

MODUL
PRAKTIKUM CLIENT SERVER

AGUS SUJARWADI

**UNIVERSITAS TEKNOLOGI YOGYAKARTA
2018**

PENDAHULUAN

i. Deskripsi Materi

Materi dalam modul ini membahas tentang pembangunan sistem untuk Perpustakaan. Bahasa pemrograman yang digunakan adalah Delphi dengan MySQL sebagai DBMS nya. Terdapat beberapa bagian/langkah-langkah pengerjaan aplikasi ini. Dimulai dengan pembahasan tentang basis data, MySQL, dan cara setting awal untuk koneksi aplikasi dengan DBMS nya serta dilanjutkan dengan membangun formulir-formulir yang diperlukan dalam sebuah Aplikasi Perpustakaan.

Modul yang disusun ini memiliki keterkaitan dengan modul pada mata kuliah lain sebagai alat bantu yang memberi kemudahan bagi mahasiswa baik untuk menyelesaikan tugas sehari-hari maupun mempersiapkan mahasiswa dalam penyusunan tugas akhir.

Setelah mempelajari modul ini, mahasiswa mampu merancang dan membangun sebuah aplikasi sederhana dengan bantuan bahasa pemrograman Delphi.

ii. Prasyarat

- Mahasiswa sudah menguasai dan lulus pada matakuliah prasyarat dari matakuliah ini.
- Mahasiswa tidak asing dengan istilah-istilah pada Delphi dan MySQL.

iii. Petunjuk Pemakaian Modul

Modul ini dapat digunakan mahasiswa dengan pertimbangan sebagai berikut :

- Mahasiswa telah memiliki modul dan telah membaca modul sebelum mata kuliah dimulai.
- Mahasiswa mempelajari serta mengidentifikasi isi modul yang diuraikan lebih rinci oleh dosen pengampu.
- Mahasiswa dan dosen mendiskusikan materi untuk mencari penyelesaian terhadap kasus tertentu.
- Mahasiswa menyimpulkan isi materi yang didiskusikan
- Mahasiswa menjawab soal latihan yang diberikan
- Pemberian pengayaan materi bagi mahasiswa yang telah memahami dan menyelesaikan soal latihan.
- Memberikan tinjauan ulang terhadap materi sekaligus mengidentifikasi kesulitan-kesulitan mahasiswa dalam memahami materi.

iv. Standar Kompetensi

1. Mahasiswa mampu menggunakan perangkat dalam Delphi
2. Mahasiswa mampu membuat dan menyelesaikan tugas-tugas pembuatan program berbasis Delphi
3. Mahasiswa mampu melanjutkan rancangan dan menyelesaikan pembangunan aplikasi perpustakaan berbasis Delphi dan MySQL

BAB 1

BASIS DATA

2.1 Mengenal MySQL

MySQL adalah sebuah sistem manajemen database relasi (relation database management system) yang bersifat “terbuka” (open source). Terbuka maksudnya adalah My SQL boleh di download oleh siapa saja, baik versi kode program aslinya (source code program) maupun versi binernya (executable program) dan bisa digunakan secara (relatif) gratis baik untuk dimodifikasi sesuai dengan kebutuhan seseorang maupun sebagai suatu program aplikasi komputer. Bila anda seorang pengembang perangkat lunak (programmer) dan ikut berpartisipasi dalam memodifikasi program asli MySQL untuk keperluan pribadi atau perusahaan sendiri.

MySQL merupakan hasil ciptaan dari Michael “Monty” Widenius, David Axmark, dan Allan Larson pada tahun 1995. Mereka lalu mendirikan perusahaan bernama MySQL AB di Swedia. Tujuan awal diciptakannya MySQL yaitu untuk mengembangkan aplikasi web yang akan digunakan oleh salah satu klien MySQL AB.

Terdapat pengkategorian dalam perintah-perintah MySQL. Pengkategorian tersebut adalah sebagai berikut:

1. *Data Definition Language* (DDL), digunakan untuk membangun objek-objek basis data, seperti: membuat tabel baru, membuat indeks primer/sekunder.
2. *Data Manipulation Language* (DML), digunakan untuk menambah, mencari, mengubah dan menghapus baris tabel.
3. *Data Control Language* (DCL), digunakan untuk menangani masalah sekuriti dalam basis data.

2.1.1 Bahasa Query Terstruktur/*Structured Query Language* (SQL)

SQL adalah bahasa yang berisi perintah-perintah untuk membuat database, menghapus database, melihat seluruh database, membuka database, melihat seluruh isi tabel dalam database, membuat tabel, menghapus tabel, mengubah nama tabel, memanipulasi tabel (menghapus, mengubah, memilih, menggabungkan data) dan

sebagainya. Standar SQL telah ada sejak 1986 dan terus diperbaharui menjadi standar pada tahun 1989, 1992, 1997, dan terakhir 1999. Pada saat ini SQL banyak digunakan pada aplikasi-aplikasi basis data karena bahasa yang efisien dan mudah dipelajari. SQL merupakan salah satu fasilitas yang digunakan dalam menuliskan query. SQL merupakan bahasa yang dapat digunakan untuk mendefinisikan skema database (DDL) dan dapat digunakan untuk memanipulasi isi dari satu atau lebih tabel yang mempunyai keterkaitan antara satu tabel dengan tabel yang lain (DML) serta digunakan sebagai pengaman database (DCL).

2.1.2 Database/Basis Data

Seperti telah disebutkan diatas bahwa database merupakan kumpulan dari beberapa tabel. Sehingga, untuk dapat membuat tabel, user/pengguna harus membuat database lebih dahulu sebagai tempat untuk menempatkan tabel-tabel. Sedangkan nantinya untuk tiap tabel akan mengandung baris-baris data atau yang lebih dikenal dengan record.

Dalam pembuatan database harus diperhatikan adalah penulisan nama database yang tidak diperbolehkan menggunakan spasi dan karakter nonstandard. Bentuk penulisan perintah/sintaks untuk membuat database baru adalah :

Create database <nama_database>

Contoh:

```
create database akademik;
```

Untuk melihat seluruh database yang ada dalam media simpan/disk adalah sebagai berikut:

```
Show databases;
```

Sedangkan setelah database terbuat untuk menggunakannya diperlukan suatu perintah (use) untuk memanggil database dan perintah ini harus selalu digunakan ketika kita akan mengakses tabel-tabel yang berada dalam database. Struktur penulisannya adalah:

Use <nama_database>

Contoh:

```
Use akademik;
```

Untuk menghapus suatu database yang pernah dibuat, dapat dilakukan dengan memberikan perintah drop, seperti sintaks berikut:

Drop database <nama_database>

Contoh:

Drop database akademik;

2.1.3 Data Definition Language (DDL)

A. Membuat Tabel

Tabel sebagai tempat penyimpanan data-data, mempunyai bagian utama yaitu nama, tipe data dan ukuran/banyak karakter yang akan mengisi data tersebut. Perlu diperhatikan dalam pembuatan tabel, apakah database sudah dipanggil sebelumnya? Jika belum, panggil database tersebut atau buat database tersebut atau buat database baru jika belum tersedia. Hal ini dilakukan agar tabel yang akan dibuat mempunyai tempat dalam database. Tabel, sebagai tempat penampungan data, mempunyai dua bagian utama, yaitu nama dan tipe data. Tipe data yang bisa digunakan dalam MySQL dapat dilihat dalam dibawah ini.

TIPE DATA	KETERANGAN
TINYINT	Ukuran terkecil dari integer Signed range : -128 Unsigned range : 0 s.d 255
SMALLINT	Ukuran kecil dan integer Signed range: -32768 s.d. 32767 Unsigned range: 0 s.d. 16777215
MEDIUMINT	Ukuran menengah dan integer Signed range: -8388608 s.d. 8388607 Unsigned range: 0 s.d. 16777215
INT, INTEGER	Integer Signed range: -2147483648 s.d. 2147483617 Unsigned range: 0 s.d. 4294967295.
BIGINT	Ukuran terbesar dan integer Signed range: -922337203654775808 s.d. 9223372036854775807. Unsigned range: 0 s.d. 18446744073709551615.
FLOAT	Presesi tunggal Range: -3.402823466E+38 s.d. -1.175494351 E38, 0, 1.175494351E-38 s.d. 3.402823466E+38

DOUBLE, DOUBLE PRECISION, REAL	Presisi ganda Range: -1.7976931348623157E+308 s.d. -2.22507385850720E-308, 0, dan 2.225073885072014E-308 s.d. 1.7976931348623157e+308
DECIMAL, NUMERIC	Pecahan dengan range seperti tipe data DOUBLE
DATE	Tanggal, Format: YVYY-MM-DD
DATETIME	Kombinasi tanggal dan jam. Range: '1000-01-01 00:00:00' s.d. '9999-12-31 23:59:59. Format: YYYY-MM-DD HH:MM:SS
TIME	Jam Range: -838:59:59 s.d. '838:59:59' Format: 'HH:MM:SS'
YEAR	Tahun Range: 1901 s.d. 2155 (4 digit), 1970-2069(2 digit) Format: 'YYYY'
CHAR	Karakter dengan panjang tetap sesuai saat pembuatan tabel, dengan karakter spasi dihilangkan pada saat penyimpanan Range : 1 s.d. 225 Karakter
NCHAR, NATIONAL CHAR	Karakter dengan panjang tetap sesuai saat pembuatan tabel, namun karakter spasi tidak dihilangkan pada saat penyimpanan Range: 1 s.d. 255 karakter
VARCHAR	Karakter dengan panjang sesuai saat panjang karakter yang ditulis, dengan karakter spasi dihilangkan pada saat penyimpanan. Range: 1 s.d. 255 karakter
TINYBLOB TINYTEXT	TEXT/BLOB dengan panjang maksimal 255 karakter
MEDIUMBLOB MEDIUMTEXT	65535
LOBLOB LONGTEXT	16777215
ENUM('value1','value2',)	Obyek string yang hanya boleh diisi dan daftar pilihan value yang diberikan, NULL, atau error value, Maksimum value: 65535
SET('value1','value2',.....)	Obyek string yang hanya boleh diisi dan daftar pilihan value yang diberikan, NULL atau error value. Maksimum value; 64

Untuk membuat tabel baru dengan SQL yaitu dengan perintah *create*. Sintaks:

Create tabel <nama tabel>
(<field 1><type data>(size), <field2><type data>(size), <field3><type data>(size)<fieldn><type data>(size));

Contoh:

Membuat tabel mahasiswa dengan record nim, nama, dan tanggal lahir

Create tabel mahasiswa,

Nim varchar(20),

Nama varchar(25),

Tanggal_lahir varchar(10);

B. Melihat Struktur Tabel

Untuk dapat melihat struktur table terlebih dahulu dipastikan bahwa kita sudah berada pada database yang mempunyai table tersebut. Perintah yang diberikan untuk melihat struktur table adalah :

Describe <nama_tabel>

Contoh:

Describe mahasiswa;

C. Mengubah Struktur Tabel

Untuk mengubah struktur tabel dalam SQL, harus dipastikan dahulu bahwa memang sudah berada dalam database yang mengandung table yang akan dimodifikasi. Untuk memodifikasi digunakan perintah *Alter*.

