

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Sistem adalah unsure yang bekerja sama secara berkelompok dan berhubungan erat satu sama lain sehingga dapat bekerja sama dalam menjalankan fungsi untuk mencapai tujuan (Kadek Indah Ratnaningsih ; 2014 : 5).

##### **II.1.1. Sistem Informasi**

Informasi didefinisikan sebagai data yang diolah dan dapat digunakan sebagai alat pembuatan keputusan. Informasi dikatakan sebagai data yang diolah menjadi sesuatu yang berguna bagi para penerimanya atau dapat berupa sesuatu yang berguna dan dapat dipahami sebagai dasar pengambilan keputusan (Kadek Indah Ratnaningsih ; 2014 : 6).

##### **II.1.2. Akuntansi**

Medefinisikan akuntansi sebagai proses pencatatan, pebgklasifikasian, penjurnalan, pengikhtisaran, dan penganalisisan data keuangan suatu organisasi. Menurut American Intitute of Cerified Public Accounting (AICPA) akuntansi adalah suatu proses mencatat, mengaklasifikasi, meringkas, mengolah dan menyajikan data transaksi serta kejadian yang berhubungan dengan keuangan sehingga mudah dimengerti untuk pengambilan suatu keputusan (Kadek Indah Ratnaningsih ; 2014 : 6).

### **II.1.3. Sistem Informasi Akuntansi**

Medefinisikan system informasi akuntansi sebagai komponen sumber daya, yang digunakan untuk mengolah data menjadi informasi yang berguna. Sistem informasi akuntansi merupakan gabungan dari manusia dan sumber daya lainnya yang bertanggungjawab dalam menyediakan informasi keuangan serta informasi yang diperoleh melalui pengumpulan dan pengolahan data transaksi dalam suatu organisasi (Kadek Indah Ratnaningsih ; 2014 : 6).

### **II.2. Sistem Informasi Penjualan**

Dalam jurnalnya mariani (2010) menjelaskan penerapan teknologi informasi akan bermanfaat jika penerapannya sesuai dengan tujuan, visi dan misi perusahaan dengan menetapkan strategi bisnis dan strategi sistem teknologi informasi. Dengan adanya sistem informasi penjualan, perusahaan dapat memberikan pelayanan yang lebih baik bagi pelanggannya, dan diharapkan loyalitas pelanggan dapat meningkat dan penjualan perusahaan juga meningkat (Budi Sasongko ; 2012 : 4).

### **II.3. Dreamweaver**

Dreamweaver merupakan software aplikasi yang digunakan sebagai editor HTML 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 yang cukup rumit untuk membuat sebuah halaman web. Selain itu, Dreamweaver juga memberikan

keleluasaan kepada anda untuk menggunakannya sebagai media penulisan bahasa pemrograman web (Madcoms Madiun ; 2011 : 2).

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

PHP merupakan singkatan dari “ *Hypertext Preprocessor*”. Pada awalnya PHP merupakan kependekan dari Personal Home Page (situs personal) dan PHP itu sendiri pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995, dan pada saat PHP masih bernama FI (Form Interpreter), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum. PHP adalah sebuah bahasa scripting yang terpasang pada HTML. Sebagian besar sintaksnya mirip dengan bahasa pemrograman C, Java, Asp dan Perl, ditambah beberapa fungsi PHP yang spesifik dan mudah dimengerti.

PHP digunakan untuk membuat tampilan web menjadi lebih dinamis, dengan PHP Anda bisa menampilkan atau menjalankan beberapa file dalam 1 file dengan cara di- include atau require. PHP itu sendiri sudah dapat berinteraksi dengan beberapa database walaupun dengan kelengkapan yang berbeda, yaitu seperti: DBM, FilePro (Personic, Inc), Informix, ingres, interbase, Microsoft Access, MSSQL, MYSQL, Oracle, PostgrSQL, dan Sybase.

Dari uraian di atas maka dapat diambil 4 point utama tentang PHP:

1. PHP adalah singkatan dari : Hypertext Proprocessor
2. PHP adalah bahasa scripting server – side, artinya di jalankan di server, kemudian outputnya dikirim ke client (browser)

3. PHP digunakan untuk membuat aplikasi web
4. PHP mendukung banyak database (MYSQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, dll).

