

BAB II

LANDASAN TEORI

II.1. Sistem Informasi Akuntansi

Sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan transaksi. Misalnya, salah satu input dari sistem informasi pada sebuah toko baju adalah transaksi penjualan (Anastasia Diana & Lilis Setiawati ; 2011 : 3).

Sistem informasi akuntansi adalah kumpulan sumberdaya, seperti manusia dan peralatan, yang diatur untuk mengubah data menjadi informasi. Informasi ini dikomunikasikan kepada beragam pengambil keputusan. SIA mewujudkan perubahan ini secara manual atau terkomputerisasi (Agustinus Mujilan ; 2012 : 3).

Akuntansi didefinisikan sebagai proses mengidentifikasi, mencatat, dan mengkomunikasikan peristiwa ekonomis pada suatu organisasi pada pihak yang membutuhkan. Dalam prakteknya akuntansi menyediakan dua macam laporan bagi pengguna eksternal dan internal. Untuk pengguna eksternal, informasi yang dihasilkan biasanya berupa laporan keuangan yang terdiri dari neraca, rugi laba, dan perubahan arus kas. Jenis laporan, macam laporan, jumlah, dan cara persiapan maupun penyajian laporan akuntansi diatur dengan standart akuntansi keuangan. Pengguna eksternal menggunakan laporan keuangan tersebut untuk membuat keputusan investasi, perpajakan, pemberian kredit, dan lain-lain. Untuk pengguna internal, laporan yyang dihasilkan oleh akuntansi digunakan untuk kepentingan

pengelola organisasi, laporan yang dihasilkan oleh akuntansi digunakan untuk kepentingan pengelolaan organisasi.

II.1.1. Tujuan Sistem Informasi Akuntansi

Lingkup sistem informasi akuntansi dapat dijelaskan dari manfaat yg didapat dari informasi akuntansi. Manfaat atau tujuan sistem informasi akuntansi tersebut adalah sebagai berikut (Anastasia Diana & Lilis Setiawati ; 2011 : 5-6) :

1. Mengamankan harta/kekayaan perusahaan

Harta//kekayaan disini 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 yang diminati oleh konsumen.

3. Menghasilkan informasi untuk pihak internal

Setiap pengelola usaha memiliki kewajiban membayar pajak. Besarnya pajak tergantung dari besarnya omset penjualan atau tergantung pada laba rugi usaha. Tanpa sistem yang baik, bisa jadi pengelola kesulitan untuk menentukan besarnya omset dan besarnya laba rugi usaha.

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.

5. Menyediakan data masa lalu untuk kepentingan audit(pemeriksaan)

Data yang tersimpan dengan baik memudahkan proses audit. Sat hal yang penting, audit bukan eksklusif milik perusahaan publik. Semua perusahaan mesti siap untuk menghadapi pemeriksaan, karna kantor pajak punya wewenang untuk melakukan pemeriksaan terhadap wajib pajak.

6. Menghasilkan informasi untuk penyusunan dan evaluasi anggaran perusahaan.

Anggaran merupakan alat yang sering digunakan perusahaan untuk mengendalikan pengeluaran kas.

7. Menghasilkan informasi yg diperlukan dalam kegiatan perencanaan dan pengendalian.

Selain berguna untuk 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.

II.1.2. Siklus Pemrosesan Transaksi dalam Sistem Informasi Akuntansi

Pada umumnya Sistem Informasi Akuntansi meliputi beberapa siklus pemrosesan transaksi, antara lain sebagai berikut :

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 perubahan sumberdaya 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. Aktiva Tetap

Aktiva tetap adalah kekayaan yang dimiliki perusahaan yang fisiknya nampak dan sifatnya relatif permanen. Syarat lain untuk dapat diklasifikasikan sebagai aktiva tetap selain aktiva itu dimiliki perusahaan, juga harus digunakan dalam operasi yang bersifat permanen (aktiva tersebut mempunyai umur kegunaan jangka panjang atau tidak akan habis pakai dalam satu periode kegiatan perusahaan). Menurut standar akuntansi keuangan, suatu benda berwujud harus diakui sebagai suatu aktiva dan dikelompokkan sebagai aktiva tetap bila :

1. Besar kemungkinan (*probable*) bahwa manfaat ekonomi dimasa yang akan datang yang berkaitan dengan aktiva tersebut akan mengalir ke dalam perusahaan.
2. Biaya perolehan aktiva dapat diukur secara andal.

