BAB II
LANDASAN TEORI

2.1 Komputer

Komputer merupakan suatu perangkat elektronika yang dapat menerima dan mengolah data menjadi informasi, menjalankan program yang tersimpan dalam memori, serta dapat bekerja secara otomatis dengan aturan tertentu (Haryono, 2003).

2.2 Komputerisasi

Komputerisasi adalah suatu pengolahan data dengan sistem input, proses dan output yang menggunakan alat bantu komputer yang hasilnya berupa informasi. Dari pengertian tersebut dapat disimpulkan bahwa untuk mengolah suatu informasi diperlukan alat bantu yaitu komputer. Menurut Hartono (2000),

Perangkat komputer adalah sebagai berikut:

a. *Hardware* (perangkat keras) adalah peralatan di sistem komputer yang secara fisik terlihat dan di jamah
b. *Software* (perangkat lunak) adalah perogram berisi perintah – perintah untuk melakukan pengolahan data.

c. *Brainware* adalah manusia yang terlibat dalamnya, mengoperasikan serta mengatur sistem komputer.

2.3 Pengertian Aplikasi

Andri Kristanto (2003), menerangkan bahwa aplikasi adalah *software* program yang memiliki aktivitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu.

2.4 Pengertian Pengolahan Data

Pengolahan data merupakan proses pengumpulan data mentah berupa dokumen yang diinput kedalam sistem komputerisasi, dan diolah untuk menghasilkan suatu informasi yang berguna.

Menurut Jogiyanto Hartono (2004), pengolahan Data (*data processing*) adalah manipulasi dari data kedalam bentuk yang lebih berguna dan lebih berarti, yang berupa suatu informasi.

Pengolahan data adalah manipulasi dari data atau data yang diolah dari kegiatan pikiran dengan bantuan tenaga atau peralatan dengan mengikuti serangkaian langkah - langkah perumusan atau pola - pola tertentu untuk mengubah data sehingga lebih baik dan lebih berguna untuk mencapai tujuan tertentu.

2.5 Basisdata (*Database*)

Basisdata adalah sekumpulan informasi mengenai suatu subjek tertentu, yang memiliki keterkaitan logis, lengkap, dan terstruktur. Basis data menyediakan

struktur bagi informasi, dan memungkinkannya untuk digunakan bersama-sama oleh berbagai aplikasi yang berbeda (Haryono, 2003).

2.5.1 Klasifikasi Basisdata

Dalam perkembangannya, terdapat berbagai model basis data, yaitu :

1. Model hirarkis.
2. Model jaringan.
3. Model relasional.
4. Model berorientasi objek.

Dari beberapa model tersebut, yang paling berkembang dan banyak digunakan pada saat ini adalah model relasional. Model berorientasi objek biasanya dipergunakan untuk kebutuhan tertentu yang lebih spesifik.

2.5.2 Pengenalan Basisdata Relasional

Dalam model relasional, basisdata terdiri atas relasi-relasi atau tabel-tabel, yang dapat saling terhubung satu sama lain. Berikut bagian-bagian dari satu model tabel.

Field

Header1

Field1	Field2	...	Fieldn

Record

Gambar 2.1 Model Tabel

Field / kolom menyimpan data dengan jenis yang sama. Elemen-elemen dari *field* adalah :

1. Nama.

   Dalam satu tabel tidak diperkenankan adanya *field* dengan nama yang sama.

2. Tipe data.

   Tipe data dapat berupa tipe data standar atau tipe lain yang disediakan oleh basisdata.

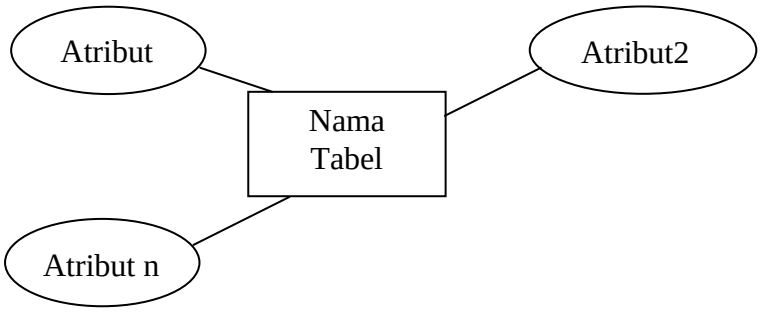
3. Ukuran, Ukuran *field* biasanya ditentukan berdasarkan tipenya, kecuali untuk tipe *string* (untaian karakter). Pada tipe string, pengguna menentukan sendiri ukuran yang digunakan.

