

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Sistem merupakan serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu. Suatu sistem pasti tersusun dari sub-sub sistem yang lebih kecil yang juga saling tergantung dan bekerja sama untuk mencapai tujuan. Sebagai contoh, sistem administrasi universitas terdiri dari sub-sub sistem administrasi fakultas dan sub-sistem fakultas terdiri dari sub-sub sistem administrasi jurusan.

Tujuan dasar suatu sistem tergantung pada jenis sistem itu sendiri. Sebagai contoh, sistem peredaran darah manusia merupakan sistem biologi yang memiliki tujuan untuk mengedarkan darah yang mengandung oksigen dan sari makanan ke seluruh tubuh. Sedangkan sistem buatan manusia seperti sistem yang terdapat di sekolah, organisasi bisnis, atau instansi pemerintah juga mempunyai tujuan yang berbeda. Organisasi bisnis biasanya memiliki tujuan yang lebih jelas, seperti yang telah disebutkan pada bagian sebelumnya, yaitu mendapatkan laba.

Sistem informasi yang kadang kala disebut sebagai sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen, baik manual ataupun berbasis komputer yang terintegrasi untuk mengumpulkan, menyimpan dan mengelola data serta menyediakan informasi kepada pihak yang berkepentingan sebagai pemakai informasi tersebut (Anastasia Diana ; 2011 : 3).

## **II.2. Informasi**

Informasi adalah data yang berguna yang telah diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat. Informasi sangat penting bagi organisasi. Pada dasarnya informasi adalah penting seperti sumber daya yang lain, misalnya peralatan, bahan, tenaga, dsb.

Informasi yang berkualitas dapat mendukung keunggulan kompetitif suatu organisasi. Dalam sistem informasi akuntansi, kualitas dari informasi yang disediakan merupakan hal penting dalam kesuksesan sistem.

Secara konseptual seluruh sistem organisasional mencapai tujuannya melalui proses alokasi sumber daya, yang diwujudkan melalui proses pengambilan keputusan manajerial. Informasi memiliki nilai ekonomik pada saat ia mendukung keputusan alokasi sumber daya, sehingga dengan demikian mendukung sistem untuk mencapai tujuan.

Pemakai informasi akuntansi dapat dibagi dalam dua kelompok besar: ekstern dan intern. Pemakai ekstern mencakup pemegang saham, investor, kreditor, pemerintah, pelanggan, pemasok, pesaing, serikat pekerja, dan masyarakat. Pemakai intern terutama para manajer, kebutuhannya bervariasi tergantung pada tingkatannya (Agustinus ; 2012 : 1).

## **II.3. Akuntansi**

Akuntansi merupakan bahasa bisnis. Sebagai bahasa bisnis akuntansi menyediakan cara untuk menyajikan dan meringkas kejadian-kejadian bisnis dalam bentuk informasi keuangan kepada pemakainya. Informasi akuntansi merupakan bagian terpenting dari seluruh informasi yang diperlukan oleh

manajemen. Informasi akuntansi yang dihasilkan oleh suatu sistem dibedakan menjadi dua, yaitu informasi akuntansi keuangan dan informasi akuntansi manajemen.

Pemakai informasi akuntansi pun terdiri dari dua kelompok, yaitu pemakai eksternal dan pemakai internal. Yang dimaksud dengan pemakai eksternal mencakup pemegang saham, investor, kreditor, pemerintah, pelanggan, pemasok, pesaing, serikat kerja dan masyarakat. Sedangkan pemakai internal adalah pihak manajer dari berbagai tingkatan dalam organisasi bersangkutan (Kusrini ; 2012 : 1).

#### **II.4. Sistem Informasi Akuntansi**

Sistem informasi akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan saksi keuangan. Lingkup sistem informasi akuntansi dapat dijelaskan dari manfaat yang didapat dari informasi akuntansi. Manfaat atau tujuan sistem informasi akuntansi tersebut adalah sebagai berikut :

1. Mengamankan harta / kekayaan perusahaan. Harta / kekayaan di sini meliputi kas perusahaan, persediaan barang dagangan, termasuk aset tetap perusahaan.
2. Menghasilkan beragam informasi untuk pengambilan keputusan. misal, pengelola toko swalayan memerlukan informasi mengenai barang apa saja yang diminati oleh konsumen. Membeli barang yang kurang laku berarti kas akan terjebak dalam persediaan dan berarti kehilangan kesempatan untuk membeli barang dagangan yang laku.

