

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Konsep Dasar Sistem**

Dalam bab ini diperlukan teori-teori yang dapat mendukung kemudahan dalam mempelajari rancang sistem aplikasi yang diharapkan dapat digunakan seoptimal mungkin. Dengan demikian akan sangat membantu untuk mempermudah proses persediaan barang pada PT Amanah Sejahtera Keluarga Abadi Cikampek.

##### **2.1.1. Sistem**

Sistem berasal dari bahasa Yunani, yaitu *systema*, yang artinya himpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan suatu keseluruhan. Selain itu, bisa diartikan sekelompok elemen yang independen, namun saling berkaitan sebagai suatu kesatuan. Sistem terdiri atas struktur dan proses. Struktur sistem merupakan unsur-unsur yang membentuk sistem tersebut, sedangkan proses sistem menjelaskan cara kerja setiap unsur sistem dalam mencapai tujuan. Ada beberapa ahli yang mendefinisikan sistem sebagai berikut:

“Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas sama-sama”. Menurut I Putu Agus Eka Pratama (2014:7).

Menurut Pratama (2016:75), dalam jurnal Prabowo dan Mamay Syani menjelaskan bahwa: “Sistem didefinisikan sebagai kumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama”.

Menurut Kadir (2014:61), “Sekumpulan elemen yang saling terkait dan terpadu yang dimaksud untuk mencapai suatu tujuan”.

Didalam suatu sistem memiliki karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen (*component*), batas sistem (*boundary*), lingkungan luar sistem (*environment*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*) dan sasaran (*objectives*) atau tujuan (*goal*).

#### 1. Komponen (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem dapat berupa benda nyata ataupun abstrak. Komponen sistem disebut juga subsistem.

#### 2. Batas Sistem (*Boundary*)

Batas Sistem diperlukan untuk membedakan antara satu sistem dengan sistem yang lainnya. Tanpa adanya batas sistem, sangat sulit untuk memberikan batasan scope tinjauan terhadap sistem.

#### 3. Lingkungan Luar sistem (*Environment*)

Lingkungan sistem adalah segala sesuatu yang berada di luar sistem lingkungan sistem yang dapat menguntungkan ataupun merugikan.

#### 4. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya.

#### 5. Masukan Sistem (*Input*)

Masukan sistem merupakan energi yang di masukan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance*) dan masukan sinyal (*signal*).

#### 6. KeluaranSistem(*Output*)

Keluaran (*Output*) adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

#### 7. PengolahSistem(*Process*)

Suatu sistem dapat memiliki bagian pengolah atau sistem tersebut sebagai pengolahnya. Pengolah yang akan mengubah masukan menjadi keluaran.

#### 8. SasaranatauTujuan(*goal*)

Suatu sistem dikatakan berhasil apabila mengenai sasaran atau tujuannya.

SistemBerorientasiObjek (OOB) merupakan suatucarabagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengolahan kompleksitas. Menurut Rosa A.S dan M. Shalahuddin (2016:100), “Berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan oprasi yang diperlakukan terhadapnya”.

#### 2.1.2. Website

Padadasarnyaweb site adalahkepanjangandari *World Wide Web* (WWW).informasi WWW inidisimpanpadaweb server untukdapatdiaksesdarijaringan *browser* terlebihdahulu, seperti *Internet Explorer* atau *Mozilla Firefox*.

Menurut Kadir (2014:310), *Word Wide Web* (WWW) adalah sistem pengakses informasi dalam *internet* yang biasadikenaldengan istilah *web*.

*Web* menggunakan protokol yang disebut HTTP (*HyperText Transfer Protocol*) yang berjalan pada TCP/IP. Dengan menggunakan *HyperText*, pemakai dapat melompat dari suatu dokumen ke dokumen lain dengan mudah, dengancukupmengklik *text-text* khusus yang pada awalnya ditandai dengan garis bawah.

