

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah sistem berbasis komputer yang membantu para pengambil keputusan mengatasi berbagai masalah melalui interaksi Pengambilan keputusan (*Decision Making*) adalah melakukan penilaian dan menjatuhkan pilihan. Keputusan ini diambil setelah melalui beberapa perhitungan dan pertimbangan alternatif. *Decision support system* atau sistem pendukung keputusan secara umum didefinisikan sebagai sebuah sistem yang mampu memberikan kemampuan baik kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah semi terstruktur. Secara khusus, Sistem pendukung keputusan didefinisikan sebagai sebuah sistem yang mendukung kerja seorang manager maupun sekelompok manager dalam memecahkan masalah semi terstruktur dengan cara memberikan informasi ataupun usulan menuju pada keputusan tertentu (Muhammad Yudin Ritonga ; 2014 : 19).

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 atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan (Dicky Nofriansyah ; 2014 : 1).

II.1.1. Karakteristik Sistem Pendukung Keputusan

Karakteristik dan kapabilitas sistem pendukung keputusan antara lain :

1. Dukungan untuk pengambilan keputusan, terutama pada situasi semi terstruktur dan tak terstruktur, dengan menyertakan penilaian manusia dan informasi terkomputerisasi. Masalah-masalah tersebut tidak dapat dipecahkan oleh sistem komputer lain atau oleh metode atau alat kuantitatif standar.
2. Dukungan untuk semua level manajerial, dari eksekutif puncak sampai manajer lini.
3. Dukungan untuk individu dan kelompok. Masalah yang kurang terstruktur sering memerlukan keterlibatan individu dari departemen dan tingkat organisasional yang berbeda atau bahkan dari organisasi lain. DSS mendukung im virtual melalui alat-alat Web kolaboratif.
4. Dukungan untuk keputusan independen dan atau sekuensial. Keputusan dapat dibuat satu kali, beberapa kali, atau berulang (dalam interval yang sama).
5. Dukungan disemua fase proses pengambilan keputusan : intelegensi, desain, pilihan, dan implementasi.

6. Dukungan diberbagai proses dan gaya pengambilan keputusan.
7. Adaptivitas sepanjang waktu. Pengambil keputusan seharusnya reaktif, dapat menghadapi perubahan kondisi secara cepat, dan dapat mengadaptasi DSS untuk memenuhi perubahan tersebut. DSS bersifat fleksibel dan karena itu pengguna dapat menambahkan, menghapus, menggabungkan, mengubah, atau menyusun kembali elemen-elemen dasar, DSS juga fleksibel dalam hal dapat dimodifikasi untuk memecahkan masalah lain yang sejenis.
8. Peningkatan terhadap keefektifan pengambilan keputusan (akurasi, *timeliness*, kualitas).
9. Kontrol penuh oleh pengambil keputusan terhadap semua langkah proses pengambilan keputusan dalam memecahkan suatu masalah. DSS secara khusus menekankan untuk mendukung pengambil keputusan, bukannya menggantikan (Gusti Ngurah Anom Cahyadi Putra ; 2011 : 37-38).

II.1.2. Komponen Sistem Pendukung Keputusan

Sistem Pendukung Keputusan disusun dari beberapa subsistem yaitu :

1. Subsistem manajemen data
Basis data yang relevan dan dikelola menggunakan *software* yang disebut *database management system* (DBMS)
2. Subsistem manajemen model
Adalah paket *software* yang berisi model-model yang disebut dengan *modelbase management system* (MBMS)
3. Subsistem manajemen pengetahuan
Subsistem yang memberikan intelegensi dan mendukung subsistem yang lain.

4. Subsistem antarmuka pengguna

Pengguna berkomunikasi dan memerintah SPK melalui *system* ini.

5. Pengguna

Orang yang berhadapan dengan pengambil keputusan (Gusti Ngurah Anom Cahyadi Putra ; 2011 : 37-38).

