

## BAB II

### TINJAUAN PUSTAKA

#### II.1. Penelitian Terkait

Penelitian yang dilakukan Teguh Bambang Sunardi dan Danny Kriestanto (2016) dengan judul Perbandingan AHP dan SAW untuk Pemilihan Pegawai Terbaik (Studi Kasus: STMIK AKAKOM Yogyakarta), bertujuan untuk membuat sebuah sistem pendukung keputusan untuk membantu dalam pengambilan keputusan menentukan pegawai terbaik di STMIK AKAKOM Yogyakarta dengan metode AHP dan SAW. Selain itu, dari aplikasi yang dibangun digunakan untuk melakukan analisis metode AHP dan SAW untuk mencari metode manakah yang cocok digunakan dalam pemilihan pegawai terbaik di STMIK AKAKOM Yogyakarta. Melibatkan beberapa kriteria yaitu kesetiaan, prestasi kerja, tanggung jawab, ketaatan, kejujuran, kerja sama, prakarsa, dan kepemimpinan. Penelitian ini dapat membantu penggunanya dalam menemukan pegawai terbaik di STMIK AKAKOM Yogyakarta.

Penelitian yang dilakukan oleh Bayu Riyanto (2016) dengan judul Sistem Pendukung Keputusan Penerimaan Karyawan Menggunakan Metode *Analytical Hierarchy Process* (AHP) Studi Kasus : RB. Nilam Sari Tembilahan, penelitian ini bertujuan mengembangkan sistem pendukung keputusan (SPK) dengan memanfaatkan metode *Analytical Hierarchy Process* (AHP) sebagai proses dalam seleksi penerimaan karyawan. Dalam proses seleksi ini digunakan beberapa

kriteria untuk menentukan calon pelamar yang mana yang akan diterima. SPK ini membantu pimpinan perusahaan dalam memutuskan pelamar mana yang akan dipilih. Penelitian ini lebih menitik beratkan kepada bagaimana merancang dan mengimplementasikan program serta dimaksudkan agar memudahkan dalam hal perhitungan. AHP digunakan sebagai metode dalam perhitungan dalam seleksi penerimaan karyawan ini.

## **II.2. Sistem**

Menurut Mulyadi (2008) sistem adalah Sekelompok dua atau lebih komponen – komponen yang saling berkaitan (sub sistem – sub sistem yang bersatu untuk mencapai tujuan yang sama. (Rini Asmara, 2016)

### **II.2.1. Elemen Sistem**

Ada beberapa elemen-elemen yang saling terkait dalam membentuk sebuah sistem, (Abdul Kadir, 2013) yaitu:

#### **1. Tujuan**

Setiap sistem memiliki tujuan (goal), entah hanya satu atau banyak. Tujuan inilah yang akan menjadi motivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali.

#### **2. Masukan**

Masukan (Input) sistem adalah segala sesuatu yang masuk yang masuk ke dalam sistem yang selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tidak tampak.

### 3. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna, misalnya informasi dan produk tetapi juga bisa hal-hal yang tidak berguna, misalnya sisa pembuangan atau limbah.

### 4. Keluaran

Keluaran (*Output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainya.

### 5. Mekanisme pengendalian, dan Umpan balik

Mekanisme pengendalian (*Controll Mekanism*) diwujudkan dengan menggunakan umpan balik (*feedback*), yang mencerminkan keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. (Vicky Ariandi, dkk, 2018)

## **II.3. Pengambilan Keputusan**

Pengambilan keputusan merupakan proses pemilihan alternatif tindakan untuk mencapai tujuan atau sasaran tertentu. Pengambilan keputusan dilakukan dengan pendekatan sistematis terhadap permasalahan melalui proses pengumpulan data menjadi informasi serta ditambah dengan faktor-faktor yang perlu di pertimbangkan dalam pengambilan keputusan (Rika Yunitarini, 2013).

### **II.3.1. Fase Pengambilan Keputusan**

Dalam pengambilan keputusan sebuah sistem harus mampu melewati beberapa fase-fase proses pengambilan keputusan. Adapun tahapan Pengambilan

Keputusan Menurut Herbert A. Simon (Kadarsah, 2002:15-16), tahap–tahap yang harus dilalui dalam proses pengambilan keputusan sebagai berikut:

1. Tahap Pemahaman (*Intelligence Phace*)

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses dan diuji dalam rangka mengidentifikasi masalah.

