

BAB II

TINJAUAN PUSTAKA

II.1 Sistem Informasi Akuntansi

II.1.1 Pengertian Sistem

Anastasia dan Lilis (2011:3), menyatakan bahwa sistem merupakan bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu. Suatu sistem tersusun dari sub-sub sistem yang lebih kecil yang juga saling tergantung dan bekerja sama untuk mencapai tujuan.

II.1.2 Pengertian Sistem Informasi

Anastasia dan Lili (2011:4), menyatakan bahwa sistem informasi, yang kadang kala disebut sebagai sistem pemroses data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen, baik manual maupun berbasis komputer yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi kepada pihak-pihak yang berkepentingan sebagai pemakai informasi tersebut. Komponen suatu sistem informasi terdiri dari :

- a. Input : Data yang relevan untuk menghasilkan informasi yang diinginkan.
- b. Proses : Langkah-langkah yang perlu dilakukan untuk mengolah data informasi.
- c. Output : Berupa informasi yang merupakan hasil dari pemrosesan data.

II.1.3 Pengertian Akuntansi

Suwardjono (2013:4), mengatakan kata akuntansi berasal dari kata bahasa Inggris *to account* yang berarti memperhitungkan atau mempertanggungjawabkan. Kata akuntansi diserap dari kata *accountancy* yang berarti hal-hal yang bersangkutan dengan *accountant* (akuntan) atau bersangkutan dengan hal-hal yang dikerjakan oleh akuntan dalam menjalankan profesinya.

Akuntansi adalah seni pencatatan, penggolongan dan peringkasan transaksi dan kejadian yang bersifat keuangan dengan cara yang berdaya guna dan dalam bentuk satuan uang dan menginterpretasikan hasil proses tersebut. Pengertian seni dalam definisi tersebut dimaksudkan untuk menunjukkan bahwa akuntansi bukan merupakan ilmu pengetahuan eksakta atau sains karena dalam proses penalaran dan perancangan akuntansi banyak terlibat unsur pertimbangan (*judgment*). Seni dalam definisi di atas lebih mempunyai konotasi sebagai kerajinan dan keterampilan atau pengetahuan terapan yang isi dan strukturnya disesuaikan dengan kebutuhan untuk mencapai suatu tujuan. Ismail (2010:2), mengatakan akuntansi adalah seni dalam mencatat, menggolongkan dan mengikhtisarkan semua transaksi-transaksi yang terkait dengan keuangan yang telah terjadi dengan cara yang bermakna dan dalam satuan uang.

II.1.4 Pengertian Sistem Informasi Akuntansi

Menurut Anastasia Diana dan Lilis Setiawan (2011:4) 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 kedalam jurnal. Kemudian, secara periodik Sistem Informasi Akuntansi akan menghasilkan output berupa laporan keuangan yang terdiri dari Neraca dan Laporan Laba Rugi.

II.1.5 Tujuan Sistem Informasi Akuntansi

Menurut Anastasia Diana dan Lilis Setiawan (2011:5), Tujuan dari sistem informasi akuntansi adalah sebagai berikut:

1. Mengamankan harta/kekayaan perusahaan.

Harta/kekayaan disini meliputi kas perusahaan, persediaan barang dagangan, termaksud aset tetap/aktiva tetap perusahaan. Tidak ada pemilik yang senang jika uang perusahaan dicuri orang (entah itu karyawan maupun orang asing). Contoh, seorang memiliki usaha jasa persewaan komik. Pemilik menempatkan seorang kasir ditempat 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 berikut-nya.

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 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 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 penghitungan 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 lagi bagi satu kegiatan usaha untuuk 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 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 akan menghasilkan output berupa laporan keuangan yang terdiri dari Neraca dan Laporan Laba Rugi.

II.2 Persekutuan

II.2.1. Pengertian Persekutuan

Persekutuan (*Partnership*) merupakan gabungan dua orang atau lebih yang memiliki dan menjalankan usaha untuk mendapatkan laba James (2013:91). Persekutuan memiliki beberapa karakteristik dengan pengaruh akuntansi. Di Indonesia, terdapat tiga jenis persekutuan yang diakui oleh Kitab Undang-undang Hukum Perdata (KUHP), yaitu persekutuan perdata, firma, dan persekutuan terbatas (CV).