3. Menghasilkan informasi untuk pihak eksternal. Setiap pengelola usaha memiliki kewajiban untuk membayar pajak. Besarnya pajak yang dibayar tergantung pada omset penjualan (jika pengelola memilih menggunakan norma dalam perhitungan pajaknya) atau tergantung pada laba rugi usaha (jika pengelola memilih untuk tidak menggunakan norma dalam perhitungan pajaknya).
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi. Sistem informasi dapat juga dimanfaatkan untuk penilaian kinerja karyawan atau divisi.
5. Menyediakan data masa lalu untuk kepentingan audit (pemeriksaan). Data yang tersimpan dengan baik sangat memudahkan proses audit (pemeriksaan).
6. Menghasilkan informasi untuk penyusunan dan evaluasi anggaran perusahaan. Anggaran merupakan alat yang sering digunakan perusahaan untuk mengendalikan pengeluaran kas.
7. Menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian. Selain berguna untuk membandingkan informasi yang berkaitan dengan anggaran dan biaya standar dengan kenyataan seperti yang telah dikemukakan (Anastasia Diana ; 2011 : 6).

## **II.5. Penjualan**

Penjualan merupakan tujuan utama dilakukannya kegiatan perusahaan. Perusahaan dalam menghasilkan barang/jasa, mempunyai tujuan akhir yaitu menjual barang/jasa tersebut kepada masyarakat. Oleh karena itu, penjualan memegang peranan penting bagi perusahaan agar produk yang dihasilkan oleh perusahaan dapat terjual dan memberikan penghasilan bagi perusahaan. Penjualan yang dilakukan perusahaan bertujuan untuk menjual barang/jasa yang diperlukan sebagai sumber pendapatan untuk menutup semua ongkos guna memperoleh laba.

Penjualan adalah pemindahan hak milik barang atau pemberian jasa yang dilakukan penjualan kepada pembeli dengan harga yang disepakati bersama dengan jumlah yang dibebankan kepada pelanggan dalam penjualan barang/jasa dalam suatu periode akuntansi (Freddy Rangkuti ; 2010 : 206).

## **II.6. Metode Cash Basis**

Akrual Basis merupakan salah satu konsep yang sangat penting dalam akuntansi, dimana Pencatatan basis kas adalah teknik pencatatan ketika transaksi terjadi dimana uang benar-benar diterima atau dikeluarkan. Dengan kata lain 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.7. *Client Server***

Aplikasi *database client/server* terdiri dari dua bagian, yaitu *server* dan *client*. *Server*, yang sering disebut sebagai *back-end*, akan berhubungan dengan media penyimpanan data/*storage device*. Tugasnya adalah mengatur agar semua data dapat tersimpan dengan benar pada media penyimpanannya. *Server* menyediakan fungsi-fungsi untuk melakukan proses *inquiry* maupun manipulasi data. Berhubung *server* ini berhubungan sangat erat dengan data, maka banyak orang menyebutnya sebagai *database server*. Oleh karena fungsi-fungsi pada *server* sangat rumit, sangat disarankan agar kita membeli dari *vendor-vendor* berpengalaman, jadi kita tidak perlu membuat sendiri. Bagian lain dari aplikasi *database client/server* adalah *client*. *Client*, yang sering disebut sebagai *front-end*, berinteraksi dengan para pemakai aplikasi. Fungsi dari *client* ini sangatlah bervariasi, perlu disesuaikan dengan keperluan para pemakainya. Dalam melaksanakan fungsinya, *client* akan membuat hubungan ke *database server* dan memanggil fungsi-fungsi yang tersedia pada *server*.