   Record merupakan representasi dari satu entitas yang dicatat dalam tabel. Dalam sebuah tabel (mestinya) tidak ada dua atau lebih *record* yang sama persis. Setiap *record* bersifat unik dalam tabel yang memuatnya.

2.6 Model E-R (*Entity-Relationship*)

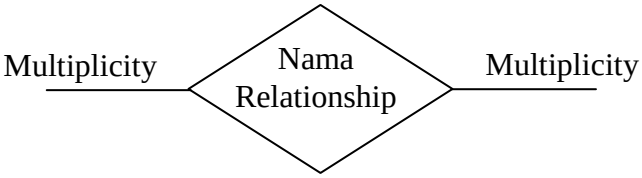
Dalam melakukan perancangan basisdata relasional, biasa dipergunakan diagram E-R. Komponen-komponen dalam model E-R adalah sebagai berikut.

1. *Entity Set*, merupakan kumpulan dari *entity* yang memiliki atribut-atribut yang sama. *Entity* dapat diartikan sebagai sesuatu yang dapat dibedakan dari yang lain. Dalam model E-R, biasa digambarkan sebagai berikut.



Gambar 2.2 *Entity Set*

2. *Relationship Set*, merupakan himpunan hubungan-hubungan antar entitas-entitas dari dua *entity set*. *Relationship Set* dilambangkan sebagai berikut.



Gambar 2.3 *Relationship Set*

2.7 Perancangan *Database*

Dalam merancang hubungan antar *file* dan model *database* digunakan model Data Relation (*Entry Relation Ship*). Pada ini hubungan antar *file* direlasikan dengan kunci relasi (*Relation Key)*, yang merupakan kunci utama –utama dari masing –masing *file.* Untuk membantu gambaran relasi secara lengkap, terdapat tiga macam relasi dalam hubungan atribut antar tabel sebagai berikut.

1. *One to One Relationship*

   Hubungan antara *file* pertama dengan *file* yang lain adalah satu banding satu. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukan tabel dan relasi antarkeduanya diwakilkan secara tunggal.

2. *One to many Relationship*

   Hubungan antara *file* pertama dengan *file* kedua adalah satu banding banyak atau banyak berbanding satu. Relasi antara keduanya diwakili dengan panah ganda untuk menunjukkan hubungan banyak tersebut.

3. *Many to Many Relationship*

   Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding banyak. Keduanya diwakili dengan panah ganda untuk menujukan hubungan banyak tersebut.

2.8 MySQL

Pada bagian ini akan dibahas tentang server *database*. MySQL merupakan sebuah server *database* yang banyak digunakan di internet karena kehandalannya dan sifatnya yang *shareware*. Sejarah MySQL adalah dengan dimulainya percobaan MySQL untuk menghubungkan tabel menggunakan rutin level (ISAM). Kenyataannya setelah melalui proses testing, ternyata MySQL tidaklah sesuai dengan yang diharapkan. MySQL adalah bahasa yang didesain khusus untuk pemakaian *multiuser* dan *server database* SQL *multi-thread*.

Beberapa alasan dipilih MySQL sebagai *database server* adalah sebagai berikut :

a. Dukungan berbagai bahasa

   Database server MySQL (MySQL daemon) dapat memberikan pesan error dalam berbagai bahasa seperti Cekoslowakia, Belanda, Inggris, Estonian, Perancis, Jerman, Hungaria, Italia, Norwegia, Polish, Portugis, Spanyol dan Swedia. MySQL secara default menggunakan set karakter 150-8859-1

(Latin1) untuk data dan sorting. Set karakter tersebut dapat diubah ketika dilakukan *compiling*.

b. Mampu membuat tabel berukuran sangat besar

Tabel-tabel di MySQL disimpan di direktori dan file terpisah. Ukuran maksimum dan setiap table adalah diantara 4 GB sampai ukuran *file* yang dapat ditangani oleh sistem operasi yang dipakai.

c. Kecepatan, kehandalan dan kemudahan dalam penggunaannya.