Jadi Anda tidak perlu susah – susah menampilkan postingan anda dengan cara lama, yaitu dengan cara link ke file lain atau mengganti file, jadi tidak repot – repot membuat file baru. Anda dapat membuat 1 file, jadi tetapi dalam 1 file itu bisa menampilkan banyak data, yaitu dengan bantuan database (Madcoms Madiun ; 2011 : 228).

## **II.5. MySQL**

Database MySQL dapat dibuat dengan menggunakan tampilan jendela phpMyAdmin atau menggunakan sebuah script PHP. Penulis menyarankan untuk membuat dan menyiapkan database dengan menggunakan jendela phpMyAdmin karena akan lebih mudah langkah – langkah penggunaannya.

Penyimpanan data yang fleksibel dan cepat aksesnya sangat dibutuhkan dalam sebuah website yang interaktif dan dinamis. Database sendiri berfungsi sebagai penampung data yang anda inputkan melalui form website. Selain itu dapat juga dibalik dengan menampilkan data yang tersimpan dalam database kedalam website di internet sebagai bank data adalah MySQL. MySQL menggunakan SQL dan bersifat gratis, selain itu MySQL dapat berjalan di berbagai platform, antara lain linux, windows dan sebagainya.

Sebagai contoh, jika Anda memberikan fasilitas kepada pengunjung website untuk mengisi sebuah form buku tamu, maka Anda dapat menampung data dalam buku

tamu tersebut dalam sebuah database. Contoh lain adalah jika Anda ingin menampilkan artikel yang dinamis dan mudah untuk di – update, Anda dapat menyimpan artikel tersebut dalam sebuah database dan menampilkan dalam halaman web (Madcoms Madiun ; 2011 : 260).

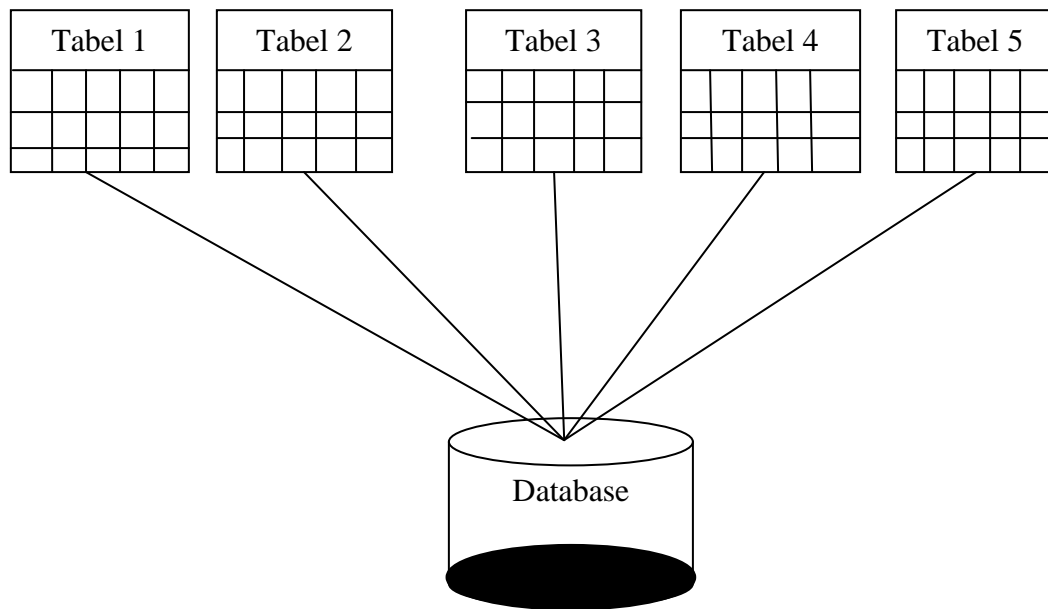
## II.6. Database

*Database* atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer. Data itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. *Database* sering digunakan untuk melakukan proses terhadap data – data tersebut untuk menghasilkan informasi tertentu. Misalnya dari data nama siswa dan tanggal lahir siswa Anda bisa mendapatkan informasi nama siswa yang berulang tahun pada hari ini. Tentu saja informasi tersebut akan didapatkan dari software pemroses database dengan cara Anda memberikan perintah dalam bahasa tertentu yaitu SQL (*Structured Query Language*) (Wahana Komputer ; 2010 : 24).