Dalam menentukan apakah suatu pos tersebut memenuhi kriteria pertama untuk suatu perusahaan harus menilai tingkat kepastian aliran manfaat ekonomi

masa yang akan datang berdasarkan bukti yang tersedia pada waktu pengakuan awal. Adanya kepastian yang cukup bahwa manfaat ekonomi masa yang akan datang mengalir ke perusahaan membutuhkan suatu kepastian bahwa perusahaan akan menerima imbalan dan menerima resiko terkait. Kepastian ini biasanya hanya tersedia jika resiko dan imbalan telah diterima perusahaan. Sebelum hal ini terjadi, transaksi untuk memperoleh aktiva biasanya dapat dibatalkan tanpa sanksi yang signifikan dan karenanya aktiva tidak diakui. Kriteria kedua untuk pengakuan biasanya dapat dipenuhi langsung kerana transaksi pertukaran mempunyai bukti pembelian aktiva mengidentifikasi biayanya. Dalam keadaan suatu aktiva yang dikonstruksi sendiri, suatu pengukuran yang dapat diandalkan atas biaya dapat dibuat dari transaksi dengan pihak eksternal dan perusahaan untuk perolehan bahan baku, tenaga kerja dan input lain yang digunakan dalam proses konstruksi aktiva tetap dapat diperoleh dengan berbagai cara. Berikut ini adalah cara perolehan aktiva tetap yang ada dalam berbagai literatur.

1. Pembelian Tunai

Aktiva tetap berwujud yang diperoleh dari pembelian tunai dicatat dalam buku-buku dengan jumlah sebesar uang yang dikeluarkan. Dalam jumlah uang yang dikeluarkan untuk memperoleh aktiva tetap termasuk harga faktur dan semua biaya yang dikeluarkan agar aktiva tetap tersebut.

2. Pembelian Angsuran

Apabila aktiva tetap diperoleh dari pembelian angsuran, maka harga perolehan aktiva tetap tidak boleh termasuk bunga. Hal ini sesuai dengan sak

16 revisi 2007 paragraf 23, yang menyebutkan “biaya perolehan aset tetap adalah setara dengan nilai tunai dan diakui pada saat terjadinya. Jika pembayaran untuk suatu aset ditangguhkan hingga melampaui jangka waktu kredit normal, perbedaan antara nilai tunai dengan pembayaran total diakui sebagai beban bunga selama periode kredit” dengan kata lain aset tetap yang dibeli dengan cara angsuran harus diakui sebesar nilai tunai dan selisih antara nilai tunai dengan total biaya yang dikeluarkan diakui sebagai biaya bunga.

3. Ditukar dengan surat-surat berharga

Aktiva tetap yang diperoleh dengan cara ditukar dengan saham atau obligasi perusahaan, dicatat dalam buku sebesar harga pasar saham atau obligasi yang digunakan sebagai penukar. Apabila harga pasar saham atau obligasi itu tidak diketahui, harga perolehan aktiva tetap ditentukan sebesar harga pasar aktiva tersebut. Kadang-kadang harga pasar surat berharga dan aktiva tetap yang ditukar kedua-duanya tidak diketahui. Dalam keadaan seperti ini nilai pertukaran ditentukan oleh keputusan pimpinan perusahaan. Nilai pertukaran ini dipakai sebagai dasar pencatatan harga perolehan aktiva tetap dan nilai surat-surat berharga yang dikeluarkan.

4. Ditukar dengan aktiva tetap

Pertukaran aktiva dipisahkan menjadi dua, yaitu :

a. Pertukaran aktiva tetap tidak sejenis

Pertukaran aktiva yang tidak sejenis adalah pertukaran aktiva tetap yang sifat dan fungsinya tidak sama seperti pertukaran tanah dengan mesin-mesin,

tanah dengan gedung, dan lain-lain. Dalam sak nomor 16 tahun 2004 paragraf 20 menyebutkan “suatu aset tetap dapat diperoleh dari pertukaran atau pertukaran sebagian untuk suatu aset tetap yang tidak serupa atau aset lain. Biaya dari pos semacam itu diukur pada nilai wajar aset yang dilepaskan atau yang diperoleh, yang mana yang lebih andal ekuivalen dengan nilai wajar aktiva yang dilepaskan setelah disesuaikan dengan jumlah setiap kas atau setara kas yang ditransfer”.

