

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Peneliti Terdahulu**

Berdasarkan penelitian yang dilakukan oleh Octaviani, dkk (2012) mengenai Tinjauan Atas Metode Pencatatan Dan Penilaian Persediaan Barang Pada Direktorat Aerostructure PT. Dirgantara Indonesia (PERSERO) maka Octaviani, dkk menyimpulkan bahwa Prosedur perediaan barang masuk dan persediaan barang keluar pada Direktorat Aerostructure PT. Dirgantara Indonesia pada umumnya telah berjalan dengan baik sesuai dengan kebijakan yang telah ditetapkan oleh perusahaan.

Berdasarkan penelitian yang dilakukan oleh Pakaya (2012) mengenai Penerapan Pencatatan Akuntansi Persediaan Barang maka Pakaya menyimpulkan bahwa penerapan persediaan barang pada Meubel Pusita masih dilakukan secara manual. Meubel tersebut juga belum menerapkan pencatatan jurnal persediaan barang dalam transaksi pembelian dan penjualan. Sedangkan untuk penilaian persediaan barang dan kartu persediaan belum diterapkan pada Meubel ini.

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

Berikut ini adalah landasan teori yang berkaitan dengan penulisan pada penelitian ini :

### **II.2.1. Sistem Informasi Akuntansi**

Sistem Informasi Akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Misalnya, salah satu input dari Sistem Informasi Akuntansi pada sebuah toko baju, seperti pada contoh sebelumnya, adalah transaksi penjualan. Kita memproses transaksi dengan mencatat penjualan tersebut ke dalam jurnal penjualan, mengklasifikasikan transaksi dengan menggunakan kode rekening, dan memposting transaksi ke dalam jurnal. Kemudian, secara periodik Sistem Informasi Akuntansi dan menghasilkan output berupa laporan keuangan yang terdiri dari Neraca dan Laporan Laba Rugi (Diana dan Setiawati, 2011 : 4).

#### **II.2.1.1. Tujuan Sistem Informasi Akuntansi**

Lingkup Sistem Informasi Akuntansi dapat dijelaskan dari manfaat yang didapat dan informasi akuntansi. Manfaat dan tujuan Sistem Informasi Akuntansi tersebut adalah sebagai berikut :

1. Mengamankan Harta/ Kekayaan Perusahaan.

Harta/kekayaan disini meliputi kas perusahaann, persediaan barang dagangan, termasuk asset perusahaan. Tidak ada pemilik yang senang jika uang dicuri orang (entah itu karyawan maupun orang asing). Contoh, seorang memiliki usaha jasa persewaan komik. Pemilik menempatkan seorang kasir di tempat persewaan tersebut. Setiap malam, pemilik akan mengambil kas hasil persewaan. Tentunya, pemilik tidak suka jika kasir tersebut tidak menyetorkan seluruh kas yang diterima. Kesempatan untuk mencuri uang perusahaan

seperti ini dapat diminimalkan, jika pemilik persewaan komik tersebut membangun sistem yang baik. Bagaimana caranya, akan dibahas lebih lanjut di ulasan-ulasan berikutnya.

## 2. Menghasilkan Beragam Informasi Untuk Pengambilan Keputusan.

Misal, pengelola toko swalayan memerlukan informasi mengenai barang apa yang diminati oleh konsumen. Membeli barang dagangan yang kurang laku berarti kas akan terjebak dalam persediaan (yang sulit laku tersebut) dan berarti kehilangan kesempatan untuk membeli barang dagangan yang laku. Hal ini sangat penting, karena toko swalayan pada dasarnya tidak dapat mengambil margin laba yang tinggi (karena ketatnya persaingan antar toko swalayan). Jadi toko swalayan lebih mengandalkan pada perputaran persediaan yang cepat. Oleh karena itu informasi mengenai persediaan yang laris merupakan kunci sukses sebuah toko swalayan. Informasi semacam ini dapat diakses dengan mudah jika toko swalayan tersebut membangun sistem informasi yang baik.

## 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). Tanpa sistem yang baik, bisa jadi pengelola kesulitan untuk menentukan besarnya omset dan besarnya laba rugi usaha. Selain untuk kepentingan perpajakan, adakalanya pengelola usaha juga terlibat

dengan kegiatan utang piutang dengan bank atau koperasi simpan pinjam. Bank membutuhkan informasi omset dan laba rugi usaha untuk memutuskan besarnya utang yang akan diberikan.