II.2. Pembelian

Pembelian adalah jumlah barang yang dibeli pada suatu periode. Pembelian dapat dilakukan secara tunai maupun kredit, yang masing-masing akan mendapatkan perlakuan berbeda. Biasanya untuk pembelian tunai mendapatkan potongan tunai. Selain itu, beberapa pembelian kredit juga diberikan syarat 2/10, n/30. Syarat 2/10 ini mempunyai arti apabila pembayaran dilakukan dalam tempo 10 hari sejak tanggal pembelian maka akan mendapatkan potongan sebesar 2%. Sementara n/30 berarti komitmen untuk melunasi pembayaran dalam tempo maksimal 30 hari.

Potongan pembelian biasa juga disebut potongan tunai pembelian. Potongan ini diberikan berupa pengurangan sejumlah harga dari harga pembelian. Seperti dijelaskan diatas, pemberian potongan ini dijelaskan melalui syarat pembayaran tertentu (Himayati ; 2011 : 25)

II.3. Metode Perbandingan Eksponensial

Metode Perbandingan Eksponensial (MPE) merupakan salah satu metode untuk menentukan urutan prioritas alternatif keputusan dengan kriteria jamak. Eksponensial Dalam menggunakan Metode Perbandingan Eksponensial ada

beberapa tahap yang harus dilakukan, yaitu: menyusun alternatif-alternatif keputusan yang akan dipilih, menentukan kriteria atau perbandingan keputusan yang penting untuk dievaluasi, menentukan tingkat kepentingan dari setiap kriteria keputusan, melakukan penilaian terhadap semua alternatif pada setiap kriteria, menghitung skor atau nilai total setiap alternatif, dan menentukan urutan prioritas keputusan didasarkan pada skor atau nilai total masing-masing alternatif (Imam Subarkah ; 2012 : 4).

II.3.1 Perhitungan Metode Perbandingan Eksponensial

Formulasi perhitungan skor untuk setiap alternatif dalam metode perbandingan eksponensial adalah sebagai berikut :

$$TN_i = \sum_{j=1}^m (RK_{ij})^{BJ}$$

(Sumber : Imam Subarkah ; 2012 : 4)

Dengan penjabaran :

Tni = Total nilai alternatif ke i

Rkij = Derajat Kepentingan relatif kriteria ke j pada pilihan keputusan i

TKKj = Derajat kepentingan kriteria keputusan ke j; TKKj > 0; bulat

n = Jumlah pilihan keputusan

m = Jumlah kriteria keputusan

Penentuan tingkat kepentingan kriteria dilakukan dengan cara wawancara dengan pakar atau melalui kesepakatan curah pendapat. Sedangkan Penentuan skor alternatif pada kriteria tertentu dilakukan dengan memberi nilai setiap alternatif berdasarkan nilai kriterianya.

Semakin besar nilai alternatif semakin besar pula skor alternatif tersebut. Total skor masing-masing alternatif keputusan akan relatif berbeda secara nyata karena adanya fungsi eksponensial. Fungsi eksponensial adalah salah satu fungsi yang paling penting dalam matematika, fungsi tersebut berfungsi sebagai bahan perbandingan untuk mengambil suatu keputusan (Imam Subarkah ; 2012 : 4).

II.3.2. Pengembangan Tahap Pengambilan

Keputusan dengan Metode Perbandingan Eksponensial Dengan melihat tahapan-tahapan Metode Perbandingan Eksponensial diatas maka penulis akan mengembangkan lagi sesuai dengan kebutuhan dalam pembuatan sistem.

a. Menyusun alternatif-alternatif keputusan yang akan dipilih

Alternatif-alternatif keputusan yang akan digunakan dalam Sistem Informasi Pendukung Keputusan ini adalah nilai – nilai pendukung yang sudah ada, dengan maksud mengambil beberapa kebutuhan dalam pengambil keputusan dengan bahan yang sudah ada sebelumnya.

b. Menentukan kriteria atau perbandingan keputusan yang penting untuk di evaluasi.

Dari data pada poin pertama maka ditentukanlah kriteria atau perbandingan keputusan yang akan dievaluasi, dan dapat dijadikan sebagai alternatif dalam melakukan pemilihan provider.