II.2.2. Persekutuan Komanditer

Menurut James (2013:93), CV (*Commanditaire Vennootschap*) merupakan persekutuan yang terdiri atas satu rekan umum (rekan aktif) atau lebih dan satu rekan pasif atau lebih. Rekan aktif secara pribadi bertanggung jawab atas seluruh utang persekutuan, sedangkan rekan pasif, yang hanya menyumbangkan modal untuk persekutuan, hanya bertanggung jawab sebesar jumlah kontribusinya saja. Akan tetapi, meskipun rekan pasif tidak terlibat secara aktif dalam bisnis, mereka dapat mengawasi tata kelola bisnis secara internal.

Status rekan pasif memiliki perbedaan yang cukup besar dengan kreditor. Jika kreditor dapat mempertahankan klaim atas CV bahkan setelah aset persekutuan berkurang, rekan pasif hanya memiliki hak atas bagian dari persekutuan jika perusahaan menghasilkan laba. Saat persekutuan membukukan rugi, rekan pasif juga menanggungnya, sehingga dalam kedua kasus, rekan pasif hanya mendapatkan laba atau menanggung rugi sebesar jumlah kontribusi dalam persekutuan.

II.2.3 Pendirian Persekutuan

Dalam mendirikan persekutuan, investasi masing-masing rekan dicatat dalam ayat jurnal terpisah. Aset yang diserahkan oleh seorang rekan didebit ke akun aset persekutuan. Jika kewajiban diambil alih oleh persekutuan, maka akun kewajiban persekutuan yang dikredit. Akun modal milik rekan akan dikreditkan sebesar jumlah bersihnya, James (2013:94).

Sebagai contoh, diasumsikan Joko Suhendro dan Endang Fauzi, pemilik toko komputer yang semula bersaing, sepakat untuk menggabungkan usaha mereka dalam sebuah persekutuan. Masing-masing pihak menyumbangkan sejumlah uang tunai dan aset lainnya. Suhendro dan Fauzi juga sepakat bahwa persekutuan akan mengambil alih kewajiban dari perusahaan masing-masing. Ayat jurnal untuk mencatat aset yang disumbangkan dan kewajiban yang dialihkan oleh joko adalah sebagai berikut :

Tabel II.1 Jurnal Pendirian Persekutuan

Tahun		Keterangan	Debit	Kredit
April	01			
		Piutang Usaha	Rp7.200.000	
		Persediaan Barang Dagang	Rp16.300.000	
		Peralatan Toko	Rp28.700.000	
		Peralatan Kantor	Rp5.400.000	
		Penyusutan hutang tak tertagih		Rp1.500.000
		Utang Usaha		Rp2.600.000
		Modal Joko Suhendro		Rp55.000.000

(Sumber : James; 2013:94)

II.2.4. Pembagian Laba

James (2013:95), menyatakan bahwa pembagian laba dapat dibagi dengan menggunakan cara berikut :

1. Pembagian Laba-Jasa Rekan

Salah satu metode dalam mengakui perbedaan kemampuan rekan dan jumlah waktu yang diberikan kepada persekutuan adalah dengan memberikan tunjangan gaji kepada para rekan. Oleh karena rekan secara hukum bukan merupakan karyawan persekutuan, maka tunjangan semacam ini diperlakukan sebagai pembagian laba bersih dan dikreditkan ke akun modal rekan.

Sebagai contoh, diasumsikan perjanjian persekutuan antara Jamila Sari dan Chandra Mono menjanjikan tunjangan gaji bulanan. Jamila Sari akan menerima tunjangan tahunan sebesar 4% (Rp6.000.000), dan

Chandra akan menerima 2% (Rp3.000.000). Laba bersih yang tersisa setelah pembagian tunjangan gaji akan dibagi sama rata. Diasumsikan pula laba bersih tahun tersebut adalah sebesar Rp150.000.000.

