

BAB III

LANDASAN TEORI

Landasan teori merupakan penjelasan dari teori-teori yang relevan dengan Rancang Bangun Aplikasi Inventarisasi Aset Tetap pada PT. Indoberka Investama. Di bagian ini terdapat teori yang mendukung dalam analisa, perancangan, dan pembuatan aplikasi.

3.1 Inventarisasi Aset

Menurut Hambudi (2015), inventarisasi aset adalah serangkaian kegiatan untuk melakukan pendataan, pencatatan, pelaporan hasil pendataan aset, dan mendokumentasikannya, baik aset berwujud maupun aset tidak berwujud pada suatu waktu tertentu. Beberapa tujuan yang ingin dicapai dari kegiatan inventarisasi aset adalah sebagai berikut :

1. Menciptakan tertib administrasi.
2. Pengamanan dan perlindungan aset.
3. Memberikan data dan informasi dalam rangka memudahkan pengawasan dan pengendalian barang.
4. Sebagai data dan informasi untuk dijadikan bahan atau pedoman dalam penyaluran barang.
5. Sebagai data dan informasi dalam menentukan keadaan barang (barang yang rusak atau tua) yang dijadikan dasar untuk menetapkan penjualan, penghapusan, atau pemusnahan aset.

Pencatatan inventarisasi di beberapa perusahaan ada yang menjadi tanggung jawab bagian *finance accounting* dengan alasan bahwa semua informasi barang berupa transaksi pembelian, hingga nilai penyusutan aset merupakan tanggung jawab *finance accounting*. Namun, beberapa perusahaan lainnya ada yang membagi tanggung jawab dengan bagian umum (*general affair*), dikarenakan sebagian aktivitas seperti penempatan, pemindahan, pencatatan, hingga pengawasan merupakan hal yang harus diketahui dan menjadi tanggung jawab bagian umum. Informasi-informasi yang penting dalam kegiatan inventarisasi aset dapat meliputi : jenis barang, lokasi, usia barang, pihak yang menggunakan, masih berfungsi atau tidak, dan proses pengawasannya.

Ketentuan-ketentuan umum dalam proses pelaksanaan inventarisasi aset adalah sebagai berikut :

1. Barang-barang inventaris harus diberi tanda dengan menggunakan kode-kode barang sesuai dengan petunjuk yang terdapat dalam panduan administrasi barang.
2. Memberi koding pada barang-barang yang diinventarisasikan, misalnya dengan menggunakan *barcode system*. Pemberian koding akan sangat berguna dalam proses administrasi dan melakukan *stock opname* atau inventarisasi barang.
3. Melakukan audit, *stock opname*, atau inventarisasi barang secara berkala (tiga atau enam bulan sekali). Hal-hal yang harus diperhatikan adalah sebagai berikut :
 - a) Membuat laporan jika terjadi mutasi barang, kehilangan, atau ganti kepemilikan.

- b) Melakukan saran untuk penyimpanan gudang dan peletakan sesuai fungsinya.
- c) Melakukan saran untuk barang yang tidak layak pakai, nilai buku telah habis, atau barang secara fungsi sudah tidak digunakan.

3.2 Aset Tetap

Aset tetap adalah aset berwujud milik perusahaan yang :

- a) Dimiliki untuk digunakan dalam produksi atau penyediaan barang atau jasa, untuk direntalkan kepada pihak lain, atau untuk tujuan administratif; dan
- b) Diperkirakan untuk digunakan selama lebih dari satu periode.

Aset tetap sendiri dapat diklasifikasikan menjadi dua kategori yaitu aset tetap berwujud dan aset tetap tidak berwujud (Ikatan Akuntan Indonesia, 2014).

Aset tetap berwujud adalah aset-aset yang berwujud yang digunakan dalam kegiatan perusahaan yang normal dan sifatnya relatif permanen, atau secara akuntansi dapat dikatakan memiliki umur lebih dari satu periode akuntansi. Aset tetap berwujud yang dimiliki oleh suatu perusahaan dapat mempunyai bermacam-macam bentuk seperti tanah, bangunan, mesin-mesin dan alat-alat, kendaraan, mebelair, dan lain-lain (Baridwan, 2008).

