BAB II

TEORI DASAR

Bab ini akan menjelaskan mengenai teori dasar yang berisikan tori-teori yang mendukung dalam perancangan Sistem yang akan dibuat

2.1 Konsep Dasar Sistem

Setiap organisasi atau kegiatan tidak terlepas dari sistem. Sehingga suatu Departemen dapat dikatakan sebagai suatu sistem karena merupakan suatu organisasi dengan kelompok individu yang memiliki tujuan yang sama.

Pendefinisian sistem meliputi dua pendekatan prosedurnya dan pendekatan terhadap komponen atau elemennya.

Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut:

" Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu " **(Jogiyanto H.M., 1993:1).**

Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem. Prosedur (*procedure*) didefinisikan oleh Richard F. Neuschel sebagai berikut:

" Suatu prosedur adalah suatu urutan-urutan operasi klerikal (tulis menulis), biasanya melibatkan beberapa orang di dalam satu atau lebih

departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi " **(Jogiyanto H.M., 1993:1).**

Pengertian sistem dalam pendekatan terhadap elemen atau komponen yaitu sebagai berikut:

" Sistem adalah kumpulan elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu ". **(Jogiyanto H.M., 1993:2)**

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu sebagai berikut **(Jogiyanto H.M., 1993:3-5)**:

- Komponen-komponen (*Components*).

  Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

- Batas Sistem (*Boundary*).

  Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

- Lingkungan Luar (*Environment*).

  Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

- Penghubung Sistem (*System Interface*).

  Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya.

- Masukan Sistem (*System Input*).

  Masukan (*Input*) adalah energi yang dimasukkan ke dalam sistem.

- Keluaran Sistem (S*ystem Output*).

Keluaran (*Output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

- Pengolah Sistem.

  Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

- Sasaran Sistem.

  Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objective*).

2.2. Konsep Dasar informasi

Informasi ibarat darah yang mengalir di dalam tubuh suatu organisasi, sehingga informasi ini sangat penting di dalam suatu organisasi. Informasi dapat didefinisikan sebagai berikut:

" Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya ". **(Jogiyanto H.M., 1993 : 8)**

Sumber dari informasi adalah data. Data merupakan pernyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*Event*) adalah sesuatu yang terjadi pada saat yang tertentu. Kesatuan nyata (*Fact*) adalah berupa suatu objek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi.

2.2.1 Kategori Informasi

Dari kategorinya informasi dikelompokkan menjadi tiga macam yaitu **(Leman, 1998:2-3)**:

- Informasi Strategis, informasi ini digunakan untuk mengambil keputusan jangka panjang, mencakup informasi eksternal (tindakan pesaing, langganan), rencana perluasan perusahaan dan sebagainya.
- Informasi Taktis, informasi ini dibutuhkan untuk mengambil keputusan jangka menengah seperti informasi trend penjualan yang dapat dipakai untuk menyusun rencana-rencana penjualan.
- Informasi Teknis, informasi yang dibutuhkan untuk keperluan operasional sehari-hari, informasi persediaan stock, laporan kas harian.

2.2.2 Kualitas Informasi

Kualitas dari sistem informasi ditentukan oleh tiga hal yaitu **(Jogiyanto H.M., 1993:10)**:

- Akurat (*Accurate*), berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.
- Tepat pada waktunya (*Timeliness*), berarti informasi yang datang pada penerima tidak boleh terlambat.
- Relevan (*Relevance*), berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi untuk tiap-tiap orang bebeda.

2.2.3 Nilai Informasi

Nilai informasi (*value of information*) ditentukan dari dua hal, yaitu **(Jogiyanto H.M., 1993:11)**:

- manfaat
- biaya mendapatkannya

Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan didalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Sebagian besar informasi tidak dapat ditaksir keuntungannya dengan suatu nilai uang, tetapi dapat ditaksir nilai efektifitasnya.

2.3 Konsep Dasar Sistem Informasi

Sistem informasi pada suatu sistem meliputi pemasukan data (*input*), yang kemudian diolah melalui suatu model dalam pemrosesan data, hasil informasi akan ditangkap kembali sebagai input dan seterusnya, sehingga membentuk siklus informasi yang dapat diperoleh dari sistem informasi (*information system*), sebagai sistem khusus dalam organisasi untuk mengolah informasi tersebut.

