

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pendukung Keputusan

Sistem pendukung keputusan (SPK) atau *Decision Support System* (DSS) adalah sistem informasi interaktif yang menyediakan informasi, pemodelan, dan manipulasi data yang digunakan untuk membantu pengambilan keputusan pada situasi yang semiterstruktur dan situasi yang tidak terstruktur di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. DSS dibuat sebagai reaksi atas ketidakpuasan terhadap TPS dan MIS. Sebagaimana diketahui, TPS lebih memfokuskan diri pada perekaman dan pengendalian transaksi yang merupakan kegiatan yang bersifat berulang dan terdefinisi dengan baik, sedangkan MIS lebih berorientasi pada penyediaan laporan bagi manajemen yang sifatnya tidak fleksibel (Abdul Kadir ; 2014 : 108).

Sistem pendukung keputusan (SPK) biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang. Aplikasi sistem pendukung keputusan (SPK) digunakan dalam pengambilan keputusan. Aplikasi sistem pendukung keputusan (SPK) menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur.

Sistem pendukung keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (respositori pengetahuan domain masalah yang ada pada sistem pendukung keputusan atau sebagai data atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan) (Dicky Nofriansyah ; 2014 : 1)

II.1.1. Karakteristik Sistem Pendukung Keputusan

Berikut ini adalah uraian beberapa karakteristik SPK, yaitu:

1. Kapabilitas interaktif

Yaitu SPK memberi pengambil keputusan akses cepat ke data dan informasi yang dibutuhkan.

2. Fleksibilitas

Yaitu SPK dapat menunjang para manajer pembuat keputusan di berbagai bidang fungsional (keuangan, pemasaran, operasi produksi, dan lain-lain).

3. Kemampuan menginteraksi model

Yaitu SPK memungkinkan para pembuat keputusan berinteraksi dengan model-model, termasuk memanipulasi model-model tersebut sesuai dengan kebutuhan.

4. Fleksibilitas output

Yaitu SPK mendukung para pembuat keputusan dengan menyediakan berbagai macam output, termasuk kemampuan grafik menyeluruh atas pertanyaan-pertanyaan pengandaian (Suhermin ; 2012 : 4).

II.1.2. Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan dibangun oleh lima komponen yaitu subsistem manajemen data, subsistem antarmuka pengguna, dan subsistem manajemen berbasis pengetahuan. Berdasarkan definisi, sistem pendukung keputusan harus mencakup tiga komponen utama yaitu subsistem manajemen model, subsistem manajemen moden dan antarmuka pengguna. Sistem Pendukung Keputusan disusun dari beberapa subsistem yaitu :

1. Subsistem manajemen data

Basis data yang relevan untuk suatu situasi dan dikelola oleh *database management system* (DBMS). Subsistem ini dapat diinterkoneksi dengan data *warehouse* perusahaan yang relavan untuk pengambilan keputusan.

2. Subsistem manajemen model

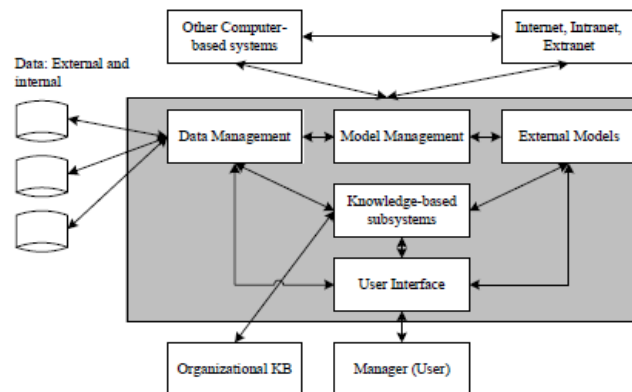
Merupakan paket perangkat lunak yang menyimpan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lainnya yang memberikan kemampuan analitik yang tepat. Perangkat lunak ini sering disebut *modelbase management system* (MBMS) dan dapat diimplementasikan pada sistem pengembangan web untuk berjalan pada *server* aplikasi.

3. Subsistem antarmuka pengguna

Merupakan dukungan komunikasi antara dengan pengguna. *Web browser* menjadi salah satu antarmuka yang menampilkan dalam bentuk grafis dan interaktif dengan pengguna.

