BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Menurut Jogianto (2000) sistem adalah :

Suatu sistem dapat didefinisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan.

Suatu sistem dapat terdiri dari sistem-sistem bagian (subsystems). Subsistem perangkat keras (hardware) dapat terdiri dari alat masukan, alat pemroses, alat keluaran dan simpanan luar. Subsistem-subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai. Interaksi dari subsistem-subsistem sedemikian rupa, sehingga dicapai. Interaksi dari subsistem-subsistem sedemikian rupa, sehingga dicapai suatu kesatuan yang terpadu atau terintegrasi (integrated).

Terdapat dua kelompok pendekatan yang digunakan dalam mendefinisikan sistem, yaitu (Al-Bahra Bin Ladjamudin, 2005) :

1. Lebih menekankan pada prosedur yang digunakan dalam sistem, diantaranya Jery FitzGerald, Ardra F. FitzGerald, Warren D. Stallings mendefinisikan sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

2. Lebih menekankan pada elemen atau komponen penyusun sistem, diantaranya :
   a. McLeod berpendapat sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan.
   b. Robert G. Murdick mendefinisikan sistem sebagai seperangkat elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan bersama.

Pengertian tentang sistem pertama kali dapat diperoleh dari definisinya. Dengan demikian definisi ini akan mempunyai peranan yang penting di dalam pendekatan untuk mempelajari suatu sistem. Pendekatan sistem yang merupakan kumpulan elemen-elemen atau komponen-komponen atau subsistem-subsistem merupakan definisi yang lebih luas. Definisi ini lebih banyak diterima karena kenyataannya suatu sistem dapat terdiri dari beberapa subsistem atau sistem-sistem bagian. Komponen-komponen atau subsistem-subsistem dalam suatu sistem tidak dapat berdiri sendiri. Komponen-komponen atau subsistem-subsistem saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran tersebut dapat tercapai.

2.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary),* lingkungan sistem (*environments),* penghubung (*interface),* masukan (*input),* keluaran (*output),* pengolah (*process)* dan sasaran (*objectivies)* atau tujuan (*goal).*

2.2 Konsep Dasar Sistem Informasi

Sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdik.(Jogiyanto, 2000).

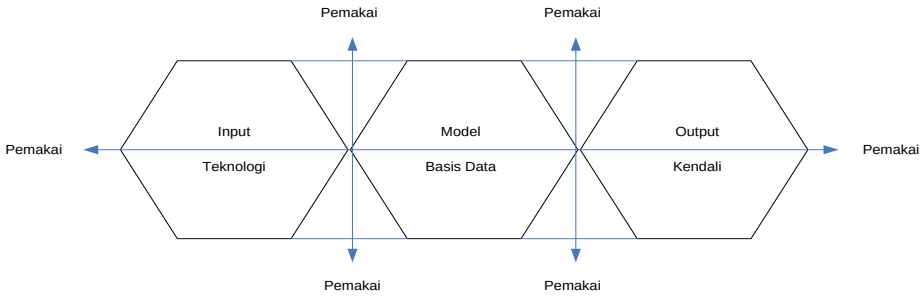
Informasi dalam sebuah organisasi sangat penting peranannya. Suatu sistem yang kurang mendapatkan informasi akan menjadi lemah dan akhirnya berakhir. Menurut Gordon B. Davis, informasi itu sendiri dapat didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya untuk mengambil keputusan masa kini maupun masa yang akan datang (Al-Bahra Bin Ladjamudin, 2005).

Sumber dari informasi adalah data. Data merupakan kenyataan yang menggambarkan suatu kejadian dan kesatuan nyata. Kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu. Kesatuan nyata adalah berupa suatu objek

nyata seperti tempat, benda dan orang yang sesungguhnya ada dan terjadi (Al-Bahra Bin Ladjamudin, 2005).

2.2.1 Komponen Sistem Informasi

Sistem informasi memiliki sejumlah komponen tertentu. Menurut John Burch dan Gary Grudnitski, komponen-komponennya disebut dengan istilah blok bangunan yang kemudian dibagi menjadi blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*data base block*) dan blok kendali (*control block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasaran (Jogiyanto HM, 2005).



Gambar 2.1. *Building Block*

1. Blok Input

   Input mewakili data yang masuk ke dalam sistem informasi. Input merupakan metode-metode dan media untuk menangkap data yang akan dimasukkan, yang berupa dokumen-dokumen dasar.

2. Blok Model

   Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Output

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi

   Teknologi merupakan kotak alat dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan serta bantuan pengendalian dari sistem secara keseluruhan.

5. Blok Basis Data

   Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan tersimpan dalam perangkat keras komputer dan

digunakan perangkat lunak untuk memanipulasinya yang disebut *Database Management System* (DBMS)

6. Blok Kendali

Pengendalian perlu dirancang dan diterapkan dan meyakinkan hal-hal yang dapat merusak sistem, dapat mencegah terjadinya kesalahan-kesalahan yang dapat langsung diatasi.