Penggunaan *HyperText* pada *web* jugatelahdikembangkan lebih jauh menjadi *HyperMedia*. Dengan menggunakan pendekatan *HyperMedia*, tidak hanya *text* yang dapat dikaitkan, melainkan juga gambar, suara, dan bahkan video.

Beberapa istilah yang ada dalam *website*:

#### 1. Internet

*Internet* mempunyai jaringan yang besar dan luas yang saling berhubungan dari jaringan-jaringan komputer yang menghubungkan orang-orang dan komputer-komputer diseluruh dunia, melalui telepon, satelit dan sistem-sistem komunikasi yang lain.

Menurut Priyanto (2015:A-85) dalam jurnal Samudi dkk menjelaskan bahwa: “*Internet* adalah (*Interconnected Network*) kumpulan jaringan komputer diseluruh dunia yang saling berhubungan satu dengan yang lainnya”.

#### 2. Web Browser

Menurut Arief (2014:128) dalam jurnal Eka Wida Fridayanthie dan Tias Mahdiati menjelaskan bahwa: “Aplikasi yang mampu menjalankan dokumen-dokumen *web* dengan cara diterjemakan”.

Prosesnya dilakukan oleh komponen yang terdapat didalam aplikasi *browser* yang biasa disebut *Web Engine*. Semua dokumen *web* ditampilkan oleh *browser* dengan cara diterjemahkan. Beberapa jenis *browser* yang populer saat ini diantaranya adalah *Internet Explorer* yang diproduksi oleh *Microsoft*, *Mozilla Firefox*, *Opera*, dan *Safari* yang diproduksi oleh *Apple*.

### 3. Web Server

Menurut Pratama (2015:448), *Web Server* dapat diartikan sebagai perangkat lunak yang dijalankan di sistem oprasi pada komputer *server* maupun dekstop, yang berfungsi untuk menerima permintaan (request) dalam bentuk protokol pada jaringan komputer.

#### 2.1.3. Basis Data (*database*)

Basis data adalah kumpulan *file-file* yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap *file* yang ada. Satu basis data menunjukkan kumpulan data yang dipakai dalam satu lingkup informasi. Dalam satu *file* terdapat beberapa *record-record* yang sejenis, sama besar, sama bentuk, merupakan satu kumpulan *entity* yang seragam. Satu *record* terdiri dari *field-field* yang saling berhubungan untuk menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan direkam dalam satu *record*. Suatu basis data berisi satu koleksi data yang saling berelasi dan satu set program untuk mengakses data tersebut.

Menurut Rosa A.S dan M. shalahuddin (2016:43), “Sistemterkomputerisasi yang tujuanutamanyaadalahmemelihara data yang sudahdiolahatauinformasidanmembuatinformasitersediaaatdibutuhkan”.

Menurut Abdul Kadir (2014:218), “Suatupengorganisasiansekumpulan data yang salingterkaitsehinggamemudahkanaktivitasuntukmemperolehinformasi”.

SedangkanmenurutPratama, (2016:79), “Tempatuntukpenyimpanandata,penyimpananinformasi (data yang telahdiolah), dan proses untukmanipulasi (pengelolaan) data menjadiinformasi, sertapengaksesan data daninformasi”.

#### 2.1.4. Model PengembanganPerangkatLunak

Menurut Rosa A.S dan M. Shalahuddin (2016:28-29), “Model *waterfall* sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*) model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau turut dimulai dari analisis, desain, pengkodean dan pengujian”. Yang terbagi menjadi lima tahapan, yaitu:

##### 1. AnalisaKebutuhanPerangkatLunak

Dalam proses pengumpulan kebutuhan yang dilakukan penulis secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahapan ini perlu untuk didokumentasikan.

##### 2. Desain

Desain perangkat lunak adalah proses multi langka yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Pada tahap

ini penulis melakukan mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

### 3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat penulis pada tahap desain.

### 4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan penulis untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

### 5. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang dilakukan penulis, tapi tidak untuk membuat perangkat lunak baru.

