

BAB II

LANDASAN TEORI

2.1. Sistem Pendukung Keputusan

SPK sebagai sebuah sistem berbasis komputer yang membantu dalam proses pengambilan keputusan. SPK sebagai sistem informasi berbasis komputer yang adaptif, interaktif, fleksibel, yang secara khusus dikembangkan untuk mendukung solusi dari permasalahan manajemen yang tidak terstruktur untuk meningkatkan kualitas pengambilan keputusan. Dengan demikian dapat ditarik satu definisi tentang SPK yaitu sebuah sistem berbasis komputer yang adaptif, fleksibel, dan interaktif yang digunakan untuk memecahkan masalah-masalah tidak terstruktur sehingga meningkatkan nilai keputusan yang diambil (Khoirudin, 2010).

Melwin Syafrizal (2011), mendeskripsikan SPK dapat digambarkan sebagai sistem yang berkemampuan mendukung analisis ad hoc data dan pemodelan keputusan, berorientasi keputusan, orientasi perencanaan masa depan, dan digunakan pada saat-saat yang tidak biasa. Sistem pendukung keputusan dibuat sebagai suatu cara untuk memenuhi kebutuhan seorang manajer dalam membuat keputusan yang spesifik dalam memecahkan permasalahan yang spesifik pula. Empat tahapan dalam pembuatan keputusan adalah: *Intelligence* (menemukan apa yang harus diperbaiki), *Design* (menemukan pilihan perbaikan), *Choice* (mengambil keputusan pilihan perbaikan), dan *Implementation* (mengimplementasikan perbaikan).

2.1.1. Konsep Sistem Pendukung Keputusan

Menurut Mat dan Waston, Sistem Pendukung Keputusan (SPK) merupakan suatu sistem interaktif yang membantu pengambilan keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah yang sifatnya semi terstruktur dan tidak terstruktur. Sedangkan menurut Moorre dan Chang, SPK adalah sistem yang dapat dikembangkan, mampu mendukung analisis data dan pemodelan keputusan, berorientasi pada perencanaan masa mendatang. Serta tidak bisa direncanakan interval (periode) waktu pemakainya.

2.1.2. Karakteristik Sistem Pendukung Keputusan

Karakteristik dari Sistem Pendukung keputusan yang membedakan dari sistem informasi lainnya adalah :

- a. SPK dirancang untuk membantu pengambilan keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun tidak terstruktur.
- b. Dalam proses pengolahannya, SPK mengkombinasikan penggunaan model-model analisa teknik pemasukan data konvensional serta fungsi pencari interogasi informasi.
- c. SPK dirancang sedemikian rupa sehingga dengan mudah digunakan oleh orang yang tidak memiliki dasar kemampuan pengoperasian computer yang tinggi. Oleh karena itu pendekatan yang digunakan biasanya model interaktif.

2.1.3. Keuntungan Sistem Pendukung Keputusan

Sistem pendukung keputusan dapat memberikan berbagai manfaat atau keuntungan bagi pemakainya, antara lain:

- a. Memperluas kemampuan pengambilan keputusan dalam memproses data bagi pemakainya.
- b. Membantu pengambilan keputusan dalam hal penghematan waktu yang dibutuhkan untuk memecahkan masalah terutama masalah yang kompleks dan tidak terstruktur.
- c. Dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan
- d. Walaupun suatu sistem pendukung keputusan, mungkin saja tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun dapat menjadi stimulan bagi pengambil keputusan dalam memahami persoalannya, karena sistem pendukung keputusan mampu menyajikan berbagai alternative.

2.2. Prestasi

Gagne menyatakan bahwa prestasi belajar dibedakan menjadi lima aspek, yaitu : kemampuan intelektual, strategi kognitif, informasi verbal, sikap dan keterampilan. Menurut Bloom dalam Suharsimi Arikunto (1990:110) bahwa hasil belajar dibedakan menjadi tiga aspek yaitu kognitif, afektif dan psikomotorik. Menurut W.J.S Winkel Purwadarmtinto(2010), siswa berprestasi adalah siswa yang mampu menaati peraturan yang ada yang menjadi kebijakan yang ada,serta dapat menuntaskan kriteria minimal

dari KKM, belajar dilakukan dengan sungguh-sungguh sehingga mencapai hasil yang terbaik dan maksimal.