1. Menambah field atau kolom

Digunakan untuk menambah satu atau lebih kolom baru ke dalam sebuah tabel.

Untuk menambah field dalam tabel dengan SQL yaitu dengan perintah *Add*.

Sintaks:

Alter tabel <nama tabel>**Add**<nama field><type data>;

Contoh:

Menambahkan field alamat pada tabel mahasiswa

Alter tabel mahasiswa

Add Alamat varchar(50);

2. Mengganti nama field atau kolom

Digunakan untuk mengubah nama kolom lama dengan nama kolom yang baru. Untuk mengubah nama kolom dalam SQL yaitu dengan perintah *to* dan *change*. Sintaks:

```
Alter<tabel>Alter<nama field lama>to<nama field baru>;  
Alter table <nama_tabel> change <nama_field_lama>  
<nama_field_baru> <tipe_data_baru>;
```

Contoh:

Merubah nama field alamat pada tabel mahasiswa dengan nama field baru yaitu alamatasal.

```
Alter mahasiswa Alter alamat to alamatasal;
```

```
Alter table mahasiswa change alamat alamatasal varchar(25);
```

3. Mengubah tipe data field atau kolom

Digunakan untuk mengubah tipe field/kolom tertentu. Tipe data yang baru langsung disebutkan dibelakang nama field/kolom tanpa harus menyebutkan tipe data lama. Bentuk perintahnya adalah:

```
Alter table <nama_tabel> modify <nama_field> <tipe_data_baru>;
```

Contoh:

```
Alter table mahasiswa modify nim longint;
```

4. Menghapus field atau kolom

Penghapusan kolom yang tidak digunakan dimaksudkan untuk menghemat memori database. Untuk menghapus kolom dalam sebuah tabel dalam SQL yaitu dengan perintah *drop* Sintaks:

```
Alter tabel<nama tabel>drop<nama field>;
```

Contoh:

Menghapus kolom alamatasal pada tabel mahasiswa

```
Alter tabel mahasiswa drop alamatasal
```


D. Mengganti Nama Tabel

Dalam pembuatan database dan table seringkali ditemukan harus mengganti penamaan sebuah table. Oleh karena itu DDL sudah menyiapkan suatu sintaks yang dipakai untuk mengubah nama suatu table.

Struktur penulisannya:

```
Alter table <nama_tabel_lama> rename <nama_tabel_baru>
```

Contoh:

```
Alter table mahasiswa rename siswa;
```

E. Menghapus Tabel

Untuk penghematan memori, maka tabel-tabel yang sudah tidak digunakan lagi dapat dihapus. Dalam SQL penghapusan sebuah tabel dapat dilakukan dengan perintah *drop* Sintaks:

```
Drop tabel <nama tabel>;
```

Contoh:

```
Drop tabel mahasiswa
```

2.1.4 Data Manipulation Language (DML)

A. Memasukkan Data

Untuk penambahan data/record pada sebuah tabel dapat dilakukan dengan perintah/sintaks:

```
Insert into <nama tabel (field1, field2,...)> values <(data field1, data field2,...)>
```

Contoh :

```
Insert into mahasiswa (nim, nama_mhs, tgl_lahir) values ('0012052',  
'Farrel Ahmad', '20/08/1983')
```

Nama field/atribut setelah nama tabel dapat diabaikan/tidak dituliskan jika kita ingin mengisi nilai ke semua field dengan urutan field sesuai dengan pendefinisian struktur yang telah dilakukan pada saat pembuatan tabel. Tetapi jika urutan field berbeda atau

terdapat field yang tidak ingin diisi nilainya, maka nama field harus dituliskan. Jika kita melakukan penambahan record dengan perintah :

```
Insert into mahasiswa (nim, nama_mhs) values ('0012052', 'Farrel  
Ahmad')
```

Maka field *tgl_lahir* yang tidak disebutkan dalam perintah insert tersebut akan diisi nilai *null*.

B. Mengubah Data

Untuk pengubahan data pada sebuah tabel, dapat dilakukan dengan perintah/sintaks:

```
Update <nama tabel> set <nama field = nilai baru> where <kreteria  
pemilihan record yang diubah>
```

Contoh:

```
Update mahasiswa set nama_mhs = 'Agus S' where nim='0012052'
```

Jika pada perintah update dengan klausa *where* tidak dituliskan maka perubahan data akan dilakukan untuk semua record/data.

C. Menghapus Data

Penghapusan data/record pada sebuah tabel, dapat dilakukan dengan perintah/sintaks:

```
Delete from <nama tabel> where <kreteria pemilihan>
```

Contoh:

```
Delete from mahasiswa where nim='0012052'
```

Jika pada perintah delete dengan klausa *where* tidak dituliskan maka penghapusan data akan dilakukan untuk semua record/data.

D. Menampilkan Data

Untuk menampilkan data pada SQL secara dasar terbagi atas 3 klausa, yaitu: `select`, `from` dan `where`.

- Klausa `select` digunakan untuk menetapkan daftar atribut/field yang diinginkan sebagai hasil query
- Klausa `from` digunakan untuk menetapkan tabel/gabungan tabel yang akan ditelusuri selama query data dilakukan.
- Klausa `where` bersifat opsional digunakan sebagai kriteria yang harus dipenuhi dalam memperoleh hasil query.

Cara penulisan/sintaks dari ekspresi SQL dasar dengan 3 klausa tersebut adalah:

```
Select<field1,field2>from tabel <nama tabel> where <kriteria>
```

Contoh:

```
Select nim, nama from mahasiswa
```

Dengan perintah diatas maka akan ditampilkan semua data yang ada pada tabel mahasiswa dengan field/kolom nim dan nama sebagai header pada masing-masing kolom dari tabel yang ditampilkan. Jika kita ingin mengubah header kolom tabel yang akan ditampilkan pada hasil query maka tambahkan perintah **as** seperti contoh berikut:

```
Select nim as Nomor_Induk, nama as Nama_Mahasiswa from Mahasiswa
```

Jika kita ingin menampilkan semua field yang terdapat pada tabel, kita dapat memberikan tanda asterik (*) sebelum klausa `from`, seperti contoh berikut:

```
Select * from mahasiswa
```

Jika ingin menampilkan record-record tertentu, misalnya dengan nim = '0012032' kita dapat memberikan perintah sebagai berikut:

```
Select * from mahasiswa where nim = '0012032'
```

Tanda petik pada nim menunjukkan tipe field nim yang bertipe teks/string. Kita juga dapat memberikan operator relasi lainnya seperti : `<`, `>`, `<=`, `>=`, `<>`. Dan dapat juga kita kombinasikan dengan operator logika (`and`, `or` dan `not`). Misalnya kita ingin menampilkan

Mata Kuliah yang diselenggarakan pada semester 5 dengan jumlah sks lebih besar dari 2, maka ekspresi SQL-nya:

```
Select * from kuliah where semester = 5 and sks>2
```

Dapat juga kita menampilkan data dengan kriteria yang berbentuk *range* nilai tertentu dengan menggunakan tambahan klausa **between**, misalnya kita akan menampilkan record-record Mata Kuliah yang diselenggarakan antara semester 1 hingga semester 4

```
Select * from kuliah where semester between 1 and 4
```

Khusus untuk field yang bertipe string kita dapat melakukan pencarian dengan pola tertentu, dengan memanfaatkan karakter '%' (persen) atau '_' (*underscore*) dan tambahan klausa **like** pada klausa where. Tanda % berarti cocok untuk semua substring, sedang tanda _ berarti cocok untuk semua karakter pada posisi yang sesuai.

Berikut adalah perintah untuk menampilkan record mahasiswa dengan nama yang diawali huruf 'D' :

```
Select * from mahasiswa where nama like 'D%'
```

Bentuk lain dari perintah diatas, adalah sebagai berikut:

```
Select * from mahasiswa where left(nama,1)='d'
```

Sedang ekspresi berikut digunakan untuk menampilkan record pada tabel mahasiswa dengan huruf kedua 'd' :

```
Select * from mahasiswa where nama like '_d%'
```

Bentuk lain perintah untuk menampilkan record pada table mahasiswa dengan huruf kedua d adalah :

```
Select * from mahasiswa where substring(nama,2,1)='d'
```

Atau dapat juga diberikan perintah sebagai berikut:

```
Select * from mahasiswa where mid(nama,2,1)='d'
```

Untuk menghasilkan tampilan data dengan nama yang diakhiri huruf 'd' saja, dengan sintaks sebagai berikut:

```
Select * from mahasiswa where nama like '%d'
```

Atau dengan perintah:

```
Select * from mahasiswa where right(nama,1)='d'
```

Untuk menampilkan data-data dengan kondisi yang tidak sesuai, kita dapat menuliskan simbol relasi dengan tanda <> atau dapat juga kita menggunakan klausa **not like**.