Laporan pembagian laba bersih dapat disajikan sebagai laporan terpisah untuk mendukung neraca dan laporan laba rugi, atau dapat diungkapkan dalam laporan modal persekutuan. Format lainnya adalah dengan menambahkan pembagian laba di bagian bawah laporan laba rugi. Jika format terakhir yang digunakan, maka bagian bawah laporan laba rugi akan muncul sebagai berikut :

Laba bersih Rp150.000.000

Pembagian laba bersih :

	Jamila Sari	Chandra Mono	Jumlah
Tunjangan gaji tahunan	Rp6.000.000	Rp3.000.000	Rp9.000.000
Sisa laba	Rp70.500.000	Rp70.500.000	Rp141.000.000
Laba bersih	Rp76.500.000	Rp73.500.000	Rp150.000.000

2. Pembagian Laba- Jasa Rekan dan Investasi

Para rekan dapat menyepakati bahwa rencana pembagian laba yang paling adil adalah menyediakan (1) tunjangan gaji dan (2) bunga atas investasi modal. Sisa laba bersih kemudian dibagi sesuai kesepakatan. Sebagai contoh, diasumsikan perjanjian persekutuan untuk Jamila Sari dan Chandra Mono membagi laba sebagai berikut :

- a. Tunjangan gaji bulanan sebesar 4% untuk Jamila Sari dan 2% untuk Chandra Mono.

- b. Bunga sebesar 12% atas saldo modal masing-masing rekan per 1 Januari.
- c. Sisa laba bersih dibagi rata di antara para rekan.

Jamila Sari memiliki saldo kredit sebesar Rp160.000.000 dalam akun modalnya per 1 Januari tahun fiskal berjalan, dan Chandra Mono memiliki saldo kredit sebesar Rp120.000.000 dalam saldo akun modal. Laba bersih tahun tersebut sebesar Rp150.000.000 dibagi berdasarkan skema berikut :

Laba bersih Rp150.000.000

Pembagian laba bersih :

	Jamila Sari	Chandra Mono	Jumlah
Tunjangan gaji tahunan	Rp6.000.000	Rp3.000.000	Rp9.000.000
Tunjangan bunga	Rp19.200.000 ¹	Rp14.400.000 ²	Rp33.600.000
Sisa laba	Rp53.700.000	Rp53.700.000	Rp107.400.000
Laba bersih	Rp78.900.000	Rp71.100.000	Rp150.000.000

¹ 12% x Rp160.000.000

² 12% x Rp120.000.000

3. Pembagian Laba-Tunjangan Melebihi Laba Bersih

Dalam contoh sebelumnya, jumlah laba bersih melebihi total jumlah tunjangan gaji dan bunga. Jika jumlah laba bersih kurang dari total tunjangan, sisa saldo menjadi jumlah negatif. Jumlah tersebut harus dibagi diantara para rekan sebagaimana terjadi rugi bersih.

Sebagai contoh, diasumsikan jumlah tunjangan gaji dan bunga adalah sama seperti dalam contoh sebelumnya, tetapi laba bersihnya

sebesar Rp40.000.000. Total tunjangan gaji dan bunga untuk Jamila Sari sebesar Rp25.200.000 dan untuk Chandra Mono sebesar Rp17.400.000. Jumlah keduanya, yaitu Rp42.600.000, melebihi laba bersih Rp40.000.000 sebesar Rp2.600.000. Kelebihan tersebut harus dibagi dua antara Jamila Sari dan Chandra Moto. Dalam perjanjian persekutuan, sisa laba bersih atau rugi bersih apapun setelah dikurangkan dari tunjangan-tunjangan dibagi rata antara Jamila Sari dan Chandra Mono. Dengan demikian, masing-masing rekan mendapatkan setengah bagian dari Rp2.600.000, dan Rp1.300.000 dikurangkan dari bagian tunjangan setiap rekan. Pembagian laba bersih final antara Jamila Sari dan Chandra Mono ditunjukkan seperti berikut ini:

Laba bersih Rp40.000.000

Pembagian laba bersih :

	Jamila Sari	Chandra Mono	Jumlah
Tunjangan gaji tahunan	Rp6.000.000	Rp3.000.000	Rp9.000.000
Tunjangan bunga	Rp19.200.000	Rp14.400.000	Rp33.600.000
Total	Rp25.200.000	Rp17.400.000	Rp42.600.000
Dikurangi selisih tunjangan terhadap laba	Rp1.300.000	Rp1.300.000	Rp2.600.000
Laba bersih	Rp23.900.000	Rp16.100.000	Rp100.000.000

II.3 Visual Basic

Microsoft Visual Basic.NET (VB.NET) adalah suatu pengembangan aplikasibahasapemrograman berbasis Visual Basic dan merupakan bahasa pemrograman terbaru buatan Microsoft setelah Microsoft Visual Basic 6.0. Pengembangan yang signifikan dari VB.NET ialah kemampuannya memanfaatkan platform NET, sehingga pengguna dapat membuat aplikasi Windows, aplikasi konsol, pustaka kelas, layanan NT, aplikasi web form, dan XML Web Service, yang secara keseluruhan memungkinkan integrasi tanpa batas dengan bahasa pemrograman lain sehingga berpeluang untuk berintegrasi dengan web.

