

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Sistem informasi berbasis komputer merupakan sekelompok perangkat keras dan perangkat lunak yang dirancang untuk mengubah data menjadi informasi yang bermanfaat. Jenis sistem informasi berbasis komputer

1. Pengolahan Data. Pengolahan data elektronik – *Electronic Data Processing (EDP)* adalah pemanfaatan teknologi komputer untuk melakukan pengolahan data transaksi-transaksi dalam suatu organisasi. EDP adalah aplikasi sistem informasi akuntansi paling dasar dalam setiap organisasi. Sehubungan dengan perkembangan teknologi komputer, istilah pengolahan data mulai dikenal dan mempunyai arti yang sama dengan istilah EDP.
2. Sistem Informasi Manajemen (SIM), menguraikan penggunaan teknologi komputer untuk menyediakan informasi bagi pengambilan keputusan para manajer.
3. Sistem Pendukung Keputusan – *Decision Support Systems (DSS)*. DSS diarahkan untuk melayani permintaan informasi tertentu, khusus, dan tidak rutin dari manajemen. Contohnya adalah penggunaan *spreadsheet* untuk melakukan analisis “*what if*” dari data operasi atau anggaran.
4. Sistem Pakar – *Expert Systems (ES)* adalah sistem informasi berbasis pengetahuan yang memanfaatkan pengetahuannya tentang bidang aplikasi tertentu untuk bertindak seperti seorang konsultan ahli bagi pemakainya.

5. Sistem Informasi Eksekutif – *Executive Information Systems* (EIS). EIS dibuat bagi kebutuhan informasi strategik manajemen tingkat puncak.
6. Sistem Informasi Akuntansi – sistem berbasis komputer yang dirancang untuk mengubah data akuntansi menjadi informasi (Agustinus ; 2012 : 2).

II.2. Pengertian Informasi

Informasi adalah data yang berguna yang telah diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat. Sistem Informasi sangat penting bagi organisasi. Pada dasarnya informasi adalah penting seperti sumber daya yang lain, misalnya peralatan, bahan, tenaga, dsb.

Informasi yang berkualitas dapat mendukung keunggulan kompetitif suatu organisasi. Dalam sistem informasi akuntansi, kualitas dari informasi yang disediakan merupakan hal penting dalam kesuksesan sistem.

Secara konseptual seluruh sistem organisasional mencapai tujuannya melalui proses alokasi sumber daya, yang diwujudkan melalui proses pengambilan keputusan manajerial. Informasi memiliki nilai ekonomik pada saat ia mendukung keputusan alokasi sumber daya, sehingga dengan demikian mendukung sistem untuk mencapai tujuan.

Pemakai informasi akuntansi dapat dibagi dalam dua kelompok besar: ekstern dan intern. Pemakai ekstern mencakup pemegang saham, investor, kreditor, pemerintah, pelanggan, pemasok, pesaing, serikat pekerja, dan masyarakat. Pemakai intern terutama paramanager, kebutuhannya bervariasi tergantung pada tingkatannya (Agustinus ; 2012 : 1).

II.3. Sistem Informasi Akuntansi

Sistem informasi akuntansi adalah kumpulan sumber daya, seperti manusia dan peralatan, yang diatur untuk mengubah data menjadi informasi. Informasi ini dikomunikasikan kepada beragam pengambil keputusan. SIA mewujudkan perubahan ini secara manual atau terkomputerisasi. SIA juga merupakan sistem yang paling penting di organisasi dan merubah cara menangkap, memproses, menyimpan, dan mendistribusikan informasi (Xu, 2009). Saat ini, digital dan informasi online semakin digunakan dalam sistem informasi akuntansi. Organisasi perlu menempatkan sistem di lini depan, dan mempertimbangkan baik segi sistem ataupun manusia sebagai faktor yang terkait ketika mengatur sistem informasi akuntansi.

SIA pada umumnya meliputi beberapa siklus pemrosesan transaksi:

1. Siklus pendapatan. Berkaitan dengan pendistribusian barang dan jasa ke entitas lain dan pengumpulan pembayaran-pembayaran yang berkaitan.
2. Siklus pengeluaran. Berkaitan dengan perolehan barang jasa dari entitas lain dan pelunasan kewajiban yang berkaitan.
3. Siklus produksi. Berkaitan dengan pengubahan sumberdaya menjadi barang dan jasa.
4. Siklus keuangan. Kejadian-kejadian yang berkaitan dengan perolehan dan manajemen dana-dana modal, termasuk kas. (Agustinus ; 2012 : 3).