Contoh:

```
Select * from mahasiswa where nama not like '%d'
```

Atau:

```
Select * from mahasiswa where right(nama,1)<>'d'
```

Kedua perintah ini digunakan untuk menampilkan semua record dalam table mahasiswa yang tidak berakhiran d untuk nama mahasiswa.

Selain klausa where yang digunakan untuk menyatakan kondisi, kita dapat juga menggunakan klausa **having** dalam pernyataan suatu kondisi tertentu.

Contoh:

```
Select * from mahasiswa having mid(nama,2,1)='d'
```

Klausa from digunakan untuk menetapkan tabel yang kita jadikan sebagai sumber pencarian data. Seperti kita ketahui bahwa basis data terdiri dari sekumpulan tabel yang saling berhubungan. Karena itu, sangat sering nantinya kita temui kebutuhan informasi/data yang mengharuskan penggabungan dari beberapa tabel yang ada (relasi). Contoh sederhananya adalah kebutuhan data akan kuliah dengan dosen yang mengajar kuliah tersebut. Dimana masing-masing tabel dapat dihubungkan/direlasikan dengan kode_dosen pada tabel kuliah dan ode_dosen pada tabel dosen, seperti sintaks berikut:

```
Select * from kuliah, dosen where kuliah.kode_dosen = dosen.kode_dosen
```

Sintaks diatas dapat kita sederhanakan dengan menggunakan alias untuk menyederhanakan penulisan ekspresi query nya, seperti:

```
Select * from kuliah k, dosen d where k.kode_dosen = d.kode_dosen
```

Tanda asterik diatas digunakan sebagai pengganti perintah untuk menampilkan semua field pada masing-masing tabel. Jika kita ingin menampilkan field-field tertentu saja, maka nama tabel atau aliasnya dapat kita gunakan untuk memperjelas asal field yang kita tampilkan tersebut, misalnya:

```
Select k.kode_kul, k.nama_kul, d.nama_dosen from kuliah k, dosen d where  
k.kode_dosen = d.kode_dosen
```

E. Mengurutkan Tampilan

Klausa **Order by** digunakan untuk mengurutkan hasil. Pengurutan dibedakan atas dua macam yaitu: pengurutan secara ascending (**asc**) yaitu pengurutan dari kecil ke besar dan pengurutan secara descending (**desc**) yaitu pengurutan dari besar ke kecil. Contoh perintahnya adalah:

```
Select * from mahasiswa order by nim asc;
```

Selain fungsi-fungsi yang disebutkan diatas terdapat beberapa fungsi lain yang dapat digunakan dalam DML ini, diantaranya:

MAX → untuk mencari nilai maksimal dari suatu kolom

MIN → untuk mencari nilai minimal dari suatu kolom

AVG → mencari nilai rata-rata

SUM → mencari jumlah

COUNT → mencari cacah

GROUP BY → mengelompokkan record

DISTINCT → meniadakan duplikasi hasil record

2.1.5 Data Control Language (DCL)

DCL terdiri dari dua perintah utama yaitu : Grant dan Revoke

Grant digunakan untuk mengijinkan user mengakses table dalam database tertentu. Pemberian hak akses database ini diberikan oleh user lain yang mempunyai hak untuk menggunakan grant.

Struktur penulisannya:

```
Grant <privileges> on <nama_database.nama_tabel> to <nama_user>
```

Bentuk-bentuk privileges yang disediakan oleh SQL diantaranya adalah:

Alter, Create, Delete, Drop, File, Reload, Index, Insert, Process, References, Reload, Select, Shutdown, Update, Usage. Jika ingin memberikan semua hak akses kepada user, maka dapat diberikan perintah:

```
Grant all privileges on akademik.* to agus@localhost;
```

Tanda asterik digunakan untuk menggantikan nama seluruh table yang ada pada database akademik.

Sedangkan jika privileges yang akan diberikan kepada user terbatas maka tuliskan beberapa privileges yang akan diberikan, seperti contoh berikut:

```
Grant select, update on akademik.* to agus@localhost;
```

Revoke merupakan kebalikan dari grant. Revoke digunakan untuk mencabut ijin yang sudah pernah diberikan sebelumnya oleh grant.pencabutan ini dilakukan oleh user yang memiliki hak untuk menggunakan perintah revoke.

Bentuk perintah yang diberikan adalah sebagai berikut:

```
Revoke <privileges> on <nama_database.nama_tabel> from <nama_user>
```

Contoh:

```
Revoke select, update on akademik.* from agus@localhost;
```

2.2 Normalisasi Tabel dan Penentuan Kunci Primer

2.3 Membuat Basis Data Perpustakaan

2.3.1 MySQL Front

Untuk membuat basis data dan tabel perpustakaan agar lebih mudah, dalam implementasinya dalam buku ini menggunakan MySQL Front sebagai interfacenya.

Untuk menjalankan MySQL Front maka pastikan MySQL Server sudah terinstall dan WinMySQL Admin sudah *ter-run* pada komputer.

Kemudian jalankan program MySQL Front dengan klik 2x pada ikon MySQL Front. Maka akan tampil layar MySQL Front sebagai berikut:

database dan table dalam MySQL dengan bantuan MySQL Front sebagai interface-nya. MySQL Front pada dasarnya memiliki tampilan tidak jauh berbeda dibandingkan merancang database dan table menggunakan interface basis teks. Disamping memang dalam MySQL Front disiapkan layar untuk menuliskan perintah SQL juga kita dapat menyimpan dan membuka kumpulan perintah-perintah yang pernah dibuat. Untuk lebih jelasnya, kerjakan perintah-perintah berikut secara langkah demi langkah berurutan.

2.4 Koneksi Basis Data Perpustakaan

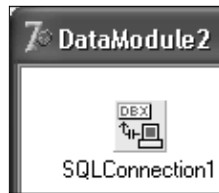
Sebuah aplikasi/form agar dapat terhubung dengan sebuah tabel membutuhkan koneksi ke sebuah basis data. Terdapat banyak pilihan perangkat koneksi, salah satunya adalah dbExpress. Dalam aplikasi perpustakaan ini, koneksi basis data menggunakannya sebagai *tool* koneksi.

Komponen dbExpress terdiri dari beberapa komponen koneksi, baik untuk basis data maupun untuk koneksi sebuah tabel maupun query. SQLConnection merupakan komponen yang digunakan untuk menghubungkan basis data dengan aplikasi.

Setiap komponen koneksi yang digunakan pada aplikasi biasanya dikelompokkan menjadi 1 tempat/kelompok. Tempat yang dipakai untuk pengelompokan tersebut dalam Delphi dikenal dengan istilah DataModule. Untuk membuat sebuah data module yang nantinya digunakan sebagai tempat komponen-komponen koneksi, ikuti langkah-langkah berikut:

- Klik Menu File
- Pilih New
- Pilih Data Module
- Maka akan muncul jendela Data Module, yang nantinya akan kita gunakan sebagai tempat menyimpan beberapa komponen untuk koneksi.

- Tambahkan komponen SqlConnection (Tab dbExpress)



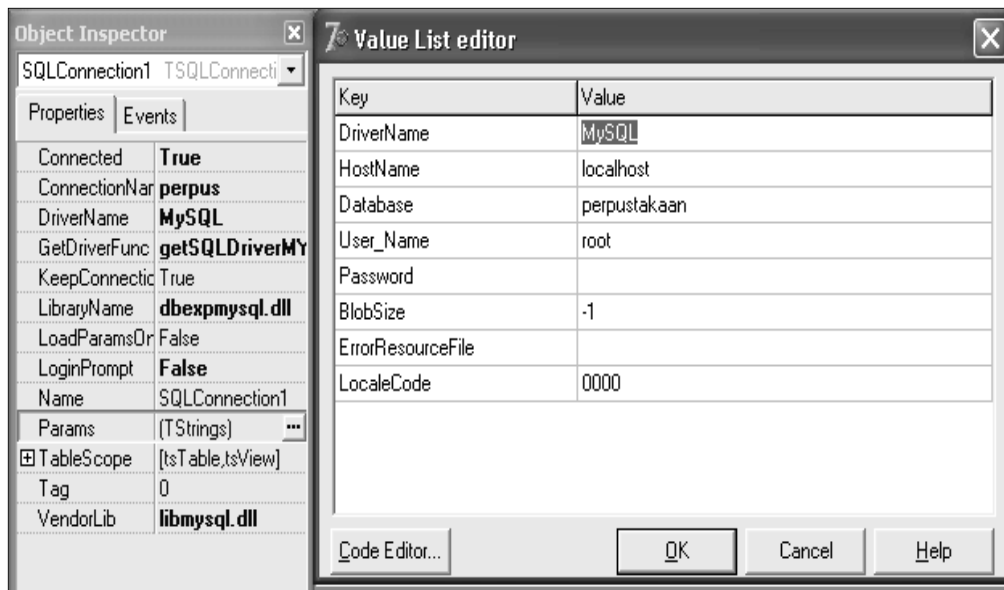
Gambar 4.1.3 DataModule2

- Setting komponen SqlConnection seperti dalam tabel berikut:

Tabel 4.1.2 Setting SQL Connection

No	Komponen	Properties		
1.	SQL Connection	ConnectionString	Perpus	
		LibraryName	Dbexpmysql.dll	
		LoginPrompt	False	
		Name	SqlConnection1	
		Params	DriverName	MySQL
			HostName	localhost
			Database	Perpustakaan
			User_Name	Root
		VendorLib	Libmysql.dll	
		Active	True	

Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini:



Gambar 4.1.4 Value List editor untuk SqlConnection1

BAB 2

GAMBARAN UMUM SISTEM PERPUSTAKAAN

3.1 Pendahuluan

Perpustakaan merupakan tempat berkumpulnya berbagai karya ilmiah dari ilmu pengetahuan. Baik karya klasik maupun kontemporer. Yang secara garis besar karya-karya tersebut berbentuk buku, jurnal, majalah, *compact disk*, dan sebagainya.

