

BAB II

LANDASAN TEORI

2.1 Pengenalan Database

Definisi :

Database adalah sebuah kumpulan dari sekelompok informasi yang diorganisasikan dengan beberapa cara logik dan saling berhubungan, database juga mempunyai definisi yaitu suatu bentuk pengorganisasian data pada media eksternal (disk) dengan tujuan mempermudah pengaksesan (penyimpanan ataupun pengambilan data), adapula yang mengartikan bahwa database dapat dianggap sebagai tempat untuk sekumpulan berkas data terkomputerisasi. Melalui sistem basis data seseorang pemakai dapat melakukan berbagai fungsi seperti : menambahkan data, menghapus data, dan mengambil data.

Salah satu model database yang banyak digunakan adalah database relasional. Pada jenis database ini, sebuah database tersusun atas sejumlah tabel.

2.2 Database Relasional

Definisi :

Database Relasional adalah suatu jenis database yang terdiri atau tersusun dari sejumlah tabel.

Sebuah database relasional mengandung satu atau lebih tabel. Setiap tabel mengandung baris dan kolom, yang disebut record dan field dalam terminologi database. Setiap record yang dimasukkan akan mengisi sebuah baris dalam tabel. Setiap item data dalam sebuah record disimpan dalam sebuah field kunci dalam tabel.

Sebagai contoh, jika dibuat sebuah database yang mengandung dua buah tabel, maka kedua tabel tersebut dapat dihubungkan dengan menggunakan sebuah field yang terdapat pada kedua tabel itu. Sebuah hubungan (relationship) terjadi diantara record-record dalam dua buah tabel jika sebuah nilai field kunci dalam sebuah record tabel sama pada tabel yang lain dan saling dihubungkan. Dengan menggunakan kapasitas dari tabel yang berbeda, dapat dibuat tabel yang mempunyai bentuk yang sederhana dan kecil, dan setiap tabel berisi sebuah kumpulan data istimewa. Fasilitas ini membuat tabel pada database dapat di-update dan diubah-ubah sesuai keperluan sehingga lebih sederhana dan efisien.

2.3 Kunci dan Indeks

Dapat dihubungkan tabel-tabel dengan mendefinisikan sebuah relationship dari field tabel yang satu dengan field tabel yang lain. Field-field ini harus mempunyai indeks. Sebuah indeks adalah sebuah file yang digunakan Database Desktop untuk menjaga lokasi record dalam sebuah tabel. Ini mempermudah Database Desktop untuk :

- Memelihara order sebuah tabel yang diurutkan.

- Mengumpulkan seperti menilainya bersama-sama.

Baik tabel paradox dan dBASE dapat mengandung indeks untuk menspesifikasikan order record diakses, meskipun dengan cara yang berbeda. Pada tabel Paradox, indeks primer (*primary indexes*) kadang disebut sebagai kunci atau *keys*.

2.3.1. Kunci

Definisi :

Kunci (*key*) adalah sebuah field dalam sebuah database yang digunakan sebagai sarana pengurutan (indeks) data di dalam database. Kunci ini sering disebut juga sebagai kunci tunggal.

Sebuah tabel paradox dapat mempunyai indeks, tapi hanya satu yang dipilih sebagai indeks primer. Dalam tabel Paradox, indeks primer disebut kunci. Sebuah tabel yang mempunyai kunci disebut tabel berkunci (*keyed table*).

Ketika membuat sebuah kunci, Database Desktop memberikan aturan tentang data yang dapat dikandung dalam field berkunci. Setiap nilai field harus unik sehingga tidak ada duplikasinya dalam tabel. Dengan demikian Database Desktop dapat mengurutkan record tabel berdasarkan nilai dalam field yang didefinisikan sebagai kunci tabel.

Jika didefinisikan sebuah kunci pada tabel yang sudah mengandung data, Database Desktop memindahkan record tabel ke dalam order pengurutan yang benar. Lokasi record secara fisik ditunjukkan oleh pengurutan nilai field kunci dalam order menaik (A ke Z dan 0 ke 9). Sehingga record baru yang ditambahkan akan diletakkan pada posisi yang sesuai atau benar pada tabel yang diurutkan. Jika digunakan lebih dari satu field sebagai kunci, maka indeksnya disebut kunci gabungan (*Composite key*).

2.3.2. Kunci Gabungan (Composite Key)

Definisi :

Kunci Gabungan (*Composite Key*) adalah kunci dalam sebuah database yang digunakan sebagai sarana pengurutan (indeks) data di dalam database dan terdiri atas lebih dari satu field (kumpulan field).

Database Desktop membiarkan pembuatan nilai yang sama (duplikasi) dalam field individual kunci gabungan, sejauh nilai tersebut tidak diduplikasikan pada semua field kunci. Karena field kunci harus mempunyai keunikan pada setiap recordnya.

Ketika dibuat sebuah kunci gabungan, Database Desktop membuat sebuah indeks gabungan primer (*primary composite index*),

yang mengelola record dengan field pertama dari kunci (tergantung dari struktur tabelnya), kemudian field berikutnya.

2.3.3. Indeks

Definisi :

Indeks adalah pengurutan data dalam sebuah database atau dalam tabel adalah pengurutan record.

Ketika dibuat sebuah indeks, Database Desktop membuat satu atau lebih file yang memuat nilai field yang diindeks dan lokasinya. Database Desktop mengacu pada file indeks ketika mealokasikan dan menampilkan record dalam sebuah tabel.