II.4. Kontrol Persediaan

Kontrol Persediaan adalah merupakan usaha-usaha yang dilakukan oleh suatu perusahaan termasuk keputusan-keputusan yang diambil sehingga kebutuhan akan bahan untuk keperluan proses produksi dapat terpenuhi secara optimal dengan resiko yang sekecil mungkin. Persediaan yang terlalu besar (over stock) merupakan pemborosan karena menyebabkan terlalu tingginya beban-beban biaya guna penyimpanan dan pemeliharaan selama penyimpanan di gudang. Disamping itu juga persediaan yang terlalu besar berarti terlalu besar juga barang modal yang menganggur dan tidak berputar. Begitu juga sebaliknya kekurangan persediaan (out of stock) dapat mengganggu kelancaran proses produksi sehingga ketepatan waktu pengiriman sebagaimana telah ditetapkan oleh pelanggan tidak terpenuhi yang ada sehingga pelanggan lari ke perusahaan lain. Singkatnya pengendalian persediaan merupakan usaha-usaha penyediaan bahan-bahan yang diperlukan untuk proses produksi sehingga dapat berjalan lancar tidak terjadi kekurangan bahan serta dapat diperoleh biaya persediaan yang sekecil-kecilnya, (Eddy Herjanto ; 237).

II.5. Metode FIFO

“Metode FIFO mengasumsikan bahwa barang dagangan yang pertama dibeli adalah barang dagangan yang pertama dijual (*the first merchandise purchased is the first merchasndise sold*), karena hargapokok penjualan dinilai berdasarkan harga pokok persediaan pertama masuk maka harga pokok persediaan yang tersisa terdiri dari harga pokok persediaan yang terakhir kali masuk.”

Semua kasus FIFO, persediaan dan harga pokok penjualan akan sama pada akhir bulan terlepas dari apakah yang dipakai adalah sistem persediaan perpetual atau periodik.

Hal ini disebabkan karena yang akan menjadi bagian dari harga pokok penjualan adalah barang-barang yang akan dibeli terlebih dahulu, dan karenanya dikeluarkan lebih dulu terlepas dari apakah harga pokok penjualan dihitung seiring barang dijual sepanjang periode akuntansi (sistem perpetual) atau sebagai residu pada akhir periode akuntansi (sistem periodik). Selain dianjurkan oleh pemerintah, metode FIFO banyak digunakan oleh perusahaan-perusahaan karena :

1. Perhitungan dan pelaksanaannya sederhana.
2. Nilai persediaan akhir pada neraca sesuai dengan harga yang berlaku

Sekarang

3. Dapat menghindari kerusakan dan keusangan persediaan.

Namun metode FIFO juga mempunyai kelemahan. Kelemahan ini terlihat jika terjadi inflasi. Dengan adanya inflasi maka barang-barang cenderung meningkat sepanjang waktu, karena biaya dari barang-barang yang dibebankan pada harga pokok barang tersebut merupakan biaya dari barang yang dibeli pertama kali sehingga *cost of good sold*-nya terlalu rendah maka laba yang dilaporkan terlalu tinggi, akibatnya pajak yang dibayar oleh perusahaan terlalu tinggi. Kelemahan yang mendasar bahwa biaya berjalan tidak ditandingkan dengan pendapatan berjalan pada laporan laba rugi.

Metode First In First Out (FIFO) adalah metode penilaian persediaan yang menganggap barang yang pertama kali masuk diasumsikan keluar pertama kali pula. Pada umumnya perusahaan menggunakan metode ini, sebab metode ini perhitungannya sangat sederhana baik sistem fisik maupun sistem perpetual akan menghasilkan penilaian persediaan yang sama.

Cara menghitung persediaan akhir adalah sebagai berikut :

Persediaan awal	xxx
Pembelian	<u>xxx</u> +
Tersedia untuk dijual	xxx
Penjualan	<u>xxx</u> -
Persediaan akhir	xxx

Metode FIFO yang didasarkan atas sistem fisik, nilai persediaan akhir ditentukan dengan cara saldo fisik yang ada dikalikan harga pokok perunit barang yang terakhir kali masuk, bila saldo fisik ternyata lebih besar dari jumlah unit terakhir masuk maka sisanya diambilkan dari harga pokok perunit yang masuk sebelumnya. Sedangkan pada sistem perpetual pencatatan persediaan dilakukan secara terus menerus dalam kartu persediaan. Pada sistem ini apabila ada transaksi penjualan maka akan dijurnal dua kali, pertama mencatat harga pokok penjualan dan yang kedua mencatat harga pokok barang yang dijual, seperti berikut ini :