## 2.2 Teori Pendukung

Teori pendukung merupakan suatu penjelasan mengenai peralatan-peralatan pendukung yang dipakai dalam membuat suatu sistem usulan. Media tools sistem merupakan alat yang dapat digunakan untuk menggambarkan bentuk logika model dari suatu sistem, dimana simbol-simbol, lambang-lambang dan diagram-diagram menunjukkan secara tepat arti fisiknya.

### 2.2.1 ERD (*Entity Relationship Diagram*)

ERD (*Entity Relationship Diagram*) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Didalam hubungan ini dinyatakan yang utama dari ERD adalah menunjukkan objek data (*Entity*) dan hubungan (*Relationship*)

Menurut Rosa A.S dan M.Shalahuddin (2016:50), “*Entity Relationship Diagram* (ERD) adalah pemodelan awal basis data yang paling banyak digunakan”.

Komponen-komponen dalam ERD (*Entity Relationship Diagram*) adalah sebagai berikut:

a. Entitas (*Entity*)

Entitas merupakan data inti yang akan disimpan; bakal table pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama table.

b. Atribut

*Field* atau kolom data yang butuh disimpan dalam suatu entitas.

c. Atribut Kunci Primer

*Filed* atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

d. Atribut Multinilai (*Multivalue*)

*Field* atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

e. Relasi

Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.

f. Asosiasi (*Associstion*)

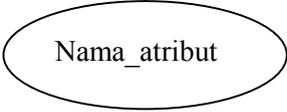
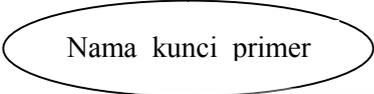
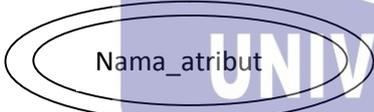
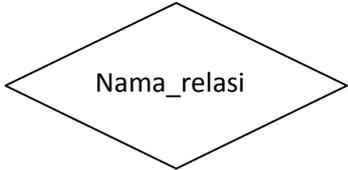
Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian.

UNIVERSITAS

Tabel II.1.

ERD (*Entity Relationship Diagram*)

Simbol	Deskripsi
<p data-bbox="304 1559 475 1592"><i>Entitas/entity</i></p> <div data-bbox="397 1753 659 1877" style="border: 1px solid black; padding: 5px; width: fit-content; margin: 20px auto;"> <p data-bbox="440 1787 608 1816">Nama_entitas</p> </div>	<p data-bbox="804 1559 1362 1957">Entitas merupakan data inti yang akan disimpan; bakal table pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi computer; penamaan entitas biasanya lebih ke kata benda dan belum</p>

	merupakan nama table
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)
Atribut multivalued/ <i>multivalued</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi/ <i>association</i> 	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah

	<p>pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas</p>
--	--

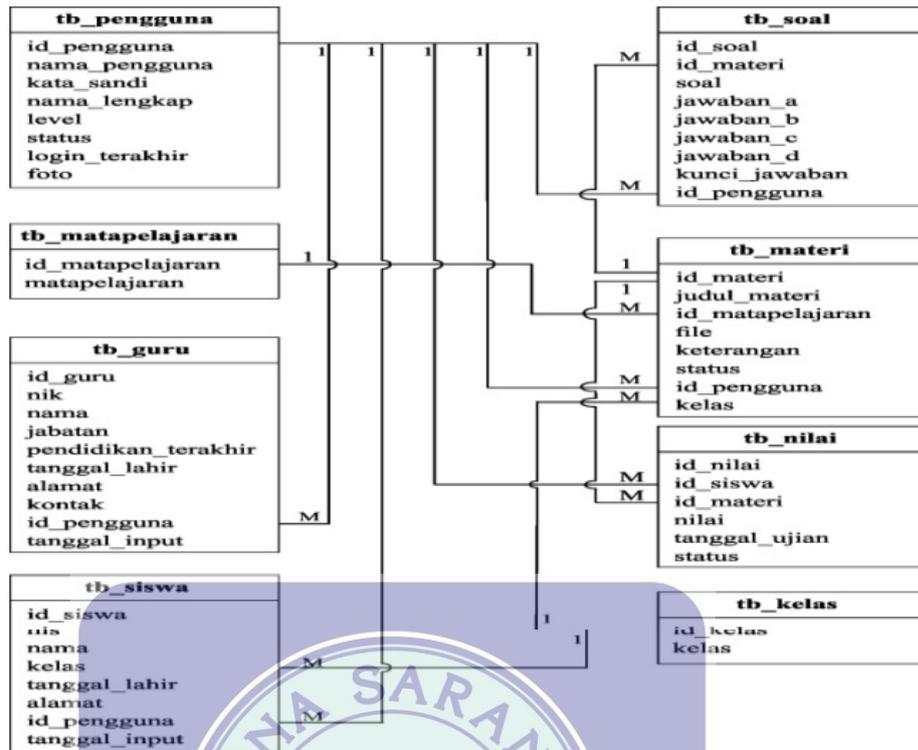
Sumber: Sukamto dan Shalahuddin (2016:50)

### 2.2.2 LRS (*logical Record Structure*)

Menurut Simarmata dan paryudi (2016:132-133) dalam jurnal Eka Wida Fridayanthie dan Tias Mahdiati menjelaskan bahwa: “*Logical Record Structured* (LRS) adalah representasi dari struktur *record-record* pada table-table yang terbentuk dari hasil relasi antar himpunan entitas”. Menentukan kardinalitas, jumlah table, dan *ForeignKey* (FK).

Berikut adalah cara membentuk skema *database* atau LRS (*Logical Record Structured*) berdasarkan *Entity Relationship Diagram* :

1. Jika relasinya satu-ke-satu maka *foreign key* diletakan pada salah satu dari dua entitas yang ada atau menyatukan kedua entitas tersebut.
2. Jika relasinya satu-ke-banyak, maka *foreign key* diletakan pada entitas *Many*
3. Jika relasinya banyak-ke-banyak, maka dibuat “*file konektor*” yang berisi dua *foreign key* yang berasal dari kedua entitas.

