2. LANDASANTEORI

2.1. *Activity Based Costing (ABC)*

Sistem tradisional memiliki kelemahan dalam mengidentifikasi dan mengukur biaya apa saja yang masuk kedalam sebuah produk berdasarkan penggunaan sumber daya yang dikonsumsi oleh produk tersebut sampai menjadi barang jadi. Hal ini terjadi karena besarnya biaya ***overhead*** yang dialokasikan tidak hanya mencerminkan volume produk yang dihasilkan, tetapi juga banyaknya jenis produk dan tingkat kerumitan produk yang dihasilkan. Menurut Hansen & Mowen (2000:**113),** ada dua faktor utama yang mempengaruhi ketidakmampuan sistem ini untuk membebankan biaya ***overhead*** secara akurat, yaitu besarnya bagian biaya ***overhead nonunit*** (seperti biaya ***setup)*** dan tingginya tingkat diversitas produk (perbedaan proporsi aktivitas ***overhead*** yang dikonsumsi oleh setiap produk). Hal ini mengakibatkan laporan biaya produk terdistorsi dan memberikan efek negatif terhadap kinerja perusahaan secara keseluruhan dalam laporan keuangan. Dengan demikian, menangani alokasi biaya ***overhead*** merupakan salah satu tugas yang vital dalam menentukan besarnya biaya produk secara akurat. Hal ini menjadi sangat penting bagi perusahaan ***yang*** memiliki tingkat ***operating feverage*** yang sangat ***tinggi,*** karena biaya ***overhead*** menjadi amat besar dalam industri padat modal yang menggunakan teknologi canggih dalam proses produksinya. ABC adalah metode perhitungan biaya dengan cara mengidentifikasi aktivitas apa saja yang terlibat dalam menyebabkan timbulnya biaya dan mengalokasikannya kepada produk berdasarkan tingkat penggunaan aktivitas oleh produk tersebut **(Maher 1997:236).**

Aktivitas yang dimaksud adalah semua tugas yang dilakukan perusahaan untuk membuat sebuah produk dari pemakaian sumber daya perusahaan seperti mendesain produk, menyetel dan menjalankan mesin, serta mendistribusikan produk tersebut. Jika manajemen perusahaan **menginginkan** produk yang kompetitif, maka mereka harus tahu aktivitas-aktivitas yang terlibat dalam pembuatan produk dan biaya yang **timbul** dari tiap aktivitas tersebut. Dalam ABC, biaya dilaporkan bukan berdasarkan alokasi per-departemen melainkan

berdasarkan alokasi dari aktivitas-aktivitas. Sistem ABC umumnya menggunakan 4 kelompok tingkatan biaya untuk mengidentifikasi dasar alokasi biaya sebagai pemicu dalam pool biaya aktivitas, yaitu: (Horngren 1998:143)

a. facility-sustaining costs

Berhubungan dengan aktivitas untuk mempertahankan kapasitas produk yang dimiliki perusahaan, mempertahankan proses produksi pabrik secara umum dan mendukung operasional perusahaan secara keseluruhan. Contoh: biaya administrasi umum, penyusutan bangunan, amortisasi, asuransi. keamanan. pemeliharaan bangunan dan manajemen pabrik. Biaya ini dibebankan atas kapasitas normal perusahaan.

b. product-sustaining (service-sustaining)costs

Berhubungan dengan penelitian dan pengembangan produk serta biaya lain yang dikeluarkan untuk mempertahankan dan mendukung berbagai produk yang dihasilkan oleh perusahaan. Contoh: biaya pengembangan produk. desain produk, desain proses pengolahan produk dan pengujian produk. Biaya ini dibebankan berdasarkan taksiranjumlah unit produk yang akandihasilkau selama umur produk tersebut ***(product life cycle).***

c. batch-level costs

Berhubungan dengan jumlah kelompok dari unit produk yang dihasilkan Contoh biaya ***setup*** dan perlengkapan lain yang dibutuhkan sebelum suatu ***order*** diproses, inspeksi, penjadwalan produksi, penanganan material. Besar kecilnya biaya ini berdasarkan frekwensi***order*** produksi dan tidak dipengaruhi oleh jumlah unit produkyang diproses dalamsetiap order produksi.

d. output unit-level costs

Dipengaruhi jumlah unit produkyang dihasillran. Contoh: biaya bahanbaku langsung, penyusutan mesin, machining, assembly. Biaya ini dibebankan berdasarkan biaya per-unit dikalikan dengan jumlah produk yang dihasilkan.

Sistem ABC menggunakan dasar alokasi ***unit based*** dan ***nonunit based activity driver*** untuk membebankan biaya pada produk, dimana *driver* in harus mencerminkan adanya hubungan sebab-akibat. Hal ini biasanya menyebabkan jumlah ***driver*** yang digunakandalam ABC menjadi lebih banyak daripada jumlah ***unit bused driver*** yang biasa digunakan dalam sistem tradisional, sehingga sistem

ini mampu untuk menghasilkan informasi yang lebih akurat mengenai biaya produk.

