

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Sistem merupakan sekumpulan elemen-elemen yang saling terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan yang telah ditetapkan. Karakteristik sistem terdiri dari :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem

Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran Sistem

Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya (Sulindawati ; 2010 : 135).

II.2. Pengertian Sistem Informasi

Sistem informasi bukan merupakan hal yang baru, yang baru adalah komputerisasinya. Sebelum ada komputer, teknik penyaluran informasi yang memungkinkan manajer merencanakan serta mengendalikan operasi telah ada. Komputer menambahkan satu atau dua dimensi, seperti kecepatan, ketelitian dan

penyediaan data dengan volume yang lebih besar yang memberikan bahan pertimbangan yang lebih banyak untuk mengambil keputusan.

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu (Tata Sutabri ; 2012 : 38).

II.3. Akuntansi

Akuntansi merupakan bahasa bisnis. Sebagai bahasa bisnis akuntansi menyediakan cara untuk menyajikan dan meringkas kejadian-kejadian bisnis dalam bentuk informasi keuangan kepada pemakainya. Informasi akuntansi merupakan bagian terpenting dari seluruh informasi yang diperlukan oleh manajemen. Informasi akuntansi yang dihasilkan oleh suatu sistem dibedakan menjadi dua, yaitu informasi akuntansi keuangan dan informasi akuntansi manajemen.

Pemakai informasi akuntansi pun terdiri dari dua kelompok, yaitu pemakai eksternal dan pemakai internal. Yang dimaksud dengan pemakai ekseternal mencakup pemegang saham, investor, kreditor, pemerintah, pelanggan, pemasok, pesaing, serikat kerja dan masyarakat. Sedangkan pemakai internal adalah pihak manajer dari berbagai tingkatan dalam organisasi bersangkutan (Kusrini ; 2012 : 1).

II.4. Sistem Informasi Akuntansi

Informasi akuntansi diberikan sebagai alat atau sarana untuk membantu manajer menjalankan fungsi-fungsi manajemen sehingga tujuan organisasi dapat tercapai. Fungsi manajemen, seperti perencanaan (*planning*), pengorganisasian (*organizing*), pengarahan karyawan (*directing*) dan pengendalian (*controlling*) tidak dapat dilakukan tanpa informasi yang memadai. Informasi dalam sebuah organisasi merupakan perekat yang mengikat fungsi-fungsi manajemen dalam sebuah sistem sehingga memungkinkan organisasi bertindak koheran dan harmonis antar berbagai fungsi.

1. Perencanaan (*Planning*)

Tahap pertama dalam perencanaan adalah mengidentifikasi alternatif-alternatif yang tersedia, kemudian memilih salah satu dari berbagai alternatif tersebut yang paling baik dan cocok dengan tujuan organisasi. Rencana manajemen biasanya diekspresikan dalam sebuah bentuk formal yang disebut dengan anggaran (*budgets*). Oleh karena itu, istilah penganggaran (*budgeting*) sering pula disebut sebagai proses perencanaan. Anggaran biasanya dibuat di bawah kendali dan koordinasi seorang manajer atau kepala bagian akuntansi. Umumnya anggaran dibuat setiap tahun dan mencerminkan rencana manajemen dalam format yang spesifik dan kuantitatif. Anggaran ini dibuat berdasarkan data yang dikumpulkan, dianalisis dan diikhtisarkan oleh akuntan manajemen.

2. Pengorganisasian Sumber-sumber Daya (*Organizing*)

Langkah selanjutnya adalah pengorganisasian sumber-sumber daya, dimana manajer menetapkan koordinasi antar bagian (*organizationchart*). Perkembangan Pengorganisasian hanya merupakan alat untuk mencapai tujuan yang telah ditentukan.

3. Pengarahan Karyawan (*Directing and Motivating*)

Setelah rencana kerja untuk periode mendatang disusun, manajer harus melakukan pemantauan terus menerus terhadap kegiatan sehari-hari dan menjaga agar organisasi dapat berfungsi sebagaimana mestinya. Untuk melakukan hal tersebut, maka manajer harus memiliki kemampuan untuk memotivasi dan mengarahkan karyawan secara efektif.

4. Pengendalian dan Pengawasan (*Controlling*)

Dalam melaksanakan tugas pengendalian, manajer sebenarnya mencari jaminan bahwa rencana yang telah disusun dalam bentuk anggaran sudah dilaksanakan sebagaimana mestinya (Suprihatmi : 2009 : 57).

II.5. Metode Perpetual

Sistem pencatatan metode perpetual disebut juga metode buku adalah sistem dimana setiap persediaan yang masuk dan keluar dicatat di pembukuan.