Sistem perpustakaan merupakan serangkaian proses yang saling terkait mulai dari penciptaan koleksi yang lebih dikenal sebagai penerbit, pencatatan data koleksi, penentuan tempat koleksi (rak), pembuatan katalog, pencatatan data anggota, pencatatan data peminjaman koleksi, pencatatan pengembalian koleksi, pencatatan denda, dan lain-lain.

Serangkain kegiatan diatas dapat dikatakan merupakan cara manual, dimana untuk masing-masing kegiatan membutuhkan waktu yang cukup lama. Hal tersebut dapat diperpendek masa prosesnya dengan system yang dibantu komputer. System yang berbantuan computer tentunya akan lebih cepat dalam beberapa hal dibandingkan dengan system manual. Sebagai contoh adalah pencarian suatu koleksi akan lebih mudah dan cepat dibandingkan dengan cara manual.

Terdapat beberapa proses dalam membuat system terkomputerisasi atau yang sering dikenal sebagai system informasi. Proses yang secara umum dilakukan adalah pengambilan data pada obyek, analisa data, perancangan system, implementasi system, pengujian system, sampai pada penarikan kesimpulan terhadap system.

Tahapan perancangan system terdiri dari beberapa bagian, diantaranya adalah perancangan diagram konteks, perancangan bagan berjenjang, perancangan diagram aliran data, perancangan tabel-tabel, perancangan tampilan masukan data dan keluaran data. Untuk tahapan implementasi system lebih kearah pengkodean dengan bahasa pemrograman dan penyiapan basis data yang digunakan sebagai tempat tabel-tabel. Sedangkan tabel-tabel tersebut yang dipakai sebagai tempat penampungan data-data.

Oleh karenanya proses-proses dalam system perpustakaan secara umum tidak berubah, yang sedikit bergeser adalah system manual hanya mengandalkan pencatatan semua data secara manual, sedang untuk system terkomputerisasi, pencatatan-pencatatan

yang dilakukan secara manual digantikan oleh computer dengan beberapa perhitungan dapat dilakukan secara otomatis, seperti : kapan tanggal pengembalian, berapa jumlah denda, katalog perpustakaan yang dapat diakses secara online dengan data yang terbaru/*update*.

3.2 Komponen-komponen Perpustakaan

Perpustakaan merupakan suatu system yang terdiri dari sekumpulan obyek yang saling berinteraksi. Obyek-obyek perpustakaan secara umum terdiri atas : koleksi, petugas/pengelola dan anggota.

Sedangkan dengan perkembangan kebutuhan akan informasi dan kemudahan dalam pengaksesan informasi tersebut, system dalam perpustakaan juga dituntut untuk dapat memenuhi kebutuhan informasi yang ada. Sehingga selain obyek-obyek yang disebutkan diatas, terdapat beberapa komponen lain yang terkait dilibatkan dalam system perpustakaan. Komponen-komponen tersebut antara lain : penerbit, penggolongan kategori koleksi, rak koleksi, transaksi peminjaman, transaksi pengembalian, dan lain sebagainya.

Secara detail dapat dijelaskan sebagai berikut:

A. Anggota

Anggota merupakan suatu obyek utama dalam system perpustakaan, yang menjadi pelaku utama baik sebagai peminjam koleksi maupun sebagai pengembali koleksi perpustakaan. Terdapat beberapa data yang digunakan sebagai data pokok dari obyek anggota, diantaranya : nama, tanda pengenal, nomor telepon, alamat dan data lainnya yang dianggap penting dalam kaitannya sebagai pengikat untuk keanggotaannya dalam system perpustakaan.

Data pengikat yang dimaksudkan adalah untuk meminimalkan kemungkinan penggelapan koleksi perpustakaan. Hal tersebut memang dapat dilakukan secara sengaja maupun tidak sengaja, sehingga untuk mengurangi kemungkinan penggelapan koleksi maka diharapkan data yang tercatat dalam system dapat digunakan sebagai data awal pelacakan keberadaan anggota.

B. Klasifikasi Koleksi

Klasifikasi koleksi perpustakaan merupakan suatu metode yang sudah ada dan baku. Memang yang menjadi acuan hingga saat ini adalah buku, ini dikarenakan sampai sekarang buku merupakan koleksi utama.

Untuk pengklasifikasian buku pada perpustakaan terdapat beberapa aturan. Aturan atau klasifikasi DEWEY merupakan aturan atau acuan yang secara luas digunakan di beberapa perpustakaan.

Untuk system komputerisasi perpustakaan, klasifikasi koleksi diserahkan pada acuan yang digunakan pada perpustakaan tersebut. Sehingga masing-masing perpustakaan akan memiliki klasifikasi yang sedikit berbeda dengan yang lainnya.

C. Penerbit

Penerbit merupakan obyek yang menerbitkan atau mengeluarkan koleksi dengan suatu lisensi yang legal. dimasukkan dalam system digunakan untuk memudahkan operator system dalam pemasukan data-data penerbit yang sama. Ini dikarenakan satu penerbit sangat mungkin menerbitkan banyak judul koleksi, sehingga pemasukan data yang sama sangat mungkin akan berulang.

D. Rak

Rak koleksi merupakan suatu tempat peletakan/menyimpan koleksi. Data rak dibutuhkan untuk pencetakan keterangan koleksi yang berada pada rak tersebut, sehingga untuk memudahkan dalam pengidentifikasian lokasi koleksi maka data rak merupakan salah satu data utama dalam system informasi perpustakaan.

E. Bahasa

Suatu koleksi selain memiliki judul dan hal-hal utama lainnya, juga terkadang anggota perpustakaan akan mencari informasi tentang bahasa apa yang digunakan oleh koleksi tersebut. Untuk memudahkan penyediaan informasi tersebut maka diawal pemasukan data, bahasa yang digunakan merupakan data acuan utama dalam system yang akan dibangun.

F. Petugas

Selain hal-hal yang telah disebut dan dijelaskan diatas juga terdapat beberapa hal lainnya. Satu diantaranya yakni data petugas. Ini penting untuk dientrikan karena untuk ketersediaan data, siapa yang bertanggung jawab tentang sirkulasi pada saat tertentu akan diketahui jika data petugas disediakan.

Beberapa komponen lainnya tidak dibahas dalam buku ini, karena untuk system yang dibangun hanya meliputi obyek-obyek yang utama dari system informasi perpustakaan secara umum. Oleh karena itu sangat dimungkinkan suatu system informasi perpustakaan akan berbeda antara satu dan lainnya.

3.3 Klasifikasi Buku/Koleksi Perpustakaan

Klasifikasi buku pada perpustakaan secara umum di Indonesia, salah satunya menggunakan metode DDC (*Dewey Decimal Classification*) atau sering disebut dengan Klasifikasi Persepuluhan Dewey. Seperti yang disinggung diatas perkembangan teknologi juga berimbas pada perkembangan lainnya seperti juga perpustakaan dimana koleksi sebuah perpustakaan pada saat ini bukan hanya terdiri atas buku saja, melainkan banyak yang lainnya seperti *Compact Disk* (CD), DVD, dan lain-lainnya. Oleh karena itu dalam membangun sebuah system informasi perpustakaan tentunya system harus dapat mengadopsi perkembangan tersebut.

Sehingga untuk memudahkan operator dan system dapat menyediakan data/informasi terkait dengan klasifikasi koleksi maka dalam membangun system informasi ini, khususnya terkait dengan klasifikasi koleksi diberi kelonggaran dalam pemasukan data klasifikasi, ini berarti klasifikasi koleksi satu perpustakaan dengan perpustakaan lainnya akan berbeda menurut kebijakan yang diberlakukan pada perpustakaan bersangkutan.



Sehingga validitas data klasifikasi akan sangat tergantung pada kebijakan dan kebenaran operator/petugas dalam memasukkan data klasifikasi koleksi.

BAB 3 DATA RAK

4.1 Rancangan Tabel

Data rak mengandung 2 buah field, kode rak dan nama rak. Untuk masing-masing field memiliki tipe data varchar dengan jumlah karakter 5 dan 20 karakter. Kode rak digunakan sebagai kunci dalam index datanya. Secara detail, rancangan table data rak dapat dilihat sebagai berikut:

Tabel 4.1.1 Tabel Rak

Name	Type	Null	Default
 kode_rak	varchar(5)	No	0
 nama_rak	varchar(20)	No	0

4.2 Koneksi Tabel

Sebuah aplikasi/form, dapat terhubung dengan tabel membutuhkan beberapa *tool*/perangkat koneksi. Terdapat banyak pilihan perangkat koneksi, salah satunya adalah dbExpress. Untuk koneksi basisdatanya, setiap basisdata biasanya hanya membutuhkan sebuah SQLConnection (seperti dijelaskan pada bab 1).

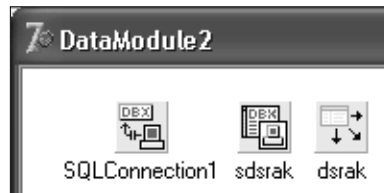
Sehingga untuk aplikasi perpustakaan ini, hanya membutuhkan sebuah SQLConnection. Sedangkan untuk koneksi tabel maupun query, akan membutuhkan sebuah SQLDataset atau sebuah SQLQuery atau dapat juga berbentuk SimpleDataset.

Komponen lainnya yang biasa digunakan untuk sebuah aplikasi adalah DataSource yang berada pada Tab DataAccess. Komponen ini berguna untuk pengaksesan data secara langsung ke tabel dengan menggunakan komponen-komponen pada Tab Data Controls.