4. Subsistem manajemen

Berbasis pengetahuan bertindak sebagai komponen independen yang memberikan kemampuan intelegensi untuk memperbesar pengetahuan pengambil keputusan. Perusahaan memiliki sistem manajemen pengetahuan. Keterhubungan subsistem ini dengan sistem pendukung keputusan dapat melalui interkoneksi dengan *web server* (Karismariyanti ; 2011 : 55).



Gambar II.1 Komponen Sistem Pendukung Keputusan
(Sumber : Karismariyanti ; 2011 : 55)

II.2. Metode *Weighted Product* (WP)

Metode *Weighted Product* (WP) menggunakan perkalian untuk menghubungkan rating atribut, dimana setiap atribut harus dipangkatkan dahulu dengan bobot atribut yang bersangkutan. Preferensi untuk alternatif A_i diberikan sebagai berikut :

$$S_i = \prod_{j=1}^n (X_{ij})^{W_j} \dots\dots\dots(1)$$

Keterangan :

Keterangan :

S_i = Preferensi alternatif dianalogikan sebagai vektor S

X_{ij} = Rating Alternatif setiap atribut

w_j = Bobot kriteria

n = Banyaknya kriteria

i = Alternatif

j = Atribut

$\prod_{j=1}^n X_{ij}$ = Perkalian Ranting alternatif per atribut dari $j=1$ - n Pada alternatif ini

dimana $\sum w_j = 1$

W_j adalah pangkat bernilai positif untuk atribut keuntungan , dan bernialai negatif untuk atribut biaya.

II.2.1. Langkah – Langkah Perhitungan Metode Weighted Product

Langkah-langkah dalam perhitungan metode Weighted Product (WP) adalah sebagai berikut :

1. Proses normalisasi bobot kriteria (W):

$$W_j = \frac{w_j}{\sum w_j} \dots\dots\dots(1)$$

Keterangan :

W_j : Bobot Atribut

$\sum w_j$: Penjumlahan Bobot Atribut

2. Preferensi untuk alternatif (S) diberikan :

$$S_i = \prod_{j=1}^n (X_{ij})^{w_j} \dots\dots\dots(2)$$

Keterangan :

S_i = Preferensi alternatif dianalogikan sebagai vektor S

X_{ij} = Rating Alternatif setiap atribut

w_j = Bobot kriteria

n = Banyaknya kriteria

i = Alternatif

j = Atribut

$\prod_{j=1}^n X_{ij}$ = Perkalian Ranting alternatif per atribut dari $j=1$ - n Pada alternatif ini
dimana $\sum w_j = 1$

W_j adalah pangkat bernilai positif untuk atribut keuntungan , dan bernialai negatif untuk atribut biaya.

3. Preferensi untuk alternatif (V) diberikan

$$V_i = \frac{\prod_{j=1}^n X_{ij} w_j}{\prod_{j=1}^n (X_j^*) w_j}$$

V_i = hasil Preferensi alternatif ke- i

X_{ij} = Rating Alternatif per atribut

W_j = Bobot atribut

i = Alternatif

j = Atribut

$\prod_{j=1}^n X_{ij}$ = Perkalian rating alternatif per atribut

$\prod_{j=1}^n (X_j^*) w_j$ = Penjumlahan hasil Perkalian rating alternatif per atribut.

II.3. 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.

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

II.4. Basis Data

Database (Basis data) merupakan kumpulan data yang saling berhubungan satu dengan lainnya yang tersimpan di perangkat keras komputer dan diperlukan suatu perangkat lunak untuk memanipulasi basis data tersebut. Buku telepon, katalog film merupakan contoh dari database. Sistem manajemen basis data (*database management sistem/DBMS*) adalah perangkat lunak yang digunakan untuk mengendalikan data, termasuk penyimpanan data, pengambilan data, keamanan data, dan integritas data. Fungsi utama DBMS adalah untuk menyediakan lingkungan yang

nyaman dan efisien untuk digunakan dalam pengambilan dan penyimpanan informasi di basis data.

