

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Konsep Dasar Sistem**

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. (Asbon Hendra:2012:157)

- a. Menurut Jerry FithGerald, sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan dan berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.
- b. Menurut Ludwig Von Bartalanfy, sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan.
- c. Menurut Anatol Raporot, sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lain.

Menurut L. Ackof, sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya. (Asbon Hendra:2012:157-158)

### II.1.1. Karakteristik Sistem

#### a. Komponen (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapa pun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut *supra sistem*.

#### b. Batas sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (*Scope*) dari sistem tersebut.

#### c. Lingkungan luar sistem (*Environment*)

*Environment* merupakan segala sesuatu diluar batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus

dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan atau dikendalikan agar tidak mengganggu operasi sistem.

d. Penghubung sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya yang mengalir dari subsistem yang satu ke subsistem lainnya. Dengan kata lain, *output* dari suatu subsistem akan menjadi *input* dari subsistem lainnya.

e. Masukan sistem (*Input*)

Merupakan energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) adalah energi yang dimasukkan supaya sistem itu dapat beroperasi. Masukan sinyal ( *Signal input* ) adalah energi yang diproses untuk didapatkan keluaran.

f. Keluaran sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh sistem, meliputi *output* yang berguna, contohnya informasi yang dikeluarkan oleh komputer. Dan output yang tidak dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer.

g. Pengolahan sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

h. Tujuan Sistem (*Goal*)

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi *input* yang dibutuhkan dan *output* yang dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya. Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. (Asbon Hendra:2012:158-160)

## II.2. Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) adalah merupakan sistem informasi yang berbasis ruang, Sistem Informasi Geografis ini dibangun oleh dua jenis data yang saling berintegrasi (data spasial dan data atribut) sehingga dapat memberikan informasi terhadap satu ruang/lokasi secara utuh. Data tersebut terstruktur dalam satu sistem basis data dan dijalankan dalam satu perangkat lunak tertentu (*Software GIS*). (Masril Syukur;2012;17)

Sistem komputer untuk SIG terdiri dari perangkat keras (*hardware*), perangkat lunak (*software*) dan prosedur untuk penyusunan input data, pengolahan, analisis, pemodelan (*modeling*), dan penanyangan data geospasial adalah peta digital, foto udara, citra satelit, table statistic dan dokumen lain yang berhubungan. Data geospasial dibedakan menjadi data grafis ( atau disebut juga data geometris dan data atribut (data tematik). Data grafis mempunyai tiga

elemen: titik (*node*), garis (*arc*) dan luasan (*polygon*) dalam bentuk vector ataupun raster yang mewakili geometri topologi, ukuran, bentuk, posisi dan arah. Fungsi *user* adalah untuk memilih informasi yang diperlukan, membuat standar, membuat jadwal pemutakhiran (*updating*) yang efisien menganalisa hasil yang dikeluarkan untuk kegunaan yang diinginkan dan merencanakan aplikasi. SIG memudahkan dalam melihat fenomena kebumihan dengan perspektif yang lebih baik. (Danny Manongga,dkk ; 2009 ;3)

### **II.2.1 Basisdata SIG**

Basisdata merupakan kumpulan berbagai data spesifik yang dapat dikontrol dalam satu sistem. Basisdata SIG terdiri dari lapisan-lapisan data (*layer-layer*) yang disusun sedemikian rupa dalam software SIG berdasarkan kepada penggunaannya (*aplikasi*). (Masril Syukur;2012;17)

Masing-masing lapisan data tersebut mempunyai tema tersendiri yang terdiri dari pasangan data spasial (*geografisi*) dan non spasial (*atribut*). Data spasial merupakan peta digital yang dibangun berdasarkan jenis grafik (*titik, garis dan area*), sedangkan data non spasial merupakan tabel-tabel yang berintegrasi dengan data spasialnya. Pembangunan basisdata dalam SIG merupakan pekerjaan yang paling besar dan utama (70 % dari seluruh pembangunan SIG), karena tanpa basisdata yang lengkap dan benar maka SIG tidak ada apa-apanya.