2. Tahap Perancangan (*Design Phace*)

Tahap ini merupakan proses pengembangan dan pencarian alternatif tindakan/solusi yang dapat diambil. Tersebut merupakan representasi kejadian nyata yang disederhanakan, sehingga diperlukan proses validasi dan vertifikasi untuk mengetahui keakuratan model dalam meneliti masalah yang ada.

3. Tahap pemilihan (*Choice Phase*)

Tahap ini dilakukan pemilihan terhadap diantara berbagai alternatif solusi yang dimunculkan pada tahap perencanaan agar ditentukan/dengan memperhatikan kriteria–kriteria berdasarkan tujuan yang akan dicapai.

4. Tahap Impelementasi (*Implementation Phace*)

Tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap perancangan serta pelaksanaan alternatif tindakan yang telah dipilih pada tahap pemilihan. (Fazliani, dkk, 2017).

#### **II.4. Sistem Pendukung Keputusan**

Menurut Herbert A.Simon Sistem Pendukung Keputusan merupakan penggabungan sumber-sumber kecerdasan individu dengan kemampuan

komponen untuk memperbaiki kualitas keputusan. Sistem Pendukung Keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah–masalah semi struktur.

Dengan pengertian diatas dapat dijelaskan bahwa sistem pendukung keputusan bukan merupakan alat pengambilan keputusan, melainkan merupakan sistem yang membantu pengambil keputusan dengan melengkapi mereka dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat. Sehingga sistem ini tidak dimaksudkan untuk menggantikan pengambilan keputusan dalam proses pembuatan keputusan. (Rika Yunitarini, 2013).

#### **II.4.1. Komponen Sistem Pendukung Keputusan**

Sistem Pendukung Keputusan terdiri atas 4 komponen utama atau sub-sistem yaitu :

1. *Data Management* (Manajemen Data)

Merupakan komponen SPK sebagai penyedia data bagi sistem, yang mana data disimpan dalam *Database Management System* (DBMS), sehingga dapat diambil dan diekstraksi dengan cepat.

2. *Model Management* (Manajemen Model)

Melibatkan model finansial, statistikal, manajemen *science*, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analistis, dan manajemen *software* yang diperlukan.

### 3. *Communication* (dialog sub sistem)

*User* dapat berkomunikasi dan memberikan perintah pada SPK melalui subsistem ini. Ini berarti menyediakan antarmuka.

### 4. *Knowledge Management* (Manajemen Pengetahuan)

Sub sistem optional ini dapat mendukung sub sistem lain atau bertindak sebagai komponen yang berdiri sendiri. ( Eko Darmanto, dkk, 2014)

## II.4.2. Karakteristik Sistem Pendukung Keputusan

Karakteristik dari sistem pendukung keputusan adalah sebagai berikut :

1. Sistem pendukung keputusan memberikan dukungan bagi pengambil keputusan pada situasi semi terstruktur dan tak terstruktur dengan memadukan pertimbangan manusia dan informasi terkomputerisasi.
2. Dukungan untuk semua level manajerial, dari *eksekutif* puncak sampai manajer lini.
3. Dukungan untuk individu dan kelompok.
4. Dukungan untuk keputusan independen dan sekuensial.
5. Dukungan di semua fase proses pengambilan keputusan, yaitu *intelligence*, *design*, *choice*, dan *implementation*.
6. Dukungan di berbagai proses dan gaya yang berbeda-beda.
7. Adaptivitas sepanjang waktu.
8. Mudah untuk digunakan *user*.
9. Peningkatan efektivitas dari pengambilan keputusan daripada efisiensi.

10. Kontrol penuh oleh pengambil terhadap semua langkah proses pengambilan keputusan.
11. Pengguna akhir bisa mengembangkan dan memodifikasi sendiri sistem sederhana.
12. Biasanya, model-model digunakan untuk menganalisis situasi pengambilan keputusan.
13. Akses disediakan untuk berbagai sumber daya, format, dan tipe, mulai dari sistem informasi sampai sistem berorientasi objek.
14. Dapat digunakan sebagai *standalone* oleh seorang pengambil keputusan pada satu lokasi atau di distribusikan di suatu organisasi secara keseluruhan dan di beberapa organisasi sepanjang rantai persediaan. (Hetty Rohayani, 2013).