Menurut Hansen & Mowen (2000:117), ada dua tahap prosedur dalam mengalokasikan biaya kepada produk yaitu:

a Tahap Pertama

Mengidentifikasi semua aktivitas (biaya dihubungkan dengan aktivitas yang terjadi dan aktivitas tersebut juga dihubungkan dengan biaya yang muncul). Identifikasi ini memerlukan sebuah daftar yang berisi tentang semua jenis pekerjaan yang ada seperti inspeksi, penanganan material, proses produksi, maupun perbaikan produk. Ketika sebuah aktivitas ditemukan, biaya aktivitas tersebut ditentukan dengan menggunakan penelusuran langsung atau resource driver-nya, kemudian menentukan activity driver yang berhubungan dengan aktivitas itu dan menghitung tarif aktivitas tersebut. Untuk mengurangi banyaknya jumlah tarif itu, biaya aktivitas yang ada dikelompokkan dalam beberapa homogeneous *cost pool* berdasarkan karakteristiknya yang sejenis yaitu secara logis saling berhubungan dan mempunyai rasio konsumsi yang sama wtuk semua produk (rasio konsumsi adalah bagian dari tiap aktivitas yang dikonsumsi oleh sebuah produk). Kemudian menghitung tarif overhead dari *pool* ini dengan membagi total biaya overhead *pool* tersebut dengan sebuah activity driver-nya.

b. Tahapkedua

Membebankan biaya overhead dari tiap *pool* kepada produk dengan cara mengalikan tarif pool dan jumlah unit bused driver yang dikonsumsi oleh produk yang dihasilkan.

Secara umum, Activity Based Costing menawarkan banyak kesempatan untuk memberikan nilai tambah bagi perusahaan, terutama melalui:

- Informasi yang lebih baik tentang biaya produk

  ABC menggunakan data yang lebih banyak daripada metode tradisional dan menyediakan estimasi biaya produk badasarkan informasi yang lebih akurat, karena ABC mengidentifikasi aktivitas-aktivitas yang menimbulkan biaya dan menentukan besarnya biaya dari aktivitas-aktivitas tersebut, sehingga *ABC* lebih jelas dalam mengidentifikasi dan mengukur biaya sebuah

produk daripada metode tradisional. Informasi biaya produk yang lebih baik membantu manajer untuk membuat perencanaan dan mengambil keputusan yang lebih baik dan tepat, seperti dalam penetapan harga dan melakukan pengembangan atau penghentian produk tertentu. Meski manajemen harus tanggap terhadap kondisi pasar, mereka juga harus mempertimbangkanbiaya produk mereka dalam menetapkan harganya.

Secara umum, manajer pemasaran sering menetapkan harga produknya di bawah harga pasar untuk memperoleh bagian yang lebih besar dalam pangsa pasarnya, atau menawarkan harga khusus untuk pelanggan tertentu. Informasi biaya produk yang akurat dapat membantu pihak manajemen untuk memutuskan seberapa jauh mereka dapat menurunkan harga. Manajemen juga dapat menggunakan informasi ini untuk memutuskan apakah tetap menjual produk tertentu, jika profit margin produk tersebut terlalu rendah atau bahkan merugikan perusahaan, sangat mungkin manajemen akan memutuskan untuk menghentikan penjualan produk tersebut. Namun keputusan ini sulit untuk dilakukan dan dibutuhkan informasi terbaik untuk hal ini.

- Informasi yang lebih baik mengenai biaya dari aktivitas-aktivitas danproses-proses yang terjadi

ABC menyediakan informasi biaya yang dihubungkan dengan berbagai aktivitas untuk menghasilkan produk. Hal ini mempermudah pihak manajemen dalam mendapatkan inforamsi yang relevan dalam pengambilan keputusan yang berkaitan denganbisnis mereka. Selain itu, pihak manajemen dapat menggunakan informasi ini untuk mengidentifikasi aktivitas-aktivitas yang tidak memberikan nilai tambah (tetapi menghabiskan sumber daya perusahaan) dan mendorong mereka untuk melakukan pemantauan dan perbaikan metode produksi yang mahal (perbaikan kinerja produksi agar efisien dan efektif) secara terus menerus untuk menurunkan biaya ***overhead.*** Misalnya, manajemen dapat menyadari betapa mahalnya salah satu dari aktivitas yang terjadi dan mengambil langkah-langkah untuk mengurangi biaya tersebut seperti menghilangkan, menyeleksi, mengurangi atau memodifikasiaktivitas tersebut. Selain itu, ***ABC*** juga menguntungkan bagian produksi karena informasi ini membantu mereka untuk mengidentifikasi

aktivitas pemicu biaya yang sebelumnya tidak diketahui dan belajar untuk mengatur dan mengontrol aktivitas tersebut sehingga biaya produksi secara keseluruhan dapat terkontrol dengan baik.

Dengan mengidentifikasi biaya dari berbagai aktivitas, manajemen memperoleh informasi yang berguna yang sebelumnya tidak diketahui dalam sistem akuntansi terdahulu. Mungkin mereka akan menemukan berbagai informasi yang menarik dan berguna ataupun hanya mengkonfirmasikan apa yang telah diketahuinya, tetapi mereka *tidak akan* tahu seberapa besar biaya dari aktivitas-aktivitas yang mereka jalankan secara tepat dan akurat jika tidak mengimplementasikan ABC.

Namun, *ABC* juga memiliki beberapa kelemahan seperti membutuhkan data (yang diperoleh dari dokumentasi/catatan perusahaan) yang lebih banyak sehingga membutuhkan waktu yang lebih lama dan biaya yang lebih besar akibat tambahan pekerjaan yang harus dilakukan, seperti biaya akuntan dan orang lain untuk menerapkan ABC, biaya pencatatan, biaya *software,* biaya konsultasi, bahkan sampai mengubah aturan/kebiasaan perusahaan yang telah ada.