Setiap jenis barang dibuatkan kartu persediaan dan di dalam pembukuan dibuatkan rekening pembantu persediaan. Rincian dalam buku pembantu bisa diawasi dari rekening kontrol persediaan barang dalam buku besar. Rekening yang digunakan untuk mencatat persediaan ini terdiri dari beberapa kolom yang dapat dipakai untuk mencatat pembelian, penjualan dan saldo persediaan. Setiap

perubahan dalam persediaan diikuti dengan pencatatan dalam rekening persediaan sehingga jumlah persediaan sewaktu-waktu dapat diketahui dengan melihat kolom saldo dalam rekening persediaan. Masing-masing kolom dirinci lagi untuk kuantitas dan harga perolehannya.

Penggunaan metode buku akan memudahkan penyusunan neraca dan laporan laba rugi jangka pendek, karena tidak perlu lagi mengadakan perhitungan fisik untuk mengetahui jumlah persediaan akhir.

Ciri-ciri terpenting dalam sistem perpetual pada perjurnalan adalah :

- a. Pembelian barang dagangan dicatat dengan mendebet rekening persediaan.
- b. Harga pokok penjualan dihitung untuk tiap transaksi penjualan dan dicatat dengan mendebet rekening HPP pada persediaan.
- c. Persediaan merupakan rekening kontrol dan dilengkapi dengan buku pembantu persediaan yang berisi catatan untuk setiap jenis persediaan. Buku pembantu persediaan menunjukkan kuantitas dan harga perolehan untuk setiap jenis barang yang ada dalam persediaan. (Adi Pamungkas ; 2012 : 2)

II.6. Pengertian Macromedia Dreamweaver

Macromedia Dreamweaver adalah sebuah *software web design software web design* yang menawarkan cara mendesain *website* dengan dua langkah sekaligus dalam satu waktu, yaitu mendesain dan program. Dreamweaver memiliki satu jendela mini yang disebut *HTML Source*, tempat kode-kode HTML tertulis. Setiap kali kita mendesain *web*, seperti menulis kata-kata, meletakkan gambar, membuat table dan proses lainnya, tag-tag HTML akan

tertulis secara langsung mengiringi proses pengaturan *website*. Artinya kita memiliki kesempatan untuk mendesain. *Website* sekaligus mengenal tag-tag HTML yang membangun *website* itu. Di lain kesempatan kita juga dapat mendesain *website* hanya dengan menulis tag-tag dan teks lain di jendela HTML *Source* dan hasilnya dapat dilihat langsung di layar (M. Suyanto ; 2009 : 244).

II.7. Pengertian PHP

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

PHP adalah kependekan dari *HyperText Preprocessor* (suatu akronim rekursif) yang dibangun oleh Rasmus Lerdorf pada tahun 1994. Dahulu, pada awal pengembangannya PHP disebut sebagai kependekan dari *Personal Home Page*. PHP merupakan produk *Open Source* sehingga anda dapat mengakses *source code*, menggunakan dan mengubahnya tanpa harus membayar sepeser pun. Gratis (Antonius Nugraha Widhi Pratama ; 2010 : 9).

II.8. Pengertian Database

Secara sederhana *database* (basis data/pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Pengertian akses dapat mencakup pemerolehan data maupun manipulasi data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat disimpan dalam media pengingat yang disebut *harddisk*. Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk *database*.

Pengaplikasian *database* dapat kita lihat dan rasakan dalam keseharian kita. *Database* ini menjadi penting untuk mengelola data dari berbagai kegiatan. Misalnya, kita bisa menggunakan mesin ATM (*anjungan tunai mandiri/ automatic teller machine*) bank karena bank telah mempunyai *database* tentang nasabah dan rekening nasabah. Kemudian data tersebut dapat diakses melalui mesin ATM ketika bertransaksi melalui ATM. Pada saat melakukan transaksi, dalam konteks *database* sebenarnya kita sudah melakukan perubahan (*update*) data pada *database* di bank. Ketika kita menyimpan alamat dan nomor telepon di HP, sebenarnya juga telah menggunakan konsep *database*. Data yang kita simpan di HP juga mempunyai struktur yang diisi melalui formulir (*form*) yang disediakan. Pengguna dimungkinkan menambahkan nomor HP, nama pemegang, bahkan kemudian dapat ditambah dengan alamat *email*, alamat *web*, nama kantor, dan sebagainya (Agustinus Mujilan ; 2012 : 23).

II.9. Pengertian MySQL

Mysql pertama kali dirintis oleh seorang programmer *database* bernama Michael Widenius, yang dapat anda hubungi di emailnya monty@analytikerna.

