

BAB II

TINJAUAN PUSTAKA

II.1. 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 dataprocessing (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. 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. Konsep DSS dikemukakan pertama kali oleh Scott-Morton pada tahun 1971. Beliau mendefinisikan cikal bakal DSS tersebut sebagai “Sistem berbasis komputer yang interaktif, yang membantu pengambil keputusan menggunakan data dan model untuk memecahkan persoalan-persoalan tidak terstruktur” (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 atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan (Nofriansyah ; 2014 : 1).

II.2.1. Definisi Sistem Pendukung Keputusan

Definisi lain mengenai sistem pendukung keputusan (SPK) yang ideal yaitu :

- a. SPK adalah sebuah sistem berbasis komputer dengan antarmuka antara mesin/komputer dan pengguna.
- b. SPK ditujukan untuk membantu pembuat keputusan dalam menyelesaikan suatu masalah dalam berbagai level manajemen dan bukan untuk mengganti posisi manusia sebagai pembuat keputusan.
- c. SPK mampu memberi alternatif solusi bagi masalah semi/tidak terstruktur baik bagi perseorangan atau kelompok dan dalam berbagai macam proses dan gaya pengambilan keputusan.
- d. SPK menggunakan data, basis data dan analisa model-model keputusan.
- e. SPK bersifat adaptif, efektif, interaktif , *easy to use* dan fleksibel
- f. SPK menyediakan akses terhadap berbagai macam format dan tipe sumber data (*data source*) (Sulistiyo ; 2012 : 2).

II.2.2. Fase Pengambilan Keputusan

Model adalah representasi/abstraksi sederhana dari realitas karena realitas terlalu kompleks dan tidak relevan untuk memecahkan masalah khusus. Proses pemodelan menggunakan empat fase pengambilan keputusan yaitu fase inteligensi, fase desain, fase pilihan, dan fase implementasi.

a. Fase Inteligensi

Inteligensi meliputi pemindaian lingkungan pada suatu waktu tertentu maupun secara periodik yang mencakup identifikasi masalah atau peluang masalah maupun monitoring hasil dari fase implementasi. Keluaran dari fase ini adalah pernyataan masalah.

b. Fase Desain

Fase desain meliputi penemuan atau pengembangan dan menganalisis tindakan yang mungkin dilakukan. Sebuah model pengambilan keputusan dibangun, diuji dan divalidasi.

c. Fase Pemilihan

Fase pemilihan meliputi pencarian evaluasi dan rekomendasi terhadap suatu solusi yang tepat untuk model. Solusi untuk model menghasilkan sebuah solusi yang direkomendasikan untuk masalah. Masalah dianggap dipecahkan hanya jika solusi yang direkomendasikan sukses diterapkan pada fase berikutnya.

d. Fase Implementasi

Implementasi dapat diartikan membuat suatu solusi yang direkomendasikan bisa bekerja (Karismariyanti ; 2011 : 56).

II.2.3. 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 denagn pengguna. *Web browser* menjadi salah satu antarmuka yang manampilkan 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).

II.3. Metode *Smart*

SMART merupakan metode dalam pengambilan keputusan multiatribut. Teknik pengambilan keputusan multiatribut ini digunakan untuk mendukung pembuat keputusan dalam memilih beberapa alternatif. Setiap pembuat keputusan harus memiliki sebuah alternatif yang sesuai dengan tujuan yang dirumuskan. Setiap alternatif terdiri dari sekumpulan atribut dan setiap atribut mempunyai nilai-nilai. Nilai ini dirata-rata dengan skala tertentu. Setiap atribut mempunyai bobot yang menggambarkan seberapa penting skala tertentu. Setiap atribut mempunyai bobot yang menggambarkan seberapa penting suatu atribut dibandingkan dengan atribut lain. Pembobotan dan pemberian peringkat ini digunakan untuk menilai setiap alternatif agar diperoleh alternatif terbaik.

SMART menggunakan linier adaptif model untuk meramal nilai setiap alternatif. SMART lebih banyak digunakan karena kesederhanaannya dalam merespon kebutuhan pembuat keputusan dan caranya menganalisa respon. Analisis yang terbaik adalah transparan sehingga metode ini memberikan pemahaman masalah yang tinggi dan dapat diterima oleh pembuat keputusan.

Pembobotan pada SMART menggunakan skala 0 sampai 1, sehingga mempermudah perhitungan dan perbandingan nilai pada masing-masing alternatif.

Model yang digunakan dalam SMART yaitu (Rika Yunitarini ; 2013 : 46) :

$$\text{Maximize } \sum_{j=1}^k w_j \cdot u_{ij} , \forall i = 1 \text{ to } n \dots\dots\dots(1)$$

II.3.1. Langkah-Langkah Perhitungan Metode *Smart*

Adapun teknik atau langkah-langkah dalam proses SMART, antara lain :

1. Identifikasi user yang nantinya bertanggung jawab dalam mengambil keputusan.
2. Identifikasi Permasalahan yang ada Dengan melihat akar permasalahan dan batasan-batasan yang ada agar nantinya tidak menyimpang dari tujuan yang ingin dicapai.
3. Identifikasi alternatif yang ada untuk mencapai tujuan dari sistem yang akan dibuat.
4. Identifikasi kriteria-kriteria yang akan mendukung pengambilan keputusan
5. Memberikan peringkat atau bobot untuk setiap kriteria. Pemberian peringkat atau bobot ini ditentukan oleh user dan sistem akan memberikan bobot skala default dari peringkat yang dimasukkan.
6. Memberikan penilaian setiap kriteria untuk setiap alternatif. Juga dilakukan oleh user dimana penilaian setiap kriteria dapat bersifat kualitatif maupun kuantitatif untuk setiap alternatifnya.
7. Mengembangkan Single-Attribute Utilities yang mencerminkan seberapa baik setiap alternatif dilihat dari setiap kriteria. Tahap ini adalah memberikan suatu

nilai pada semua kriteria untuk setiap alternatif dengan nilai yang berskala 0 sampai 1.