Basisdata SIG terdiridari berbagai lapisan data yang menggambarkan semua satuan objek atau struktur ruang dalam kawasan. Lapisan-lapisan data

tersebut (*layer*) disusun dan dipisah-pisahkan berdasarkan jenis grafis (garis, poligon dan area) dan tema-tema dari objek. (Masril Syukur;2012;17-18)

## II.2.2 Sumber Data SIG

Sumber data SIG secara garis besarnya terdiri dari 2 bagian, yaitu sumber data spasial dan sumber data atribut.

### 1. Sumber Data Spasial

Sumber data spasial adalah merupakan sumber data keruangan atau berbentuk grafis (*feature*) dari permukaan bumi (geografis) yang biasanya berupa peta-peta. Berbagai sumber data spasial yang dapat digunakan sebagai basis data SIG antara lain adalah :

- a. Peta garis berbentuk kertas (hardcopy)
- b. Peta garis dalam format digital (biasanya format Cad)
- c. Peta Foto (raster/ image)
- d. Peta *Remote Sensing*

### 2. Sumber Data Atribut

Sumber data atribut merupakan sumber data yang terdiri dari keterangan-keterangan atau diskripsi yang berkaitan dengan objek atau aspek keruangan. Berbagai data atribut yang dijadikan sebagai sumber data adalah:

- a. Tabel-tabel tentang tema objek spasial

b. Hasil survey lapangan

c. Deskripsi wilayah

d. dlsb.

Dalam pengolahannya, sumber data atribut tersebut akan diintegrasikan dengan basisdata spasialnya menjadi basisdata SIG. (Masril Syukur;2012;18)

### **II.2.3 Software dan Hardware SIG**

Secara umum pengetahuan orang terhadap SIG identik dengan perangkat lunak (software) SIG yang beredar dipasaran. Untuk menganalisis ruang, diperlukan perangkat lunak SIG yang memiliki analitical power tertentu. Semakin kompleks daya analisis satu perangkat lunak, berarti semakin tinggi analitical power-nya dan semakin mahal harga perangkat lunak tersebut.

Berbagai software SIG yang beredar di Indonesia antara lain ; *ArcInfo*, *MapInfo*, *Erdas*, *ILWIS*, *Idrisi*, *Integrapph*, dlsb., dimana masing-masing perangkat lunak tersebut mempunyai kelebihan dan kekurangannya. Namun perangkat lunak yang paling banyak digunakan saat ini adalah *ArcInfo* dan *MapInfo*, dimana kedua software ini (terutama *ArcInfo*) memiliki analitical power yang lengkap dan mampu berintegrasi dengan data lain seperti multi media. (Masril Syukur;2012;18)

Untuk membangun satu aplikasi SIG yang lengkap dibutuhkan software-software lain yang sifatnya mendukung dari segi pembangunan

basisdata, analisis data dan penyajian informasi. Perangkat lunak (software) tersebut antara lain adalah :

- a. Visual Basic
- b. Microsoft SQL/ Acces/ Oracle
- c. Map Object/ Map X
- d. ER-Mapper (Masril Syukur;2012;18)

### **II.3. Sarana Bantu Navigasi Pelayaran ( SBNP )**

Sarana Bantu Navigasi Pelayaran adalah peralatan atau sistem yang berada diluar kapal yang didesain dan dioperasikan untuk meningkatkan keselamatan dan efisiensi bernavigasi kapal dan/atau lalu lintas kapal. (Nomor PM 25 ; 2011 ; 3)

Rambu suar adalah Sarana Bantu Navigasi-Pelayaran tetap yang bersuar dan mempunyai jarak tampak sama atau lebih dari 10 (sepuluh) mil laut yang dapat membantu para navigator adanya bahaya/rintangan navigasi antara lain karang, air dangkal, gosong, dan bahaya terpencil serta menentukan posisi dan/atau haluan kapal serta dapat dipergunakan sebagai tanda batas wilayah negara.

