

BAB II

TINJAUAN PUSTAKA

II.1. Akuntansi

Akuntansi adalah sistem informasi yang menghasilkan informasi keuangan kepada pihak-pihak yang berkepentingan mengenai aktivitas ekonomi dan kondisi suatu perusahaan. Dalam proses menghasilkan informasi yang dibutuhkan oleh berbagai pihak yang berkepentingan, akuntansi harus melewati beberapa tahapan proses. Proses tersebut dimulai dari mengumpulkan dokumen dasar transaksi, mengklasifikasikan jenis transaksi, menganalisis, meringkasnya dalam catatan, hingga melaporkannya dalam bentuk laporan keuangan yang dibutuhkan (Rudianto ; 2012 : 16).

II.1.1 Siklus Akuntansi

Adapun beberapa siklus akuntansi, yaitu :

- a. Transaksi adalah peristiwa bisnis yang dapat diukur dengan menggunakan satuan moneter dan yang menyebabkan perubahan disalah satu unsur posisi keuangan perusahaan. Umumnya, transaksi selalu disertai dengan perpindahan hak milik dari pihak-pihak yang melakukan transaksi tersebut.
- b. Dokumen dasar adalah beberapa formulir yang menjadi bukti telah terjadinya transaksi tertentu. Berbagai formulir yang biasanya menjadi dokumen dasar

antara lain : faktur, kwitansi, nota penjualan dan lain-lain. Dokumen dasar merupakan titik tolak dilakukannya proses akuntansi dalam perusahaan. Tanpa dokumen dasar, tidak bisa dilakukan pencatatan dalam akuntansi.

- c. Jurnal (*journal*) adalah buku yang digunakan untuk mencatat transaksi perusahaan secara kronologis, sedangkan menjurnal adalah aktivitas meringkas dan mencatat transaksi perusahaan di buku jurnal dengan menggunakan urutan tertentu berdasarkan dokumen dasar yang dimiliki.
- d. Akun (*Account*) adalah kelas informasi dalam sistem akuntansi. Atau media yang digunakan untuk mencatat informasi sumber daya perusahaan dan informasi lainnya berdasarkan jenisnya.
- e. Posting adalah aktivitas memindahkan catatan di buku jurnal kedalam buku besar sesuai dengan jenis transaksi dan nama akun masing-masing.
- f. Buku Besar (*General Ledger*) adalah kumpulan dari semua akun yang dimiliki perusahaan beserta saldonya. Seluruh akun yang dimiliki perusahaan saling berhubungan satu dengan lainnya dan merupakan satu kesatuan (Rudianto ; 2012 : 16).

II.1.2. Laporan Keuangan

Pada akhir siklus akuntansi, akuntan perusahaan harus membuat laporan keuangan untuk berbagai pihak yang membutuhkan. Sesuai dengan standar akuntansi keuangan yang berlaku di indonesia, laporan keuangan terdiri dari :

1. Laporan laba rugi yaitu laporan yang menunjukkan kemampuan perusahaan dalam menghasilkan laba selama suatu periode akuntansi atau satu tahun. Secara umum, laporan laba rugi terdiri dari unsur pendapatan dan unsur beban usaha. Pendapatan usaha dikurangi dengan beban usaha akan menghasilkan laba usaha.
2. Laporan arus kas (*Statement of Cash Flow*) adalah laporan yang menunjukkan aliran uang yang diterima dan yang digunakan perusahaan selama satu periode akuntansi, beserta sumber-sumbernya (Rudianto ; 2012 : 19).

II.1.3. Terminologi Dalam Akuntansi

Akuntansi sering menggunakan terminologi khusus yang biasa dipakai di bidang akuntansi. Untuk mempelajari akuntansi pada tahap awal, pertama kita harus mengenal beberapa istilah yang biasa dipakai di bidang akuntansi. Beberapa istilah dasar yang biasa dipakai di bidang akuntansi antara lain :

1. Kas (*Cash*) adalah alat pembayaran yang dimiliki perusahaan dan siap digunakan, seperti cek kontan serta uang tunai (uang kertas atau uang logam).
2. Piutang Usaha (*Account Reveivable*) adalah tagihan perusahaan kepada pihak lain akibat transaksi yang terjadi di masa lalu, seperti tagihan atas penjualan dan tagihan kepada pegawai atas pinjamannya ke perusahaan.