4. Menghasilkan Informasi Untuk Penilaian Kinerja Karyawan Atau Divisi

Sistem informasi dapat juga dimanfaatkan untuk penilaian kinerja karyawan atau divisi. Sebagai contoh, pengelola toko swalayan dapat memanfaatkan data penjualan untuk menilai kinerja kasir. Kasir mana yang lebih cepat dan lebih cermat dalam melayani pelanggan. Apresiasi pada karyawan yang rajin berguna untuk memotivasi karyawan dan meminimalkan sikap malas-malasan di tempat kerja.

5. Menyediakan Data Masa Lalu Untuk Kepentingan Audit (Pemeriksaan).

Data yang tersimpan dengan baik sangat memudahkan proses audit (pemeriksaan). Satu hal yang penting, audit bukan eksklusif milik perusahaan publik. Semua perusahaan mesti siap untuk menghadapi pemeriksaan (sekalipun perusahaan perseorangan), karena kantor pajak punya wewenang untuk melakukan pemeriksaan terhadap wajib pajak. Jadi, tidak ada alasan bagi satu kegiatan usaha untuk mendapat perkecualian bebas dari pemeriksaan. Benar, belum tentu dalam lima tahun, perusahaan kena giliran diperiksa, tetapi tidak ada salahnya jika perusahaan selalu siap dengan data dan dokumen pendukung yang rapi sehingga mudah diaudit. Tambahan lagi, sekalipun tidak ada pemeriksaan dari kantor pajak, baik jika sekali waktu perusahaan diaudit oleh pihak eksternal. Audit semacam ini berguna bagi perusahaan untuk evaluasi diri serta untuk menimbulkan kewaspadaan

(kehati-hatian) pada karyawan administrasi bahwa apa yang mereka kerjakan suatu saat akan diperiksa oleh pihak lain.

6. Menghasilkan Informasi Untuk Penyusunan Dan Evaluasi Anggaran Perusahaan.

Anggaran merupakan alat yang sering digunakan perusahaan untuk mengendalikan pengeluaran kas. Anggaran membatasi pengeluaran seperti yang telah disetujui dan menghindari pengeluaran yang seharusnya tidak dikeluarkan, dan berapa besarnya. Anggaran bermanfaat untuk mengalokasikan dana yang terbatas. Anggaran berperan dalam menerapkan skala prioritas pengeluaran sesuai tujuan perusahaan. Sistem informasi dapat dirancang untuk mempermudah pengawasan pengeluaran, apakah melewati batas anggaran yang telah disetujui.

7. Menghasilkan Informasi Yang Diperlukan Dalam Kegiatan Perancangan Dan Pengendalian.

Selain berguna membandingkan informasi yang berkaitan dengan anggaran dan biaya standar dengan kenyataan seperti yang telah dikemukakan sebelumnya, data historis yang diproses oleh sistem informasi dapat digunakan untuk meramal pertumbuhan penjualan dan aliran kas atau untuk mengetahui tren jangka panjang beserta korelasinya (Diana dan Setiawati, 2011 : 7).

### II.2.1.2. Perbandingan Antara Sistem Informasi Akuntansi Manual Dan Terotomatis

Perbandingan antara Sistem Informasi Akuntansi manual dan terotomatisasi terletak pada teknologi yang digunakan. Pada Sistem Informasi Akuntansi terotomatisasi, input data penjualan menggunakan alat pemindai *barcode* (*barcode scanner*), sehingga proses entri menjadi lebih cepat dan akurat dari pada di lakukan secara manual. Begitu juga dengan pemrosesan datanya, Sistem Informasi Akuntansi terotomatisasi menggunakan program aplikasi seperti *Microsoft Excel* atau bahkan menggunakan paket *software* seperti MYOB. Tabel II.1 membantu memperjelas perbedaan antara kedua hal tersebut (Diana dan Setiawati, 2011 : 7).

**Tabel II.1. Perbandingan Siklus Akuntansi Manual Dan Terotomatis.**

<b>Siklus Akuntansi Manual</b>	<b>Siklus Akuntansi Terotomatis</b>
Menjurnal: mencatat transaksi dalam jurnal, misalnya transaksi penjualan dicatat dalam jurnal penjualan.	<i>Input</i> : mencatat transaksi ke dalam file transaksi, misalnya dokumen sumber dari transaksi penjualan dicatat dalam file transaksi penjualan.
Memposting: memposting setiap entri dari jurnal ke dalam buku pembantu.	Proses: mencatat setiap transaksi ke dalam file master, misalnya mencatat setiap transaksi penjualan ke dalam file master Piutang.
Memposting: memposting total jurnal (misalnya total jurnal penjualan) ke buku besar.	Proses: mentotal transaksi dalam file transaksi (misalnya transaksi penjualan ke dalam file transaksi penjualan) dan mencatat ke dalam file master buku besar.
Meringkas: menyiapkan Neraca Lajur	<i>Output</i> : memanggil file master buku besar dan mencetak Neraca Lajur.