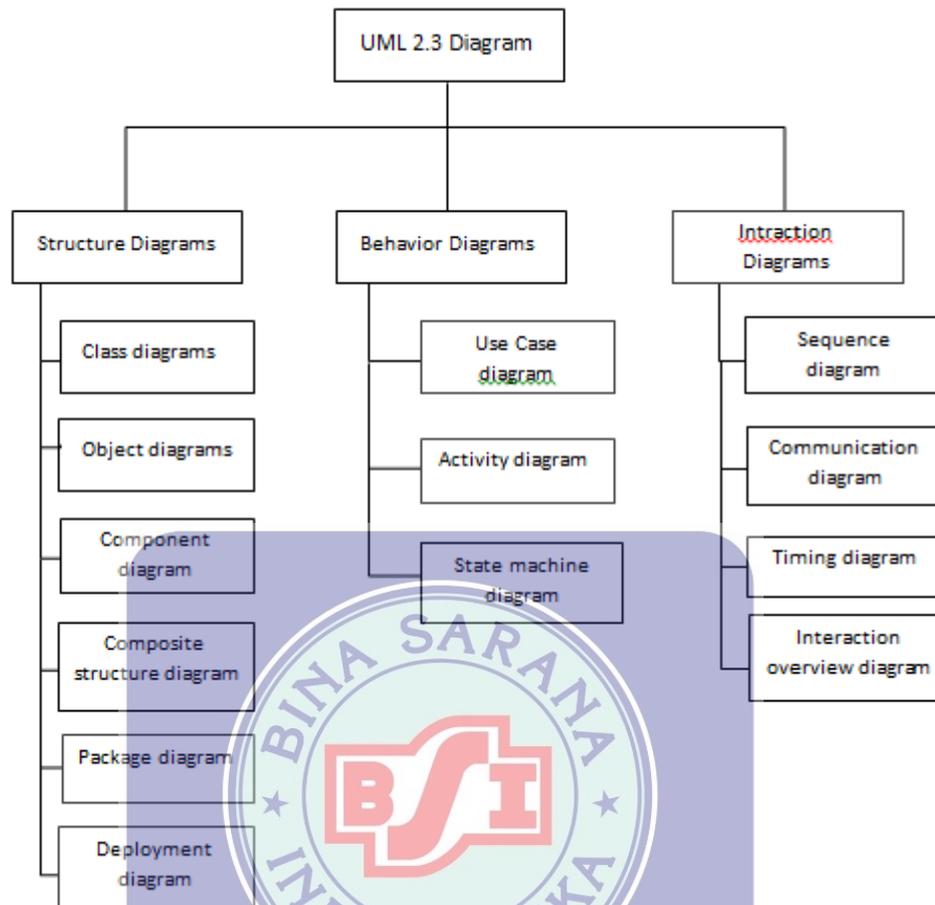


Sumber: Kuryanti J Sandra (2017:89)

Gambar II.1.  
Contoh LRS

### 2.2.3 UML (Unified Modeling Language)

Menurut Rosa A.S dan M.Shalahuddin (2016:133) UML (*unified Modeling Language*) adalah “Salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek”.



Sumber : Sukamto dan Shalahuddin (2016:140)

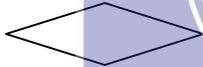
Gambar II.2.  
Diagram UML

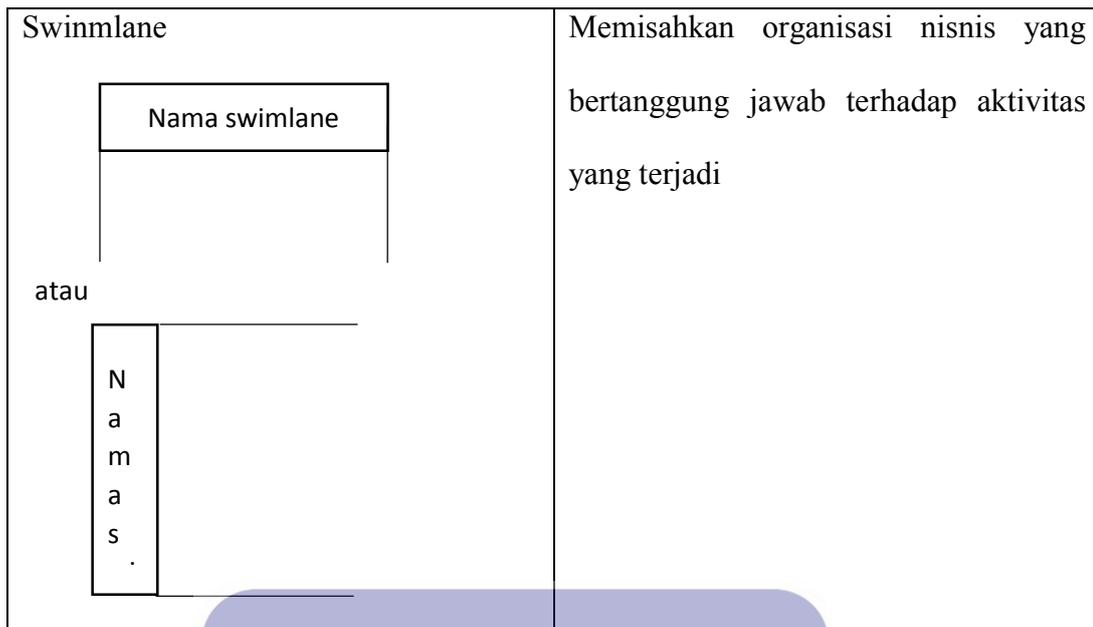
## 1. Activity Diagram

“Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Menurut (Rosa A.S dan M.Shalahudin 2016:161).