2.3. *FMADM (Fuzzy Multiple Atrtribute Decision Making)*

2.3.1. **Himpunan Fuzzy**

Fuzzy Multiple Attribute Decision Making FMADM adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Inti dari FMADM adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan. Pada dasarnya, ada 3 pendekatan untuk mencari nilai bobot atribut, yaitu pendekatan subyektif, pendekatan obyektif dan pendekatan integrasi antara subyektif & obyektif. Masing masing pendekatan memiliki kelebihan dan kelemahan. Pada pendekatan subyektif, nilai bobot ditentukan berdasarkan subyektifitas dari para pengambil keputusan, sehingga beberapa faktor dalam proses perankingan alternatif bisa ditentukan secara bebas. Sedangkan pada pendekatan obyektif, nilai bobot dihitung secara matematis sehingga mengabaikan subyektifitas dari pengambil keputusan. (Kusumadewi, 2011).

Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah FMADM. antara lain (Kusumadewi, 2012):

- a. Simple Additive Weighting Method (SAW)
- b. Weighted Product (WP)
- c. ELECTRE

- d. Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)
- e. Analytic Hierarchy Process (AHP)

2.3.2. *Algoritma FMADM*

Algoritma FMADM adalah:

1. Memberikan nilai setiap alternatif (A_i) pada setiap kriteria (C_j) yang sudah ditentukan, dimana nilai tersebut di peroleh berdasarkan nilai crisp; $i=1,2,\dots,m$ dan $j=1,2,\dots,n$
2. Memberikan nilai bobot (W) yang juga didapatkan berdasarkan nilai crisp.
3. Melakukan normalisasi matriks dengan cara menghitung nilai rating kinerja ternormalisasi (r_{ij}) dari alternatif A_i pada atribut C_j berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan atau benefit = MAKSIMUM atau atribut biaya atau cost= MINIMUM). Apabila berupa artibut keuntungan maka nilai crisp (X_{ij}) dari setiap kolom atribut dibagi dengan nilai crisp MAX ($\text{MAX } X_{ij}$) dari tiap kolom, sedangkan untuk atribut biaya, nilai crisp MIN ($\text{MIN } X_{ij}$) dari tiap kolom atribut dibagi dengan nilai crisp (X_{ij}) setiap kolom.
4. Melakukan proses perankingan dengan cara mengalikan matriks ternormalisasi (R) dengan nilai bobot (W).
5. Menentukan nilai preferensi untuk setiap alternatif (V_i) dengan cara menjumlahkan hasil kali antara matriks ternormalisasi (R) dengan

nilai bobot (W). Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih. (Kusumadewi , 2010).

2.3.3. *Simple Additive Weighting (SAW)*

Metode *Simple Additive Weighing* sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap Alternative pada semua atribut. Metode *Simple Additive Weighting* membutuhkan proses normalisasi matriks keputusan (x) ke suatu skala yang dapat diperbandingkan dengan semua rating Alternative yang ada.

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i x_{ij}} & \text{jika j adalah atribut keuntungan (benefit)} \\ \frac{\text{Min}_i x_{ij}}{x_{ij}} & \text{jika j adalah atribut biaya (cost)} \end{cases}$$

Keterangan :

r_{ij}	= nilai rating kinerja ternormalisasi
x_{ij}	= nilai atribut yang dimiliki dari setiap kriteria
$\text{Max}_i x_{ij}$	= nilai terbesar dari setiap kriteria i
$\text{Min}_i x_{ij}$	= nilai terkecil dari setiap kriteria i
<i>Benefit</i>	= jika nilai terbesar dalah terbaik
<i>Cost</i>	= jika nilai terkecil dalah terbaik

Dimana (r_{ij}) adalah rating kinerja ternormalisasi dari Alternative A, pada attribut $C_i : i=1,2,\dots,m$ dan $j= 1,2,\dots,n$. Nilai preferensi untuk setiap Alternative V_i diberikan sebagai :

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

Keterangan :

V_i = ranking untuk setiap Alternative

W_j = nilai terbobot dari setiap kriteria

r_{ij} = nilai rating kinerja ternormalisasi

Nilai V_i yang lebih besar mengindikasikan bahwa Alternative A, lebih terpilih.

2.3.4. Langkah Penyelesaian

Dalam penelitian ini menggunakan FMADM metode SAW. Adapun langkah-langkahnya adalah:

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C_i .
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria (C_i), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R .
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A_i) sebagai solusi. (Kusumadewi, 2010).

