

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem Informasi Akuntansi**

Menurut Kusrini (2010:9), Sistem Informasi Akuntansi merupakan sistem informasi yang mengubah data transaksi bisnis yang menjadi informasi keuangan yang berguna bagi pemakainya.

Tujuan dari Sistem Informasi Akuntansi adalah :

- a. Mendukung operasi sehari-hari
- b. Mendukung pengambilan keputusan manajemen
- c. Memenuhi kewajiban yang berhubungan dengan pertanggungjawaban

Komponen-komponen yang terdapat dalam Sistem Informasi Akuntansi adalah sebagai berikut :

- a. Orang-orang yang mengoperasikan sistem tersebut.
- b. Prosedur-prosedur, baik manual maupun terotomatisasi, yang dilibatkan dalam pengumpulan, pemrosesan dan penyimpanan data aktivitas-aktivitas organisasi.
- c. Data tentang proses-proses bisnis.
- d. Software yang dipakai untuk memproses data organisasi.
- e. Infrastruktur teknologi informasi.

Didalam organisasi Sistem Informasi Akuntansi berfungsi untuk :

- a. Mengumpulkan dan menyimpan aktivitas yang dilaksanakan disuatu organisasi, sumber daya yang dipengaruhi oleh aktivitas-aktivitas tersebut dan para pelaku aktivitas tersebut.
- b. Mengubah data dan informasi yang berguna bagi manajemen.
- c. Menyediakan pengendalian yang memadai.

Sistem Informasi Akuntansi merupakan pendukung aktivitas organisasi.

Yang termasuk pendukung aktivitas organisasi adalah :

1. Infrastruktur perusahaan, akuntansi, hukum dan administrasi umum.
2. Sumber daya manusia : perekrutan, pengontrolan, pelatihan dan kompensasi kepada pegawai.
3. Teknologi : Peningkatan produk dan jasa (Penelitian).
4. Pembelian.

Sementara itu aktivitas utamanya adalah :

- a. *Inbount Logistics*, penerimaan, penyimpanan dan distribusi bahan-bahan masukan.
- b. Operasi : aktivitas untuk mengubah masukan menjadi barang atau jasa.
- c. *Outbount Logistics* : distribusi produk kepelanggan.
- d. Pemasaran dan Penjualan.
- e. Pelayanan : Dukungan purna jual maintenance.

### II.1.1. Siklus Sistem Informasi Akuntansi

Menurut Kusri (2010:11), Sistem Informasi Akuntansi memiliki beberapa sistem bagian (*sub system*) yang berupa siklus akuntansi. Siklus akuntansi menunjukkan prosedur akuntansi, mulai dari sumber data sampai ke proses pencatatan/pengolahan akuntansinya. Berikut ini adalah pembagian dari siklus akuntansi.

a. Siklus pendapatan

Siklus pendapatan merupakan prosesur pendapatan yang dimulai dari bagian penjualan otorisasi kredit, pengambilan barang, penerimaan barang, penagihan sampai dengan penerimaan kas.

b. Siklus pengeluaran kas

Siklus pengeluaran kas merupakan prosedur pengeluaran kas yang dimulai dari proses pembelian sampai proses pembayaran.

c. Siklus konversi

Siklus konversi merupakan siklus produksi, dimulai dari bahan mentah sampai barang jadi.

d. Siklus Manajemen Sumber Daya Manusia (SDM)

Siklus Manajemen Sumber Daya Manusia (SDM) merupakan siklus yang melibatkan proses penggajian pada karyawan.

e. Siklus buku besar dan laporan keuangan

Siklus ini berupa prosedur pencatatan dan perekaman ke jurnal dan buku besar dan pencetakan laporan keuangan yang datanya diambil dari buku besarnya.

Didalam sebuah Sistem Informasi Akuntansi, tidak semua siklus harus diimplementasikan. Yang wajib ada dalam system tersebut adalah siklus buku besar dan laporan keuangan. Transaksi-transaksi yang termasuk dalam siklus tetapi tidak diimplementasikan, misalnya penggajian, dapat dimasukkan dalam siklus buku besar.

## **II.2. Aktiva Tetap (Aset Tetap)**

Menurut Trio Mandala Putra (2013:192), Aset tetap adalah kekayaan perusahaan yang memiliki wujud, mempunyai manfaat ekonomis lebih dari satu tahun, dan diperoleh perusahaan untuk melaksanakan kegiatan perusahaan, bukan untuk dijual kembali.

Aset tetap disebut juga *plant asset* atau *fixed assets* dan mendefinisikannya sebagai berikut: Harta berwujud (*tangible asset*) yang memiliki masa manfaat ekonomis lebih dari satu tahun bernilai material, dan digunakan untuk kegiatan operasi normal perusahaan dan merupakan pengeluaran yang nilainya besar atau material. Aktiva tetap adalah aktiva-aktiva dengan sifat yang tidak bias dikonversikan menjadi uang tunai dalam siklus operasi perusahaan. Termasuk didalamnya adalah bangunan, peralatan dan tanah atau properti berwujud yang diinginkan dalam suatu bisnis, tetapi tidak diharapkan untuk digunakan habis atau dikonfirmasikan menjadi dana tunai dalam perjalanan bisnis biasa.