### 1. Tabel

Data dalam *database* akan diklasifikasikan berdasarkan jenisnya dan disimpan di dalam wadah tersendiri, yang disebut table. Oleh karena itu, banyak juga yang mendefinisikan *database* sebagai suatu kumpulan table. Tabel itu sendiri adalah suatu entitas yang tersusun atas kolom dan baris. Dalam dunia *database*, kolom disebut *field* dan baris disebut *record*.

Berikut adalah gambar relasi dalam table :

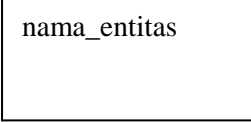
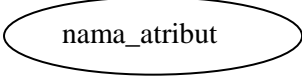
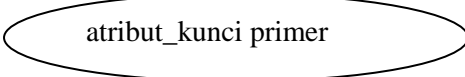
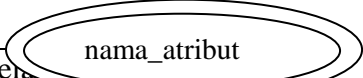
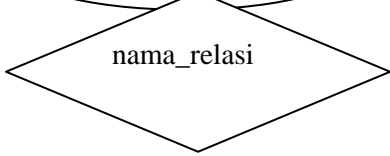


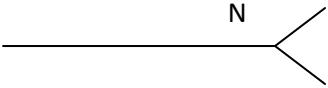
**Sumber : Budi Raharjo (2011: 4)**

## II.7. ERD

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 menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi *Crow's Foot*, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen. Berikut adalah symbol – symbol yang digunakan pada ERD dengan notasi Chen (Rosa A.S & M. Shalahudin ; 2013 : 50)

Berikut adalah simbol – simbol *Entity Relantionship Diagram* :

Simbol	Deskripsi
<p data-bbox="300 387 467 421">Entitas/<i>Entity</i></p> 	<p data-bbox="774 387 1302 640">Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar diakses oleh aplikasi komputer. Penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.</p>
<p data-bbox="300 678 395 712">Atribut</p> 	<p data-bbox="774 678 1302 745"><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.</p>
<p data-bbox="300 969 571 1003">Atribut kunci primer</p> 	<p data-bbox="774 969 1302 1249"><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, misalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik.</p>
<p data-bbox="300 1261 643 1294">Atribut multinilai/<i>multivalue</i></p> 	<p data-bbox="774 1261 1302 1361"><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.</p>
<p data-bbox="300 1373 363 1406">Relasi</p> 	<p data-bbox="774 1373 1302 1473">Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.</p>

<p>Asosiasi/<i>association</i></p> 	<p>Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i>. Kemungkinan jumlah maksimum keterhubungan antara entitas yang lain tersebut disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dengan entitas B</p>
--	---

Sumber : Rossa A.S & M. Shalahudin (2013 : 50 – 51)

## II.8. Normalisasi Data

Setelah melalui tahapan di atas, maka hasil pada diagram tersebut mulai direalisasikan pada table – table database. Untuk itu diperlukan sebuah tahapan yang disebut normalisasi. Normalisasi data adalah proses di mana tabel – table pada database dites dalam hal kesalingtergantungan di antara field – field pada sebuah table. Misalnya jika pada sebuah table tersebut, maka table tersebut harus dipecah menjadi banyak table. Banyaknya table pecahannya bergantung pada seberapa banyak ketergantungannya. Tiap table hanya boleh memiliki sebuah field kunci yang menjadi ketergantungan dari field lainnya dalam table tersebut.

Pada proses normalisasi data, aturan yang dijadikan acuan adalah metode ketergantungan fungsional. Teorinya adalah bahwa tiap kolom dalam sebuah tabel selalu memiliki hubungan yang unik dengan sebuah kolom kunci.