2.4 Link

Definisi :

Link adalah hubungan (*relationship*) antara tabel dengan kata lain adalah hubungan fisis atau konseptual antar objek.

Hubungan antar tabel dilakukan jika nilai dari field yang satu dengan field yang diindeks pada tabel lain sama, maka Database Desktop akan secara otomatis menghubungkan data dari satu tabel ke tabel yang lain.

Satu atau lebih field yang digunakan untuk mendefinisikan link harus diindeks karena Database Desktop harus memadankan (*matching*) nilai dari field tersebut. Apabila sebuah indeks digunakan, maka Database Desktop mempunyai nilai terpelihara yang mempunyai file terpelihara yang mempunyai daftar lokasi dari semua record dalam tabel. Sehingga Database Desktop dapat menemukan dan menghubungkan (*link*) record dengan cara cepat dan efisien.

2.5 Objek dalam Database Desktop

Definisi :

Pada Database Desktop, komponen database yang mengirimkan, menampilkan, *me-trieve*, dan mempresentasikan data disebut **objek**.

Objek utama yang sering dikerjakan pada Database Desktop adalah tabel, query, dan file SQL.

Database Desktop menggunakan objek untuk mengirimkan, menampilkan, dan mempresentasikan informasi.

Suatu objek mengandung dua komponen yaitu :

- File dalam sebuah disk
- Tabel, query dan file SQL.

Database Desktop menggunakan ikon objek untuk mempresentasikan objek ketika diminimize. Setiap objek mempunyai ekstensi file yang berbeda.

2.5.1. Tabel

Definisi :

Tabel adalah suatu sarana dalam database yang digunakan untuk menyimpan data.

Tabel mempunyai baris dan kolom. Setiap baris memuat informasi tentang sebuah item penting seperti orang, tempat, benda, serta hal yang lainnya. Sehingga baris bisa disebut sebagai **record**. Setiap kolom memuat satu kategori data yang mendukung record dan sering disebut **field**.

2.5.2. Query

Definisi :

Sebuah query Database Desktop adalah sebuah pertanyaan tentang data pada tabel yang dapat digunakan untuk:

- Menemukan atau memilih data dari sebuah tabel
- Menggabungkan data lebih dari satu tabel
- Membuat perhitungan dengan data pada tabel

Database Desktop memberikan cara sederhana untuk menanyakan tentang data tabel. Dalam jendela Database Desktop Query, tinggal dipilih tabel yang ingin ditanyakan. Kemudian diberikan contoh data

yang diinginkan, dan Database Desktop menjawab berdasarkan contoh. Hal ini disebut dengan **Query By Example (QBE)**.

Database Desktop menyediakan tampilan query yang digunakan untuk mendefinisikan dan menjalankan sebuah query. Di dalam Query, dapat dideskripsikan sebuah fasilitas pengeditan data sehingga akan dihasilkan sebuah tampilan tabel hasil dari pendeskripsian data yang dituliskan dalam Query tersebut. Tampilan Query menyederhanakan cara untuk membatasi tampilan data.

2.5.3. File SQL

Definisi :

File SQL adalah sebuah objek yang mengandung kode yang ditulis dalam SQL (*Structured Query Language*).

SQL editor dapat digunakan untuk menulis kode SQL guna mengoperasikan data remote yang menggunakan Borland SQL Links. Dapat pula ditulis skrip menggunakan SQL yang dapat dijalankan pada data lokal Paradox atau dBASE.

2.6 Bekerja Dengan Database Desktop

Untuk memanipulasi database secara interaktif (tanpa melalui form), Delphi menyediakan tool yang disebut Database Desktop. Melalui tool inilah database bisa dimanipulasi, sebelum memasuki ke tahap pemrograman.

Setelah mengaktifkan Database Desktop maka akan terlihat tampilan seperti berikut :



Gambar 1. Form Database Desktop

2.6.1. Pengaturan Direktori Kerja Dan Alias

Direktori Kerja

Definisi :

Sebuah direktori kerja Database Desktop adalah Direktori yang digunakan untuk membuka dan menyimpan file.

Direktori kerja mengontrol setiap file yang ditampilkan dalam kotak dialog selama operasi Open dan Save. Direktori kerja defaultnya adalah WORKING dibawah direktori sistem, sehingga dapat diubah direktori kerja sesuai dengan keinginan pemakai.

Direktori Privat

Definisi :

Sebuah direktori yang digunakan menyimpan tabel temporer pada direktori yang tidak dapat digunakan bersama-sama dengan user lain.

Dalam lingkungan multi user, dibutuhkan tempat untuk meletakkan objek temporer. Dibutuhkan tempat untuk menyimpan tabel temporer pada direktori yang tidak dapat digunakan bersama-sama dengan user lain. Semua user Database Desktop membutuhkan direktori privat ketika menjalankan Database Desktop.

Direktori Privat defaultnya adalah PRIVATE, yang dibuat dibawah direktori Database Desktop utama pada hard disk. Direktori privat dapat pula diubah seperti yang diinginkan.

Alias

Definisi :

Sebuah alias adalah nama yang dibuat sebagai shortcut ke sebuah direktori.

Untuk defaultnya, direktori kerja mempunyai alias: WORK: dan direktori privat : PRIV:.

