

BAB II

TINJAUAN PUSTAKA

II.1. Aktiva Tetap

Aktiva tetap sebagai aset berwujud yang dimiliki untuk digunakan dalam produksi atau penyediaan barang dan jasa, untuk disewakan kepada pihak lain, atau untuk tujuan administratif, dan diharapkan untuk digunakan selama lebih dari satu periode. Aset tetap adalah aset yang digunakan perusahaan sebagai entitas untuk menciptakan pendapatan. Aset tetap berasal dari aktivitas investasi perusahaan. Aset tetap merupakan aset yang memiliki wujud fisik dan memberikan manfaat ekonomis kepada entitas bisnis selama lebih dari satu periode akuntansi pada masa-masa yang akan datang. Aset tetap dapat diperoleh dengan cara pembelian, pembangunan kembali atau pertukaran dengan aset tetap atau aset jenis lainnya yang dimiliki oleh perusahaan lain. (Mirawati Florce Sihombing; 2016).

II.2. Penyusutan

Penyusutan adalah alokasi secara periodik dan sistematis dari harga perolehan aset selama periode-periode berbeda yang memperoleh manfaat dari penggunaan aset bersangkutan, dalam bukunya berjudul pengantar akuntansi-adaptasi indonesia menyatakan bahwa penyusutan atau depresiasi adalah pemindahan biaya ke beban secara berkala. Faktor-faktor yang mempengaruhi beban penyusutan adalah :

1. Nilai Perolehan Aset (*asset cost*), mencakup seluruh pengeluaran yang terkait dengan perolehannya dan persiapan sampai aset dapat digunakan.
2. Nilai Residu/Nilai Sisa (*residual or salvage value*), merupakan estimasi nilai realisasi pada saat aset tidak dipakai lagi.
3. Umur Ekonomis (*economic life*), dapat diartikan sebagai suatu periode atau umur fisik dimana perusahaan dapat dimanfaatkan aset tetapnya.
4. Pola Pemakaian (*pattern of use*), pola pemakaian ini seringkali diabaikan dalam menghitung besarnya beban penyusutan periodek mengingat sulitnya dalam mengidentifikasi pola pemakaian dimaksud, (Mirawati Florce Sihombing; 2016).

II.3. Metode Penyusutan Aktiva Tetap

Beberapa metode penyusutan aktiva tetap antara lain, Metode garis lurus (*straight-line*), metode saldo menurun ganda (*double declining method*), metode jumlah angka tahun (*sum of the years digits method*), metode jam kerja (*service hours method*), metode unit produksi (*productive output method*). Penyusutan metode garis lurus adalah bahwa aset yang bersangkutan akan memberikan manfaat yang sama untuk setiap periodenya sepanjang umur aset dan pembebanannya tidak dipengaruhi oleh perubahan produktivitas maupun efisiensi aset. Penyusutan dengan menggunakan metode jam kerja, metode ini adalah bahwa pembelian suatu aset menunjukkan pembelian sejumlah jam jasa langsung, (Mirawati Florce Sihombing; 2016). Perhitungan metode jam kerja sebagai berikut:

Tarif Penyusutan = (Harga Perolehan – Estimasi Nilai Residu) /Estimasi Total
Jam Jasa

Sedangkan untuk menghitung besaran penyusutan per periode adalah sebagai berikut caranya:

Penyusutan = (Jumlah jam kerja periode x tarif penyusutan) / jam, (Mirawati Florce Sihombing; 2016)..

II.4. *Hypertext Markup Language (HTML)*

Hypertext Markup Language (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web internet dan *formatting hypertext* sederhana yang ditulis kedalam format ASCII agar dapat menghasilkan tampilan wujud yang terintergrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan kedalam format ASCII normal sehingga menjadi *home page* dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standart Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium (W3c)*, HTML dibuat oleh kolaborasi *Caillau TIM* dengan *Berners lee robert* ketika mereka bekerja di CERN pada tahun 1989 CERN adalah lembaga penelitian fisika energi tinggi di Jenewa. (Harison; 2016).

II.5. PHP

PHP adalah singkatan dari "PHP: Hypertext Preprocessor ", yang merupakan sebuah bahasa scripting yang terpasang pada HyperText Markup Language (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. (Nursahid; 2015).