8. Menghitung penilaian terhadap setiap alternative (Rika Yunitarini ; 2013 : 46).

II.4. Metode Weight Product

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 :

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

Keterangan :

S = Preferensi alternatif dianalogikan sebagai vektor S

X_{ij} = Nilai variabel dari alternatif pada setiap atribut

w_j = Nilai bobot kriteria

n = Banyaknya kriteria

i = Nilai alternatif

j = Nilai kriteria (Lusiana Dwi Jayanti ; 2014 ; 3).

II.2.1. Langkah – Langkah Perhitungan Metode Weighted Product

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

1. Siapkan terlebih dahulu bobot dan lakukan perbaikan bobot dengan $W_j =$

$$W_j / \sum W_j$$

2. Normalisasi matriks X berdasarkan persamaan

$$S_i = \prod_{j=1}^n (X_{ij})^{w_j} \text{ sehingga diperoleh matriks ternormalisasi } S$$

3. w_j adalah pangkat bernilai positif untuk atribut keuntungan, dan bernilai negatif untuk atribut biaya.
4. Lakukan perangkingan dengan menggunakan persamaan :

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

5. Ditemukan urutan alternatif terbaik yang akan menjadi keputusan

II.5. Pengertian Visual Basic

VB.NET adalah salah satu bahasa pemrograman tingkat tinggi yang mendekati bahasa manusia. Kemunculan bahasa VB.NET ini sebagai jawaban untuk menyederhanakan bahasa pemrograman pada platform .NET yang diluncurkan tahun 2002 dan untuk menjembatani programmer Visual Basic. Bahasa VB.NET secara teknis mengadopsi sintak bahasa Visual Basic. Konsistensi API membuat bahasa VB.NET menjadi pilihan dalam membuat kode program diatas platform Windows. Fitur baru bahasa VB.NET dibandingkan Visual Basic bahwa bahasa VB.NET mendukung object-oriented dan juga dynamics programming. Ini menambah daftar kemudahan untuk belajar bahasa VB.NET.

Ibaratnya seperti ikan dan air yang tidak dipisahkan, ini sama halnya pada VB.NET dan .NET Framework. Bahasa VB.NET memerlukan .NET Framework agar dapat dikompilasi dan dijalankan. .NET Framework merupakan framework

yang membungkus kompleksitas OS Windows sehingga konsisten API dapat diperoleh dan tidak dipusingkan dengan beragam API tiap OS Windows. Buku ini tidak akan membahas .NET Framework. Pembaca dapat mempelajari buku yang khusus belajar mengenai .NET Framework. Pembaca juga dapat mengunjungi website resminya yaitu <http://www.microsoft.com/net> (Kurniawan ; 2013 : 10).

II.6. Pengertian SQL Server 2008

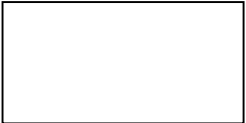
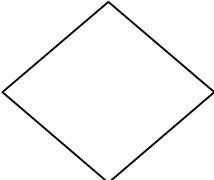
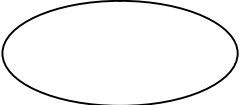

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang *database*. SQL Server adalah DBMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. SQL Server 2008 dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data. Microsoft merilis SQL Server 2008 dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sebagai berikut (Widya ; 2012 : 3)

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 (Simarmata ; 2010 : 67).

Tabel II.1. Simbol ERD

Simbol	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 yang berfungsi sebagai key diberi garis bawah)
	Garis, sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut

(Sumber : Janner Simarmata ; 2010 : 67)

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

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

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		

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.

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

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

II.9. 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.

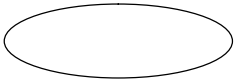
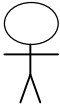

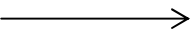
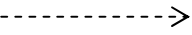
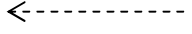
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu 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.7. Simbol Use Case




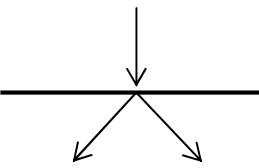
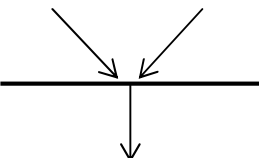
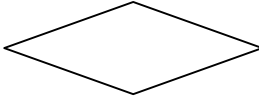
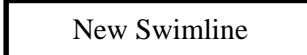
Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang 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>.</p>
	<p>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.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><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.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

(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.8. Simbol *Activity Diagram*

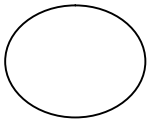
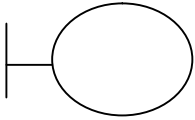


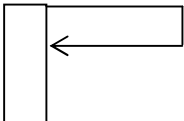


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan 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> .
 New Swimlane	<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.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 : 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 : Gata, 2013 : 9)