

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Sistem**

##### **II.1.1. Pengertian Sistem**

Mempelajari suatu sistem akan lebih mengena bila mengetahui terlebih dahulu apakah sistem itu. Pengertian tentang sistem pertama kali dapat diperoleh dari definisi sistem itu sendiri. Jika kita perhatikan dengan seksama diri kita juga terdiri dari berbagai sistem yang berfungsi untuk mengantarkan kita pada tujuan hidup kita. (Tam Sutabri, 2012 : 14).

Gordon B. Davis dalam bukunya menyatakan bahwa sistem bisa berupa abstrak atau fisik. Sistem yang abstrak adalah susunan gagasan-gagasan atau konsepsi yang teratur yang saling bergantung. Misalnya sistem teologi adalah susunan yang teratur dari gagasan-gagasan tentang Tuhan, Manusia dan lain sebagainya. Sedangkan sistem yang bersifat fisik adalah serangkaian unsur yang bekerja sama untuk mencapai suatu tujuan. (Tata Sutabri, 2012 : 6).

Berikut ini merupakan pengertian sistem dari beberapa ahli :

1. Jerry FilhGerald “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan dan berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu”.
2. Ludwig Von Bartalanfy “Sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan”.

3. Anatol Rapoport “Sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lain”.
4. L. Ackoff “Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya”.

### **II.1.3. Karakteristik Sistem**

Model umum sebuah sistem terdiri dari input, proses dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. (Tata Sutabri, 2012 : I3-14).

Adapun karakteristik yang dimaksud adalah sebagai berikut :

#### **1. Komponen Sistem (*component*)**

Suatu sistem terdiri dari sejumlah komponen yang berinteraksi, yang bekerja sama membentuk satu kesatuan. komponen komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem sistem yang lebih besar yang disebut dengan supra sistem.

#### **2. Batasan Sistem (*boundary*)**

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lain atau sistem dengan lingkungan luarnya. Batasan sistem

ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. **Lingkungan Luar Sistem**  
(Environment)

Bentuk apapun yang ada diluar lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem, lingkungan luar sistem ini dapat menguntungkan dan dapat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara sedangkan lingkungan luar yang merugikan harus dikendalikan kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

4. **Penghubung Sistem (*interface*)**

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem *interface*. Penghubung ini memungkinkan sumber sumber daya mengalir dari satu subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung, dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. **Masukan Sistem (*input*)**

Energi yang dimasukkan kedalam sistem tersebut masukan sistem yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh didalam suatu unit sistem komputer "program" adalah *maintenance*

*input* yang digunakan untuk mengoperasikan komputer, sementara "data" adalah *signal* yang akan diolah menjadi informasi.

6. Keluaran Sistem (*output*)

Hasil dari energi yang diolah dan diklarifikasikan menjadi keluaran yang berguna, keluaran ini merupakan bagian subsistem yang lain seperti contoh sistem informasi keluaran yang dihasilkan adalah informasi dimana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal lain yang merupakan input bagi subsistem lainnya.

7. Pengolahan Sistem (*Procces*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran sebagai contoh sistem akuntansi sistem ini akan mengolah data transaksi menjadi laporan laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya, suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

## **II.2. Sistem Pendukung Keputusan**

Sistem pendukung keputusan adalah suatu sistem berbasis komputer dengan antar muka antara mesin/komputer dengan pengguna. Sistem pendukung keputusan ditujukan untuk membantu dalam menyelesaikan suatu masalah dalam berbagai level manajemen dan bukan untuk menggantikan manusia sebagai

pembuat keputusan. Sistem pendukung keputusan mampu memberi alternatif solusi bagi masalah semi terstruktur atau tidak terstruktur baik bagi perseorangan atau kelompok dan dalam berbagai macam proses dan gaya pengambilan keputusan. Sistem pendukung keputusan menggunakan data, basis data, dan analisa model model keputusan. Sistem pendukung keputusan bersifat efektif, adiktif, interaktif, *easy to use* dan *fleksibel* serta menyediakan akses terhadap berbagai macam format dan tipe sumber data (*data source*). (kardiaman, 2014 : 64).

### **II.3. Metode *Profile Matching***

*Profile matching* merupakan suatu proses yang sangat penting dalam manajemen SDM dimana terlebih dahulu ditentukan kompetensi (kemampuan) yang diperlukan oleh suatu jabatan. Kompetensi/kemampuan tersebut haruslah dapat dipenuhi oleh pemegang/calon pemegang jabatan.