Pelampung suar adalah Sarana Bantu Navigasi-Pelayaran apung yang bersuar dan mempunyai jarak tampak sama atau lebih 4 (empat) mil laut yang dapat membantu para navigator adanya bahaya/rintangan navigasi antara lain karang, air dangkal, gosong, kerangka kapal dan/atau untuk menunjukkan perairan

aman serta pemisah alur, dan dapat dipergunakan sebagai tanda batas wilayah negara. (Nomor PM 25 ; 2011 ; 3)

#### **II.4. PHP *Hypertext Preprocessor***

PHP ( PHP : *Hypertext Preprocessor* ) adalah bahasa *script* yang ditanam di sisi *server* . Prosesor PHP dijalankan di *server* ( Windows atau Linux ). Saat sebuah halaman dibuka dan mengandung kode PHP, prosesor itu akan menerjemahkan dan mengeksekusi semua perintah dalam halaman tersebut, dan kemudian menampilkan hasilnya ke browser sebagai halaman HTML biasa. Karena penerjemahan ini terjadi di server, sebuah halaman ditulis dengan PHP dapat dilihat dengan menggunakan semua jenis browser, di sistem operasi apapun.

Seperti sebagian besar bahasa *script* lainnya, PHP dapat ditanamkan langsung ke HTML. Kode PHP dipisahkan dari HTML dengan menggunakan tanda *start* dan *End* . Ketika sebuah dokumen dibaca, prosesor PHP hanya menerjemahkan area yang ditandai saja, dan menampilkan hasilnya pada tempat yang sama. (Adhi Prasetio:2012:122 ).

##### **II.4.1. Mengenal Script Dasar PHP**

Dalam penulisannya. Script php tidak harus berdiri sendiri namun dapat disisipkan diantara kode HTML. Script PHP harus selalu diawali dengan `<? Atau <?php` dan diakhiri dengan `?>`.

Perhatikan contoh penulisan script PHP berikut:

`<?`

.....

?>

Atau

<?php

.....

?>

Sedangkan untuk menampilkannya ke dalam browser digunakan fungsi **echo “data”;**.

<?php

Echo”Dreamweaver CS5”;

\$nilai=20

Echo”\$nilai”;

?>

Semua teks yang diketik setelah tanda buka script(<?) dan tanda tutup script (?>) akan dieksekusi sebagai suatu script PHP. Anda dapat membuat keterangan atau komentar didalam script PHP dan komentar tersebut tidak akan dieksekusi sebagai sebuah script.( MADCOMS:2010:350-351 )

## **II.5. Macromedia Dreamweaver CS5**

Dreamweaver merupakan software aplikasi yang digunakan sebagai HTML editor professional untuk mendesain web secara visual. Aplikasi ini juga yang biasa dikenal dengan istilah WYSIWYG (*What You See Is What You Get*). Yang intinya adalah bahwa anda tidak harus berurusan dengan tag-tag HTML untuk membuat sebuah situs. Selain itu, Dreamweaver juga memberikan keleluasaan kepada anda untuk menggunakannya sebagai media penulisan bahasa pemrograman web. ( MADCOMS:2010:1 )

### **II.5.1 Mengenal Alur Kerja Dreamweaver**

Berikut ini adalah alur kerja pembuatan sebuah website dengan menggunakan Dreamweaver.

1. Membuat rancangan dan mencari lokasi situs
2. Melakukan organisasi dan menejemen file situs
3. Membuat layout halaman web
4. Membuat halam dengan kode program
5. Mengatur web menjadi aplikasi dengan isi yang dinamis
6. Menciptakan halaman dinamis
7. Menguji situs dan melakukan publis. ( MADCOMS:2010:9 )

## II.6. MySQL

### II.6.1. Dasar Pemrograman MySQL

Sebuah website yang dinamis membutuhkan tempat penyimpanan data agar dapat memberikan komentar, saran dan masukan atas website yang dibuat. Tempat penyimpanan data berupa informasi dalam sebuah table tersebut dengan database adalah MySQL yang memiliki sekumpulan prosedur dan struktur sedemikian rupa sehingga mempermudah dalam menyimpan, mengatur dan menampilkan data. ( Anhar:2010:45)

### II.6.2. Pengertian MySQL

MySQL (*My Structure Query Language*) adalah salah satu *Database Management System* (DBMS) dari sekian banyak DBMS dari oracle, MS SQL, Postagre SQL, dan lainnya. MySQL berfungsi untuk mengolah database menggunakan bahasa SQL. MySQL bersifat open source sehingga kita bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung/*Support* dengan database MySQL. ( Anhar:2010:45)

### II.6.3. Database

Database adalah sekumpulan table-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom. Struktur file yang menyusun sebuah database adalah *Data Record* dan *Field*.