Penekanan file pada basis data adalah kemampuan untuk mengakses data dengan cepat dan efisien dalam menggunakan media simpanan luarnya. Faktor yang mempengaruhi hal ini adalah organisasi dari file basis data. Organisasi file basis data ini mencoba meningkatkan struktur dari data antara satu file dengan file yang lainnya, dengan menunjukkan hubungan antardata yang ada :

- a. Struktur data berjenjang (*hierarchial data structure*) atau disebut juga dengan nama data pohon (*tree data structure*) menunjukkan hubungan antara data membentuk suatu jenjang seperti pohon.
- b. Struktur data jaringan (*network data structure*) disebut juga dengan plex data struktur. Pada struktur data pohon tiap-tiap node tidak dapat mempunyai lebih dari satu orang tua, sedangkan struktur data jaringan ini, tiap-tiap node punya banyak orang tua.
- c. Struktur data hubungan adalah meletakkan semua hubungan dalam bentuk tabel dua dimensi (Junidar ; 2012 : 19).

II.5. MySQL

MySQL adalah suatu sistem manajemen basis data relasional (RDBMS-*Relational Database Management System*) yang mampu bekerja dengan cepat, kokoh, dan mudah digunakan. Contoh RDBMS lain adalah *Oracle*, *Sybase*. Basis data memungkinkan anda untuk menyimpan, menelusuri, menurutkan dan mengambil data secara efisien. *Server MySQL* yang akan membantu melakukan fungsionalitas

tersebut. Bahasa yang digunakan oleh MySQL tentu saja adalah SQL-*standar* bahasa basis data relasional di seluruh dunia saat ini.

MySQL dikembangkan, dipasarkan dan disokong oleh sebuah perusahaan Swedia bernama MySQL AB. RDBMS ini berada di bawah bendera GNU GPL sehingga termasuk produk *Open Source* dan sekaligus memiliki lisensi komersial. Apabila menggunakan MySQL sebagai basis data dalam suatu situs Web. Anda tidak perlu membayar, akan tetapi jika ingin membuat produk RDBMS baru dengan basis MySQL dan kemudian menguainya, anda wajib bertemu dengan lisensi komersial (Antonius Nugraha Widhi Pratama ; 2010 : 10)

II.6. Normalisasi

Normalisasi adalah proses mengubah relasi dari bentuk tidak normal menjadi bentuk normal atau proses untuk mengidentifikasi dan menghilangkan anomali. Proses ini dilakukan dengan memecah sebuah relasi menjadi beberapa relasi lain yang lebih kecil, relasi yang dihasilkan memiliki jumlah atribut lebih sedikit. Tiga bentuk normal yaitu bentuk normal pertama (1NF), bentuk normal kedua (2NF), bentuk normal ketiga (3NF). Tetapi dalam perkembangan muncul bentuk-bentuk normal yang baru. Definisi tentang bentuk normal sebagai berikut:

1. Tahap tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

No_mhs	Nama	Prg_Studi	Kode_mk	Nama_mk	SKS	Kd_Dsn	Dosen
0231	Cahyo	I Komputer	PAM211	Kalkulus Lanjut I	3	MT002	Yasir
			PAAM261	Prg. Terstruktur I	3	IK003	Kamal

Gambar II.4. Tidak Normal
Sumber : Tawar ; 2011 : 7

2. Tahap normal tahap pertama (1st Normal Form)

Sebuah table disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut.
- Masing-masing cell bernilai tunggal

MHS(<u>No_mhs</u> , Nama, Prg_Studi)		
No_mhs	Nama	Prg_Studi
0231	Cahyo	I Komp.
0232	Hoho	Statistik
0233	Budi	Matematika

DAFTAR_MK(<u>No_mhs</u> , <u>Kode_mk</u> , Nama_Mk, SKS, Kd_Dsn, Dosen)					
No_mhs	Kode_mk	Nama_mk	SKS	Kd_Dsn	Dosen
0231	PAM211	Kalkulus Lanjut I	3	MT002	Yasir
0231	PAAM261	Prg. Terstruktur I	3	IK003	Kamal
0231	PAM367	Simulasi	3	IK002	Jack
0232	PAM333	Prg. Linier	3	MT003	Andri
0232	PAM241	Met. Statistik I	3	ST002	Fendi
0232	PAM345	Analisis Data	3	ST003	Hasbi
0233	PAM337	Fungsi Khas	3	MT001	Jaya
0233	PAM522	Topologi	3	MT003	Andri

Gambar II.5. Normalisasi 1NF
Sumber : Tawar ; 2011 : 8

3. Tahap normal tahap kedua (2nd normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

AMBIL(<u>No_mhs</u> , <u>Kode_mk</u>)	
No_mhs	Kode_mk
0231	PAM211
0231	PAAM261
0231	PAM367
0232	PAM333
0232	PAM241
0232	PAM345
0233	PAM337
0233	PAM522
0233	PAM432