Sistem informasi didefinisikan oleh Robert A. Leitch dan K.Roscoe Davis sebagai berikut:

" Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan

menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan ". **(Jogiyanto H.M., 1993 : 11)**

2.3.1 Komponen Sistem Informasi

Komponen sistem informasi terdiri dari (**Leman, 1998 :3**):

- *Hardware*, terdiri dari komputer, printer dan jaringan.
- *Software*, merupakan kumpulan dari perintah/fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer untuk melaksanakan tugas tertentu.
- Data, merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi.
- Manusia, yang terlibat dalam komponen manusia seperti operator dan lain sebagainya.
- Prosedur, seperti dokumen prosedur/proses sistem, buku penuntun operasional dan tehnis.

2.3.2 Kegiatan Sistem informasi

Kegiatan yang terdapat di dalam sistem informasi mencakup antara lain **(Leman, 1998 : 4)**:

- *Input*, menggambarkan suatu kegiatan untuk menyediakan data untuk diproses.
- Proses, menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah.

- *Output*, suatu kegiatan untuk menghasilkan laporan dari proses diatas tersebut.
- Penyimpanan, suatu kegiatan untuk memelihara dan menyimpan data.
- *Control*, suatu aktivitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan.

2.4 Desain Database

Database merupakan suatu hal yang sangat penting dalam pembuatan program, karena nantinya akan memudahkan pengaksesan *file* didalam program itu sendiri. Kesulitan utama dalam desain *database* adalah bagaimana agar *database* dapat memuaskan keperluan saat ini dan masa mendatang. Dalam perancangan model konseptual dilakukan disamping perancangan model phisik. Pada perancangan konseptual akan menunjukkan *entity* dan relasinya berdasarkan proses yang diinginkan oleh organisasi.

Definisi dari *database* adalah sebagai berikut:

***"Database adalah kumpulan file-file yang saling berelasi, relasi tersebut biasanya ditunjukkan dengan kunci dari tiap file yang ada "*(Ir.Harianto Kristanto, 1998:1)**. Satu *database* menunjukkan satu kumpulan data yang dipakai dalam satu lingkup Perusahaan, Instansi.

Penerapan *database* dalam sistem informasi disebut dengan sistem *database* (*database system*). Dengan sistem dasar *database* ini tiap-tiap orang atau bagian dapat memandang *database* dari beberapa sudut pandangan yang berbeda.

Dalam *database* biasanya terdiri dari: **(Ir.Harianto Kristanto, 1998:2-3)** *entity*, atribut, *data value*, *record* dan *file*. *Entity* adalah orang, tempat, kejadian atau konsep yang informasinya direkam. Atribut adalah setiap *entity* mempunyai atribut atau sebutan untuk mewakili suatu *entity*. Data value adalah data aktual atau informasi yang disimpan pada tiap data elemen yang saling berkaitan menginformasikan tentang suatu *entity* secara lengkap. *File* adalah kumpulan *record-record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama namun berbeda data valuenya.

Model *database* bukanlah pendekatan proses informasi seorang *programmer* aplikasi, tetapi merupakan kombinasi beberapa cara untuk memproses data untuk beberapa aplikasi. Pada model *database* penekanan tinjauan dilakukan pada struktur dan data relasi antar *file*. Tidaklah perlu dipikirkan tentang terapan dan operasi yang akan dilakukan pada *database*.

Pendekatan yang dilakukan pada model *database* adalah menggunakan model data *relational*. Pada model *database* dibagi menjadi dua buah teknik antara lain **(Ir. Harianto Kristanto, 1998:18)**:

- Teknik Normalisasi.
- *Teknik Entity Relationship*.

2.4.1 Teknik Normalisasi

Proses normalisasi merupakan proses pengelompokkan data elemen menjadi tabel-tabel yang menunjukkan *entity* dan relasinya.

Pada proses normalisasi selalu diuji pada beberapa kondisi, apakah ada kesulitan pada saat menambah/*insert*, menghapus/*delete*, mengubah/*update*, membaca/*retrieve* pada satu *database*. Bila ada kesulitan pada pengujian tersebut maka relasi tersebut dipecahkan pada beberapa tabel lagi atau dengan kata lain perancangan belumlah mendapat *database* yang optimal.