Ada beberapa langkah dalam normalisasi tabel, yaitu:

1. **Decomposition**, dekomposisi adalah proses mengubah bentuk tabel syarat tertentu sebagai tabel yang baik. Dekomposisi dapat dikatakan berhasil

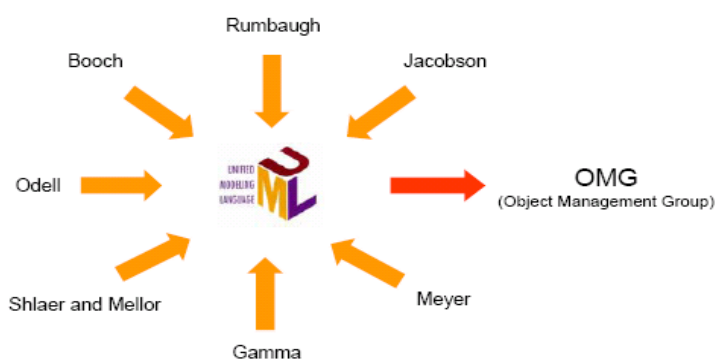
jika tabel yang dikenai dekomposisi bila digabungkan kembali dapat menjadi tabel awal sebelum di-dekomposisi. Dekomposisi akan sering dilakukan dalam proses normalisasi untuk memenuhi syarat – syaratnya.

2. **Bentuk tidak normal**, pada bentuk ini semua data yang ada pada tiap entity (diambil atributnya) masih ditampung dalam satu tabel besar. Data yang ada pada tabel ini masih ada yang redundansi dan ada juga yang kosong (Wahana Komputer ; 2010 : 32).
3. **Normal form pertama (1 Normal Form)**, pada tahapan ini tabel di-dekomposisi dari tabel bentuk tidak normal yang kemudian dipisahkan menjadi tabel – tabel kecil yang memiliki kriteria tidak memiliki atribut yang bernilai ganda dan komposit. Semua atribut harus bersifat atomik (Wahana Komputer ; 2010 : 33).
4. **Normal Form kedua (2 Normal Form)**, pada tahapan ini tabel dianggap memenuhi normal kedua jika pada tabel tersebut semua atribut yang bukan kunci primer tersebut. Dalam hal ini semua tabel diatas telah memenuhi bentuk normal form yang kedua, Adapun kunci primer masing – masing tabel adalah sebagai berikut:
  - Untuk tabel anggota kunci primernya kode\_agt.
  - Untuk tabel buku kunci primernya kd\_bk.
  - Untuk tabel transaksi kunci primernya no\_trans (Wahana komputer ; 2010 : 34).
5. **Normal Form ketiga (3 Normal Form)**, setiap atribut pada tabel selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama.

Bentuk normal ketiga biasanya penggunaan bentuk normal (normalisasi) hanya sampai pada bentuk ketiga, dan tabel yang dihasilkan telah memiliki kualitas untuk membentuk sebuah database yang dapat diandalkan. Semua tabel di atas juga telah memenuhi bentuk normal tahap ketiga (Wahana Komputer ; 2010 : 35).

## II.9. Unified Modeling Language (UML)

*Unified Modeling Language (UML)* adalah sebuah “bahasa” yang telah menjadi standar dalam industry untuk visual, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Yuni Sugiarti ; 2013 : 34)



**Gambar. II.8. Metodeologi Pemodelan Berorientasi Objek**

**Sumber : Yuni Sugiarti (2013 : 35)**

*Unified Modeling Language (UML)* biasa digunakan untuk:

1. Menggambarkan batasan sistem dan fungsi – fungsi sistem secara umum, dibuat dengan use case dan actor.

2. Menggambarkan kegiatan dan proses bisnie yang dilaksanakan secara umum, dibuat dengan interaction diagrams.
3. Menggambarkan representasi struktur static sebuah sistem dalam bentuk class diagrams.
4. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan state transition diagrams.
5. Menyatakan arsitektur implementasi fisik menggunakan component and development diagrams.
6. Menyampaikan atau memperluas functionality dengan stereotypes (Yuni Sugiarti ; 2013 : 36).