Sebuah *database* ABC merupakan sekumpulan *datu set* yang diatur dan dihubungkan untuk digunakan oleh sistem informasi ABC perusahaan. Sedangkan *data set* merupakan sebuah kelompok data yang secara logis berhubungan. Menurut Hansen & Mowen (1995:314), untuk memhuat sebuah database ABC membutuhkan tiga tahap. Pertama, membuat definisi dan model dari obyek yang ada dalam operasi sistem ABC, *seperti* aktivitas yang terjadi dan produk yang dihasilkan. **Kedua,** membuat **pengertian konseptual** *yang* **mencerminkan** obyek dan hubungan logis yang ada diantaranya. Ketiga, mengidentifikasikan atribut yang seharusnya ada dan berhubungan dengan setiap obyeknya Atribut ini ditentukan berdasarkan **tujuan** sistem informasi **yang** dibuat dan **kebutuhan** dari penggunanya, misalnya atribut kapasitas aktivitas dari ***activity driver*** dibutuhkan untuk menghitung **tarif *homogeneous cost pool*.**

Kegunaan analisis informasi *ABC* yang lebih baik mengenai biaya aktivitas untuk membantu manajemen dalam membuat keputusan dikenal dengan istilah ***Activity Based Management*** (ABM). Analisis aktivitas menampilkan cara yang sistematik bagi perusahaan untuk berpikir tentang proses yang digunakan dalam

menyediakan produk bagi pelanggan. ABM dapat *digunakan* untuk mengidentifikasi dan menghilangkan aktivitas yang menambah biaya, tetapi tidak memberikan nilai tambah kepada produk, sehingga aktivitas tersebut dapat dihilangkan tanpa mengurangi kualitas, kinerja, atau nilai dari produk tersebut. Beberapa aktivitas yang tidak memberikan nilai tambah kepada produk, seperti: (Maher 1997:269)

- Proses produksi yang tidak atau kurang efisien (banyak setup, inspeksi, atau produk cacat).
- Pemindahan barang (bahan baku, dan lain-lain) dari tempat penyimpanan ke tempat produksi atau dari departemen satu ke departemen lainnya
- Pekerja yang menunggu untuk proses selanjutnya karena adanya waktu kosong ***(idle time)***.
- Penyimpanan bahan baku, setengah jadi ataupun barang jadi. Manajemen dapat menggunakan filosofi ***Just*** in ***Time*** dalam pembelian dan produksi mereka.

ABC dapat dipakai untuk kegunaan strategik bagi perusahaan dengan cara menjadi produsen yang berbiaya rendah melalui pengurangan biaya aktivitas yang mahal dan yang tidak memberikan nilai tambah bagi perusahaan. ABC memegang peranan penting dalam strategi perusahaan dan rencana jangka panjang perusahaan untuk membangun keunggulan kompetitif (dalam biaya produk). Salah satu pengaruh yang besar dalam penetapan harga adalah biaya, selain pelanggan dan pesaing. Pengembangan yang dilakukan terus menerus dan kegunaan analisis informasi ABC memegang peran penting dalam mengurangi biaya, sehingga perusahaan dapat meurunkan harga, tetapi tetap menguntungkan.

2.2. Sistem Informasi Manajemen (SIM)

SIM didefinisikan sebagai suatu sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai dengan kebutuhan yang serupa. Informasi tersebut tersedia dalam bentuk laporan periodik, laporan khusus atau ***output*** dari simulasi matematika dan umumnya dihasilkan dari ***suatu shared database*** yang menyimpan data dari banyak sumber (McLeod 1995:78). ***Output*** informasi ini

digunakan oleh manajer maupun bukan manajer dalam perusahaan saat mereka membuat perencanaan, pengawasan dan pengontrolan operasi perusahaan, bahkan dalam pengambilan keputusan untuk memecahkan masalah.

Sistem lnformasi Manajemen memiliki 3 fungsi mendasar, yaitu:

- Mengumpulkan dan memproses data mengenai aktivitas usaha secara efektif dan efisien.

  Setiap perusahaan pasti melakukan aktifitas operasionalnya dandari aktivitas tersebut harus dapat dikumpulkan data yang diperlukan supaya menjadi informasi yang berguna bagi pihak manajemen. Dalam SIM, pemrosesan data sangat dibantu oleh komputer.

- Menyediakan informasi yang berguna untuk pengambilan keputusan.

  Dalam menyediakan suatu informasi harus dilihat untuk siapa informasi itu ditujukan. Ada dua kategori utama infomasi yang mempunyai fungsi yang berbeda, yaitu laporan keuangan dan laporan manajerial. Pada laporan keuangan, informasi yang ditampilkan merupakan ringkasan data yang ada dan digunakan untuk melihat kinerja perusahaan dari sisi keuangan. Laporan ini dilihat oleh pihak luar. Sedangkan pada laporan manajerial, informasi yang ditampilkan merupakan rincian dari setiap data yang muncul dan terjadi di perusahaan dan digunakan untuk menentukan strategi perusahaan masa mendatang.

- Memiliki pengendalian yang memadai.

  Hal ini diperlukan untuk memberikan keyakinan yang memadai bahwa data mengenai aktivitas usaha telah dicatat dan diproses secara baik.