Keuntungan menggunakan Alias :

- Tidak perlu diketikkan nama path yang panjang
- Dengan menggunakan alias, dapat dihubungkan ataupun tidak dihubungkan dari server database remote.
- Dapat merubah definisi sebuah alias kapan saja. Semua objek Database Desktop yang memakai alias secara otomatis akan memakai alias yang baru.

Ada dua macam alias yang perlu diperhatikan :

- Public aliases
- Project aliases

Public dan Project Alias

Public alias disimpan pada file konfigurasi BDE (Borland Database Engine). Public alias tersedia dari direktori kerja dan terlihat pada beberapa aplikasi yang menggunakan BDE. Sedangkan Project alias tersedia apabila digunakan Database Desktop, dan pada direktori kerja yang dibuat.

Jika direktori kerja dirubah, maka Database Desktop tidak akan me-load semua project alias yang terhubung pada direktori kerja lama. Hal ini terjadi karena Database Desktop hanya me-load project alias yang spesifik dengan direktori kerja baru.

Jika sebuah project alias mempunyai nama sama dengan public alias, maka Database Desktop tidak akan me-load project alias.

Kotak dialog Alias Manager dapat digunakan untuk :

- Membuat alias baru
- Memodifikasi alias yang sedang digunakan
- Menghapus alias

2.6.2. Pembuatan Database Dengan Tabel Paradox

Sebagai tambahan, ketika dibuat tabel paradox, dapat dilakukan hal-hal seperti :

- Membuat kunci, atau indeks primer pada tabel

- Membuat indeks sekunder pada tabel
- Mendefinisikan validity checks untuk field individual
- Membangun tabel lookup untuk tabel lain
- Menentukan password untuk tabel atau field individual

Setelah dibuat sebuah tabel, dapat diurutkan recordnya atau mencari data dengan perintah dan query.

Apabila ingin menambah atau menghapus field atau membuat perubahan struktur tabel, dapat dibuat struktur ulang tabel tersebut.

Perencanaan Pembuatan Tabel Pada Bahasa Pemrograman Delphi

Perencanaan adalah langkah pertama dalam membuat sebuah tabel. Perlu diputuskan apa yang akan dikandung oleh sebuah tabel atau membuat layoutnya. Oleh karena itu perlu digunakan langkah-langkah berikut:

- Digunakan sedikit mungkin informasi pada setiap field untuk pemeliharaan data yang fleksibel dan mempercepat query. Misal pada field alamat dapat dibagi menjadi field jalan, kota telepon, kode pos dan lain-lain.
- Membuat tabel yang lengkap. Tabel yang lengkap adalah tabel yang berisi informasi yang dibutuhkan. Jika kemudian ditemukan field (informasi) yang lain, dapat ditambahkan pada tabel.
- Membuat tabel sesederhana mungkin untuk menghindari kompleksitas data tabel.

- Menghindari penduplikasian (redundancy) data. Contoh apabila akan dihubungkan dua buah tabel, perlu dihindari penggabungan field yang sama dari kedua tabel tersebut.
- Memahami jenis tabel yang dibutuhkan (Paradox atau dBASE) berdasarkan keuntungan masing-masing. Jika tabel Paradox mendukung password, validity checks, referential integrity, dan tipe field yang banyak jenisnya. Sedangkan tabel dBASE mendukung penghapusan dan kompatibel dengan dBASE yang ada sekarang. Jadi perlu dipertimbangkan sebelum memilih jenis tabel yang digunakan.

Untuk memilih tipe tabel yang akan digunakan, perlu dipertimbangkan :

- Jenis tabel apa yang akan dipinjam strukturnya nanti.
- Beberapa aturan untuk struktur tabel, seperti nama field, tipe, ukuran, dan aturan untuk menemukan field kunci yang diperbolehkan. Misalnya, Paradox memperbolehkan spasi dan penggunaan tanda baca pada penamaan tabel, sedangkan dBASE tidak.

Dengan demikian, ada beberapa fasilitas dari tipe tabel yang dibutuhkan. Fasilitas-fasilitas tersebut akan memberikan pertimbangan sebelum membuat tabel. Untuk itu dapat dibuat sebuah tabel pertimbangan untuk memilih Paradox atau dBASE pada bahasa pemrograman Borland Delphi seperti di bawah ini:

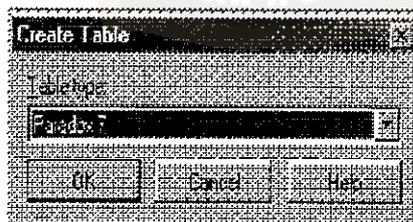
Tabel 1. Perbandingan Fasilitas Paradox dan dBASE Pada Bahasa Pemrograman Borland Delphi

Fasilitas	Paradox	dBASE
Memiliki validity checks, referential integrity, dan tabel look up	Ya	Tidak
Me-refresh data yang diubah	Ya	Tidak
Memiliki urutan penomoran dalam tabel	Ya	Tidak
Memformat field memo sebaik field long integer, time, timestamp, autoincrement, dan bytes	Ya	Tidak

Membuat Tabel Paradox

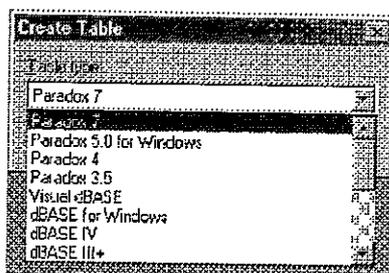
Untuk membuat sebuah tabel Paradox dari jendela Database Desktop dapat dilakukan langkah-langkah sebagai berikut :

- 1) Memilih **New** dari menu File, kemudian dipilih **Table**.