Jika kita membuat aplikasi dengan Visual Basic maka kita akan mendapatkan *file* yang menyusun aplikasi tersebut, yaitu :

1. File Project (*.vbp)

File ini merupakan kumpulan dari aplikasi yang kita buat. File project bisa berupa file *.frm, *.dsr atau file lainnya.

2. File Form (*.frm)

File ini merupakan file yang berfungsi untuk menyimpan informasi tentang bentuk form maupun *interface* yang kita buat (Edy Winarno ; 2010 : 83).

II.4 Microsoft SQL Server

Stefano (2014:4), menyatakan bahwa *Microsoft SQL Server* adalah sebuah manajemen basis data relasional (RDBMS) produk *microsoft*.

Bahasa kueri utamanya adalah *Transact-SQL* yang merupakan implementasi dari *SQL* standar *ANSI/ISO* yang digunakan oleh *microsoft* dan *Sybase*. Umumnya, *SQL* digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai menengah, tetapi kemudian berkembang dengan digunakannya *SQL Server* pada basis data besar.

Microsoft *SQL Server* juga mendukung *ODBC (Open Database Connectivity)*, dan mempunyai *driver JDBC* untuk bahasa pemrograman java. Fitur lain dari *SQL Server* adalah kemampuannya membuat basis data *mirroring* dan *clustering*.

Beberapa versi *SQL Server* yang telah berjalan :

1. *SQL Server 2008R2 Datacenter*
2. *SQL Server Enterprise Edition*
3. *SQL Server Standard Edition*
4. *SQL Server Web Edition*
5. *SQL Server Workgroup Edition*
6. *SQL Server Express Edition*
7. *SQL Server Developer Edition*

II.5 Normalisasi

II.5.1 Pengertian Normalisasi

Janer Simarmata (2012:57), menyatakan bahwa normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata 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.5.2 Bentuk-Bentuk Normalisasi

1. Bentuk normal tahap pertama (1st normal form)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

2. Bentuk normal tahap kedua (2nd normal form)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

3. Bentuk normal tahap ketiga (3rd normal form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada

pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

4. Boyce Code Normal Form (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

5. Bentuk Normal Tahap Keempat dan Kelima

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD).

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata ; 2010 : 76).

II.6 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.

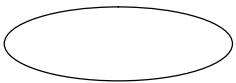
UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

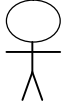


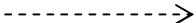
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

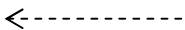
1. *Use case* Diagram

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.2 Simbol *Use Case* Diagram

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan</p>

	dengan menggunakan kata kerja di awal nama <i>use case</i> .
	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> .
	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</i>




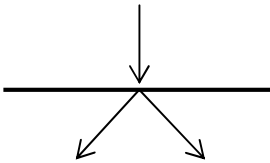
	<i>case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

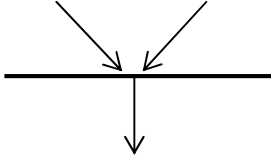
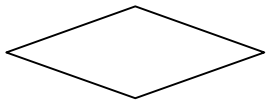

(Sumber : Windu Gata ; 2013 : 4)

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.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.

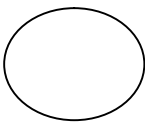
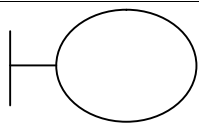
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

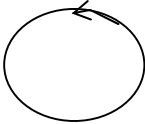
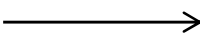
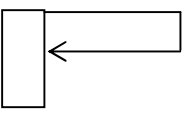


(Sumber : Windu Gata ; 2013 : 6)

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*, yaitu:

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.
	<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. Diagram Kelas (*Class Diagram*)

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 *multiplicity* atau kardinaliti.

Tabel II.5 *Multiplicity Class Diagram*

<i>Multiplicity</i>	Penjelasan
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 maksimal 4

(Sumber : Windu Gata ; 2013:9)