

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Aplikasi**

Program aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

- a. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu (Rahmatillah ; 2011 : 3).

#### **II.2. Jasa**

Jasa adalah sesuatu yang diproduksi dan dikonsumsi secara simultan. Jadi, jasa tidak pernah ada dan hasilnya dapat dilihat setelah terjadi. Misal : bila anda potong rambut, jasa dikonsumsi ketika diproduksi, tetapi hasil jasa tampak dan akan berakhir beberapa waktu. Kesenjangan produksi dan konsumsi merupakan perbedaan yang penting. Jasa tidak dapat diproduksi di satu tempat dan dikirim ke tempat lain seperti barang, juga tidak dapat disimpan. Semua

karakteristik ini dapat dihubungkan dengan keserentakan produksi dan konsumsi, jasa terdiri dari tindakan dan interaksi yang merupakan kontak sosial. Jasa lebih dari sekedar hasil sesuatu yang tak terhalang, dan jasa merupakan interaksi sosial antara produsen dan konsumen (Hery Prasetya ; 2012 : 72).

### **II.3. Penyewaan**

Penyewaan dapat diartikan sebagai peminjaman jasa atau barang dengan tidak mengabaikan suatu ketentuan atau kesepakatan dan syarat yang berlaku didalam organisasi tersebut guna mencapai satu tujuan. Barang yang dapat disewa bermacam-macam, tarif dan lama sewa juga bermacam-macam. Rumah umumnya disewa dalam satuan tahun, mobil dan gedung dalam satuan hari, permainan komputer seperti *playstation* disewa dalam satuan jam, buku yang disewa dalam satuan hari, dan kontainer yang disewa dalam satuan muatan (Era Sari Munthe ; 2011 : 90).

### **II.4. Metode Cash Basis**

Akuntansi Kas Basis adalah basis akuntansi yang mengakui pengaruh transaksi dan peristiwa lainnya pada saat kas atau setara kas diterima atau dibayar yang digunakan untuk pengakuan pendapatan, belanja dan pembiayaan. Kas Basis akan mencatat kegiatan keuangan saat kas atau uang telah diterima misalkan perusahaan menjual produknya akan tetapi uang pembayaran belum diterima maka pencatatan pendapatan penjualan produk tersebut tidak dilakukan, jika kas telah diterima maka transaksi tersebut baru akan dicatat seperti halnya dengan

“dasar akrual” hal ini berlaku untuk semua transaksi yang dilakukan, kedua teknik tersebut akan sangat berpengaruh terhadap laporan keuangan, jika menggunakan dasar akrual maka penjualan produk perusahaan yang dilakukan secara kredit akan menambah piutang dagang sehingga berpengaruh pada besarnya piutang dagang sebaliknya jika yang di pakai cash basis maka piutang dagang akan dilaporkan lebih rendah dari yang sebenarnya terjadi. Kas Basis juga mendasarkan konsepnya pada dua pilar yaitu :

1) Pengakuan Pendapatan :

Pengakuan pendapatan, saat pengakuan pendapatan pada cash basis adalah pada saat perusahaan menerima pembayaran secara kas. Dalam konsep *cash basis* menjadi hal yang kurang penting mengenai kapan munculnya hak untuk menagih. Makanya dalam *cash basis* kemudian muncul adanya metode penghapusan piutang secara langsung dan tidak mengenal adanya estimasi piutang tak tertagih.

2) Pengakuan Biaya :

Pengakuan biaya, pengakuan biaya dilakukan pada saat sudah dilakukan pembayaran secara kas. Sehingga dengan kata lain, pada saat sudah diterima pembayaran maka biaya sudah diakui pada saat itu juga. Untuk usaha-usaha tertentu masih lebih menggunakan *cash basis* ketimbang *accrual basis*, contoh : usaha relative kecil seperti toko, warung, mall (*retail*) dan praktek kaum spesialis seperti dokter, pedagang *informal*, panti pijat (malah ada yang pakai credit card-tapi ingat credit card dikategorikan juga sebagai kas basis).

