

BAB II

LANDASAN TEORI

II.1. Sistem Informasi

Sistem Informasi adalah suatu sistem yang tujuannya menghasilkan informasi. Sebagai suatu sistem untuk dapat memahami sistem informasi, akan lebih baik jika konsep dari sistem dipahami terlebih dahulu. Tujuan dari sistem informasi adalah menghasilkan informasi. Informasi adalah data yang diolah menjadi bentuk yang berguna bagi para pemakainya. (Jogiyanto HM ; 2009 : 36).

Komponen – komponen sistem informasi terdiri dari :

1. Komponen *Input*

Input merupakan data yang masuk ke dalam sistem informasi. Komponen ini merupakan bahan dasar dalam pengolahan informasi. Sistem informasi tidak akan dapat menghasilkan informasi jika tidak mempunyai komponen *input*. *Input* yang masuk ke dalam sistem informasi dapat langsung diolah menjadi informasi.

2. Komponen *Output*

Output merupakan komponen yang harus ada di sistem informasi. Sistem informasi yang tidak pernah menghasilkan output, tetapi selalu menerima input dikatakan bahwa *input* yang diterima masuk ke dalam lubang yang dalam (*deep hole*). *Output* dari sistem informasi dibuat dengan menggunakan data yang ada di basis data dan diproses menggunakan model yang tertentu.

3. Komponen Basis Data (*Database*)

Basis Data (Database) adalah kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

4. Komponen Model

Informasi yang dihasilkan oleh sistem informasi berasal dari data yang diambil dari basis data yang diolah lewat suatu model-model tertentu. Model- model yang digunakan di sistem informasi dapat berupa model logika yang menunjukkan suatu proses perbandingan logika atau model matematik yang menunjukkan proses perhitungan matematika. (Jogiyanto HM ; 2009 : 49).

II.2. Sistem Pendukung Keputusan

Sistem pendukung keputusan didefinisikan sebagai sistem yang digunakan untuk mendukung dan membantu pihak manajemen melakukan pengambilan keputusan pada semiterstruktur dan tidak terstruktur. Pada dasarnya konsep sistem pendukung keputusan hanyalah sebatas pada kegiatan membantu para manajer melakukan penelitian serta menggantikan posisi serta peran manajer. Sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan yang memanipulasi data. Sistem ini digunakan untuk membantu mengambil keputusan dalam situasi yang semi terstruktur dan tak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat.

Sistem pendukung keputusan adalah berbasis komputer sistem informasi yang mengandung bisnis atau organisasi pengambilan keputusan kegiatan. Sistem pendukung keputusan melayani manajemen, operasional, dan tingkat perencanaan organisasi (manajemen biasanya pertengahan dan lebih tinggi) dan membantu untuk membuat keputusan, yang mungkin cepat berubah dan tidak mudah ditentukan dimuka (*unstructured* dan masalah keputusan semiterstruktur). Sistem pendukung keputusan dapat berupa sepenuhnya terkomputerisasi, manusia atau keduanya. Sementara secara akademis telah dianggap, sistem pendukung keputusan sebagai alat untuk mendukung proses keputusan, pengguna sistem pendukung keputusan melihat pengguna sistem pendukung sebagai alat untuk memfasilitasi proses organisasi. (Novita Endah Wulandari ; 2015 : 179) .

II.2.1. Tahap – Tahap dalam Sistem Penunjang Keputusan

Penyusunan sebuah tahapan keputusan merupakan suatu cara untuk mengembangkan hubungan-hubungan logis yang mendasarin persoalan keputusan kedalam suatu bentuk model matematis yang mencerminkan hubungan di antara faktor-faktor yang terlibat. Proses ini terdiri dari empat fase, yaitu :

1. Penelusuran (*Intelligence*)

Tahap ini merupakan tahap pendefinisian masalah serta indentifikasi informasi yang dibutuhkan yang berkaitan dengan persoalan yang dihadapi serta keputusan yang akan diambil

2. Perancangan (*Design*)

Tahap ini merupakan suatu proses untuk merepresentasikan model sistem yang akan dibangun berdasarkan pada asumsi yang telah ditetapkan. Dalam hal ini, suatu model dari masalah dibuat, diuji, dan divalidasi.