Di dalam PSAK 16 yang dimaksud dengan aset tetap adalah aset berwujud yang dimiliki untuk digunakan dalam produksi atau penyediaan barang dan jasa

untuk direntalkan kepada pihak lain, atau untuk tujuan administrative, dan Diharapkan untuk digunakan selama lebih dari satu periode.

### **II.2.1. Klasifikasi Aktiva tetap (Aset Tetap)**

Menurut Trio Mandala Putra (2013:192), Jenis-jenis aset tetap terdiri dari:

1. Lahan, yaitu bidang tanah terhampar baik yang merupakan tempat bangunan maupun yang masih kosong. Dalam akuntansi, apabila ada lahan yang didirikan bangunan di atasnya, maka pencatatan antara bangunan dan lahan harus dipisahkan. Khusus untuk bangunan yang dianggap sebagai bagian dari lahan atau konstruksi yang dapat meningkatkan nilai lahan itu sendiri, maka pencatatannya dapat digabungkan dengan nilai lahan.
2. Gedung, adalah bangunan yang berdiri di atas lahan baik yang berdiri di atas tanah maupun di atas air. Tidak seperti tanah yang tidak pernah disusutkan, maka gedung mengalami penyusutan dari tahun ke tahun sehingga nilainya akan berkurang tiap periodenya.
3. Mesin, yaitu alat mekanis yang dikuasai perusahaan dalam kegiatannya baik untuk dagang maupun jasa. Pencatatannya dilakukan dengan menambahkan nilai dari peralatan-peralatan yang menjadi bagian dari mesin itu.
4. Kendaraan, merupakan sarana angkutan yang dimiliki perusahaan untuk mendukung kegiatan operasionalnya. Misalnya, truk, mobil dinas, kendaraan roda dua, serta jenis kendaraan lain yang dapat digunakan sebagai sarana transportasi.
5. Inventaris, perlengkapan yang melengkapi isi kantor misalnya. Termasuk perlengkapan pabrik, kantor, ataupun alat-alat besar yang digunakan dalam

perusahaan. Contoh: inventaris kantor, inventaris pabrik, inventaris laboratorium, serta inventaris gudang.

Dari uraian diatas bisa diketahui pada umumnya, aset tetap digolongkan kedalam beberapa kelompok besar aset yaitu, Tanah, gedung, mesin, kendaraan dan inventaris.

### **II.2.2. Perolehan Aset Tetap**

Dalam PSAK No. 16 (Revisi 2011) paragraf 6 menyatakan Biaya perolehan adalah jumlah kas atau setara kas yang dibayarkan atau nilai wajar dari imbalan lain yang diserahkan untuk memperoleh suatu aset pada saat perolehan atau konstruksi atau, jika dapat diterapkan, jumlah yang distribusikan ke aset pada saat pertama kali diakui sesuai dengan persyaratan tertentu dalam PSAK lain, misalnya PSAK 53 (revisi 2010): Pembayaran berbasis saham. (Trio Mandala Putra, 2013:193).

Aset tetap dapat diperoleh melalui beberapa cara, diantaranya adalah :

1. Pembelian tunai.
2. Pembelian secara kredit.
3. Pembelian dengan surat berharga.
4. Diterimadarisumbangan.
5. Dibangun sendiri dan.
6. Pertukaran atau tukar tambah.

### II.3. Penyusutan

Menurut Samuel Mairuhu dan Jantje J. Tinangon (2014:406), menyatakan bahwa 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*). Faktor-faktor yang harus diperhatikan dalam menghitung beban penyusutan adalah :

1. Biaya perolehan (*initial cost/capitalized cost*), yaitu jumlah keseluruhan biaya-biaya yang dikeluarkan oleh sebuah organisasi bisnis untuk memperoleh aset tetap.
2. Umur manfaat (*usefull life*), yaitu estimasi atau perkiraan lamanya waktu penggunaan aset tetap tersebut.
3. Nilai sisa/residu (*residual value/scrap value/salvage value/trade-in value*), yaitu estimasi nilai tunai aset tetap yang diharapkan pada akhir umur manfaatnya.
4. Jumlah biaya yang dapat disusutkan/jumlah tersusutkan (*asset's depreciable cost*), yaitu selisih antara biaya perolehan aset tetap dengan nilai residunya. Jumlah ini kemudian akan dialokasikan secara sistematis sebagai beban penyusutan.
5. Jumlah tercatat/nilai buku (*book value*) adalah selisih antara biaya perolehan dengan akumulasi penyusutan.

Standar Akuntansi Keuangan (2012) dalam jurnal Samuel Mairuhu dan Jantje J. Tinangon (2014:406), menyatakan bahwa “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*).