Kas/ Piutang Dagang	xxx
Penjualan	xxx
HPP	xxx
Persediaan barang	xxx

II.6. Pengertian Java

Java Language Spesification adalah definisi teknis dari bahasa pemrograman *Java* yang di dalamnya terdapat aturan penulisan sintaks dan semantik *Java*. Referensi lengkap tentang *Java Language Spesification* dapat anda temui pada website resmi *Java*, yaitu <http://java.sun.com/docs/books/jl> (Wahana Komputer ; 2010 : 3).

Java Memiliki banyak kelebihan, antara lain :

a. Sederhana

Untuk mempelajari Java tidaklah sulit. Anda bisa mempelajari dan membuat program secara cepat dengan menggunakan Java.

b. Berorientasi Objek

Meskipun banyak bahasa pemrograman yang menyebut dirinya berorientasi objek, Java bukan merupakan turunan dari bahasa pemrograman manapun dan sama sekali tidak kompatibel dengan source-code bahasa apapun.

c. Aman

Banyak bahasa yang pada awalnya dirancang dengan tingkat keamanan yang hampir tidak ada. Fasilitas-fasilitas yang diberikan seringkali dapat dimanfaatkan untuk disusupi oleh berbagai virus. Berbeda dengan Java yang disusun sejak awal dengan prinsip keamanan.

d. Interaktif

Java dirancang memenuhi kebutuhan dunia nyata untuk menciptakan program jaringan yang interaktif.

e. Kokoh

Java membatasi anda dari berbagai hal kunci supaya dapat menemukan kesalahan lebih cepat saat mengembangkan program.

f. Terinterpretasi dan Berkinerja Tinggi

Java dapat diterjemahkan oleh sistem apapun yang memiliki program Java di dalamnya. Java dirancang dengan hati-hati sehingga mudah diterjemahkan ke dalam bahasa asli suatu sistem untuk menghasilkan kinerja yang tinggi.

g. Terdistribusi

Java dapat dengan mudah bekerja dalam lingkungan terdistribusi seperti internet/intranet karena kemampuannya menangani protokol TCP/IP.

h. Dinamis

Karena berorientasi objek, Java mudah dalam hal pemeliharaan dan pengembangan, karena kita tidak harus membedah isi program untuk mengubah dan mengembangkan program dengan skala lebih besar (Yuniar Supardi ; 2010 : 2-3).

II.7. Pengertian NetBeans

NetBeans merupakan salah satu proyek *open source* yang disponsori oleh *SunMicrosystem*. Proyek ini berdiri pada tahun 2000 dan telah menghasilkan 2 produk, yaitu NetBeanss IDE dan NetBeans Platform. NetBeans IDE merupakan produk yang digunakan untuk melakukan pemrograman baik menulis kode, meng-*compile*, mencari kesalahan dan mendistribusikan program. Sedangkan NetBeans Platform adalah sebuah modul yang merupakan kerangka awal / pondasi dalam bangun aplikasi desktop yang besar.

NetBeans juga menyediakan paket yang lengkap dalam pemrograman dari pemrograman standar (aplikasi desktop), pemrograman enterprise, dan pemrograman perangkat mobile. Saat ini NetBeans telah mencapai versi 6.8 (Wahana Komputer ; 2010 : 15).

II.8. Pengertian Database

Database adalah sekumpulan *file* data yang saling berhubungan dan diorganisasi sedemikian rupa sehingga memudahkan untuk mendapat dan memproses data. Lingkungan sistem *database* menekankan data yang tidak tergantung (*idenpendent data*) pada aplikasi yang akan menggunakan data. Data adalah kumpulan fakta dasar (mentah) yang terpisah.

Sebuah *database* harus dibuat dengan rapi agar data yang dimasukkan sesuai dengan tempatnya. Sebagai contoh, di sebuah perpustakaan, penyimpanan buku dikelompokkan berdasar jenis atau kategori-kategori tertentu, misalnya kategori buku komputer, buku pertanian, dan lain-lain. Kemudian dikelompokkan lagi berdasarkan abjad judul buku, ini dilakukan agar setiap pengunjung dapat dengan mudah mencari dan mendapatkan buku yang dimaksud (Wahana Komputer ; 2006 ; 1).