Sedangkan aset tetap tidak berwujud adalah aset yang tidak dapat diamati secara langsung. Bukti adanya aset ini terdapat di dalam bentuk perjanjian, kontrak, atau kadang-kadang paten, tetapi aset ini sendiri tidak mempunyai wujud nyata. Aset yang termasuk kategori aset tidak berwujud diantaranya : paten, hak

cipta, hak monopoli (*franchise*), cap dan merek dagang (*trademark*), biaya pendirian, biaya pengembangan *software*, dan *goodwill* (Smith & Skousen, 2005).

3.3 Depresiasi Aset Tetap

Depresiasi adalah proses pengalokasian biaya perolehan aset tetap menjadi beban selama masa manfaatnya dengan cara yang rasional dan sistematis. Pengalokasian biaya perolehan diperlukan agar dapat dilakukan penandingan yang tepat antara pendapatan dengan beban, sebagaimana diminta oleh prinsip penandingan. Depresiasi didefinisikan sebagai proses pengalokasian biaya perolehan, bukan proses penilaian aset. (Jusup, 2011)

3.3.1 Metode-metode Perhitungan Depresiasi Aset Tetap

Berbagai metode depresiasi (penyusutan) dapat digunakan untuk mengalokasikan jumlah yang disusutkan secara sistematis dari suatu aset selama umur manfaatnya. Metode tersebut antara lain metode garis lurus (*straight line method*), metode saldo menurun (*diminishing balance method*), dan metode jumlah unit (*sum of the unit method*).

Metode garis lurus menghasilkan pembebanan yang tetap selama umur manfaat aset jika nilai residunya tidak berubah (Ikatan Akuntan Indonesia, 2014). Penentuan beban penyusutan dengan metode garis lurus dapat digunakan dengan rumus berikut :

$$\text{Beban Penyusutan} = \frac{\text{Biaya Perolehan} - \text{Nilai residu}}{\text{Umur Manfaat}}$$

Metode saldo menurun menghasilkan pembebanan yang menurun selama umur manfaat aset. Penentuan tarif penyusutan dengan metode saldo menurun dapat dilakukan dengan menggunakan persamaan berikut (Yulianti, 2014) :

$$\text{Beban Penyusutan} = \text{Tarif Penyusutan} \times \text{Nilai Tercatat}$$

Metode jumlah unit menghasilkan pembebanan berdasarkan pada penggunaan atau output yang diharapkan dari suatu aset. Persamaan yang digunakan untuk menghitung tarif penyusutan berdasarkan metode jumlah unit adalah sebagai berikut :

$$\text{Beban Penyusutan} = \frac{\text{Biaya Perolehan} - \text{Nilai residu}}{\text{Total Jumlah Produksi}}$$

Selain itu, untuk menghitung depresiasi dapat juga menggunakan metode lainnya yaitu penyusutan berdasarkan jumlah angka tahun (*sum of the years digits method*) (Mairuhu & Tinangon, 2014). Menurut Arifin (2009), metode jumlah angka tahun menghasilkan biaya penyusutan atau depresiasi pada tahun pertama lebih besar daripada tahun-tahun berikutnya. Dengan kata lain, beban penyusutan aset dari tahun ke tahun semakin menurun. Hal ini didasarkan pada anggapan yang menyatakan bahwa aset yang baru lebih efisien jika dibanding dengan aset yang dipakai lebih dulu. Penyusutan aset dengan metode ini dihitung dengan cara mengalikan bagian pengurang (*reducing fractions*) yang setiap tahun selalu menurun dengan harga perolehan dikurangi dengan nilai sisa yang ditetapkan.

Metode penyusutan aset dipilih berdasarkan ekspektasi pola konsumsi manfaat ekonomik masa depan dari aset dan diterapkan secara konsisten dari

periode ke periode kecuali ada perubahan dalam ekspektasi pola konsumsi manfaat ekonomik masa depan dari aset tersebut.