2.2.2 Pengembangan Sistem Informasi

Dalam pengembangan sistem informasi ada beberapa metode pendekatan yang dapat digunakan, yaitu (Jogiyanto HM, 2005):

1. *Top Down*

Metode *top down* mampu mengembangkan suatu aliran informasi yang baik dan menyusun sistem informasi yang sesuai untuk mendukung aliran informasi tersebut. Langkah-langkah yang harus dilakukan untuk menggunakan metode ini adalah :

a. Analisis tujuan, lingkungan dan pembatas.
b. Identifikasi fungsi-fungsi atau aktivitas-aktivitas di dalam organisasi.
c. Identifikasi proses-proses pengambilan keputusan dan tindakan-tindakan.
d. Identifikasi jenis informasi yang dibutuhkan untuk setiap keputusan dan tindakan.

e. Pengelompokan proses pengambilan keputusan dan kebutuhan informasi kedalam sub-sub sistem dan modulnya.

f. Menentukan prioritas untuk pengembangan basis data, subsistem dan modulnya.

2. Bottom Up

Metode ini berangkat dari pengertian bahwa pada dasarnya pengolahan data adalah pengolahan proses-proses transaksi dan proses peremajaan file. Oleh sebab itu, sistem informasi pada awalnya disusun dengan mulai memenuhi modul-modul operasi pada proses transaksi dan peremajaan file. Kemudian dilengkapi agar dapat menunjang proses perencanaan, pengendalian, dan pengambilan keputusan serta modul lain yang dianggap perlu. Langkah-langkah yang harus dilakukan untuk menggunakan metode ini adalah :

a. Menyediakan file yang terpisah untuk setiap pemakaian yang menunjang operasi, terutama untuk proses-proses transaksi, peremajaan file dan laporan sederhana.

b. Mengintegrasikan file-file tersebut ke dalam basis data dan memakai sistem manajemen basis data untuk mengatur penyimpanan dan jangkauan data.

c. Menambahkan beberapa modul keputusan dan perencanaan yang terpisah untuk melengkapi kegiatan perencanaan, pengendalian manajemen dan pengambilan keputusan.

d. Mengintegrasikan modul-modul tambahan tersebut ke dalam basis data serta mengembangkan model-model yang sudah ada.

e. Penambahan data-data dan model-model perencanaan strategi.

3. Kombinasi

Metode ini pertama kali diterapkan oleh IBM dengan nama *Business System Planning* (BSP). Lahirnya metode ini berawal dari kesalahan-kesalahan yang berkaitan dengan sistem informasi tradisional dari dalam perusahaan IBM maupun dari pengalaman atas hal yang sama pada perusahaan lain. Kesalahan tersebut antara lain tidak dikelolanya perawatan (*maintenance*) dan antar muka (*interface*) sub-sub sistem dalam perusahaan. Hal ini semakin terasa dengan dibentuknya Departemen Perencanaan dan Pengendalian, sistem yang bertanggung jawab terhadap pemrosesan data di seluruh bagian perusahaan, yang meliputi bagian engineering, manufacturing, marketing, dan sebagainya.

Dalam prakteknya hanya sedikit koordinasi yang diberikan kepada keterkaitan data antar bagian dalam perusahaan. Kebanyakan aktivitas pengolahan dibatasi pada lokasi dan unit dalam masing-masing bagian. Konsekuensinya, masing-masing bagian mengoperasikan sistemnya sendiri-sendiri, akibatnya banyak terjadi redundansi data. Dari hal itu muncullah perbaikan yang kemudian mengarahkan pada kebutuhan bisnis secara menyeluruh, bukan pada kebutuhan per bagian, yang meminta konsekuensi harus ada penanggung jawab data yang tetap dan terpusat

baik dalam perencanaan maupun pengendalian sistem informasi. Selain tujuan tersebut, tujuan lain yang ingin dicapai sehubungan dengan perbaikan sistem informasi tersebut adalah :

a. Tidak ada suatu data yang sama dibuat oleh lebih dari satu bagian, tetapi data yang sumbernya tunggal ini tetap dapat dipakai secara bersama-sama oleh beberapa bagian yang berbeda.
b. Data yang tergantung pada struktur organisasi atau hal-hal yang bersifat organisasional.
c. Penggunaan metode BSP ini yaitu dalam perencanaan sistem informasi digunakan metode *top down* dan pada saat perancangan dan implementasi digunakan metode *bottom up*.

2.3 Pengertian Perancangan Sistem

Pada tahap perancangan sistem ini akan dirancang suatu sistem dalam suatu bagan yang menunjukkan prosedur-prosedur dari sistem tersebut. Perancangan sistem dapat didefinisikan sebagai berikut :

1. Menurut Robert J. Verzello / John Reuter III mendefinisikan perancangan sistem sebagai tahap setelah analisis dari siklus pengembangan sistem; pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi; menggambarkan bagaimana suatu sistem dibentuk.