II.9. Pengertian MySQL

Mysql pertama kali dirintis oleh seorang *programmerdatabase* bernama Michael Widenius, yang dapat anda hubungi di emailnya monty@analytikerna Mysql *databaseserver* adalah RDBMS (*Relasional Database Management System*) yang dapat menangani data yang bervolume besar. meskipun begitu, tidak menuntut *resource* yang besar. Mysql adalah *database* yang paling populer diantara *database* yang lain.

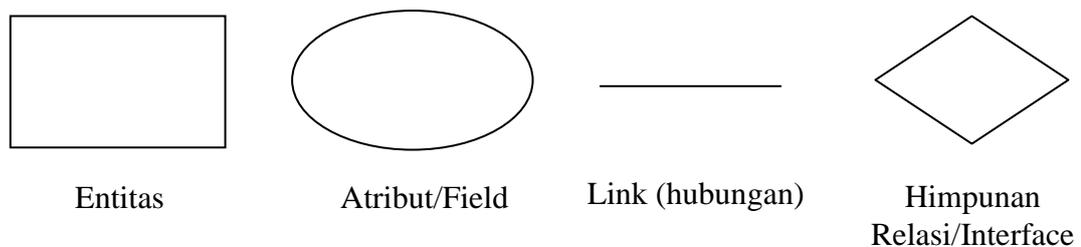
MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multiuser*. MySQL memiliki dua bentuk lisensi, yaitu *freeware* dan *shareware*. penulis sendiri dalam menjelaskan buku ini menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/GPL (*general public license*), yang dapat anda download pada alamat resminya <http://www.mysql.com>. MySQL sudah cukup lama dikembangkan, beberapa *fase* penting dalam pengembangan MySQL adalah sebagai berikut :

- MySQL dirilis pertama kali secara internal pada 23 Mei 1995

- Versi windows dirilis pada 8 Januari 1998 untuk windows 95 dan windows NT.
- Versi 3.23 : beta dari Juni 2000, dan dirilis pada January 2001.
- Versi 4.0 : beta dari Agustus 2002, dan dirilis pada Maret 2003 (unions)
(Wahana Komputer ; 2010).

II.10. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram atau ERD merupakan salah satu alat (tool) berbentuk grafis yang populer untuk *desain database*. Tool ini relatif lebih mudah dibandingkan dengan Normalisasi. Kebanyakan sistem analis memakai alat ini, tetapi yang jadi masalah, kalau kita cermati secara seksama, tool ini mencapai 2NF (Yuniar Supardi ; 2010 : 448).



Gambar. II.2 Bentuk Simbol ERD
(Sumber : Ir. Yuniar Supardi ; 2010 : 448)

II.11. Kamus Data

Karena DBMS menyimpan kumpulan beberapa *item* data yang terpisah yang dapat digunakan pemakai pada beberapa aplikasi secara bersama-sama, adalah penting bahwa beberapa mekanisme digunakan untuk menyediakan

informasi mengenai beberapa *item* data bersangkutan. Itu adalah fungsi dari kamus data.

Kamus data adalah suatu *file* yang terpisah yang menyimpan informasi seperti :

- a. Nama setiap *item*/jenis/kolom data.
- b. Struktur data untuk tiap *item*.
- c. Program yang menggunakan tiap *item*
- d. Tingkat keamanan untuk setiap *item* (Zulkifli Amsyah ; 2005 : 382).

Pemakai yang perlu memperoleh informasi dari *database* dapat menuju ke kamus data untuk mendapatkan nama dari *item* data yang digunakan pada penelusuran (*search*). Dan yang mendesain aplikasi dapat menggunakan kamus untuk menentukan apakah *item* data sudah disimpan di komputer, dan kalau sudah dengan nama apa *item* data tersebut dapat dipanggil dan aplikasi apa yang digunakan.

Kamus data berguna khusus bagi perlindungan timbulnya kelebihan data. Tanpa kamus data, pemakai dari lain bagian mungkin menyimpan versi identik dari *item* data yang sama pada beberapa lokasi, dimana masing-masing *item* data mempunyai nama yang berbeda.

Operasional komputer dalam organisasi dewasa ini banyak yang sudah menggunakan model kerja jaringan (*network*). Dengan menggunakan data dasar yang sama untuk keperluan informasi masing-masing unit dan manajemen organisasi secara keseluruhan.