2.4. Implementasi SAW

Suatu rumah sakit membutuhkan seorang IT support untuk menjadi karyawan tetap. Ada lima kriteria yang digunakan untuk melakukan pemilihan, yaitu:

1. C1 = skill programing
2. C2 = skill jaringan
3. C3 = tes kepribadian
4. C4 = tes keagamaan
5. C5 = tes psikotes

Ada lima orang karyawan yang menjadi kandidat (Alternative) untuk dipromosikan sebagai kepala cabang, yaitu:

- A1 = Andre,
 A2 = Aan,
 A3 = Habib,
 A4 = Dede,
 A5 = Yunus.

Perhitungan dengan *Simple Additive Weighting* :

1. memberikan bobot untuk setiap kriteria sebagai berikut:

C1= 30%; C2= 25%; C3= 20%; C4= 15% dan C5= 10%.

Tabel 2.1: nilai pada setiap kriteria

Kriteria	C1	C2	C3`	C4	C5
Nilai					
Bobot	0.30	0.25	0.20	0.15	0.10

2. Pemberian nilai pada attribute

Tabel 2.2 : nilai pada setiap attribute

Alternative	Kriteria				
	C1	C2	C3	C4	C5
Andre	70	50	60	70	40
Aan	70	70	50	60	60
Habib	75	60	40	70	80
Dede	60	40	70	60	70
Yunus	50	60	80	80	50

3. Membuat Tabel Normalisasi

$$r_{ij} = \frac{X_{ij}}{\max \{X_{ij}\}}$$

a. Alternative 1

$$r_{11} = \frac{70}{\max\{70;70;75;60;50\}} = \frac{70}{75} = 0.93$$

$$r_{12} = \frac{50}{\max\{50;70;60;40;60\}} = \frac{50}{70} = 0.71$$

$$r_{13} = \frac{60}{\max\{60;50;40;70;80\}} = \frac{60}{80} = 0.75$$

$$r_{14} = \frac{70}{\max\{70;60;70;60;80\}} = \frac{70}{80} = 0.88$$

$$r_{15} = \frac{40}{\max\{40;60;80;70;50\}} = \frac{40}{80} = 0.5$$

b. Alternative 2

$$r_{11} = \frac{70}{\max\{70;70;75;60;50\}} = \frac{70}{75} = 0.93$$

$$r_{12} = \frac{70}{\max\{50;70;60;40;60\}} = \frac{70}{70} = 1$$

$$r_{13} = \frac{50}{\max\{60;50;40;70;80\}} = \frac{50}{80} = 0.63$$

$$r_{14} = \frac{60}{\max\{70;60;70;60;80\}} = \frac{60}{80} = 0.75$$

$$r_{15} = \frac{60}{\max\{40;60;80;70;50\}} = \frac{460}{80} = 0.75$$

c. Alternative 3

$$r_{11} = \frac{75}{\max\{70;70;75;60;50\}} = \frac{75}{75} = 1$$

$$r_{12} = \frac{60}{\max\{50;70;60;40;60\}} = \frac{60}{70} = 0.86$$

$$r_{13} = \frac{40}{\max\{60;50;40;70;80\}} = \frac{40}{80} = 0.5$$

$$r_{14} = \frac{70}{\max\{70;60;70;60;80\}} = \frac{70}{80} = 0.88$$

$$r_{15} = \frac{80}{\max\{40;60;80;70;50\}} = \frac{80}{80} = 1$$

d. Alternative 4

$$r_{11} = \frac{60}{\max\{70;70;75;60;50\}} = \frac{60}{75} = 0.8$$

$$r_{12} = \frac{40}{\max\{50;70;60;40;60\}} = \frac{40}{70} = 0.57$$

$$r_{13} = \frac{70}{\max\{60;50;40;70;80\}} = \frac{70}{80} = 0.88$$

$$r_{14} = \frac{60}{\max\{70;60;70;60;80\}} = \frac{60}{80} = 0.75$$

$$r_{15} = \frac{70}{\max\{40;60;80;70;50\}} = \frac{70}{80} = 0.88$$

e. Alternative 5

$$r_{11} = \frac{50}{\max\{70;70;75;60;50\}} = \frac{50}{75} = 0.67$$

$$r_{12} = \frac{60}{\max\{50;70;60;40;60\}} = \frac{60}{70} = 0.86$$

$$r_{13} = \frac{80}{\max\{60;50;40;70;80\}} = \frac{80}{80} = 1$$

$$r_{14} = \frac{80}{\max\{70;60;70;60;80\}} = \frac{80}{80} = 1$$

$$r_{15} = \frac{50}{\max\{40;60;80;70;50\}} = \frac{50}{80} = 0.63$$

Tabel 2.3: Normalisasi Alternative

Alternative	C1	C2	C3	C4	C5
A1	0.93	0.71	0.75	0.88	0.5
A2	0.93	1	0.63	0.75	0.75
A3	1	0.86	0.5	0.88	1
A4	0.8	0.57	0.88	0.75	0.88
A5	0.67	0.86	1	1	0.63