Setiap komponen koneksi yang digunakan pada aplikasi biasanya dikelompokkan menjadi 1 tempat/kelompok. Tempat yang dipakai untuk pengelompokkan tersebut dalam Delphi dikenal dengan istilah DataModule. Untuk lebih jelasnya, mari kita koneksikan tabel rak dengan beberapa komponen koneksi yang digunakan pada form rak, secara detail sebagai berikut:

Untuk mengaktifkan data module langkah-langkahnya adalah sebagai berikut:

- Klik Menu File
- Pilih New
- Pilih Data Module
- Maka akan muncul jendela Data Module, yang nantinya akan kita gunakan sebagai tempat menyimpan beberapa komponen untuk koneksi.
- Tambahkan komponen SimpleDataSet (Tab dbExpress) & DataSource (Tab DataAccess)

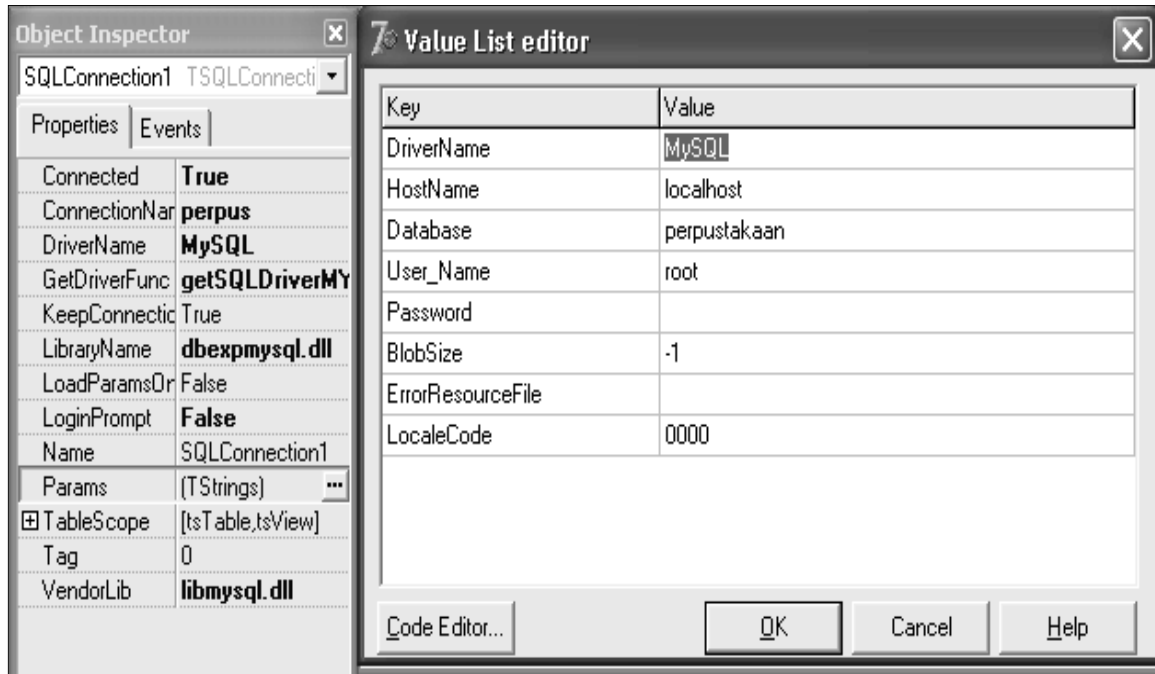


Gambar 4.1.3 DataModule2

- Setting masing-masing komponen sebagai berikut:

Tabel 4.1.2 Komponen Delphi dalam Data Module2 untuk Rak Koleksi

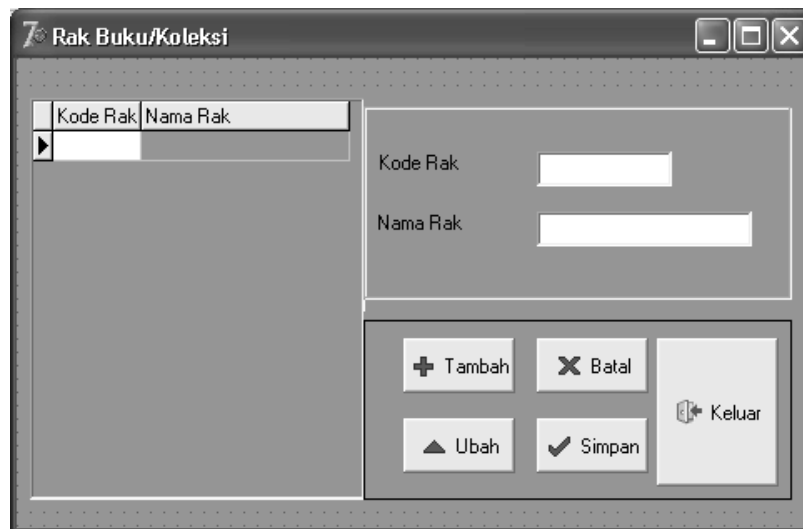
No	Komponen	Properties			
1.	SQL Connection	ConnectionName	Perpus		
		LibraryName	Dbexpmysql.dll		
		LoginPrompt	False		
		Name	SQLConnection1		
		Params	DriverName	MySQL	
			HostName	localhost	
			Database	Perpustakaan	
			User_Name	Root	
			VendorLib	Libmysql.dll	
Active	True				
2.	SimpleDataSet	Connection	SQLConnection1		
		Dataset	CommandType	ctTable	
			CommandText	Rak_buku1	
		Name	sdsrak		
		Active	True		
3.	DataSource	Dataset	Sdsrak		
		Name	Dsrak		



Gambar 4.1.4 Value List editor untuk TSQLConnection1

4.1 Rancangan Form Rak

Form Rak akan mengandung beberapa komponen dari Delphi, untuk mempermudah pengaplikasiannya maka komponen-komponen yang dipakai merupakan komponen-komponen standard Delphi. Secara detail komponen-komponen yang dipakai pada form rak adalah sebagai berikut:



Gambar 4.1.1 Form Rak


Untuk rancangan form diatas membutuhkan beberapa komponen, secara detail lihat table berikut:

Tabel 4.1.1 Komponen Delphi dalam Rak Koleksi

No	Komponen	Properties	
1.	Form	Caption	Rak Buku/Koleksi
2.	DBGrid1	Columns	*
		Datasource	Datamodule2.dsarak
3.		GroupBox	Caption
4.	Label1	Caption	Kode Rak
5.	Label2	Caption	Nama Rak
6.	DBEdit1	DataSource	DataModule2.dsarak
		DataField	Kode_rak
		Name	DBEdit1
7.	DBEdit2	DataSource	DataModule2.dsarak
		DataField	Nama_rak
		Name	DBEdit2
8.	Bitbtn1	Caption	Tambah
		Glyph	Sesuaikan gambar yang ada
		Name	Btambah
9.	Bitbtn2	Caption	Batal
		Glyph	Sesuaikan gambar yang ada
		Name	Bbatal
10.	Bitbtn3	Caption	Ubah
		Glyph	Sesuaikan gambar yang ada
		Name	Bubah
11.	Bitbtn4	Caption	Simpan
		Glyph	Sesuaikan gambar yang ada
		Name	Bsimpan
12.	Bitbtn5	Caption	Keluar
		Glyph	Sesuaikan gambar yang ada
		Name	Bkeluar

Penjelasan:

* Columns pada DBGrid1 terdiri 2 kolom yakni Kode Rak dan Nama Rak, untuk mengaktifkan kolom tersebut ikuti langkah-langkah sebagai berikut:

- Pada Properties DBGrid1 klik GridColumns  ...
- Maka akan muncul jendela baru seperti berikut :



Gambar 4.1.2 Editing DBGrid

- Untuk kode_rak dan nama_rak dapat disetting propertiesnya sesuai dengan yang akan ditentukan.

Untuk Datasource, dalam program ini menggunakan datamodule yang diberi nama datamodule2,

Komponen-komponen yang dibutuhkan pada form rak sudah terpenuhi, maka selanjutnya kita akan menambahkan program yang diperlukan, seperti yang dijelaskan pada bagian berikutnya.

4.2 Proses Tambah Data Rak

Proses tambah data rak merupakan proses yang terjadi pada form rak yang merupakan proses-proses bagian dari 5 proses yang ada. Proses tambah diharapkan menjadi suatu proses mampu digunakan untuk menambah data pada Button/Tombol Tambah yang diberi nama pada properties-nya adalah btambah.

Untuk itu event handler pada Onclick button tambah, kemudian tuliskan programnya sebagai berikut:

```
procedure TForm12.btambahClick(Sender: TObject);
begin
groupbox1.Enabled :=true;
dbedit1.enabled :=true;
dbedit2.Enabled :=true;
datamodule2.sdsrak.Append ;
btambah.Enabled :=false;
bubah.Enabled :=false;
bbatal.Enabled :=true;
bsimpan.Enabled :=true;
bkeluar.Enabled :=false;
end;
```

4.3 Proses Ubah Data Rak

Proses ubah data rak diharapkan menjadi suatu proses mampu digunakan untuk mengubah data pada Button/Tombol Ubah yang diberi nama pada properties-nya adalah bubah.

Untuk itu event handler pada Onclick button ubah, kemudian tuliskan programnya sebagai berikut:

```
procedure TForm12.bubahClick(Sender: TObject);
begin
groupbox1.Enabled :=true;
dbedit1.Enabled :=true;
dbedit2.Enabled :=true;
datamodule2.sdsrak.Edit ;
btambah.Enabled :=false;
bubah.Enabled :=false;
bbatal.Enabled :=true;
bsimpan.Enabled :=true;
bkeluar.Enabled :=false;
end;
```

4.4 Proses Batal Data Rak

Proses batal data rak diharapkan menjadi suatu proses mampu digunakan untuk membatalkan suatu proses lain yang terjadi pada form. Proses Batal pada properties-nya diberi nama adalah bbatal.

Untuk itu event handler pada Onclick button batal, kemudian tuliskan programnya sebagai berikut:

```
procedure TForm12.bbatalClick(Sender: TObject);
begin
groupbox1.Enabled :=false;
dbedit1.Enabled :=false;
dbedit2.Enabled :=false;
datamodule2.sdsrak.Cancel ;
btambah.Enabled :=true;
bubah.Enabled :=true;
bbatal.Enabled :=false;
```

```

bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
end;

```

4.5 Proses Simpan Data Rak

Proses simpan data rak diharapkan menjadi suatu proses mampu digunakan klarifikasi tentang kebenaran data serta untuk menyimpan data yang sudah diklarifikasi oleh operator sistem. Proses Simpan pada properties-nya diberi nama adalah bsimpan.