Penyusutan secara garis lurus merupakan metode pembebanan/alokasi sistematis dari biaya perolehan (harga beli) aset tetap menjadi beban penyusutan dalam laporan rugi laba secara konstan /tetap selama umur manfaat aset tetap tersebut. Penentuan beban penyusutan dapat dilakukan dengan menggunakan rumus berikut:

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

**(Sumber : Analisis Penerapan Metode Penyusutan Aktiva Tetap Dan Implikasinya Terhadap Laba Perusahaan Pada Perum Bulog Drive Sulut Dan Gorontalo ; 2014)**

Penyusutan dengan saldo menurun ganda merupakan aset tetap yang dipercepat, dimana dasar penggunaan tarif garis lurus dikalikan dengan 2 (dua) dipakai untuk menentukan tarif penyusutan. Dengan persamaan sebagai berikut:

$$\text{Tarif Penyusutan} = \text{Tarif Garis Lurus} \times 2$$

Metode unit produksi merupakan metode yang menghitung penyusutan aset tetap dengan pertimbangan faktor penggunaan atau keluaran aset tetap tersebut. Dengan persamaan sebagai berikut:

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

**(Sumber : Analisis Penerapan Metode Penyusutan Aktiva Tetap Dan Implikasinya Terhadap Laba Perusahaan Pada Perum Bulog Drive Sulut Dan Gorontalo ; 2014)**

Metode Jumlah Angka Tahun adalah metode ini menghitung beban penyusutan dengan cara membagi biaya perolehan dikurangi nilai residunya dengan jumlah keseluruhan dari umur manfaat aset tetap tersebut. Dengan persamaan sebagai berikut:

$$\text{Dasar Penyusutan} = 1+2+3+4+5 = 15$$

**(Sumber : Analisis Penerapan Metode Penyusutan Aktiva Tetap Dan Implikasinya Terhadap Laba Perusahaan Pada Perum Bulog Drive Sulut Dan Gorontalo ; 2014)**

#### **II.4. Client Server**

Menurut Wahana Komputer (2010:5), *Client Server* adalah salah satu model komunikasi dua computer atau lebih yang berfungsi melakukan pembagian

tugas. *Client* bertugas melakukan melakukan input, *update*, penghapusan data dan menampilkan data dari dalam sebuah database. Sementara server bertugas menyediakan pelayanan untuk melakukan manajemen yaitu penyimpanan dan mengolah database.

Aplikasi *Client Server* merupakan jawaban atas berkembangnya teknologi informasi, dimana sebuah perusahaan memiliki banyak departemen dan harus terhubung satu sama lain dalam melakukan akses data.

## **II.5. Unified Modeling Language (UML)**

Adi Nugroho (2010:6), *Unified Modeling Language (UML)* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma “berorientasi objek” . Pemodelan (*Modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Dalam hal ini sasaran model sesungguhnya adalah abstraksi segala sesuatu yang ada diplanet bumi menjadi gambaran-gambaran umum yang lebih mudah dipahami dan dipelajari. Adapun tujuan pemodelan (dalam rangka pengembangan system/perangkat lunak aplikasi) sebagai sarana analisis, pemasahan visualisasi dan komunikasi antar anggota tim pengembang.

### **II.5.1. Pengenalan UML**

Adi Nugroho (2010), *UML* sebagai sebuah bahasa yang memberikan *vocabulary* dan tatanan penulisan kata-kata dalam ‘*MS Word*’ untuk kegunaan komunikasi. Sebuah bahasa model adalah sebuah bahasa yang mempunyai

*vocabulary* dan konsep tatanan / aturan penulisan serta secara fisik mempresentasikan dari sebuah sistem. Seperti halnya *UML* adalah sebuah bahasa *standard* untuk pengembangan sebuah *software* yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan *software*.

*UML* tidak hanya merupakan sebuah bahasa *pemograman visual* saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemograman, seperti *JAVA*, *C++*, *Visual Basic*, atau bahkan dihubungkan secara langsung ke dalam sebuah *object-oriented database*.

Begitu juga mengenai pendokumentasian dapat dilakukan seperti: *requirements*, *arsitektur*, *design*, *source code*, *project plan*, *tests*, dan *prototypes*. Untuk dapat memahami *UML* membutuhkan bentuk konsep dari sebuah bahasa model, dan mempelajari 3 (tiga) elemen utama dari *UML* seperti *building block*, aturan-aturan yang menyatakan bagaimana *building block* diletakkan secara bersamaan, dan beberapa mekanisme umum (*common*).