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

**Tabel II.2**  
**Activity diagram**

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari aktivitas digabungkan menjadi satu
Status akhir 	Status berakhir yang dilakukan sistem, sebuah diagram aktivitas memiliki status akhir



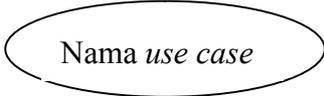
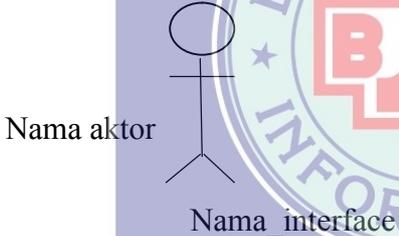
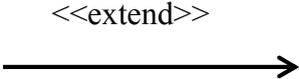
Sumber : Sukamto dan Shalahuddin (2013:162)

## 2. Use Case Diagram

“*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”. Menurut Rosa A.S dan M.Shalahuddin (2016:155)

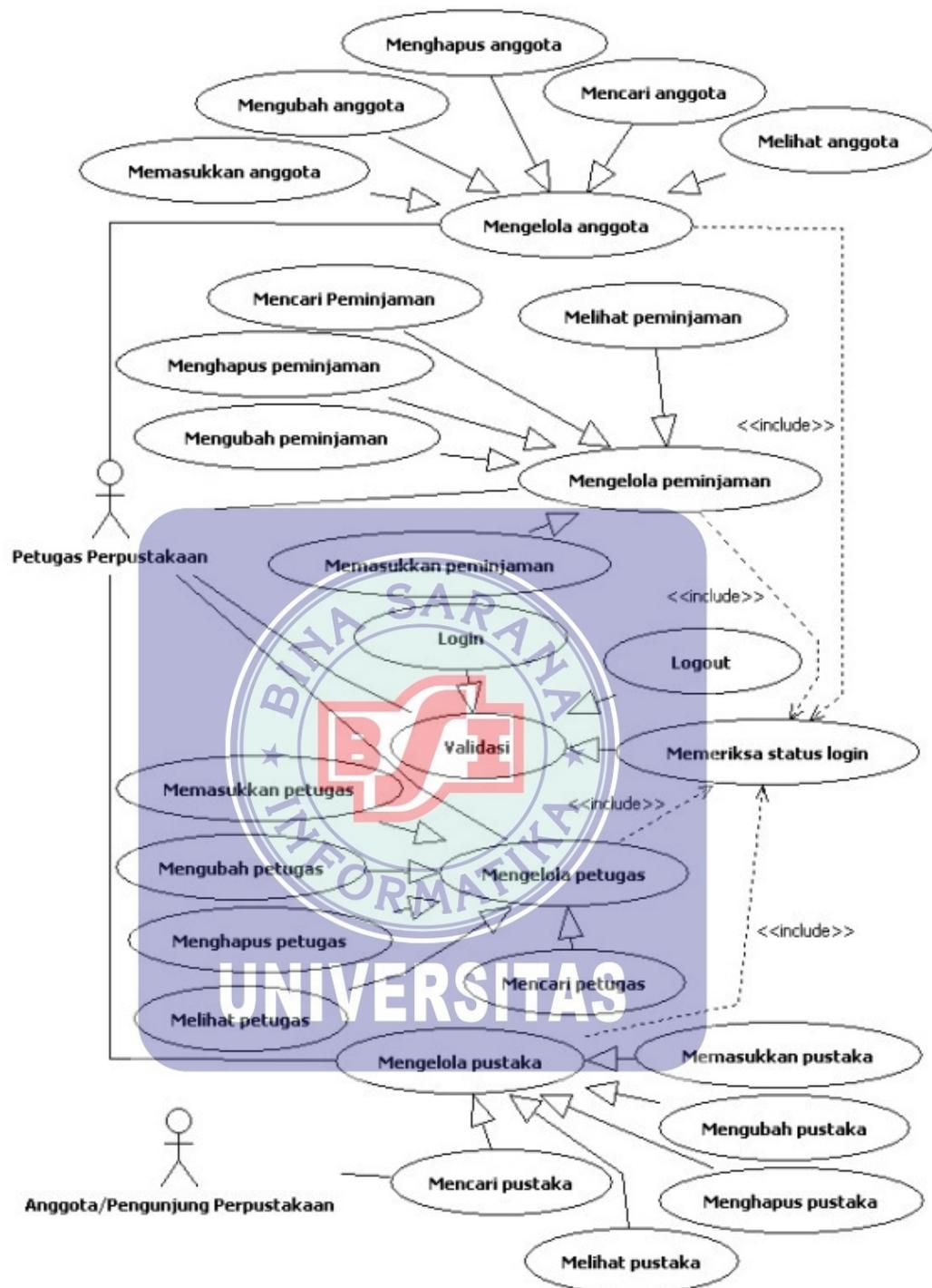
Berikut ini adalah simbol-simbol yang ada pada diagram *use case* :

