

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Sistem penunjang keputusan (SPK)**

Menurut Asfi dan Sari (2010:132), Sistem Penunjang Keputusan (SPK) adalah bagian dari sistem informasi berbasis komputer termasuk sistem berbasis pengetahuan untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan. SPK juga dapat merupakan sistem komputer yang mengolah data menjadi informasi untuk mengambil keputusan dari masalah semi-terstruktur yang spesifik. SPK dapat menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian mereka. SPK ditujukan untuk keputusan-keputusan yang memerlukan penilaian atau pada keputusan-keputusan yang sama sekali tidak dapat didukung oleh algoritma.

Menurut Jahanshahlo dalam Masruro (2013:07-39), sistem penunjang keputusan adalah sebuah sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semi terstruktur.

Menurut Morton dalam Sarwindah (2013:09-39), sistem berbasis komputer interaktif, yang membantu para pengambil keputusan untuk menggunakan data dan berbagai model untuk memecahkan masalah-masalah yang tidak terstruktur”.

Dari ketiga definisi di atas, maka dapat disimpulkan sistem penunjang keputusan adalah sistem yang membantu para pengambil keputusan manajerial

dalam situasi semi terstruktur dan tidak terstruktur dengan berbasis komputer interaktif.

### **II.1.1. Karakteristik Sistem penunjang keputusan**

Beberapa karakteristik dari Sistem Pendukung Keputusan menurut (Turban : 2010) adalah sebagai berikut :

1. Sistem penunjang keputusan dirancang untuk membantu pengambil keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun tidak terstruktur.
2. Dalam proses pengolahannya, sistem penunjang keputusan mengkombinasikan penggunaan model-model / teknik-teknik analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari / interogasi informasi.
3. Sistem Pendukung Keputusan, dirancang sedemikian rupa sehingga dapat digunakan / dioperasikan dengan mudah oleh orang-orang yang tidak memiliki dasar kemampuan yang tinggi. Oleh karena itu pendekatan yang digunakan biasanya model interaktif.
4. Sistem penunjang keputusan dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi. Sehingga mudah disesuaikan dengan berbagai perubahan lingkungan yang terjadi dan kebutuhan pemakai.

Dengan berbagai karakter khusus seperti yang dikemukakan di atas, sistem penunjang keputusan dapat memberikan berbagai manfaat atau keuntungan bagi pemakainya. Keuntungan yang dimaksud di antaranya meliputi :

1. Sistem penunjang keputusan memperluas kemampuan pengambil keputusan dalam memproses data / informasi bagi pemakainya.
2. Sistem penunjang keputusan membantu pengambil keputusan dalam hal penghematan waktu yang dibutuhkan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.
3. Sistem penunjang keputusan dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.
4. Walaupun Sistem penunjang keputusan telah ada, mungkin saja tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun dapat disajikan kesimpulan bagi pengambil keputusan dalam memahami persoalannya. Karena sistem ini mampu menyajikan berbagai alternatif.
5. Sistem penunjang keputusan dapat menyediakan bukti tambahan untuk memberikan pembenaran sehingga dapat memperkuat posisi pengambil keputusan.

Di samping berbagai keuntungan dan manfaat yang dikemukakan di atas, Sistem penunjang keputusan juga memiliki keterbatasan diantaranya adalah sebagai berikut :

1. Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat dimodelkan, sehingga model yang ada dalam sistem tidak semuanya mencerminkan persoalan sebenarnya.
2. Kemampuan suatu SPK terbatas pada pembendaharaan pengetahuan yang dimilikinya (pengetahuan dasar serta model dasar).

3. Proses-proses yang dapat dilakukan oleh SPK biasanya tergantung juga pada kemampuan perangkat lunak yang digunakannya.
4. SPK tidak memiliki kemampuan intuisi seperti yang dimiliki oleh manusia.

Karena walau bagaimanapun canggihnya suatu SPK, tetap saja berupa kumpulan dari perangkat keras, perangkat lunak dan sistem operasi yang tidak dilengkapi dengan kemampuan berpikir.