b. Pertukaran aktiva tetap sejenis

Pertukaran aktiva tetap yang sejenis adalah pertukaran aktiva tetap yang sifat dan fungsinya sama seperti pertukaran mesin produksi merk a dengan merk b, truk merk a dengan truk merk b, dan seterusnya. Perlakuan akuntansi dalam masalah ini dapat mengacu pada SAK no 16 tahun 2004 paragraf 21 yang menyebutkan “suatu aset tetap dapat diperoleh dari pertukaran atas suatu aset yang serupa yang memiliki manfaat yang serupa dalam bidang usaha yang sama dan memiliki suatu nilai wajar serupa. Suatu aset juga dapat dijual dalam pertukaran dengan kepemilikan aset yang serupa. Dalam keadaan tersebut, karena proses perolehan penghasilan (*earning process*) tidak lengkap, tidak ada keuntungan dan kerugian yang diakui dalam transaksi. Sebaliknya, biaya perolehan aset baru adalah jumlah tercatat dari aset yang dilepaskan. Tetapi nilai wajar aset yang diterima dapat menyediakan bukti dari suatu pengurangan (*impairment*) aset yang dilepas. Dalam keadaan ini aset yang dilepaskan diturun nilai bukukan (*written down*) dan nilai turun buku (*written down*) ini ditetapkan untuk nilai aset baru”.

5. Diperoleh dari hadiah/donasi

Perlakuan akuntansi dalam masalah ini dapat mengacu pada sak no 16 tahun 2004 paragraf 21 yang menyebutkan “aktiva tetap yang diperoleh dari sumbangan harus dicatat sebesar harga taksiran atau harga pasar yang layak dengan mengkredit akun “modal donasi”.

6. Aktiva yang dibuat sendiri

Ada dua cara dalam membebankan biaya overhead pabrik, yaitu:

- a. Biaya *overhead* pabrik dialokasikan dengan tarif kepada pembuatan aktiva dan produksi.
- b. Kenaikan biaya *overhead* pabrik yang dibebankan pada aktiva yang dibuat.

Apabila digunakan cara yang pertama harga pokok aktiva merupakan jumlah semua biaya langsung ditambah dengan tarif yang menjadi beban aktiva yang dibuat. Apabila cara yang kedua yang dipergunakan, maka harga pokok aktiva yang dibuat adalah biaya langsung untuk membuat aktiva itu ditambah dengan kenaikan biaya *overhead* pabrik. Hal yang perlu diperhatikan bahwa harga pokok aktiva yang dibangun sendiri, jumlah biaya atas pembangunan tersebut tidak boleh melebihi harga pasar dari aktiva yang dibangun (Iskandar, Efendi dan Syamsu Rizal ; Jurnal Akuntansi & Keuangan : 2011 : 101-103).

II.2. Penyusutan Aset Tetap

Dr. Erhans Anggawirya “penyusutan adalah alokasi jumlah suatu aktiva yang dapat disusutkan sepanjang masa manfaat yang diestimasi (PSAK No. 17 tahun 2002). Jumlah aktiva yang dapat disusutkan adalah biaya perolehan

dikurangi nilai residu. Jika nilai residu tidak ada, maka jumlah aktiva yang dapat disusutkan sebesar biaya perolehan (Nisa UI Kholqiah & Nining R ; 2011 : 40).

Aktiva tetap kecuali tanah atau hak atas tanah pada waktu digunakan dalam operasi perusahaan yang dimaksudkan untuk memperoleh laba, kegunaannya akan semakin menurun. Penurunan kegunaan aktiva tetap tersebut mengakibatkan nilainya harus disusutkan. Proses itu dinamakan penyusutan untuk aktiva berwujud yang dapat diganti. Proses penyusutan ini penekanan utamanya adalah pada pengalokasian biaya dari *cost* aktiva tetap ke biaya periode untuk ditandingkan dengan pendapatan yang dilaporkan pada masing-masing periode selama digunakan aktiva tetap tersebut (Arif Rohman & Nining R ; 2010 : 3-4).

