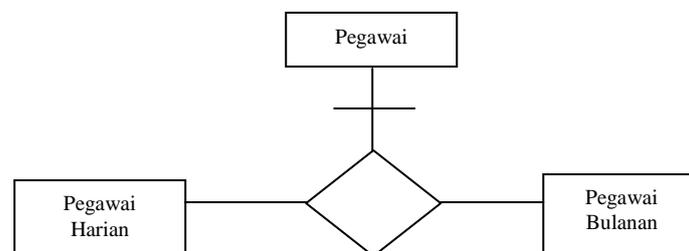
Sumber : Linda Marlinda, S. Kom (2004:19)

b. Indicator tipe supertipe

Terdiri dari suatu object dan sub kategori atau lebih yang dihubungkan dengan dihubungkan dengan relationship yang tidak bernama (Linda Marlinda, S. Kom 2004:17-19).

Untuk melihat gambar *indicator tipe supertipe* ini, lihat pada gambar II.6. sebagai berikut :

Contoh :



Gambar II.6. Indicator Tipe SuperTipe

Sumber : Linda Marlinda, S. Kom (2004:19)

II.7. Kamus Data

Kamus Data (KD) atau data dictionary (DD) atau disebut juga dengan istilah System data dictionary adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi”(Jogiyanto : 2005 ; 725).

KD dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem.

II.8. Normalisasi

Normalisasi adalah sebagai teknik yang menstrukturkan/mendekmbinasikan data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud

adalah berkaitan dengan penyimpangan-penyimpangan (anomallies) yang terjadi akibat adanya kerangkapan data dalam relasi dan in-efisiensi pengolahan.

Teori normaisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu.ada bentuk normall yang telah ditemukan (Edhy sutanta ; 2011: 175).

II.8.1. Tahap-Tahap Normalisasi

a. Bentuk tidak normal (*Unnormalized Form*)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu. Dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya dengan saat menginput.

Contoh data :

Tabel II.1. Relasi Supplier dalam bentuk UNF (*Unnormalized Form*)

<u>Kode_supplier</u>	<u>Status</u>	<u>Kota</u>	<u>Kode_Barang</u>	<u>Jumlah_barang</u>
S01	10	Jakarta	B01	100
			B02	150
			B03	200
S02	20	Surabaya	B02	250
S03	30	Yogyakarta	B05	150

Sumber : Edhy Sutanta (2011 : 179)

b. Bentuk Normal Pertama (1 NF/ *Fisrt Normal Form*)

Suatu relasi 1 NF dan hanya jika sifat dan setiap relasi atributenya bersifat *atomic*. Atom adalah zat terkecil yang masih memiliki sifat induknya. Bila dipecah lagi maka ia tidak memiliki sifat induknya.

Tabel II.2. Relasi Supplier_1 (1 NF/ First Normal Form)

Kode_supplier	Status	Kota	Kode_Barang	Jumlah_barang
S01	10	Jakarta	B01	100
S01	10	Jakarta	B02	150
S01	10	Jakarta	B03	200
S02	20	Surabaya	B02	250
S03	30	Yogyakarta	B05	150

Sumber : Edhy Sutanta (2011 : 180)

c. Bentuk Normal Kedua (2 NF/ Second Normal Form)

Fungsional (*functional dependeny*) yang bersifat sebagian (*partial functional dependeny*) telah dihilangkan.

Tabel II.3. Relasi Supplier_2 (2NF/ Second Normal Form)

Kode_supplier	Status	Kota
S1	10	Jakarta
S2	20	Surabaya
S3	30	Yogyakarta

Sumber : Edhy Sutanta (2011 : 181)

d. Bentuk Normal Ketiga (3 NF/Third Normal Form)

Untuk menjadi bentuk normal ketiga maka relasi dalam bentuk normal kedua dan semua *attribute* bukan primer tidak punya hubungan yang transistif. Dengan kata lain, setiap *attribute* bukan kunci haruslah bergantung hanya pada *primary key* dan *primary key* secara menyeluruh.