### **II.1.2. Komponen-Komponen Sistem penunjang keputusan**

Menurut Sudyantoro dalam Pratiwi (2013:09-46), komponen-komponen dalam SPK meliputi 8 (delapan) bagian, yaitu :

1. *Hardware Resources*
2. *Software Resources*
3. Sumber Data
4. Sumber Model
5. Sumber Daya Manusia
6. Model Sistem penunjang keputusan
7. *Electronic Spreadsheet*
8. Sistem penunjang keputusan Kelompok

### **II.1.3. Tujuan Sistem penunjang keputusan**

Menurut Sudyantoro dalam Pratiwi (2013:09-46), tujuan dari sistem penunjang keputusan adalah :

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 daripada perbaikan efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.
5. Peningkatan produktivitas. Membangun satu kelompok pengambil keputusan, terutama para pakar, bisa sangat mahal. Pendukung terkomputerisasi bisa mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berada di berbagai lokasi yang berbeda beda (menghemat biaya perjalanan). Selain itu, produktivitas staf pendukung (misalnya analisis keuangan dan hukum) bisa ditingkatkan. Produktivitas juga bisa ditingkatkan menggunakan peralatan optimalisasi yang menentukan cara terbaik untuk menjalankan sebuah bisnis.

## **II.2. Metode SAW (*Simple Additive Weighting*)**

Metode SAW (*Simple Additive Weighting*) adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Inti dari SAW 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, et al., 2006).

Adapun langkah penyelesaian dalam menggunakan metode SAW menurut Kusumadewi (2006) adalah:

1. Menentukan alternatif (kandidat), yaitu  $A_i$ .
2. Menentukan kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu  $C_j$ .
3. Memberikan nilai rating kecocokan setiap alternatif pada setiap kriteria.
4. Menentukan bobot preferensi atau tingkat kepentingan ( $W$ ) setiap kriteria.

$$W = [ W_1, W_2, W_3, \dots, j, ] \quad (2.1)$$

5. Membuat tabel rating kecocokan dari setiap alternatif pada setiap kriteria.
6. Membuat matrik keputusan  $X$  yang dibentuk dari tabel rating kecocokan dari setiap alternatif pada setiap kriteria. Nilai  $x$  setiap alternatif ( $A_i$ ) pada setiap kriteria ( $C_j$ ) yang sudah ditentukan, dimana,  $i=1,2,\dots,m$   $j=1,2,\dots,n$ .

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1j} \\ \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{ij} \end{bmatrix} \quad (2.2)$$

7. Melakukan normalisasi matrik keputusan X dengan cara menghitung nilai rating kinerja ternormalisasi ( $r_{ij}$ ) dari alternatif  $A_i$  pada kriteria  $C_j$ .

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i x_{ij}} \\ \frac{\text{Min}_i x_{ij}}{x_{ij}} \end{cases} \quad (2.3)$$

Jika j adalah kriteria keuntungan (*benefit*)

Jika j adalah kriteria biaya (*cost*)

Keterangan :

- a. Dikatakan kriteria keuntungan apabila nilai  $x_{ij}$  memberikan keuntungan bagi pengambil keputusan, sebaiknya kriteria biaya apabila  $x_{ij}$  menimbulkan biaya bagi pengambil keputusan.
  - b. Apabila berupa kriteria keuntungan maka nilai  $x_{ij}$  dibagi dengan nilai  $\text{Max}(x_{ij})$  dari setiap kolom, sedangkan untuk kriteria biaya, nilai  $\text{Min}(x_{ij})$  dari setiap kolom dibagi dengan nilai  $x_{ij}$ .
8. Hasil dari nilai rating kinerja ternormalisasi ( $r_{ij}$ ) membentuk matrik ternormalisasi (R)

$$R = \begin{bmatrix} r_{11} & \cdots & r_{1j} \\ \vdots & \ddots & \vdots \\ r_{i1} & \cdots & r_{ij} \end{bmatrix} \quad (2.4)$$

9. Hasil akhir nilai preferensi ( $V_i$ ) diperoleh dari penjumlahan dari perkalian elemen baris matrik ternormalisasi ( $R$ ) dengan bobot preferensi ( $W$ ) yang bersesuaian elemen kolom matrik ( $W$ ).

$$V_i = \sum_{j=1}^n W_{ij} r_{ij} \quad (2.5)$$

Hasil perhitungan nilai ( $V_i$ ) yang lebih besar mengindikasikan bahwa alternatif  $A_i$  merupakan alternatif terbaik.

10. Menentukan Nilai Indikasi

Nilai indikasi dilakukan pada *hidden layer*, yang berfungsi sebagai nilai pengangkatan jabatan karyawan menggunakan kriteria jenjang pendidikan, tingkat kehadiran, disiplin, kerjasama tim, komunikasi, pengetahuan dan kerapian dalam berpakaian.

