

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem merupakan sekumpulan elemen-elemen yang saling terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan yang telah ditetapkan.

Karakteristik sistem terdiri dari :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem

Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran Sistem

Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya (Sulindawati ; 2010 : 135).

II.2 Sistem Informasi Akuntansi

Sistem informasi akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Lingkup sistem informasi akuntansi dapat dijelaskan dari manfaat yang didapat dari

informasi akuntansi. Manfaat atau tujuan sistem informasi akuntansi tersebut adalah sebagai berikut :

1. Mengamankan harta / kekayaan perusahaan. Harta / kekayaan di sini meliputi kas perusahaan, persediaan barang dagangan, termasuk aset tetap perusahaan.
2. Menghasilkan beragam informasi untuk pengambilan keputusan. Misal, pengelola toko swalayan memerlukan informasi mengenai barang apa saja yang diminati oleh konsumen. Membeli barang yang kurang laku berarti kas akan terjebak dalam persediaan dan berarti kehilangan kesempatan untuk membeli barang dagangan yang laku.
3. Menghasilkan informasi untuk pihak eksternal. Setiap pengelola usaha memiliki kewajiban untuk membayar pajak. Besarnya pajak yang dibayar tergantung pada omset penjualan (jika pengelola memilih menggunakan norma dalam perhitungan pajaknya) atau tergantung pada laba rugi usaha (jika pengelola memilih untuk tidak menggunakan norma dalam perhitungan pajaknya).
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi. Sistem informasi dapat juga dimanfaatkan untuk penilaian kinerja karyawan atau divisi.
5. Menyediakan data masa lalu untuk kepentingan audit (pemeriksaan). Data yang tersimpan dengan baik sangat memudahkan proses audit (pemeriksaan).
6. Menghasilkan informasi untuk penyusunan dan evaluasi anggaran perusahaan. Anggaran merupakan alat yang sering digunakan perusahaan untuk mengendalikan pengeluaran kas.
7. Menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian. Selain berguna untuk membandingkan informasi yang berkaitan

dengan anggaran dan biaya standar dengan kenyataan seperti yang telah dikemukakan (Diana dan Setiawati ; 2011 : 6).

II.3 Penyusutan

Seiring dengan waktu pemakaian sebuah aset tetap, maka pada saat yang sama aset tetap tersebut akan mulai berkurang kemampuannya atau mulai mengalami keusangan (*obsolescence*) untuk menciptakan barang dan jasa. Berkurangnya kemampuan aset tetap ini disebut sebagai penyusutan atau depresiasi (*depreciation*). Jumlah yang dapat disusutkan dialokasikan ke setiap periode akuntansi selama masa manfaat aktiva dengan berbagai metode yang sistematis. Metode manapun yang dipilih, konsistensi dalam penggunaannya adalah perlu, tanpa memandang tingkat profitabilitas perusahaan dan pertimbangan perpajakan, agar dapat menyediakan daya banding hasil operasi perusahaan dari periode ke periode. Aktiva tetap berwujud dapat disusutkan dalam beberapa metode, beberapa jenis metode penyusutan atas aset tetap menurut PSAK 16 yang dapat diterapkan di Indonesia adalah metode penyusutan garis lurus (*straight line method*), saldo menurun ganda (*double declining balance method*), dan metode unit produksi (*units of production method*). Serta tambahan metode penyusutan lainnya yaitu penyusutan berdasarkan jumlah angka tahun (*sum of the years digits method*) (Mairuhu dan Tinangon ; 2014 : 404).

II.4 Perhitungan Beban Penyusutan

Perhitungan terhadap beban penyusutan aset tetap, dapat menggunakan metode penyusutan yang sesuai dengan standar akuntansi keuangan (SAK) maupun peraturan perpajakan. Metode penyusutan menurut SAK digunakan untuk menilai kinerja perusahaan dan keadaan finansialnya. Metode penyusutan berdasarkan peraturan

perpajakan digunakan untuk kepentingan pajak. Adanya perbedaan pengakuan beban penyusutan, akan mengakibatkan terjadinya koreksi fiskal. Tujuan penelitian ini untuk mengetahui penerapan metode penyusutan yang digunakan dalam laporan keuangan perusahaan, baik menurut SAK maupun ketentuan perpajakan dan pengaruh terhadap perbedaan perhitungannya. Metode analisis yang digunakan metode deskriptif. Hasil penelitian menunjukkan beban penyusutan aset tetap yang dihitung perusahaan dengan metode garis lurus setiap bulannya telah dilakukan dengan baik, namun belum konsisten. Tidak adanya kegiatan operasional dan kerugian yang dialami, membuat perusahaan tidak menghitung beban penyusutan sesuai prinsip yang berlaku. Adanya perbedaan pengakuan beban penyusutan menurut standar akuntansi keuangan dan perpajakan, maka ditemukan koreksi fiskal negatif yang mengakibatkan adanya penambahan biaya yang telah diakui dalam laporan laba-rugi komersial. Sebaiknya pimpinan perusahaan menggunakan metode penyusutan yang sesuai dengan kondisi perusahaan, misalnya metode jam jasa atau metode saldo menurun (Ajeng Citralarasati Mardjani ; 2015).