Tabel II.4. Relasi Supplier_3 (3 NF/ Third Normal Form)

Kode_supplier	Status
S1	10
S2	20
S3	30

Sumber : Edhy Sutanta (2011 : 182)

Tabel II.5. Relasi Kota (3 NF/ *Third Normal Form*)

Status	Kota
10	Jakarta
20	Surabaya
30	Yogyakarta

Sumber : Edhy Sutanta (2011 : 182)

e. *Boyee-Cood Normal Form* (BCNF)

BCNF mempunyai paksaan lebih kuat dan bentuk normal ketiga untuk menjadi BCNF, relasi harus dalam bentuk normal kesatu dan setiap *attribute* harus bergantung fungsi pada *attribute superkey*.

f. Bentuk Normal Ke Empat (4NF)

Relasi R adalah bentuk 4 NF jika dan hanya jika relasi tersebut juga termasuk BCNF dan semua ketergantungan *multivalued* adalah juga ketergantungan fungsional.

g. Bentuk Normal Kelima (5 NF)

Disebut juga PINF (*Projection Join Normal Form*) dan 4 NF dilakukan dengan menghilangkan ketergantungan *join* yang merupakan kunci kandidat.

II.9. Sekilas Tentang Visual Basic

Visual Basic versi sebelumnya adalah Visual Basic 6 yang diluncurkan Microsoft tahun 1998. kemudian pada bulan Juli 2000, Microsoft memaparkan pengembangan Microsoft.NET dalam PDC (Professional Developer Conference) di Orlando, Florida, AS. Setelah dilakukan pengembangan cukup lama, akan pada Februari 2002 secara resmi Microsoft merilis VS.NET di mana salah satu bahasa pemrogramannya adalah VB.NET. Lalu pada tahun 2003, Microsoft merilis VS.NET 2003 yang memperbaiki performa dan aspek keamanan dari VS.NET

2002. Pada VS.NET 2003 telah menggunakan .NET Framework 1.1 yang terbaru. Pada awal tahun ini Microsoft juga telah merilis VS.NET 2005 dengan menggunakan .NET Framework 2.0, kemudian merilis VS.NET 2008 dan yang terbaru VS.NET 2010 dengan menggunakan Framework 4.0 (Wahana Komputer : 2011 ; 2).

II.10. Sekilas Tentang SQL Server

SQL Server 2008 adalah sebuah terobosan baru dari microsoft dalam bidang database. SQL Server adalah sebuah DBMS(database Management System) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengelolaan data mentusul pendahulunya seperti IBM dan Oracle. SQL server 2008 dibuat pada saat kemajuan dalam bidang hardware sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data”(Wahana Komputer : 2010 ; 2).

Microsoft merilis SQL Server 2008 dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju.

Versi-versi tersebut adalah:

- a. Versi Compact, ini adalah versi “Tipis” dari semua versi yang ada. Versi ini seperti versi desktop pada SQL Server 2000.
- b. Versi Express, ini adalah versi “Ringan” dari semua versi yang ada dan paling cocok untuk latihan para pengembang aplikasi.

- c. Versi Workgroup (Workgroup Edition), versi ini dirancang untuk kalangan bisnis berskala kecil dan biasanya digunakan pada level dalam departemen saja.
- d. Versi standar (Standard Edition), versi ini menyediakan apa yang dimiliki oleh versi workgroup.
- e. Versi Enterprise (Enterprise Edition), versi ini memiliki semua fasilitas yang ada pada versi standard, tetapi versi ini mampu menangani user yang banyak.
- f. Versi Developer (Developers Edition), versi ini memiliki semua keuntungan dari versi enterprise.