11. Perangkingan dilakukan dengan cara mengalikan nilai SAW dengan nilai indikasi dan hasil akhir dari nilai akan dirangking sesuai urutan hasil yang mempunyai nilai paling besar sampai yang terkecil.

### II.3. Visual Basic .Net

Visual basic 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh Microsoft, yaitu microsoft Visual Studio 2010. Sebagai produk lingkungan pengembangan terintegrasi atau IDE andalan yang di keluarkan oleh microsoft, visual studio 2010 menambahkan perbaikan-perbaikan fitur dan fitur baru yang lebih lengkap visual studio pendahulunya, yaitu mirosoft visual studio 2008. (Wahana Komputer;2010;2)

Sedangkan menurut Aswan (2012 : 1) Visual basic 2010 adalah salah satu bagian dari microsoft visual studio 2010. Sebuah alat yang digunakan oleh pengembang windows dari berbagai level untuk mengembangkan dan membangun aplikasi yang bergerak diatas sistem .NET Framework, dengan menggunakan bahasa BASIC. Visual Basic menyediakan cara cepat dan mudah untuk membuat aplikasi.

Setiap generasi baru dari perangkat lunak bahasa pemrograman datang karena adanya keterbatasan dari generasi sebelumnya. Teknologi *device, hardware, network* dan internet baru yang muncul menyebabkan bahasa pemrograman yang ada tidak lagi menjadi alat yang ideal untuk mengembangkan perangkat lunak yang dapat bekerja dengan teknologi baru tersebut. Sekarang untuk pertama kalinya, platform pengembang perangkat lunak yang lengkap, Microsoft .NET telah didesain dari dasar dengan internet sebagai fokus utamanya (walaupun tidak secara eksklusif hanya untuk pengembang internet saja). Banyak inovasi baru yang berada dalam platform ini akan mengatasi keterbatasan dari tool-tool dan teknologi lama. Visual Basic .NET adalah pengembangan dari Visual basic sebelumnya. Kelebihan VB .NET 2010 terletak pada tampilannya yang lebih canggih dibandingkan dengan edisi Visual Basic sebelumnya. Selain memiliki kelebihan, VB .NET 2010 memiliki kekurangan. Kekurangan VB .NET 2010 yang terlihat jelas adalah beratnya aplikasi ini apabila dijalankan pada komputer yang memiliki spesifikasi sederhana.

## II. 4. 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. Menurut cara pemrosesan data pada prosesor maka *Microsoft* mengelompokkan produk ini berdasarkan 2 jenis yaitu :

- Versi 32-bit(x86), yang biasanya digunakan untuk komputer dengan *single prosesor* (Pentium 4) atau lebih tepatnya *prosesor* 32 bit dan sistem operasi *Windows XP*.
- Versi 64-bit(x64), yang biasanya digunakan untuk komputer dengan lebih dari satu *prosesor* (Misalnya *Core 2 Duo*) dan sistem operasi 64 bit seperti *Windows XP 64*, *Vista*, dan *Windows 7*. Sedangkan secara keseluruhan terdapat versi-versi seperti berikut ini:
  - *Versi Compact*, ini adalah versi “Tipis” dari semua versi yang ada. Versi ini seperti versi *desktop* pada *SQL Server 2000*. Versi ini juga digunakan pada *handled drvice* seperti *Pocket PC*, *PDA*, *SmartPhone*, *Tablet PC*.

- *Versi Express*, ini adalah versi “Ringan” dari semua versi yang ada (tetapi versi ini berbeda dengan versi *compact*) dan paling cocok untuk latihan, *Express Manager* standar, integrasi dengan *CLR* dan *XML*.

*Microsoft SQL Server* merupakan salah satu *database relational* yang banyak digunakan oleh dunia usaha. (Agus Tinus Setiawan, Rika Putri Permadani: 2016).

## **II.5. Pemasok**

Pemasok merupakan suatu perusahaan atau individu yang mampu untuk menyediakan sumber daya, baik dalam bentuk barang atau jasa yang dibutuhkan oleh perusahaan lainnya. Pemasok merupakan komponen penting dibidang logistik dan manajemen produksi. Untuk memperoleh pemasok yang mampu memenuhi barang atau jasa sesuai permintaan, diperlukan proses pemilihan pemasok yang baik. Tujuan dari pemilihan pemasok yaitu untuk memperoleh pemasok yang tepat sehingga dapat mengurangi biaya pembelian barang atau jasa. Pemilihan pemasok yang salah, dapat merugikan perusahaan. Untuk itu pemilihan pemasok merupakan komponen penting yang harus dilakukan dalam suatu perusahaan. (Suci Oktri Viarani; 2015)

## **II.6. UML (Unified Modelling Language)**

*Unified Modeling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan

diagram dan teks-teks pendukung.