Dalam proses *profile matching* secara garis besar merupakan proses membandingkan antara kompetensi individu kedalam kompetensi jabatan sehingga dapat diketahui perbedaan kompetensinya (disebut juga *gap*), semakin kecil *gap* yang dihasilkan maka bobot nilainya semakin besar yang berarti memiliki peluang lebih besar untuk karyawan menempati posisi tersebut. (Safrianto, 2015 : 61 ).

#### **II.3.1. Proses Perhitungan Metode *Profile Matching***

Menurut (Nina Sherly : 2013 : 44 - 45) adapun langkah langkah dalam penyelesaian metode *profile matching* adalah sebagai berikut :

### 1. Perhitungan *Gap* Kompetensi

Setelah proses pemilihan kandidat, proses berikutnya adalah menentukan kandidat mana yang paling cocok menduduki tempat sebagai karyawan yang akan menerima bonus yang diajukan oleh perusahaan. Dalam kasus ini penulis menggunakan perhitungan pemetaan *gap* kompetensi dimana yang dimaksud dengan *gap* disini adalah beda antara profil bonus dengan profil karyawan atau dapat ditunjukkan pada rumus di bawah ini:

$$\text{Gap} = \text{Profil Karyawan} - \text{Profile Bonus} \dots\dots\dots(1)$$

### 2. Perhitungan Pemetaan *GAP*

Kompetensi Berdasarkan Aspek Aspek Untuk perhitungan pemilihan karyawan pengumpulan *gap-gap* yang terjadi itu sendiri pada tiap aspeknya mempunyai perhitungan yang berbeda-beda. Setelah didapatkan tiap *gap* masing-masing karyawan maka tiap profil karyawan diberi bobot nilai dengan patokan tabel bobot nilai *gap* seperti yang dapat dilihat pada tabel II.1.

**Tabel II.1. Keterangan Bobot Nilai Gap**

No	Selisih	Bobot Nilai	Keterangan
1	0	5	Tidak ada selisih (kompetensi sesuai dengan yang dibutuhkan ).
2	1	4,5	Kompetensi individu kelebihan 1 tingkat/level
3	-1	4	Kompetensi individu kekurangan 1 tingkat/level
4	2	3,5	Kompetensi individu kelebihan 2 tingkat/level

5	-2	3	Kompetensi individu kekurangan 2 tingkat/level
6	3	2,5	Kompetensi individu kelebihan 3 tingkat/level
7	-3	2	Kompetensi individu kekurangan 3 tingkat/level
8	4	1,5	Kompetensi individu kelebihan 4 tingkat/level
9	-4	1	Kompetensi individu kekurangan 4 tingkat/level

(Sumber : Nina Sherly : 2013)

### 3. Perhitungan dan Pengelompokan *Core* dan *Secondary Factor*

Setelah menentukan bobot nilai *gap* untuk setiap aspek. Kemudian tiap aspek dikelompokkan menjadi 2 (dua) kelompok yaitu kelompok *Core Factor* dan *Secondary Factor*. Untuk perhitungan *core factor* dapat ditunjukkan pada rumus di bawah ini:

$$NCF = \frac{NC(I,s,p)}{IC} \dots\dots\dots(2)$$

Keterangan:

NCF : Nilai rata-rata *core factor*

NC(*i, s, p*) : Jumlah total nilai *core factor*

(*Intelektual, Sikap kerja, Perilaku*)

IC : Jumlah *item core factor*

Sedangkan untuk perhitungan *secondary factor* dapat ditunjukkan pada rumus di bawah ini:

$$NSF = \frac{NS(I,s,p)}{IS} \dots\dots\dots(3)$$

Keterangan:

NSF : Nilai rata-rata *secondary factor*

NS(*i, s, p*) : Jumlah total nilai *secondary factor*  
(*Intelektual, Sikap kerja, Perilaku*)

IS : Jumlah *item secondary factor*.