## II.5. Metode SAW (*Simple Additive Weighting*)

Metode SAW merupakan metode untuk menentukan nilai terbobot. Pada konsep SAW adalah mencari nilai penjumlahan terbobot dari ranting setiap alternatif untuk semua atribut. Metode SAW didalam terdapat proses normalisasi matrix keputusan ( $x$ ) ke suatu kondisi yang dapat dibandingkan dengan semua alternatif yang ada.

Berikut ini rumus untuk melakukan normalisasi matrix tersebut adalah :

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_{ij}} \\ \frac{\text{Min}_{ij}}{x_{ij}} \end{cases}$$

Keterangan

Jika  $j$  adalah keuntungan atribut ( *benefit* )

Jika  $j$  adalah untuk biaya atribut ( *cost* )

Yang dimana :

$R_{ij}$  = ranting kinerja ternormalisasi

$Max_{ij}$  = nilai maksimum dari setiap baris dan kolom

$Min_{ij}$  = nilai minimum dari setiap baris dan kolom

$X_{ij}$  = baris dan kolom dari matriks

Dengan  $r_{ij}$  adalah ranting kinerja ternormalisasi dari 2 Alternatif  $A_i$  pada atribut

$C_j$ ;  $i = 1, 2, \dots, m$  dan  $j = 1, 2, \dots, n$ .

Nilai referensi untuk setiap alternatif ( $V_i$ ) dirumuskan seperti dibawah ini :

$$V_i = \sum_{j=1}^n W_j r_{ij}$$

Yang dimana:

$V_i$  = nilai akhir alternatif

$W_j$  = nilai bobot yang telah ditentukan

$R_{ij}$  = nilai normalisasi matrixn (Soeb Aripin, dkk, 2018)

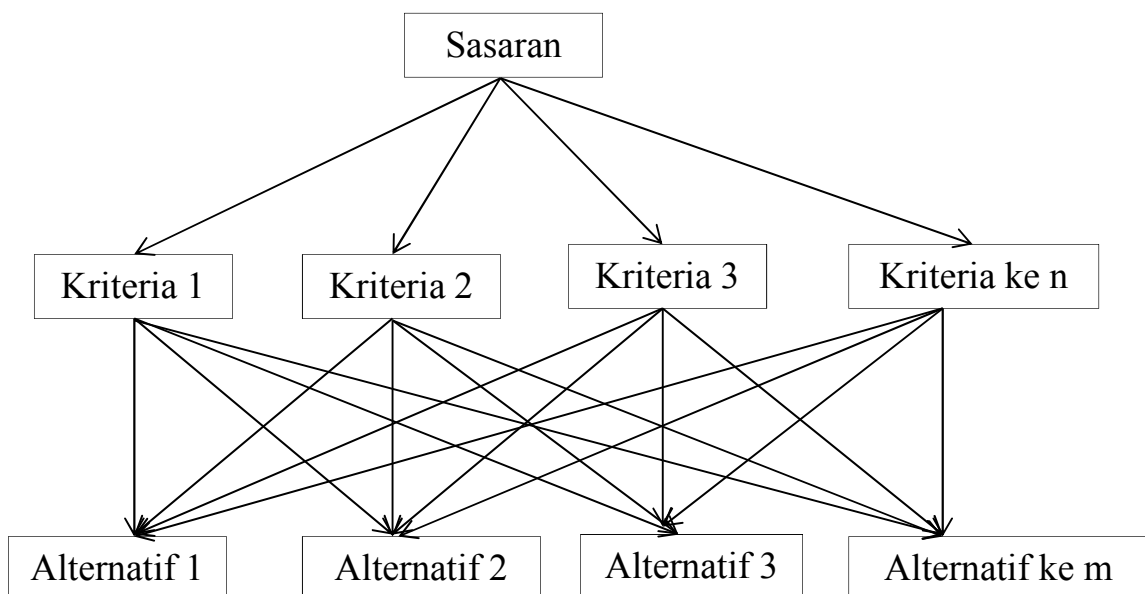
## II.6. Metode AHP (*Analytic Hierarchy Process*)