Sebuah *database server* dapat dihubungkan dengan lebih dari satu *client*. Bisa jadi, *client-client* tersebut memanggil fungsi-fungsi yang sama dalam satu waktu. *Server* mengatur bagaimana cara melaksanakan fungsi-fungsi itu sehingga data yang di proses dapat selalu dalam keadaan benar (David Ciang ; 2011 : 24).

### **II.7.1. Konsep *Client Server***

Konsep *client server* yang berbasis pada aturan bahwa komputer *server* hanya akan mengirim data yang dibutuhkan oleh *workstation/client*, di mana proses penyiapan data dilakukan pada komputer *server*. Proses tersebut sedikit banyak dapat mengurangi beberapa permasalahan, baik dari segi lalu lintas data maupun sumber daya dan biaya komputerisasi, karena kini sebuah perusahaan dapat menggunakan komputer berkemampuan rendah sebagai *workstation* dan memberi alokasi dana lebih besar untuk memperoleh komputer *server* dengan kemampuan lebih baik. Disamping itu kemampuan data lebih terjamin, salah satu contoh paling populer dari konsep *client server* adalah sistem jaringan internet, di mana dengan menggunakan komputer sederhana atau lewat ponsel, kita dapat mengakses data pada jaringan komputer lainnya (Harip Santoso ; 2011 : 2).

### **II.7.2. Arsitektur *Client Server***

Berdasarkan pada cara PC *Client* dihubungkan ke komputer *server*, dikenal dua macam tingkatan arsitektur yaitu model Dua Tier dan Model Tiga Tier. Dua Tier/Dua Tingkatan adalah proses di mana komputer *workstation* membuat hubungan ke komputer *server* dan mempertahankan hubungan tersebut sampai proses selesai. Misalnya pada proses pemasukan data atau membuat

sebuah laporan. Hubungan yang terbentuk sangat konsumtif dalam sumber daya (alokasi memori, ruang *harddisk*, kontrol, dan lainnya), karenanya tidak efektif jika dilakukan untuk hubungan yang melibatkan banyak pemakai (contohnya pada sistem jaringan internet). Oleh karena itu tidak heran jika konsep Dua Tingkatan hanya pengguna digunakan model Tiga Tier.

Tiga Tier/Tiga Tingkatan adalah model proses di mana kita menambahkan sebuah komputer yang bertugas untuk berhubungan dengan komputer *server* (SQL *Server*) sehingga *workstation/client* tidak dapat berhubungan langsung dengan komputer *server* (SQL *Server*). Contoh paling populer dari model Tiga Tier adalah sistem jaringan internet, di mana semua permintaan *client* akan ditangani oleh komputer web *server* dan dari web *server* hubungan ke SQL *server* dilakukan. Model ini cukup efektif, ia dapat membantu meningkatkan keamanan data karena begitu permintaan data ke SQL *Server* selesai dilakukan (informasi/tabel data di copy-kan ke komputer web *server*), hubungan ke SQL *server* diputus (Harip Santoso ; 2011 : 3).

## **II.8. Pengertian Java**

Bahasa pemrograman Java merupakan karya SunMicrosystemInc. Rilis resmi level *beta* dilakukan pada November 1995. Dua bulan berikutnya Netscape menjadi perusahaan pertama yang memperoleh lisensi bahasa Java dari Sun. Java telah berkembang dari semula ditujukan untuk pemrograman *applet web browser* menjadi bahasa pemrograman pengembangan aneka ragam aplikasi, mulai dari yang berjalan di *handheld devices* seperti *handphone*, PDA (*Personal Digital Assitant*) sampai aplikasi terbesar skala *enterprise* diberagam komputer *server*.

Java merupakan bahasa orientasi objek untuk pengembangan aplikasi cerdas yang dapat berkomunikasi lewat *internet*/ jaringan komunikasi.

