

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terdahulu**

Demi kesempurnaan penelitian, maka penulis perlu melakukan perbandingan untuk mengetahui gambaran dari penelitian terdahulu.

Berdasarkan Gloria Stefanie Rotikan (2013) dengan judul “Penerapan Metode *Activity Based Costing* Dalam Penentuan Harga Pokok Produksi Pada PT. Tropica Cocoprime”. Tujuan penelitian ini adalah untuk menganalisis Harga Pokok Produksi PT. Tropica Cocoprime dengan menggunakan Metode *Activity Based Costing*. Untuk perusahaan manufaktur, diharuskan dapat memproduksi produk yang berkualitas tinggi dengan harga yang dapat bersaing di pasar. Oleh karena itu, proses penentuan harga sangat penting dalam setiap perusahaan, termasuk pada PT. Tropica Cocoprime. Metode *Activity Based Costing* (ABC) merupakan sistem perhitungan harga pokok produksi yang didasarkan pada aktivitas-aktivitas dalam perusahaan sehingga dapat menghasilkan harga pokok produksi yang lebih akurat. Metode ini diharapkan dapat diterapkan pada PT. Tropica Cocoprime yang masih menggunakan Sistem Tradisional dengan metode *full costing*.

Menurut Maria Sifra Rumampuk (2013) dengan judul “Perbandingan Perhitungan Harga Pokok Produk Menggunakan Metode *Activity Based Costing* Dan Metode Konvensional Pada Usaha Peternakan Ayam CV. Kharis Di Kota Bitung”. Tujuan penelitian ini untuk mengetahui bagaimana perhitungan

harga pokok produk dan membandingkan apakah terdapat perbedaan perhitungan metode *activity based costing* dan metode konvensional pada CV. Kharis di Kota Bitung. *Activity based costing* merupakan metode penentuan harga pokok produk yang dapat mengukur secara cermat konsumsi sumber daya dalam setiap aktivitas yang digunakan dalam menghasilkan produk. Tujuan dari penelitian ini adalah untuk membandingkan perhitungan harga pokok produk berdasarkan perhitungan metode *activity based costing* dan metode konvensional yang diterapkan CV. Kharis. Hasil penelitian menunjukkan tidak terdapat perbedaan yang signifikan antara hasil perhitungan dengan menggunakan metode *activity based costing* dan metode konvensional. Manajemen disarankan menggunakan metode *activity based costing* karena akurat dalam menghitung aktivitas-aktivitas yang terjadi di perusahaan

## **II.2. Landasan Teori**

### **II.2.1. Sistem**

Sistem merupakan sekelompok perangkat keras dan perangkat lunak yang dirancang untuk mengubah data menjadi informasi yang bermanfaat. Jenis sistem informasi berbasis komputer

- a. Pengolahan Data. Pengolahan data elektronik – *electronic data processing (EDP)* adalah pemanfaatan teknologi komputer untuk melakukan pengolahan data transaksi-transaksi dalam suatu organisasi. EDP adalah aplikasi sistem informasi akuntansi paling dasar dalam setiap organisasi. Sehubungan dengan

perkembangan teknologi komputer, istilah pengolahan data mulai dikenal dan mempunyai arti yang sama dengan istilah EDP.

- b. Sistem Informasi Manajemen (SIM), menguraikan penggunaan teknologi komputer untuk menyediakan informasi bagi pengambilan keputusan para manajer.
- c. Sistem Pendukung Keputusan – *Decision Support Systems (DSS)*. DSS diarahkan untuk melayani permintaan informasi tertentu, khusus, dan tidak rutin dari manajemen. Contoh adalah penggunaan *spreadsheet* untuk melakukan analisis “*what if*” dari data operasi atau anggaran. (Agustinus Mujilan : 2012).

### **II.2.2. Informasi**

Informasi adalah sebuah perusahaan (organisasi) merupakan energi (daya pendorong) yang dibutuhkan sehingga dapat diinartkan sebagai dara yang mengalir didalam tubuh manusia. Informasi yang mendukung kelangsungan perkembangan usaha. Jika dalam masa tertntu informasi kurang didapatkan maka perusahaan akan mengalami ketidakmampuan dalam mengendalikan sumberdaya yang ada. Akibatnya pada saat penambilan keputusan strategis akan terganggu. Dal hal ini bisa mengakibatkan kekalahan ditengah – tengah persainan dunia usaha karena tida mempunyai daya saing (*competitiveness*). Disamping itu dengan adanya informasi internal, manajer bisa meningkatkan efisiensi dan efektifitas kinerja perusahaan informasi merupakan satu hal yang sangat penting bagi organisasi modren, seperti informasi laba atau rugi perusahaan pada suatu periode tertentu. (Mei Hotma Mariati : 2016).

