

# MODUL PRAKTIKUM SISTEM BASIS DATA



**FAKULTAS BISNIS & TEKNOLOGI INFORMASI  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
2017**

## DAFTAR ISI

<b>Halaman Judul</b> .....	i
<b>Daftar Isi</b> .....	ii
<b>MODUL I</b> MENGENAL DATABASE PADA MS ACCESS .....	1
<b>MODUL II</b> MENGENAL TIPE DATA LOOKUP & QUERY .....	13
<b>MODUL III</b> MEKANISME PENGAKSESAN DATABASE ACCESS DENGAN DELPHI & PENCARIAN DATA .....	21
<b>MODUL IV</b> NAVIGASI RECORD .....	28
<b>MODUL V</b> MEMFORMAT DAN MEMVALIDASI DATA .....	34
<b>MODUL VI</b> DATA MODUL .....	37
<b>MODUL VII</b> DCL (Data Control Language) .....	40
<b>MODUL VIII</b> Aplikasi Sederhana pengakses MySQL .....	45

# **MODUL PRAKTIKUM APLIKASI BASISDATA**

## **Deskripsi Materi**

Mata Kuliah Praktikum Pemrograman Database membahas tentang Konsep Dasar Pembuatan Database yang dimulai dari Pengenalan Database dalam MS Access, dilanjutkan dengan pengenalan data dengan tipe lookup dan pembuatan query, mekanisme pengaksesan database dengan Delphi, navigasi record dalam Delphi, memformat dan validasi data, penggunaan data modul, dan penggunaan DBMS MySQL.

## **Prasyarat**

Untuk mengikuti mata kuliah ini mahasiswa harus lulus mata kuliah prasyarat Pemrograman Dasar dan Perancangan Basis Data. Agar mata kuliah ini dapat berhasil dengan baik, mahasiswa disarankan untuk sering berlatih dan menganalisa alur suatu permasalahan pemrograman dengan algoritma kemudian mengimplementasikan ke dalam bahasa pemrograman.

## **Petunjuk Pemakaian Modul**

Untuk dapat memahami praktikum ini dengan baik mahasiswa disarankan membaca secara berurutan, karena modul ini disusun secara terstruktur diawali dengan tujuan instruksional dan kompetensi yang dihasilkan, pengantar teori, contoh kasus dan solusinya serta diakhiri dengan tugas mandiri yang berfungsi melatih mahasiswa untuk mencoba menyelesaikan suatu permasalahan sesuai dengan metode penyelesaian masing-masing.

## **Standar Kompetensi**

Setelah mahasiswa mengambil matakuliah praktikum pemrograman database dengan menggunakan bantuan modul ini diharapkan mahasiswa memiliki kemampuan menganalisa dan mendefinisikan suatu masalah, mencari kemungkinan solusi-solusi yang dapat digunakan dan memilih solusi yang terbaik, menyusun algoritma program kemudian menyusunnya ke dalam bahasa pemrograman dan juga dengan menggunakan bahasa DBMS yang dikuasai.

# MODUL I

## MENGENAL DATABASE PADA MS ACCESS

### A. Kompetensi dan Indikator

Kompetensi Dasar: Memahami tentang Database, Tabel, Record dan Field

Indikator : - Mampu membuat database & tabel dengan menggunakan MS Access

- Mampu menentukan type dan ukuran sebuah field
- Mampu membuat relasi antar tabel

### B. Dasar Teori

#### 1. Pengenalan Database

Definisi *database* sangatlah bervariasi. C.J. Date menyatakan bahwa *database* dapat dianggap sebagai tempat untuk sekumpulan berkas data terkomputerisasi.

implementasi *database* dapat dibagi menjadi 2 :

1. sebuah database disimpan pada sebuah direktori dan **terdiri atas sejumlah berkas table.**

2. sebuah database dinyatakan dengan sebuah berkas.

Model pertama pada model pengolahan database keluarga Dbase. Pada visual FoxPro misalnya, kumpulan berkas berekstensi. DBF. Model kedua bisa di jumpai pada Microsoft Access. **Pada Access, sebuah *database* akan disimpan pada dengan ekstensi . MDB.**

Dalam terminologi *database* yang bersifat relasional, suatu database tersusun atas sejumlah table.

#### 2. Database MS Access.

*Database* atau basis data adalah kumpulan data yang berhubungan dengan suatu obyek, topic atau tujuan khusus tertentu.

*Database* pada Access terdiri atas beberapa **tabel, query, form, report, page, makro, dan modul** yang semuanya saling berhubungan atau saling terkait.

❖ **TABLES**

Kumpulan data yang merupakan komponen utama dari sebuah *database*.

❖ **QUERIES**

Untuk mencari dan menampilkan data yang memenuhi syarat tertentu.

❖ **FORMS**

Dipergunakan untuk menampilkan data, mengisi data, dan mengubah data yang ada pada sebuah tabel.

❖ **REPORTS**

Untuk menampilkan laporan hasil analisis data.

❖ **PAGES**

Untuk membuat halaman web (page) berupa data access page yang dapat ditempatkan diserver system jaringan **Intranet** atau **Internet**.

❖ **MACROS**

Untuk mengotomasisasi perintah-perintah yang sering digunakan.

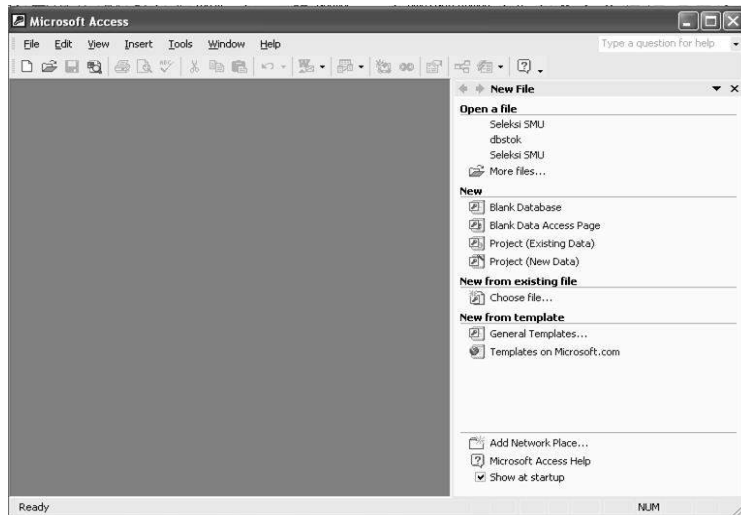
❖ **MODULES**

Digunakan untuk perancangan berbagai modul aplikasi pengolahan *database*.