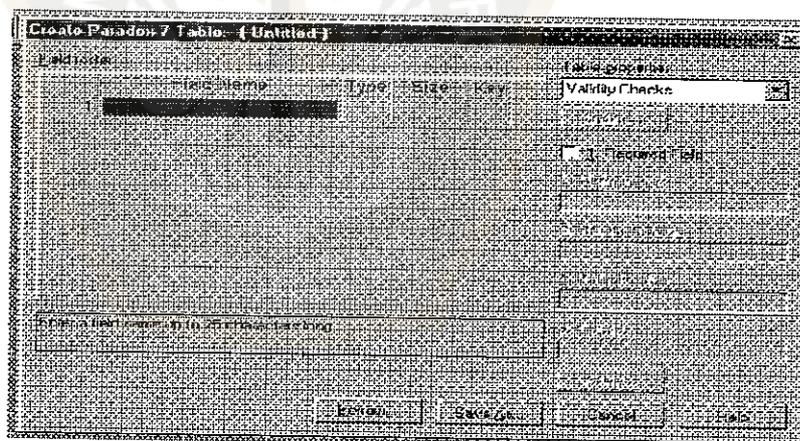
Gambar 2. Form Pembuatan Tabel Baru Pada Database Desktop.

- 2) Klik anak panah pada kotak pilihan Tabel Type untuk melihat daftar jenis-jenis tabel yang ada.]



Gambar 3. Form Pemilihan Tipe Tabel.

- 3) Pilih tabel Paradox versi 7.0 dan klik tombol **OK**. Selanjutnya Database Desktop akan menampilkan kotak dialog Create Paradox 7.0 Table. Kotak dialog ini adalah kotak dialog modal sehingga tidak dapat digunakan fasilitas DATABASE DESKTOP lain sampai mengklik tombol OK atau Cancel.



Gambar 4. Form Pengisian Field

Mendefinisikan Field

Untuk mendefinisikan sebuah field terdapat beberapa aturan :

- Panjang maksimum sebuah nama field adalah 25 karakter.

- Sebuah nama field tidak dapat dimulai dengan spasi kosong, tetapi dapat mengandung spasi kosong di dalamnya.
- Setiap nama field dalam sebuah tabel harus unik.
- Sebuah nama field seharusnya tidak mengandung karakter penting seperti koma (,), pipa (|), dan tanda seru (!).

Field dan Ukurannya

Tipe dan ukuran field pada tabel Paradox diperlihatkan pada tabel di bawah ini :

Tabel 2. Tipe-Tipe Data Pada Database Paradox

Simbol	Ukuran	Tipe	Keterangan
A	1 – 225	Alpha	Dapat berupa karakter apa saja. Panjang maksimal 255 karakter. Misalnya untuk menyimpan nama orang.
N		Number	Dapat berupa bilangan positif atau negatif dengan jangkauan dari -10^{307} sampai 10^{308} . Ketelitian sampai 15 digit
\$		Money	Biasa digunakan untuk menyatakan uang.
S		Short	Berguna untuk menyimpan bilangan bulat antara $-32,767$ sampai dengan $32,767$
I		Long Integer	Berguna untuk menyimpan bilangan

Simbol	Ukuran	Tipe	Keterangan
			bulat antara -214783648 sampai dengan 2147483647
#	0 – 32*	BCD	Untuk meyimpan data numerik dalam format BCD (Binary Coded Decimal)
D		Date	Untuk menyimpan data tanggal
T		Time	Untuk menyimpan data waktu dimulai dari tengah malam. Satuan milidetik, terbatas sampai 24 jam.
@		Timestamp	Untuk menyimpan data jam dan tanggal.
M	1 – 240**	Memo	Untuk menyinpan data alphanumerik yang terlalu panjang kalau disimpan dengan memakai tipe Aplha. Data disimpan pada berkas dengan ekstensi .MB
F	0 – 240**	Formatted Memo	Seperti tipe Memo, tetapi dapat menyimpan format teks seperti jenis, bentuk, ukuran dan warna.
G	0 – 240***	Graphic	Berguna untuk menyimpan data gambar. Format yang didukung: .BMP, .PCX, .TIF, .GIF, dan .EPS.
O	0 – 240***	OLE	Untuk menyinpan data OLE.

Simbol	Ukuran	Tipe	Keterangan
L		Logical	Untuk menyimpan data yang nilainya True atau False.
±		Autoincrement	Nilai akan dinaikkan sebesar satu terhadap setiap kali sebuah <i>record</i> diciptakan.
B	0 – 240***	Binary	Bila digunakan untuk menyimpan data dalam format biner (misalnya data suara).
Y	1 – 255	Bytes	Biasa digunakan untuk menyimpan kode kartu magnetik magnetik atau <i>barcode</i> .

Keterangan :

- * Jumlah digit di belakang desimal
- ** Field Memo dan Formatted Memo dapat dipilih panjang karakternya (1 sampai 240 karakter untuk memo dan 0 sampai 240 karakter untuk formatted memo)
- *** Optional

Field BLOP

Tipe field pada Paradox seperti memo, formatted memo, graphic, OLE, dan binary mengandung file binary large object atau disingkat sebagai BLOP.