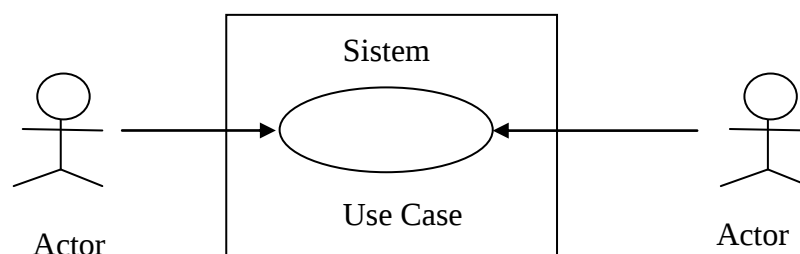
Alasan mengapa *UML* digunakan adalah pertama, *scalability* dimana objek lebih mudah dipakai untuk menggambarkan sistem yang besar dan kompleks. Kedua, *dynamic modeling*, dapat dipakai untuk pemodelan sistem dinamis dan *real time*. Sebagaimana dalam tulisan pertama, penulis menjelaskan konsep mengenai obyek, *OOA&D (Obyek Oriented Analyst/ Design)* dan pengenalan *UML*, maka dalam tulisan kedua ini lebih ditekankan pada cara bagaimana *UML*

digunakan dalam merancang sebuah pengembangan software yang disertai gambar atau contoh dari sebuah aplikasi.

Adapun jenis-jenis dari tipe diagram *UML* adalah sebagai berikut :

### 1. *Use Case Diagram*

*Use Case* adalah deskripsi fungsi dari sebuah sistem dari *perspektif* pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Model *use case* adalah bagian dari *requirement*. *Use case* adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Dengan demikian diharapkan akan bisa dibangun suatu sistem yang bisa membantu pengguna, perlu diingat bahwa *use case* mewakili pandangan diluar sistem. Diagram *Use Case* menunjukkan 3 aspek dari sistem yaitu : *actor*, *use case* dan *system/sub system boundary*. *Actor* mewakili peran orang, *system* yang lain atau alat ketika berkomunikasi dengan *use case*. Berikut gambar notasi *use case* :

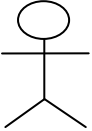
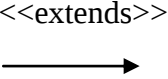
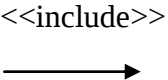


**Gambar II.1. Notasi *Use Case Diagram***

**(Sumber : *Rekayasa Perangkat Lunak Berorientasi Objek Dengan Metode USDP ; 2010*)**

Berikut ini gambar *table* simbol-simbol dari *Use Case Diagram* adalah:

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

| <b>Nama<br/>Komponen</b> | <b>Deskripsi</b>                                                                | <b>Gambar</b>                                                                         |
|--------------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| <i>Use Case</i>          | Menunjukkan proses yang terjadi pada system                                     |    |
| <i>Actor</i>             | Menunjukkan user yang akan menggunakan sistem.                                  |    |
| <i>Association</i>       | Sebuah garis yang berfungsi menghubungkan <i>Actor</i> dengan <i>Use Case</i> . |  |
| <i>Extend</i>            | Perluasan dari <i>Use Case</i> lain jika kondisi atau syarat terpenuhi.         |  |
| <i>Include</i>           | Menjelaskan bahwa <i>Use Case</i> termasuk dalam <i>Use Case</i> lain.          |  |

(Sumber : *Rekayasa Perangkat Lunak Berorientasi Objek Dengan Metode USDP ; 2010*)

## 2. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok :

- a. Nama (dan *stereotype*)
- b. *Atribut*
- c. *Metoda*

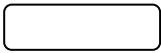
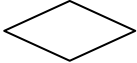
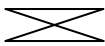
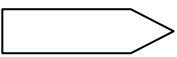
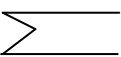
*Atribut* dan *metode* dapat memiliki salah satu sifat berikut :

- a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
- b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
- c. *Public*, dapat dipanggil oleh siapa saja.
- d. *Class* dapat merupakan implementasi dari sebuah *interface*, yaitu *class abstrak* yang hanya memiliki *metoda*. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi *metoda* pada saat *run-time*.

### 3. *Activity Diagram*

*Activity Diagram* adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Berikut adalah simbol yang ada pada *activity diagram*.

**Tabel II.2. Simbol Yang Ada Pada Activity Diagram**

| SIMBOL                                                                              | KETERANGAN                                                                                                                                       |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
|    | Titik Awal                                                                                                                                       |
|    | Titik Akhir                                                                                                                                      |
|    | Activity                                                                                                                                         |
|    | Pilihan untuk pengambilan keputusan.                                                                                                             |
|    | <i>Fork</i> : digunakan untuk menunjukkan kegiatan yang dilakukan secara 29cenario atau untuk menggabungkan dua kegiatan 29cenario menjadi satu. |
|  | <i>Rake</i> ; menunjukkan adanya dekomposisi                                                                                                     |
|  | Tanda waktu                                                                                                                                      |
|  | Tanda pengiriman                                                                                                                                 |
|  | Tanda penerimaan                                                                                                                                 |
|  | Aliran Akhir ( <i>Flow Final</i> )                                                                                                               |

(Sumber : *Rekayasa Perangkat Lunak Berorientasi Objek Dengan Metode USDP ; 2010*)

#### 4. Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu.

## 5. *Collaboration Diagram*

*Collaboration diagram* juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*.