### **3. Membuat Database Baru dengan MS Access**

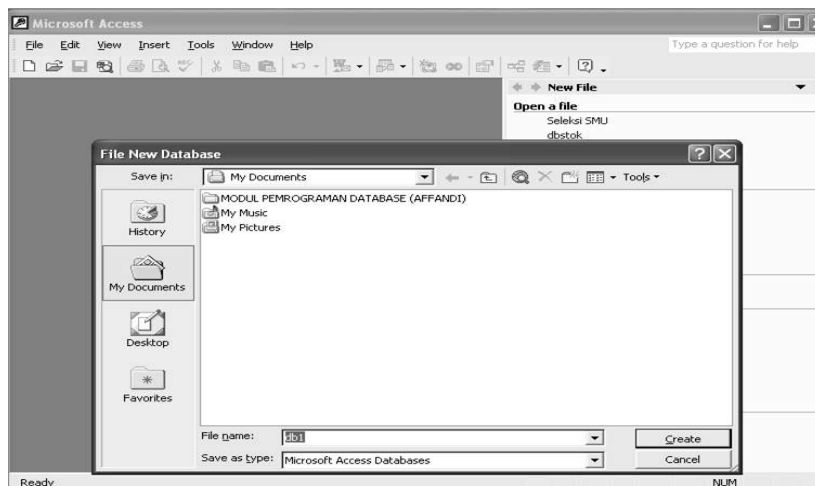
Untuk membuat *database* baru yang masih kosong ikuti langkah berikut ini :

- 1.** Jalankan atau aktifkan program aplikasi Access. Jendela kerja Access akan ditampilkan yang dilengkapi dengan **jendela Task Pane-New File** disebelah kanannya.



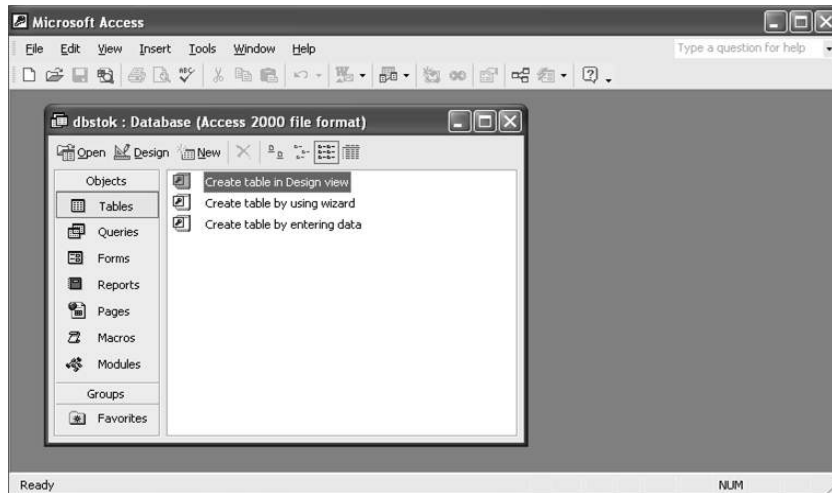
**Gambar 1.1.** Jendela kerja Access.

2. Pilih dan **klik Blank Database**.



**Gambar 1.2.** Jendela database baru yang masih kosong.

3. Pada tombol pilihan **Save In**, pilih **drive** dan **folder** tempat akan menyimpan file.
4. Pada kotak isian **File Name**, ketikkan **nama file** yang diinginkan.
5. Klik tombol **perintah Create**. Tunggu sampai *database* baru yang masih kosong terbentuk.

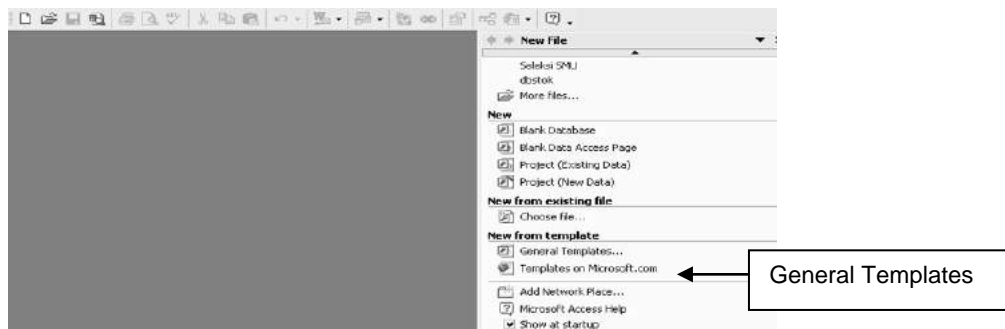


**Gambar 1.3.** Jendela Kerja Database Baru yang Masih Kosong

#### 4. Membuat Database Baru dengan Fasilitas Template Database Wizard.

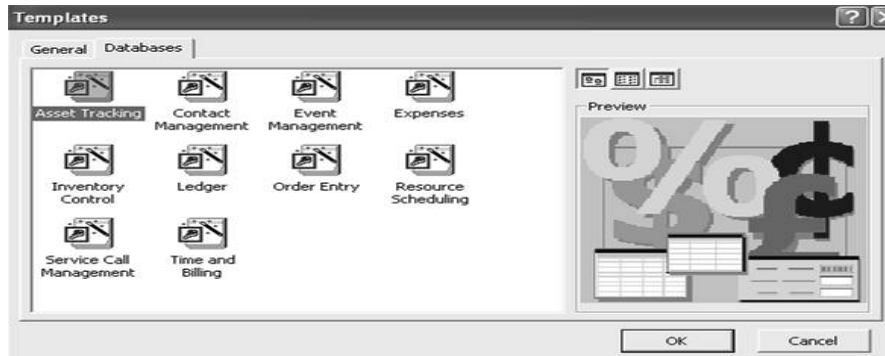
Ikuti langkah berikut ini :

1. Jalankan atau **aktifkan program aplikasi MS Access.**
2. Pada bagian New from templet yang ada di jendela **Task Pane – New File**, pilih dan klik **General Templates**. Kotak dialog templets akan di tampilkan.