Metode penyusutan menurut Ketentuan Perundang-undangan Perpajakan sebagaimana telah diatur dalam Pasal 11 Undang-Undang Pajak Penghasilan adalah sebagai berikut (Direktorat Jenderal Pajak, 2013) :

1. Metode garis lurus (*straight line method*), atau metode saldo menurun (*diminishing balance method*) untuk aset tetap berwujud bukan bangunan.
2. Metode garis lurus untuk aset tetap berwujud berupa bangunan.

3.4 Aplikasi

Menurut Santoso (2005), yang dimaksud dengan aplikasi adalah suatu kelompok *file* (*form, class, report*) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi *payroll*, aplikasi *fixed assets*, dan lain-lain.

Solichin (2004) mengklasifikasikan aplikasi menjadi aplikasi berbasis *web*, aplikasi berbasis *desktop*, dan aplikasi berbasis *mobile*. Aplikasi berbasis *desktop* merupakan aplikasi yang memerlukan proses instalasi di setiap komputer yang akan menggunakannya. Sementara itu, aplikasi berbasis *web* tidak memerlukan instalasi di setiap komputer karena aplikasi berada pada suatu *server*, sehingga untuk mengakses aplikasi cukup menggunakan *browser* yang terhubung melalui jaringan ke *server*. Jenis aplikasi yang ketiga yaitu aplikasi berbasis *mobile* merupakan aplikasi yang hanya dapat dijalankan pada perangkat bergerak (*mobile device*) seperti telepon seluler, *smartphone*, dan PDA.

Mengacu kepada dua definisi tersebut, dapat disimpulkan bahwa aplikasi adalah kumpulan *file* komputer yang saling sinergis yang bertujuan untuk melakukan aktivitas tertentu dan dapat dikategorikan berdasarkan *platform* dimana aplikasi tersebut dapat dijalankan, menjadi aplikasi *web*, aplikasi *desktop*, dan aplikasi *mobile*.

3.5 Siklus Hidup Pengembangan Sistem

Menurut McLeod dan Schell (2008), *System Development Life Cycle* atau yang disingkat SDLC adalah metoda tradisional yang digunakan untuk membangun, memelihara dan mengganti suatu sistem informasi. SDLC terdiri dari tujuh fase, yaitu :

a. *Project Identification and Selection*

Fase dimana kebutuhan sistem informasi secara keseluruhan diidentifikasi dan analisa.

b. *Project Initiation and Planning*

Fase dimana suatu proyek sistem informasi yang potensial dilakukan dan direncanakan terinci dikembangkan untuk pengembangan sistem.

c. *Analysis*

Fase dimana sistem yang sedang berjalan dipelajari dan alternatif sistem baru diusulkan.

d. *Logical Design*

Fase dimana semua kegiatan fungsional dari sistem yang diusulkan untuk dikembangkan dan digambarkan secara independent.

e. *Physical Design*

Fase rancangan logis dari sebelumnya diubah dalam bentuk teknis yang terinci dimana pemrograman dan bentuk sistem dapat dibuat.

f. *Implementation*

Fase dimana sistem informasi diuji dan digunakan untuk mendukung suatu organisasi.

g. *Maintenance*

Fase dimana sistem informasi secara sistematis diperbaiki dan dikembangkan.

Siklus hidup pengembangan sistem merupakan suatu bentuk yang digunakan untuk menggambarkan tahapan-tahapan utama, dan langkah-langkah didalam tahapan tersebut dalam proses pengembangannya. Tiap-tiap pengembangan sistem itu dibagi menjadi beberapa tahapan kerja. Tiap tahapan ini mempunyai karakteristik tersendiri. Sebagai awal dari pelaksanaan pengembangan sistem adalah proses kebijaksanaan dan perencanaan sistem, dimana kebijaksanaan sistem merupakan landasan dan dukungan dari manajemen puncak untuk membuat perencanaan sistem. Sedangkan perencanaan sistem merupakan pedoman untuk melakukan pengembangan dari sistem tersebut.