Java adalah bahasa yang dapat dijalankan disembarang *platform*, diberagam lingkungan : *internet*, *consumer electronic products*, dan *computer applications*. *The java 2 Platform* tersedia dalam tiga edisi untuk keperluan berbeda berikut :

1. Java 2 Standard Edition (J2SE)
2. Java 2 Enterprise Edition (J2EE)
3. Java 2 Micro Edition (J2ME).

Pada pengembangan *enterpriseapplications*, kita menggunakan sejumlah besar paket. Pada *consumerelectronicproduct*, hanya sejumlah kecil bagian bahasa yang digunakan. Masing-masing edisi berisi *Java 2 Software Development KIT* (SDK) untuk mengembangkan aplikasi dan *Java 2 Runtime Environment* (JRE) untuk menjalankan aplikasi.

a. *Standard Edition*(J2SE)

*The Java 2 Platform, Standard Edition*(J2SE) menyediakan lingkungan pengembangan yang kaya fitur, stabil, aman dan *cross-platform*. Edisi ini mendukung konektivitas basis data, rancangan antarmuka pemakai, masukan/keluaran dan pemrograman jaringan dan termasuk sebagai paket-paket dasar bahasa Java.

b. *EnterpriseEdition*(J2EE)

*The Java 2, EnterpriseEdition*(J2EE) menyediakan kakas untuk membangun dan menjalankan *multitierenterpriseapplications*. J2EE berisi paket-paket di

J2SE ditambah paket-paket untuk mendukung pengembangan *EnterpriseJavaBeans*, *Java Servlets*, *JavaServer*, *Pages*, *XML*, dan kendali transaksi yang fleksibel.

c. *Micro Edition*(J2ME)

*The Java 2, Micro Edition*(J2ME) untuk beragam *consumerelectronicproduct*, seperti *pager*, *smartcard*, *cellphone*, *handheldPDA*, dan *set-topbox*. J2ME sembari menyediakan bahasa Java yang sama, unggul dalam portabilitas kemampuan dijalankan di mana pun dan *safenetworkdelivery* seperti J2SE dan J2EE (Bambang Hariyanto ; 2011 : 3).

## II.9. Pengertian NetBeans

NetBeans merupakan salah satu proyek *open source* yang disponsori oleh *Sun Microsystem*. Proyek ini berdiri pada tahun 2000 dan telah menghasilkan 2 produk, yaitu NetBeanss IDE dan NetBeans Platform. NetBeans IDE merupakan produk yang digunakan untuk melakukan pemrograman baik menulis kode, meng-*compile*, mencari kesalahan dan mendistribusikan program. Sedangkan NetBeans Platform adalah sebuah modul yang merupakan kerangka awal / pondasi dalam bangun aplikasi desktop yang besar.

NetBeans juga menyediakan paket yang lengkap dalam pemrograman dari pemrograman standar (aplikasi desktop), pemrograman *enterprise*, dan pemrograman perangkat mobile. Saat ini NetBeans telah mencapai versi 6.8 (Wahana Komputer ; 2010 : 15).

## II.10. Pengertian Database

*Database* merupakan kumpulan data yang saling berhubungan, hubungan antar data dapat ditunjukkan dengan adanya *field* kunci dari setiap tabel yang beda.

Dalam satu *file* atau tabel terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* terdiri dari *field* yang saling berhubungan menunjukkan bahwa *fiel* tersebut satu pengertian yang lengkap dan disimpan dalam satu *record*. Basis data mempunyai beberapa kriteria penting yaitu :

1. Bersifat data oriented dan bukan program oriented.
2. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya.
3. Dapat dikembangkan dengan mudah, baik *volume* maupun strukturnya.
4. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah.
5. Dapat digunakan dengan cara-cara yang berbeda.

Prinsip utama *database* adalah pengaturan data dengan tujuan utama fleksibel dan kecepatan pada saat pengambilan data kembali. Adapun ciri-ciri basis data di antaranya adalah sebagai berikut :

1. Efisiensi meliputi kecepatan, ukuran dan ketepatan.
2. Data dalam jumlah besar.
3. Berbagi pakai (dipakai bersama-sama atau *sharebility*).

Mengurangi bahkan menghilangkan terjadinya duplikasi dan data yang tidak konsisten (Windu Gata ; 2013 : 19).