3. Pemilihan (*Choice*)

Tahap ini merupakan suatu proses melakukan pengujian dan memilih keputusan terbaik berdasarkan kriteria tertentu yang telah ditentukan dan mengarah kepada tujuan yang akan dicapai.

4. Implementasi (*Implementation*)

Tahap ini merupakan tahap pelaksanaan dari keputusan yang telah diambil. Pada tahap ini perlu disusun tindakan yang terencana, sehingga hasil keputusan dapat dipantau dan disesuaikan apabila diperlukan perbaikan-perbaikan. (Novita Endah Wulandari ; 2015 : 180) .

II.2.2. Prosedur Pengambilan Keputusan

Ada beberapa prosedur pengambilan keputusan yaitu sebagai berikut :

1. Mengidentifikasi masalah.
2. Mengklarifikasi tujuan-tujuan khusus yang diinginkan.
3. Memeriksa berbagai kemungkinan untuk mencapai tujuan yang telah ditetapkan.

4. Mengakhiri proses itu dengan menetapkan pilihan bertindak dengan dasar fakta dan nilai (keputusan dan kemungkinan yang timbul). (Novita Endah Wulandari ; 2015 : 180) .

II.2.3. Tujuan dari Sistem Pendukung Keputusan

Banyaknya definisi yang dikemukakan mengenai pengertian dan penerapan dari sebuah sistem pendukung keputusan, maka terdapat tujuan sistem pendukung keputusan yaitu sebagai berikut :

1. Membantu manajer dalam pengambilan keputusan atas masalah semi terstruktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektivitas keputusan yang diambil manajer lebih dari perbaikan efisiensinya.
4. Kecepatan komputasi, komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya rendah.
5. Peningkatan produktivitas, membangun satu kelompok pengambil keputusan, terutama para pakar bisa sangat mahal. Pendukung terkomputerisasi bias mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berada diberbagai lokasi yang berbeda-beda (menghemat biaya perjalanan).
6. Dukungan kualitas, komputer bisa meningkatkan kualitas keputusan yang dibuat.

7. Berdaya saing.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.
(Novita Endah Wulandari ; 2015 : 180).

II.3. Metode *Analytical Hierarchy Process* (AHP)

II.3.1. *Analytical Hierarchy Process* (AHP)

AHP adalah sebuah metode memecah permasalahan yang kompleks/ rumit dalam situasi yang tidak terstruktur menjadi bagian-bagian komponen. Mengatur bagian atau variabel ini menjadi suatu bentuk susunan hierarki, kemudian memberikan nilai numerik untuk penilaian subjektif terhadap kepentingan relatif dari setiap variabel dan mensintesis penilaian untuk variabel mana yang memiliki prioritas tertinggi yang akan mempengaruhi penyelesaian dari situasi tersebut. AHP menggabungkan pertimbangan dan penilaian pribadi dengan cara yang logis dan dipengaruhi imajinasi, pengalaman, dan pengetahuan untuk menyusun hierarki dari suatu masalah yang berdasarkan logika, intuisi dan juga pengalaman untuk memberikan pertimbangan. AHP merupakan suatu proses mengidentifikasi, dan memberikan perkiraan interaksi sistem secara keseluruhan. (Tominanto ; 2012 : 3).

II.3.2. Prosedur Kegiatan *Analytical Hierarchy Process* (AHP)

Prosedur dalam metode AHP terdiri dari beberapa tahap , yaitu:

1. Menyusun hirarki dari permasalahan yang dihadapi.

Penyusunan hirarki yaitu dengan menentukan tujuan yang merupakan sasaran sistem secara keseluruhan pada level teratas. Level berikutnya terdiri dari kriteria-kriteria untuk menilai atau mempertimbangkan alternatif-alternatif yang ada dan menentukan alternatif-alternatif tersebut. setiap kriteria dapat memiliki subkriteria dibawahnya dan setiap kriteria dapat memiliki nilai intensitas masing-masing.