Salah satu strategi yang dibangun agar perusahaan memperoleh keunggulan kompetitif adalah melalui strategi kepemimpinan biaya, artinya menekan biaya sehingga menjadi produsen yang berbiaya rendah dalam industri sejenis, dengan bantuan penggunaan strategik teknologi informasi.

2.3. *System Analysis and Design Methods*

Suatu sistem informasi dapat terbentuk tanpa bantuan seorang analis karena sistem ini merupakan produk sampingan yang alami dari kejasama seseorang

dengan orang lain. Dalam suatu organisasi pasti terdapat orang-orang yang aktif dan bekerja bersama dalam menyelesaikan pekerjaannya sehari-hari seperti transaksi pembelian, penjualan, pembuatan, pengembangan, pengiriman suatu barang, pembayaran, penerimaan kas dan sebagainya sehingga mereka secara otomatis akan membentuk suatu sistem tertentu untuk memastikan bahwa kegiatan operasionalnya dapat berjalan dengan baik. Untuk mengembangkan sistem ini, maka diperlukan adanya suatu metode desain dan analisis sistem yang tepat. Menurut Whitten & Bentley, suatu sistem informasi mempunyai definisi seperti berikut ini:

> ***"An information system is an arrangement of people, data, processes, interfaces and geography that are integrated for the purposes of supporting and improving the day by day operations in a business, as well as fulfilling the problem solving and decision making information needs of business managers."*** (Whitten & Bentley **1998:38)**

Ada suatu perbedaan yang mendasar antara data dan informasi. Data merupakan suatu fakta mengenai suatu organisasi dan transaksi bisnisnya, sedangkan informasi merupakan data yang telah disaring dan diatur oleh suatu pemrosesan yang cerdas sehingga menghasilkan informasi yang **berguna,** misalnya bagi manajemen dalam mengambil keputusan bisnis. Ringkasnya, sistem informasi mengubah data menjadi informasi yang berguna Kemampuan untuk mengelola dan memanfatkan data dan informasi ini menjadi faktor penting dalam kesuksesan perusahaan.

Beberapa prinsip yang perlu diperhatikan dalam mengembangkan suatu sistem antara lain, mengusahakan keterlibatan ***owners*** dan ***users***, menggunakan suatu pendekatan untuk memecahkan masalahnya, membentuk tahap-tahap dan aktivitas yang akan dilakukan, membuat *standar* untuk dokumentasi dan pengembangan yang **konsisten,** memperlakukan suatu sistem sebagai investasi modal, tidak takut untuk menghapus atau mengubah batasannya, membagi suatu sistem kedalam subsistem agar lebih mudah dikuasai, danmendesain suatu sistem yang terbuka untuk pengembangan dan perubahan.

Prototyping mempakan suatu pendekatan dalam mendesain suatu sistem dengan membuat suatu model kerja yang sederhana ***(prototype)*** dari pengembangan sebuah sistem informasi. Menurut kamus Webster, ***"a prototype is***

an original or model on which something is patterned" and / or "afirst full scale and usually functional form of new type or design of a construction, as an airplane."

Suatu***prototype*** dapat dibuat untuk*output* sederhana, dialog komputer, fungsi **kunci,** keseluruhan subsistem atau bahkan keseluruhan dari sistem. ***Prototyping*** merupakan sebuah tehnik rekayasa yang digunakan untuk mengembangkan bagian dari sebuah sistem atau aplikasi namun dalam versi fungsionalnya. Hal ini membuat sebuah ***prototype*** dapat menjadi sebuah sistem yang final, dapat diimplementasikan ketika dikembangkan lebih lanjut dalam tahap desain dan konstruksi sistem.

Prototype juga dapat membawa *users* untuk menentukan apakah mereka menyukai atau tidak terhadap sistem yang dibuat. Berdasarkan reaksi dan masukan mereka, pengembang mengubah sistem tersebut dan mempresentasikan kembali kepada ***user.*** Proses ini berlanjut sampai ***user*** merasa puas bahwa sistem tersebut telah cukup memadai untuk memenuhi kebutuhan mereka. Premis dasar dari ***prototyping*** adalah lebih mudah bagi ***user*** untuk menyukai atau tidak mengenai suatu sistem yang nyata daripada hanya memikirkan apa yang benar-benar mereka sukai dalam sebuah sistem karma ***prototyping*** adalah suatu *tindakan* untuk membuat skala kecil dari model kerja yang mencerminkan kebutuhan ***user*** untuk menemukan atau membuktikan kebutuhan tersebut ***("they will*** know ***what they need when they see it")***.

Tahap-tabap dalam mengembangkan suatu sistem dari suatu prototype: (Romney 1997:408)

a. mengidentifikasikan kebutuhan dasar suatu sistem dengan bertemu *user* untuk menyetujui ukuran dan batasan dari sistem dan memutuskan apa yang seharusnya ada atau tidak dalam sistem tersebut. Pengembang dan *user* juga menentukan ***output*** pemrosesan transaksi, beserta ***input*** dan data yang dibutuhkan untuk menghasilkan ***output*** tersebut. Pengembang harus yakin bahwa harapan *user* tersebut realistis dan dapat dipenuhi.
b. membuat suatu ***prototype*** awal yang memenuhi kebutuhan tersebut. Kemudian mendemonstasikannya pada *user* dan meminta masukan untuk perubahan yang diperlukan.

c. proses perubahan ***prototype*** ini terus berlanjut sampai ***user*** puas dengannya.
d. menjalankan ***prototype*** yang telah disetujui oleh ***user*** dalam operasional mereka dengan tetap melakukan kontrol, meningkatkan efisiensinya, menyediakan ***backup*** dan ***recovery,*** dan mengintegrasikannya dengan sistem yang berhubungan.