PENGAJAR(<u>Kode_mk</u> , <u>Nama_mk</u> , <u>SKS</u> , <u>Kd_Dsn</u> , <u>Dosen</u>)				
Kode_mk	Nama_mk	SKS	Kd_Dsn	Dosen
PAM211	Kalkulus Lanjut I	3	MT002	Yasir
PAAM261	Prg. Terstruktur I	3	IK003	Kamal
PAM367	Simulasi	3	IK002	Jack
PAM333	Prg. Linier	3	MT003	Andri
PAM241	Met. Statistik I	3	ST002	Fendi

Gambar II.6. Normalisasi 2NF
Sumber : Tawar ; 2011 : 9

4. Tahap normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow Y \rightarrow Z$, dimana Y mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- X haruslah superkey pada tabel tersebut.
- Atau Y merupakan bagian dari primary key pada tabel tersebut.

KULIAH(Kode_mk, Nama_mk, SKS, Kd_Dsn)			
Kode_mk	Nama_mk	SKS	Kd_Dsn
PAM211	Kalkulus Lanjut I	3	MT002
PAAM261	Prg. Terstruktur I	3	IK003
PAM367	Simulasi	3	IK002
PAM333	Prg. Linier	3	MT003
PAM241	Met. Statistik I	3	ST002
PAM345	Analisis Data	3	ST003
PAM337	Fungsi Khas	3	MT001
PAM522	Topologi	3	MT003
PAM432	Teori Optimasi	3	MT004

Gambar II.7. Normalisasi 3NF
Sumber : Tawar ; 2011 : 10

5. Boyce Code Normal Form (BCNF)

- Memenuhi 1st NF
- Relasi harus bergantung fungsi pada atribut superkey

NILAI(No_mhs, No_Rkng, Kd_Mk, Nilai)		
No_mhs	Kode_mk	Nilai
0231	PAM211	A
0231	PAAM261	B
0231	PAM367	A
0232	PAM333	C
0232	PAM241	A
0233	PAM345	B
0233	PAM337	A
0235	PAM522	B
0237	PAM432	B

REKENING(No_mhs, No_Rkng)	
No_mhs	No_Rkng
0231	88681
0232	88682
0233	88683
0235	88685
0237	88687

Gambar II.8. Normalisasi BCNF
Sumber : Tawar ; 2011 : 12

6. Tahap Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF)

BAHASA(<u>No Mhs</u>, <u>Prg Studi</u>, <u>Bhs Asing</u>)		
No_mhs	Prg_Studi	Bhs_Asing
0232	Komputer	Inggris
0232	Akuntasnsi	Jerman
0232	Komputer	Jerman
0232	Akuntasnsi	Inggris
0236	Statistik	Perancis
0236	Hukum	Belanda
0236	Statistik	Belanda
0236	Hukum	Perancis
BAHASA2(<u>No Mhs</u>, <u>Prg Studi</u>, <u>Bhs Asing</u>)		
No_mhs	Prg_Studi	Bhs_Asing
0232	Komputer	Inggris
0232	Akuntasnsi	Jerman

Gambar II.9. Normalisasi 4NF
Sumber : Tawar ; 2011 : 13

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

Menurut 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.

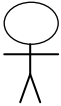
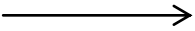
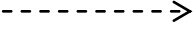
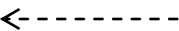
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu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

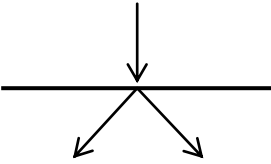
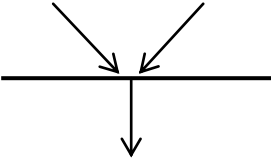
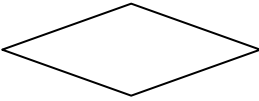
(Sumber : Gata, 2013 : 4)

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.

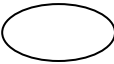
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , 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.

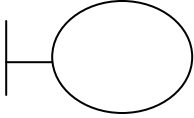
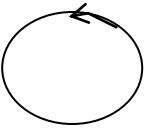
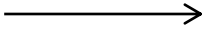
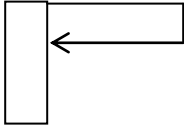
(Sumber : Gata, 2013 : 6)

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. 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 : 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 : Gata, 2013 : 9)