## II.6. Pengertian Basis Data (*Database*)

Basis data merupakan kumpulan dari data-data yang saling terkait dan saling berhubungan satu dengan yang lainnya. Basis data adalah kumpulan-kumpulan *file* yang saling berkaitan.

Menurut Wahana Komputer (2010:140), Database diartikan sebagai representasi fakta dunia nyata yang mewakili sebuah objek, misalnya manusia, hewan, barang, peristiwa, konsep dan lain sebagainya yang direkam dalam bentuk huruf, teks, symbol, angka, suara, gambar dan lainnya. Sedangkan basis data dapat diartikan sebagai tempat berkumpul, sarang atau gudang untuk menyimpan sesuatu. Dengan demikian basis data/database dapat diartikan sebagai tempat berkumpul, menyimpan data-data suatu benda atau kejadian yang saling berhubungan.

Menurut Kusri (2010, p2), pengertian Basis Data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau symbol).

Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.

2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpan elektronik.

#### **II.6.1. Tujuan Basis Data**

Menurut Kusriani (2010, p2), Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuan, syarat basisdata yang baik adalah sebagai berikut :

- a. Tidak adanya redudansi dan inkonsistensi data

Redudansi terjadi jika suatu informasi disimpan di beberapa tempat. Misalnya ada data mahasiswa yang memuat nim, nama, alamat dan atribut lainnya, sementara kita punya data lain tentang data KHS mahasiswa yang isinya terdapat NIM, nama, mata kuliah dan nilai. Pada kedua data tersebut kita temukan atribut nama.

- b. Kesulitan pengaksesan data

Basis data memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan query ataupun dari tool yang melibatkan tabelnya. Dengan fasilitas ini, bisa segera langsung melihat data dari software DBM-nya.

- c. Multiple user

Basis data memungkinkan penggunaan data secara bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda. Dengan

meletakkan basis data pada bagian server yang bisa diakses dari banyak client, sudah menyediakan akses kesemua pengguna dari komputer client ke sumber informasi yaitu basis data.

### **II.6.2. Manfaat/Kelebihan Basis Data**

Menurut Kusrini (2010, p5), Banyak manfaat yang diperoleh dengan menggunakan basis data, Manafaat/Kelebihan Basis Data dan kelebihan basis data diantaranya adalah :

a. Kecepatan dan kemudahan

Dengan menggunakan basis data pengambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokkan, mengurutkan bahkan perhitungan dengan metematika. Dengan perancangan yang benar maka penyajian informasi dapat dilakukan dengan cepat dan mudah.

b. Kebersamaan pemakai (*sharability*)

Sebuah basis data dapat digunakan oleh banyak user dan banyak aplikasi. Untuk data yang diperlukan oleh banyak bagian/orang, tidak perlu dilakukan pencacatan dimasing-masing bagian/orang, tetapi cukup dengan satu basis data untuk dipakai bersama.

c. Pemusatan kontrol data

Karena cukup satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan disatu tempat saja.

d. Efisiensi ruang penyimpanan

Dengan pemakaian bersama, tidak perlu menyediakan tempat penyimpanan diberbagai tempat tetapi cukup satu saja, sehingga ini dapat menghemat ruang penyimpanan yang dimiliki oleh sebuah organisasi.

e. Keakuratan (*Accuracy*)

Penerapan secara tepat acuan tipe data, domain data, keunikan data, hubungan antar data, dan lain-lain, dapat menekan ketidakakuratan dalam pemasukan/penyimpanan data.

f. Ketersediaan (*Availability*)

Dengan basis data, semua data dapat dibackup, memilah-milah data mana yang masih diperlukan yang perlu disimpan ke tempat lain. Hal ini mengingat pertumbuhan transaksi sebuah organisasi dari lain waktu ke waktu membutuhkan penyimpanan yang semakin besar.

g. Keamanan (*Security*)

Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna. Pengguna diberi hak akses yang berbeda-beda sesuai dengan kepentingan dan posisinya. Basis data bisa diberikan password untuk membatasi orang yang diaksesnya.

h. Kemudahan dalam pembuatan program aplikasi baru

Penggunaan basis data merupakan bagian dari perkembangan teknologi. Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan dari DBMS. Sehingga membuat aplikasi tidak perlu mengurus penyimpanan data, tetapi cukup mengatur interface untuk pengguna.

i. Pemakaian secara langsung

Basis data memiliki fasilitas yang lengkap untuk melihat datanya secara langsung dengan tools yang disediakan oleh DBMS.

j. Kebebasan data

Perubahan dapat dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasinya.

k. User View

Basis data menyediakan pandangan yang berbeda-beda untuk tiap-tiap pengguna.