(Sumber : Diana dan Setiawati, 2011 : 8)

Terdapat 2 macam cara untuk meng-*update* file master dalam Sistem Informasi Akuntansi terotomatisasi, yaitu dengan pemrosesan transaksi dan pemeliharaan file (*file maintenance*). Pemrosesan transaksi meliputi fungsi pemrosesan data yang berkaitan dengan kejadian ekonomis seperti transaksi akuntansi, kegiatan operasional internal (produksi), dan penyusunan laporan keuangan. Sedangkan, pemeliharaan file meliputi kegiatan yang berkaitan dengan menambah, menghapus, atau mengganti data pada file master seperti mengubah alamat pelanggan pada file Piutang atau mengubah harga jual pada file persediaan (Diana dan Setiawati, 2011, 8).

### **II.2.1.3. Keterkaitan Sistem Informasi Akuntansi Dengan Proses Bisnis Dan Organisasi**

Sistem Informasi Akuntansi memiliki peranan yang penting dalam proses bisnis karena Sistem Akuntansi mengidentifikasi, mengukur, dan mencatat proses bisnis tersebut dalam suatu model yang sedemikian rupa sehingga informasi yang dihasilkan dapat dimanfaatkan oleh pihak-pihak yang berkepentingan. Dari sudut pandang akuntansi, model proses bisnis tersebut diwujudkan dalam bentuk siklus transaksi. Pengelompokan siklus transaksi biasanya berkaitan dengan beberapa kejadian yang berurutan. Sebagai contoh, siklus transaksi penjualan pada perusahaan dagang dimulai dari pemesanan barang oleh pelanggan, diikuti dengan pengiriman barang yang dipesan, lalu pembuatan laporan penjualan, dan dilanjutkan dengan penagihan.

Untuk lebih memperjelas keterkaitan Sistem Informasi Akuntansi dengan proses bisnis, marilah kita bahas mengenai organisasi sebagai pelaku

dalam proses bisnis tersebut. Organisasi merupakan suatu sistem yang tersusun dari sub-sub sistem seperti yang telah dibahas sebelumnya. Sub-sub sistem dalam organisasi meliputi manajemen, operasi, dan informasi. Sub-sub sistem tersebut berkaitan baik dengan pihak internal perusahaan, seperti karyawan, maupun dengan pihak eksternal perusahaan, seperti pelanggan dan instansi pemerintah.

Sub sistem manajemen terdiri dari orang, wewenang, kebijakan, dan prosedur yang bertujuan untuk menyusun rencana, strategi, dan pengendalian operasional organisasi. Sedangkan sub sistem operasi terdiri dari orang, peralatan, kebijakan, dan prosedur yang bertujuan untuk menjalankan organisasi. Sub sistem operasi biasanya terdiri dari produksi, personalia, penyimpanan, distribusi, pemasaran, dan penjualan.

Sub sistem informasi, termasuk Sistem Informasi Akuntansi, berguna untuk mendukung fungsi operasional dan pengambilan keputusan manajemen. Dengan memperoleh informasi yang benar, manajemen dapat menggunakan informasi tersebut untuk merencanakan dan mengendalikan kegiatan perusahaan. Sedangkan, bagi fungsi operasi, informasi yang benar sangat membantu dalam perencanaan produk pemesanan bahan baku, penyimpanan di gudang, dan penagihan (Diana dan Setiawati, 2011 : 9).

#### **II.2.1.4. Neraca, Laporan Laba Rugi, Dan Laporan Perubahan Modal**

Sebelum masuk ke pembahasan mengenai Laporan Keuangan (financial statement), berikut ini yang termasuk dalam laporan ini hanyalah rekening aktiva, kewajiban, dan modal. Pada Neraca, jumlah rekening pada

Laporan Perubahan Modal telah dimasukkan semua pendapatan dan biaya dari Laporan Laba Rugi, sehingga Neraca akan menjadi seimbang.

Laporan Laba Rugi merupakan selisih antara pendapatan dan biaya. Dengan demikian, rekening yang termasuk dalam laporan ini hanyalah rekening pendapatan dan rekening biaya.