II.11. Pengenalan UML

Unified Modelling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. (Prabowo puajo widodo & Herlawati ; 2011 : 6)

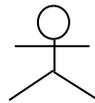
UML dikembangkan oleh 3 pendekar 'berorientasi objek', yaitu *Grady Booch*, *Jim Rumbaugh*, dan *Ivar Jacobson*. UML menjadi bahasa yang bisa digunakan untuk berkomunikasi dalam perspektif obyek antara *user* dengan *developer*, antara *developer* dengan *developer*, antara *developer* analisis dengan *developer* disain, dan antara *developer* disain dengan *developer* pemrograman. UML memungkinkan *developer* melakukan permodelan secara *visual*, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Permodelan *visual* membantu untuk menangkap struktur dan kelakuan dari obyek, mempermudah

penggambaran interaksi antara elemen dalam sistem, dan mempertahankan konsistensi antara disain dan implementasi dalam pemrograman.

UML menyediakan standar pada notasi dan artifak (*diagram*) yang bisa digunakan untuk memodelkan suatu sistem. Berikut ini adalah notasi yang ada pada UML, yaitu sebagai berikut:

1. *Actor*

Actor adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat. Jadi walaupun simbol dari *actor* adalah gambar orang, tapi *actor* belum tentu merupakan orang.

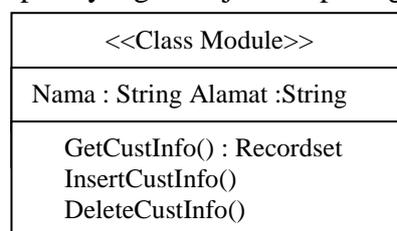


Gambar II.7. Notasi *Actor*

Sumber : Rosa A.S - M.Shalahuddin (2011 : 120)

2. *Class*

Class merupakan pembentuk utama dari sistem berorientasi objek karena class menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama. Notasi *class* berbentuk persegi panjang berisi 3 bagian : persegi paling atas untuk nama *class*, persegi panjang tengah untuk atribut, dan persegi panjang paling bawah untuk operasi. Seperti yang ditunjukkan pada gambar II.8. sebagai berikut :

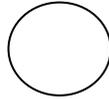


Gambar II.8. Notasi *Class*

Sumber : Rosa A.S - M.Shalahuddin (2011 : 122)

3. *Interface*

Interface merupakan kumpulan operasi tanpa implementasi dari suatu *class*.



Gambar II.9. Notasi Interface

Sumber : Rosa A.S - M.Shalahuddin (2011 : 122)

4. *Use Case*

Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antarunit atau *actor*.

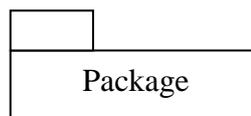


Gambar II.10. Notasi Use Case

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

5. *Package*

Package merupakan sebuah bungkusan dari satu atau lebih kelas atau elemen diagram UML lainnya. Tujuannya adalah untuk mempermudah penglihatan (*visibility*) dari model yang sedang dibangun.

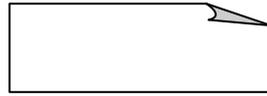


Gambar II.11. Notasi Package

Sumber : Rosa A.S - M.Shalahuddin (2011 : 130)

6. *Note*

Note digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model.