*UML* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

*UML* hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek (Rosa A.S dan M. Shalahudin, 2014:133).

### **II.6.1. Sejarah *UML***

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS.

Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Karena banyaknya metodologi-metodologi yang berkembang pesat saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep, seperti konsep *Object Modeling Technique (OMT)* dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators (CRC)* dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep lainnya dimana

James R. Rumbaigh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama *Rational Software Corporation* menghasilkan bahasa yang disebut dengan *Unified Modeling Language (UML)*.

Pada tahun 1996, *Object Management Group (OMG)* mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 *UML* diakomodasi oleh *OMG* sehingga sampai saat ini *UML* telah memberikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait di dalamnya (Rosa A.S dan M. Shalahudin, 2014:138).

### **II.6.2. Diagram UML**

Rosa A.S dan M. Shalahudin (2014:140), pada *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1. *Structure diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure diagram* terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram* dan *deployment diagram*.
2. *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. *Behavior diagram* terdiri dari *Use case diagram*, *Activity diagram*, *State Machine System*.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar

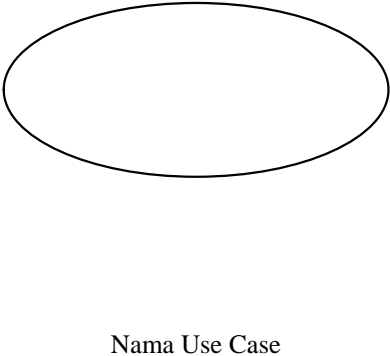
subsistem pada suatu sistem. *Interaction diagram* terdiri dari *Sequence Diagram*, *Communication Diagram*, *Timing Diagram*, *Interaction Overview Diagram*

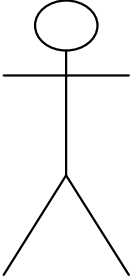

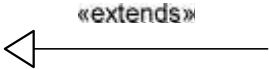
### II.6.3. Use Case Diagram

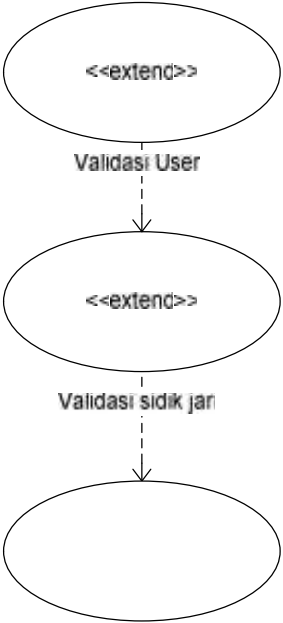
Rosa dan M. Shalahudin (2014:155), *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

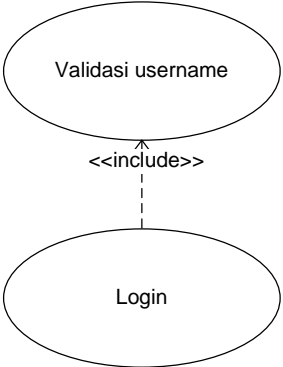
Adapun simbol-simbol yang ada di diagram *use case* pada tabel II.1 :