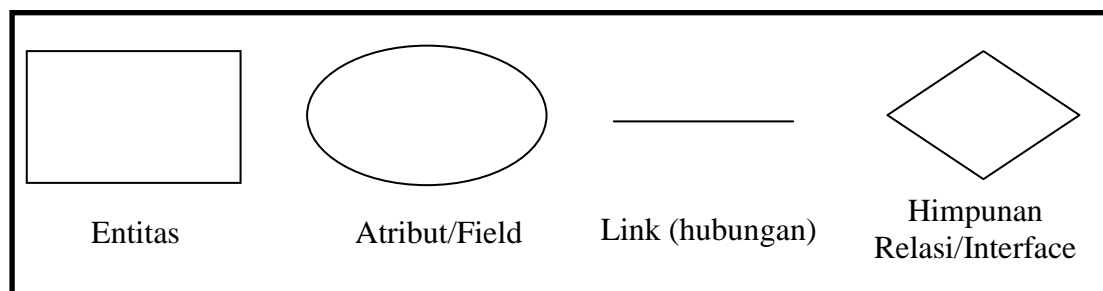
Mysql *database server* adalah RDBMS (*Relasional Database Management System*) yang dapat menangani data yang bervolume besar. meskipun begitu, tidak menuntut *resource* yang besar. Mysql adalah *database* yang paling populer di antara *database* yang lain.

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 publiclicense*), 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 :

1. MySQL dirilis pertama kali secara internal pada 23 Mei 1995
2. Versi *windows* dirilis pada 8 Januari 1998 untuk *windows 95* dan *windows NT*.
3. Versi 3.23 : beta dari Juni 2000, dan dirilis pada Januari 2001.
4. Versi 4.0 : beta dari Agustus 2002, dan dirilis pada Maret 2003 (*unions*)
(Wahana ; 2010 : 5).

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 (Ir. Yuniar Supardi ; 2010 : 448).



Gambar. II.1 Bentuk Simbol ERD
(Sumber : Ir. Yuniar Supardi ; 2010 : 448)

II.11. Kamus Data

Dalam tahapan ini unsur-unsur data yang diperlukan ditentukan untuk selanjutnya dijelaskan lagi dikamus data (*data dictionary*). Pengertian kamus data itu sendiri adalah suatu *ensiklopedik* dari informasi yang berkaitan dengan data perusahaan atau dapat juga kita katakan bahwa komputer (*computer-based catalog or directory*) yang berisi data perubahan (*metadata*) yang berkenaan dengan tahapan penjelasan data ini adalah system kamus data (*data dictionary system/DDS*) dan bahasa pendeskripsi data (*data description language/DDL*).

Sistem kamus data berbentuk perangkat lunak yang fungsinya adalah penciptaan dan pemerilahaan serta penyediaan kamus data agar dapat digunakan. Kamus data dapat berbentuk kertas atau punarsip (*file*) komputer. DDS dapat kita

peroleh dalam paket perangkat lunak terpisah ataupun dalam bentuk modul seperti yang ada dalam DBMS (*database management system*) dan CASE (teknik perangkat lunak tambahan komputer/ *computer aided software engineering*) (Ian Somerville ; 2010 : 344).

II.12. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

II.12.1. Bentuk-bentuk Normalisasi

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

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

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

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah

tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

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

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah table berada pada bentuk normal ketiga (3NF) jika table sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

4. BoyceCode Normal Form (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan kantabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan kanentitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

5. Bentuk Normal Tahap Keempat dan Kelima

Sebuah table relasional berada pada bentuk normal keempat (4NF) jika didalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD). Sebuah table berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah table lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima

berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata ; 2010 : 76).

II.12.2. 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.

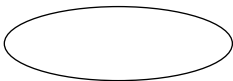



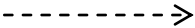

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

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. Adapun Simbol-simbol yang digunakan dalam *usecase* diagram, yaitu :

Tabel II.1 Simbol *Use Case*


Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>usecase</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>usecase</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 mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>usecase</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>usecase</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.



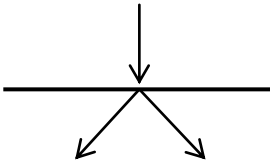
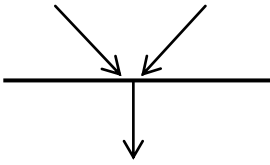
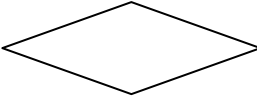

(Sumber : Windu Gata ; 2013 : 4)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Adapun 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.

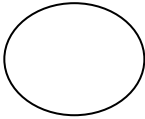
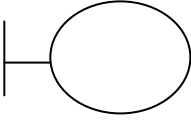
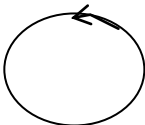

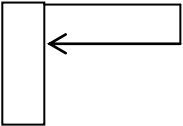


	<i>Endpoint</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>DecisionPoints</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

3. 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. Adapun simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.3 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)

4. *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.4 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)