**Gambar 1.4.** General Templates.

3. Pada kotak dialog tersebut, **klik tab Databases**. Lihat Gambar



**Gambar 1.5.** Kotak Dialog Templates-Tab Databases.

4. Pada kotak dialog tersebut, pilih dan klik icon **file database** yang menurut anda paling sesuai, kemudian **OK**.
5. Pada pilihan **Save In**, pilih **drive** dan **folder** tempat menyimpan file.
6. Pada kotak isian **File Name**, ketikkan **nama File**.
7. Klik tombol Perintah **Create**. Kotak dialog **Database Wizard** yang pertama di tampilkan.



**Gambar 1.6.** Kotak dialog database Wizard

8. Disini akan ditampilkan berbagai keterangan, bila sudah di baca klik tombol perintah **Next**.
9. Pada kotak dialog tersebut lakukan pemilihan **nama Field** yang akan digunakan., kemudian **klik Next**.
10. Pilih gaya tampilan, **Klik Next**.
11. Klik gaya tampilan Laporan, **pilih Next**.



12. Ketikkan **judul database**.
13. Pada kotak dialog tersebut, beri tanda atau klik kotak **cek Yes, Star the database**, bila database yang terbentuk langsung ingin ditampilkan dilayar.
14. Tunggu hingga database **yang dipilih terbentuk**.

## 5. Mulai Menciptakan Tabel.

**Tabel** adalah kumpulan data yang berhubungan dengan topik tertentu/kumpulan dari beberapa entity yang saling berhubungan, misalnya daftar penjualan, daftar pembelian.

Sebagai contoh gambar dibawah, yang memperlihatkan suatu database yang diberi nama **DBstok**. Database ini mengandung tiga buah tabel dengan masing-masing bernama **BARANG, PEMBELIAN, dan PENJUALAN**.

Database: DBstok

TABEL BARANG.

KODE_BARANG	NAMA_BARANG	SATUAN	JUMLAH
A1A01	Monitor Samsung 14 inc	Unit	7
A2B01	Monitor Yundai 14 inc	Unit	2

TABEL PEMBELIAN.

KODE_BARANG	HARGA_JUAL_BARANG	JUMLAH
A1A01	Rp 500.000	9
A1A02	Rp 700.000	5
A2B01	Rp 600.000	6

TABEL PENJUALAN.

KODE_BARANG	HARGA_BELI_BARANG	JUMLAH
A1A01	Rp. 600.000	2

**Gambar 1.7.** Gambaran database, tabel, dan isi tabel.

Suatu tabel memiliki sejumlah field atau kolom. Sebagai contoh, tabel barang memiliki 4 buah field :

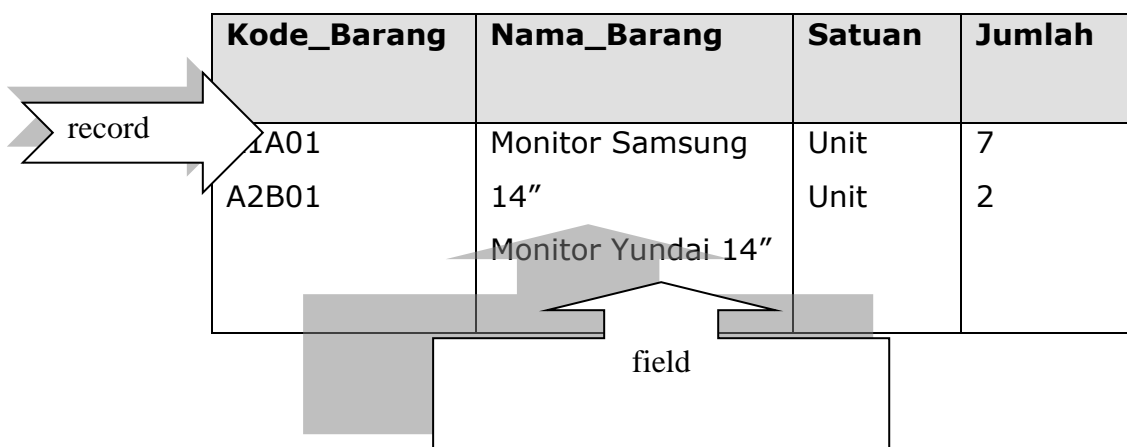
- KODE\_ BARANG.
- NAMA\_ BARANG.
- SATUAN.
- JUMLAH

Tabel pembelian dan penjualan memiliki tiga buah field.

Setiap baris pada sebuah tabel dikenal dengan istilah sebuah *record*.

Sebuah **record** digunakan untuk menyatakan sebuah data.

#### TABEL **BARANG**.



Kode_Barang	Nama_Barang	Satuan	Jumlah
A1A01	Monitor Samsung	Unit	7
A2B01	14" Monitor Yundai 14"	Unit	2

The diagram illustrates a table with four columns: **Kode\_Barang**, **Nama\_Barang**, **Satuan**, and **Jumlah**. A large arrow labeled 'record' points to the first row of data. A smaller arrow labeled 'field' points to the 'Nama\_Barang' column. The second row of data contains '14"' and 'Monitor Yundai 14"' in the 'Nama\_Barang' column, indicating a multi-line entry.

**Gambar 1.8.** Tabel, Record, dan Field.

- **Field**, tempat dimana data atau informasi dalam kelompok yang sama atau sejenis di masukkan. Field tersimpan secara vertical pada tabel.
- **Record**, merupakan data lengkap dalam jumlah tunggal yang biasanya tersimpan dalam bentuk baris secara horizontal pada tabel.

Setiap tabel memiliki kunci. **Kunci primer** digunakan untuk mengidentifikasi sebuah record. Kunci ini bersifat **unik**, yang berarti bahwa dalam sebuah tabel tidak akan ada nilai kunci primer yang sama. Kunci yang tersusun dari sebuah *field* biasa disebut kunci sederhana, sedangkan yang tersusun atas beberapa *field* biasa disebut **kunci komposit**.

## 6. Membuat tabel dengan Fasilitas Create Table in Design View

Sebelum membuat sebuah tabel dengan beberapa data yang ada, terlebih dahulu tentukan struktur tabel yang nantinya menjelaskan jenis data untuk setiap field dan ukuran data yang bisa di tampung. Selain itu perlu menentukan kunci primer yang digunakan.