MySQL lebih cepat tiga sampai empat kali dan pada *database server* komersial yang beredar saat ini. MySQL juga dapat dengan mudah diatur (*easy to manage*). Tidak memerlukan seorang *database* administrator yang ahli untuk mengatur administrasi pemasangan MySQL.

d. Lebih Murah

MySQL adalah RDBMS (basis data relasional) yang *open source*. MySQL didistribusikan dengan gratis tanpa biaya untuk UNIX platform, OS/2 dan *windows platform*. Jadi biaya untuk menggunakan MySQL akan jauh lebih murah daripada menggunakan database komersial.

e. Melekatnya integrasi PHP dengan MySQL

Meskipun PHP banyak mendukung beberapa *database server* Iainnya seperti Oracle, SQL Server, PostgreSQL dan lain-lain, tetapi penulis lebih memilih MySQL karena selain database server ini sering digunakan para programmer PHP, juga keterikatan yang sangat kuat antar PHP dengan MySQL yang sama-sama *software open-source*, membuat koneksi yang terjadi adalah lebih cepat jika dibandingkan dengan menggunakan *database server* lainnya.

2.9 Designer

DBDesigner merupakan aplikasi yang berbentuk visual database yang mengintegrasikan desain database, model, penciptaan dan pemeliharaan menjadi satu. *DBDesigner* digunakan untuk membantu mengatur database, tabel, dan lain-lain. Dengan program ini pengguna dapat membuat tabel dan relasinya dengan mudah dan cepat. Fitur-fiturnya antara lain *reverse engineering of MySQL*, Oracle, MSSQL, bermacam-macam ODBC database, *drag and drop*, XML database model, melakukan pencetakan, dan lain-lain. Aplikasi ini bersifat *open source* dan memiliki *interface* yang profesional disertai detailnya.

DBDesigner dikembangkan dan dioptimalkan untuk open source MySQL. Semua fitur khusus MySQL telah dibangun untuk menawarkan cara yang paling nyaman untuk merancang dan menjaga kontrol terhadap MySQL-Database. Selain itu DBDesigner telah terintegrasi dengan database MySQL. Jadi kita bisa men-generate database secara langsung dari model yang telah kita buat.

2.11 Konsep Dasar Borland Delphi 7.0

Borland Delphi 7.0 adalah bahasa pemrograman yang dirancang untuk bekerja dalam MS-Windows. Borland Delphi 7.0 menggunakan bahasa Object Pascal yang sangat terkenal dan mendukung pemograman berorientasi obyek yang dikenal sebagai OOP (*Object Oriented Proramming*).

Selain menyediakan berbagai fasilitas pemrograman yang bersifat umum, misalnya kemampuan operasi numerik, kemampuan grafik dan lain-lain, Borland Delphi sangat dikenal sebagai bahasa pemrograman yang menyediakan fasilitas

pengolahan database yang canggih dan hebat. Berbagai format database bisa diolah dengan Borland Delphi 7.0 baik database dalam komputer tunggal, database server lokal maupun database dalam *server remote*. Format database yang dapat diolah dengan Borland Delphi 7.0 antara lain format database Paradox, dBase, MS-Access, ODBC, SyBASE, Oracle dan lain-lain.

2.11.1 Komponen-komponen Borland Delphi 7.0

Beberapa istilah dan komponen-komponen Borland Delphi 7.0 yang dipakai dalam membuat program aplikasi, antara lain :

1. *Project*

   Project adalah sekumpulan form, unit dan beberapa hal lain. Singkatnya Project adalah program aplikasi itu sendiri. *File* utama proyek disimpan dalam *file* berakhiran ***.dpr***. Selain itu disaat merancang proyek secara otomatis dibuat oleh Borland Delphi 7.0 sejumlah file yang diperlukan proyek tersebut, yaitu sebagai berikut :

a. *File Unit (.pas)*

   File *unit* dipakai untuk menyimpan program (kode program). Biasanya *unit* berhubungan langsung dengan form, tetapi kadangkala unit hanya berupa *procedure* atau *function* yang tidak berhubungan dengan form.

b. *File Form (.dfm)*

   Dipakai untuk menyimpan semua informasi mengenai form.

c. *File Project Option (.dfo)*

   Dipakai untuk menyimpan semua setting option (pilihan) proyek.