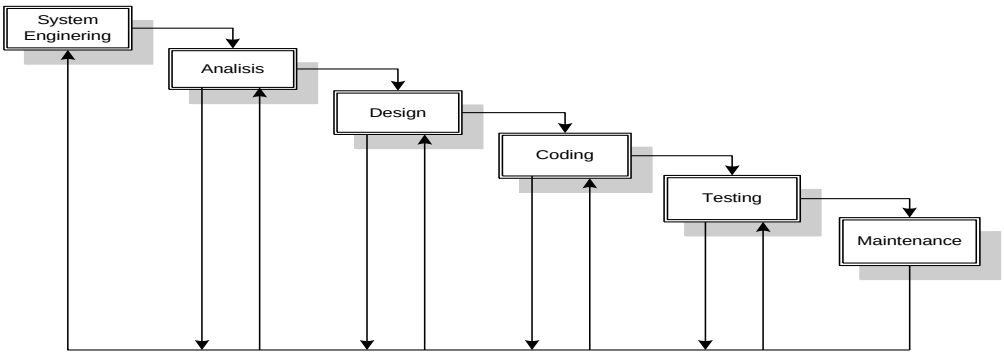
2. Menurut John Burch dan Gary Grudnitski, perancangan sistem adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.

3. Menurut George M. Scott, perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan, tahap ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem.

Tahap perancangan sistem mempunyai dua maksud atau tujuan utama (Jeffry L. Whitten, Lonnie D. Bentley, Thomas I.M. Hon, 1986), yaitu :

1. Untuk memenuhi kebutuhan pemakai sistem.
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik lainnya yang terlibat.

Metoda yang digunakan dalam proses *software* menggunakan *Metoda Waterfall*, model ini adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Berikut ini ada dua gambaran dari waterfall model. Sekalipun keduanya menggunakan nama-nama fase yang berbeda, namun sama alam intinya.Fase-fase dalam Waterfall Model menurut referensi R. S. Pressman:



Gambar 2.2

Pengembangan Sistem Dengan Model Waterfall

("Software Engineering *A Practitioner's Approach* " *Roger S. Pressman,Ph.D, Mc Graw Hill, 1997*)

Model yang terdapat beberapa proses sebagai berikut :

1. Rekayasa Sistem (*System Engineering)*

Proses menentukan segala hal yang diperlukan dalam pengembangan sistem.

2. Analisis *(analysis).*

Penguraian dari sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud megidentifikasikan dan mengevaluasikan permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Tahap analisa merupakan tahap yang sangat penting, karena jika terjadi kesalahan didalam tahap ini, maka akan menyebabkan kesalahan juga di tahap selanjutnya.

3. Perancangan *(design).*

   Perancangan ini terdiri dari dua bagian, yaitu : perancangan awal berupa transformasi keperluan kedalam arsitektur data dan perangkat lunak, kemudian dilanjutkan dengan perancangan rinci berupa perbaikan model arsitektur dalam pembuatan struktur data dan algoritma secara rinci dari perangkat lunak.

4. Pengkodean *(coding).*

   Pada tahap ini dilakukan proses penerjemahan dari keperluan data atau pemecahan masalah yang telah dirancang kedalam bahasa pemrograman komputer yang ditentukan.

5. Pengujian (*Testing*)

   Setelah program selesai dibuat, maka tahap berikutnya adalah uji coba program tersebut.

6. Pemeliharaan *(maintenance).*

   Tahap ini merupakan tahap terakhir dimana perangkat lunak yang sudah selesai dan dapat mengalami perubahan-perubahan atau penambahan-penambahan sesuai dengan *user*.

2.3.1 *Database Server*

Sebuah *database server* adalah sebuah pengolahan basis data yang dapat mengelola data dalam jumlah yang cukup besar dalam lingkungan *multi-user* secara konkuren (pada saat yang sama). Sebuah *database server* dibutuhkan kemampuan pengolahan data yang tinggi sehingga sebuah *database server* haruslah sebuah sistem (perangkat lunak) yang berdiri sendiri dan terpisah dengan program aplikasi atau aplikasi basis data.

2.3.2 Sistem *Client Server*

Sistem *client-server* ditujukan untuk mengatasi kelemahan-kelemahan yang terdapat pada sistem tersentralisasi (baik sentralisasi pada aplikasi dan basis data maupun sentralisasi hanya pada basis data). Kelemahan pada bentuk sistem tersentralisasi yang pertama, yaitu beratnya beban server yang harus menangani semua proses, diatasi dengan membagi beban itu menjadi 2 bagian : *client* (yang menjalankan aplikasi basis data) pada mesin yang berbeda. Sedang kelemahan pada bentuk sistem tersentralisasi yang kedua, yaitu padatnya lalulintas data antara server dan *work-station* diatasi dengan mekanisme transfer data yang lebih efisien.