Untuk itu event handler pada Onclick button simpan, kemudian tuliskan programnya sebagai berikut:

```

procedure TForm12.bsimpanClick(Sender: TObject);
begin
if   messagedlg('Apakah      Datanya      Sudah      Benar      dan      Lengkap
?',mtconfirmation,[mbytes,mbno],0)=mryes then
begin
dbedit1.Enabled :=false;
dbedit2.Enabled :=false;
datamodule2.sdsrak.Post ;
datamodule2.sdsrak.ApplyUpdates(0);
datamodule2.sdsrak.Refresh ;
btambah.Enabled :=true;
bubah.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
groupbox1.Enabled :=false;
end;
end;

```

Listing Program secara lengkap menjadi:

```

unit rak;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, Mask, DBCtrls, Grids, DBGrids;

```

```

type
  TForm12 = class(TForm)
    DBGrid1: TDBGrid;
    GroupBox1: TGroupBox;
    Label2: TLabel;
    Label1: TLabel;
    DBEdit2: TDBEdit;
    DBEdit1: TDBEdit;
    GroupBox2: TGroupBox;
    btambah: TBitBtn;
    bbatal: TBitBtn;
    bubah: TBitBtn;
    bsimpan: TBitBtn;
    bkeluar: TBitBtn;
    procedure btambahClick(Sender: TObject);
    procedure bbatalClick(Sender: TObject);
    procedure bubahClick(Sender: TObject);
    procedure bsimpanClick(Sender: TObject);
    procedure bkeluarClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormActivate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form12: TForm12;

implementation

uses dm;

{$R *.dfm}

procedure TForm12.btambahClick(Sender: TObject);
begin
  groupbox1.Enabled :=true;
  dbedit1.enabled :=true;
  dbedit2.Enabled :=true;
  datamodule2.sdsrak.Append ;

```

```

btambah.Enabled :=false;
bubah.Enabled :=false;
bbatal.Enabled :=true;
bsimpan.Enabled :=true;
bkeluar.Enabled :=false;
end;

procedure TForm12.bbatalClick(Sender: TObject);
begin
groupbox1.Enabled :=false;
dbedit1.Enabled :=false;
dbedit2.Enabled :=false;
datamodule2.sdsrak.Cancel ;
btambah.Enabled :=true;
bubah.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
end;

procedure TForm12.bubahClick(Sender: TObject);
begin
groupbox1.Enabled :=true;
dbedit1.Enabled :=true;
dbedit2.Enabled :=true;
datamodule2.sdsrak.Edit ;
btambah.Enabled :=false;
bubah.Enabled :=false;
bbatal.Enabled :=true;
bsimpan.Enabled :=true;
bkeluar.Enabled :=false;
end;

procedure TForm12.bsimpanClick(Sender: TObject);
begin
if      messagedlg('Apakah      Datanya      Sudah      Benar      dan      Lengkap
?',mtconfirmation,[mbyes,mbno],0)=mryes then
begin
dbedit1.Enabled :=false;
dbedit2.Enabled :=false;
datamodule2.sdsrak.Post ;
datamodule2.sdsrak.ApplyUpdates(0);

```

```

datamodule2.sdsrak.Refresh ;
btambah.Enabled :=true;
bubah.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
groupbox1.Enabled :=false;
end;
end;

procedure TForm12.bkeluarClick(Sender: TObject);
begin
close;
end;

procedure TForm12.FormCreate(Sender: TObject);
begin
groupbox1.Enabled :=false;
btambah.Enabled :=true;
bubah.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
end;

procedure TForm12.FormActivate(Sender: TObject);
begin
groupbox1.Enabled :=false;
btambah.Enabled :=true;
bubah.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
end;

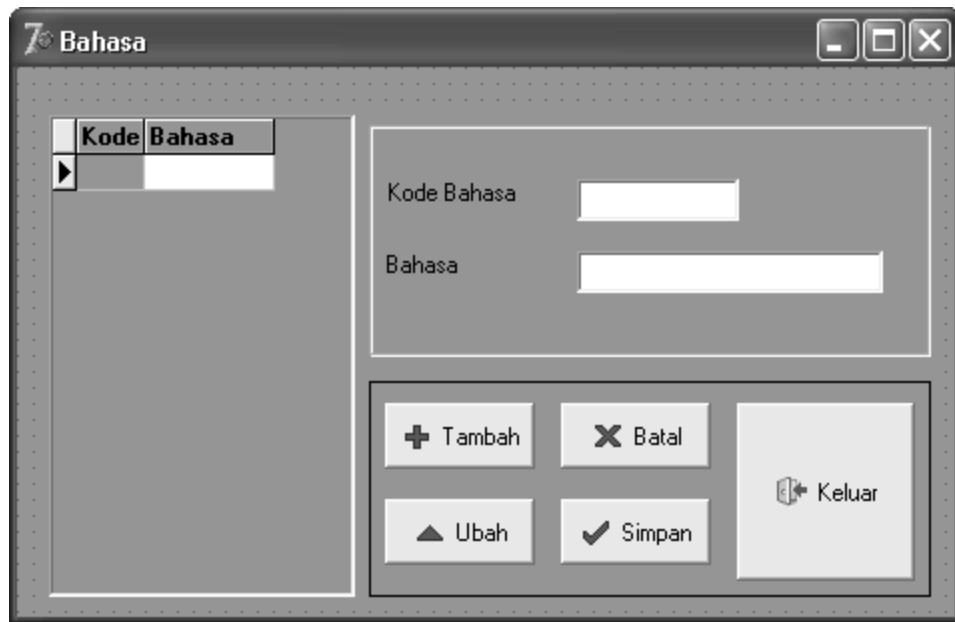
end.

```

BAB 4 DATA BAHASA

5.1 Rancangan Form Bahasa

Seperti pada form rak, form bahasa akan mengandung beberapa komponen dari Delphi yang merupakan komponen-komponen standard Delphi. Secara detail komponen-komponen yang dipakai pada form bahasa adalah sebagai berikut:



Gambar 5.1.1 Form Bahasa


Untuk rancangan form diatas membutuhkan beberapa komponen, secara detail lihat tabel berikut:

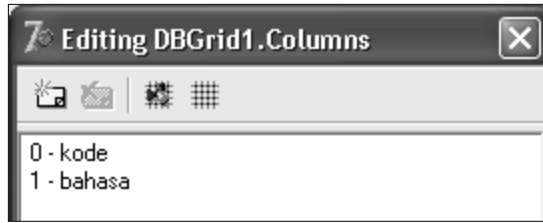
Tabel 5.1.1 Komponen Delphi dalam Bahasa

No	Komponen	Properties	
1.	Form	Caption	Bahasa
2.	DBGrid1	Columns	*
		Datasource	Datamodule2.dsbhs
3.		GroupBox	Caption
4.	Label1	Caption	Kode Bahasa
5.	Label2	Caption	Bahasa
6.	DBEdit1	DataSource	DataModule2.dsbhs
		DataField	Kode
		Name	DBEdit1
7.	DBEdit2	DataSource	DataModule2.dsbhs
		DataField	Bahasa
		Name	DBEdit2
8.	Bitbtn1	Caption	Tambah
		Glyph	Sesuaikan gambar yang ada
		Name	Btambah
9.	Bitbtn2	Caption	Batal
		Glyph	Sesuaikan gambar yang ada
		Name	Bbatal
10.	Bitbtn3	Caption	Ubah
		Glyph	Sesuaikan gambar yang ada
		Name	Bubah
11.	Bitbtn4	Caption	Simpan
		Glyph	Sesuaikan gambar yang ada
		Name	Bsimpan
12.	Bitbtn5	Caption	Keluar
		Glyph	Sesuaikan gambar yang ada
		Name	Bkeluar

Penjelasan:

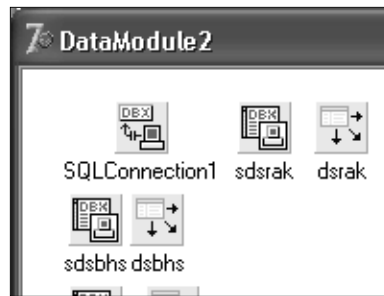
* Columns pada DBGrid1 terdiri 2 kolom yakni Kode dan Bahasa, untuk mengaktifkan kolom tersebut ikuti langkah-langkah sebagai berikut:

- Pada Properties DBGrid1 klik GridColumns  ...
- Maka akan muncul jendela baru seperti berikut :



Gambar 5.1.2 Editing DBGrid

- Untuk kode dan bahasa dapat disetting propertiesnya sesuai dengan yang akan ditentukan.
- Tambahkan 2 buah komponen pada datamodule2, yakni sebuah simpledataset dan sebuah datasource, seperti pada form rak, sehingga datamodule2 akan menjadi seperti gambar berikut:



Gambar 5.1.3 DataModule2

- Setting masing-masing komponen sebagai berikut:

Tabel 5.1.2 Komponen Delphi dalam Data Module2 untuk Bahasa

No	Komponen	Properties		
2.	SimpleDataSet	Connection	SQLConnection1	
		Dataset	CommandType	ctTable
			CommandText	Bahasa 1
		Name	Sdsbhs	
		Active	True	
3.	DataSource	Dataset	Sdsbhs	
		Name	Dsbhs	

Komponen-komponen untuk rancangan pada form bahasa sudah cukup, maka langkah selanjutnya adalah menambahkan source code yang diperlukan. Tidak jauh berbeda pada form rak, form bahasa memiliki proses-proses yang sama seperti pada form rak.

4.2 Proses Tambah Bahasa

Proses tambah bahasa dari sisi pemrograman/listing programnya akan sama dengan listing program pada rak buku, karena setiap proses yang sama sedikit banyaknya akan diharapkan memiliki dampak yang sama pada keseluruhan program, hal ini akan terkait dengan konsistensi pemrograman secara keseluruhan.