Ide dasar normalisasi berkaitan dengan kunci relasi dan dependensi antar kunci. Secara sederhana, relasi seharusnya hanya menyimpan fakta tentang kunci relasi. Karena itu penting untuk mengetahui lebih jauh mengenai normalisasi, ada beberapa konsep yang harus diketahui lebih dahulu yaitu (**Ir. Harianto Kristanto, 1998:19**):

- *Field*/Atribut Kunci.
- Ketergantungan Fungsi (*Functional Dependency*).

Selanjutnya akan dijelaskan istilah yang berkaitan dengan kedua konsep diatas, agar supaya memudahkan dalam merancang *database*.

- **Field/Atribut Kunci.**

Setiap file terdapat kunci dan file berupa *database field* atau satu set *field* yang dapat mewakili *record*. *Field* ini akan digunakan sebagai kunci dalam pencarian *field* lainnya. Pada dasarnya kunci (*key*) adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (*row*) dalam tabel secara unik. Artinya jika suatu atribut dijadikan sebagai kunci, maka tidak boleh ada dua atau lebih baris data dengan nilai yang sama untuk atribut tersebut.

Ada empat macam *key* yang dapat diterapkan pada suatu tabel yaitu (**Ir. Harianto Kristanto, 1998:19-21**):

- *Alternate Key* (Kunci Alternatif), merupakan kunci kandidat yang tidak dipakai sebagai *primery key*.
- *Candidate Key* (Kunci Kandidat), merupakan satu *atribute* atau satu set minimal *atribute* yang mengidentifikasikan secara unik suatu kejadian *spesific* dari *entity*.
- *Primary Key* (Kunci Primer), merupakan satu *atribute* atau satu set minimal *atribute* yang tidak hanya mengidentifikasikan secara unik suatu kejadian *spesific*, tapi juga dapat mewakili setiap kejadian dari suatu *entity*.
- *Foreign Key* (Kunci Tamu), merupakan satu *atribute* yang melengkapi satu *relationship* (hubungan) yang menunjukkan ke induknya.

- **Ketergantungan Fungsi (*Functional Dependency*)**

Dalam ketergantungan fungsi, biasanya kunci dari file (*atribut key*) dijadikan untuk menunjukkan sebagai fungsi dari suatu tabel relasi, misalnya nim sebagai atribut *key*, juga menunjukkan fungsi nama. Untuk lebih jelasnya ketergantungan fungsi dapat dilihat pada nama dibawah ini:

Nim ⟶ Nama atau Nama = f (Nim)

Yang berarti bahwa atribut nama hanya bergantung pada atribut nim. Hal ini dibutuhkan dari fakta : untuk setiap nilai nim yang sama, maka pasti nilai nama juga sama.

2.4.2 Teknik Entity Relationship

Relationship adalah hubungan atau asosiasi antara semua entitas dengan dirinya sendiri atau dengan entitas lainnya. Pada model data *relational* hubungan antar *file* direlasikan dengan kunci relasi (*relation key*), yang merupakan kunci utama dari masing-masing *file*. *Relationship* digambarkan sebagai garis yang menghubungkan entitas-entitas yang dipandang memiliki hubungan antara satu dengan lainnya. Relasi antara dua *file* atau dua tabel dapat dikategorikan menjadi tiga macam, yaitu **(Ir. Harianto kristanto, 1998:35)**:

1) *One to One Relationship*

   Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding satu. Seperti ada pelajaran privat dimana satu guru mengajar satu siswa dan satu siswa hanya diajar oleh satu guru. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakili dengan tanda panah tunggal.

2) *One to Many Relationship*

   Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik menjadi banyak lawan satu. Seperti pada sistem pengajaran di sekolah dasar dimana satu guru mengajar banyak siswa dan siswa hanya diajar oleg satu guru. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.

3) Many to Many Relationship

Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding banyak. Seperti sistem pengajaran di Perguruan Tinggi dimana satu guru mengajar banyak siswa dan siswa diajar oleh banyak guru. Hubungan tersebut digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda.