### **II.6.3. Operasi Dasar Database**

Menurut Kusrini (2010, p9), Beberapa operasi dasar basis data yaitu :

- a. Pembuatan basis data
- b. Penghapusan basis data
- c. Pembuatan file/tabel
- d. Penghapusan file/tabel
- e. Pengubahan tabel
- f. Penambahan/pengisian
- g. Pengambilan data
- h. Penghapusan data

### **II.6.4. Pemodelan Basis Data**

Menurut Samiaji Sarosa (2010:4), Model diperlukan untuk mendapatkan penyederhanaan dari kenyataan dan memungkinkan desainer program program

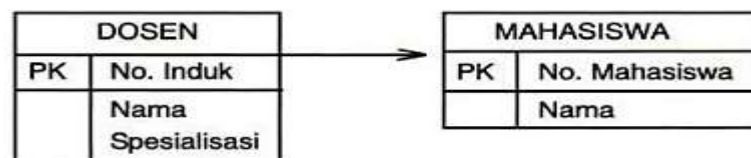
aplikasi bereksperimen dengan berbagai macam variable sebelum diaplikasikan ke system yang berjalan. Untuk merancang suatu aplikasi basis data alat yang biasa digunakan adalah *Entity Relationship Diagram (ERD)*. ERD didasarkan dari artikel yang dipublikasikan oleh Peter Phin Shan Chen.

Ada beberapa case tool menamakan notasi ERD yang digunakan sebagai *chen ERD*. *Entity Relationship Model* adalah abstraksi konseptual yang mewakili struktur dari suatu basis data.

Dalam perkembangannya banyak diciptakan notasi ERD yang berbeda-beda seperti terlihat gambar dibawah ini :



**Gambar II.2. Diagram Dengan Notasi Crows Foot**  
(Sumber : *Sistem Informasi Akuntansi ; 2010*)



**Gambar II.3. Diagram Dengan Notasi Relational**  
(Sumber : *Sistem Informasi Akuntansi ; 2010*)

## II.6.5. Normalisasi

Menurut Samiaji Sarosa (2010:5), Normalisasi adalah teknik yang dirancang untuk merancang tabel basis data relasional untuk meminimalkan

duplikasi data dan menghindarkan basis data tersebut anomali. Suatu basis data dikatakan tidak normal jika terjadi 3 (tiga) anomali berikut :

a. *Insertion Anomaly*

Anomali yang terjadi jika ada data yang tidak bisa disisipkan kedalam table.

b. *Deletion anomaly*

Anomali yang terjadi jika ada satu data dihapus pada satu table akan menghapus data tentang keberadaan suatu anggota entitas.

c. *Update/Modification anomaly*

Anomali yang terjadi jika ada perubahan pada suatu item data maka harus mengubah lebih dari satu baris data.

Langkah-langkah normalisasi sampai pada bentuk 3NF adalah sebagai berikut :

a. *First Normal Form (1NF)*

Untuk menjadi 1NF suatu table harus memenuhi dua syarat. Syarat pertama tidak ada kelompok data atau *field* yang berulang. Syarat kedua harus ada *primary key (PK)* atau kunci unik, atau kunci yang membedakan satu baris dengan baris yang lain dalam satu table. Pada dasarnya sebuah table selamat tidak ada kolom yang sama merupakan bentuk table dengan 1NF.

b. *Second Normal Form (2NF)*

Untuk menjadi 2NF suatu table harus berada dalam kondisi 1NF dan tidak memiliki *partial dependencies*. *Partial dependencies* adalah suatu kondisi jika atribut non kunci (Non PK) tergantung sebagian tetapi bukan seluruhnya pada PK.

c. *Third Normal Form (3NF)*

Untuk menjadi 3NF suatu table harus berada dalam kondisi 2NF dan tidak memiliki *transitive dependencies*. *Transitive dependencies* adalah suatu kondisi dengan adanya ketergantungan fungsional antara 2 atau lebih atribut non kunci (Non PK).

## II.7. Visual Basic 2010

Menurut Edi Winarno ST, M.Eng dkk (2010:1), Visual Basic adalah bahasa pemrograman klasik, legendaris yang paling banyak dipakai oleh programmer didunia. Pemograman ini dipakai oleh jutaan programmer dan tercatat sebagai program yang paling disukai oleh mayoritas orang.

Visual Studio 2010 pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Visual Studio 2010 selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (tool) untuk menghasilkan program-program aplikasi berbasis windows.

Beberapa kemampuan atau manfaat dari Visual Studio 2010 diantaranya seperti :

1. Untuk membuat program aplikasi berbasis windows.
2. Untuk membuat objek-objek pembantu program seperti, misalnya : kontrol ActiveX, file Help, aplikasi Internet dan sebagainya.

3. Menguji program (debugging) dan menghasilkan program berakhiran EXE yang bersifat executable atau dapat langsung dijalankan.

### II.7.1. Antar Muka Visual Basic 2010

Saat menjalankan Visual Basic 2010 pertama kali muncul jendela *chose default envirointment settings*. Disini bisa memilih apakah ingin memilih antar muka di Visual Studio. Untuk *programmer* Visual Basic lebih baik memilih *Visual Basic Development Centre*.