### II.2.3. Sistem Informasi

Sistem informasi adalah suatu kumpulan dari komponen – komponen dalam perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi. System informasi dapat juga didefinisikan sebagai suatu system yang menerima sumber data sebagai *input* dan mengolahnya menjadi produk informasi sebagai *output*. System informasi merupakan suatu system yang terdiri dari beberapa subsistem atau komponen *hardware*, *software* dan *brainware*, data dan prosedur untuk menjalankan *input*, proses, *output*, penyimpanan dan pengontrolan yang mengubah sumber data menjadi informasi. Atau dapat juga didefinisikan sebagai suatu system di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung oprasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan – laporan yang diperlukan (Marimin ; 2012 : 18).

### II.2.4. Akuntansi

Akuntansi berasal dari bahasa Inggris yaitu “*To account*” yang artinya menghitung atau mempertanggungjawabkan sesuatu yang ada kaitannya dengan pengelolaan bidang keuangan dari suatu perusahaan kepada pemiliknya atas kepercayaan yang telah diberikan kepada pengelola tersebut untuk menjalankan kegiatan perusahaan. Pengertian lain akuntansi merupakan kumpulan prosedur berupa kegiatan mencatat, mengikhtisarkan, mengklasifikasikan dan melaporkan keuangan dalam bentuk laporan keuanga dalam dalam periode waktu. Laporan keuangan yang dihasilkan harus dapat di pertanggungjawabkan kepada pihak-

pihak yang berkepentingan. Akuntansi adalah proses dari transaksi yang dibuktikan dengan faktur, lalu dari transaksi dibuat jurnal, buku besar, neraca lajur kemudian akan menghasilkan Informasi dalam bentuk laporan keuangan yang digunakan pihak-pihak tertentu. (Wiratna Sujarweni : 2016 : 1)

### **II.2.5. Sistem Informasi Akuntansi**

Sistem informasi akuntansi adalah kumpulan sumberdaya, seperti manusia dan peralatan, yang diatur untuk mengubah data menjadi informasi. Informasi ini dikomunikasikan kepada beragam pengambil keputusan. Sistem Informasi Akuntansi mewujudkan perubahan ini secara manual atau terkomputerisasi. Sistem Informasi Akuntansi juga merupakan sistem yang paling penting di organisasi dan merubah cara menangkap, memproses, menyimpan, dan mendistribusikan informasi. Saat ini, digital dan informasi *online* semakin digunakan dalam sistem informasi akuntansi. Organisasi perlu menempatkan sistem di lini depan, dan mempertimbangkan baik segi sistem ataupun manusia sebagai factor yang terkait ketika mengatur sistem informasi akuntansi. Sistem Informasi Akuntansi pada umumnya meliputi beberapa siklus pemrosesan transaksi :

1. Siklus pendapatan. Berkaitan dengan pendistribusian barang dan jasa ke entitas lain dan pengumpulan pembayaran-pembayaran yang berkaitan.
2. Siklus pengeluaran. Berkaitan dengan perolehan barang jasa dari entitas lain dan pelunasan kewajiban yang berkaitan.

3. Siklus produksi. Berkaitan dengan pengubahan sumber daya menjadi barang dan jasa.
4. Siklus keuangan. Kejadian-kejadian yang berkaitan dengan perolehan dan manajemen dana-dana modal, termasuk kas. (Agustinus Mujilan : 2012 : 3).