**Tabel II.3.**  
*Use case diagram*

Simbol	Deskripsi
	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i></p>
<p>Aktor /actor</p> 	<p>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; biasanya dinyatakan dengan menggunakan kata benda diawal frase nama aktor</p>
<p>Asosiasi/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>
<p>Ekstensi/<i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <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 pemogram berorientasi objek;</p>

	<p>biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, missal arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya</p>
<p>Generalisasi/ <i>generalization</i></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: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p>&lt;&lt;include&gt;&gt;</p>  <p>&lt;&lt;uses&gt;&gt;</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> :</p>

Sumber: Sukamto dan Shalahuddin (2016:156)



Sumber : Sukamto dan Shalahuddin (2016:205)

**Gambar II.3**  
Contoh Use Case

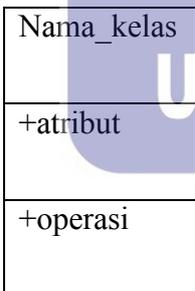
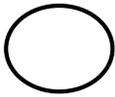
### 3. Class Diagram

Menurut Rossa A.S dan M.shalahuddin (2016:141) “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 metode atau oprasi”. Berikut penjelasan tentang atribut dan metode:

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Oprasi atau metode adalah fuungsi-fungsi yang dimiliki oleh suatu kelas

Berikut adalah simbol-simbol yang ada pada diagram kelas:

**Tabel II.4.**  
*Class diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Antarmuka/ <i>interface</i>  Nama_interface	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.

<p>Asosiasi / <i>association</i></p> 	<p>Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Asosiasi berarah / <i>directed association</i></p> 	<p>Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Generalisasi</p> 	<p>Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).</p>
<p>Kebergantungan / <i>dependency</i></p> 	<p>Relasi antarkelas dengan makna kebergantungan antarkelas.</p>
<p>Agregasi / <i>aggregation</i></p> 	<p>Relasi antarkelas dengan makna semua bagian (<i>whole-part</i>).</p>

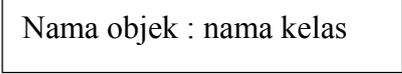
Sumber : Sukamto dan Shalahuddin (2016:146)

#### 4. Sequence Diagram

Menurut Rosa A.S dan M.Shalahuddin (2016:165) “Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek”.

Berikut adalah simbol-simbol yang ada pada diagram sekuen:

**Tabel II.5.**  
***Sequence diagram***

Simbol	Deskripsi
<p>Aktor</p>  <p>atau</p> <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol darin aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p>	<p>Menyatakan objek dalam keadaan aktif</p>

	dan berinteraksi, semua yang berhubungan dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
---	--

Sumber : Sukanto dan Shalahuddin (2016:165)