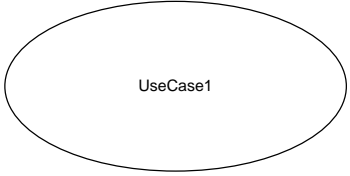
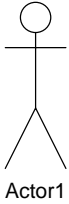
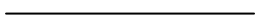
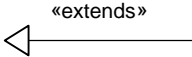
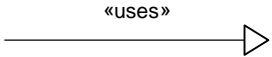
#### II.8.1. Diagram Dasar *Unified Modeling Language* (UML)

Berikut ini adalah penjelasan mengenai berbagai diagram UML serta tujuannya:

##### 1. *Use Case Diagram*

Kebutuhan fungsional akan digambarkan melalui sebuah diagram yang dinamakan diagram *use case*. *Use Case Diagram* atau diagram use case merupakan pemodelan untuk menggambarkan kelakuan (behavior) sistem yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat (Yuni Sugiarti ; 2013 : 41).

Berikut adalah simbol – simbol yang ada pada diagram *Use Case*

Simbol	Deskripsi
<p data-bbox="300 383 424 412">Use Case</p> 	<p data-bbox="683 383 1339 524">Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja awal frase awal <i>use case</i></p>
<p data-bbox="300 674 379 703">Aktor</p> 	<p data-bbox="683 674 1339 925">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 awal frase nama aktor.</p>
<p data-bbox="300 1010 564 1039">Asosiasi/association</p> 	<p data-bbox="683 1010 1339 1115">Komunikasi antar dua aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p data-bbox="300 1155 395 1184">Extend</p> 	<p data-bbox="683 1155 1339 1440">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 pemrograman berorientasi objek; biasanya <i>use case</i> memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, arah panah menunjuk pada <i>use case</i> yang dituju.</p>
<p data-bbox="300 1581 400 1610">Include</p> 	<p data-bbox="683 1581 1339 1722">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.</p>

Sumber : Yuni Sugiarti (2013: 42)

## 2. *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.

- Atribut merupakan variable – variable yang dimiliki oleh suatu kelas
- Atribut mendeskripsikan property dengan sebaris teks didalam kotak tersebut.
- Operasi atau metode adalah fungsi – fungsi yang dimiliki oleh suatu kelas.

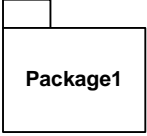
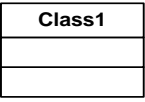

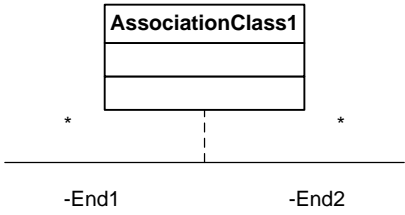
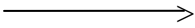
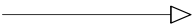
Diagram kelas mendiskripsikan jenis - jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan – batasan yang terdapat dalam hubungan – hubungan objek tersebut.

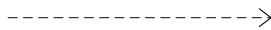
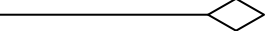
Diagram kelas menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain – lain.

Kelas memiliki tiga area pokok

1. Nama
2. Atribut
3. Operasi (Yuni Sugiarti ; 2013 :57).

Berikut adalah simbol – simbol dari *Class Diagram*

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkusan dari satu atau lebih kelas
Operasi 	Kelas pada struktur sistem
Antarmuka/ <i>Interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
Asosiasi berarah/ <i>directed</i> asosiasi 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh makna kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi
Kebergantungan/Depedency	Relasi antar kelas dengan makna kebergantungan antar kelas

	
Agregasi 	Relasi antar kelas dengan makna semua bagian


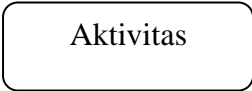
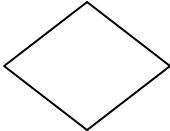


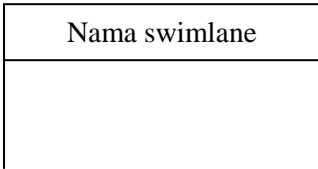
**Sumber : Yuni Sugiarti (2013 : 59)**

### 3. *Activity Diagram*

Diagram aktivitas atau activity diagram menggambarkan *workflow* ( aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada 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.