#### **II.2.6. Harga Pokok Produksi**

Harga pokok produksi (*cost of goods manufactured*) ialah kalkulasi biaya produk jadi per unit yang terdiri dari unsur-unsur persediaan awal barang dalam proses ditambah biaya produksi dalam periode sekarang dikurangi persediaan akhir barang dalam proses. Sedangkan menurut Bustami dan Nurlela (2010) harga pokok produksi adalah kumpulan biaya produksi yang terdiri dari bahan baku langsung, tenaga kerja langsung dan biaya overhead pabrik ditambah persediaan produk dalam proses awal dan dikurang persediaan produk dalam proses akhir. Harga pokok produksi terikat pada periode waktu tertentu. Harga pokok produksi akan sama dengan biaya produksi apabila tidak ada persediaan produk dalam proses awal dan akhir. (Yohana Saputri : 2016)

#### **II.2.7. Metode *Activity Base Costing***

Metode *Activity Based Costing* (ABC) adalah suatu metode perhitungan yang sederhana untuk menentukan harga pokok produk/jasa dengan dasar bahwa aktivitaslah yang menyebabkan biaya itu timbul, bukan dari produk dan produklah yang mengkonsumsi aktivitas. Biaya-biaya tidak dapat langsung ditentukan

melalui aktivitas yang dilaluinya dan biaya untuk masing–masing aktivitas tersebut kemudian dibebankan produk atas dasar konsumsi masing–masing produk pada aktivitas dalam *Activity Based Costing* (ABC). Sistem *Activity Based Costing* dapat menyediakan informasi perhitungan biaya yang lebih baik dan dapat membantu manajemen mengelola perusahaan secara efisien serta memperoleh pemahaman yang lebih baik atas keunggulan kompetitif, kekuatan, dan kelemahan perusahaan. Sehingga dengan metode *Activity Based Costing* dapat menyajikan informasi harga pokok produksi secara cermat dan akurat bagi kepentingan manajemen. (Steven M Mamitoho : 2015)

Menurut Hansen dan Mowen dalam Budiman (2012:23) rumus yang digunakan dalam penghitungan tarif dengan menggunakan metode *activity based costing* antara lain:

1. Menentukan tarif per unit *cost driver*

$$\text{Tarif Per Unit Cost Driver} = \frac{\text{Jumlah Aktivitas}}{\text{Cost Driver}}$$

2. Menghitung pembebanan biaya *overhead* dari setiap aktivitas

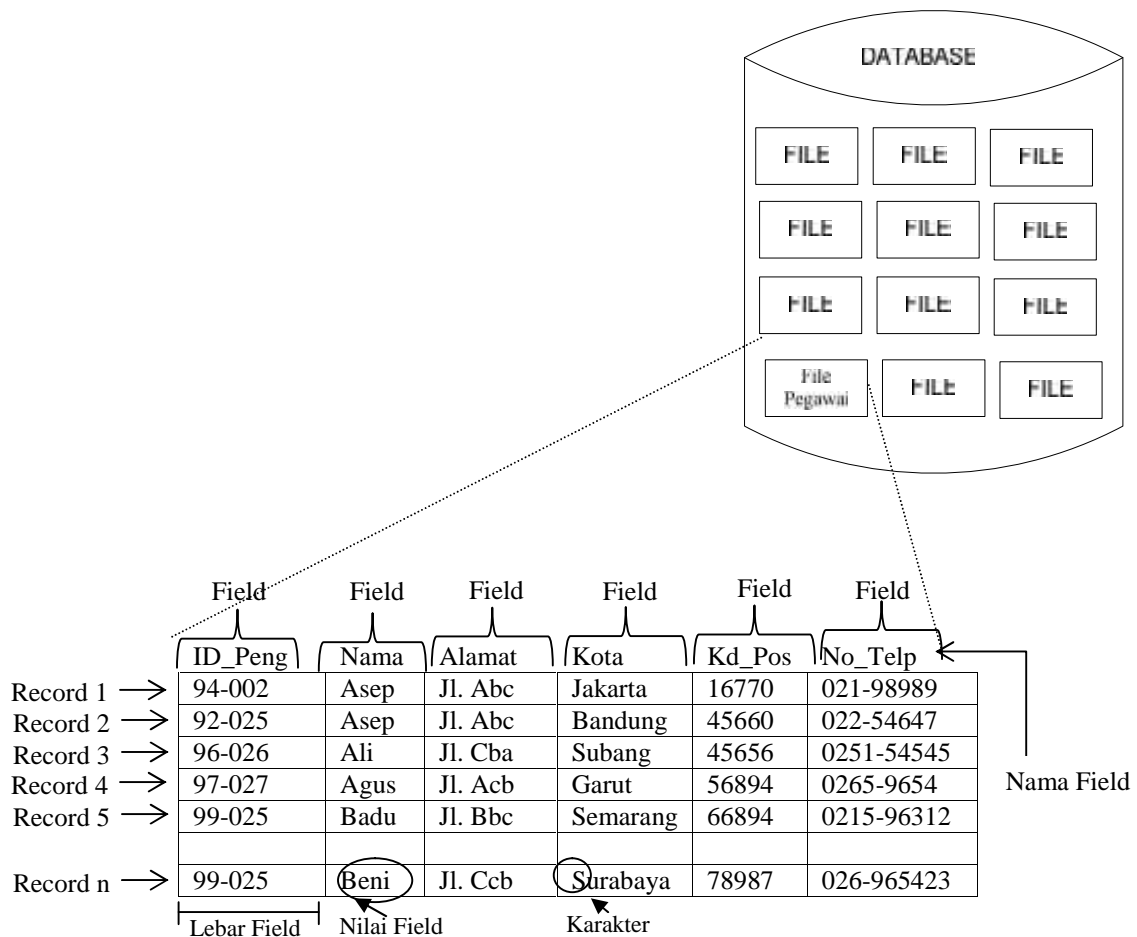
$$\text{BOP Yang dibebankan} = \frac{\text{Tarif Unit Cost Driver}}{\text{Cost Driver}}$$