Sistem ini terdiri atas dua komponen (mesin) utama, yaitu *client* dan *server*. Client berisi aplikasi basis data dan *server* berisi DBMS dan basis data. Setiap aktivitas yang dikehendaki para pemakai akan lebih dulu ditangani oleh *client*. *Client* selanjutnya akan mengupayakan agar semua proses sebisa mungkin ditangani sendiri. Jika ada proses yang harus melibatkan data yang tersimpan pada basis

data, barulah *client* melakukan kontak dengan *server*. Katakanlah ada kebutuhan untuk mendapatkan baris-baris data *(row)* dengan kriteria tertentu *(proses query)* dari tabel data bernama T. pada bentuk sistem tersentralisasi yang kedua, kebutuhan ini dijawab dengan mentransfer lebih dulu semua baris data *(row)* yang ada di tabel T tersebut dari *server* ke *work-station*, barulah kemudian di *work-station* dilakukan pemilihan baris data *(query)* terhadap tabel T tersebut. Sementara pada sistem *client-server*, untuk memenuhi kebutuhan itu *client* akan mengirimkan *message* (perintah) *query* pengambilan data. Selanjutnya, *server* yang menerima *message* tersebut akan menjalankan *query* tersebut (melakukan pencarian baris data terhadap tabel T) dan hanya hasilnya (jadi tidak seluruh isi tabel T yang akan dikirimkan kembali ke *client*. Dengan begitu transfer datanya jauh lebih efisien.

Ada 2 macam bentuk sistem *client-server* yaitu :

1. Lebih sederhana dan dapat diterapkan pada sebuah jaringan komputer lokal (LAN), dimana fungsi *client* (untuk menangani sebagian besar proses pengolahan data seperti perhitungan, perulangan, pembandingan, dll) dan fungsi *work-station* untuk menangani interaksi dengan pemakai, menerima data masukan dan menayangkan hasil pengolahan) disatukan. Adanya pemilahan fungsi *client* dan fungsi *server*, disamping meningkatkan kompleksitas tersendiri dalam pembangunan aplikasi *(system)* secara keseluruhan, juga menimbulkan kelemahan lain, yaitu aktivitas pemasangan apliksi *(deployment)* yang tidak praktis. Jika terjadi perubahan atau perbaikan

aplikasi basis data, maka kita harus mengulang pekerjaan instalasi di semua mesin *client* yang digunakan. Karena itu pekerjaan ini menjadi sangat tidak praktis jika harus diterapkan pada sistem jaringan yang lebar *(WAN).*

2. Untuk itulah, varian sistem *client-server* yang kedua dapat kita terapkan. Pada varian sistem ini aplikasi basis data tidak ditempatkan di setiap *work-station* tetapi dipasang di setiap *client* yang jumlahnya jauh lebih sedikit. *Client* ini ditempatkan pada lokasi yang lebih dekat pada kumpulan *worksation*. Jadi setiap *client* dan sejumlah *work station* membentuk sebuah LAN tersendiri. Karena *client* ini merupakan basis tempat aplikasi basis data disimpan dan turut menangani proses-proses dalam aplikasi, maka bagi *workstation*, *client* ini dapat dipandang sebagai *server* aplikasi. Tidak sebagaimana *work-station* yang dapat diaktifkan dan di*non*aktifkan oleh para pemakai, *client* tersebut (sebagaimana juga DBMS server) harus selalu dalam keadaan aktif dan terkoneksi dalam sebuah jaringan yang lebih besar (WAN). Dengan begitu, tahap instalasi aplikasi (khususnya untuk instalasi hasil perbaikan aplikasi) dapat dilakukan secara *remote* (instalasi jarak jauh) dari lokasi lain, sehingga kelemahan dari sisi instalasi dapat diatasi.

2.4 Pengertian Persediaan

Setiap perusahaan, baik itu perusahaan jasa maupun perusahaan manufaktur selalu memerlukan persediaan. Tanpa adanya persediaan para pengusaha akan dihadapkan pada resiko bahwa perusahaannya pada suatu waktu tidak dapat memenuhi keinginan para pelanggannya.

Hal ini bisa saja terjadi karena tidak selamanya barang atau jasa tersedia setiap saat. Berarti, pengusaha akan kehilangan kesempatan memperoleh keuntungan yang seharusnya ia dapatkan. Jadi persediaan sangat penting untuk setia perusahaan, baik yang menghasilkan barang maupun jasa.

Pengertiaan persediaan adalah bahan-bahan, bagian yang disediakan, dan bahan-bahan dalam proses yang terdapat dalam perusahaan untuk proses produksi, serta barang-barang jadi atau produk yang disediakan untuk memenuhi permintaan dari konsumen atau pelanggan setiap waktu.(Freddy Rangkuti, 2004).

2.4.1 Tujuan diadakannya persediaan

Pada setiap perusahaan manufaktur harus selalu memiliki persediaan bahan baku. Tanpa adanya persediaan,perusahaan akan dihadapkan pada resiko tidak terpenuhinya jadwal produksi yang telah ditetapkan karena tidak selamanya persediaan tersebut tersedia setiap saat dan pada tempat yang diinginkan.

Jadi mempertahankan suatu jumlah persediaan bahan baku tertentu dalam perusahaan manufaktur pada dasarnya bertujuan untuk mempelancar atau mempermudah jalannya operasi perusahaan dan menghilangkan resiko tertundanya proses produksi, terutama jika terjadi kondisi-kondisi sebagai berikut:

1. Terjadi keterlambat datangnya barang atau bahan-bahan yang dibutuhkan.
2. Barang atau bahan-bahan yang dipesan tidak memenuhi kualitas yang diinginkan, sehingga harus dikembalikan.
3. Memberikan pelayanan kepada pelanggan dengan sebaik-baiknya dimana keinginan pelanggan pada suatu waktu dapat dipenuhi dengan memberikan jaminan tetap tersedianya barang tersebut.