- c. Menentukan tingkat kepentingan dari setiap kriteria keputusan

Setelah didapatkan alternatif-alternatif keputusan dan kriteria sebagai penilaian, tahap ketiga dalam Metode Perbandingan Eksponensial ini adalah menentukan tingkat kepentingan dari setiap kriteria keputusan, yaitu dilakukan pemberian bobot penilaian, Penentuan tingkat kepentingan kriteria dilakukan dengan cara wawancara dengan pakar atau melalui kesepakatan curah pendapat (Imam Subarkah ; 2012 : 4).

II.3.3. Analisa Perbandingan Metode Perbandingan Eksponensial

1. Menentukan Alternatif

Untuk menganalisa perbandingan kecepatan antara Algoritma Brute Force dengan Algoritma Boyer Moore dalam pencarian Word Suggestion, adapun yang menjadi Alternatif adalah Algoritma Brute Force dan Algoritma Boyer Moore.

2. Menentukan Kriteria

Untuk dapat membandingkan kedua alternatif diatas, maka selanjutnya yang perlu dilakukan adalah menentukan kriteria dalam analisa ini.

3. Menentukan Bobot Kriteria

Penentuan bobot merupakan salah satu komponen yang sangat berpengaruh terhadap hasil dari analisa, untuk itu penulis menetapkan bobot kriteria berdasarkan tingkat pengaruh dalam menentukan kecepatan dari algoritma dalam pencarian *Word Suggestion*.

4. Pemberian Nilai Pada Setiap Kriteria

Tahap ini adalah tahap dimana setiap kriteria yang telah terbentuk diberi nilai. Untuk dapat memberikan nilai, berikut adalah contoh hasil simulasi pencarian *Word Suggestion* yang diambil dari pembahasan analisa Algoritma *Brute Force* dan analisa Algoritma *Boyer Moore* sebelumnya.

5. Menghitung Skor

Setelah nilai pada setiap kriteria dimasukkan, maka tahapan selanjutnya adalah melakukan perhitungan dengan menggunakan rumus Metode Perbandingan Eksponensial.

6. Menentukan Prioritas Keputusan

Setelah Total Nilai dari setiap alternatif dihitung maka yang tahapan selanjutnya adalah tahapan terakhir yaitu menentukan prioritas keputusan berdasarkan total nilai dari setiap alternatif (Andri Januardi ; 2013 : 22).

II.4. Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai jenis komputer dan berbagai sistem operasi termasuk telepon genggam. Java dikembangkan oleh *Sun Microsystem* dan dirilis tahun 1995. Java merupakan suatu teknologi perangkat lunak yang digolongkan *multi platform*. Selain itu, Java juga merupakan suatu *platform* yang memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan suatu program.

Bahasa pemrograman java pertama lahir dari *The Green Project*, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek

tersebut belum menggunakan *versi* yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, serta Sembilan pemrograman lainnya dari *Sun Microsystem*. Salah satu hasil proyek ini adalah mascot Duke yang dibuat oleh Joe Palrang (Wahana Komputer ; 2010 : 1).

II.5. NetBeans

NetBeans merupakan salah satu proyek *open source* yang disponsori oleh *Sun Microsystem*. 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.6. 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 *multi user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* 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*) (Wahana Komputer; 2010 : 5).

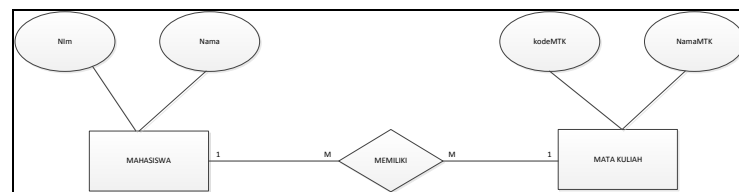
II.7. Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau ERD adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antarentitas. Proses memungkinkan analisis menghasilkan struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien (Janner Simarmata ; 2010 : 67).

Tabel II.1. Simbol ERD

Notasi	Keterangan
	Entitas , adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi , menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut , berfungsi mendeskripsikan karakter entitas (atribut yg berfungsi sebagai key diberi garis bawah)
	Garis , sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

(Sumber : Janner Simarmata ; 2010 : 67)



Gambar. II.1. Diagram ERD

(Sumber : Janner Simarmata ; 2010 : 60)

II.8. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data 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.8.1. 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.