Ikuti langkah berikut :

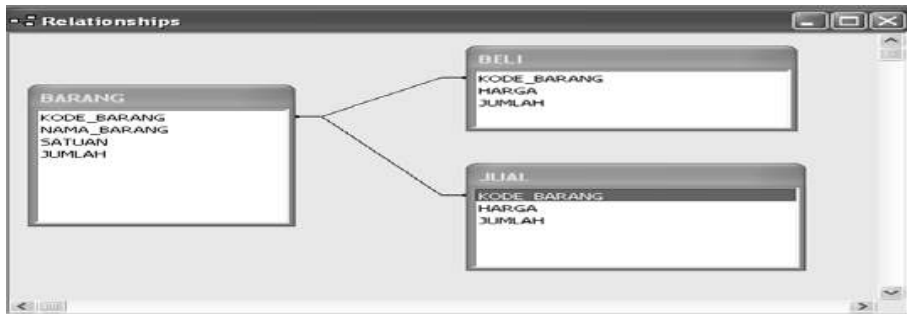
1. Buka file database yang diinginkan, kemudian pada jendela kerja database , **klik Tables** yang ada dibawah Objects .
2. Pada jendela kerja database tersebut, klik dua kali **Create table in Design view**.
3. Pada kotak dialog tersebut, lakukan **pendefinisian struktur** tabel.
4. Isi **field property** yang ada dibawahnya.
5. Setelah selesai, simpan hasil pendefinisian tersebut dengan cara memilih dan **mengklik menu File, Save ( Ctrl+S )** atau klik tombol **toolbar Save**.
6. Pada kotak isian **Table Name**, ketikkan nama untuk table tersebut.
7. Tutup **jendela pendefinisian** tabel tersebut.

## 7. Relationships Antar Tabel.

Jenis relasi antar tabel :

- **One-to-many** relationship (relasi satu ke banyak ).
- **Many-to-many** relationship (relasi banyak ke banyak).
- **One-to-one** relationship ( relasi satu ke satu ).

Sebagai contoh lihat gambar 9, pada database **dbstok** terdapat tiga buah tabel diantaranya: barang, pembelian, dan penjualan.



**Gambar 1.9.** Relasi antar tabel

Hasil dari relasi antar tabel dapat dilihat di gambar berikut.

KODE_BARANG	NAMA_BARANG	SATUAN	BANYAK
- A1A01	MONITOR SAMSUNG 14 INC	UNIT	
	HARGA	JUMLAH	
	\$400,000.00	9	
*	\$0.00		
- A2B01	MONITOR YUNDAI 14 INC	UNIT	
	HARGA	JUMLAH	
	\$300,000.00	3	
*	\$0.00		
- A2B02	MONITOR YUNDAI 15 INC	UNIT	
	HARGA	JUMLAH	
	\$600,000.00	2	
*	\$0.00		

**Gambar 1.10.** hasil relasi antara beberapa tabel

### C. Praktek

#### 1. Struktur Tabel

Buat struktur tabel dibawah ini dengan ketentuan :

Database name : dbstok.

Tabel name : 1. tabel barang. 2.tabel pembelian. 3. tabel penjualan.

Struktur tabel **BARANG**.

FIELD NAME	TYPE	FIELD SIZE	PRIMARY KEY
KODE_BARANG	TEXT	6	*
NAMA_BARANG	TEXT	25	-
SATUAN	TEXT	12	-
JUMLAH	NUMBER	Long Integer	-

Struktur tabel **PEMBELIAN**.

FIELD NAME	TYPE	FIELD SIZE	PRIMARY KEY
KODE_BARANG	TEXT	6	*
HARGA_BELI	Currency	Currency	-
JUMLAH	NUMBER	Long Integer	-

Struktur tabel **PENJUALAN**.

FIELD NAME	TYPE	FIELD SIZE	PRIMARY KEY
KODE_BARANG	TEXT	6	*
HARGA_JUAL	Currency	Currency	-
JUMLAH	NUMBER	Long Integer	-

2. Memasukan Data

TABEL **BARANG**

KODE_BARANG	NAMA_BARANG	SATUAN	JUMLAH
A1A01	Monitor Samsung	Unit	7
A2B01	14 inc Monitor Yundai 14 inc	Unit	2

TABEL **PEMBELIAN**.

KODE_BARANG	HARGA_JUAL_BARANG	JUMLAH
A1A01	Rp 500.000	9
A1A02	Rp 700.000	5
A2B01	Rp 600.000	6

TABEL **PENJUALAN**.

KODE_BARANG	HARGA_BELI_BARANG	JUMLAH
A1A01	Rp. 600.000	2

3. Tambahkan empat data lagi untuk masing-masing tabel. Dan rubah nama field dalam tabel pembelian dan tabel penjualan ( jumlah menjadi jumlah pembelian & jumlah penjualan ).

TABEL **PEMBELIAN.**

<b>KODE_BARANG</b>	<b>HARGA_JUAL_BARANG</b>	<b>JUMLAH_PEMBELIAN</b>
A1A01	Rp 500.000	9
A1A02	Rp 700.000	5
A2B01	Rp 600.000	6

TABEL **PENJUALAN.**

<b>KODE_BARANG</b>	<b>HARGA_BELI_BARANG</b>	<b>JUMLAH_PENJUALAN</b>
A1A01	Rp. 600.000	2

4. Buat relasi antar tabel tersebut



## MODUL II.

### MENGENAL TIPE DATA LOOKUP & QUERY

#### A. Kompetensi dan Indikator

Kompetensi Dasar: Memahami tentang Tipe Data Lookup & Query

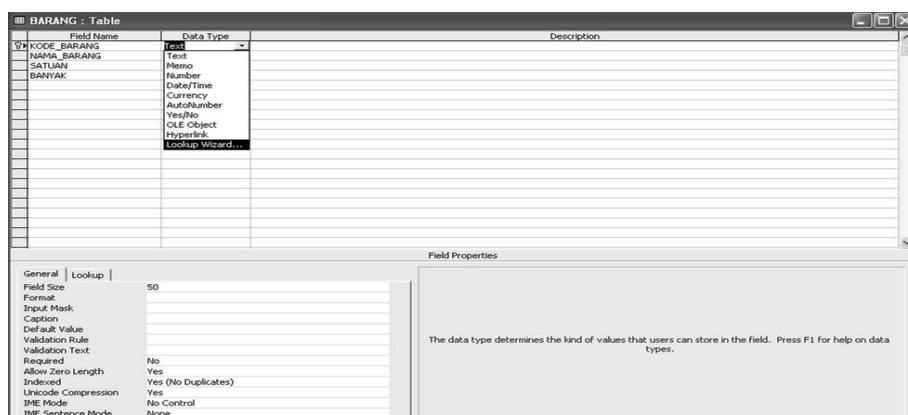
Indikator : - Mampu membuat tabel dengan tipe lookup  
- Mampu membuat query 1 tabel dan antar beberapa tabel