3.5.1 Analisis Sistem

Menurut Fatta (2007), analisis sistem adalah sebuah istilah yang secara kolektif mendeskripsikan fase-fase awal pengembangan sistem. Analisis sistem adalah teknik pemecahan masalah yang menguraikan bagian-bagian komponen dengan mempelajari seberapa bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan mereka.

Analisis sistem merupakan tahapan paling awal dari pengembangan sistem yang menjadi fondasi menentukan keberhasilan sistem informasi yang dihasilkan nantinya. Tahapan ini sangat penting karena menentukan bentuk sistem yang harus dibangun. Tahapan ini bisa merupakan tahap yang mudah jika klien sangat paham dengan masalah yang dihadapi dalam organisasinya dan tahap ini bisa menjadi tahap yang paling sulit jika klien tidak bisa mengidentifikasi kebutuhannya atau tertutup terhadap pihak luar yang ingin mengetahui detail proses-proses bisnisnya.

3.5.2 Perancangan Sistem

Menurut Yakub (2012), perancangan sistem dapat didefinisikan sebagai penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Tujuan dari rancangan sistem adalah untuk memenuhi kebutuhan kepada pemakai sistem, dan untuk memberikan gambaran secara jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik lainnya.

Menurut Koniyo dan Kusrini (2007), perancangan sistem adalah proses pengembangan spesifikasi baru berdasarkan hasil rekomendasi analisis sistem. Dalam tahap perancangan, tim kerja harus merancang spesifikasi yang dibutuhkan dalam berbagai kertas kerja. Kertas kerja tersebut harus memuat berbagai uraian mengenai input, proses, dan output dari sistem yang diusulkan. Perancangan sistem dapat diartikan sebagai :

1. Tahap setelah analisis dari siklus pengembangan sistem.
2. Pendefinisian atas kebutuhan-kebutuhan fungsional.
3. Persiapan untuk rancang bangun implementasi.

4. Menggambarkan bagaimana sistem dibentuk, berupa penggambaran perencanaan, pembuatan sketsa, pengaturan dari beberapa elemen terpisah ke dalam satu kesatuan yang utuh dan berfungsi.
5. Konfigurasi komponen *software* dan *hardware* sistem.

IEEE Computer Society (2014) menjelaskan bahwa, hasil dari perancangan sistem memberikan gambaran mengenai arsitektur sistem, yaitu tentang bagaimana sistem didekomposisi dan diorganisasikan ke dalam komponen-komponen yang lebih kecil dan antarmuka antar komponen-komponen tersebut. Desain sistem juga harus memberikan gambaran komponen sistem pada level yang lebih terperinci untuk mendukung proses pengembangan sistem. *Tools* yang digunakan dalam perancangan sistem antara lain adalah bagan alir (bagan alir dokumen dan bagan alir sistem), *Data Flow Diagram* (DFD), dan *Entity Relationship Diagram* (ERD).

3.5.3 Bagan Alir

Menurut Jogiyanto (2005), *flowchart* (bagan alir) adalah bagan (*charts*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Dalam perancangan sistem, bagan alir terdiri atas dua jenis, yaitu bagan alir dokumen dan bagan alir sistem.

A. Bagan Alir Dokumen (*Document Flow*)

Menurut Jogiyanto (2005), bagan alir dokumen (*document flowchart*) atau bisa disebut juga sebagai bagan alir formulir yang merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.

Dalam pembuatannya, *document flowchart* memiliki ketentuan-ketentuan yang harus diperhatikan. Salah satunya adalah notasi-notasi yang ada di dalamnya. Notasi yang umumnya ada pada *document flowchart* dijelaskan pada tabel 3.1.