2. Menentukan prioritas elemen.

a. Langkah pertama dalam menentukan prioritas elemen adalah membuat perbandingan berpasangan, yaitu membandingkan elemen secara berpasangan sesuai kriteria yang di berikan dengan menggunakan bentuk matriks. Matriks bersifat sederhana, berkedudukan kuat yang menawarkan kerangka untuk memeriksa konsistensi, memperoleh informasi tambahan dengan membuat semua perbandingan yang mungkin dan menganalisis kepekaan prioritas secara keseluruhan untuk merubah pertimbangan. Proses perbandingan berpasangan dimulai dari level paling atas hirarki untuk memilih kriteria, misalnya C, kemudian dari level dibawahnya diambil elemen-elemen yang akan dibandingkan, misal A1, A2, A3, A4, maka susunan elemen-elemen pada sebuah matrik seperti Tabel II.1.

Tabel II.1 Matriks Perbandingan Berpasangan

C	A1	A2	A3	A4
A1	1			

A2		1		
A3			1	
A4				1

(Sumber : Tominanto ; 2012 : 3).

b. Mengisi matrik perbandingan berpasangan yaitu dengan menggunakan bilangan untuk merepresentasikan kepentingan relatif dari satu elemen terhadap elemen lainnya yang dimaksud dalam bentuk skala dari 1 sampai dengan 9. Skala ini mendefinisikan dan menjelaskan nilai 1 sampai 9 untuk pertimbangan dalam perbandingan berpasangan elemen pada setiap level hirarki terhadap suatu kriteria di level yang lebih tinggi. Apabila suatu elemen dalam matrik dan dibandingkan dengan dirinya sendiri, maka diberi nilai 1. Jika i dibanding j mendapatkan nilai tertentu, maka j dibanding i merupakan kebalikkannya. Berikut ini skala kuantitatif 1 sampai dengan 9 untuk menilai tingkat kepentingan suatu elemen dengan elemen lainnya.

Tabel II.2 Skala Penilaian Perbandingan Berpasangan

Intensitas Kepentingan	Arti/Makna	Penjelasan
1	Kedua elemen sama pentingnya	Dua elemen mempunyai pengaruh yang sama besar terhadap tujuan
3	Elemen yang satu sedikit lebih penting dari pada elemen yang lainnya.	Pengalaman dan penilaian sedikit menyokong satu elemen dibandingkan elemen yang lainnya
5	Elemen yang satu lebih penting dari pada elemen yang lainnya.	Pengalaman dan penilaian sangat kuat menyokong satu elemen dibandingkan elemen yang lainnya.
7	Satu elemen jelas lebih mutlak penting dari pada elemen yang lainnya.	Satu elemen yang kuat di sokong dan dominan terlihat dalam praktek
9	Satu elemen mutlak penting	Bukti yang mendukung elemen yang satu terhadap elemen lain memiliki tingkat

	dari pada elemen yang lainnya.	penegasan tertinggi yang mungkin Menguatkan
2, 4, 6, 8	Nilai-nilai antara 2 nilai pertimbangan yang berdekatan.	Nilai ini diberikan bila ada dua kompromi diantara 2 pilihan
Kebalikan	Jika aktifitas i mendapat satu angka dibanding aktifitas j, maka j mempunyai nilai kebalikkannya dibanding dengan i	

(Sumber : Tominanto ; 2012 : 4).

c. Sintesis.

Pertimbangan-pertimbangan terhadap perbandingan berpasangan di sintesis untuk memperoleh keseluruhan prioritas.

- 1) Menjumlahkan nilai-nilai dari setiap kolom pada matriks.
- 2) Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks.
- 3) Menjumlahkan nilai dari setiap matriks dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata.
- 4) Mengukur konsistensi.

Konsistensi penting untuk mendapatkan hasil yang valid dalam dunia nyata. AHP mengukur konsistensi pertimbangan dengan rasio konsistensi (consistency ratio). Nilai Konsistensi rasio harus kurang dari 5% untuk matriks 3x3, 9% untuk matriks 4x4 dan 10% untuk matriks yang lebih besar. Jika lebih dari rasio dari batas tersebut maka

nilai perbandingan matriks di lakukan kembali. Langkah-langkah menghitung nilai rasio konsistensi yaitu:

- a) Mengkalikan nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua, dan seterusnya.
- b) Menjumlahkan setiap baris.
- c) Hasil dari penjumlahan baris dibagi dengan elemen prioritas relatif yang bersangkutan.
- d) Membagi hasil diatas dengan banyak elemen yang ada, hasilnya disebut eigen value ($\lambda_{\max}$).
- e) Menghitung indeks konsistensi (consistency index) dengan rumus :

$$CI = (\lambda_{\max} - n) / n$$

Dimana CI : Consistensi Index

$\lambda_{\max}$: Eigen Value

n : Banyak elemen

- f) Menghitung konsistensi ratio (CR) dengan rumus :

$$CR = CI / IR$$

Dimana CR : *Consistency Ratio*

CI : *Consistency Index*

IR : *Index Random Consistency*

(Sumber : Tominanto ; 2012 : 5)

g) Memeriksa Konsistensi Hierarki.