#### B. Dasar Teori

##### 1. Membuat Field Untuk Data Berjenis Lookup.

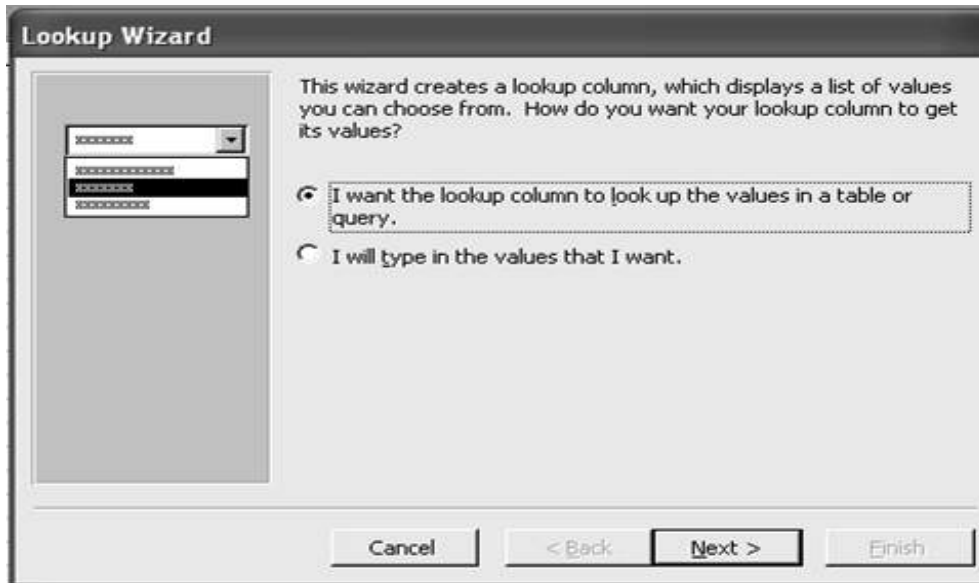
Untuk membuat field yang berjenis lookup wizard ikuti langkah berikut ini ;

- a. **Buka database** yang di inginkan, kemudian pada jendela kerja database, **klik Tables** yang ada di bawah **Object** atau pilih dan **klik menu View, database Object, Tables.**
- b. Apabila ingin menambahkan field baru yang ingin di jadikan data lookup wizard **klik tombol toolbar Design.**
- c. Lakukan perubahan sesuai dengan keinginan semisal, field kode\_barang akan di jadikan data **bertipe Lookup Wizard**, setelah selesai memilih kotak dialog Lookup wizard yang pertama akan di tampilkan.



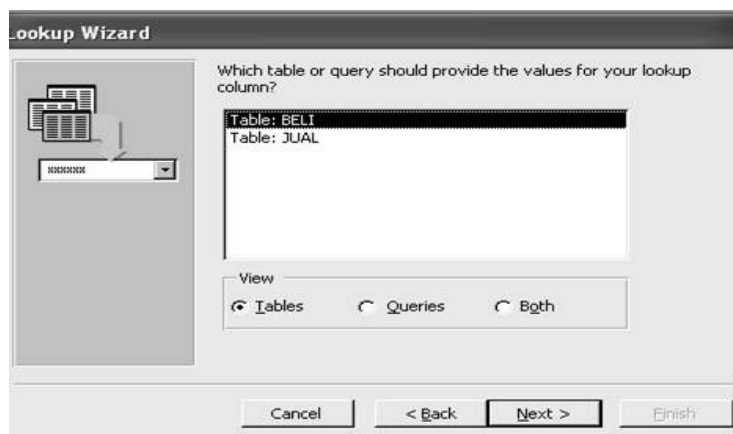
**Gambar 2.1.** Jendela Design Struktur Tabel.





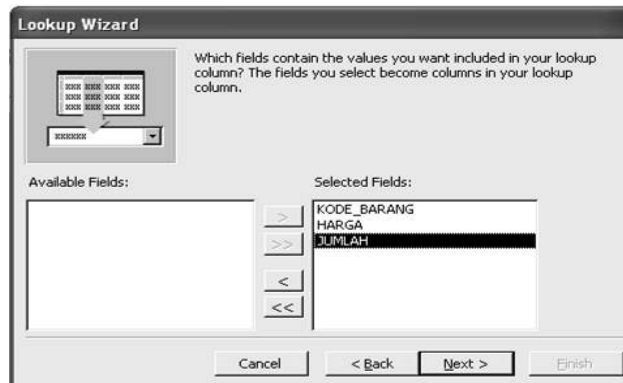
**Gambar 2.2.** Kotak Informasi Lookup Wizard.

- d. Pada kotak dialog tersebut, beri tanda pada salah satu tombol .
- **I want the lookup column to lookup up the values in a table or query**, bila ingin mengambil data tabel atau query yang sudah ada pada database.
  - **I will type in the values that I want**, bila ingin mengetikkan datanya.
- e. Jika sudah **klik perintah Next**.

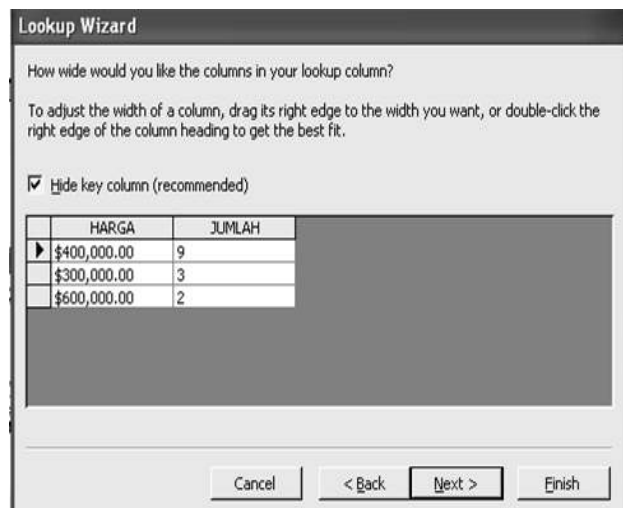


- f. Pada kotak **Available Field**, pilih dan klik nama Filed yang ingin di tampilkan, kemudian **klik tombol >** . dengan langkah ini nama

field yang di pilih akan di pindah kekotak daftar pilihan Selected Fields, kemudian **klik tombol Next**.