Mengindeks

Database Desktop mengelola record tabel Paradox. Tabel tersebut mempunyai kunci yang menunjuk pada nilai dalam field kunci atau bisa disebut sebagai indeks primer. Sebagai defaultnya, semua indeks dikelola dan diakses dalam perintah menaik (A sampai Z, 0 sampai 9).

Ketika didefinisikan kunci gabungan untuk sebuah tabel, Database Desktop membuat sebuah indeks gabungan primer, yang mengelola record dengan field pertama sebagai kunci (tergantung struktur tabeknya), field kedua dan seterusnya.

Pada tabel Paradox, indeks sekunder mendefenisikan sebuah perintah tampilan alternatif untuk mengubah penampilan perintah record, meskipun lokasi record secara fisik dalam tabel tidak berubah, indeks sekunder juga digunakan untuk query, untuk mempercepat kemampuan, dan membangun hubungan antar tabel.

Aturan Membuat Field Kunci

Untuk mendefinisikan sebuah kunci harus diikuti aturan-aturan sebagai berikut :

- Sebuah tabel hanya mempunyai sebuah kunci. Kunci ini dapat dibuat dari satu atau lebih field.
- Sebuah kunci tidak dapat mengandung memo, formatted memo, graphic, OLE, binary, logical, atau field bytes.

- Jika sebuah kunci didefinisikan sebagai field tunggal, maka field tersebut harus field pertama pada Field Roster.
- Jika didefinisikan lebih dari satu field sebagai kunci, maka berarti dibuat sebuah composite key. Field-field ini dianggap sebagai grup, dan harus unik untuk setiap record dalam tabel. Composite key harus dimulai dari field pertama dalam Field Roster.
- Jika ditambah kunci pada sebuah tabel yang sebelumnya tidak mempunyai kunci atau mempunyai kunci yang lain, maka pengacauan kunci (*key violation*) akan terjadi.
- Data yang sudah dimasukkan ke dalam tabel mengacaukan aturan yang dibuat oleh kunci baru. Database Desktop menulis record yang kuncinya kacau (*key violating*) pada tabel temporer khusus yang disebut Keyviol.
- Record yang kuncinya dihapus dari tabel. Dapat diubah record dalam tabel Keyviol sehingga mereka membutuhkan kunci, kemudian ditambahkan pada tabel asli dengan menggunakan Tools, Utilities, Add.
- Tabel tipe Paradox mempunyai lebih dari satu field yang didefinisikan sebagai field kunci. Field-field tersebut diatur seperti group atau gabungan. Field kunci gabungan harus field pertama dari tabel. Dapat digunakan field kunci gabungan apabila semua nilai adalah unik.

- Ketika sebuah tabel mempunyai sebuah field kunci gabungan, nilai duplikasi dibiarkan dalam sebuah field kunci individu, sepanjang tidak terduplikasi pada semua kunci atau harus unik.
- Database Desktop mengurutkan tabel yang mempunyai field kunci gabungan dengan memulai dari field pertama, kemudian baru field selanjutnya.

Validity Check

Pada tabel Paradox, validity checks adalah aturan yang dibebankan pada sebuah field untuk meyakinkan bahwa data yang dientri ke dalam field adalah suatu kebutuhan penting. Cara mendefinisikan sebuah validity check menjelaskan data apa yang dapat dimasukkan dalam sebuah field. Database Desktop menyediakan lima macam validity checks sebagai berikut:

Tabel 3. Macam Validity Check

Validity Check	Artinya
Required Field	Setiap record tabel harus mempunyai sebuah nilai dalam field
Minimum Value	Nilai yang dimasukkan ke dalam field harus sama atau lebih besar daripada minimum yang dispesifikasikan.
Maximum Value	Nilai yang dimasukkan ke dalam field harus sama atau lebih kecil daripada maksimum yang dispesifikasikan.
Default Value	Nilai yang dispesifikasikan di sini akan dimasukkan secara

	otomatis di dalam field ini, jika tidak ada nilai yang dimasukkan.
Picture	Menentukan sebuah karakter string yang bertindak selaku template untuk nilai yang dapat dimasukkan pada field ini.

Ketika menyimpan tabel yang dibuat, Database Desktop akan menyimpan validity checks dalam sebuah file dengan nama tabel dan berekstensi. VAL.

Ketika memilih sebuah field dalam Field Roaster, Database Destop menampilkan validity Checks pada sisi kanan bawah jendela. Setelah memilih field, validity checks mengubah untuk merefleksikan constraint pada field yang terpilih.

Referential Integrity

Referential Integrity adalah sebuah field atau kumpulan field pada satu tabel (tabel anak) yang harus menunjuk pada kunci tabel yang lain (tabel induk). Hanya nilai yang aktif pada kunci tabel induk yang valid dengan field tabel anak. Database Desktop akan memeriksa keabsahan nilai dari sebuah tabel sebelum menerimanya pada tabel referential integrity.

Konversi Field Paradox

Pengkonversian field Paradox dari satu tipe ke tipe yang lain, akan digunakan tabel untuk memutuskan kekompatibelan tipe field.

Tabel 4. Konversi Tipe Data Paradox.