2.4.2 Metode penilaian persediaan

Dalam menilai suatu persediaan ada beberapa cara dapat digunakan, diantaranya:

1. Cara First-In, First- Out (FIFO-Method)

   Cara ini didasarkan atas asumsi bahwa harga pokok barang yang digunakan pertama adalah yang dibeli pertama. Dengan demikian persediaan akhir dinilai menurut harga pembelian barang yang terakhir masuk

2. Cara Last-In, First-Out (LIFO-Method)

   Cara ini didasarkan atas asumsi bahwa harga barang yang telah terjual dinilai menurut harga pembelian barang yang terakhir masuk. Jadi persediaan akhir dinilai berdasarkan harga pembelian barang terdahulu.

3. Cara Rata-rata tertimbang (Weighted Average Method)

   Cara ini didasarkan atas harga rata-rata tertimbang. Harga tersebut dipengeruhi oleh jumlah barang yang diperoleh pada masing-masing harganya. Dengan demikian persediaan dinilai berdasarkan harga rata-rata.

Kebijakan pemilihan metode mana yang akan digunakan oleh perusahan tergantung pada pemimpin perusahaan dengan mempertimbangkan situasi dan kondisi yang berlaku.

2.4.3 Sistem pencatatan persediaan

Secara umum kita hanya mengenal 2 sistem pencatatan untuk persediaan, yaitu :

1. Sistem persediaan perpeptual (perpeptual inventory system)

   Sistem ini mencatat persediaan secara berkesinambungan. Setiap terjadi pembelian dan pengeluaran barang selalu langsung dicatat ke perkiraan persediaan. Harga pokok penjualan langsung dihitung pada saat terjadi pengeluaran barang. Jumlah persediaan barang yang ada dapat langsung diketahui dari catatan dan pengeluaran barang tanpa harus melakukan perhitungan secara fisik.

2. Sistem persediaan periodik (periode inventory system)

   Sistem ini tidak mengikuti mutasi persediaan secara berkesinambungan. pembelian akan dicatat ke dalam perkiraan pembelian. Dan pada pengeluaran barang tidak diadakan perhitungan harga pokok penjualan maupun pengurangan terhadap jumlah persediaan yang ada.

Dari kedua sistem pencatatan persediaan tersebut biasanya dalam penggunaannya tergantung dari kebijakan pimpinan perusahaan dalam memilihnya.

2.5 Perangkat Lunak Pendukung

Perangkat lunak yang penulis gunakan dalam membuat program aplikasi ini adalah SQL Server 2000 sebagai pengolah *database* dan pembuatan program aplikasi dengan menggunakan Borland Delphi 7.0.

2.5.1 Sejarah Borland Delphi 7.0

Ide munculnya Delphi sebenarnya berasal dari bahasa pemrograman Pascal. Bahasa pemrograman Pascal diciptakan tahun 1971 oleh ilmuwan dari Swiss, yaitu Niklaus Wirth. Nama Pascal diambil dari ahli matematika dan filsafat dari Prancis, yaitu Balaise Pascal (1623 – 1662).

Beberapa versi Pascal diantaranya Turbo Pascal yang muncul pertama kali hanya dapat dijalankan di DOS yang dirilis oleh Borland International Incorporation tahun 1983. dalam perkembangan selanjutnya dirilis Turbo pascal yang berjalan di windows 3.x, yaitu Turbo Pascal for Windows. Tahun 1992, Borland International menggabungkan Turbo Pascal for DOS dengan Turbo Pascal for Windows yang kemudian dikenal dengan nama Borland Pascal versi 7. tahun 1993 Borland International mengembangkan bahasa Pascal yang bersifat visual, yang hasilnya pada tahun 1995 dirilis Borland Delphi.

Sekitar satu tahun sesudahnya, yaitu tahun 1996 Borland Internasional merilis Borland Delphi 2.0 yang dikhususkan untuk sistem operasi Windows 95 dan Windows NT. Tahun 1997 dirilis Borland Delphi 3.0, tahun 1999, dikeluarkan Borland Delphi 6.0 dan sekarang sudah sampai dengan Borland Delphi 7.0 yang telah dipromosikan sebagai salah satu bahasa pemrograman.

Borland Delphi adalah sebuah aplikasi untuk pengembangan yang memanfaatkan keistimewaan konsep-konsep antar muka dalam grafik Microsoft Windows. Aplikasi yang dihasilkan Borland Delphi berkaitan erat dengan windows itu sendiri sehingga butuh pengetahuan bagaimana cara kerja windows bila dipandang dari kacamata *user*. Dengan memahami windows maka pemrogram dapat menghasilkan aplikasi yang lebih efektif.

Borland Delphi merupakan perangkat lunak yang digunakan untuk menyusun program aplikasi yang berdasarkan bahasa pemrograman Pascal. Dalam merancang suatu program aplikasi, Borland Delphi sangat mudah. penggunaannnya karena telah disediakan komponen-komponen yang siap pakai, sehingga akan menghemat penulisan program.