- g. Selanjutnya lakukan pengaturan pada **lebar kolom** agar sesuai dengan ke inginan, dilanjutkan dengan mengklik **tombol Next**.



- h. Pada kotak dialog terakhir ketikkan label untuk field atau nama kolom yang diinginkan. Kemudian **klik tombol perintah Finish**.

## 2. Mengenal Query.

Ada beberapa macam query yang terdapat didalam MS Access, di antaranya :

- Select query**, merupakan pasilitas untuk menampilkan suatu tabel dengan kreteria tertentu.
- Parameter query**, adalah query yang jika di jalankan menampilkan kotak dialog yang menanyakan informasi yang akan

di gunakan. Parameter query juga berguna sebagai basis bagi form dan report.

- c. **Crosstab Query**, menampilkan nilai-nilai yang telah diolah.
- d. **Action Query**, ada 4 macam action query yaitu :
  1. *Delete Query*, yang digunakan untuk menghapus sekumpulan record.
  2. *Update Query*, digunakan untuk membuat perubahan umum terhadap sekumpulan record.
  3. *Append Query*, di gunakan untuk menambahkan sekelompok record dari satu atau lebih tabel dengan kreteria tertentu.
  4. *Make-Table Query*, untuk membuat tabel baru dari seluruh atau sebagian data dalam satu atau lebih tabel.
  
- e. **SQL Query**, adalah query yang dapat dibuat dengan pernyataan query. SQL Query terdiri atas :
  1. *Union*, digunakan untuk menggabungkan *field* (kolom) dari satu atau lebih tabel atau query kedalam satu atau lebih tabel.
  2. *Pass-Trough*, jenis perintah ini mengirim perintah secara langsung ke database ODBC.
  3. *Data-Definition*, untuk membuat object database.

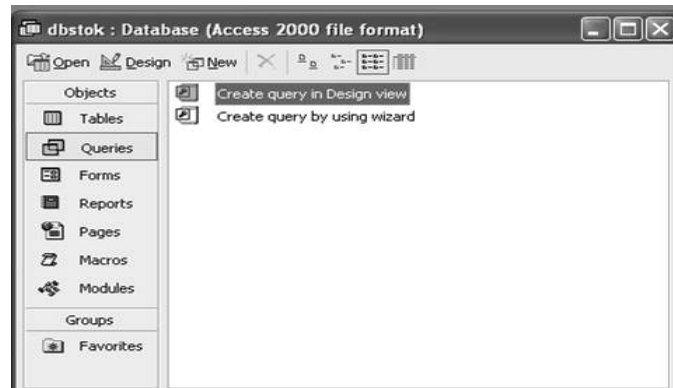
### **3. Membuat Dan Menggunakan Query.**

Query dapat di buat dengan fasilitas **Create query in design view** atau fasilitas **Create query by using wizard**.

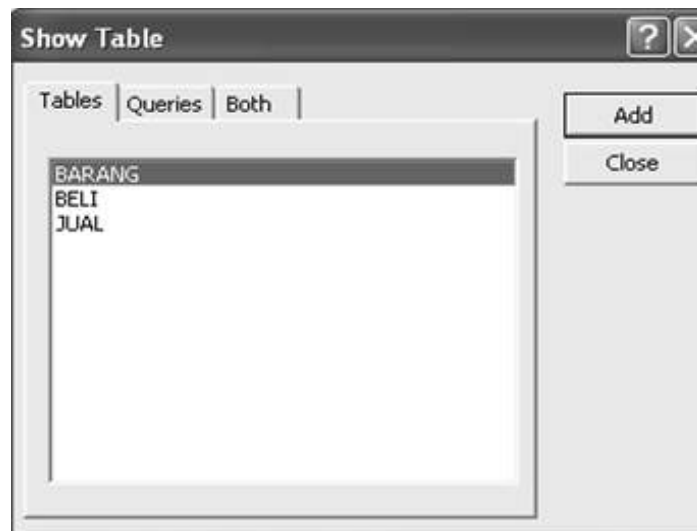
Dengan menggunakan fasilitas Create query in design view.

Untuk membuat, ikuti langkah berikut :

1. Buka file *database* yang diinginkan, kemudian pada jendela kerja database klik Queries yang ada di bawah Objects atau klik **menu view, database Objects, Queries**.



2. Klik dua kali pada bagian **Create query in design view**. Kotak dialog show table akan di tampilkan.



3. Pada kotak dialog **show table**, pilih dan klik tabel atau query lain yang ingin di libatkan, kemudian klik tombol perintah **Add**.
4. Untuk menutup kotak dialog Show table, pilih dan klik tombol perintah **Close**.



5. Pada jendela kerja **select query** tersebut, tempatkan field-field pada kolom yang sudah disediakan.
- Pada baris field pilih dan **klik field** yang ingin di tampilkan. Dengan menggunakan langkah yang sama , lanjutkan mengisi kolom –kolom berikutnya.
  - Pada kotak daftar pilihan field yang ada di bagian atas jendela kerja select query , pilih dan **geser nama field** yang ingin di tampilkan.
  - Lanjutkan langkah tersebut untuk field-field yang ingin di tampilkan.
  - Untuk melihat hasil query yang baru dibentuk, pilih dan klik **menu Query, Run** atau klik tombol **toolbar Run**.

#### 4. Memodifikasi Rancangan Query

Untuk menyesuaikan dengan kebutuhan, rancangan query yang sudah di buat dapat di modifikasi dengan berbagai cara.