Tabel 3.1. Simbol-simbol *document flowchart*


No.	Simbol	Nama Simbol	Fungsi
1.		<i>Terminator</i>	Merupakan bentuk dari <i>terminator</i> simbol yang digunakan pada awal pembuatan <i>document flowchart</i> sebagai mengawali (<i>Start</i>) dan mengakhiri (<i>End</i>) <i>flowchart</i> .
2.		<i>Manual process</i>	Merupakan notasi dari proses manual yang pada <i>document flowchart</i> . Dinyatakan sebagai proses manual karena dalam notasi <i>document flowchart</i> segala bentuk proses masih belum dilakukan oleh komputer.
3.		<i>Document</i>	Merupakan notasi dari dokumen pada <i>document flowchart</i> . Notasi <i>document</i> ini umumnya digambarkan sebagai bentuk lain dari arsip, laporan atau dokumen lainnya yang berbentuk kertas.
4.		<i>Decision</i>	Merupakan notasi dari suatu keputusan dalam pengerjaan <i>document flowchart</i> . Dalam penggambaran notasi <i>decision</i> ini selalu menghasilkan dengan

No.	Simbol	Nama Simbol	Fungsi
			keputusan ya atau tidak.

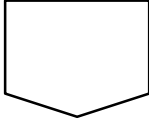
B. Bagan Alir Sistem (*System Flow*)

Menurut Jogiyanto (2005), bagan alir sistem (*system flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem. Pembuatan *system flowchart* memiliki aturan dan ketentuan yang harus diikuti. Seperti halnya dalam pembuatan *document flowchart*, *system flowchart* memiliki notasi-notasi sebagai representasi dari proses kerja suatu sistem. Sebagian notasi dalam *system flowchart* memiliki kesamaan dengan notasi yang ada pada *document flow* seperti, *terminator* (*start/end*), dan notasi laporan. Selain kedua notasi tersebut terdapat perbedaan secara bentuk dan fungsinya. Simbol-simbol *system flowchart* dijelaskan pada tabel 3.2

Tabel 3.2. Simbol-simbol *system flowchart*

No.	Simbol	Nama Simbol	Fungsi
1.		Proses komputerisasi	Simbol di samping merupakan representasi dari proses sistem. Menggambarkan sistem yang dikerjakan oleh komputer (otomatis).
			Gambar disamping adalah representasi dari Database yang

No.	Simbol	Nama Simbol	Fungsi
2.		<i>Database</i>	mana fungsinya untuk menyimpan data dari proses sebelumnya.
3.		<i>Connector</i>	<i>Connector</i> difungsikan sebagai penunjuk arah aliran dari satu proses ke proses yang lainnya yang saling berkaitan.
4.		<i>Sub-Process</i>	Simbol <i>Sub-Process</i> difungsikan untuk menunjukkan adanya proses yang lebih rinci dari suatu proses utama
5.		<i>Document</i>	Simbol <i>Document</i> menunjukkan tentang dokumen yang dihasilkan
6.		<i>Decision</i>	Simbol di samping difungsikan sebagai langkah pengambil keputusan . Keputusan yang ada terkait “ya” atau “tidak” keputusan diambil.
7.		<i>Input/Output</i>	Simbol di samping difungsikan untuk menunjukkan masukan data (<i>input</i>) dan data yang dihasilkan (<i>output</i>).
8.		<i>Connector (On Page Reference)</i>	Untuk menunjukkan hubungan simbol yang saling terkait dalam <i>system flowchart</i> . Selain itu juga sebagai pengganti garis <i>connector</i> untuk menghubungkan simbol yang




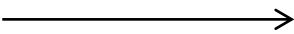
No.	Simbol	Nama Simbol	Fungsi
			saling berjauhan.
9.		<i>Connector (Off-Page Reference)</i>	Fungsinya hampir sama dengan <i>connector (on page reference)</i> perbedaannya adalah untuk menghubungkan simbol yang berada pada halaman yang berbeda.

3.5.4 Data Flow Diagram (DFD)

Menurut Jogiyanto (2005), DFD adalah diagram yang menggunakan notasi simbol untuk menggambarkan arus data sistem. DFD menggambarkan komponen-komponen yang ada pada suatu sistem serta aliran-aliran data. Fungsi utama dari DFD yaitu untuk memudahkan analisis sistem untuk menggambarkan suatu sistem sebagai satu fungsi yang saling berhubungan antara satu dengan yang lain.