AHP (*Analytic Hierarchy Process*) adalah suatu teori umum tentang pengukuran yang digunakan untuk menemukan skala rasio, baik dari perbandingan berpasangan yang diskrit maupun kontinyu. AHP menguraikan masalah multi faktor atau multi kriteria yang kompleks menjadi suatu hirarki. Hirarki didefinisikan sebagai suatu representasi dari sebuah permasalahan yang

kompleks dalam suatu struktur multi level dimana level pertama adalah tujuan, yang diikuti level faktor, kriteria, sub kriteria, dan seterusnya ke bawah hingga level terakhir dari alternatif. Dengan hirarki, suatu masalah yang kompleks dapat diuraikan ke dalam kelompok-kelompoknya yang kemudian diatur menjadi suatu bentuk hirarki sehingga permasalahan akan tampak lebih terstruktur dan sistematis. (Eko Darmanto, dkk. 2014).

### II.6.1. Tahapan tahapan dalam AHP (*Analytic Hierarchy Process*)

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan.
2. Membuat struktur hierarki yang diawali dengan tujuan umum, dilanjutkan dengan kriteria-kriteria dan alternatif- alternatif pilihan
3. Gambar 2 dibawah ini adalah gambar struktur hirarki AHP.



**Gambar II.I. Struktur Hierarki AHP**

**Sumber : (Eko Darmanto, dkk 2014)**

4. Membuat matrik perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap tujuan atau kriteria yang setingkat di atasnya. Perbandingan dilakukan berdasarkan pilihan atau judgement dari pembuat keputusan dengan menilai tingkat kepentingan suatu elemen dibandingkan elemen lainnya.
5. Menormalkan data yaitu dengan membagi nilai dari setiap elemen di dalam matrik yang berpasangan dengan nilai total dari setiap kolom.
6. Menghitung nilai eigen vector dan menguji konsistensinya, jika tidak konsisten maka pengambilan data (preferensi) perlu diulangi. Nilai eigen vector yang dimaksud adalah nilai *eigen vector* maksimum yang diperoleh.
7. Mengulangi langkah 3, 4 dan 5 untuk seluruh tingkat hirarki.
8. Menghitung eigen vector dari setiap matriks perbandingan berpasangan. Nilai *eigen vector* merupakan bobot setiap elemen.
9. Menguji konsistensi hirarki. Jika tidak memenuhi dengan  $CR < 0,100$  maka penilaian harus diulangi kembali. (Eko Darmanto, dkk 2014)

## **II.7. Basis Data (*Database*)**

Menurut Asrianda dalam Urva, Gellysa (2008) *Database* adalah sekumpulan tabel-tabel yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap tabel yang ada. Satu database menunjukkan satu lingkup perusahaan atau instansi. (Ade Hendini, 2016).

## II. 8. *Entity Relationship Diagram (ERD)*

Menurut (Yasin, 2012) *Entity Relationship Diagram (ERD)* adalah suatu rancangan atau bentuk hubungan suatu kegiatan di dalam sistem yang berkaitan langsung dan mempunyai fungsi di dalam proses tersebut. ERD adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan *entity* lainnya (Diah Puspitasari, 2016)

## II.9. *Microsoft Visual Basic .Net*

*Visual Basic 2010* merupakan salah satu bagian dari produk pemograman yang dikeluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio 2010*. Sebagai produk pengembangan atau *Integrated Development Environment (IDE)* andalan yang dikeluarkan oleh *Microsoft*, *Visual Studio 2010* berisi beberapa IDE pemograman seperti *Visual Basic*, *Visual C++*, *Visual Web Developer*, *Visual C#*, dan *Visual F#*. Semua IDE tersebut sudah mendukung penuh implementasi *.NET Framework* terbaru, yaitu *.Net Framework 4.0* yang merupakan pengembangan dari *.Net Framework 3.5*. Adapun *database* standar yang disertakan adalah *Microsoft SQL Server 2008 Express*. (Mustakim, Krisnawati, 2013).

## II.10. *SQL Server*

*SQL Server* adalah sebuah database relasional yang dirancang untuk mendukung aplikasi dengan arsitektur *client/server* dimana database terdapat pada komputer pusat yang disebut *server*, dan informasi digunakan bersama-sama oleh beberapa *user* yang menjalankan aplikasi didalam komputer lokalnya yang disebut dengan *client* arsitektur semacam ini memberikan integritas data yang tinggi karena semua *user* bekerja dengan informasi yang sama. Melalui aturan aturan bisnis, kendali diterapkan kepada semua *user* mengenai informasi yang ditambahkan ke dalam *database*. (Rika Yunitarini, 2013).