- a. Data adalah satuan informasi yang akan diolah. Sebelum diolah, data dikumpulkan didalam suatu file database.

- b. Record adalah data yang isinya merupakan satu kesatuan seperti NamaUser dan Password. Setiap keterangan yang mencakup NamaUser dan Password dinamakan satu record. Setiap record diberi nomor urut yang disebut nomor record.
- c. Field adalah sub bagian dari record. Dari contoh isi record diatas meka terdiri dari 2 field yaitu : field NamaUser dan Password. (Anhar:2010:45-46)

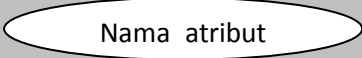
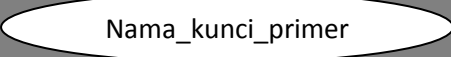
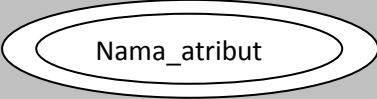
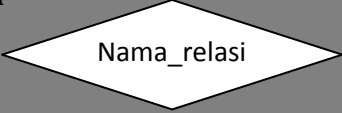
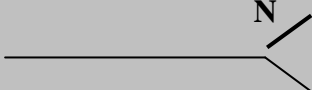
#### II.6.4 ERD ( *Entity Relationship Diagram* )

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD) . ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ( Rosa A.S M.Shalahuddin:2013:50 )

Berikut adalah simbol-simbol yang digunakan pada ERD dapat kita lihat pada Tabel II.1 dibawah ini:

**Tabel II.1 Simbol-simbol pada ERD**

Simbol	Deskripsi
<p><b>Entitas/entity</b></p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;">Nama_entitas</div>	<p>Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh</p>

	<p>aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.</p>
<p><b>Atribut</b></p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas</p>
<p><b>Atribut kunci primer</b></p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> 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)</p>
<p><b>Atribut multivalai / <i>multivalue</i></b></p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.</p>
<p><b>Relasi</b></p> 	<p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.</p>
<p><b>Asosiasi / <i>association</i></b></p> 	<p>Penghubung antara relasi dan entitas dimana dikedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.</p> <p>Kemungkinan jumlah maksimum</p>

	keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas.
--	-----------------------------------------------------------------------------------------

**Sumber : Rosa A.S M.Shalahuddin:2013:51**

## II.7. Unified Modelling Language (UML)

UML (*Unified Modelling Language*) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. 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 M.Shalahuddin:2013:137-138)

### II.7.1. Diagram Unified Modelling Language (UML)

#### 1. *Class Diagram*

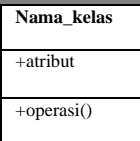
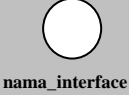


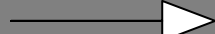


Diagram kelas atau *class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

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

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.( Rosa A.S M.Shalahuddin:2013:141-142)

Berikut ini adalah simbol-simbol yang ada pada diagram kelas dapat kita lihat pada Tabel II.2 dibawah ini :

**Tabel II.2 Simbol-simbol pada *Class Diagram***

Simbol	Deskripsi
<p><b>Kelas</b></p> 	Kelas pada struktur sistem.
<p><b>Antarmuka/<i>interface</i></b></p> 	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
<p><b>Asosiasi/<i>Association</i></b></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p><b>Asosiasi berarah/<i>directed Association</i></b></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p><b>Generalisasi</b></p> 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
<p><b>Kebergantungan/<i>dependency</i></b></p> 	Relasi antar kelas dengan makna kebergantungan antar kelas..
<p><b>Agregasi/<i>agregation</i></b></p> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )

Sumber : Rosa A.S M.Shalahuddin:2013: 146-147

## 2. Use Case Diagram


*Use Case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour*) 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. ( Rosa A.S M.Shalahuddin:2013:155)

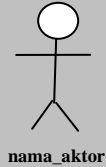

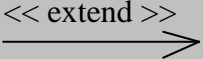
Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut actor dan *use case*.