Disamping itu, pencatatan akuntansi dengan metode kas basis juga mempunyai beberapa keunggulan dan kelemahan yaitu sebagai berikut :

1) Keunggulan Pencatatan Akuntansi Secara Kas Basis

- a) Metode kas basis digunakan untuk pencatatan pengakuan pendapatan, belanja dan pembiayaan.
- b) Beban/biaya belum diakui sampai adanya pembayaran secara kas walaupun beban telah terjadi, sehingga tidak menyebabkan pengurangan dalam penghitungan pendapatan.
- c) Pendapatan diakui pada saat diterimanya kas, sehingga benar-benar mencerminkan posisi yang sebenarnya.
- d) Penerimaan kas biasanya diakui sebagai pendapatan.
- e) Laporan Keuangan yang disajikan memperlihatkan posisi keuangan yang ada pada saat laporan tersebut.
- f) Tidak perlunya suatu perusahaan untuk membuat pencadangan untuk kas yang belum tertagih.

2) Kelemahan Pencatatan Akuntansi Secara Kas Basis

- a) Metode kas basis tidak mencerminkan besarnya kas yang tersedia.
- b) Akan dapat menurunkan perhitungan pendapatan bank, karena adanya pengakuan pendapatan sampai diterimanya uang kas.
- c) Adanya penghapusan piutang secara langsung dan tidak mengenal adanya estimasi piutang tak tertagih.
- d) Biasanya dipakai oleh perusahaan yang usahanya relative kecil seperti toko, warung, mall (retail) dan praktek kaum spesialis seperti dokter,

pedagang *informal*, panti pijat (malah ada yang pakai credit card-tapi ingat credit card dikategorikan juga sebagai kas basis).

- e) Setiap pengeluaran kas diakui sebagai beban.
- f) Sulit dalam melakukan transaksi yang tertunda pembayarannya, karena pencatatan diakui pada saat kas masuk atau keluar.
- g) Sulit bagi manajemen untuk menentukan suatu kebijakan kedepannya karena selalu berpatokan kepada kas (H. Rahmansyah Ritonga ; 2012 : 2).

## II.5. *Database*

*Database* adalah suatu kumpulan data yang terintegrasi yang diorganisasikan untuk memenuhi kebutuhan para pemakai didalam suatu organisasi. semua pengaksesan *kedatabase* ditangani oleh suatu piranti lunak yang disebut *database management system*. *Database* merupakan himpunan kelompok kata yang saling berkaitan. Data tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi yang tidak perlu, sehingga dapat diolah atau dieksplorasi secara cepat dan mudah untuk menghasilkan *informasi*.

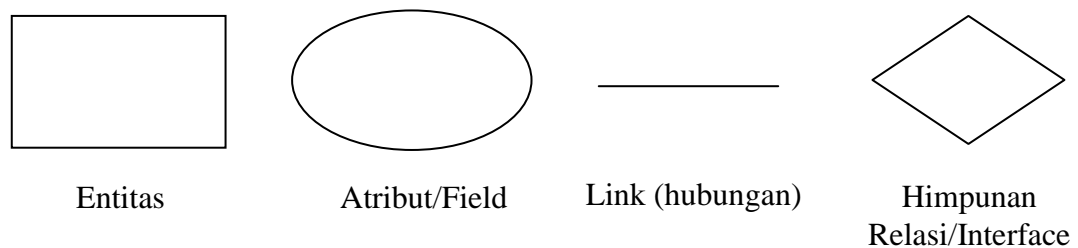
Secara fisik suatu *database* terdiri atas entity-entity yang biasanya disebut table. Tiap-tiap table tersebut mempunyai atribut-atribut yang disebut field. Dan isi dari table tersebut disebut tuple atau record yang merupakan tumpukan baris yang mempunyai minimal satu atribut dari table tersebut.