	A	N	S	D	S	M	F	B	G	O	L	I	T	@	#	+	Y
A	Y	P	P	P	P	Y					P	P	P	P	P		Y
N	Y	Y	Y		P						Y	Y			Y	<	
S	Y	Y	Y		Y						Y	Y			Y		
D	Y				Y										Y		
S	Y	Y	Y		Y						Y	Y			Y	<	
M	Y					Y	Y	Y									
F						Y	Y	Y									
B								Y									
G								Y	Y								
O								Y		Y							
L	Y	Y	Y		Y						Y	Y			Y	<	
I	Y	Y	Y		Y						Y	Y			Y		
T	Y												Y	Y			
@	Y			Y									Y	Y			
#	Y	Y	Y		Y						Y	Y			Y		
+	Y	Y	Y		Y						Y	Y			Y	Y	
Y	Y																Y

Keterangan :

Y : Boleh dikonversi, tetapi harus me-trim data. Jika DBD harus me-trim data, maka akan dilihat kotak dialog Restructure Warning, yang menyatakan konfirmasi pengkonversian.

Kosong : Tipe field tidak dapat dikonversi.

P : Konversi diperbolehkan, tapi mungkin harus membuat tabel problem

< : Konversi ke autoincrement hanya dari sebuah field kunci tunggal yang mengandung data seperti < 2147483647.

2.7 Desain Sistem Perangkat Lunak

Terdapat dua bagian dalam pembagian desain sistem, yaitu desain sistem secara umum (general system design) dan desain sistem terperinci (details system design).

Arti Desain Sistem

Desain sistem dapat didefinisikan sebagai berikut :

Menurut Robert J. Verzello/John Reuter III :

The stage of the development cycle which follow analysis : definition of functional requirement and preparation on implementation spesification; describing how a system is to constructed.

(Tahap setelah analisis dari siklus pengembangan sistem:
pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk

rancang bangun implementasi; menggambarkan bagaimana suatu sistem dibentuk).

Menurut John Burch & Gary Grudnitski :

System design can be defined as drawing, planning, sketching, or arranging of many separate into a viable, runified a whole.

(Desain sistem dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi).

Menurut George M. Scott :

System design determines how a system will accomp what it must accomplish; it involves configuring the software and hardware component of a system so that after the installation to the system will fully satisfy the system specifications esthablished at the end of the system analysis phase.

(Desain 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).

Dengan demikian desain sistem dapat diartikan sebagai berikut :

1. Tahap setelah analisis dari suatu siklus pengembangan sistem;
2. Pendefinisian dari kebutuhan-kebutuhan fungsional;
3. Persiapan untuk rancang bangun implementasi;
4. Menggambarkan bagaimana sistem dibentuk; yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi; Termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem.

Tujuan Desain Sistem Perangkat Lunak

Tahap desain sistem mempunyai dua tujuan utama, yaitu sebagai berikut :

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

Tujuan yang kedua lebih condong pada desain sistem yang terinci, yaitu pembuatan rancang bangun yang jelas dan lengkap untuk nantinya digunakan untuk pembuatan program komputernya.

Untuk mencapai tujuan tersebut, analisis sistem harus dapat mencapai sasaran-sasaran sebagai berikut :

1. Desain sistem harus berguna, mudah dipahami dan nantinya mudah digunakan. Ini berarti bahwa data harus mudah diterapkan dan informasi harus mudah dihasilkan serta mudah dipahami dan digunakan.
2. Desain sistem harus dapat mendukung tujuan utama sesuai yang telah didefinisikan pada tahap perencanaan sistem yang dilanjutkan pada tahap analisis sistem.
3. Desain sistem harus efisien dan efektif untuk dapat mendukung pengolahan transaksi, pelaporan manajemen dan mendukung keputusan yang akan dilakukan oleh manajemen, termasuk tugas-tugas yang lainnya yang tidak dilakukan oleh komputer.
4. Desain sistem harus dapat mempersiapkan rancang bangun yang terinci untuk masing-masing komponen dari sistem informasi yang meliputi data dan informasi, simpanan data, metode-metode, prosedur-prosedur, orang-orang, perangkat keras, perangkat lunak dan pengendalian intern.

2.7.1 Desain Sistem Perangkat Lunak Secara Umum

Analisis sistem dapat mendesain model dari sistem informasi yang diusulkan dalam bentuk *physical system* dan *logical model*. Bagan alir sistem (*system flowchart*) merupakan alat yang tepat digunakan untuk menggambarkan *physical system*. Simbol-simbol bagan alir sistem ini menunjukkan secara tepat arti fisiknya, seperti simbol terminal, *hardisk*, laporan-laporan.

Logical model dari sistem informasi lebih menjelaskan kepada user bagaimana nantinya fungsi-fungsi di sistem informasi secara logika akan bekerja, logical model dapat digambar dengan menggunakan diagram arus data (data flow diagram). Arus dari data di DAD dapat dijelaskan dengan kamus data (data dictionary).

Sketsa dari physical system dapat menunjukkan kepada user bagaimana nantinya sistem secara fisik akan digunakan atau diterapkan. Pengolahan data dari sistem informasi berbasis komputer membutuhkan metode-metode dan prosedur-prosedur. Metode-metode dan prosedur-prosedur ini merupakan bagian dari model sistem informasi (model prosedur) yang akan mendefinisikan urutan-urutan kegiatan untuk menghasilkan output dari input yang ada. Metode adalah suatu cara untuk melakukan suatu kegiatan. Suatu prosedur merupakan rencana tahap demi tahap untuk menerapkan suatu metode. Bagan alir sistem merupakan alat berbentuk grafik yang dapat digunakan untuk menunjukkan urutan kegiatan dari sistem informasi yang berbasis komputer.