Perbedaan yang ada diantara kedua listing program hanya pada akses simpledataset saja. Pada form rak simpledataset yang digunakan adalah sdsrak sedangkan pada form bahasa yang digunakan sdsbhs, secara lengkap dapat dilihat pada event handler pada Onclick button tambah, sebagai berikut:

```
procedure TForm1.btambahClick(Sender: TObject);
begin
groupbox1.Enabled :=true;
dbedit1.enabled :=true;
dbedit2.Enabled :=true;
datamodule2.sdsbhs.Append ;
datamodule2.sdsbhs.FieldName('kode').AsString
:=inttostr(datamodule2.sdsbhs.recordcount+1);
btambah.Enabled :=false;
bubah.Enabled :=false;
bbatal.Enabled :=true;
bsimpan.Enabled :=true;
bkeluar.Enabled :=false;
end;
```

Pada penggalan program diatas terdapat perintah untuk penjumlahan otomatis pada kode, artinya kode yang ditampilkan akan secara otomatis dihitung oleh bagian ini, untuk dijadikan kode baru, jumlah record yang ada akan ditambah 1 untuk digunakan sebagai kode record selanjutnya.

5.3 Proses Ubah Bahasa

Proses ubah bahasa juga demikian, akan sama pada data rak, yang membedakan hanya pada simpledataset yang digunakan. Selain itu pada form bahasa tidak dilengkapi dengan proses hapus, mengapa? Pada aplikasi ini diasumsikan bahwa setiap data yang telah tersimpan dianggap sudah benar, sehingga tidak perlu dihapus. Dan jika memang sangat dibutuhkan untuk menghilangkan/menghapus sebuah data, maka lakukan perubahan data saja, sehingga hal ini tidak berakibat pada perbedaan antara jumlah record dengan kode otomatisnya.

Untuk itu event handler pada Onclick button ubah adalah sebagai berikut:

```
procedure TForm11.bubahClick(Sender: TObject);
begin
groupbox1.Enabled :=true;
dbedit1.Enabled :=true;
dbedit2.Enabled :=true;
datamodule2.sdsbhs.Edit ;
btambah.Enabled :=false;
bubah.Enabled :=false;
bbatal.Enabled :=true;
bsimpan.Enabled :=true;
bkeluar.Enabled :=false;
end;
```

4.4 Proses Batal BahasaData Rak

Proses batal bahasa merupakan suatu proses yang digunakan untuk membatalkan suatu proses lain yang terjadi pada form bahasa. Proses Batal pada properties-nya diberi nama adalah bbatal.

Untuk itu event handler pada Onclick button batal, kemudian tuliskan programnya sebagai berikut:

```
procedure TForm11.bbatalClick(Sender: TObject);
begin
groupbox1.Enabled :=false;
dbedit1.Enabled :=false;
dbedit2.Enabled :=false;
datamodule2.sdsbhs.Cancel ;
btambah.Enabled :=true;
```

```

bubah.Enabled :=true;
bhapus.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
end;

```

4.5 Proses Simpan Bahasa

Proses simpan bahasa juga merupakan suatu proses yang mampu digunakan untuk klarifikasi tentang kebenaran data serta untuk menyimpan data yang sudah diklarifikasi oleh operator sistem. Proses Simpan pada properties-nya diberi nama adalah bsimpan.

Untuk itu event handler pada Onclick button simpan, kemudian tuliskan programnya sebagai berikut:

```

procedure TForm1.bsimpanClick(Sender: TObject);
begin
if   messagedlg('Apakah      Datanya      Sudah      Benar      dan      Lengkap
?',mtconfirmation,[mbytes,mbno],0)=mryes then
begin
dbedit1.Enabled :=false;
dbedit2.Enabled :=false;
datamodule2.sdsbhs.Post ;
datamodule2.sdsbhs.ApplyUpdates(0);
datamodule2.sdsbhs.Refresh ;
btambah.Enabled :=true;
bubah.Enabled :=true;
bhapus.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
groupbox1.Enabled :=false;
end;
end;

```

Listing Program secara lengkap menjadi:

```
unit bhs;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, Mask, DBCtrls, Grids, DBGrids;

type
  TForm11 = class(TForm)
    DBGrid1: TDBGrid;
    GroupBox1: TGroupBox;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    DBEdit1: TDBEdit;
    Label1: TLabel;
    GroupBox2: TGroupBox;
    btambah: TBitBtn;
    bbatal: TBitBtn;
    bubah: TBitBtn;
    bsimpan: TBitBtn;
    bkeluar: TBitBtn;
    procedure btambahClick(Sender: TObject);
    procedure bbatalClick(Sender: TObject);
    procedure bubahClick(Sender: TObject);
    procedure bsimpanClick(Sender: TObject);
    procedure bkeluarClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form11: TForm11;

implementation

uses dm;
```

```

{$R *.dfm}

procedure TForm1.btambahClick(Sender: TObject);
begin
groupbox1.Enabled :=true;
dbedit1.enabled :=true;
dbedit2.Enabled :=true;

datamodule2.sdsbhs.Append ;
datamodule2.sdsbhs.FieldByName('kode').AsString
:=inttostr(datamodule2.sdsbhs.recordcount+1);
btambah.Enabled :=false;
bubah.Enabled :=false;
bbatal.Enabled :=true;
bsimpan.Enabled :=true;
bkeluar.Enabled :=false;
end;

procedure TForm1.bbatalClick(Sender: TObject);
begin
groupbox1.Enabled :=false;
dbedit1.Enabled :=false;
dbedit2.Enabled :=false;
datamodule2.sdsbhs.Cancel ;
btambah.Enabled :=true;
bubah.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
end;

procedure TForm1.bubahClick(Sender: TObject);
begin
groupbox1.Enabled :=true;
dbedit1.Enabled :=true;
dbedit2.Enabled :=true;
datamodule2.sdsbhs.Edit ;
btambah.Enabled :=false;
bubah.Enabled :=false;
bbatal.Enabled :=true;
bsimpan.Enabled :=true;
bkeluar.Enabled :=false;
end;

```

```

end;

procedure TForm11.bsimpanClick(Sender: TObject);
begin
if messagedlg('Apakah Datanya Sudah Benar dan Lengkap
?',mtconfirmation,[mbyes,mbno],0)=mryes then
begin
dbedit1.Enabled :=false;
dbedit2.Enabled :=false;
datamodule2.sdsbhs.Post ;
datamodule2.sdsbhs.ApplyUpdates(0);
datamodule2.sdsbhs.Refresh ;
btambah.Enabled :=true;
bubah.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
groupbox1.Enabled :=false;
end;
end;

procedure TForm11.bkeluarClick(Sender: TObject);
begin
close;
end;

procedure TForm11.FormCreate(Sender: TObject);
begin

groupbox1.Enabled :=false;

btambah.Enabled :=true;
bubah.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
end;

end.

```


BAB 5 DATA PENERBIT

5.1 Rancangan Form Penerbit

Form penerbit terdiri dari 3 field, yakni: kode penerbit, nama penerbit dan alamat. Field kode digunakan sebagai kunci/*key* dalam table penerbit, sedangkan nama penerbit dan alamat penerbit merupakan identitas yang melekat pada suatu penerbit. Data ini dapat ditambahkan dengan nomor telepon, atau kalau hanya mengandung 3 field seperti ini, dapat menggunakan field alamat untuk tempat menuliskan nomor telepon/nomor kontak.

Seperti pada form-form sebelumnya, form penerbit mengandung beberapa komponen dari Delphi yang merupakan komponen-komponen standard Delphi. Seperti pada gambar berikut:

Kode Penerbit	Nama Penerbit
2	Elex Media Komputindo
3	Informatika
4	Salemba Infotek
5	J & J Learning Yogyakarta
6	The McGraw-Hill Companies, Inc

Gambar 6.1.1 Form Penerbit


Untuk rancangan form diatas membutuhkan beberapa komponen, secara detail lihat tabel berikut:

Tabel 6.1.1 Komponen Delphi dalam Data Penerbit

No	Komponen	Properties	
1.	Form	Caption	Penerbit Buku
2.	DBGrid1	Columns	*
		Datasource	Datamodule2.dspenerbit
3.		GroupBox	Caption
4.	Label1	Caption	Kode
5.	Label2	Caption	Penerbit
6.	Label3	Caption	Alamat
7.	DBEdit1	DataSource	DataModule2.dspenerbit
		DataField	Kode
		Name	DBEdit1
8.	DBEdit2	DataSource	DataModule2.dspenerbit
		DataField	Nama
		Name	DBEdit2
9.	DBEdit3	DataSource	DataModule2.dspenerbit
		DataField	Alamat
		Name	DBEdit3
10.	Bitbtn1	Caption	Tambah
		Glyph	Sesuaikan gambar yang ada
		Name	Btambah
11.	Bitbtn2	Caption	Batal
		Glyph	Sesuaikan gambar yang ada
		Name	Bbatal
12.	Bitbtn3	Caption	Ubah
		Glyph	Sesuaikan gambar yang ada
		Name	Bubah
13.	Bitbtn4	Caption	Simpan
		Glyph	Sesuaikan gambar yang ada
		Name	Bsimpan
14.	Bitbtn5	Caption	Keluar
		Glyph	Sesuaikan gambar yang ada
		Name	Bkeluar

Penjelasan:

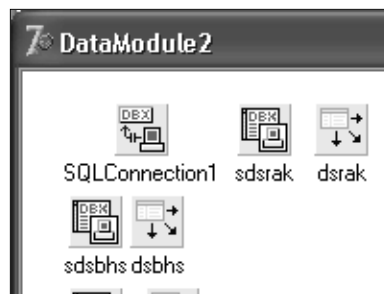
* Columns pada DBGrid1 terdiri 3 kolom yakni Kode, Nama dan Alamat, untuk mengaktifkan kolom tersebut ikuti langkah-langkah sebagai berikut:

- Pada Properties DBGrid1 klik GridColumns  ...
- Maka akan muncul jendela baru seperti berikut :



Gambar 6.1.2 Editing DBGrid

- Untuk kode, nama dan alamat dapat disetting propertiesnya sesuai dengan yang akan ditentukan.
- Tambahkan 2 buah komponen pada datamodule2, yakni sebuah simpledataset dan sebuah datasource, seperti pada form rak, sehingga datamodule2 akan menjadi seperti gambar berikut:



Gambar 5.1.3 DataModule2

- Setting masing-masing komponen sebagai berikut:

Tabel 5.1.2 Komponen Delphi dalam Data Module2 untuk Bahasa

No	Komponen	Properties		
2.	SimpleDataSet	Connection	SQLConnection1	
		Dataset	CommandType	ctTable
			CommandText	Bahasa1
		Name	Sdsbhs	
		Active	True	
3.	DataSource	Dataset	Sdsbhs	
		Name	Dsbhs	

Komponen-komponen untuk rancangan pada form bahasa sudah cukup, maka langkah selanjutnya adalah menambahkan source code yang diperlukan. Tidak jauh berbeda pada form rak, form bahasa memiliki proses-proses yang sama seperti pada form rak.