Prototyping lebih cocok dilakukan ketika ***user*** tidak mengerti dengan baik kebutuhan mereka atau adanya pembahan pikiran yang cepat dari ***user*** secara kontinu adanya kesulitan untuk mendefinisikan kebutuhan dari sistemnya, ***input*** maupun ***output*** dari sistem masih belum jelas, tugas yang dijalankan tidak terstruktur atau agak terstruktur, pembuatnya tidak yakin dengan teknologi yang akan digunakan, sistem ini penting dan mendesak waktunya, resikonya tinggi dalam mengembangkan sistem yang **salah,** pentingnya reaksi ***user*** dalam pertimbangan pengembangan sistem, banyaknya strategi desain yang harus diuji, sistem tersebut akan digunakan tidak secara rutin, dan adanya kemungkinan tingkat kegagalan yang cukup tinggi.

Pendekatan ini mempunyai beberapa keunggulan seperti mendorong dan membutuhkan partisipasi aktif dari ***user*** sehingga memperoleh definisi yang lebih baik tentang kebutuhan mereka, dapat mengatasi kecenderungan ***user*** untuk berubah pikiran dalam pengembangan suatu sistem dengan lebih baik, membantu ***user*** yang seringkali tidak sungguh-suugguh mengetahui kebutuhannya sampai mereka melihat sistem tersebut **dijalankan,** memberikan kepuasan yang lebih karena ***prototype*** merupakan suatu model yang aktifdandapat dilihat, disentuh, dirasakan dan dialami oleh ***user,*** dapat mendeteksi kesalahan lebih dulu sehingga sistem lebih dapat diandalkan dan lebih **murah** untuk dikembangkan, dapat meningkatkan kreativitas karena memperoleh umpan balik yang lebih cepat dari user sehingga menghasilkan solusi yang lebih baik, *serta* dapat mempercepat beberapa tahap secara langsung sehingga waktu pengembangan yang dibuthkan lebih sedikit.

Tahap–tahap yang dilalui dalam membuat ***prototype*** dan mendesain ***input*** komputer: (Whitten & Bentley **1998:451-457)**

a. memeriksa kembali kebutuhan ***input***

Kita dapat menetapkan kebutuhan ***input*** dengan mempelajari desain atau kebutuhan dari ***output*** dan ***database.***

b. memilih kontrol GUI ***(Graphical User Interface)***

Kita dapat memilih kontrol yang tepat untuk tiap atribut yang akan ditampilkan setelah mengetahui ***input*** yang kita butuhkan.

c. membuat ***prototype*** tampilan ***input***

Tahap ini meliputi pengembangan tampilan ***prototype*** bagi ***user*** yang ***akan*** dikaji dan diuji oleh mereka. Masukan mereka **mungkin akan** menyebabkan kebutuhan **untuk** kembali pada tahap pertama dan kedua **untuk** menambah atribut baru dan memenuhi harapan ***user.***

Tahap-tahap yang dilalui dalam membuat ***prototype*** dan mendesain ***output*** komputer: (Whitten & Bentley **1998:47**1473)

a. mengidentifikasi ***output*** dari sistem

Perancang sistem harus bertemu, bertanya dengan user dan mengerti ***output*** yang diharapkan dari sistem tersebut. Mereka juga **harus** memikirkan kemungkinan ***output*** yang akan dihasilkan dengan memeriksa berbagai macam entitas data, isi dan hubungannya dengan entitas lain.

b. memilih format dan medium ***output***

Keputusan ini dibuat dengan menentukan medium dan format yang terbaik **untuk** desain dan implementasi yang **berdasar** pada bentuk dan kegunaan ***output,*** juga kemungkinan ekonomis **dan** teknisnya.

c. membuat ***protoype output*** untuk pemakai sistem

Bentuk atau **susunan** dari suatu output ***akan*** secara langsung mempengaruhi kemampuan pemakai sistem untuk membaca dan menginterpretasikannya Oleh karena itu, kita perlu menunjukkan prototype output ini kepada pemakai sistem untuk memperoleh masukan dan mengubahnya sampai benar-benar sesuai dengan keinginan mereka.

2.4. Microsoft Access

Database Access merupakan **suatu** metode **untuk** menyimpan dan memanggil kembali sejumlah data. ***Database*** ini *digunakan* untuk menyimpan, mengatur

informasi, dan mempunyai kemampuan untuk menambah, mengubah, memanipulasi dan memanggilnya dengan cepat. Aplikasi ***database*** adalah suatu ***database*** yang dilengkapi dengan ***user interface*** dan pemanggilan kembali data serta pilihan untuk mencetaknya (Groh & Harding 1994:243). ***Database*** Access disusun dari beberapa obyek yang disebut tabel, *query*, *form*, *macro*, *report* dan modul. Tabel merupakan komponen penyimpanan utama dari suatu ***database Access*** yang dibuat dari ***row (records)*** dan ***column*** *(field)* data yang diatur oleh ***Access.*** Query digunakan untuk menjawab pertanyaan yang kita tanyakan dari ***database,*** bagaimana kita memanggil informasi yang dipilih dari tabel dalam ***database.***

2.5. *Visual Basic (VB)*

Visual Basic merupakan suatu ligkungan pemrograman, yaitu suatu program yang didesain secara khusus untuk memfasilitasi pembuatan program-program baru. Semua lingkungan pemrograman menyediakan 2 hal bagi pembuat program yang menggunakannya, yaitu: (Burrows & Langford 1998:7)

- seperangkat alat pemrograman yang memampukan pembuat program untuk merakit dan mengatur ulang komponen program yang sedang dikerjakan.
- bahasa pemrograman yang memampukan pembuat program untuk menyusun perintah yang memberitahu komputer bagaimana melaksanakan tugas yang dibutuhkan oleh program yang sedang dikerjakan.