Setiap perusahaan, baik bergerak dalam bidang jasa, perdagangan maupun industri pasti memiliki aktiva tetap untuk menjalankan kegiatan operasional setiap harinya. Aktiva tetap merupakan harta perusahaan yang masa penggunaannya lebih dari satu periode normal akuntansi (biasanya di atas satu tahun penggunaan) dan menjadi subjek manajemen dengan mempertimbangkan kualitas dengan cara pemakaiannya, demikian juga dengan penyusutan (depresiasi). Semakin bertambahnya jumlah aktiva tetap perusahaan menuntut adanya sistem penghitungan penyusutan yang baik. Sistem penghitungan yang sedang berjalan saat ini masih menggunakan Ms. Excel. Dengan

adanya sistem penghitungan penyusutan aktiva tetap diharapkan akan mempercepat penghitungan penyusutan aktiva tetap. Karena dalam sistem yang diajukan sudah disediakan database untuk keakuratan data aktiva tetap. Sistem ini menggunakan Data Flow Diagram (DFD) sebagai metode analisisnya dan dibangun dengan menggunakan bahasa pemrograman Visual Basic 6.0 dan Ms. Access sebagai Database Management System (DBMS) (Daemi ; 2012).

II.5 Metode Hasil Produksi

Beban penyusutan aset (aktiva) dengan metode Satuan Hasil Produksi (*productive output method*) dari tahun ke tahun tidak selalu sama karena tergantung dari produk yang dihasilkan untuk suatu periode. Perhitungan penyusutan aset dengan metode ini didasarkan pada anggapan bahwa aset (terutama mesin) akan cepat rusak jika dipakai *full time* daripada *part time*. Dalam penggunaan metode ini, umur kegunaan aset ditaksir dengan jumlah unit produk yang dihasilkan selama masa manfaat (umur ekonomis). Langkah untuk menghitung penyusutan metode Satuan Hasil Produksi adalah sebagai berikut :

- a. Tentukan penyusutan aset per unit produk.
- b. Distribusikan taksiran produksi selama umur (masa) manfaat aset.
- c. Beban penyusutan periode bersangkutan dihitung berdasarkan perkalian penyusutan aset per unit produk dengan jumlah produksi yang dihasilkan dalam suatu periode akuntansi.

Besarnya beban penyusutan aset tetap dihitung dengan cara mengurangi taksiran nilai residu dari harga perolehannya, dan membagi hasilnya dengan taksiran jumlah

produk yang akan dihasilkan dari aset tetap tersebut selama umur ekonomisnya. Dari hasil pembagian tersebut akan diketahui beban penyusutan per unit produk. Jumlahnya lalu dijadikan dasar untuk mengalikan dengan jumlah unit produk yang dihasilkan secara aktual selama suatu periode, sehingga diketahui beban penyusutan aset tetap pada suatu periode.

Rumus untuk menghitung beban penyusutan per unit produk sebagai berikut :

Beban penyusutan = (Harga Perolehan – Nilai Sisa) / Taksiran Hasil Produksi (Rudianto ; 2012 : 263).

II.6 Visual Basic

VB.NET adalah salah satu bahasa pemrograman tingkat tinggi yang mendekati bahasa manusia. Kemunculan bahasa VB.NET ini sebagai jawaban untuk menyederhanakan bahasa pemrograman pada platform .NET yang diluncurkan tahun 2002 dan untuk menjembatani programmer Visual Basic. Bahasa VB.NET secara teknis mengadopsi sintak bahasa Visual Basic. Konsistensi API membuat bahasa VB.NET menjadi pilihan dalam membuat kode program diatas platform Windows.

Fitur baru bahasa VB.NET dibandingkan Visual Basic bahwa bahasa VB.NET mendukung object-oriented dan juga dynamics programming. Ini menambah daftar kemudahan untuk belajar bahasa VB.NET (Kurniawan ; 2013 : 10).

II.7 SQLServer 2008

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang didevelop oleh Microsoft, yang digunakan untuk menyimpan dan

mengolah data. Pada SQL Server 2008, kita bisa melakukan pengambilan dan modifikasi data yang ada dengan cepat dan efisien. Pada SQL Server 2008, bisa membuat *object* yang sering digunakan pada aplikasi bisnis, seperti membuat *database*, *table*, *function*, *stored procedure*, *trigger* dan *view*. Selain *object*, berfungsi juga menjalankan perintah SQL (*Structured Query Language*) untuk mengambil data (Solution dan Community ; 2010 : 101)

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.

II.8.1 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.

5. 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.

6. 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*) (Simarmata dan Paryudi ; 2010 : 76).

II.9 UML (*Unified Modeling Language*)

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

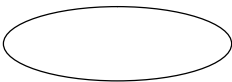
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

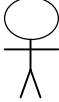

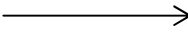
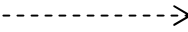
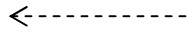
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>



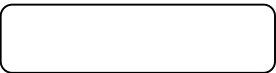
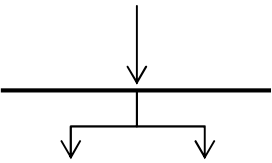
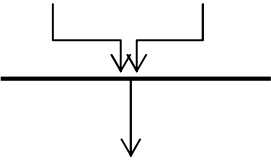
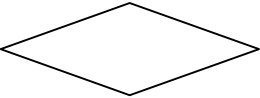

	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. 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 mengindikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan 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 : Gata ; 2013 : 4)

2. Diagram Aktivitas (*Activity Diagram*)

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

Tabel II.2. Simbol *Activity Diagram*

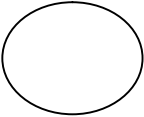
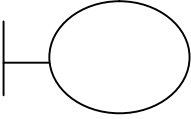
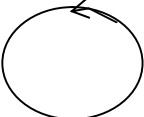
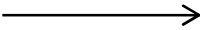
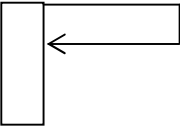


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
 New Swimline	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : 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.3. 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 : 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.4. 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 : Gata ; 2013 : 9)