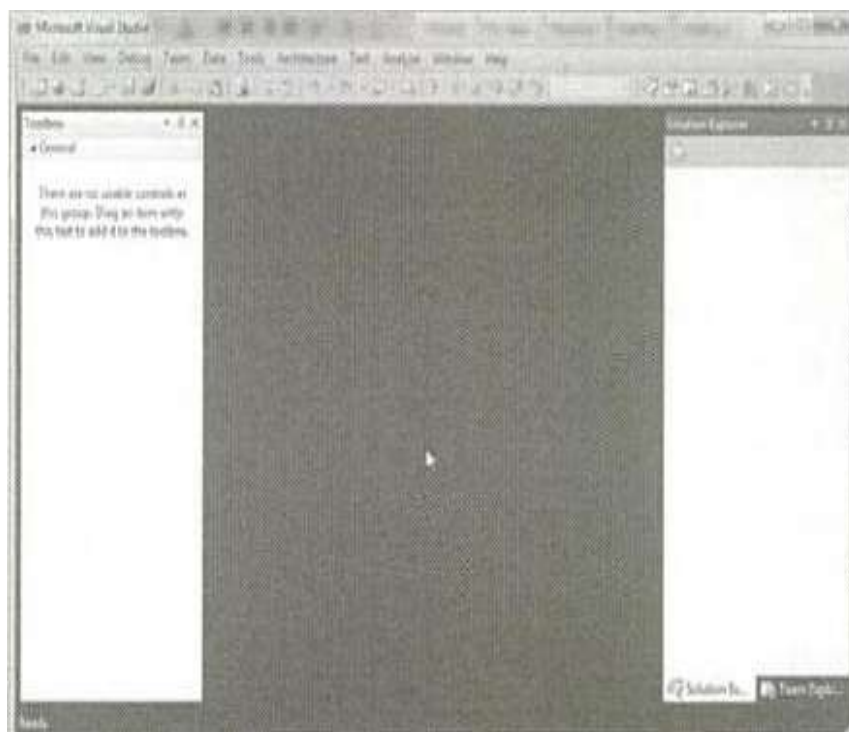
**Gambar II.4. Form Chose Default Envirointment Settings**  
(Sumber : Dasar-Dasar Pemrograman Dengan Visual Basic 2010 ; 2010)

Dibagian awal visual basic, bisa memilih *Start Page*. *Start Page* adalah halaman yang mencantumkan informasi-informasi seputar program dan juga informasi RSS dari sumber tertentu. Jika tidak ingin menampilkan hal ini hilangkan tanda centang pada *Show Page On Startup*.



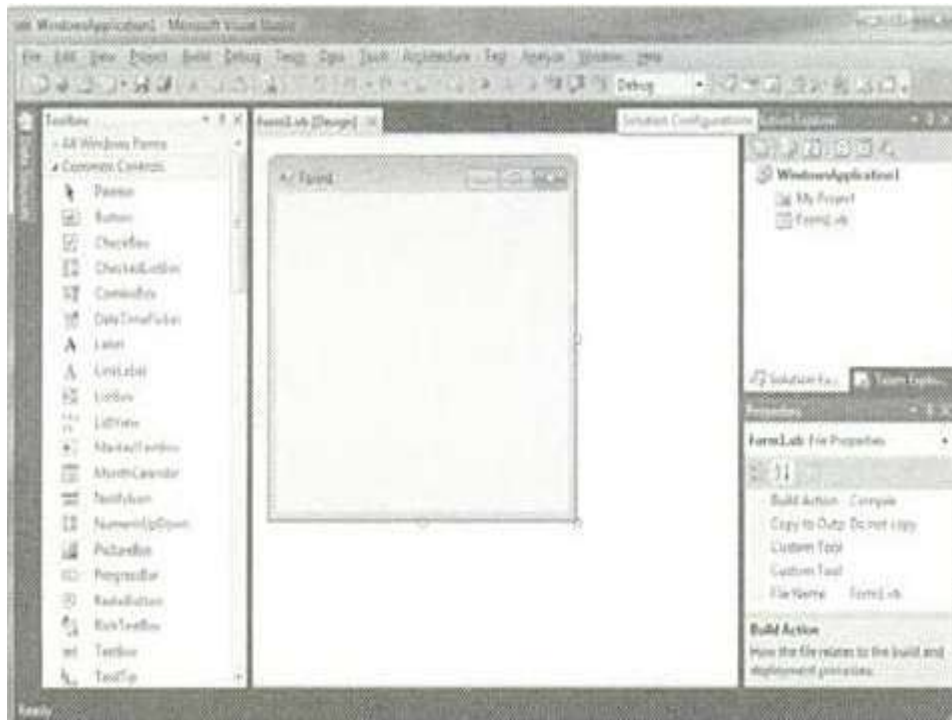
**Gambar II.5. Start Page Visual Basic 2010**  
*(Sumber : Dasar-Dasar Pemrograman Dengan Visual Basic 2010 ; 2010)*

Jika start page ditutup terlihat tampilan sebagai berikut :



**Gambar II.6. Tampilan IDE (Integrated Development Environment) setelah Start Page ditutup**  
*(Sumber : Dasar-Dasar Pemrograman Dengan Visual Basic 2010 ; 2010)*

Jika ada sebuah form yang terlihat, tampilan lengkap IDE seperti gambar berikut ini.