Jika nilainya lebih dari 100%, maka penilaian data judgment harus diperbaiki. Namun jika rasio konsistensi (CI/IR) kurang atau sama dengan 0,1, maka hasil perhitungan bisa dinyatakan (Jurnal Ilmiah Rekam Medis dan Informatika Kesehatan).

Matriks random dengan skala penilaian 1 sampai 9 beserta kebalikkannya sebagai *index random consistency* (IR).

Berdasarkan perhitungan saaty menggunakan 500 sampel, jika pertimbangan memilih secara acak dari skala 1/9, 1/8, ..., 1, 2, ..., 9 akan diperoleh rata-rata konsistensi untuk matriks yang berbeda seperti pada Tabel II.3.

Tabel II.3 Daftar Nilai Rata-Rata Konsistensi

Ukuran Matriks	Konsistensi Acak (Nilai IR)
1	0,00
2	0,00
3	0,58
4	0,90
5	1,12
6	1,24
7	1,32
8	1,41

9	1,45
10	1,49

(Sumber : Tominanto ; 2012 : 5).

II.4. Database

Database merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan diperangkat lunak untuk memanipulasinya. *Database* merupakan salah satu komponen yang sangat penting dalam suatu sistem informasi, karena merupakan basis sistem dalam menyediakan informasi bagi para pemakai.

Sistem manajemen data merupakan perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dan utility data dalam jumlah yang besar. *Database* juga merupakan kumpulan data –data yang mempunyai fasilitas akses yang berhubungan satu sama yang lainnya sehingga dapat mengelola data-data tersebut.

Database ada dua jenis yaitu :

1. Database Hierarki

Yaitu suatu data yang tersusun dengan bentuk hirarki pohon. Susunan yang seperti ini terdiri dari beberapa unsur komponen yang saling mempengaruhi dan tidak dapat dipisahkan, jenis *database* ini merupakan hubungan satu komponen dengan banyak komponen .

2. Database relasi

Adalah suatu data yang disusun dalam bentuk tabel yang terdiri dari dua definisi dan tersusun secara terstruktur. Bentuk susunan dua dimensi ini terdiri dari beberapa kolom dan *record* yang tersusun berbentuk baris dari kiri kekanan. Data-data yang susunannya berbentuk baris adalah susunan yang menurun kebawah. Dimana pada setiap baris berisikan data-data yang saling berkaitan satu sama lainnya. Artinya setiap pemasukan data yang tersimpan pada field merupakan kesatuan dalam bentuk satu baris

Komponen utama dari sistem database terdiri dari antara lain :

1. Data yaitu diutamakan data yang bersifat *integrity* (kesatuan) dan *share* (pemakaian bersama)
2. *Hardware* yaitu semua yang menyangkut media penyimpanan eksternal, piranti input dan output .
3. *Software* yaitu berupa *database* manajemen sistem seperti SQL, yang merupakan penghubung antara alumni dengan data yang tersimpan didalam media penyimpanan secara fisik.
4. *Database* yaitu merupakan kumpulan dari *file* yang saling berhubungan satu dengan yang lainnya. *Database* secara fisik terdapat dalam media penyimpanan seperti sistem komputer. (Indra Warman dkk ; 2012 : 45)

II.4. SQL SERVER 2008 R2

SQL (*Structured Query Language*) adalah bahasa *non procedural* untuk mengakses data pada *database relasional*. SQL adalah bahasa *database* yang dipergunakan dalam menyelesaikan permasalahan dalam *database* serta mempunyai kelebihan dalam mengolah data. Dengan menggunakan SQL, kita dapat melakukan hal-hal berikut:

1. Memodifikasi struktur *database* .
2. Mengubah, mengisi, menghapus isi *database*.
3. Mentransfer data antara *database* yang berbeda.