#### 4. Perhitungan Nilai Total

Dari hasil perhitungan dari tiap aspek di atas kemudian dihitung nilai total berdasarkan presentasi dari *core* dan *secondary* yang diperkirakan berpengaruh terhadap kinerja tiap-tiap profil. Contoh perhitungan dapat dilihat pada rumus di bawah ini:

$$N(i, s, p) = (x)\% NCF(i, s, p)(x)\% NSF(i, s, p) \dots\dots\dots(4)$$

Keterangan:

(*i, s, p*) : (*Intelektual, Sikap Kerja, Perilaku*)

$N(i, s, p)$  : Nilai total dari aspek

$NCF(i, s, p)$  : Nilai rata-rata *core factor*

$NSF(i, s, p)$  : Nilai rata-rata *secondary factor*

(*x*)% : Nilai persen yang diinputkan

#### 5. Perhitungan Penentuan Hasil Akhir atau Ranking

Hasil akhir dari proses ini adalah *ranking* dari kandidat yang dijukan untuk pemberian bonus karyawan. Penentuan *ranking* mengacu pada hasil perhitungan tertentu. Perhitungan tersebut dapat ditunjukkan pada rumus di bawah ini:

$$Ha = (x)\% Ni + (x)\%Ns + (x)\%Np \dots\dots\dots(5)$$

Keterangan:

*Ha* : Hasil Akhir

*Ni* : Nilai Kapasitas Intelektual

*Ns* : Nilai Sikap Kerja

*Np* : Nilai Perilaku

(*x*)% : Nilai Persen yang *diinputkan*

#### II.4. *SQL Server 2008*

*SQL Server 2008* adalah sebuah terobosan baru dari *Microsoft* dalam bidang *database*. *SQL Server* adalah sebuah 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* membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data (Wahana Komputer,

2012: 2).

## **II.5. Microsoft Visual Studio 2010**

*Visual Studio 2010* merupakan sebuah *Integrated Development Environment* (IDE) atau lingkungan kerja yang digunakan untuk membangun aplikasi .NET dengan mudah. *Visual Studio Profesional 2010* menyediakan berbagai *tool*, antara lain *tool toolbox* yang berisi komponen visual, sehingga anda tinggal *drag and drop* komponen dan *Visual Studio 2010* akan menuliskan kode untuk anda. Selain itu masih ada jendela *wizard* membantu anda untuk melakukan pemrograman dengan sangat mudah (Wahana Komputer, 2012:7).

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

*Unified Modeling Language* (UML) adalah standarisasi bahasa pemodelan untuk membangun perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. Diagram-diagram yang digunakan pada UML antara lain adalah *class diagram*, *object diagram*, *use case diagram*, *activity diagram*, dan *sequence diagram* (Winda Aprianti dan Umi Maliha, 2016 :22).

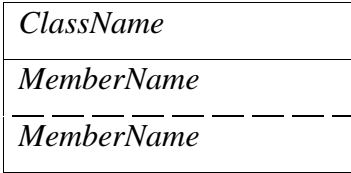


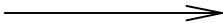

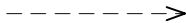

### **1. Class Diagram**

Menurut Rosa dan Shalahuddin, 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* :

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

Adapun keterangan simbol-simbol yang ada pada diagram kelas dapat

dilihat pada gambar II.1

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem
2.	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
6.	Kebergantungan/ <i>dependensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
7.	Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )

**Gambar II.1. Class Diagram**  
 (Sumber : Winda Aprianti dan Umi Maliha, 2016 : 22)

## 2. Use Case Diagram


Menurut Rosa dan Shalahuddin, *use case* atau diagram *use case* merupakan



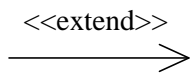
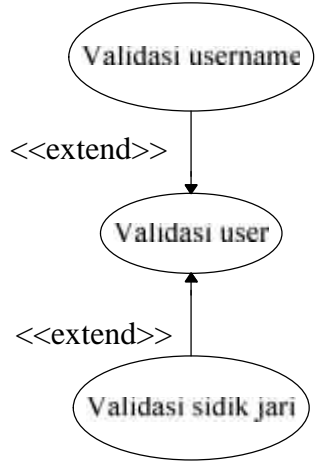
pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.


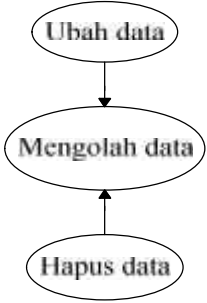
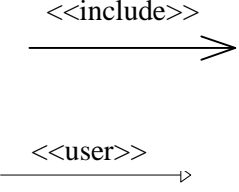
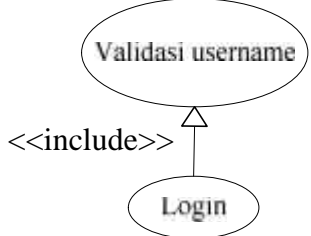
Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal pada *use case* yaitu pendefenisiaan apa yang disebut aktor dan *use case*.