### II.2.8. Database

*Database* adalah sekumpulan data mentah yang disusun menurut logika tertentu dan terorganisasi dalam bentuk yang dapat disimpan dan diproses oleh komputer. Contoh *database* dapat berisi data pegawai, data penjualan, pembayaran dan lain-lain. Data internal dari akunting, keuangan, penjualan dan bidang-bidang bisnis lainnya yang disimpan dalam suatu komputer dan disusun

menurut logika tertentu disebut dengan internal *database*. *Database* seringkali disimpan dalam suatu perangkat tertentu pada komputer, seperti *hard disk*, *compact disk*, dan sebagainya. Hubungan antar sistem *database* dan sistem *software* sangat kuat karena sistem *database* yang dipakai sangat menentukan kemudahan aksesnya data sementara *software* sendiri memungkinkan peneliti memanipulasi data untuk dianalisis. (Dermawan Wibisono : 2012)

Bahasa manipulasi yang paling banyak digunakan dewasa ini adalah *Structured Query Language (SQL)*. (Latief : 2016)



**Gambar II.1. Hirarki Database**  
 Sumber : Mukhlisulfatih Latief : 2016

Bahasa SQL tersusun atas 3 kelompok pernyataan berdasarkan fungsi dari pernyataan tersebut yaitu:

1. *Data Definition Language* (DDL) : Mendefinisikan jenis data yang akan dibuat (dapat berupa angka atau huruf), cara relasi data, validasi data dan lainnya. Contoh : *create, drop, alter table*.
2. *Data Manipulation Language* (DML) : Data yang telah dibuat dan didefinisikan tersebut akan dilakukan beberapa pengerjaan, seperti menyaring data, melakukan proses *query*. Contoh : *select, update, insert*.
3. *Data Control Language* (DCL) : Bagian ini berkenaan dengan cara mengendalikan data, seperti siapa saja yang bisa melihat isi data, bagaimana data bisa digunakan oleh banyak *user*. (Latief : 2016)

### **II.2.9. Normalisasi**

Agar model *database* relasional bisa digunakan secara efektif, maka pengelompokan yang rumit harus disederhanakan sehingga data berlebih dan relasi yang salah bisa dieliminasi. Proses membuat struktur data yang lebih kecil dan stabil dari sekelompok data yang rumit disebut Normalisasi.

Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data/*database*, teknik pengelompokan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik tanpa redundansi. Tujuan normalisasi adalah mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai, menghilangkan kerangkapan data, mengurangi kompleksitas, mempermudah modifikasi data. (Mukhlisulfatih Latief : 2016)

## 1. Proses Normalisasi

- a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu kebeberapa tingkat.
- b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

## 2. Tahapan Normalisasi :

- 1) Bentuk tidak normal : Menghilangkan perulangan grup.