VB telah digunakan dalam banyak perusahaan karena memiliki beberapa kelebihan yaitu:

- Menyediakan alat yang mempermudah pembuat program untuk membuat GUI ***(Graphical*** User ***Interface)*** yang baik sehingga waktu yang dibutuhkan lebih singkat dan ***user*** lebih puas dengan **aplikasi** akhirnya.
- Menyederhanakan tugas pembuat program dalam menulis processing script.
- Menyediakan alat yang menyederhanakan proses penyesuaian suatu aplikasi untuk memperoleh data dari suatu ***database*** sehingga dapat menghemat banyak waktu dan uang.

- Mampu memodifikasi aplikasi bisnis ***(user interface*** dan ***processing script)*** dengan cepat **untuk** memenuhi pengaturan ***database*** yang baru.
- **Mudah untuk** memperoleh bantuan dalam menghadapi masalah teknis karena banyak perusahaan dan pembuat program yang telah memakai dan mengembangkan **VB** secara kontinu.
- Program dari ***Visual Basic*** dapat dibuat dalam bentuk **exe** file sehingga kerahasiaan programnya terjamin.

Biaya **untuk** membuat **suatu** aplikasi muncul dari banyaknya **waktu** yang digunakan oleh ***user*** **untuk** menyampaikan kebutuhannya kepada pembuat program dan waktu yang dibutuhkan oleh pembuat program **untuk** mendesain dan membentuk sebuah program yang mampu untuk memenuhi kebutuhan ***user.*** Ada beberapa faktor yang harus diperhatikan dalam membuat suatu aplikasi, yaitu:

a. Aplikasi tersebut seharusnya memenuhi kebutuhan user.
b. Aplikasi tersebut seharusnya mudah **untuk** digunakan. Hal ini dapat dicapai apabila pembuat **program** mendesain suatu ***user interface*** yang sederhana dan jelas.
c. **Detil** internal **dari** aplikasi seharusnya dapat dimengerti dengan **mudah** oleh pembuat program. Untuk mencapainya, pembuat program harus menyusun ***processing script*** **secara** jelas dan **sesederhana** mungkin.
d. Aplikasi tersebut seharusnya dibuat dengan fleksibel **agar** dapat melakukan perubahan **yang diperlukan** *dimasa* mendatang. Untuk mencapai hal ini, pembuat program **harus** mampu mengantisipasi kebutuhan perubahan dimasa mendatang **dan menyusun*processing script*** yang berhubungan.

Prosedur **formal untuk** membuat **suatu** aplikasi adalah:

a. mengidentifikasi kebutuhan ***user.***

   Dalam tahap ini, pembuat program mencari **tahu untuk** apa aplikasi **itu** dibuat, **ciri-ciri** apa yang harus **disediakan,** dan bagaimana aplikasi tersebut dijalankan.

b. membuat desain aplikasi (GUI dan fungsinya).

   Setelah mengerti kebutuhan ***user,*** pembuat program mengidentifikasi dan mengatur komponen yang **dibutuhkan** serta membuat kepastian **bahwa** aplikasi tersebut mampu dibuat dalam biaya yang dapat diterima.

Category	Primary Controls	Others
Trigger Initiate processing	*Command Button* *Menu* *Timer*	*Text Box* *Picture Box* *List Box* *Form*
Input: Get data from user	*Text Box* *Option Button* *Check Box* *List Box* *Combo Box*	*Common Dialog* *File List Box* *Directory List Box* *Drive List Box*
Output: Display results to user	*Label* *FIexGrid* *Image* *Picture Box*	*Text Box* *List Box* *Scroll Bar* *Form*
Organize: Group other controls	*Form* *Frame* *SSTab*	*Picture Box*
Beautify: Simple Graphics	*Line* *Shape*	
Data Access: Interface with database	*Data* *DBList* *DBCombo* *DBGrid*	
Integrate: Interface with other applications	*OLE*	

Data Control memampukan kita untuk berpindah dari satu *record* ke *record* yang lain dan dapat menampilkan serta memanipulasi data dari *record* tersebut dalam *Bound Control. DataList, DataCombo, DataGrid,* dan *MSHFlexGrid control* mampu untuk mengatur sejumlah records ketika dikaitkan dengan sebuah *Data control.* Semua kontrol ini dapat menampilkan dan memanipulasi beberapa *record* dengan cepat. Kontrol *Picture, Label, TextBox, CheckBox, Image, OLE, ListBox* dan *ComboBox* dapat dikaitkan pada sebuah *field* dari *Recordset* yang diatur oleh *Data control. Data control* mengakses data dengan menggunakan *Microsoft Jet Database engine, database engine* yang sama untuk *Microsoft Access.* Teknologi ini memberikan akses ke banyak format *database* standar dan dapat membuat aplikasi data tanpa menulis kode apapun. *Data control* paling baik digunakan untuk *database* yang kecil seperti *database Access* dan ISAM. *Data control* digunakan untuk membuat aplikasi yang dapat menampilkan, menyunting, memperbaiki informasi dari banyak tipe *database* yang ada seperti Microsoft *Access, Btrieve, dBASE, Microsoft Foxpro®* dan *Paradox.* Kita juga dapat menggunakannnya untuk mengakses *Microsoft Excel,* Lotus *2-2-3,* dan *standard ASCII text files* jika mereka memang adalah suatu *database.* Sebagai tambahan, *data control* juga dapat mengakses dan memanipulasi *remote Open Database Connectivity (ODBC) database* seperti *Microsoft SQL Server* dan *Oracle.*