DFD digunakan untuk menyajikan sistem dalam beberapa tingkat perincian dari yang sangat umum ke yang sangat terperinci. DFD banyak digunakan oleh analisis sistem untuk mewakili elemen logis dari sistem. Akan tetapi, teknik ini tidak mewakili sistem fisik. Dengan kata lain, DFD menunjukkan tugas logis yang sedang dilakukan, namun tidak menunjukkan cara melakukannya atau siapa (atau apa) yang melakukannya. Simbol-simbol yang digunakan dalam membuat DFD dijelaskan pada tabel 3.3.

Tabel 3.3. Simbol-simbol *Data Flow Diagram* (DFD)

No.	Simbol	Nama Simbol	Fungsi
1.		<i>External entity</i>	Untuk menggambarkan asal atau tujuan data di luar sistem. Sebagai penggambaran dari entitas eksternal (orang, sekelompok orang, organisasi, departmen organisasi, dsb) yang sama tetapi di luar kendali sistem yang akan dibuat
2.		<i>Process</i>	Untuk menggambarkan bagian dari sistem yang memproses inputan menjadi output. Setiap proses diberi nama dengan menggunakan kata kerja transitif, seperti ; menghitung gaji, menghitung penjualan, dsb.
3.		<i>Data store</i>	Untuk menggambarkan media penyimpanan data, seperti file atau database. Media penyimpanan berkaitan dengan penyimpanan secara komputerisasi.
4.		<i>Data Flow</i>	Untuk menggambarkan arah keluar masuknya dari suatu proses. Alur data digunakan untuk menjelaskan proses perpindahan data/informasi

No.	Simbol	Nama Simbol	Fungsi
			dari satu proses ke proses lain.

3.5.5 Entity Relationship Diagram (ERD)

Menurut Hall (2007), diagram relasi entitas atau ERD adalah teknik dokumentasi yang digunakan untuk menyajikan relasi antar entitas. Entitas (*entity*) adalah sumber daya fisik (mobil, kas, atau persediaan), kegiatan (memesan persediaan, menerima kas, atau mengirim barang), pelaku (staf penjualan, pelanggan, atau pemasok) yang akan digunakan oleh organisasi untuk mendapatkan data. Salah penggunaan umum dari ERD adalah untuk memodelkan basis data organisasi

Simbol bujur sangkar digunakan dalam ERD untuk mewakili entitas dalam sistem. Garis penghubungnya mewakili sifat hubungan antara dua entitas. Tingkat hubungannya disebut kardinalitas (*cardinality*), yaitu pemetaan numerik antara entitas. Hubungannya bisa satu ke satu (1:1), satu ke banyak (1:M), atau banyak ke banyak (M:M). Kardinalitas pada ERD mencerminkan peraturan bisnis umum serta kebijakan organisasional.

3.6 Tools Pemrograman

Tools pemrograman adalah bahasa pemrograman atau program komputer yang digunakan oleh pengembang sistem untuk membuat, men-*debug*, me-*maintain*, atau untuk mendukung sistem yang dikembangkan. Dalam rancang bangun aplikasi inventarisasi aset tetap pada PT. Indoberka Investama, *tools*

pemrograman yang digunakan adalah Notepad++, HTML, PHP, dan perangkat lunak manajemen basis data MySQL.

3.6.1 Definisi Notepad++

Menurut Kurniawan dan Java Creativity (2010), Notepad++ adalah sebuah program *freeware* (bebas lisensi) yang berfungsi sebagai editor pengganti Notepad *default* bawaan Windows. Notepad ++ ditulis dalam bahasa C++ yang menjamin kecepatan eksekusi lebih tinggi dan ukuran program lebih kecil. Editor ini biasa digunakan untuk mengedit halaman *web* berformat HTML standar menggantikan Dreamweaver. Beberapa keunggulan Notepad++ adalah sebagai berikut :