Borland Delphi memiliki fasilitas IDE (*Integrated Development Environment*). Dengan IDE, fungsi-fungsi yang diperlukan selama penulisan program telah disediakan dalam suatu tampilan. Untuk menuliskan kode program misalnya, tinggal mengklik objek yang akan diprogram kemudian dituliskan program pada kode editor yang telah terbuka dan siap pakai bila diperlukan.

2.5.1.1 Pengenalan Borland Delphi 7.0

Saat pertama kali Borland Delphi 7.0 dipanggil maka akan muncul empat windows yaitu :

1. Windows Utama

   Terdapat tiga buah elemen yang masing-masing memiliki fungsi khusus yaitu *menu bar*, *speed bar* dan *component pallete*

   a. *Menu Bar*

   Untuk mengatur semua window yang ada di lingkungan kerja Delphi, juga untuk mengelola proses desain aplikasi.



Gambar 2.3. *Menu Bar*

   b. *Speed Bar*

   Menyediakan akses yang cepat bagi operasi-operasi yang sering digunakan, seperti membuka file, menyimpan file, fasilitas *cut* and *paste*, memilih *window* di dalam Delphi, eksekusi program aplikasi, serta membuka dan memilih *form* dalam unit.



Gambar 2.4. *Speed Bar*

c. *Component Pallete*

Digunakan untuk memilih objek yang digunakan dalam pembuatan aplikasi dan kemudian meletakkan objek tersebut pada rancangan *form*.



Gambar 2.5. *Component Pallete*

Component pallete adalah sebuah katalog yang berisi objek-objek visual dan objek aplikasi lainnya yang dapat ditempatkan pada *form*.

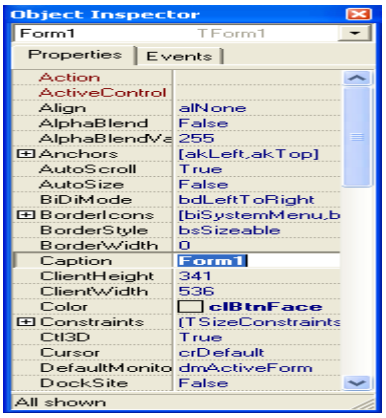
2. *Windows Object Inspector*

Terletak di bagian kiri bawah. Terdiri dari dua halaman yaitu halaman properties dan halaman *events*. Halaman *properties* menampilkan semua *properties* untuk objek yang terpilih dalam *windows form*. Halaman *events* mendaftar semua *events* yang akan ditanggapi oleh objek yang bersangkutan.

a. Halaman *Properties*

Halaman *properties* digunakan untuk menentukan *setting* suatu objek. Suatu objek biasanya mempunyai beberapa *properties* yang dapat diatur langsung dari lembar *properties* pada jendela *Object Inspector* maupun diatur lewat kode program. *Setting properties*

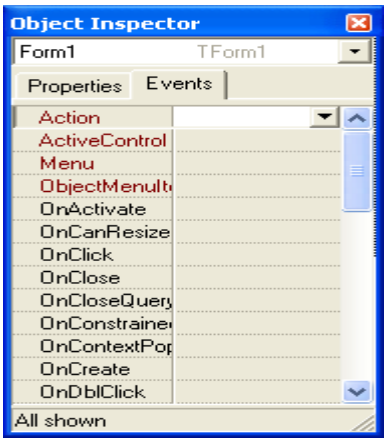
akan menentukan cara kerja dari objek yang bersangkutan saat program aplikasi dijalankan.



Gambar 2.6. *Object Inspector Properties*

b. Halaman *Events*

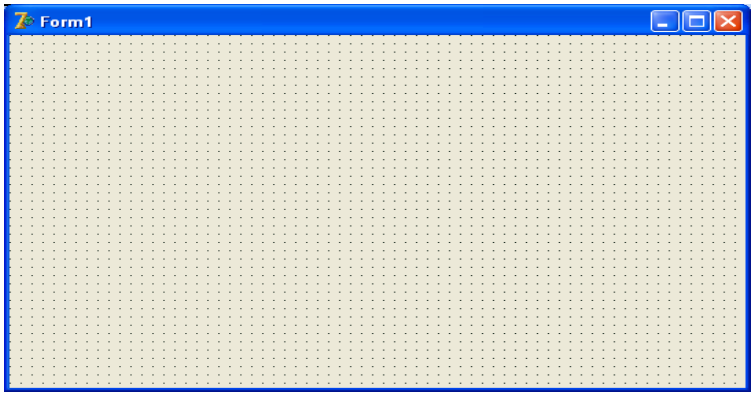
Event adalah peristiwa yang diterima oleh suatu objek, misalnya *klik*, *drag*, tunjuk dan lain-lain. *Event* yang diterima objek akan membuat Borland Delphi 7.0 menjalankan kode program yang ada di dalamnya.