## II.11. Teknik Normalisasi

Normalisasi merupakan proses pengelompokan elemen data menjadi tabel-tabel yang menunjukkan *entity* dan relasinya. Pada proses normalisasi selalu diuji pada beberapa kondisi, apakah ada kesulitan pada saat menambah / *insert*, menghapus / *delete*, mengubah / *update*, dan membaca / *retrieve* pada suatu Database. (Canggih Ajika Pamungkas dan Sudarmaji, 2015)

### II.11.1. Bentuk-Bentuk Normalisasi

Tahapan dalam normalisasi dimulai dari tahap paling ringan (1NF) hingga paling ketat (5NF). Biasanya hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik. Adapun penjelasannya sebagai berikut :

### 1. Bentuk tahap pertama (1<sup>st</sup> Normal Form)

Sebuah model data dikatakan memenuhi bentuk normal pertama apabila setiap atribut yang dimilikinya memiliki satu dan hanya satu nilai. Apabila ada atribut yang memiliki nilai lebih dari satu, atribut tersebut adalah kandidat untuk menjadi entitas.

**Tabel II.1. Contoh bentuk normal pertama (1NF)**

Kode Transaksi	ID Pelanggan	Nama Pelangan	Tgl Masuk	Tgl Keluar	Jumlah Cucian (Kg)	Pemba- yaran	Ket	ID User	Nama User
M300001	220001	Fanidia	2015-09-01	2015-09-04	5	23000	Lunas	3	Boim
M300002	220001	Fanidia	2015-09-01	2015-09-02	5	23000	Lunas	3	Boim
M300003	110005	Alwan	2015-09-01	2015-09-04	2	18000	Lunas	2	Joni
M300004	420003	Nisa	2015-09-01	2015-09-03	2	18000	Belum Lunas	3	Boim

(Sumber : Refika Khoirunnissa, et al , 2016)

### 2. Bentuk tahap kedua (2<sup>nd</sup> Normal Form)

Sebuah model data dikatakan memenuhi bentuk normal kedua apabila ia memenuhi bentuk normal pertama dan setiap atribut non-identifier sebuah entitas bergantung sepenuhnya hanya pada semua identifier entitas tersebut.

**Tabel II.2. Contoh bentuk normal kedua (2NF)**

Tabel Data User

ID User	Nama User
1	Refika
2	Joni
3	Boim

Tabel Data Pelanggan

ID Pelanggan	Nama Pelanggan
220001	Fanidia
110005	Alwan
420003	Nisa

Tabel Data Transaksi

Kode Transaksi	ID Pelanggan	Tgl Masuk	Tgl Keluar	Jumlah Cucian (Kg)	Pembayaran	Ket	ID User
M300001	220001	2015-09-01	2015-09-04	5	23000	Lunas	3
M300002	220001	2015-09-01	2015-09-02	5	23000	Lunas	3
M300003	110005	2015-09-01	2015-09-04	2	18000	Lunas	2
M300004	420003	2015-09-01	2015-09-03	2	18000	Belum Lunas	3

(Sumber : Refika Khoirunnissa, dkk , 2016)

### 3. Bentuk tahap ketiga (3<sup>rd</sup> Normal Form)

Sebuah model data dikatakan memenuhi bentuk normal ketiga apabila ia memenuhi bentuk normal kedua dan tidak ada satupun atribut *non-identifying* (bukan pengidentifikasi unik) yang bergantung pada atribut *non-identifying* lain. Apabila ada, pisahkan salah satu atribut tersebut menjadi entitas baru, dan atribut yang bergantung padanya menjadi atribut entitas baru tersebut.