ADO Data control menggunakan *Microsoft ActiveX Data Objects (ADO)* untuk secara cepat membuat hubungan antara *data-bound control* dan *data provider. Datu-bound control* adalah kontrol yang mempunyai suatu *DataSource property. Data provider* dapat merupakan sumber manapun yang termasuk dalam spesifikasi OLE DB. Kita juga dapat dengan mudah membuat data provider sendiri dengan menggunakan modul kelas VB.Waktu mendesain, kita dapat membuat suatu hubungan dengan menentukan properti ConnectionString pada sebuah hubungan yang benar, kemudian menentukan properti *RecordSource* pada suatu pernyataan yang s e e dengan pengaturan *database.* Kita juga dapat langsung mengatur properti *ConnectionString* pada sebuah *namafile* yang akan dihubungkan. *File* ini dihasilkan oleh *DataLink dialog box* yang akan muncul ketika kita menekan *ConnectionString* pada properti *window.* Kemudian hubungkan *ADO Data Control* pada sebuah *data-bound control* seperti *DataGrid,*

DataCombo, atau *DataList control* dengan mengatur properti *DataSource* pada ADO *Data Control.* Meskipun kita menggunakan ADO secara langsung dalam aplikasi, ADO *Data control* mempunyai keunggulan yaitu dapat juga menjadi suatu kontrol *grafik* (adanya tombol *back danforward)* dan suatu *interface* yang mudah digunakan untuk membuat aplikasi *database* hanya dengan kode yang minim. Saat dijalankan, kita juga dapat mengatur properti *ConnectionString* dan *RecordSource* secara dinamis untuk mengubah *database.* Hal ini tidak dapat dilakukan oleh data *control* karena *tidak* menggunakan*ConnectionString.*

Data Control dan *ADO Datu Control* merupakan kontrol VB yang mempunyai konsep yang sama, yaitu *"datacontrols"* yang menghubungkan suatu sumber data kepada suatu kontrol pembatasan data. Kontrol-kontrol ini mempunyai penampakan tombol yang *sama* yaitu satu set dari empat tombol yang dapat digunakan untuk menuju pada awal *recordset,* akhir *recordset,* maju dan mundur diantara *recordset.* Menurut MSDN™ Library, *"both controls are stili included with Visual Basic ,for backward compatibiliy. However, because of the flexibility of ActiveX Data Objects (ADO), it's recommended that you create new database applications using the ADO Data control".* Hal yang sama juga diungkapkan oleh literatur lainnya yaitu "Kontrol *ADO* secara visual terlihat seperti kontrol data namun kontrol ini lebih fleksibel karena dapat membuat hubungan dengan data provider lain" (Wahana 2001:175).

DataGrid menampilkan dan dapat memanipulasi data dari suatu kelompok baris dan kolom yang mewakili *record dan field* dari sebuah obyek *Recordset. DataGrid control* terlihat sama seperti *Grid control. Kita* dapat mengatur properti *DataSource-nya* pada sebuah *Data control,* sehingga *kontrol* ini secara otomatis mengisi kolom dan barisnya dari sebuah obyek *Recordset Data control. DataGrid control* terdiri dari sejumlah kolom yang tetap dengan baris *yang tak* terbatas. Sebuah *DataGrid control* dapat memiliki sebanyak mungkin baris dan maksimal 32767 kolom. Waktu mendesain, kita dapat mengatur panjang kolom dan tingginya baris, serta dapat membuatnya tidak dapat dilihat oleh *user.* Kita juga dapat mencegah *user* untuk mengubah formatnya ketika dijalankan.

DataCombo control merupakan sebuah *data-bound combo box yang* secara otomatis dihasilkan dari suatu *field* dalam sebuah *data source, dan* dapat

mengubahfield tersebut dalam tabel yang berhubungan dari *data source* lainnya. *DataCombo control* dapat bekerja lebih optimal dengan *ActiveX Data Objects (ADO).*