d. *File Resource (.res)*

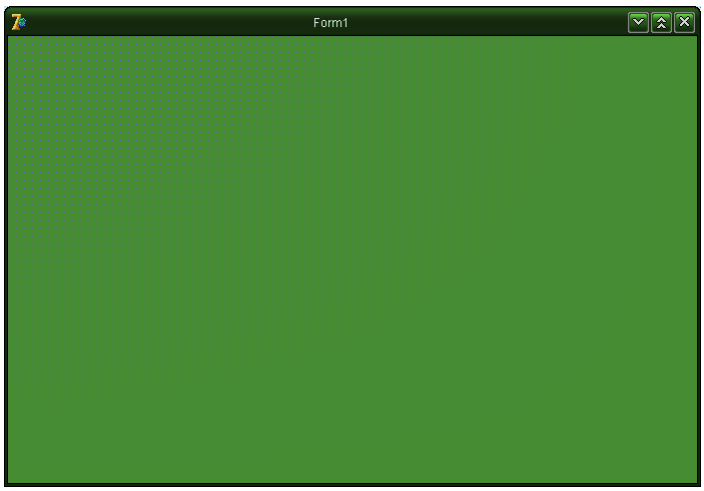
Dipakai untuk menyimpan ikon yang dipakai proyek.

e. *File Backup(.~dp, .~df, .~pa)*

Dipakai untuk menyimpan file *backup* proyek, *form* dan *unit*.

2. *Form*

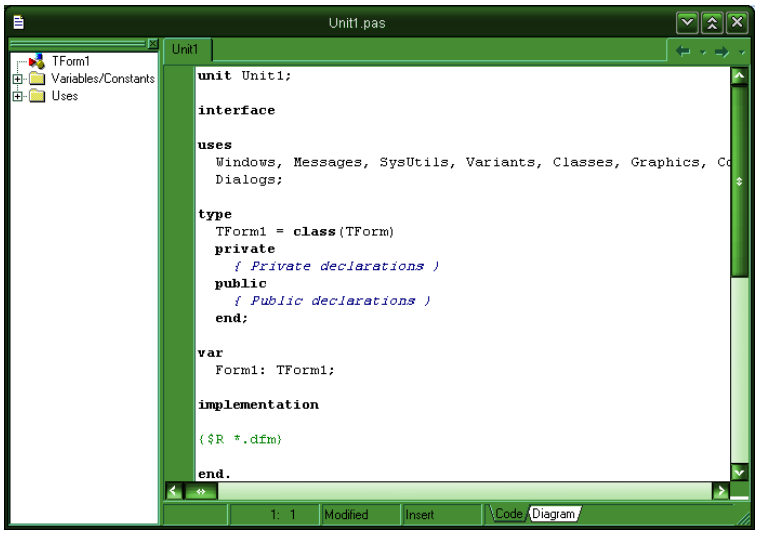
Form adalah suatu obyek yang dipakai sebagai tempat bekerja program aplikasi. Form berbentuk jendela dan dapat dibayangkan sebagai kertas atau meja kerja yang dapat dilukis atau diletakkan kedalamnya obyek-obyek lain. Pada saat pembuatan program aplikasi (proyek) baru, akan secara otomatis tersedia satu form dan disebut *Form1*. Biasanya pada saat mendesain form, terdapat titik yang disebut *grid*. Setiap form mengandung *unit*, *unit* dalam form dipakai untum mengatur dan mengendalikan form.



Gambar 2.4 Tampilan from

3. *Unit*

Unit adalah modul kode program. Satu program mungkin mempunyai satu atau lebih *unit*. Unit biasanya dipakai untuk mengatur dan mengendalikan segala sesuatu yang berhubugan dengan form.



Gambar 2.5 Tampilan unit

4. *Program*

Program dibangun dari satu atau lebih unit. Sebuah program, secara umum mempunyai stuktur sebagai berikut :

a. Heading Program, yaitu bagian yang menunjukan nama program tersebut.

b. Pernyataan User, yang berisi daftar *unit* yang dipakai program.

c. Blok deklarasi dan pernyataan, yaitu bagian yang berisi deklarasi dan pernyataan program yang dilaksanakan pada saat program dijalankan. bagian ini harus diakhiri dengan pernyataan *end* diikuti dengan titik.