II.12. Teknik Normalisasi

Proses normalisasi menyediakan cara sistematis untuk meminimalkan terjadinya kerangkapan data diantara relasi dalam perancangan logikal basis data.

Format normalisasi terdiri dari lima bentuk, yaitu:

II.12.1. Bentuk-bentuk Normalisasi

a. Bentuk normal tahap pertama (1st Normal Form)

Suatu tabel dikatakan sudah 1NF jika telah memenuhi ketentuan sebagai berikut:

- Tidak ada atribut mempunyai nilai berulang atau nilai array
- Tidak mempunyai baris yang rangkap

Bentuk unnormal mengijinkan nilai-nilai pada suatu atribut dapat berulang.

b. Bentuk normal tahap kedua (2nd normal form)

Relasi dapat dikatakan format normal kedua jika sudah dalam format normal pertama dan diikuti kondisi sebagai berikut:

- Key terdiri dari atribut tunggal
- Setiap atribut nonkey ketergantungan fungsional pada semua key atau tidak terjadinya ketergantungan pada key *composite*.

Misalnya tabel UNIV berada dalam normal kedua dengan mengasumsikan DNO sebagai key, kecuali CRSE. Jika ditentukan CNO dan SECNO sebagai key composite, atribut nonkey CNAME tergantung hanya pada CNO, bukan pada SECNO, sehingga CNAME

tidak secara ketergantungan fungsional penuh terhadap key (CNO, SECNO).

c. Bentuk normal tahap ketiga (3rd normal form)

Relasi dikatakan format normal ketiga jika sudah dalam format normal kedua dan tidak ada ketergantungan transitif diantara atribut. Misalnya tabel STUDNT mempunyai atribut SSNO sebagai *key* (2NF). Ketergantungan transitif terjadi diantara DNO dan COLREG. Saat DNO determinan COLREG tanpa melibatkan *key* SSNO. Contohnya, DNO='CS' termasuk COLREG='Arts/Sc.' tidak tergantung oleh atribut SSNO, sehingga STUDNT belum termasuk 3NF. Yang menjadi catatan, ketergantungan transitif tidak akan terjadi jika ada ketergantungan fungsional diantara atribut-atribut *non-key* yang melibatkan *key*.

d. Boyce Code Normal Form (BCNF)

BCNF menentukan setiap determinan adalah kunci kandidat (*candidate key*). Misalnya UNIV mempunyai dua determinan yaitu DNO dan DNAME yang merupakan *kunci kandidat* sehingga termasuk ke dalam BCNF. Di lain pihak CRSLST dalam 3NF tetapi tidak dalam BCNF. Atribut komposisinya (CNO, SECNO, SID, OFRNG) sebagai kunci-kunci kandidat dan tidak ada ketergantungan transitif, sehingga CRSLST termasuk ke dalam 3NF. Namun atribut CNO adalah determinan saat SECNO tergantung penuh secara fungsional terhadap

CNO, walaupun CNO bukan kunci kandidat, sehingga CRSLST belum termasuk BCNF.

e. Bentuk Normal Tahap Keempat dan Kelima

Bentuk ini adalah bentuk normal ketiga atau BCNF dengan nilai atribut tidak tergantung pada nilai banyak (*multivalued dependency*).

Konsep pada bentuk ini adalah ketergantungan pada gabungan beberapa atribut (*join dependency*) (Haidar Dzacko ; 2007 : 12).

II.13. UML (*Unified Modeling Language*)

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

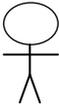
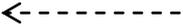
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara

satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.1. Simbol Use Case

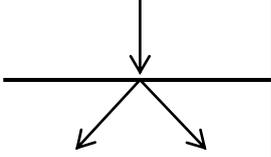
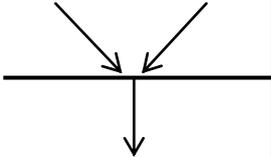
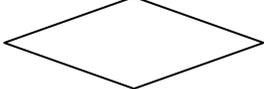
Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang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

(Sumber : Windu Gata ; 2013 : 4)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

3. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.3. *Multiplicity Class Diagram*

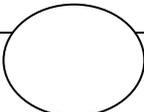
Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

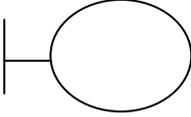
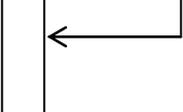
(Sumber : Windu Gata ; 2013 : 9)

4. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang

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

(Sumber : Windu Gata ; 2013 : 7)