Sedangkan laporan Perubahan Modal merupakan laporan mengenai perubahannya posisi modal akibat laba atau rugi yang terjadi. Dengan demikian, pada laporan ini, modal awal akan ditambah dengan laba atau dikurangi rugi yang terjadi pada periode berjalan, sehingga diperoleh modal akhir periode (Diana dan Setiawati, 2011 : 27).

### **II.2.2. *Physical Inventory***

*Physical Inventory* adalah kuantitas persediaan di tangan ditentukan, seperti yang tersirat oleh namanya, secara periodik. Semua pembelian persediaan selama periode akuntansi di catat dengan mendebet akun pembelian. Pada metode *Physical Inventory* setiap pemasukan dan pengeluaran persediaan di catat dalam perkiraan yang berbeda yaitu pembelian dan penjualan (Octaviani, 2013 : 3).

### **II.2.3. *Microsoft Visual Basic 2010***

*Visual basic* dibuat oleh Microsoft, merupakan salah satu bahasa pemrograman berorientasi objek yang mudah dipelajari. Selain menawarkan kemudahan, *Visual Basic* juga cukup andal untuk digunakan dalam pembuatan berbagai aplikasi, terutama aplikasi *database*. *Visual basic* merupakan bahasa

pemrograman *event drive*, di mana program aplikasi yang dapat berupa kejadian atau *event*, misalnya ketika *user* mengklik tombol atau menekan *enter* (Prayogi, dkk, 2015 : 3).

#### **II.2.4. Basis Data Dan DBMS**

Basis data dapat didefinisikan sebagai koleksi dari data-data yang terorganisasi sedemikian rupa sehingga data mudah disimpan dan dimanipulasi (diperbarui, dicari, diolah dengan perhitungan-perhitungan tertentu, serta dihapus). Secara teoritis, basis data tidak harus berurusan dengan komputer (misalnya, catatan belanja hari ini yang dibuat oleh seorang ibu rumah tangga juga merupakan basis data dalam bentuk yang sangat sederhana) (Nugroho, 2011, Hal : 4).

Menurut Abdul Kadir (2014) basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktifitas untuk memperoleh informasi. Basis data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas.

Untuk mengelola basis data diperlukan perangkat lunak yang disebut *Database Management System (DBMS)*. *DBMS* adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol dan mengakses basis data dengan cara yang praktis dan efisien. *DBMS* dapat digunakan untuk mengakomodasikan berbagai macam pemakai yang memiliki kebutuhan akses yang berbeda-beda (Kadir, 2014 : 218).

Umumnya *DBMS* menyediakan fitur-fitur sebagai berikut :

a. Independensi data-program

Karena basis data ditangani oleh *DBMS*, program dapat dipilih sehingga tidak tergantung pada struktur data dalam basis data. Dengan perkataan lain, program tidak akan terpenaruh sekiranya bentuk fisik data diubah.

b. Keamanan

Keamanan dimaksudkan untuk mencegah pengaksesan data oleh orang yang tidak berwenang.

c. Integritas

Hal ini ditujukan untuk menjaga agar data selalu dalam keadaan yang valid dan konsisten.

d. Konkurensi

Konkurensi memungkinkan data dapat diakses oleh banyak pemakai tanpa menimbulkan masalah.

e. Pemulihan (*recovery*)

*DBMS* menyediakan mekanisme untuk mengembalikan basis data ke keadaan semula yang konsisten sekiranya terjadi gangguan perangkat keras atau kegagalan perangkat lunak.

f. Katalog Sistem

Katalog Sistem adalah deskripsi tentang data yang terkandung dalam basis data yang dapat diakses oleh pemakai.

#### g. Perangkat Produktivitas

Untuk menyediakan kemudahan bagi pemakai dan meningkatkan produktivitas, *DBMS* menyediakan sejumlah perangkat produktivitas seperti pembangkit *query* dan pembangkit laporan (Kadir, 2014: 219).

#### II.2.5. *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 setting terhadap *SQL Server* untuk bekerja lebih optimal. Settingan juga bisa dilakukan menggunakan script untuk memudahkan developer mengubah *Setting Options* pada *SQL Server 2008* (Ruslan, 2013 : 39).

#### II.2.6. Normalisasi

Normalisasi merupakan parameter digunakan untuk menghindari duplikasi terhadap tabel dalam basis data dan juga merupakan proses mendekomposisikan sebuah tabel yang masih memiliki beberapa anomali atau ketidakwajaran sehingga menghasilkan tabel yang lebih sederhana dan

struktur yang bagus, yaitu sebuah tabel yang tidak memiliki *data redundancy* dan memungkinkan *user* untuk melakukan *insert*, *delete*, dan *update* pada baris (*record*) tanpa menyebabkan inkonsistensi data (Triyono, 2011 : 19).