Depresiasi atau penyusutan adalah pengalokasian harga perolehan aktiva tetap, seperti kendaraan, gedung, mesin produksi ataupun peralatan selama umur ekonomis aktiva tersebut. Umur ekonomis adalah berapa lama aktiva tersebut bermanfaat bagi perusahaan secara efisien. Jadi, umur ekonomis tidak sama dengan umur aktiva (Anastasia Diana dan Lilis Setiawati ; 2011 : 24).

Menurut Hery dan Widyawati Lekok (2011 : 22), penyusutan bukanlah proses dimana perusahaan mengakumulasikan dana (kas) untuk mengganti aktiva tetapnya, contohnya kendaraan. Penyusutan juga bukan cara menghitung nilai yang berlaku saat ini atas aktiva tetap. Penyusutan adalah alokasi secara periodik dan sistematis dari harga perolehan aktiva (seperti kendaraan, bangunan, dan peralatan) selama periode-periode berbeda yang memperoleh manfaat dari penggunaan aktiva bersangkutan (kendaraan). Akumulasi penyusutan adalah bukan sebuah dana pengganti aktiva, melainkan jumlah harga perolehan aktiva

II.3. Metode Jam Jasa (*Service Hours Method*)

Metode jam jasa (*service hours method*) digunakan untuk mengalokasikan beban penyusutan berdasar jangka waktu penggunaan aktiva yang sebenarnya (Arif Rohman dan Nining R ; 2010 : 5).

Teori yang mendasari metode ini adalah bahwa pembelian suatu aktiva menunjukkan pembelian sejumlah jam jasa langsung. Dalam menghitung besarnya beban penyusutan, metode ini membutuhkan estimasi umur aktiva berupa jumlah jam jasa yang dapat diberikan oleh aktiva bersangkutan. Harga perolehan yang dapat disusutkan (harga perolehan dikurangi dengan nilai estimasi nilai residu) dibagi dengan estimasi total jam jasa, menghasilkan besarnya tarif penyusutan untuk setiap jam pemakaian aktiva.

$$\text{Penyusutan} = \frac{\text{Harga Perolehan} - \text{Nilai Residu}}{\text{Estimasi Total Jam Jasa}}$$

Gambar II.1. Rumus Penyusutan
(Sumber : Hery ; 2014)

Dari rumus diatas dapat diketahui bahwa besarnya penyusutan aset tetap dihitung dengan cara mengurangkan harga perolehan dengan nilai residu, dan membagi hasilnya dengan estimasi jumlah jam pemakaian total dari aset tetap tersebut selama umur ekonomisnya. Besarnya beban penyusutan ini akan berfluktuasi setiap periodenya tergantung pada jumlah kontribusi jam jasa yang diberikan oleh aset bersangkutan.

II.4. Visual Basic 2010

Merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio 2010*. *Visual Studio* merupakan produk pemrograman andalan dari *Microsoft Corporation*, dimana di dalamnya berisi beberapa jenis IDE pemrograman seperti *Visual Basic*, *Visual C++*, *Visual Web Developer*, *Visual C#*, dan *Visual F#*. Semua IDE pemrograman tersebut sudah mendukung penuh implementasi *.Net Framework* terbaru, yaitu *.Net Framework 4.0* yang merupakan pengembangan dari *.Net Framework 3.5*. Adapun *database* standar yang disertakan adalah *Microsoft SQL Server 2008 express* (Wahana Komputer ; 2011 : 2).

II.5. SQL Server 2008

SQL Server 2008 adalah sebuah terobosan baru dari *Microsoft* dalam bidang *database*. *SQL Server* adalah DBMS (*Database Management System*) yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti *IBM* dan *Oracle*. *SQL Server 2008* dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008* membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data.

Microsoft merilis *SQL Server 2008* dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sebagai berikut. Menurut cara pemrosesan data pada prosesor maka *Microsoft* mengelompokkan produk ini berdasarkan 2 jenis yaitu :

1. Versi 32-bit(x86), yang biasanya digunakan untuk komputer dengan single prosesor (Pentium 4) atau lebih tepatnya prosesor 32 bit dan sistem operasi *Windows XP*.
2. Versi 64-bit(x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya *Core 2 Duo*) dan sistem operasi 64 bit seperti *Windows XP 64*, *Vista*, dan *Windows 7*. Sedangkan secara keseluruhan terdapat versi- versi seperti berikut ini :
 - Versi *Compact*, ini adalah versi “Tipis” dari semua versi yang ada. Versi ini seperti versi *desktop* pada *SQL Server 2000*. Versi ini juga digunakan pada *handled device* seperti *Pocket PC*, *PDA*, *SmartPhone*, *Tablet PC*.
 - Versi *Express*, ini adalah versi “Ringan” dari semua versi yang ada (tetapi versi ini berbeda dengan versi *compact*) dan paling cocok untuk latihan para pengembang aplikasi (Wenny Widya dkk; 3).