SQL ada yang dikembangkan untuk PC dan ada juga yang dikembangkan untuk dapat mengakomodasi *database* yang sangat besar. (Eka Iswandy ; 2015 : 73) .

II.5. Teknik Normalisasi

Normalisasi adalah suatu proses yang digunakan untuk menentukan pengelompokkan atribut-atribut dalam sebuah relasi sehingga diperoleh relasi yang berstruktur baik. Normalisasi sendiri dilakukan melalui sejumlah langkah. Setiap langkah berhubungan dengan bentuk normal (*normal form*) tertentu. Dalam hal ini yang disebut bentuk normal adalah “ suatu keadaan relasi yang dihasilkan oleh penerapan aturan-aturan sederhana yang berhubungan dengan dependensi fungsional terhadap relasi tersebut. Yang dikaksud dengan aturan-aturan tersebut

dan juga istilah dependensi fungsional. Ada 5 bentuk normal dalam normalisasi dapat berupa :

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

Bentuk normal pertama (1NF) adalah suatu keadaan yang membuat setiap perpotongan baris dan kolom dalam relasi hanya berisi satu nilai.

2. Bentuk normal tahap kedua (2nd Normal Form)

Bentuk normal kedua adalah suatu bentuk yang menyatakan bahwa relasi harus sudah berada dalam bentuk normal pertama dan tidak mengandung dependensi parsial.

3. Bentuk normal tahap ketiga (3rd Normal Form)

Bentuk normal ketiga adalah suatu keadaan yang menyatakan bahwa relasi harus sudah berada dalam bentuk normal kedua dan tidak mengandung dependensi transitif. Agar suatu relasi masuk ke dalam bentuk normal ketiga dependensi transif (jika ada) harus dibuang.

4. Boyce Code Normal Form (BCNF)

Merupakan keadaan yang menyatakan bahwa setiap determinan (penentu) dalam suatu relasi berkedudukan sebagai kunci kandidat. Oleh karena itu untuk mengetahui suatu relasi memenuhi BCNF atau tidak perlu dilakukan identifikasi guna memastikan bahwa semua determinan dalam relasi berkedudukan sebagai kunci kandidat.

5. Bentuk normal tahap keempat dan kelima

Suatu keadaan yang membuat relasi yang telah memenuhi bentuk normal keempat yang membuat relasi yang telah memenuhi bentuk normal keempat tidak

dapat didekomposisi menjadi relasi-relasi yang lebih kecil dengan kunci kandidat relasi-relasi pecahannya tersebut tidak sama dengan kunci kandidat relasi. (Abdul Kadir ; 2009 : 130).

II.6. Visual Basic 2010

Visual Basic Net 2010 adalah salah satu bahasa pemrograman yang berorientasi objek. *Microsoft Visual Basic .NET* adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem .NET Framework, dengan menggunakan bahasa *BASIC*. Bahasa *Visual Basic* telah digunakan secara luas karena kemudahan penggunaannya bagi orang awam dan penulisan kode di dalamnya tidak terlalu rumit dibandingkan bahasa C, Delphi, dan Java . (M. Rosidi Zamroni ; 2014 :562).

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

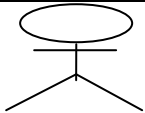
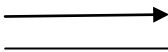
Diagram *Unified Modeling Language* (UML) adalah menjadi diagram yang populer dan telah banyak digunakan oleh para pengembang sistem berorientasi objek. Saat ini, telah banyak perangkat lunak yang membantu membuat model sistem dengan metodologi berbasis objek. (Hamim Tohari ; 2014 : 22) .

Adapun jenis – jenis dari tipe diagram UML adalah sebagai berikut :

1. *Use Case Diagram*

Merupakan rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah actor. *Use case* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi. *Use Case Diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. sebuah *Use Case Diagram* merepresentasikan sebuah interaksi antara actor dengan sistem. *Use Case Diagram* menyatakan sebuah aktivitas atas pekerjaan tertentu. Simbol yang digunakan dalam *Use Case Diagram* ditunjukkan oleh tabel II.4.