Gambar 2.7.*Object Inspector Events*

3. *Windows Form*

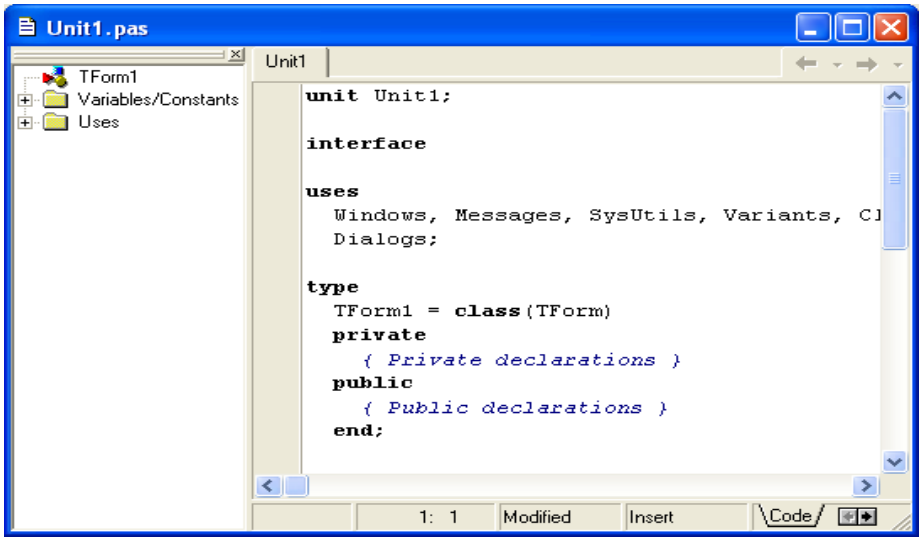
Form adalah suatu objek yang dipakai sebagai tempat bekerja program aplikasi. *Form* berbentuk jendela dan dapat dibayangkan sebagai kertas atau meja kerja yang dapat dilukisi atau diletakkan ke objek-objek lain dalamnya. Pada saat membuat suatu program aplikasi baru, biasanya ditampilkan dengan judul *Form* 1 dan terletak di bagian kanan atas *windows editor program*.



Gambar 2.8. *Windows Form*

4. *Windows Editor Program*

Dipakai untuk mengedit program objek Pascal. Memiliki beberapa halaman yang masing-masing menyimpan sebuah unit program, yang digunakan untuk menuliskan perintah yang harus dilakukan oleh komponen-komponen yang ada pada *form*.



Gambar 2.9. *Windows Editor Program*

2.5.2 SQL Server 2000

2.5.2.1. Sejarah SQL Server 2000

Pada tahun 1988 Microsoft mengeluarkan SQL Server yang dikembangkan bersama Sybase. Awalnya SQL Server dirancang untuk dapat berjalan pada *Platorm* OS/2.

Selama awal tahun 1990-an, Microsoft mulai mengembangkan versi baru dari SQL Server untuk dapat berjalan pada *Platform* NT. Selama tahap pengembangan ini, Microsoft memutuskan bahwa SQL Server bersifat *tightly coupled* (tingkat interaksinya tinggi) dengan sistem operasi NT.

Pada tahun 1993, Microsoft mengeluarkan Windows NT 3.1 dan SQL Server 4.2 untuk NT. Hal ini menunjukkan bahwa filosofi Microsoft untuk menggabungkan *high-performance database* dengan *interface* yang mudah digunakan denar-benar terbukti. Ini menyebabkan Microsoft dengan segera menjadi *vendor software relational database* yang populer.

Tahun 1994, Microsoft dan *Sybase* secara resmi mengakhiri kerjasama mereka. Pada tahun 1995, Microsoft mengeluarkan SQL Server 6.0. Rilis ini merupakan penulisan ulang teknologi utama SQL Server dengan peningkatan pada kinerjanya, replikasi *built-in,* dan pengiriman administrasi yang terpusat.

Tahun 1996 Microsoft mengeluarkan SQL Server 6.5. Versi ini membawa peningkatan dan perbaikan terhadap teknologi yang ada dengan menyediakan fitur-fitur baru. Tahun 1997, Microsoft mengeluarkan SQL Server 6.5 *Enterprise Edition*. SQL Server 7.0 yang merupakan engine database yang ditulis ulang secara lengkap dikeluarkan pada tahun1998.

Tahun 2000, Microsoft mengeluarkan SQL Server 2000. Versi ini dikembangkan berdasarkan SQL Server 7.0 Framework. Menurut tim pengembang SQL Server, perubahan pada *engine database* SQL Server 2000ini dirancang dan disiapkan untuk bertahan sampai dengan 10 tahun ke depan.

2.5.2.2 Fitur-fitur pada SQL Server 2000

Untuk mempermudah solusi manajemen data, SQL Server 2000 meningkatkan performansi dengan menghadirkan fitur-fitur yang menjadikannya pilihan terbaik sebagai *tools system management and tuning* dengan skalabilitas dan reliabilitas yang baik. Berikut adalah fitur-fitur pada SQL Server 2000 :

a. *Data Trabsformation Service*, yaitu layanan yang memudahkan transformasi, ekstraksi dan proses *load* data dari berbagai sumber.

b. *English query*, suatu layanan yang memungkinkan user untuk mengajukan query dalam bahasa Inggris sebagai pengganti penggunaan Multi Dimensional eXpression (DMX).

c. *OLAP Flexibility*, dimaksudkan untuk memanfaatkan *multiple dimension types* untuk menganalisa penyesuaian bisnis.