**Gambar II.7. Tampilan lengkap IDE**  
*(Sumber : Dasar-Dasar Pemrograman Dengan Visual Basic 2010 ; 2010)*

Komponen-komponen dari IDE adalah :

1. Dibagian kiri terdapat toolbox yang menampilkan semua objek tool yang bisa dimasukkan kedalam form untuk membuat program.
2. Dibagian tengah terdapat tempat meletakkan form dan kode, baik disaat desain ataupun pada saat program dijalankan.
3. Dibagian kanan terdapat solution explorer yang merupakan explorer untuk melihat file-file disebuah objek.
4. Dikanan bawah terdapat propertis untuk melihat properti dari nilai-nilai pada objek yang dipilih dibagian tengah. (Edi Winarno ST, M.Eng dkk, 2010:1).

## **II.8. SQL Server 2008 *Express Edition***

Menurut Wahana Komputer (2010:2), SQL Server 2008 *Express Edition* sebuah terobosan baru 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 2008 *Express Edition* dibuat pada saat kemajuan dalam bidang hardware semakin pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 *Express Edition* membawa terobosan dalam bidang pengolahan dan penyimpanan data.

### **II.8.1. Kebutuhan *Hardware***

Adapun *hardware* yang diperlukan untuk instalasi SQL Server 2008 *Express Edition* minimal adalah sebagai berikut :

- a. Prosescor minimal 1 GHz
- b. Memori minimal 512 MB
- c. Sistem Operasi Windows

Biar dapat diinstal pada system computer dengan memoti 512 MB, tetapi disarankan menggunakan memori 1 GB. Sedangkan untuk jaringannya diperlukan adalah :

- a. *Sharer Memory*
- b. TCP/IP
- c. *Named Pipes*
- d. *Virtual Interface Adapter (VIA)*(Wahana Komputer, 2010:2).

### II.8.2. Versi SQL Server 2008 *Express Edition*

Microsoft merilis SQL Server 2008 *Express Edition* dalam beberapa versi yang disesuaikan dengan segmen-segmen pasar yang dituju. Versi-versi tersebut adalah sebagai berikut :

- a. Menurut cara pemrosesan data pada prosesor maka Microsoft mengelompokkan produk ini berdasarkan dua jenis yaitu :
  1. Versi 32 Bit (x86), yang biasanya digunakan untuk komputer *single processor* (Pentium 4) atau lebih tepatnya processor 32 bit atau Windows XP
  2. Versi 64 Bit (x64), yang biasanya digunakan oleh computer yang lebih dari satu processor (Misalnya *Core 2 duo*) dan system operasi 64 bit, Vista dan Windows 7.
- b. Sedangkan secara keseluruhan terdapat versi-versi seperti berikut :
  1. Versi *Compact* ini adalah versi “tipis” dari semua versi yang ada
  2. Versi *Express* ini adalah versi “ringan”

### II.8.3. Instalasi SQL Server 2008 *Express Edition*

Proses instalasi SQL Server 2008 *Express Edition* tidak sama dengan instalasi versi-versi sebelumnya. Proses SQL Server 2008 *Express Edition* agak panjang melalui beberapa tahapan. Tahapan yang dilakukan akan membawa beberapa pilihan yang akan diisi dalam *setting* sebuah *server database*. Berikut ini adalah pilihan-pilihan yang akan dijumpai dalam proses instalasi SQL Server 2008 *Express Edition*.

1. Tempat direktori utama dan penyimpanan file database

Direktori utama adalah direktori dimana semua file program akan ditempatkan dan file-file tersebut tidak akan berubah selama anda menjalankan SQL server. Direktori utama secara standard akan berada dalam direktori “C:\Program Files\Microsoft SQL Server”.

## 2. Penggunaan *Multiple instance*

*Instance* adalah sebuah turunan dari server database SQL Server. Karena sebuah tiruan maka sebuah *Instance* memiliki fungsi yang sama dengan database server aslinya. Arti sebenarnya *Instance SQL Server* adalah sebuah server database yang tidak men-*sharing* sistemnya dan database user dengan database server lainnya yang ada dalam komputer yang sama.

## 3. Jasa *Autentification User* (Menggunakan Windows atau mixed)

*Autentification User* diperlukan supaya server tidak dapat dipergunakan oleh orang yang tidak bertanggungjawab dan tidak berhak. Dalam SQL server ada dua *Autentification User* yang dapat digunakan yaitu :

- a. Mode Windows, Pada mode ini SQL Server akan melakukan autentifikasi dengan menggunakan level login pada system operasi.
- b. Mode Mixel atau campuran, mode ini menginjinkan *user* untuk masuk kedalam system SQL server dengan menggunakan *Account* yang dibuat di system operasi windows atau juga menggunakan *account* yang di *set up* pada SQL Server (Wahana Komputer, 2010:2).