Desain Database Secara Umum

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. Database merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Penerapan database dalam sistem

informasi disebut database system. Sistem basis data ini adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam.

Langkah-langkah Desain Database Secara Umum

Untuk tahap desain database secara umum, yang perlu dilakukan adalah mengidentifikasi terlebih dahulu file-file yang diperlukan oleh sistem informasi. File-file database oleh sistem dapat dilihat pada desain model yang digambarkan dalam bentuk diagram arus data. Langkah-langkah desain database secara umum adalah sebagai berikut :

1. Menentukan kebutuhan file database untuk sistem baru.
2. Menentukan parameter dari file database.

Merancang Database merupakan sesuatu yang sangat penting. Kesulitan utama dalam merancang database adalah bagaimana merancang sehingga database dapat memuaskan keperluan saat ini dan masa datang. Perancangan model konseptual perlu dilakukan disamping perancangan model fisik. Pada perancangan konseptual akan menunjukkan entity dan relasinya berdasarkan proses yang diinginkan oleh organisasi. Ketika menentukan entity dan relasinya dibutuhkan analisis data tentang informasi yang ada dalam spesifikasi di masa datang.

Pada pendekatan model konseptual, beberapa pendekatan Relasional digunakan, namun tidak berarti konsep ini nantinya diimplementasikan ke model Relational saja tetapi dipakai pada model Hierarhical dan model Network.

Pada perancangan model konseptual penekanan tinjauan dilakukan pada struktur data dan relasi antara file. Tidak perlu dipikirkan tentang terapan dan operasi yang akan dilakukan pada database. Pendekatan yang dilakukan pada perancangan model konseptual adalah menggunakan model data relasional berupa:

1. Teknik Normalisasi

Proses normalisasi merupakan proses pengelompokan 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 dengan kata lain perancangan belum mendapat database yang optimal.

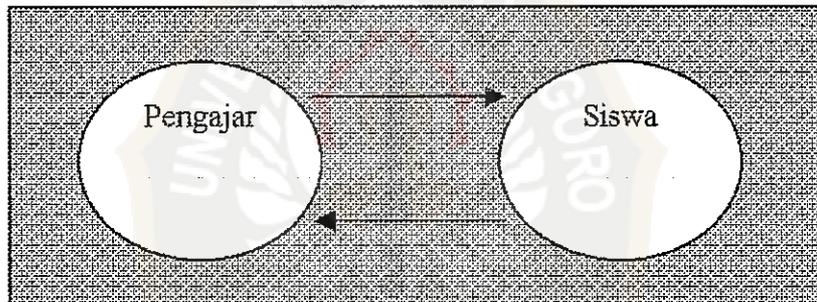
2. Teknik Entity Relationship

Database adalah kumpulan file yang saling berkaitan. Pada model data relational hubungan antar file direlasikan dengan kunci relasi (relation key), yang merupakan kunci utama dari masing-masing file. Perancangan yang tepat akan membuat database akan bekerja secara optimal.

Relasi antara dua file atau dua tabel dapat dikategorikan menjadi tiga macam. Demikian pula untuk membantu gambaran relasi secara lengkap terdapat juga tiga macam relasi dalam hubungan atribut dalam satu file.

- **One to one relationship 2 file**

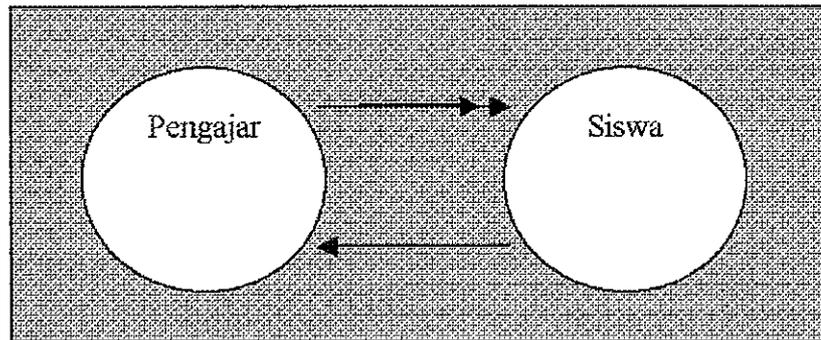
Hubungan antara file pertama dengan file kedua adalah satu berbanding satu. Seperti pada pelajaran privat dimana satu pengajar mengajar satu siswa dan satu siswa hanya diajara oleh satu guru pula. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah tunggal.



Gambar 5. Relasi One to One relationship 2 file

- **One to many relationship 2 file**

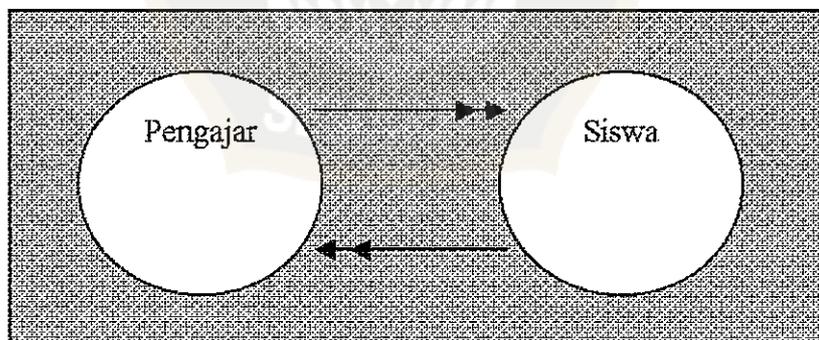
Hubungan antara file pertama dengan file kedua adalah satu berbanding banyak atau dapat pula dibalik banyak lawan satu. Seperti dalam pelajaran satu guru mengajar banyak siswa. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.