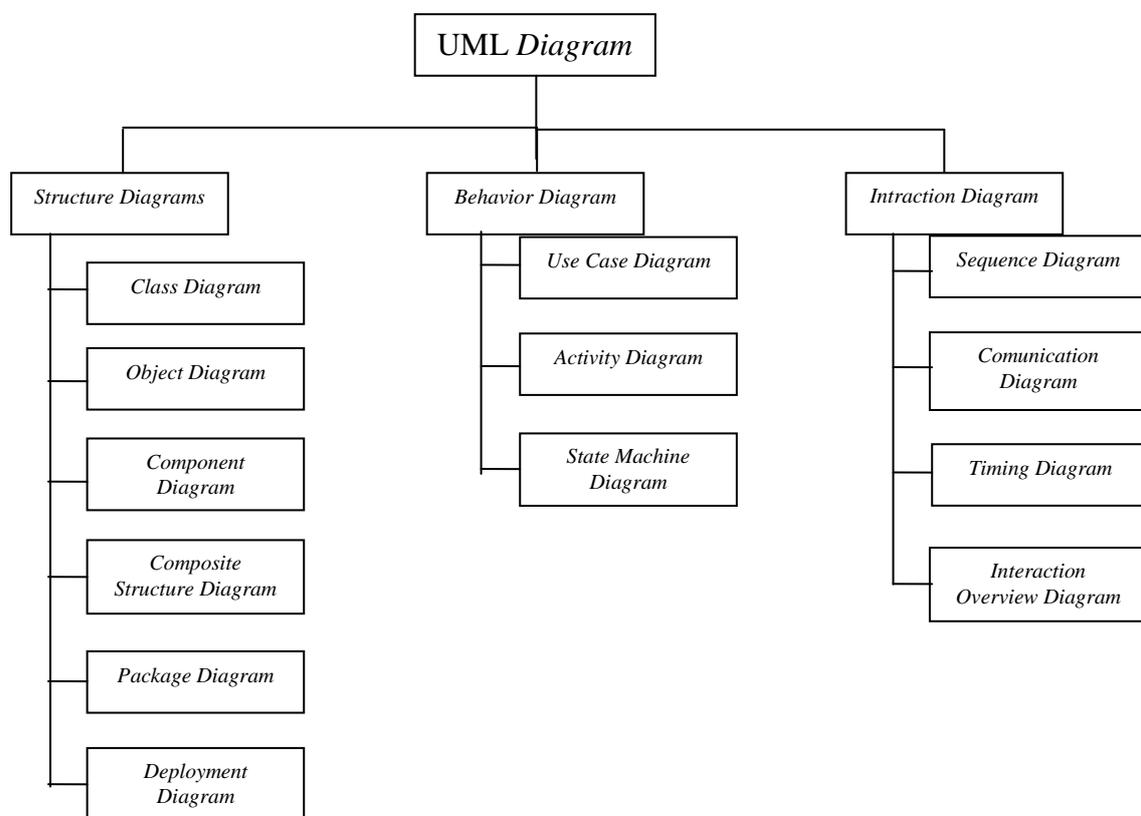


Gambar II.12. Notasi Note

Sumber : Rosa A.S - M.Shalahuddin (2011 : 130)

II.12.Diagram UML

Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar II.13. dibawah ini :



Gambar II.13. 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.

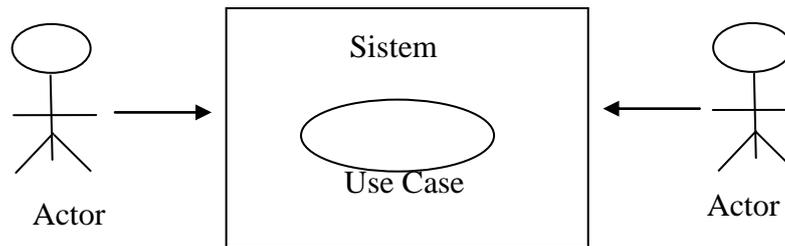
3. *Interaction Diagrams*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

Diagram – diagram pada metode UML (*Unified Modelling Language*) adalah sebagai berikut :

1. *Use Case Diagram*

Use case adalah alat bantu terbaik guna menstimulasikan pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Tidak selalu mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud menggunakan sebuah sistem. Karena sistem pengembangan tradisional sering ceroboh dalam melakukan analisis, akibatnya pengguna seringkali susah menjawabnya tatkala dimintai masukan tentang sesuatu. Ide dasarnya adalah bagaimana melibatkan penggunaan sistem di fase – fase awal analisis dan perancangan sistem. Diagram *use case* menunjukkan 3 aspek dari sistem yaitu *actor*, *use case* dan sistem / sub sistem *boundary*. *Actor* mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan *use case*. Gambar II.14. mengilustrasikan *actor*, *use case* dan *boundary*.