Tabel II.4 Simbol *Use Case Diagram*

Simbol	Keterangan
	<i>Actor</i> dapat berupa manusia, sistem atau <i>device</i> yang memiliki peranan dalam keberhasilan operasi dari sistem.
	<i>Use case</i> mengidentifikasi fitur kunci dari sistem. Tanpa <i>use case</i> sistem tidak akan memenuhi permintaan <i>user/actor</i> . Setiap <i>use case</i> mengekspesikan goal-nya dan digambarkan dengan <i>elips</i> .
	<i>Assosiation</i> mengidentifikasi

	interaksi antara setiap <i>actor</i> tertentu dengan setiap <i>use case</i> tertentu.
--	---

(Sumber : Hamim Tohari ; 2014 : 49)

2. Class Diagram

Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi dalam menghasilkan sebuah objek dan merupakan inti dari pengembangan dan perancangan berorientasi objek. Class menggambarkan keadaan (atribut/property) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). Dalam pemodelan statis dari sebuah sistem, class diagram biasanya digunakan untuk memodelkan salah satu dari tiga hal tersebut :

- a. Perbendaharaan dari sistem
- b. Kolaborasi
- c. Skema basis data *logical*

Kelas memiliki tiga area pokok :

- a. Nama dan *stereotype*
- b. Atribut
- c. Metode dan operasi

Atribut dan metode dapat memiliki salah satu sifat berikut :

- a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- b. *Proctected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- c. *Public*, dapat dipanggil oleh siapa saja

Class dapat berupa implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metode. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metode pada saat *run-time*.

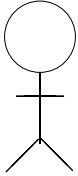
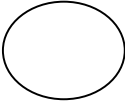
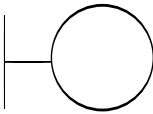
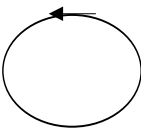
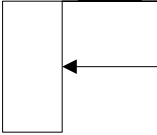
Tabel II.5 Class Diagram

Indikator	Arti
0..1	Nol atau satu
1	Hanya satu
0..*	Nola tau lebih
1..*	Satu atau lebih
N	Hanya n (dengan $n > 1$)
0..n	Nol sampai n (dengan $n > 1$)
1..n	Satu sampai n (dengan $n > 1$)

(Sumber : Hamim Tohari ; 2014 : 86)

3. *Sequene Diagram* menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek yang terjadi pada titik tertentu dalam eksekusi sistem. (Hamim Tohari ; 2014 : 101).

Tabel II.6 Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Actor</i> dapat berupa manusia, sistem atau <i>device</i> yang memiliki peranan dalam keberhasilan operasi dari sistem.
	<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 actor dengan sistem.
	<i>Control Class</i> suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas.
	<i>Message</i> simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

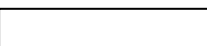
	<i>Activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi .
	<i>Lifeline</i> yaitu garis titik titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

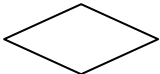
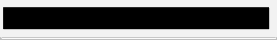
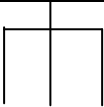
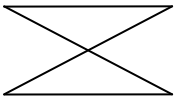
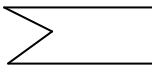
(Sumber : Gellysa Urva dkk ; 2015 : 95)

4. Activity Diagram

Activity Diagram memodelkan *workflow* proses bisnis urutan dan urutan aktivitas dalam sebuah proses. Diagram ini sangat mirip *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status .Membuat *Activity Diagram* pada awal pemodelan proses cukup menguntungkan untuk membantu memahami keseluruhan proses. *Activity Diagram* juga bermanfaat untuk menggambarkan *parallel behavior* atau menggambarkan interaksi antara beberapa *use case*.

Tabel II.7 Activity Diagram

Simbol	Keterangan
	Titik awal (<i>start</i>)
	Titik akhir (<i>akhir</i>)
	Activity

	Pilihan untuk pengambilan keputusan
	<i>Fork</i> : digunakan untuk menunjukkan kegiatan yang dilakukan secara 32 cenario atau untuk menggabungkan dua kegiatan 32 cenario menjadi satu.
	<i>Rake</i> : menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerima
	Aliran akhir (<i>Flow Final</i>)

(Sumber : Hamim Tohari ; 2014)