4. Perangkingan

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

$$\begin{aligned} \text{a. } V_1 &= (0.30 \times 0.93) + (0.25 \times 0.71) + (0.20 \times 0.75) + \\ &\quad (0.15 \times 0.88) + (0.10 \times 0.5) = 0.7885 \end{aligned}$$

$$\begin{aligned} \text{b. } V_2 &= (0.30 \times 0.93) + (0.25 \times 0.1) + (0.20 \times 0.63) + \\ &\quad (0.15 \times 0.75) + (0.10 \times 0.75) = 0.8425 \end{aligned}$$

$$\text{c. } V_3 = (0.30 \times 1) + (0.25 \times 0.86) + (0.20 \times 0.5) +$$

$$(0.15 \times 0.88) + (0.10 \times 1) = 0.8470$$

$$\begin{aligned} \text{d. } V4 &= (0.30 \times 0.8) + (0.25 \times 0.5) + (0.20 \times 0.88) + \\ & (0.15 \times 0.75) + (0.10 \times 0.88) = 0.7590 \end{aligned}$$

$$\begin{aligned} \text{e. } V5 &= (0.30 \times 0.67) + (0.25 \times 0.86) + (0.20 \times 1) + \\ & (0.15 \times 1) + (0.10 \times 0.63) = 0.8290 \end{aligned}$$

Tabel 2.4: Perangkingan Alternative

Vektor	Nilai	Rangking	Nama Alternative
V1	0.7885	4	Andre
V2	0.8425	2	Aan
V3	0.8470	1	Habib
V4	0.7590	5	Dede
V5	0.8290	3	Yunus

Dari hasil perhitungan metode *simple additive weighting* tersebut dapat disimpulkan bahwa hasil terbaik didapatkan kandidat yaitu Habib dengan perolehan nilai 0.8470 (Karno, 2013)

2.5. MySQL

Menurut Yeni Kustiyaningsih (2010), Basis data adalah sekumpulan informasi yang diatur agar mudah dicari. Dalam arti umum basis data adalah sekumpulan data yang diproses dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan tepat, yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berelasi.

MySQL merupakan suatu database. MySQL dapat juga dikatakan sebagai database yang sangat cocok bila dipadukan dengan PHP. Secara umum, database berfungsi sebagai tempat atau wadah untuk menyimpan. Mengklasifikasikan data secara profesional. MySQL bekerja menggunakan

SQL language (*Structure Query Language*) dapat diartikan bahwa MySQL merupakan standar penggunaan database di dunia untuk pengolahan data.

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Sedangkan RDBMS sendiri akan lebih banyak mengenal istilah seperti tabel, baris, dan kolom digunakan dalam perintah-perintah di MySQL, merupakan sebuah basis data yang mengandung satu atau sejumlah tabel yang terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel yang terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. Didalam PHP telah menyediakan fungsi untuk koneksi ke basis data dengan sejumlah fungsi untuk pengaturan baik menghubungkan maupun memutuskan koneksi dengan server database MySQL sebagai sarana untuk mengumpulkan informasi.

Pada umumnya, perintah yang paling sering digunakan dalam MySQL adalah *Select* (mengambil), *insert* (menambah), *update* (mengubah), dan *delete* (menghapus). Selain itu, SQL juga menyediakan perintah untuk membuat database, field ataupun index guna menambah atau menghapus data.

2.6. PHP

Menurut Agus Saputra (2011), PHP atau memiliki kepanjangan *Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu web dinamis. PHP menyat dengan kode HTML maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun ata pondasi dari kerangka *layout web*, sedangkan PHP

difungsikan sebagai prosesnya sehingga dengan adanya PHP tersebut web sangat mudah di *maintenance*.