**Tabel II.3. Contoh bentuk normal ketiga (3NF)**

Tabel Detail Transaksi

Kode Transaksi	Jumlah cuciian (kg)	Harga Cucian
M300001	5	23000
M300002	5	23000
M300003	2	18000
M300004	2	18000

Tabel Data Transaksi

Kode Transaksi	ID Pelanggan	Tgl Masuk	Pembayaran	Keterangan	ID User
M300001	220001	2015-09-01	23000	Lunas	3
M300002	220001	2015-09-01	23000	Lunas	3
M300003	110005	2015-09-01	18000	Lunas	2

Tabel Data Transaksi ( lanjutan)

M300004	420003	2015-09-01	18000	Belum Lunas	3
---------	--------	------------	-------	-------------	---

(Sumber : Refika Khoirunnissa, dkk , 2016)

#### 4. Bentuk Normal *Boyce-Codd* (BCNF)

Bentuk BCNF terpenuhi dalam sebuah tabel, jika untuk setiap functional *dependency* terhadap setiap atribut atau gabungan atribut dalam bentuk:  $X \rightarrow Y$  maka X merupakan *super key*.

#### 5. Bentuk Normal Keempat (4<sup>th</sup> Normal Form)

Bentuk normal 4NF terpenuhi dalam sebuah tabel jika telah memenuhi bentuk BCNF, dan tabel tersebut tidak boleh memiliki lebih dari sebuah *multivalued attribute*.

#### 6. Bentuk Normal Kelima (5<sup>th</sup> Normal Form)

Bentuk normal 5NF terpenuhi jika tidak dapat memiliki sebuah *lossless decomposition* menjadi tabel-tabel yang lebih kecil. (Canggih Ajika Pamungkas dan Sudarmaji, 2015)

### II.12. *Unified Modeling Language* (UML)

Menurut Windu Gata dalam Ade Hendini (2016), *Unified Modeling Language* (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.

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* 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:

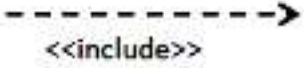
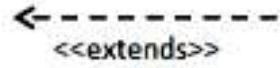
**Tabel II.4 Simbol *Use Case Diagram***

Gambar	Keterangan
	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktif, yang dinyatakan dengan menggunakan kata kerja.</p>

Tabel II.4 Simbol *Use Case Diagram* (Lanjutan)

	<p><i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem.</p> <p>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.</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 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>

**Tabel II.4 Simbol *Use Case Diagram* (Lanjutan)**




	<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 : Ade Hendini, 2016)

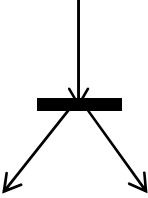
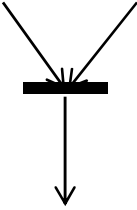
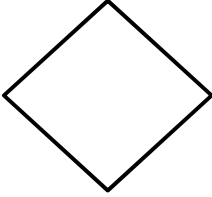
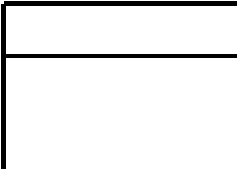
2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity Diagram* yaitu”

**Tabel II.5. Simbol *Activity Diagram***

Gambar	Keterangan
	<p><i>Start Point</i>, diletakkan pada pojok kiri atas dan merupakan awal aktivitas.</p>
	<p><i>End Point</i>, akhir aktivitas.</p>
	<p><i>Activities</i>, menggambarkan suatu proses/kegiatan bisnis.</p>

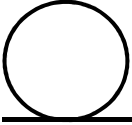
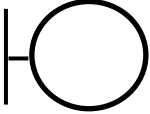
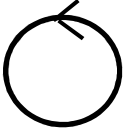
Tabel II.5. Simbol *Activity Diagram* (lanjutan)

	<p><i>Fork</i>/percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.</p>
	<p><i>Join</i> (penggabungan) atau <i>rake</i>, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>.</p>
	<p><i>Swimlane</i>, pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.</p>


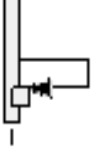


### 3. Diagram Urutan (*Sequence Diagram*)

*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.6. Simbol *Sequence Diagram*

Gambar	Keterangan
	<p><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.</p>
	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak.</p>
	<p><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.</p>

Tabel II.6. Simbol *Sequence Diagram* (lanjutan)

	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i>, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i></p>

(Sumber : Ade Hendini, 2016)

#### 4. Diagram Kelas (*Class Diagram*)

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 (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

**Tabel II.7. Multiplicity Class Diagram**

<i>Multiplicity</i>	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 maksimal 4

(Sumber : Ade Hendini, 2016)