*Database* rasional adalah salah satu *database* yang sering digunakan dan tersedia untuk aplikasi komputer yang penting. *Database* relasional menyediakan

sebuah kemampuan yang sangat kuat dan fleksibel untuk menyimpan data dari berbagai jenis (Luther ; 2011 : 132).

## II.6. *Entity Relationship Diagram (ERD)*

*Entity Relationship Diagram* atau ERD adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antarentitas. Proses memungkinkan analisis menghasilkan struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien (Janner Simarmata ; 2010 : 67).



**Gambar. II.1 Bentuk Simbol ERD**  
(Sumber : Janner Simarmata ; 2010 : 67)

## II.7. Kamus Data

Kamus data adalah suatu ensiklopedik dari *informasi* yang berkaitan dengan data perusahaan, atau dapat juga kita katakan bahwa kamus data adalah katalog atau *directory* yang berbasis komputer (*computer base catalog or directory*) yang berbasis data perubahan (metadata). Yang berkenaan dengan tahapan penjelasan data ini adalah sistem kamus data (*data description language/DDL*). Sistem kamus data berbentuk perangkat lunak yang fungsinya adalah penciptaan dan pemeliharaan serta menyediakan kamus data agar dapat

digunakan. Kamus data dapat berbentuk kertas ataupun arsip (*file*) komputer (Ian Sommerville ; 2010 : 344).

## **II.8. Teknik Normalisasi**

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan :

### **II.8.1. Bentuk-bentuk Normalisasi**

#### a. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti *format* tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

#### b. Bentuk normal tahap pertama (1<sup>st</sup> Normal *Form*)

Definisi :

Sebuah table disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut.
- Masing-masing cell bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci,

meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

c. Bentuk normal tahap kedua (2<sup>nd</sup> normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada *primary key* secara utuh.

d. Bentuk normal tahap ketiga (3<sup>rd</sup> normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi  $X \rightarrow A$ , dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- X haruslah *superkey* pada tabel tersebut.
- Atau A merupakan bagian dari *primary key* pada tabel tersebut.

e. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

f. Boyce Code Normal Form (BCNF)

- Memenuhi 1<sup>st</sup> NF
- Relasi harus bergantung fungsi pada atribut *superkey* (Janner Simarmata ; 2010 : 76).

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

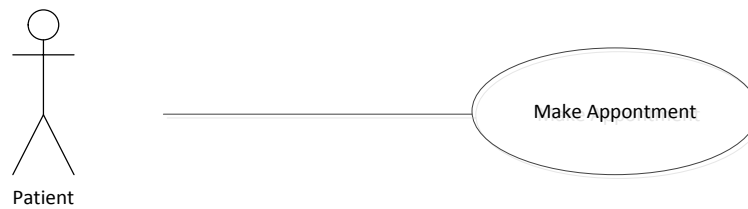
Menurut Arif Rachman (2012) Pemecahan masalah utama dari Object Oriented biasanya dengan penggambaran dalam bentuk model. Model abstrak (semu) merupakan gambaran detail dari inti masalah yang ada, umumnya sama seperti refleksi dari problem yang ada pada kenyataan. Beberapa modeling tool yang dipakai adalah bagian dari dasar UML, kependekan dari *United Modeling Language*.

UML terdiri atas beberapa diagram, yaitu :

### 1. Diagram Use Case

Diagram Use Case menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. yang menjadi persoalan itu *apa yang dilakukan* bukan *bagaimana melakukannya*. Diagram Use Case dekat kaitannya dengan kejadian-kejadian. Kejadian (scenario) merupakan contoh apa yang terjadi ketika seseorang berinteraksi dengan sistem. untuk lebih memperjelas lihat gambaran suatu peristiwa untuk sebuah klinik kesehatan di bawah ini :

“Pasien menghubungi klinik untuk membuat janji (*appointment*) dalam pemeriksaan tahunan. *Receptionist* mendapatkan waktu yang luang pada buku jadwal dan memasukkan janji tersebut ke dalam waktu luang itu.”