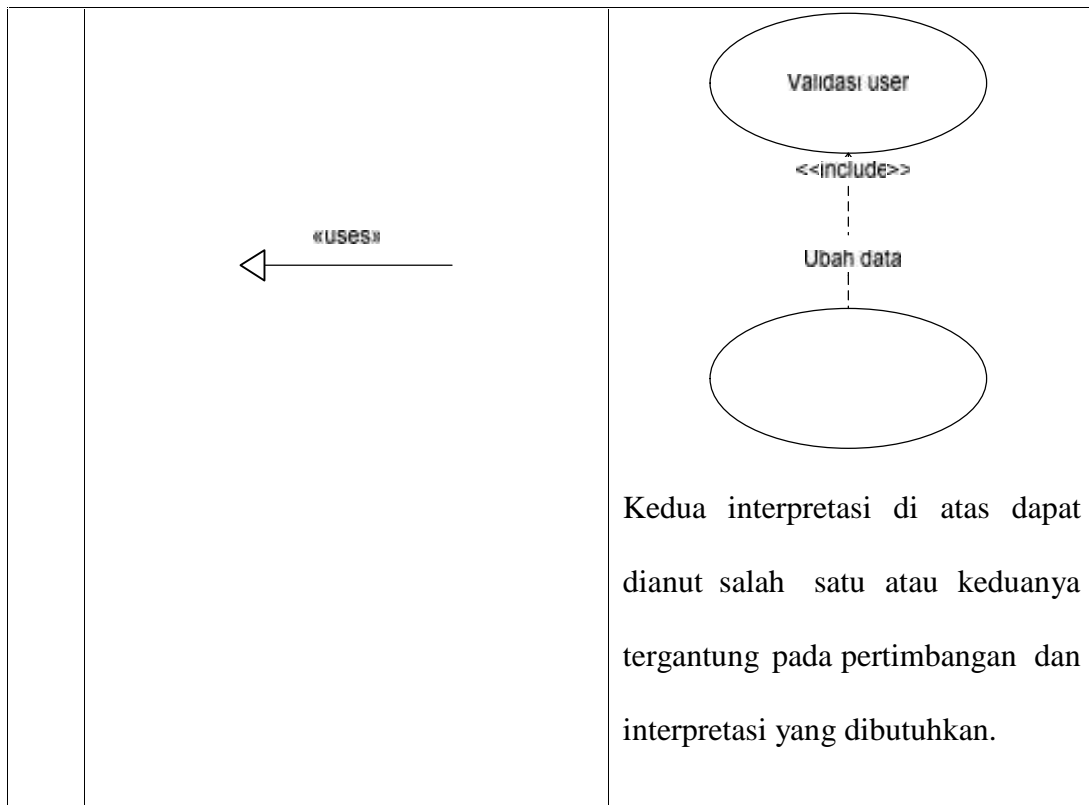
**Tabel II.1. Simbol Use Case Diagram**

No.	Simbol	Deskripsi
1.	Use Case  	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau faktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>
2.	Aktor atau <i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem

		<p>informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.</p>
3.	<p>Asosiasi atau Association</p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
4.	<p>Ekstensi atau Extend</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p>

		<p style="text-align: center;"><b>Validasi user name</b></p>  <p style="text-align: center;">Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extendnya</i> merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
5.	Menggunakan atau <i>include</i> atau <i>uses</i>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>usecase</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>usecase</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada dua sudut pandang yang</p>

	<p>&lt;&lt;include&gt;&gt;</p>	<p>cukup besar mengenai <i>include</i> di <i>use case</i> :</p> <ul style="list-style-type: none"><li>- <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut :</li></ul>  <pre>graph BT; Login((Login)) -.-&gt; &lt;&lt;include&gt;&gt;  ValidasiUsername((Validasi username));</pre> <p><i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan di jalankan, misal pada kasus berikut :</p>
--	--------------------------------	--





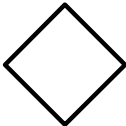


**Sumber : Rosa A.S dan M. Shalahudin (2014:162)**

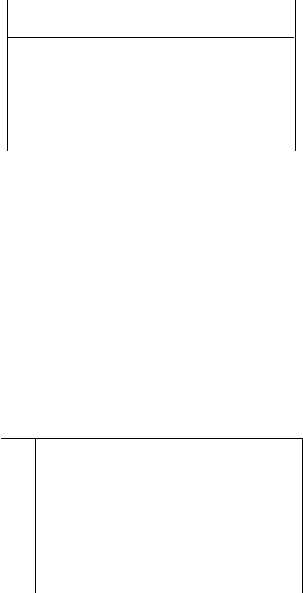
#### **II.6.4. Activity Diagram**

Rosa dan M. Shalahudin (2014:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Adapun simbol-simbol yang ada di diagram aktivitas pada tabel II.2 :

Tabel II.2. Simbol *Activity Diagram*

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktifitas sistem, sebuah diagram aktifitas memiliki sebuah status awal.
2.	Aktifitas 	Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja.
3.	Percabangan atau <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.
4.	Penggabungan atau <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktifitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktifitas memiliki sebuah status akhir.