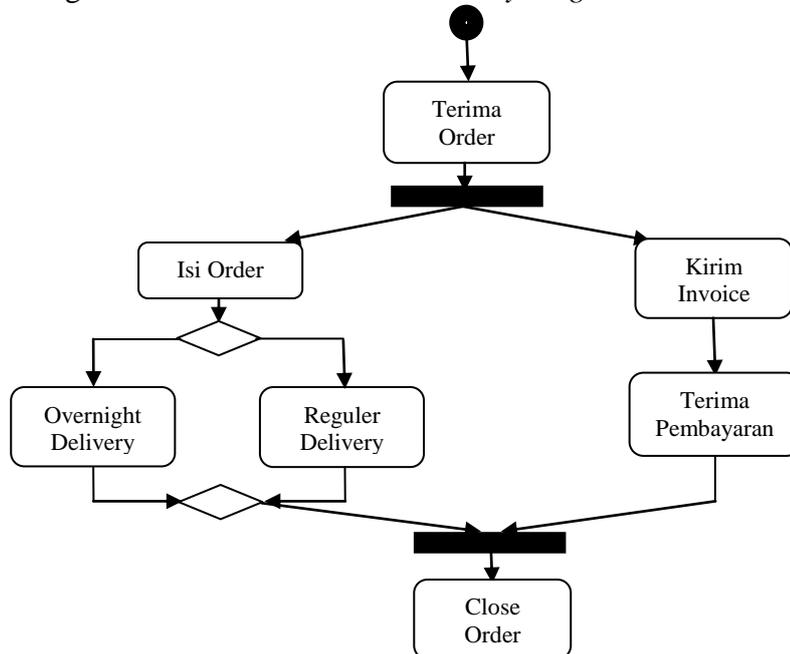


Gambar II.14. Use Case Model
Sumber : Rosa A.S - M.Shalahuddin (2011 : 131)

2. Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa.

Berikut gambar dari sederhana dari *Activity diagram*.



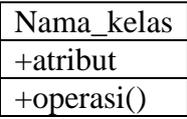
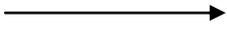
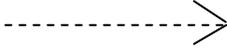
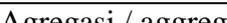
Gambar II.15. Contoh Activity Diagram Sederhana
Sumber : Munawar (2005 : 111)

3. Class Diagram

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.

- a. Atribut merupakan varabel-variabel yang dimiliki oleh suatu kelas.
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel II.6. Simbol-Simbol pada Class Diagram

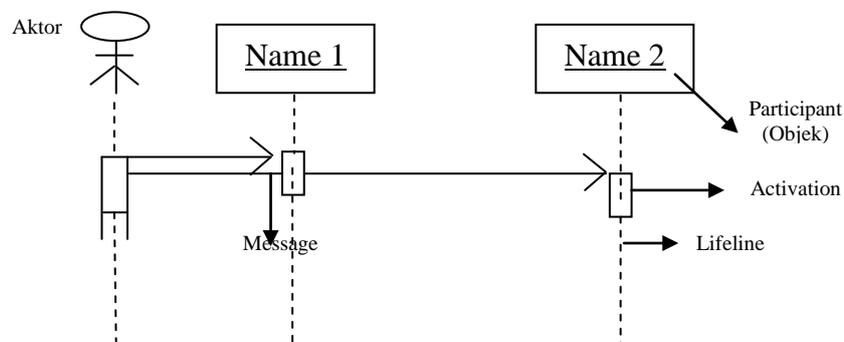
Simbol	Deskripsi
Kelas 	kelas pada struktur system
Antarmuka / interface  Nama_interface	sama dengan konsep interface dalam pemrograman berorientasi objek
Asosiasi / association 	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
Asosiasi berarah / directed association 	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
Generalisasi 	relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / dependency 	relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / aggregation 	relasi antar kelas dengan makna

Sumber : Rosa A.S.M.Shalahuddin (2011 : 123)

4. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan pesan yang diletakan diantara obyek – obyek ini di dalam *use case*.

Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical* (Munawar, 2005 : 87).



Gambar II.16. Simbol-Simbol pada Sequence Diagram
Sumber : Rosa A.S.M.Shalahuddin (2011 ; 139)