Contoh normalisasi 1NF adalah seperti pada table berikut :

Tabel 2. Tabel Bentuk Normal Pertama (1NF)

p#	Status	Kota	b#	qty
p1	20	Yogyakarta	b1	300
p1	20	Yogyakarta	b2	200
p1	20	Yogyakarta	b3	400
p1	20	Yogyakarta	b4	200
p1	20	Yogyakarta	b5	100
p1	20	Yogyakarta	b6	100
p2	10	Medan	b1	300
p2	10	Medan	b2	400
p3	10	Medan	b2	200
p4	20	Yogyakarta	b2	200
p4	20	Yogyakarta	b4	300
p4	20	Yogyakarta	b5	400

(Sumber : Janner Simarmata ; 2010 : 79)

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.

Tabel 3. Tabel Bentuk Normal Kedua (2NF)

Pemasok2			Barang		
p#	status	Kota	p#	b#	qty
P1	20	Yogyakarta	p1	b1	300
P2	10	Medan	p1	b2	200
P3	10	Medan	p1	b3	400
P4	20	Yogyakarta	p1	b4	200
P5	30	Bandung	p1	b5	100
			p1	b6	100
			p2	b1	300
			p2	b2	400
			p3	b2	200
			p4	b2	200
			p4	b4	300
			p4	b5	400

(Sumber : Janner Simarmata ; 2010 : 82)

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 tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

Tabel 4. Tabel Bentuk Normal Ketiga (3NF)

Pemasok Kota		Kota Status	
p#	Kota	Kota	status
P1	Yogyakarta	Yogyakarta	20
P2	Medan	Medan	10
P3	Medan	Yogyakarta	20
P4	Yogyakarta	Bandung	30
P5	Bandung		

(Sumber : Janner Simarmata ; 2010 : 83)

4. *Boyce Code Normal Form (BCNF)*

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF(Janner Simarmata ; 2010 : 85).

5. **Bentuk Normal Keempat (4NF)**

Sebuah tabel rasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD). Sebuah ketergantungan multivalued tiga kolom, satu kolom mempunyai banyak baris bernilai sama, tetapi kolom lain bernilai berbeda.

Tabel 5. Tabel Bentuk Normal Keempat (4NF)

Pegawai Proyek		Pegawai Ahli	
peg#	Pry#	Peg#	Ahli
1211	P1	1211	Analisis
1211	P3	1211	Perancangan
		1211	Pemrograman

(Sumber : Janner Simarmata ; 2010 : 85)

6. **Bentuk Normal Kelima**

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel 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 : 78).

Tabel 6. Tabel Bentuk Normal Kelima (5NF)

peg#	Pry#	Ahli
1211	11	Perancangan
1211	28	Pemrograman

(Sumber : Janner Simarmata ; 2010 : 86)

II.9. 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.


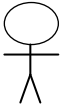

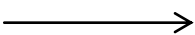
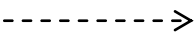
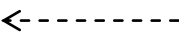
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

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.7. Simbol Use Case




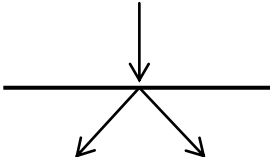
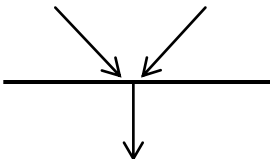
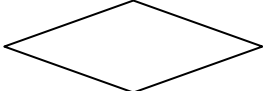
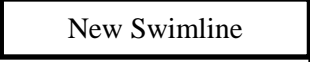
Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang 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 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>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.8. Simbol *Activity Diagram*

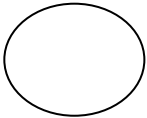
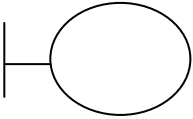
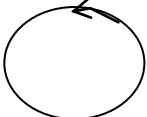
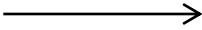
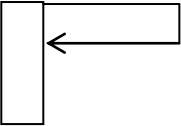


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel 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. 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.9. 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.10. *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)