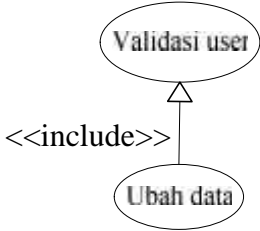
- a. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah gambar II.2 merupakan simbol-simbol yang ada pada diagram *use case*:

No	Simbol	Deskripsi
1.	<p><i>Usecase</i></p> 	Fungsionalitas yang disediakan system sebagai unit-unit yang saling bertukar pesanan tarunit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
2.	Aktor / <i>actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan system informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri,

	 Nama aktor	jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
3.	Asosiasi / <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.	Exstensi/extend 	<p>Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, missal:</p>  <pre> graph TD     A(Validasi user) -- "&lt;&lt;extend&gt;&gt;" --&gt; B(Validasi username)     C(Validasi sidik jari) -- "&lt;&lt;extend&gt;&gt;" --&gt; A       </pre> <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i></p>

		yang menjadi induknya.
5.	<p>Generalisasi</p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya.</p>
6.	<p>Menggunakan <i>/include/uses</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> :</p> <ol style="list-style-type: none"> <li>1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, missal pada kasus berikut :</li> </ol> 

		<p>2. <i>Include</i> berarti <i>use case</i> tambahan yang akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan di jalankan, misal pada kasus berikut :</p>  <pre> graph BT     UC1((Ubah data)) -- "&lt;&lt;include&gt;&gt;" --&gt; UC2((Validasi user))   </pre> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>
--	--	---






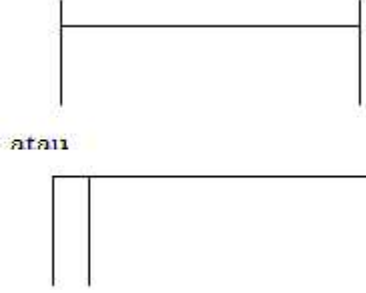
**Gambar II.2. Use Case Diagram**  
(Sumber : Winda Aprianti dan Umi Maliha, 2016 : 23)

### 3. Activity Diagram

Menurut Rosa dan Shalahuddin, 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 di sini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah gambar II.3 yaitu simbol-simbol yang ada pada diagram

aktivitas.

No	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

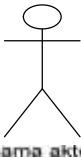

**Gambar II.3. Activity Diagram**  
(Sumber : Winda Aprianti dan Umi Maliha, 2016 : 23)



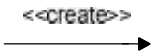

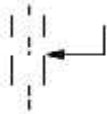
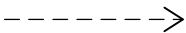
#### 4. Sequence Diagram

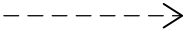
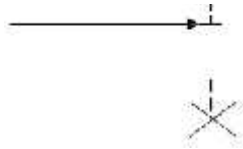
Menurut Rosa dan Shalahuddin, diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan

*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 di gambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya 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 pada diagram sekuen, yang dapat di lihat pada gambar II.4.

No	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>nama aktor</p> <p>atau <span style="border: 1px solid black; padding: 2px;">nama aktor</span></p> <p>tanpa waktu aktif</p>	<p>Orang, proses, atau system lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal</p>
2.	<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
3.	<p>Objek</p> <span style="border: 1px solid black; padding: 2px;">nama objek : nama kelas</span>	<p>Menyatakan objek yang berinteraksi</p> <p>Pesan</p>

4.	<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya:</p>  <p>Maka cekStatusLogin() dan open() dilakukan di dalam metode login(). Aktor tidak memiliki waktu aktif</p>
5.	<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
6.	<p>Pesan tipe <i>call</i></p> <p>1 : nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.</p> <p>1 : nama_metode()</p>  <p>Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi / metode yang di panggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
7.	<p>Pesan tipe <i>send</i></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.	Pesan tipe return  1 : keluaran  	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9.	Pesan tipe <i>destroy</i> << <i>destroy</i> >> 	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> .

**Gambar II.4. *Sequence Diagram***  
 (Sumber : Winda Aprianti dan Umi Maliha; 2016 : 23)