6.	<p><b>Swimlane</b></p>  <p>Atau</p>	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.</p>
----	--	---

**Sumber : Rosa A.S dan M. Shalahudin (2014:162)**

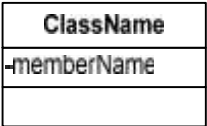

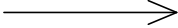
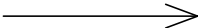
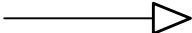
### **II.6.5. Class Diagram**

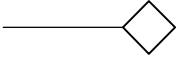
Rosa dan M. Shalahudin (2014:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method* :

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Adapun simbol-simbol yang ada di diagram kelas pada tabel II.4 :

Tabel II.3. Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.	Kelas  	Kelas pada struktur sistem.
2.	Asosiasi atau Association  	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
3.	Asosiasi berarah atau <i>directed</i>  	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Generalisasi  	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
5.	Kebergantungan atau <i>dependency</i>  	Relasi antar kelas dengan makna kebergantungan antar kelas.

6.	Agregasi atau <i>aggregation</i>  	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )
----	---	--

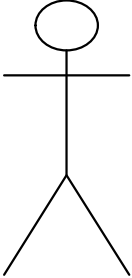
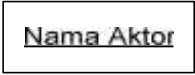

**Sumber : Rosa A.S dan M. Shalahudin (2014:146)**


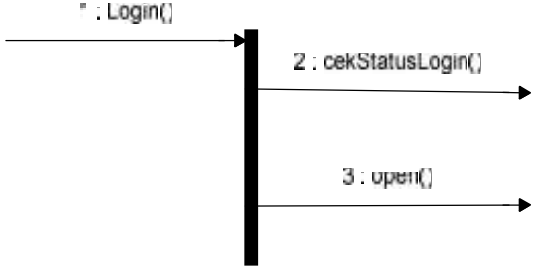
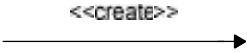
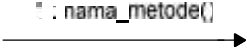
### **II.6.6. Sequence Diagram**

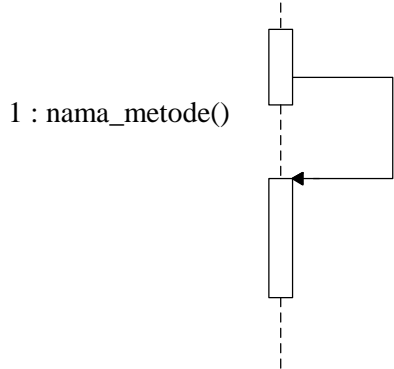
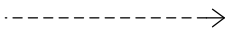
Rosa dan M. Shalahudin (2014:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

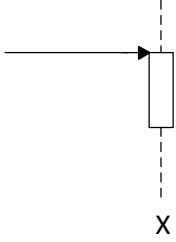
Berikut adalah simbol-simbol yang ada di diagram sekuen pada tabel II.5 :

Tabel II.4. Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<p data-bbox="379 454 454 483">Aktor</p>  <p data-bbox="379 898 432 927">Atau</p>  <p data-bbox="496 1137 719 1167">Tanpa waktu aktif</p>	<p data-bbox="774 454 1369 920">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.</p>
2.	<p data-bbox="379 1256 691 1285">Garis hidup atau <i>lifeline</i></p> 	<p data-bbox="774 1256 1238 1285">Menyatakan kehidupan suatu objek.</p>
3.	<p data-bbox="379 1648 459 1677">Objek</p> <p data-bbox="395 1783 730 1812"><u>Nama Objek : nama kelas</u></p>	<p data-bbox="774 1648 1334 1677">Menyatakan objek yang berinteraksi pesan.</p>

		
4.	Waktu aktif	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misal</p> <div style="text-align: center;">  </div> <p>Maka cek StatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif.</p>
5.	Pesan tipe create	<p>Menyatakan suatu objek membuat objek yang lain, arah panah pengarah pada objek yang dibuat.</p> <div style="text-align: center;">  </div>
6.	Pesan tipe call	<p>Menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri</p> <div style="text-align: center;">  </div>

		<p>1 : nama_metode()</p> <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
7.	<p>Pesan tipe send</p> <p>1 : masukkan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
8.	<p>Pesan tipe return</p>	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode</p>

	<p>1 : keluaran</p> <p>-----&gt;</p>	<p>menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
9.	<p>Pesan tipe destroy</p> <p>&lt;&lt;destroy&gt;&gt;</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i></p>

**Sumber : Rosa A.S dan M. Shalahudin (2014:165)**