3. Utang Usaha (*Account Payable*) adalah kewajiban untuk membayar sejumlah uang atau barang atau jasa kepada pihak lain yang timbul akibat transaksi yang dilakukan perusahaan dimasa lalu.
4. Modal Saham (*Share Capital*) adalah kontribusi pemilik pada perusahaan yang berbentuk perseroan terbatas, sekaligus menunjukkan bukti kepemilikan dan hak pemilik atas perseroan terbatas tersebut. bukti kepemilikan ini dibagi dalam bentuk lembaran saham perusahaan bersangkutan.
5. Laba Ditahan (*Retained Earning*) adalah akumulasi laba usaha yang diperoleh perusahaan selama beberapa tahun dari sebuah badan usaha berbentuk perseroan terbatas yang tidak dibagikan kepada pemegang saham.
6. Dividen (*Dividends*) adalah bagian dari laba usaha perseroan terbatas yang dibagikan kepada pemilik perusahaan (pemegang saham) sebagai imbalan atas setoran modal pemilik.
7. Pendapatan (*Revenues*) adalah kenaikan modal perusahaan akibat penjualan produk perusahaan. Istilah pendapatan biasanya digunakan oleh perusahaan jasa, sedangkan perusahaan dagang atau perusahaan manufaktur lebih banyak menggunakan istilah penjualan (*Sales*) untuk menampung transaksi yang sama.
8. Biaya (*Cost*) adalah pengorbanan ekonomis untuk memperoleh barang atau jasa, dimana manfaat barang atau jasa tersebut dinikmati dalam waktu lebih dari satu tahun (jangka panjang).

9. Beban (*Expenses*) adalah pengorbanan ekonomis untuk memperoleh barang atau jasa yang manfaatnya dinikmati hanya dalam waktu satu tahun atau satu periode akuntansi saja (Rudianto ; 2012 : 27).

II.1.4. Akun-akun perusahaan dagang

Berbeda dengan perusahaan jasa yang menjual produk yang bersifat nonfisik, perusahaan dagang membeli dan menjual produk yang memiliki bentuk fisik . karena itu, terdapat beberapa aktivitas dan beberapa akun dalam perusahaan dagang yang tidak dimiliki perusahaan jasa. Akun-akun tersebut antara lain :

1. Akun Pembelian, yaitu akun yang hanya digunakan untuk menampung aktivitas pembelian barang dagang perusahaan. Pembelian aset selain persediaan barang dagang (seperti peralatan kantor, pembelian perlengkapan kantor, dan lain-lain) tidak ditampung dalam akun ini.
2. Akun Penjualan, yaitu akun yang hanya digunakan untuk menampung penjualan barang dagang perusahaan. Aktivitas penjualan aset (seperti penjualan aset tetap) tidak ditampung dalam akun ini.
3. Akun Persediaan, yaitu akun yang digunakan untuk menunjukkan jumlah barang dagang yang dimiliki perusahaan pada awal atau akhir periode akuntansi tertentu (Rudianto ; 2012 : 114).

II.2. PHP

PHP adalah singkatan dari "*PHP Hypertext Processor*", yang merupakan sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaks mirip dengan bahasa C, Java dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat. PHP merupakan bahasa server-side yang cukup handal, yang akan disatukan dengan HTML (Hypertext Markup Language) dan berada di server. Artinya, sintaks dan perintah yang diberikan akan sepenuhnya dijalankan di server sebelum dikirim ke komputer klien. Pada awal tahun 1995, Rasmus Ledorf membuat produk bernama PHP/FI PHP ini ditulis dengan menggunakan bahasa C, dan memiliki kemampuan untuk berkomunikasi dengan database serta membuat halaman dinamis (Achmad Nazrul ; 2013).