4.2 Proses Tambah Bahasa

Proses tambah bahasa dari sisi pemrograman/listing programnya akan sama dengan listing program pada rak buku, karena setiap proses yang sama sedikit banyaknya akan diharapkan memiliki dampak yang sama pada keseluruhan program, hal ini akan terkait dengan konsistensi pemrograman secara keseluruhan.

Perbedaan yang ada diantara kedua listing program hanya pada akses simpledataset saja. Pada form rak simpledataset yang digunakan adalah sdsrak sedangkan pada form bahasa yang digunakan sdsbhs, secara lengkap dapat dilihat pada event handler pada Onclick button tambah, sebagai berikut:

```
procedure TForm1.btambahClick(Sender: TObject);
begin
  groupbox1.Enabled :=true;
  dbedit1.enabled :=true;
  dbedit2.Enabled :=true;
  datamodule2.sdsbhs.Append ;
  datamodule2.sdsbhs.FieldByName('kode').AsString
:=inttostr(datamodule2.sdsbhs.recordcount+1);
  btambah.Enabled :=false;
  bubah.Enabled :=false;
  bbatal.Enabled :=true;
  bsimpan.Enabled :=true;
  bkeluar.Enabled :=false;
end;
```

Pada penggalan program diatas terdapat perintah untuk penjumlahan otomatis pada kode, artinya kode yang ditampilkan akan secara otomatis dihitung oleh bagian ini, untuk dijadikan kode baru, jumlah record yang ada akan ditambah 1 untuk digunakan sebagai kode record selanjutnya.

5.3 Proses Ubah Bahasa

Proses ubah bahasa juga demikian, akan sama pada data rak, yang membedakan hanya pada simpledataset yang digunakan. Selain itu pada form bahasa tidak dilengkapi dengan proses hapus, mengapa? Pada aplikasi ini diasumsikan bahwa setiap data yang telah tersimpan dianggap sudah benar, sehingga tidak perlu dihapus. Dan jika memang sangat dibutuhkan untuk menghilangkan/menghapus sebuah data, maka lakukan perubahan data saja, sehingga hal ini tidak berakibat pada perbedaan antara jumlah record dengan kode otomatisnya.

Untuk itu event handler pada Onclick button ubah adalah sebagai berikut:

```
procedure TForm11.bubahClick(Sender: TObject);
begin
groupbox1.Enabled :=true;
dbedit1.Enabled :=true;
dbedit2.Enabled :=true;
datamodule2.sdsbhs.Edit ;
btambah.Enabled :=false;
bubah.Enabled :=false;
bbatal.Enabled :=true;
bsimpan.Enabled :=true;
bkeluar.Enabled :=false;
end;
```

4.4 Proses Batal BahasaData Rak

Proses batal bahasa merupakan suatu proses yang digunakan untuk membatalkan suatu proses lain yang terjadi pada form bahasa. Proses Batal pada properties-nya diberi nama adalah bbatal.

Untuk itu event handler pada Onclick button batal, kemudian tuliskan programnya sebagai berikut:

```
procedure TForm11.bbatalClick(Sender: TObject);
begin
groupbox1.Enabled :=false;
dbedit1.Enabled :=false;
dbedit2.Enabled :=false;
```

```

datamodule2.sdsbhs.Cancel ;
btambah.Enabled :=true;
bubah.Enabled :=true;
bhapus.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
end;

```

4.5 Proses Simpan Bahasa

Proses simpan bahasa juga merupakan suatu proses yang mampu digunakan untuk klarifikasi tentang kebenaran data serta untuk menyimpan data yang sudah diklarifikasi oleh operator sistem. Proses Simpan pada properties-nya diberi nama adalah bsimpan.

Untuk itu event handler pada Onclick button simpan, kemudian tuliskan programnya sebagai berikut:

```

procedure TForm11.bsimpanClick(Sender: TObject);
begin
if      messagedlg('Apakah      Datanya      Sudah      Benar      dan      Lengkap
?',mtconfirmation,[mbyes,mbno],0)=mryes then
begin
dbedit1.Enabled :=false;
dbedit2.Enabled :=false;
datamodule2.sdsbhs.Post ;
datamodule2.sdsbhs.ApplyUpdates(0);
datamodule2.sdsbhs.Refresh ;
btambah.Enabled :=true;
bubah.Enabled :=true;
bhapus.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
groupbox1.Enabled :=false;
end;
end;

```

Listing Program secara lengkap menjadi:

```
unit bhs;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, Mask, DBCtrls, Grids, DBGrids;

type
  TForm1 = class(TForm)
    DBGrid1: TDBGrid;
    GroupBox1: TGroupBox;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    DBEdit1: TDBEdit;
    Label1: TLabel;
    GroupBox2: TGroupBox;
    btambah: TBitBtn;
    bbatal: TBitBtn;
    bubah: TBitBtn;
    bsimpan: TBitBtn;
    bkeluar: TBitBtn;
    procedure btambahClick(Sender: TObject);
    procedure bbatalClick(Sender: TObject);
    procedure bubahClick(Sender: TObject);
    procedure bsimpanClick(Sender: TObject);
    procedure bkeluarClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation
```

```

uses dm;

{$R *.dfm}

procedure TForm1.btambahClick(Sender: TObject);
begin
groupbox1.Enabled :=true;
dbedit1.enabled :=true;
dbedit2.Enabled :=true;

datamodule2.sdsbhs.Append ;
datamodule2.sdsbhs.FieldName('kode').AsString
:=inttostr(datamodule2.sdsbhs.recordcount+1);
btambah.Enabled :=false;
bubah.Enabled :=false;
bbatal.Enabled :=true;
bsimpan.Enabled :=true;
bkeluar.Enabled :=false;
end;

procedure TForm1.bbatalClick(Sender: TObject);
begin
groupbox1.Enabled :=false;
dbedit1.Enabled :=false;
dbedit2.Enabled :=false;
datamodule2.sdsbhs.Cancel ;
btambah.Enabled :=true;
bubah.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
end;

procedure TForm1.bubahClick(Sender: TObject);
begin
groupbox1.Enabled :=true;
dbedit1.Enabled :=true;
dbedit2.Enabled :=true;
datamodule2.sdsbhs.Edit ;
btambah.Enabled :=false;
bubah.Enabled :=false;
bbatal.Enabled :=true;

```



```

bsimpan.Enabled :=true;
bkeluar.Enabled :=false;
end;

procedure TForm11.bsimpanClick(Sender: TObject);
begin
if messagedlg('Apakah Datanya Sudah Benar dan Lengkap
?',mtconfirmation,[mbyes,mbno],0)=mryes then
begin
dbedit1.Enabled :=false;
dbedit2.Enabled :=false;
datamodule2.sdsbhs.Post ;
datamodule2.sdsbhs.ApplyUpdates(0);
datamodule2.sdsbhs.Refresh ;
btambah.Enabled :=true;
bubah.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
groupbox1.Enabled :=false;
end;
end;

procedure TForm11.bkeluarClick(Sender: TObject);
begin
close;
end;

procedure TForm11.FormCreate(Sender: TObject);
begin

groupbox1.Enabled :=false;

btambah.Enabled :=true;
bubah.Enabled :=true;
bbatal.Enabled :=false;
bsimpan.Enabled :=false;
bkeluar.Enabled :=true;
end;

end.

```

TUGAS ANDA, MELANJUTKAN APLIKASI PERPUSTAKAAN TERSEBUT, DENGAN MENAMBAHKAN FORMULIR-FORMULIR SEBAGAI BERIKUT:

DATA KLASIFIKASI KOLEKSI

Tambah Data Klasifikasi
Ubah Data Klasifikasi
Batal Data Klasifikasi
Simpan Data Klasifikasi

PENDATAAN ANGGOTA

Tambah Data Anggota
Ubah Data Anggota
Batal Data Anggota
Simpan Data Anggota

PENDATAAN KOLEKSI PERPUSTAKAAN

Tambah Data Koleksi
Ubah Data Koleksi
Batal Data Koleksi
Simpan Data Koleksi
Pencarian Berdasarkan Kode Koleksi
Lihat Data Berdasarkan Judul Koleksi

PENDATAAN INVENTARIS KOLEKSI PERPUSTAKAAN

Tambah Data Inventaris Koleksi
Ubah Data Inventaris Koleksi
Batal Data Inventaris Koleksi
Simpan Data Inventaris Koleksi
Hapus Data Inventaris Koleksi
Cetak Label Koleksi

PENDATAAN REGISTRASI ANGGOTA

Tambah Data Registrasi
Batal Data Registrasi
Simpan Data Registrasi
Hapus Data Registrasi

TRANSAKSI

Transaksi Peminjaman Koleksi
Transaksi Pengembalian Koleksi

CETAK LABEL RAK DAN DATA KOLEKSI

Cetak Label Rak
Cetak Data Koleksi

CETAK RATING KOLEKSI & SIRKULASI PERBULAN

Cetak Rating Koleksi
Cetak Sirkulasi Perbulan