Laporan dapat dibuat dengan *Microsoft Data Report Designer,* yaitu penghasil laporan data serbaguna yang memiliki kemampuan untuk membuat laporan terpadu, sehingga kita dapat membuat laporan dari beberapa tabel yang berbeda **jika** digunakan dalam hubungan dengan suatu sumber data, seperti *Data Environment designer. Designer* ini dapat secara cepat dan mudah **untuk** membuat bangunan *hierarchical recordsets* pada **waktu** desain. Pemrograman yang menghasilkan *DataEnvironment object* memerlukan pengetahuan tentang *ADO* dan karakteristikdari *hierarchical recordsets.* Frase *"DataEnvironment designer"* menunjuk pada *designer* dan bagian-bagiannya, sedangkan frase *"Data Environment object"* menunjuk pada hasil obyek pemrograman dari *Data Environment designer.* Obyek *Data Report* merupakan obyek yang diprogramkan untuk menampilkan *Data Report designer. Data Report* menghasilkan laporan dengan menggunakan *record* dari sebuah *database.* **Cara** penggunannnya adalah membentuk sumber data seperti *Microsoft Data Environment* untuk mengakses sebuah *database,* kemudian menentukan sumber data dari obyek *Data Report* pada properti *Data Source* dan menentukan anggota data dari obyek *Data Report* pada properti *Data Member.* Lalu tambahkan kontrol dantentukan properti *Data Member* dan *Data Field-nya.* Gunakan metode *Show* untuk menampilkan *Data Report* pada waktu dijalankan. Sebagai tambahan dalam membuat laporan yang dapat dicetak, **kita** juga dapat mengekspor laporan dalam bentuk HTML, atau text *files. Data report designer* memiliki beberapabagian, yaitu: (Microsoft 2000)

a. *Drag-and-Drop Functionalityfor Fields_menarik fields* dari *Microsoft Data Environment designer* ke *Data Report designer.* Ketika **dilakukan,** secara otomatis **VB** membuat sebuah kontrol *text box* pada *data report* dan **mengatur** *DataMember* dan *DataFieldproperties darifield* yang ditarik.

b. *Toolbox Controls — Data Report designer* mempunyai **suatu** set kontrol tersendiri. Ketika sebuah *Data Report designer* ditambahkan ke proyek, kontrol-kontrolnya secara otomatis dibuat pada sebuah Toolbox *tab* **baru** yang bernama *DataReport.* Kebanyakan kontrolnya berfungsi **sama** dengan kontrol

VB lainnya seperti *Label, Shape, Image, TextBox* dan *Line control.* Kontrol keenam yaitu *Function control,* secara otomatis menghasilkan satu dari *empat* jenis informasi seperti *Sum, Average, Minimum,* atau *Maximum.*

c. *Print Preview* — Menampilkan laporan dengan menggunakan *Show method.* Laporan dihasilkan dan ditampilkan dalam layar *window* monitor. Sebuah printer harus dipasang pada komputer untuk menunjukkan laporan dalam *print preview mode.*

d. *Print Report* — Mencetak sebuah laporan dengan terprogram dengan memanggil *PrintReport method.* Ketika laporan dalam tampilan layar *window* monitor, kita juga dapat mencetak dengan menekan tombol printer pada *toolbar.*

e. *File Export* — Mengekspor informasi laporan dengan menggunakan *ExportReport method* dalam bentuk *HTML* dan *text.*

f *Export Templates* — Kita dapat membuat suatu kumpulan file *templates yang* digunakan dengan *ExportReport method.* Hal ini berguna untuk mengekspor laporan dalam berbagai bentuk tipe laporan.

Berikut ini ada beberapa petunjuk yang harus diperhatikan dalam pemrograman untuk menghasilkan laporan yang membutuhkan suatu parameter. Untuk menciptakan sebuah *parameterized query,* kita dapat membuat proyek baru dan menambahkan sebuah *Data Environment designer,* kemudian menghubungkan dengan suatu *database* dalam *Connection object* dan menambah *Command* yang dapat dinamai *Command Query* serta cari propertinya Pada *General tab,* tekan *SQL Statement* dan ketikkan seperti contoh berikut ini: *SELECT * FROMAuthors WhereAu_ID =?* Lalu tekan *Parameters tab* dan tekan *OK,* maka sebuah *Input parameter* telah diciptakan. Kemudian buat kontrol *TextBox* dan *CommandButton* pada formulirya, dan tambahkan contoh kode seperti berikut ini: **(Microsoft 2000)**

```
Option Explicit
Private Sub Commandl–Click()
' You must close the recordset before changing the parameter,
    If DataEnvironmentl.rsCommandQuery.State = adStateOpen Then
    DataEnvironment/ rsCornmandQuery.Close
    End If
' Reopen the recordset with the input parameter supplied by the TextBox
control
```

```
DataEnvironmentl.CommandQuery Textl.Text With Text2
.DataField = "AU_LName"
.DataMember = TommandQuery··
Set .Datasource = DataEnvironmentl
End With
End Sub

Private Sub Form_Load
' Supply a default value.
TextI.Text = "I72-32-1172"
'Change the CommandButton caption.
Commandl.Caption = "Run Query"
End Sub
```

Setelah itu jalankan proyeknya dan tekan tombol yang ada. Kita dapat mengubah *text in Textl* untuk menjalankan *query* lainnya.

Program aplikasi bisnis memiliki keterbatasan dalam kemampuannya untuk mengakses data, sehingga seharusnya data terpisah dengan program. Alasannya adalah program tidak perlu diubah ketika terjadi perubahan data karena hanya file data yang perlu diubah, dan program yang berbeda dapat berbagi file data yang sama, sehingga menghapuskan kebutuhan untuk meniru data dan mengurangi adanya kelebihan data yang tidak diperlukan. Hal ini bertujuan agar pengelolaan data dapat menjadi lebih efisien karena kelebihan data mempunyai beberapa dampak yang negatif seperti penyimpanan data yang berlebihan, memerlukan pembahaman data pada beberapa tempat, dan ada kemungkinan tidak semua data telah diperbaharui sehingga menghasilkan nilai yang tidak sama untuk data yang sama.