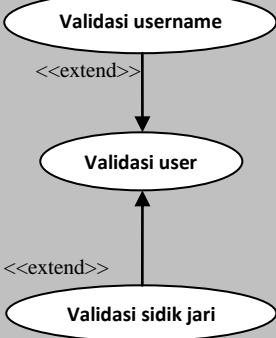

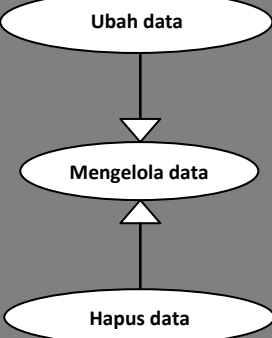
- a. Aktor, merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antra unit atau actor. ( Rosa A.S M.Shalahuddin:2013:155)

Berikut adalah symbol-simbol yang ada pada diagram *use case*, dapat kita lihat pada Tabel II.3 dibawah ini :

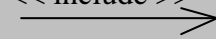
**Tabel II.3 Simbol-simbol pada *use case diagram***

Simbol	Deskripsi
<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar

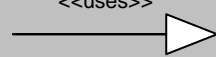
	<p>pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>
<p><b>Antarmuka/interface</b></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.</p>
<p><b>Asosiasi/Association</b></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><b>Asosiasi berarah/directed Association</b></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 pemograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p>

	 <pre> graph TD     A([Validasi username]) -- &lt;&lt;extend&gt;&gt; B([Validasi user])     C([Validasi sidik jari]) -- &lt;&lt;extend&gt;&gt; B   </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> yang menjadi induknya.</p>
<p><b>Generalisasi/ <i>generalization</i></b></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>  <pre> graph TD     A([Ubah data]) --&gt; B([Mengelola data])     C([Hapus data]) --&gt; B   </pre>
<p><b>Menggunakan <i>/include / uses</i></b></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan</p>

<< include >>



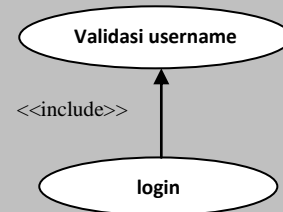
<< uses >>



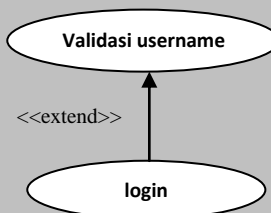
memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini

Ada dua sudut pandang yang cukup besar mengenai include di *use case* :

- *Include* berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan, misal pada kasus berikut:



- *Include* berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan, misal pada kasus berikut:



	Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.
--	--------------------------------------------------------------------------------------------------------------------------------




**Sumber : Rosa A.S M.Shalahuddin:2013: 156-158**



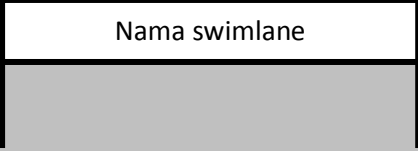
### 3. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan sistem. ( Rosa A.S M.Shalahuddin:2013:161 )

Berikut adalah simbol-simbol yang ada pada diagram aktivitas, dapat kita lihat pada Tabel II.4 dibawah ini :

**Tabel II.4 Simbol-simbol pada *activity diagram***

Simbol	Deskripsi
<b>Status awal</b> 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
<b>Aktivitas</b> 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
<b>Percabangan/decision</b> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu

<b>Penggabungan/join</b> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
<b>Status akhir</b> 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<b>Swimlane</b> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

**Sumber : Rosa A.S M.Shalahuddin:2013: 162**

#### 4. *Sequence Diagram*

Diagram sekuan menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambar diagram sekuan maka harus diketahui objek-objek yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuan 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 pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. (Rosa A.S M.Shalahuddin:2013: 165)

## II.8. Kamus Data

Kamus data (*Data Dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi:

- a. Nama – nama dari data
- b. Digunakan pada – merupakan proses-proses yang terkait data
- c. Deskripsi – merupakan deskripsi data

Informasi tambahan – seperti tipe data, nilai data, batas nilai data dan komponen yang membentuk data. (Rosa A.S M.Shalahuddin:2013: 73-74)

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan, dapat kita lihat pada Tabel II.5 dibawah ini :

**Tabel II.5 Simbol-simbol pada kamus data**

Simbol	Deskripsi
=	Disusun atau terdiri dari
+	Dan
[]	Baik....atau...
{ <sup>n</sup> }	n kali diulang / bernilai banyak
()	Data opsional
*...*	Batas komentar

Sumber : Rosa A.S M.Shalahuddin:2013: 74