**Tabel II.1. Contoh bentuk tidak normal (Unnormal)**

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
			M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C
			Akn201	Akuntansi	D310	Lia	B
			MKT300	Dasar Pemasaran	B212	Lola	A

*Sumber : Mukhlisulfatih Latief : 2016*

- 2) Bentuk Normal pertama (1NF) : Menghilangkan ketergantungan sebagian.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kesatu bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data.

**Tabel II.2. Contoh Bentuk Normal Pertama (1NF)**

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
2683	Welli	MI	M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C
5432	Bakti	AK	Akn201	Akuntansi	D310	Lia	B
5432	Bakti	AK	MKT300	Dasar Pemasaran	B212	Lola	A

*Sumber : Mukhlisulfatih Latief : 2016*

3) Bentuk Normal kedua (2NF) : Menghilangkan ketergantungan transitif.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kedua bila relasi tersebut sudah memenuhi bentuk normal kesatu dan atribut yang bukan *key* sudah tergantung penuh terhadap *key*-nya.

**Tabel II.3. Contoh Bentuk Normal Kedua (2NF)**

Kode-MK	Nama-MK	Kode Dosen	Nama Dosen
M1350	Manajemen DB	B104	Ati
M1465	Analisis Perc. Sistem	B317	Dita
M1350	Manajemen DV	B104	Ati
Akn201	Akuntansi	D310	Lia
MKT300	Dasar Pemasaran	B212	Lola

**Sumber : Mukhlisulfatih Latief : 2016**

4) Bentuk Normal ketiga (3NF) : Menghilangkan anomali-anomali hasil dari ketergantungan fungsional. Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal ketiga bila relasi tersebut sudah memenuhi bentuk normal kedua dan atribut yang bukan *key* tidak tergantung transitif terhadap *key*-nya.

**Tabel II.4. Contoh Tabel Mahasiswa Dan Tabel Kuliah (3NF)**

No_Mhs	Nama Mhs	Jurusan
2683	Welli	MI
5432	Bakti	AK

*Sumber : Mukhlisulfatih Latief : 2016*

### II.2.10. Visual Studio 2010

*Visual Studio 2010* adalah inkarnasi dari bahasa *visual basic* yang sangat populer dan telah dilengkapi dengan fitur serta fungsi yang setara dengan bahasa tingkat lainnya seperti C++. Anda dapat menggunakan *visual basic 2010* untuk membuat aplikasi *windows, mobile, web, dan office* atau kode yang telah ditulis oleh orang lain dan kemudian dimasukkan ke program lainnya. *Visual basic* menyediakan berbagai tools dan *fitur* canggih yang memungkinkan dapat menulis kode, menguji dan menjalankan program tunggal atau terkadang serangkaian program yang terkait dengan satu aplikasi. (Christopher Lee:2014:1)

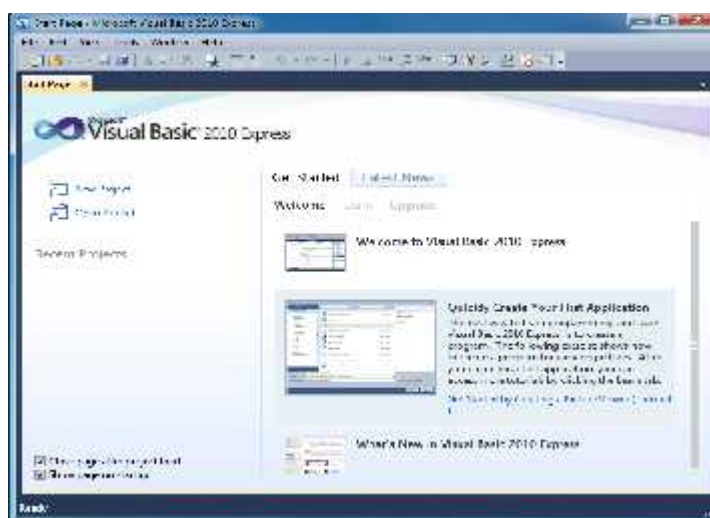
**Tabel II.5. Jendela Aplikasi Visual Studio 2010**