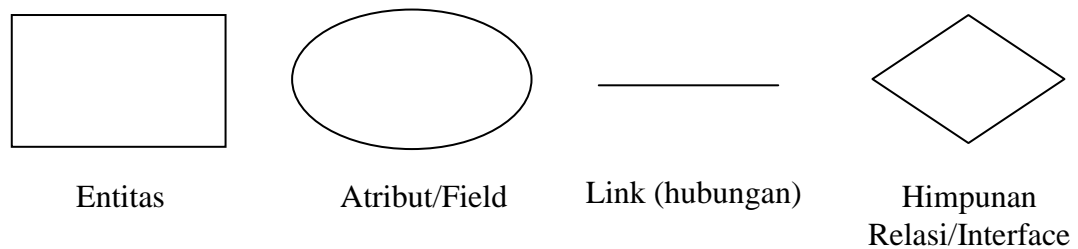
### II.11. Pengertian MySQL

Mysql *database server* adalah RDBMS (*Relasional Database Management System*) yang dapat menangani data yang bervolume besar. meskipun begitu, tidak menuntut resource yang besar. Mysql adalah *database* yang paling populer diantara *database* yang lain. MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan multi user. MySQL memiliki dua bentuk lisensi, yaitu free software dan shareware. MySQL sudah cukup lama dikembangkan, beberapa fase penting dalam pengembangan MySQL adalah sebagai berikut :

- a. MySQL dirilis pertama kali secara internal pada 23 Mei 1995
- b. Versi windows dirilis pada 8 Januari 1998 untuk windows 95 dan windows NT.
- c. Versi 3.23 : beta dari Juni 2000, dan dirilis pada January 2001.
- d. Versi 4.0 : beta dari Agustus 2002, dan dirilis pada Maret 2003 (unions)  
(Wahana Komputer ; 2010 :5).

### II.12. Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* atau ERD merupakan salah satu alat (tool) berbentuk grafis yang populer untuk *desain database*. Tool ini relatif lebih mudah dibandingkan dengan Normalisasi. Kebanyakan sistem analis memakai alat ini, tetapi yang jadi masalah, kalau kita cermati secara seksama, tool ini mencapai 2NF (Yuniar Supardi ; 2010 : 448).



**Gambar. II.1. Bentuk Simbol ERD**  
**(Sumber : Yuniar Supardi ; 2010 : 448)**

### II.13. 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.14. 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.14.1. 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.15. UML (*Unified Modeling Language*)

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML

adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

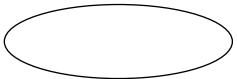
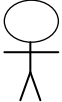
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.





Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

#### 1. *Use case* Diagram

*Use case* diagram 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. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

**Tabel II.1. Simbol *Use Case***

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang</p>



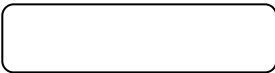
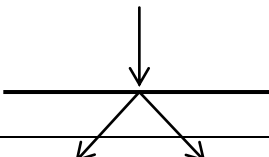
	berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

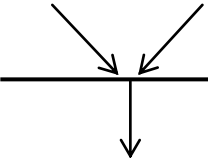
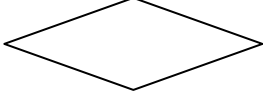

(Sumber : Windu Gata ; 2013 : 4)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

**Tabel II.2. Simbol *Activity Diagram***

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

### 3. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

*Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

**Tabel II.3. *Multiplicity Class Diagram***

<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

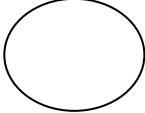
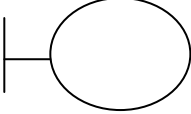
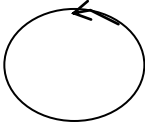

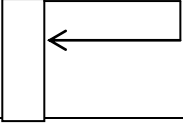

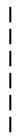
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4
------	---

(Sumber : Windu Gata ; 2013 : 9)

#### 4. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

**Tabel II.4. Simbol *Sequence Diagram***

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata ; 2013 : 7)