PHP berjalan pada sisi server sehingga PHP disebut juga sebagai bahasa *Server Side Scripting* artinya bahwa dalam setiap untuk menjalankan PHP, wajib adanya web server. PHP ini bersifat *open source* sehingga dapat dipakai secara cuma-cuma dan mampu lintas platform, yaitu dapat berjalan pada sistem operasi *Windows* maupun *Linux*. PHP juga dibangun sebagai modul pada *web server apache* dan sebagai binary yang dapat berjalan sebagai CGI.

2.7. HTML

HTML kependekan dari *HyperText Markup Language*. Dokumen HTML adalah file teks murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini dikenal sebagai web page. Dokumen ini umumnya berisi informasi atau interface aplikasi di dalam internet. Ada dua cara untuk membuat sebuah web page yaitu dengan HTML editor atau dengan editor teks biasa (misalnya *notepad*). (Herlawati, 2010).

Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). versi terakhir dari HTML adalah HTML 5.01, meskipun saat ini telah berkembang XHTML yang merupakan perkembangan dari HTML.

HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan *web browser* seperti *Mozilla Firefox* atau *Microsoft Internet Explorer*. HTML juga dapat dikenali oleh aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan browser.

2.8. CSS

CSS merupakan singkatan dari *Cascading Style Sheet*. Kegunaannya adalah untuk mengatur tampilan dokumen, contohnya seperti pengaturan jarak antar baris, teks, warna dan format border bahkan penampilan file gambar. CSS dikembangkan oleh W3C, organisasi yang mengembangkan teknologi internet. Tujuannya yaitu untuk mempermudah proses penataan web. Perlu diingat, CSS hanyalah berupa kumpulan pelengkap agar dokumen HTML bias tampil lebih menarik dan dinamis.

Sejak ditemukannya CSS pada awal dekade 90an, CSS terus dikembangkan dan diserap oleh *web developer*, hingga sekarang telah mencapai versi ke-3. Kode CSS bersifat lintas platform, yang berarti script ini dapat dibaca oleh berbagai macam sistem operasi dan browser. Hanya saja browser seperti *Internet Explorer*, seringkali salah mengartikan script CSS yang menyebabkan ketidaksempurnaanya tampilan dokumen HTML. Script CSS perlu dioptimalkan agar tampil maksimal pada *browser internet explorer*. (Jayan, 2010).

2.9. Rekayasa Perangkat Lunak

Menurut Shalahuddin (2011), perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumen kebutuhan, model desain, dan cara pengguna (user manual). Sebuah program komputer tanpa terasosiasi dengan dokumentasinya maka belum dapat disebut perangkat lunak.

Perangkat lunak adalah suatu instruksi (program komputer) yang dibuat oleh sebuah perusahaan komputer untuk mengerjakan tugas tertentu. Memberikan fungsi dan unjuk kerja seperti yang diinginkan atau mengatur struktur data memungkinkan program memanipulasi informasi secara proposional, dan mengatur dokumen yang menggambarkan operasi dan kegunaan program.

2.10. UML

Unified Modeling Language (UML) merupakan alat merancang perangkat lunak, sarana komunikasi antara perangkat lunak dengan, menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem, mendokumentasikan sistem yang ada, proses-proses dan organisasinya. (Herlawati, 2011).

Unified Modeling Language (UML) adalah salah satu bentuk language atau bahasa, menurut pencetusnya UML didefinisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model dan mendokumentasikan aspek-aspek dari sebuah sistem.

Salah satu cara untuk mengatur diagram UML adalah dengan menggunakan *view*. View adalah kumpulan diagram yang menggambarkan

aspek yang sama dari proyek. View mempunyai 3 pelengkap yaitu *Static View*, *Dynamic View* dan *Functional View*.

2.10.1. Static View

Static View termasuk diagram yang memberikan gambaran dari unsur-unsur sistem tetapi tidak memberitau bagaimana elemen akan berperilaku. Hal ini sangat mirip *blueprint* yang bersifat *komperhensif*, tetapi mereka hanya menunjukkan apa yang tetap diam, maka disebut *Static View* yang berbentuk dua diagram yaitu *Class Diagram* dan *Object Diagram*.

2.10.2. Dynamic View

Pada *Dynamic View* meliputi diagram yang mengungkapkan bagaimana benda berinteraksi dengan satu sama lain dalam respon terhadap lingkungan. Ini termasuk *Sequence Diagram* dan *Collaboration Diagram*, yang kolektif disebut sebagai diagram interaksi. Mereka secara khusus dirancang untuk menjelaskan bagaimana benda berbicara satu sama lain. Ini juga mencakup *Statechart Diagram*, yang menunjukkan bagaimana dan mengapa perubahan objek dari waktu ke waktu dalam menanggapi lingkungan.