II.3. Database

Database atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer. Data itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. Database sering digunakan untuk melakukan proses terhadap data-data tersebut untuk menghasilkan informasi tertentu. Misalnya dari data nama siswa dan tanggal lahir siswa anda bisa mendapatkan informasi nama siswa yang berulang tahun pada hari ini. tentu saja informasi tersebut akan anda dapatkan dari software pemroses database dengan cara anda memberikan perintah

dalam bahasa tertentu yaitu SQL (Structured Query Language) (Wahana Komputer ; 2010 : 24).

II.4. SQL Server

SQL Server adalah sebuah terobosan baru dari *microsoft* dalam bidang *database*. SQL Server adalah sebuah DBMS (*Database Management System*) yang dibuat oleh *microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan *Oracle*. SQL Server dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data (Wahana Komputer ; 2010 : 2).

II.5. Entity Relationship Diagram (ERD)

Pada dasarnya ERD (*Entity Relationship Diagram*) adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD di atas. ERD digunakan untuk melakukan pemodelan terhadap struktur data dan hubungannya. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah *database* yang baik.

Entity dapat berarti sebuah objek yang dapat dibedakan dengan objek lainnya. Objek tersebut dapat memiliki komponen-komponen data (atribut atau *field*) yang membuatnya dapat dibedakan dari objek yang lain. Dalam dunia *database* *entity* memiliki atribut yang menjelaskan karakteristik dari *entity*

tersebut. Ada dua macam atribut yang dikenal dalam *entity* yaitu atribut berperan sebagai kunci primer dan atribut deskriptif. Hal ini berarti setiap *entity* memiliki himpunan yang diperlukan sebuah *primary key* untuk membedakan anggota-anggota dalam himpunan tersebut (Wahana Komputer ; 2010 : 30).

Ada beberapa derajat relasi yang dapat terjadi, yaitu :

1. *One to one*, menggambarkan bahwa antara 1 anggota *entity* A hanya dapat berhubungan dengan 1 anggota *entity* B. Biasanya derajat relasi ini digambarkan dengan simbol 1-1.
2. *One to many*, menggambarkan bahwa 1 anggota *entity* A dapat memiliki hubungan dengan lebih dari 1 anggota *entity* B. Biasanya derajat relasi ini digambarkan dengan simbol 1-N.
3. *Many to many*, menggambarkan bahwa lebih dari satu anggota A dapat memiliki hubungan lebih dari satu anggota *entity* B. Simbol yang digunakan adalah N-N (Wahana Komputer ; 2010 : 31).

II.6. Normalisasi

Normalisasi adalah suatu proses yang digunakan untuk menentukan pengelompokan atribut-atribut dalam sebuah relasi sehingga diperoleh relasi yang berstruktur baik. Dalam hal ini yang dimaksud dengan relasi yang berstruktur baik adalah relasi yang memenuhi dua kondisi berikut :

1. Mengandung redundansi sesedikit mungkin, dan

2. Memungkinkan baris-baris dalam relasi disisipkan, dimodifikasi dan dihapus tanpa menimbulkan kesalahan atau ketidakkonsistenan.

Normalisasi sendiri dilakukan melalui sejumlah langkah. Setiap langkah berhubungan dengan bentuk normal (normal form) tertentu. Dalam hal ini yang disebut bentuk normal adalah suatu keadaan relasi yang dihasilkan oleh penerapan aturan-aturan sederhana yang berhubungan dengan dependensi fungsional terhadap relasi tersebut. Sekedar perlu diketahui terlebih dulu bahwa bentuk normalisasi dapat berupa :

1. Bentuk normal pertama (1NF / First normal form)

Yang disebut dengan bentuk normal pertama (1NF) adalah suatu keadaan yang membuat setiap perpotongan baris dan kolom dalam relasi hanya berisi satu nilai. Untuk membentuk relasi agar berada dalam bentuk normal pertama.