**Gambar II.2 Usecase Diagram**  
**Sumber : Arif Rachman ; 2012 : 2**

## 2. Diagram Class

Diagram *Class* memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. Diagram *Class* bersifat statis; *menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan.*

*Multiplicity* dari suatu titik *association* adalah angka kemungkinan bagian dari hubungan kelas dengan *single instance* (bagian) pada titik yang lain. *Multiplicity* berupa *single number* (angka tunggal) atau *range number* (angka batasan). Pada contoh, hanya bisa satu 'Customer' untuk setiap 'Order', tapi satu 'Customer' hanya bisa memiliki beberapa 'Order'. Tabel di bawah mengenai *multiplicity* yang sering digunakan :

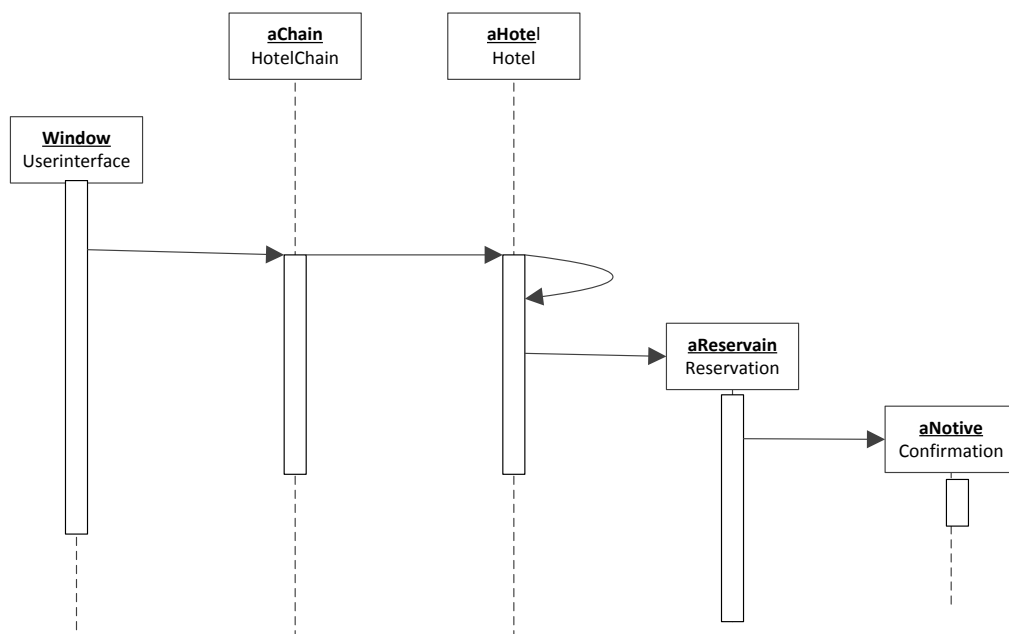
**Tabel II.1**  
Tabel *Multiplicity*.

<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

**Sumber : Arif Rachman ; 2012 : 3**

### 3. Diagram Sequence

Diagram *Class* dan diagram Object merupakan suatu gambaran *model statis*. Namun ada juga yang bersifat *dinamis*, seperti Diagram Interaction. Diagram *Sequence* merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