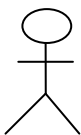


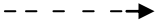
2.10.3. Functional View

Functional View terbentuk oleh *Use Case Diagram* dan *Activity Diagram*.

1. Use Case Diagram

Use Case diagram merupakan permodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah instruksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. *Use case*. Digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut actor dan *use case*. Simbol-simbol yang ada pada diagram *use case*:

Tabel 2.5: Simbol *Use Case Diagram* (Shalahuddin dan Rosa, 2011)







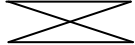


Simbol	Keterangan
	<i>Actor</i> : Sebuah peran yang dimainkan oleh seorang, sistem, atau perangkat yang memiliki saham dalam keberhasilan operasi dari sistem.
	<i>Use Case</i> : Untuk mengungkapkan tujuan bahwa sistem harus dicapai.
	<i>Association</i> : Mengidentifikasi interaksi antara actor dan <i>Use Case</i>
	<i>Dependency</i> : Mengidentifikasi hubungan komunikasi antara dua <i>Use Case</i>

2. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan workflow atau aktivitas dari sebuah sistem atau proses bisnis, yang perlu diperhatikan disini bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang digunakan aktor, jadi aktivitas yang dapat dilakukan sistem.

Berikut simbol-simbol yang ada pada *activity diagram* :

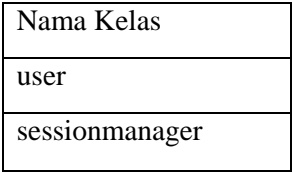


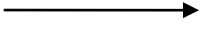
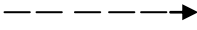
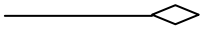
Tabel 2.6: Simbol *Activity Diagram* (Shalahuddin dan Rosa, 2011)

Simbol	Keterangan
	Titik Awal
	Titik Akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork : Untuk menunjukkan kekuatan yang dilakukan secara parallel
	Rake : Menunjukkan adanya dekomposisi
	Tanda Waktu
	Tanda Penerimaan
	Aliran Akhir (<i>Flow Final</i>)

3. Class Diagram

Kelas Diagram terdiri dari ruang persegi panjang yang mengandung informasi yang berbeda perlu untuk menjelaskan sifat-sifat satu jenis objek. Simbol yang digunakan dalam class diagram. :

Tabel 2.7 Simbol *Class Diagram* (Rosa, 2011)

Simbol	Deskripsi
Kelas 	Kelas pada struktur system
Paket atau <i>Package</i> 	Paket atau <i>package</i> merupakan sebuah bungkus dari satu atau lebih kelas (kumpulan kelas)
<i>Asosiasi</i> 	<i>Asosiasi</i> merupakan hubungan antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
<i>Generalisasi</i> 	<i>Generalisasi</i> merupakan hubungan generalisasi dan spesialisasi (umum khusus) antara dua kelas dimana fungsi yang lebih umum dari fungsi yang lainnya.
<i>Dependency</i> 	<i>Dependency</i> merupakan hubungan antar kelas yang saling bergantung membutuhkan satu sama lain
<i>Agregasi</i> 	<i>Agregasi</i> merupakan hubungan antar kelas dimana satu kelas merupakan semua bagian dari kelas-kelas yang lain



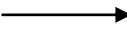
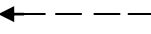


4. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-

objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram *sequence* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* telah didefinisikan interaksinya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *use case* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak. Simbol *Sequence Diagram*:

Tabel 2.8 Simbol *Sequence Diagram* (Shalahuddin dan Rosa, 2011).

Nama	Simbol	Keterangan
Objek		Objects mewakili peserta
<i>Lifeline</i>		Lifeline merupakan siklus hidup sebuah pesan berdasarkan waktu
<i>Message</i>		Pesan merupakan komunikasi antara objek yang satu dengan objek yang lain
<i>Return message</i>		Return message merupakan balasan hasil yang berisi nilai dari objek yang meminta (mengirim pesan)
<i>Self message</i>		Self message merupakan pesan dari sebuah objek kepada objek lain itu sendiri untuk melakukan suatu aksi
<i>Return self message</i>		Balasan dari self message yang berisi suatu nilai kepada objek itu sendiri