Gambar 6. Relasi One to Many relationship 2 file

- **Many to many relationship 2 file**

Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Seperti pada sistem pengajaran di perguruan tinggi dimana banyak guru mengajar banyak siswa dan siswa diajar oleh banyak guru pula. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.

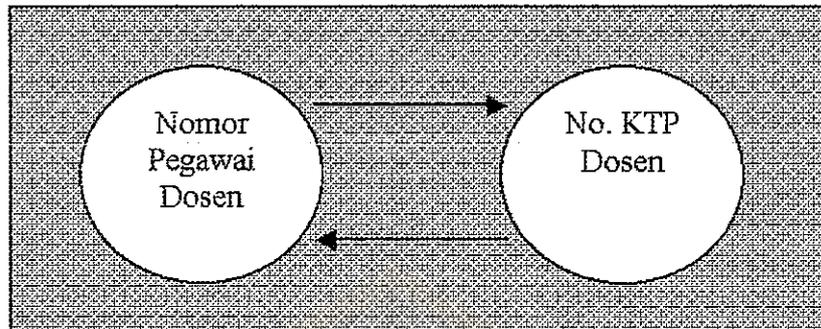


Gambar 7. Relasi Many to Many relationship 2 file

- **Relasi One to one 2 atribut dalam 1 file**

Hubungan antara satu atribut dengan atribut yang lain dalam satu file yang sama mempunyai hubungan satu lawan satu.

Misalnya atribut nomor induk pegawai dosen yang unik dan atribut nomor ktp pegawai tersebut mempunyai hubungan satu lawan satu. Satu nomor pegawai satu nomor ktp tidak ada yang berganda.



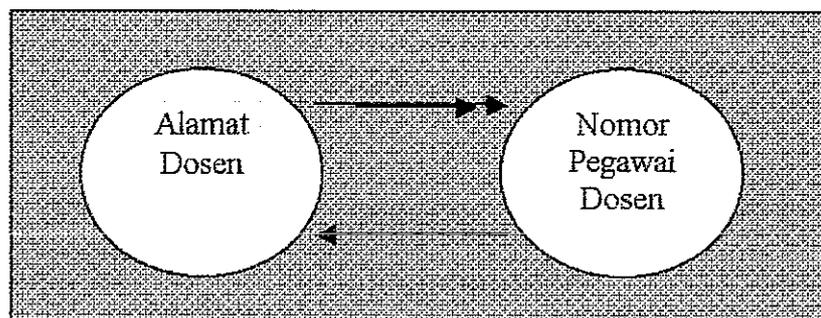
Gambar 8. Relasi One to One 2 attribute relationship 1 file

- **Relasi Many to one 2 attribute dalam 1 file**

Hubungan antara satu atribut dengan atribut yang lain dalam satu file yang sama mempunyai hubungan satu lawan banyak.

Misalnya pada satu alamat Dosen terdapat puluhan pegawai kantor tersebut.

Hubungan antara atribut alamat Dosen dengan nomor pegawai Dosen adalah satu alamat menunjukkan banyak pegawai.



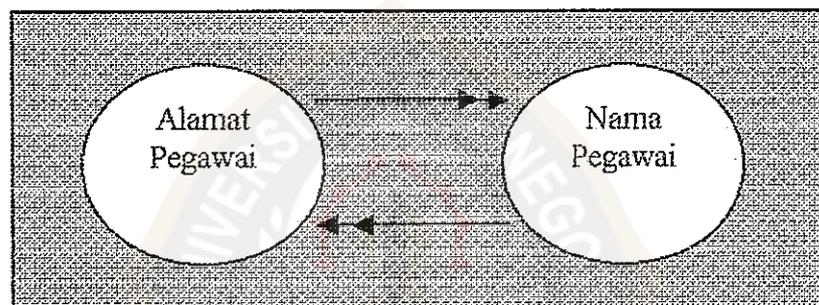
Gambar 9. Relasi Many to One 2 attribute relationship 1 file

- **Relasi Many to many 2 atribut dalam 1 file**

Hubungan antara satu atribut dengan atribut yang lain dalam satu file yang sama mempunyai hubungan banyak lawan banyak.

Misalnya pada satu alamat Dosen terdapat puluhan Dosen tersebut dan beberapa Dosen mempunyai nama sama mempunyai alamat yang berbeda.

Maka hubungan antara atribut alamat Dosen dengan nama Dosen adalah beberapa alamat menunjukkan nama Dosen yang sama dan sebaliknya.



Gambar 10. Relasi Many to Many 2 atribut relationship 1 file

2.7.2 *Desain Sistem Perangkat Lunak Secara Khusus*

Yang dimaksud dengan desain perangkat lunak secara khusus di sini adalah desain sistem yang mengacu pada penerapan langsung di lapangan dan pada saat proses pembuatan program. Desain perangkat lunak secara khusus dapat diartikan proses perancangan tampilan dan jalur arah program yang dibuat oleh programmer.