1. Tampilan lebih menarik dan menyegarkan.
2. Lebih *user friendly* dan mudah penggunaannya.
3. Mendukung *multi-tab*.

3.6.2 Bahasa Pemrograman

Utami dan Sukrisno (2005) mendefinisikan bahasa pemrograman sebagai kumpulan aturan yang disusun sedemikian rupa sehingga memungkinkan pengguna komputer membuat program yang dapat dijalankan dengan aturan tersebut. Bahasa pemrograman dapat dikelompokkan dalam berbagai macam sudut pandang. Salah satu pengelompokan bahasa pemrograman adalah pendekatan dari notasi bahasa pemrograman tersebut, apakah lebih dekat ke bahasa mesin atau ke bahasa manusia. Dengan cara ini bahasa pemrograman dapat dikelompokkan menjadi dua, yaitu bahasa tingkat rendah (*low-level languages*) dan bahasa tingkat tinggi (*high-level languages*). Dalam rancang bangun aplikasi inventarisasi aset tetap pada PT. Indoberka Investama, bahasa pemrograman yang

digunakan adalah *Hyper Text Markup Language* (HTML), *Hypertext Preprocessor* (PHP), *Cascading Style Sheet* (CSS), *Javascript*, dan *Structured Query Language* (SQL). Adapun penjelasannya adalah sebagai berikut.

A. *Hyper Text Markup Language* (HTML)

Menurut Sya'ban (2010), HTML adalah singkatan dari *Hyper Text Markup Language*. Fungsi HTML di dalam sebuah dokumen *web* adalah untuk mengatur struktur tampilan dokumen tersebut dan juga untuk menampilkan *link* atau sambungan ke halaman *web* lain yang ada di internet. Biasanya sebuah dokumen HTML disimpan dalam ekstensi *.html*. HTML disebut *markup language* karena berfungsi untuk memperindah file tulisan (*text*) biasa untuk dapat dilihat pada sebuah *web browser*.

B. *Hypertext Preprocessor* (PHP)

Menurut Anhar, ST (2010), PHP atau *hypertext preprocessor* adalah bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman tersebut diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima oleh *client* selalu terbaharui (*up to date*). Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

C. *Cascading Style Sheet* (CSS)

Sulistiyawan dkk. (2008) mendefinisikan CSS sebagai suatu bahasa *stylesheet* yang digunakan untuk mengatur *style* suatu dokumen. Pada umumnya CSS dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa

HTML dan XHTML. CSS memungkinkan *web developer* untuk memisahkan HTML dari aturan-aturan untuk membentuk tampilan sebuah *website*.

CSS adalah sebuah dokumen yang berdiri sendiri dan dapat dimasukkan dalam kode HTML atau sekedar menjadi rujukan oleh HTML dalam pendefinisian *style*. CSS menggunakan kode-kode yang tersusun untuk menetapkan *style* pada elemen HTML atau dapat juga digunakan untuk membuat *style* baru yang biasa disebut *class*. CSS dapat mengubah besar kecilnya teks, mengganti warna *background* pada sebuah halaman, atau dapat pula mengubah warna *border* pada tabel, dan masih banyak lagi hal yang dapat dilakukan oleh CSS. Singkatnya, CSS digunakan untuk mengatur susunan tampilan pada halaman HTML.

D. Javascript

Menurut Wahana Komputer (2010), *Javascript* merupakan bahasa yang berbentuk kumpulan skrip yang berfungsi untuk memberikan tampilan yang tampak lebih interaktif pada dokumen *web*. Dengan kata lain, bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan ke dalam bahasa pemrograman HTML dengan mengizinkan pengekseskusan perintah-perintah pada sisi klien, dan bukan sisi *server* dokumen *web*. *Javascript* tidak membutuhkan *compiler* karena pemasangannya diselipkan pada bahasa pemrograman HTML.

Kelebihan dari penggunaan *Javascript* adalah sebagai berikut :

1. Lebih praktis dan mudah sebab *Javascript* memiliki sedikit sintaks.
2. Koneksi cepat sebab peletakan program berada pada sisi klien, berukuran *file* sangat kecil, dan dapat dijalankan langsung di *browser*.

Sedangkan kekurangan *Javascript* terdapat pada pengelolaan objek yang terbatas.