#### 1. *First Normal Form (1 NF)*

Sudah tidak ada *repeating group* yaitu pengulangan yang terjadi pada beberapa atribut atau kolom dalam sebuah tabel, dan juga setiap atribut harus bernilai tunggal. Atribut *multivalued*, *composite*, *derive* tidak tunggal. Setiap nilai dari atribut hanya mempunyai nilai tunggal.

#### 2. *Second Normal Form (2 NF)*

Untuk menjadikan tabel normal tingkat ke 2 maka sudah 1NF dan setiap atribut yang bukan *primary key* sepenuhnya secara *functional* tergantung pada semua atribut pembentuk *primary key*.

#### 3. *Third Normal Form (3 NF)*

Tabel sudah 2NF dan tidak memiliki *transitive dependencies*, *Transitive dependency* adalah ketika ada atribut yang secara tidak langsung tergantung pada *primary key* dan atribut tersebut juga tergantung pada atribut lain yang bukan *primary key*.

#### 4. *Boyce-codd Normal Form (BCNF)*

Tabel dalam BCNF jika sudah 3NF dan semua *determinants* adalah *candidate keys*. Perbedaan 3NF dan BCNF adalah untuk *functional dependency*  $A \rightarrow B$ , 3NF memperbolehkan ketergantungan ada dalam relasi jika B adalah *Primary Key* dan A bukan merupakan *candidate key*. Sedangkan BCNF menuntut untuk ketergantungan tetap ada dalam relasi, A harus menjadi *candidate key*.

#### 5. *Fourth Normal Form (4 NF)*

Relasi berada pada bentuk normal keempat apabila memenuhi syarat BCNF dan tidak mempunyai *multivalued dependency*.

#### 6. *Fifth Normal Form (5 NF)*

Tabel bentuk normal kelima sering disebut PJNF (*Projection Join Normal Form*), penyebutan PJNF karena untuk suatu relasi akan berbentuk normal kelima jika tabel tersebut dapat dipecah atau diproyeksikan menjadi beberapa tabel dan dari proyeksi-proyeksi itu dapat disusun kembali (*join*) menjadi tabel yang sama dengan keadaan semula. Jika penyusunan ini tidak mungkin dilakukan dikatakan pada relasi itu terdapat *join dependencies* dan dikatakan bersifat *lossy join* (Triyono, 2011 : 20).

### **II.2.7. *Unified Modeling Language (UML)***

Menurut Windu Gata (2013) 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.

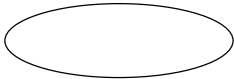
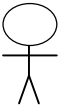
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem (Urva dan Siregar, 2015: 93).


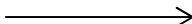
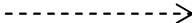
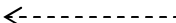
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

### 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 dapat dilihat pada Tabel II.2 dibawah ini:

**Tabel II.2. 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 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</p>




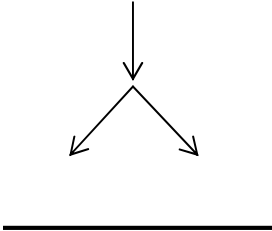
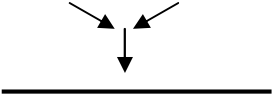
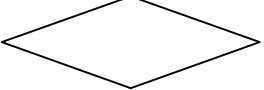

	control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan 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:Urva dan Siregar; 2015: 94)

## 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* dapat dilihat pada Tabel II.3 dibawah ini:

**Tabel II.3. Simbol *Activity Diagram***

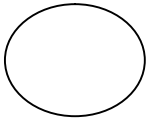
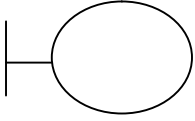
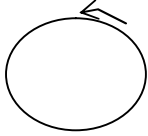

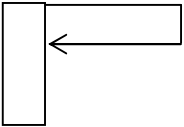
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel 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> .
 New Swimline	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.



(Sumber : Urva dan Siregar; 2015: 94)

### 3. 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* dapat dilihat pada Tabel II.4 dibawah ini :

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

Gambar	Keterangan
	<i>EntityClass</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.

	<p><i>Activation, activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Urva dan Siregar; 2015: 95)

#### 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 yang dapat dilihat pada Tabel II.5 dibawah ini:

**Tabel II.5. 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 : Urva dan Siregar; 2015 : 95)