2.5 Pemrograman Delphi

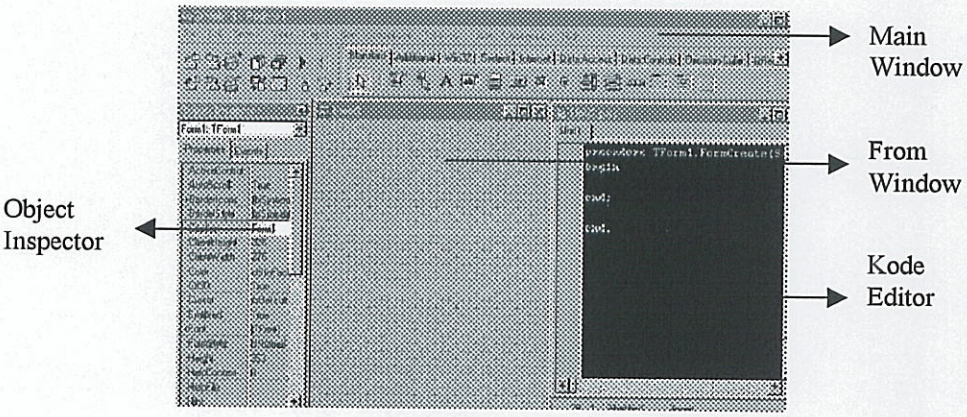
Delphi dapat didefinisikan sebagai berikut:

" perangkat lunak untuk menyusun program aplikasi yang didasarkan pada bahasa pascal, dan bekerja sama dalam lingkungan sistem operasi windows" **(Djoko Pramono, 1997:3). Dengan delphi akan merasakan begitu mudahnya menyusun program aplikasi, karena delphi menggunakan komponen-komponen yang akan menghemat penulisan program (Djoko Pramono, 1997:3).** Aplikasi yang dihasilkan delphi berkaitan erat dengan *window* itu sendiri sehingga dibutuhkan pengetahuan bagaimana cara keja dari *window* bila di pandang dari kacamata pemakai.

2.5.1 Lingkungan Kerja Utama Delphi

Secara umum dalam lingkungan kerja Delphi terdapat empat buah *window* yang dirancang sebagai sebuah aplikasi multi *window single document interface* (*SDI*) seperti dapat dilihat pada gambar 2-1 **(Joh Matcho dan David R.**

Faulkner, 1997:28). Satu *windows* utama dalam Delphi mengatur beberapa *window* yang saling berhubungan, yaitu **(Joh Matcho dan David R. Faulkner, 1997:28-30):** *windows object inspector*, *windows form* dan *windows* editor program. *Windows* utama merupakan pusat pengaturan Delphi. *Windows object inspector* akan menampilkan *properties* dan *event*. *Windows form* untuk mengedit/membuat aplikasi, biasanya ditampilkan ditengah *window* editor program. *Windows* editor program digunakan untuk menyunting program objek Pascal.



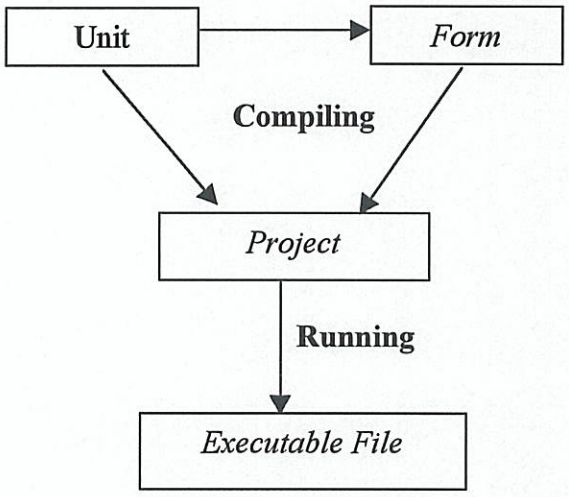
Gambar 2-1 Saat pertama kali Delphi dijalankan, Anda dapat mengubah posisi ketiga window sesuai dengan keinginan Anda.

2.5.2 Memahami Bahasa Delphi

Dasar bahasa pemrograman yang digunakan dalam Delphi adalah Pascal, sebuah bahasa yang didesain khusus oleh Niklaus Wirth untuk mengajarkan pemrograman terstruktur. Delphi sepenuhnya menerapkan *Metode Object*

Oriented Programming, suatu metode pemrograman yang mampu menangani program berukuran besar karena memiliki sistem organisasi yang baik dan bagus.