d. *Data mining*, dimaksudkan untuk menemukan pola dan kecenderungan serta prediksi terhadap bisnis aplikasi di masa yang akan datang.

e. *Indexing Views*, dimaksudkan untuk meningkatkan kinerja dengan menyimpan hasil query sehingga diharapkan dapat mengurangi *response time.*

f. *VI SAN (Virtual Ayatem Area Network)* yang akan meningkatkan kinerja keseluruhan sistem yang ada.

g. *Replication*, dimaksudkan agar dapat melakukan *merge, transactional dan snaphot replication* dengan berbagai sistem.

h. Mendukung Rich XML sehingga memudahkan penggabungan *back-end system* dengan transfer data melalui *firewall* menggunakan XML.

i. Terintegrasi dengan Windows Server System–BizTalk Server dan Commerce Server.

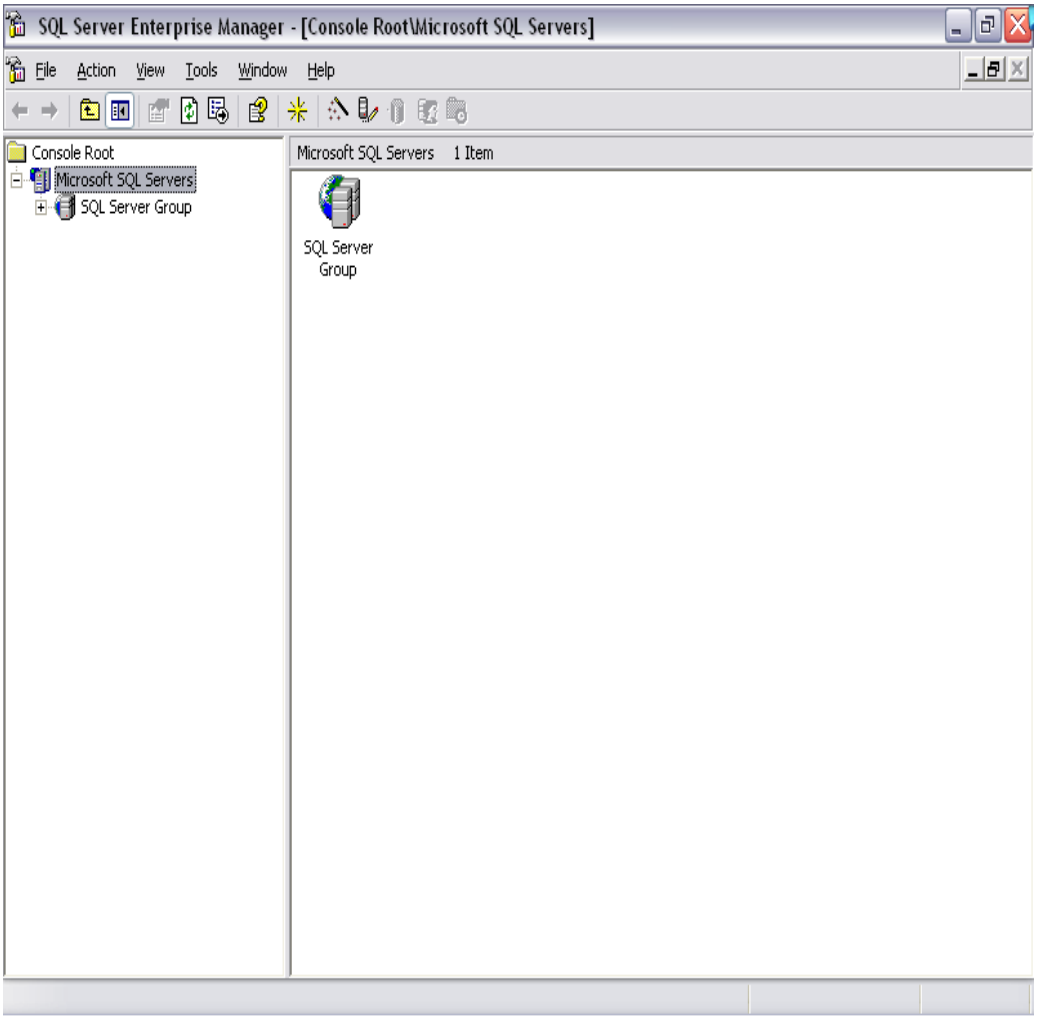
j. Memungkinkan untuk melakukan analisa data dari remote OLAP cubes yang dapat diakses web.

k. Dengan hanya menggunakan web tanpa tambahan pemrograman apapun, *user* dapat terkoneksi dengan SQL Server database dan OLAP cubes.

l. *Application Hosting*, dengan dukungan *multi-instance*, lebih dari satu aplikasi dapat berjalan pada satu *server*.

m. Menjamin sekuritas aplikasi yang berjalan pada ligkungan jaringan apapun karena adanya *role-based security* dan enkripsi terhadap *file* dan *network*.

n. Terintegrasi dengan Microsoft Office 2000 menggunakan *Pivot Table Service.*



Gambar 2.10. ***Enterprise Manag***