II.6. Database

Basis data atau database adalah kumpulan data terstruktur. Agar dapat menambahkan, mengakses, dan memproses data yang tersimpan dalam database komputer, dibutuhkan sistem manajemen basis data (database management system). Dalam pengembangan perangkat lunak tradisional yang memanfaatkan pemrosesan file, setiap kelompok pengguna menyimpan file-file-nya sendiri untuk menangani aplikasi pengolahan datanya masing-masing. Hal ini mengakibatkan adanya kerangkapan data atau disebut dengan redundancy. Redundansi dalam proses penyimpanan data yang terjadi berkali-kali dapat mengakibatkan beberapa masalah. Pertama, ada kebutuhan untuk melakukan pembaruan logis tunggal, misalnya seperti memasukkan data pada siswa baru beberapa kali: satu kali untuk setiap file tempat data siswa direkam. Hal ini menyebabkan duplikasi data. Kedua, ruang penyimpanan terbuang ketika data yang sama disimpan berulang kali, dan masalah ini mungkin serius untuk database yang besar. Ketiga, file yang mewakili data yang sama mungkin menjadi tidak konsisten. Hal ini bisa terjadi karena

update diaplikasikan pada beberapa file tapi tidak untuk file yang lain, (Cosmas Eko Suharyanto; 2017).

II.7. SQL

SQL atau disebut juga dengan SEQUEL (*Structured English Query Language*) merupakan bahasa pemrograman yang memiliki tujuan khusus dan dirancang untuk mengelola data dalam sistem manajemen *database* relasional (RDBMS-*Relational Database Management Systems*), atau untuk pengolahan aliran data dalam sistem manajemen basis data relasional. SQL memiliki tiga bagian utama yaitu bahasa pemrograman untuk mendefinisikan data (*Data Defenition Language-DDL*), untuk manipulasi dan akses data (*Data ManipulationLanguage-DML*) dan bagian yang digunakan untuk pengawasan/kontrol pemakai (*Data Control Language*). Bahasa SQL masih memiliki beberapa kekurangan diantaranya adalah ketiga bahasa tersebut harus terintegrasi menggunakan suatu bahasa pemrograman tertentu. Bahasa SQL atau SEQUEL dibangun atas dasar *Relational Algebra*. Secara luas pemakainya telah distandarkan dalam sebuah kerangka kerja yang terdaftar dalam *International Organization For Standardization* (ISO) pada tahun 1987, (Eko Darmanto; 2015).

II.8. My SQL

Salah satu contoh database management system adalah MySQL. MySQL adalah database open source terpopuler di dunia. Dengan kinerja, kehandalan dan kemudahan penggunaan yang terbukti, MySQL telah menjadi pilihan database terdepan untuk aplikasi berbasis web, yang digunakan oleh properti web profil

tinggi termasuk Facebook, Twitter, YouTube, Yahoo! dan banyak lagi [9]. Kata “SQL” dari “MySQL” adalah singkatan dari “Structured Query Language”. SQL adalah bahasa standar yang paling umum digunakan untuk mengakses database. Bergantung pada lingkungan pemrograman yang kita pakai, kita bisa memasukkan SQL secara langsung (misalnya, untuk men-generate laporan), memasukkan pernyataan SQL ke dalam kode yang ditulis dalam bahasa lain, atau menggunakan API khusus yang dapat menyembunyikan sintaks SQL, (Cosmas Eko Suharyanto; 2017).

II.9. UML (*Unified Modeling Language*)

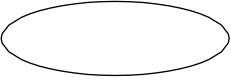
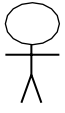


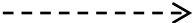

Unified Modeling Language (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem, (Ade Hendini; 2016).

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* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case* Diagram yaitu :

Tabel II.1. Tabel Simbol *Use Case Diagram*




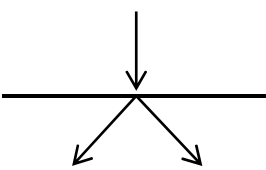
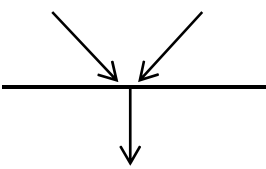
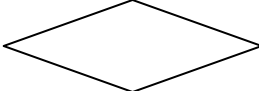

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang 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 : Ade Hendini; 2016)

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. Tabel Simbol *Activity Diagram*

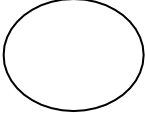
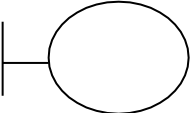


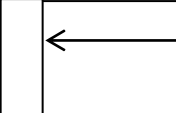


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/ kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , 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 : Ade Hendini; 2016)

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. Tabel 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 <i>formentry</i> 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 : Ade Hendini; 2016)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam 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 *cardinality*.

Tabel II.4. Tabel *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 : Ade Hendini; 2016)