No	Bagian	Keterangan
1	<i>Title Bar</i>	Menampilkan nama aplikasi yang sedang terbuka.
2	Menu Bar	Menampilkan daftar perintah yang memungkinkan anda dapat menulis, mengedit, menyimpan, mencetak, menguji dan menjalankan program <i>visual basic</i> .
3	<i>Standard Toollbar</i>	Berisi Tombol yang menjalankan perintah yang sering digunakan seperti <i>open project, new project, save, cut, copy, paste, dan undo</i>
4	<i>Toolbox</i>	Berisi komponen <i>NET</i> yang dapat anda gunakan untuk mengembangkan antarmuka pengguna grafis untuk program <i>visual studio 2010</i> .
5	Area Kerja Utama	Menampilkan item yang sedang di kerjakan
6	<i>Solution Explorer</i>	Menampilkan elemen dari <i>visual basic solution</i> , yaitu nama yang diberikan kepada program <i>visual basic</i> dan item lainnya yang dihasilkan oleh <i>visual basic 2010</i> sehingga program akan mengeksekusi dengan benar.

7	<i>Properties Window</i>	Setiap Objek dalm program <i>visual basic</i> memiliki seperangkat karakteristik yang disebut sifat-sifat objek.
---	--------------------------	--

(Sumber : Christopher Lee ; 2014 : 2)

Untuk melihat tampilan *visual basic* 2010 dapat dilihat pada gambar II.2. sebagai berikut :



**Gambar II.2. Tampilan Utama Visual Basic 2010**  
(Sumber : Christopher Lee ; 2014 ; 2)

### II.2.11. SQL Server 2008

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang sangat *powerful* dan telah terbukti kekuatannya dalam mengolah data. Dalam versi terbarunya ini, *SQL Server* 2008 memiliki banyak fitur yang bisa diandalkan untuk meningkatkan *performa database*. *SQL Server* 2008 memiliki suatu GUI (*Graphic User Interface*) yang kita gunakan untuk melakukan aktivitas sehari hari berkaitan dengan *database*, seperti menulis T-SQL, melakukan *backup* dan *restore database*, melakukan *security database* terhadap aplikasi, dan sebagainya. Pada GUI tersebut kita bisa melakukan settingan terhadap *SQL Server* untuk berkerja lebih optimal. Settingan juga bisa

dilakukan menggunakan *script* untuk memudahkan *developer* mengubah *Setting Options* pada *SQL Server 2008*. (Ruslan; 2013 : 9)


### II.2.12. UML (*Unified Modelling Language*)

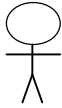

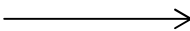
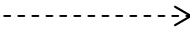
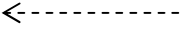
Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga 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. (Windu Gata : 2013)

#### 1. *Diagram Use Case (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.6. Simbol *Use Case Diagram***

Gambar	Keterangan
	<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>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 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>.</p>
	<p>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 mengidikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><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.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

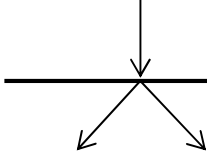
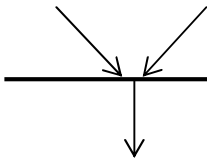
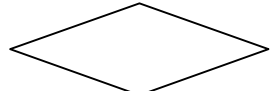

(Sumber : Windu Gata ; 2013:5)

## 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.7. Simbol Diagram Aktivitas**

Gambar	Keterangan
	<p><i>Start point</i>, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.</p>
	<p><i>End point</i>, akhir aktifitas.</p>
	<p><i>Activites</i>, menggambarkan suatu proses/kegiatan bisnis.</p>

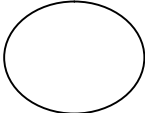
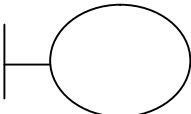
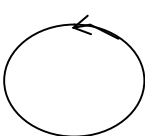
	<p><i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.</p>
	<p><i>Join</i> (penggabungan) atau <i>rake</i>, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i>, <i>false</i>.</p>
	<p><i>Swimlane</i>, pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.</p>


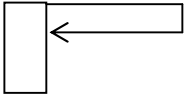


(Sumber : Windu Gata ; 2014:6)

### 3. 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.8. Simbol Sequence Diagram**

Gambar	Keterangan
	<p><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.</p>
	<p><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.</p>
	<p><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.</p>

	<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)

#### 4. 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.9. Simbol *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)*