2. Bentuk normal kedua (2NF / Second normal form)

Bentuk normal kedua adalah suatu bentuk yang mensyaratkan bahwa relasi harus sudah berada dalam bentuk normal pertama dan tidak mengandung dependensi parsial.

3. Bentuk normal ketiga (3NF / Third normal form)

Bentuk normal ketiga adalah suatu keadaan yang menyaratkan bahwa relasi harus sudah berada dalam bentuk normal kedua dan tidak mengandung dependensi transitif.

4. Bentuk normal Boyce-Codd (BCNF / Boyce-Codd normal form)

Bentuk normal Boyce-Codd adalah suatu keadaan yang menyaratkan bahwa setiap determinan (penentu) dalam suatu relasi berkedudukan sebagai kunci kandidat. Oleh karena itu untuk mengetahui suatu relasi memenuhi BCNF atau tidak perlu dilakukan identifikasi guna memastikan bahwa semua determinan dalam relasi berkedudukan sebagai kunci kandidat.

5. Bentuk normal keempat (4NF / Fourth normal form)

Bentuk normal keempat adalah suatu keadaan yang menyaratkan relasi berada pada BCNF dan tidak mengandung lebih dari satu dependensi bernilai banyak yang bersifat independen.

6. Bentuk normal kelima (5NF / Fifth normal form)

Bentuk normal kelima adalah suatu keadaan yang membuat relasi yang telah memenuhi bentuk normal keempat tidak dapat didekomposisi menjadi relasi-relasi yang lebih kecil dengan kunci kandidat relasi-relasi pecahannya tersebut tidak sama dengan kunci kandidat relasi (Abdul Kadir ; 2009 : 144).

II.7. *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

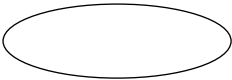
UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram (Sumber :Yuni Sugiarti ; 2013 : 34).

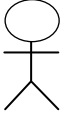


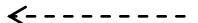

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use Case Diagram*

Use case diagram atau diagram *use case* merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem 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 data-bbox="336 1809 448 1839"><i>Use case</i></p> 	<p data-bbox="639 1809 1370 1989">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di</p>

	awal <i>frase</i> nama <i>use case</i> .
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal <i>frase</i> nama aktor.
Asosiasi 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Asosiasi 	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidentifikasi bila aktor berinteraksi secara pasif dengan sistem.
<i>Extend</i> 	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.
<i>Include</i> 	<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.

(Sumber : Yuni Sugiarti ; 2013 : 42)

2. Class Diagram (Diagram Kelas)

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. Diagram kelas menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain.

Tabel II.2. Multiplicity Class Diagram

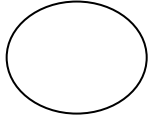
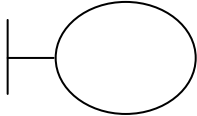
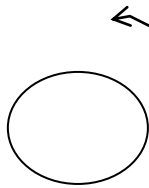

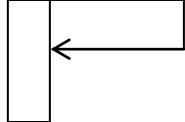


<i>Multiplicity</i>	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber :Yuni Sugiarti ; 2013 : 59)

3. *Sequence Diagram* (Diagram Urutan)

Sequence diagram menggambarkan kelakuan/perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.3. Simbol *Sequence Diagram*




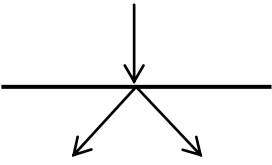
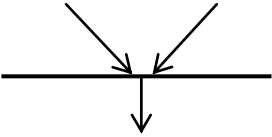
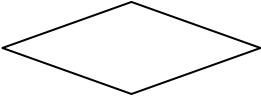
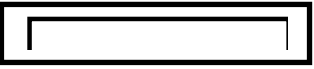
Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara 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 : Rosa A.S ; 2013 : 165)

4. Activity Diagram (Diagram Aktivitas)

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

Tabel II.4. 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 diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Rosa A.S ; 2013 : 162)