II.6. Database

Database dapat didefinisikan sebagai kumpulan dari item data (*file* atau tabel) yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan dalam perangkat keras komputer, dan digunakan perangkat lunak untuk memanipulasinya. Sedangkan, kenapa *database* diperlukan, setidaknya alasan berikut dapat menjawabnya.

1. *Database* merupakan salah satu komponen penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.



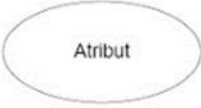

2. Mampu menentukan kualitas informasi, meliputi: akurasi, tepat pada waktunya, dan relevan. Karena informasi dapat dikatakan bernilai apabila manfaatnya lebih besar disbanding dengan biaya yang dikeluarkan.
3. Dapat mereduksi duplikasi data (*data redundancy*) dan mengurangi pemborosan tempat simpanan luar.

Proses perancangan *database* merupakan tahap penting agar sistem yang dikembangkan efisien dalam penggunaan ruang penyimpanan, pengaksesan data, terjaminnya integritas data, dan mudah dalam pemanipulasiannya. Beberapa metode populer yang digunakan untuk melakukan pemodelan data adalah Diagram ER (*Entity Relationship*) dan Normalisasi (Riyanto ; 2014 : 61).

II.7. 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).

Tabel II.1. Simbol ERD

| Notasi | Keterangan |
|---|---|
|  | Entitas , adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai. |
|  | Relasi , menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda. |
|  | Atribut , berfungsi mendeskripsikan karakter entitas (atribut yg berfungsi sebagai key diberi garis bawah) |
|  | Garis , sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut. |

(Sumber : Yuniar Supardi ; 2010 : 448)

II.8. Teknik Normalisasi

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

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

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 kuncinya.

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.9. Unified Modeling Language (UML)

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.

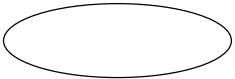
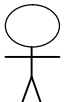

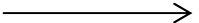

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

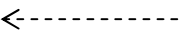
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

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 dapat dilihat pada tabel II.2 berikut ini :

Tabel II.2. Simbol Use Case

| 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> . |
|  | 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 case</i> lain, contohnya adalah pemanggilan sebuah fungsi program. |



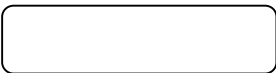
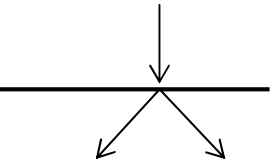
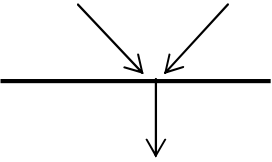
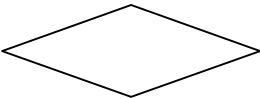
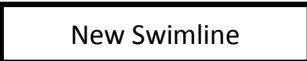
| | |
|---|---|
|  | <i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi. |
|---|---|

(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* dapat dilihat pada tabel II.3 berikut ini :

Tabel II.3. Simbol *Activity Diagram*

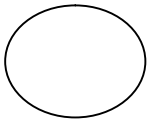
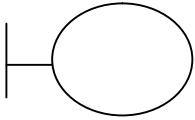
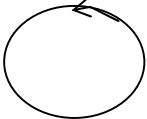

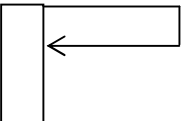


| Gambar | Keterangan |
|---|--|
|  | <i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas. |
|  | <i>End point</i> , akhir aktifitas. |
|  | <i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis. |
|  | <i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan 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 diagram</i> 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* dapat dilihat pada tabel II.4 berikut ini :

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)

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 dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.5 berikut ini :

Tabel II.5. *Multiplicity Class Diagram*

| Multiplicity | 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 maksimum 4 |

(*Sumber : Windu Gata ; 2013 : 8*)