#### 5. Menghapus sebuah field.

Untuk menghapus sebuah field tertentu yang tidak di butuhkan dari query, ikuti langkah berikut :

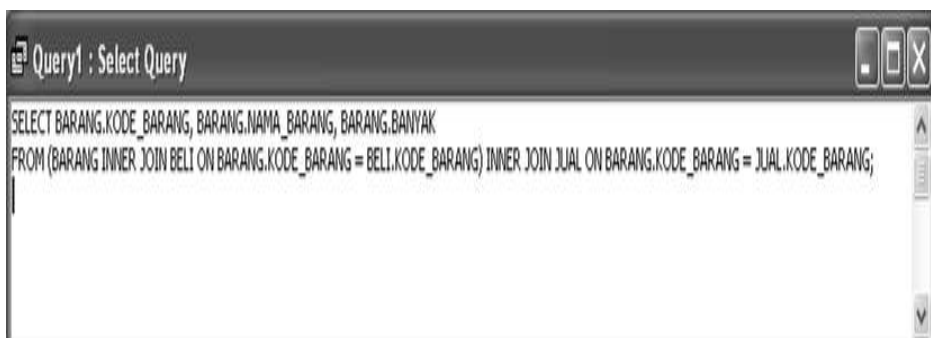
1. Buka database yang diinginkan, kemudian pada jendela kerja database, **klik Queries** yang ada di bawah Objects atau pilih dank ilk menu **View, Database Objects, Queries**.

2. Pilih klik query yang ingin di modifikasi rancangannya, kemudian **klik tombol toolbar Design**. Jendela kerja perancangan query akan di tampilkan.
3. Pilih dan **klik field** yang akan di hapus, kemudian pilih dan klik menu **Edit, Delete Columns**.
4. Setelah selesai memodifikasi jangan lupa **disimpan**.

## 6. Menampilkan Query dengan Tampilan SQL View.

Untuk menampilkan query dengan tampilan sql ikuti langkah berikut :

1. Pilih dan klik query yang ingin di modifikasi rancangannya, kemudian klik tombol **toolbar Design**. Jendela kerja perancangan query akan di tampilkan.
2. Pada jendela kerja perancangan query tersebut , pilih dan klik menu **View, SQL View**. Jendela kerja dengan tampilan SQL View akan di tampilkan.



Untuk kembali ke jendela kerja semula, pilih dan klik menu **View, Design View**.

## C. Praktek

Buat query dengan menggunakan Create Query By Using wizard.



# MODUL III

## **MEKANISME PENGAKSESAN DATABASE ACCESS DENGAN DELPHI & PENCARIAN DATA**

### **A. Kompetensi dan Indikator**

Kompetensi Dasar: Memahami tentang Database Access dengan Delphi

Indikator : - Mampu membuat koneksi antara database Access dengan Delphi

- Mampu membuat aplikasi untuk pencarian data

### **B. Dasar Teori**

#### **1. Mekanisme Pengaksesan Database**

Pada dasarnya database dapat dibedakan menjadi dua macam :

##### **1. Local database.**

**Berada** pada drive local atau sebuah jaringan local ( Local Area Network ). Yang termasuk pada local database di antaranya adalah *Paradox, dBase, FoxPro, dan Access.*

##### **2. Remote *database servers***

Dapat di golongan pada *two-tiered applications* atau *multi-tiered application*, karena aplikasi terletak pada system (tiers) yang berbeda. Yang termasuk di antaranya adalah *InterBase, Oracle, Sybase, Microsof SQL Serve, dan DB2.*

**Delphi** hanyalah salah satu perangkat pengembangan aplikasi yang sangat terkenal di lingkungan windows, yang dapat digunakan untuk membuat aplikasi yang melibatkan *database*. Delphi menggunakan bahasa pembangunan berupa **Object Pascal**. Apa yang bisa dikembangkan dengan Delphi? Tentu saja banyak hal yang bisa diimplementasikan dengan menggunakan perangkat lunak ini. Tidak hanya *database* , masalah-masalah seperti komputasi numeric, grafika



computer, dan multimedia dapat diselesaikan dengan mudah menggunakan Delphi.

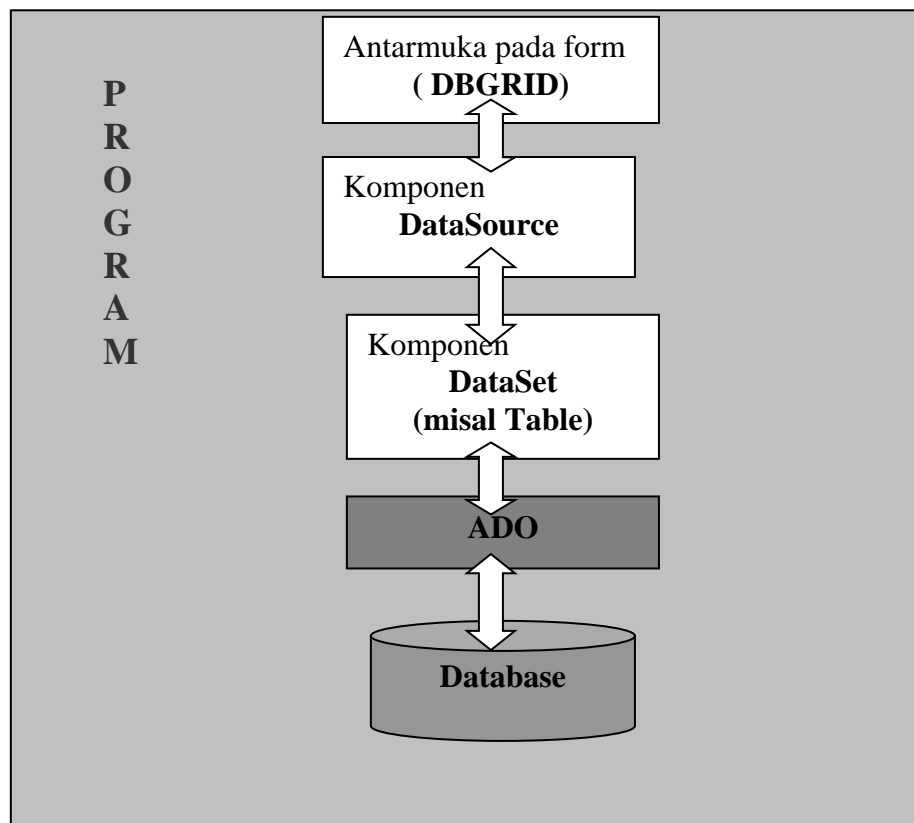
Dephi menyatakan banyak komponen untuk membuat aplikasi database.