Oleh Borland Delphi dikembangkan dengan tujuan untuk menetapkan standar baru bahasa pascal. Delphi 3.0 merupakan bahasa pemrograman yang tampil dalam IDE (*Intergreted Development Environment*) yang baru. Delphi memilah program menjadi dua bagian utama yaitu bagian primer dan bagian sekunder. Bagian primer berupa *file* program yang mengkoordinasi keseluruhan program, *file* ini dicirikan dari ekstensinya yaitu *. Dpr . Sedangkan bagian sekunder merupakan bagian yang melaksanakan tugas-tugas tertentu yang disebut unit *file* bagian yang dicirikan oleh ekstensi *. Pas. Hasil rancangan tampilan disisipan dalam *file* dengan ekstensi *. Dfm. Linking dari *file* tersebut akan *digenerate* pembuatan *file* ber*ekstension* . Exe.



Gambar 2-2 Hubungan *File* dalam Bahasa Pemrograman Delphi 3.0

2.5.3 Fasilitas Database Delphi

Delphi mempunyai komponen-komponen yang dipakai untuk mengakses tabel dan metode untuk memanipulasi *record*. Delphi memberikan fasilitas untuk memanipulasi *database* yang kompleks. Dalam hal ini fasilitas *database* delphi memberikan *tools* untuk membuat form yang menghubungkan *field* dalam tabel ke *field* edit dalam form. Fasilitas ini dinamakan *Database Form Wizard*. *Database* lokal merupakan jenis *database* standar yang digunakan delphi seperti paradox dan dbase, sedangkan format *database* yang dapat digunakan bersama disebut *Database Server*.

2.5.3.1 Borland Database Engine

Definisi dari Borland Database Engine adalah sebagai berikut:

"Borland Database Engine (BDE) adalah Engine untuk pengaksesan data " **(Djoko Pramono, 1998:22).**

Pada BDE telah menyediakan *secket* ODBC (*Open Database Connectivity*) yang menghubungkan ke *driver* ODBC sehingga dapat mengakses data format *database* populer yang biasanya menggunakan sarana ODBC seperti Ms Access ataupun Ms Foxpro.

Selain fasilitas-fasilitas tersebut, dengan BDE akan memberikan kemudahan-kemudahan berikut ini **(Djoko Pramono, 1998:22)**:

(a) Berpindah-pindah dari *platform* satu ke *platform* lainnya.

(b) Otomatisasi penerjemahan jenis data antara jenis data C++ yang digunakan Delphi dengan jenis data *platform* tabel yang sedang aktif.

(c) Kemampuan *joint* antar tabel dari *platform* yang berlainan.

(d) Pengaksesan data secara dua arah pada tabel dengan berbagai *platform*.

(e) Dapat berbagi data dengan program *database* lain yang *platform*nya tidak dikenal oleh BDE.

2.5.3.2 Database Desktop

Delphi dirancang seutuhnya untuk menjadi sebuah klien *database* dan tidak terbatas pada format *database* yang dipakai, karena delphi dikemas dengan produk Borland yang lain yaitu Database Desktop (DBD). DBD menyediakan metode untuk membuat, melihat, mengedit, mengubah struktur, mengindeks, mengurutkan, membuat *Query* dan memanipulasi tabel-tabel dalam format Paradox, Dbase dan SQL. DBD dapat dianggap sebagai Versi Mini Paradox atau Dbase untuk *Windows*.

Definisi dari Database Desktop adalah sebagai berikut:

"Database Desktop adalah program database yang canggih penuh fasilitas pendukung database bahkan juga mudah dipelajari dan digunakan" **(Hengky W. Pramana, 1999:1)**.

Penambahan pada Database Desktop Versi 7.0 (*Windows 95*) ini termasuk mendukung untuk nama *file* yang panjang adalah fasilitas kemudahan dalam penggunaan seperti **(Hengky W. Pramana, 1999:2-3)**:

(a) Mengubah properti *object* menggunakan kotak dialog baru yang dimiiki tab.

(b) Penambahan fasilitas *toolbar* yang dapat dipindah-pindah letak posisinya.

(c) Perlengkapan menu yang mempermudah dalam menemukan perintah yang dibutuhkan.

(d) Perlengkapan penanganan file menggunakan kotak dialog *windows* untuk kenyamanan dan kompatibilitas tempat (*Suite Compatibility*).

(e) Tool baru dan bisa diimprovisasi yang menambah fungsional dan kenyamanan.

(f) Perlengkapan *On-Line Help* yang memudahkan untuk menemukan informasi yang dibutuhkan.

(g) Perlengkapan SQL yang dapat langsung bekerja dengan tabel SQL pada antarmuka pengguna Database Desktop (DBD).