5. *Properti*

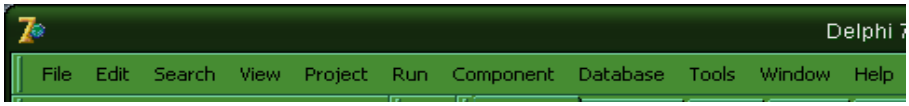
Properti digunakan untuk menentukan *setting* suatu obyek. Suatu obyek biasanya mempunyai beberapa properti, yang dapat diatur langsung dari lembar properti pada jendela *Object Inspector* maupun diatur lewat kode program. *Setting propertis* akan menentukan cara kerja dari obyek yang bersangkutan

saat program aplikasi dijalankan, misalnya menentukan warna obyek, bingkai obyek, pengambilan data dan lain-lain.

Gambar 2.6 Tampilan *Object Inspector*

6. *Menu*

Bagian menu Delphi dapat dilihat seperti pada Gambar 2.2.



Gambar 2.7 Menu Utama *Delphi*

Pada bagian menu terdapat sebelas menu utama, yaitu menu *File*, *Edit*, *Search*, *View*, *Project*, *Run*, *Component*, *Database*, *Tools*, *Window*, dan *Help*. Untuk menggunakan menu, pilih pada menu utama yang kita inginkan kemudian memilih submenu.

7. *Toolbar*

Toolbar merupakan kumpulan dari icon yang fungsi dan kegunaannya sama seperti fungsi menu. Bagian toolbar dapat dilihat seperti pada Gambar 2.3.

Gambar 2.8 *Toolbar*

Icon-icon pada toolbar adalah pilihan-pilihan pada menu yang sering digunakan dalam membuat program aplikasi. Toolbar dapat dijadikan jalan pintas untuk memilih proses yang sering dilakukan tenpa harus memilihnya pada menu.

8. *Component Palette*

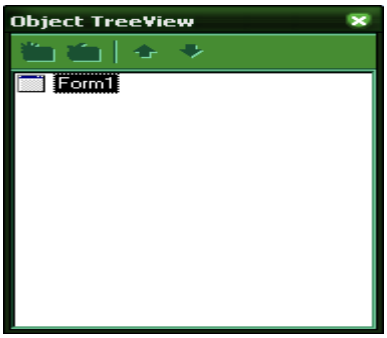
Component Palette adalah tempat dimana kontrol-kontrol dan komponen-komponen diletakkan. Kontrol-kontrol dan komponen-komponen yang terdapat pada *Component Palette* dipakai dalam pembuatan program apilkasi. Bagian *Component Palette* dapat dilihat seperti pada Gambar 2.4.

Gambar 2.9 *Component Palette*

9. *Object TreeView*

Object TreeView adalah tempat untuk melihat daftar dari objek-objek apa saja yang terdapat pada program aplikasi. Bagian *Object TreeView*

Dapat dilihat seperti pada Gambar 2.10.



Gambar 2.10 *Object TreeView*

10. *Event*

Event adalah peristiwa atau kejadian yang diterima oleh suatu obyek, misalnya klik, drag, tunjuk dan lain-lain. Event yang diterima obyek memicu Borland Delphi 7.0 menjalankan kode program yang ada di dalamnya.

2.11.2 Dasar Perancangan Program

Untuk merancang program database tidaklah gampang, banyak hal yang harus dilakukan dalam perancangan agar tampilan yang disajikan dalam program tersebut dapat berjalan dan digunakan dengan mudah.

Dalam perancangan program database ini dilakukan sesuia dengan survey di bawah ini :

1. Tahap Pengumpulan Data

   Yaitu dengan survei data, kepustakaan, wawancara dan lain-lain.

2. Tahap Perencanaan

   Sebuah informasi data yang akan disajikan kepada pengguna harus dapat dimengerti dan dipahami. Tahap ini yang dianggap sangat penting dalam

perancangan program, karena tahap ini menentukan seluruh literatur dalam sebuah program yang akan dibangun atau dibuat.

3. Penyajian Data Dalam Program

   Sebagai pengguna program harus mengerti letak dan kegunaannya, karena data yang ada di dalam program bisa dihapus atau tidak. Ada pula data yang harus diisikan oleh pengguna ke dalam program sebagai data masukan.