E. *Structured Query Language (SQL)*

Menurut Arief (2006), *Structured Query Language (SQL)* adalah salah satu bahasa generasi level ke-4 yang awalnya dikembangkan oleh IBM. SQL adalah bahasa yang bersifat *request oriented* dan bersifat non-prosedural, sehingga lebih mudah untuk dipelajari karena sintaksis yang digunakan hampir menyerupai bahasa yang digunakan oleh manusia untuk berkomunikasi. Oleh karena itu, SQL lebih fleksibel dalam penggunaannya. Selain itu, SQL juga bersifat *non case sensitive*. Banyak vendor pembuat *Database Management System (DBMS)* yang saat ini menggunakan SQL sebagai standarisasi dalam produk mereka, seperti ORACLE, Microsoft SQL Server, PostgreSQL, dan MySQL.

SQL memiliki tiga bagian utama, yaitu :

1. DDL (*Data Definition Language*), yaitu perintah yang memiliki kemampuan untuk mendefinisikan data yang berhubungan dengan pembuatan dan penghapusan objek seperti tabel, indeks, bahkan basis datanya sendiri. Misalnya : CREATE, DROP, dan ALTER.
2. DML (*Data Manipulation Language*), yaitu perintah yang berhubungan dengan proses manipulasi data pada tabel atau *record*. Misalnya : INSERT, UPDATE, SELECT, dan DELETE.
3. DCL (*Data Control Language*), yaitu perintah yang digunakan untuk mengontrol data. Misalnya : GRANT dan REVOKE.

3.6.3 Database Management System (DBMS)

Yanto (2016) mendefinisikan DBMS sebagai sebuah paket program (*software*) yang dibuat agar memudahkan dan mengefisienkan pemasukan, pengeditan, penghapusan, dan pengambilan informasi terhadap *database*. Perangkat lunak yang tergolong ke dalam DBMS antara lain, Microsoft SQL Server, MySQL, ORACLE, Microsoft Access, dan lain-lain.

Penyimpanan data dalam DBMS akan mempunyai banyak manfaat dan kelebihan, antara lain :

1. *Performance*, misalnya efisiensi dalam hal media penyimpanan dan penggunaan memori.
2. *Integrity*, misalnya kemampuan DBMS untuk mengatasi masalah *redundancy* atau kejadian berulangnya data atau kumpulan data yang sama yang dapat menyebabkan pemborosan media penyimpanan.
3. *Independency*, misalnya kemudahan dalam mengubah struktur basis data tanpa harus mengubah aplikasi yang mengaksesnya.
4. *Centralization*, misalnya kemudahan dalam mengelola basis data yang terpusat dan konsistensi data yang diakses bersama-sama lebih terjamin
5. *Security*, yaitu sistem keamanan DBMS yang lebih fleksibel daripada pengamanan pada *file* sistem operasi. Keamanan dalam DBMS memberikan keluwesan untuk memberikan hak akses kepada pengguna.

Dalam rancang bangun aplikasi inventarisasi aset tetap pada PT. Indoberka Investama, DBMS yang digunakan adalah MySQL. Adapun penjelasannya adalah sebagai berikut.

A. MySQL

Menurut Yanto (2016), MySQL merupakan sistem manajemen *database* yang bersifat *open source* atau gratis dan telah mulai dikembangkan pada tahun 1979 oleh perusahaan TcX yang berasal dari Swedia. Pada tahun 1996 MySQL 3.11.1 mulai dipublikasi di dunia dan didistribusikan untuk Linux. Sampai saat ini MySQL sudah dapat bekerja untuk banyak *platform* dengan dilengkapi *source code*. Keunggulan dari MySQL antara lain sebagai berikut :

1. Kecepatan.
2. Kemudahan bagi user dalam penggunaannya.
3. Bersifat *open source* atau gratis.
4. Mendukung bahasa *query*.
5. Akses data dapat dilakukan di setiap tempat dengan fasilitas internet.
6. MySQL mudah didapatkan karena *source code* yang disebarluaskan.