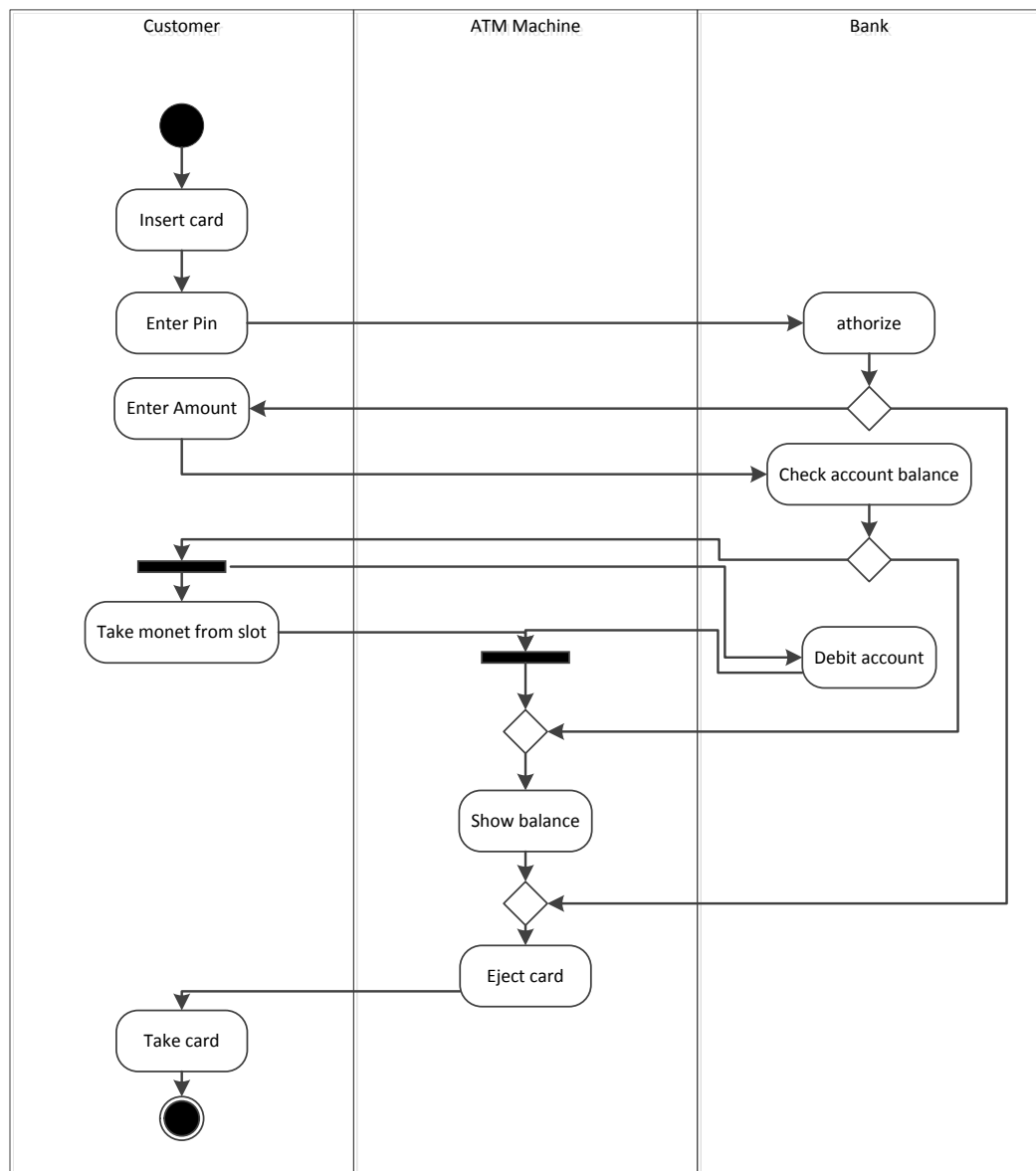


**Gambar II.3. Diagram Sequence**  
Sumber : Arif Rachman ; 2012 : 6

### 4. Diagram Activity

Pada dasarnya diagram *Activity* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram Statechart. Diagram Statechart berfokus pada *obyek*

yang dalam suatu proses (atau proses menjadi suatu obyek), diagram Activity berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain.



**Gambar II.4. Diagram Activity**  
**Sumber : Arif Rachman ; 2012 : 9**