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- Urutan atau pengelompokkan tampilan dari sistem / user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- Rancangan menu yang ditampilkan pada perangkat lunak (Rosa A.S & M.Shalahudin ; 2013 : 161)

Berikut adalah simbol – simbol yang ada pada diagram aktivi





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 satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : Rossa A.S & M. Shalahuddin (2013 : 162)

#### 4. *State Machine Diagram*

*State machine diagram* atau statechart diagram atau dalam bahasa Indonesia disebut diagram mesin status atau sering juga disebut diagram status digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek. Jika diagram sekuen digunakan untuk interaksi antar objek maka diagram status digunakan untuk interaksi di dalam sebuah objek. Perubahan tersebut digunakan untuk berinteraksi di dalam sebuah objek. Perubahan tersebut digambarkan dalam suatu graf berarah. *State machine diagram* merupakan pengembangan dari diagram *Finite State Automata* dengan penambahan beberapa fitur dan konsep baru. Diagram *Finite State Automata* (FSA) ini biasanya diajarkan dalam mata kuliah Automata (Rosa A.S & M. Shalahudin ; 2013 : 163).

Berikut adalah simbol – simbol yang ada pada diagram sekuen :


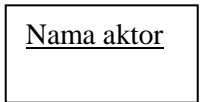

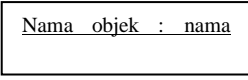
NO	GAMBAR	NAMA	KETERANGAN
1		Start / Status Awal ( <i>Initial State</i> )	Start atau initial state adalah state atau kehidupan awal pada sistem mulai hidup
2		End / Status Akhir ( <i>Final State</i> )	End atau final state adalah state keadaan akhir dari daur hidup suatu sistem
3	Event 	Event	Event adalah kegiatan yang menyebabkan berubahnya status mesin
4		State	State atau status adalah keadaan sistem pada waktu tertentu. State dapat berubah jika ada event tertentu yang memicu perubahan tersebut.

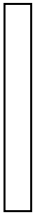
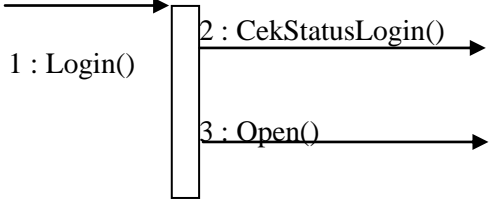
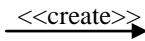
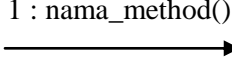
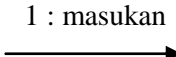
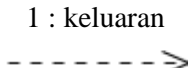
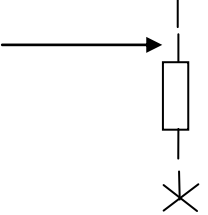
Sumber : Rossa A.S & M. Shalahuddin (2013 : 164– 16

### 5. Sequence Diagram

Diagram sekuen 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 sekuen maka harus diketahuin 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 (Rosa A.S & M. Shalahudin ; 2013 : 165).

Berikut adalah simbol – simbol yang ada pada diagram sekuen:

NO	GAMBAR	NAMA	KETERANGAN
1	<p>Atau </p> <p></p> <p>Tanpa waktu aktif</p>	<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2		Garis Hidup / <i>lifeline</i>	Menyatakan kehidupan suatu objek
3		Objek	Menyatakan objek yang berinteraksi pesan

4		Waktu Aktif	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya</p>  <p>Maka cekStatusLogin() dan open() dilakukan di dalam metode login () Aktor tidak memiliki waktu aktif</p>
5		Pesan Tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6		Pesan tipe call	Menyatakan suatu objek membuat objek memanggil operasi/ metode yang ada pada objek yang dibuat
7		Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8		Pesan tipe return	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 destroy	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

Sumber : Rossa A.S & M. Shalahuddin (2013 : 165– 167)