Komponen-komponen tersebut diantaranya :

- a. Page BDE.
- b. Page ADO.
- c. Page Interbase.
- d. Page dbExpres.
- e. Page DataAccess

Sebenarnya terdapat beberapa mekanisme untuk mengakses *database*.

Salah satu diantaranya adalah melalui **ADO**, yang dapat digunakan untuk mengakses **database MS Access**.

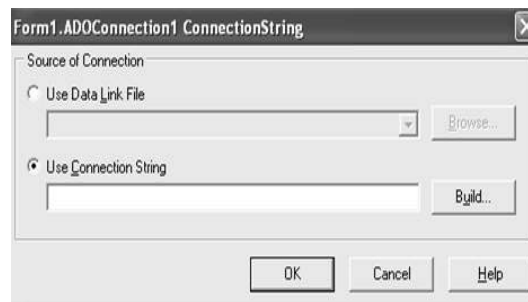


**Gambar 3.1.** Mekanisme Pengaksesan database melalui **ADO**

## 2. Memahami Komponen-Komponen Database dalam Delphi

Untuk lebih memahami penggunaan komponen TADOConnection dan TADOTable coba **latihan** berikut :

- Buat sebuah objek baru dan tambahkan **TADOConnection** pada form. **Klik tombol elipsis** yang terdapat pada property **ConnectionString**.
- Pada kotak dialog **Data Link Properties**, pilih **Use Connection String** dan klik pada tombol **Build**.



- Pada tab **Provider** terdapat beberapa provider . pilih tab **Microsoft Jet 4.0 OLE DB Provider** klik **Next**.



- Pada **Tab Connection**, klik tombol **ellipsis button** ( tombol yang mengarahkan ketempat penyimpanan *database* access ), dan klik tombol **Test Connection**.



- e. Jika sudah terconnecti klik **tombol OK**.
- f. Tambahkan komponen **TADOTable**. Ubah **property Connection** menjadi **ADOConnection1**. ubah property **TableName** menjadi **table** yang diinginkan, dan aktifkan tabel dengan mengubah property **Active** menjadi **True**.
- g. Tambahkan komponen **TDataSource** dan ubah properti **DataSet** menjadi **ADOTable1**.
- h. Tambahkan komponen **TDBGrid** dan **TDBNavigator** dan ubah property **DataSource** dari dua komponen tersebut menjadi **DataSource1**.

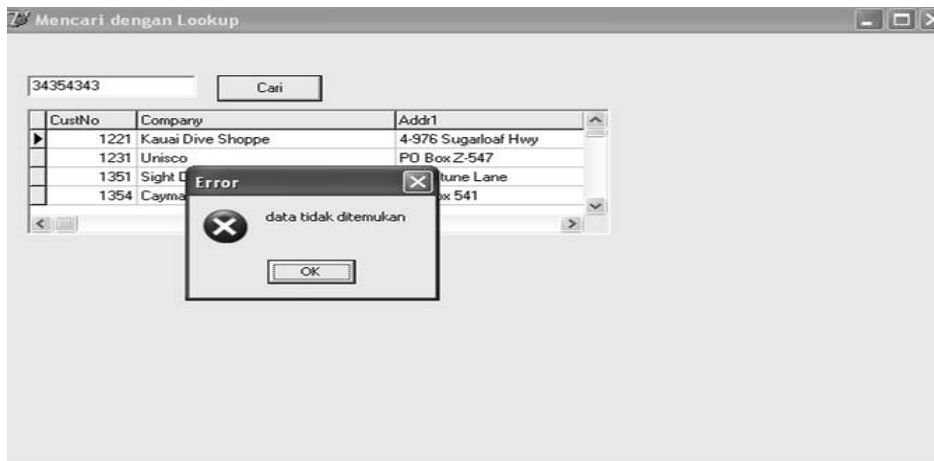
### 3. Pencarian Data

Mencari data sangat di perlukan dalam pembuatan database. Maka di dalam penggunaan **ADO dikenal 3 model pencarian** data antara lain menggunakan **method Locate, Lookup, dan seek**.

#### C. Praktek

Buat aplikasi untuk pencarian dengan menggunakan Lookup. lookup dipakai untuk mengambil nilai dari field yang sesuai dengan kreteria pencarian, contoh, seperti dibawah ini

1. Buat sebuah proyek baru dalam Delphi dan tambahkan pada form, komponen **TADOTable**, **TDataSource**, **TDBGrid**, **Tedit**, dan **TButton**.
2. Ubah beberapa property dari masing-masing komponen. Database yang di gunakan yaitu **dbdemos.mdb**. sedangkan tabelnya adalah **customer**.
3. Pada aplikasi ini jika user mengklik **tombol cari** maka akan dilakukan **proses pencarian**. **Ditampilkan kotak dialog** . jika pencarian gagal, akan muncul pesan kesalahan.



KOMPONEN	PROPERTY	NILAI
TADOTable	ConnectionString	Provider=Microsoft.jet.OLEDB.4.0 ;Data Source= C:\Program Files\Common Files\Borland Shared\Data\dbdemos.mdb
	Tablename	Customer
	Active	True
TdataSource	DataSet	ADOTable1
TDBGrid	DataSource	DataSource1
TEdit	Name	EditCari
TButton	Caption	Cari

**Listing program sebagai berikut :**

```

procedure TForm1.Button1Click(Sender: TObject);
var
hasilpencarian:variant;
begin
if Trim(editcari.Text)<>"then
begin
hasilpencarian :=ADOTable1.lookup('custno',editcari.Text,'company');
if not varIsNull (hasilPencarian) then
ShowMessage(varToStr(hasilpencarian))
else
MessageDlg('data tidak ditemukan',mterror, [mbOK], 0);
end
// editcari masih kosong maka letakkan fokus pada editcari
else
editcari.setfocus;
end;

```

#### **D. Latihan.**

1. Buat Program pencarian dengan Locate untuk database DBstok.mdb
2. Aplikasikan program pencarian dengan menggunakan method Locate dan seek



## MODUL IV

### NAVIGASI RECORD


#### A. Kompetensi dan Indikator

Kompetensi Dasar: Memahami tentang Navigasi untuk Record

Indikator : - Mampu membuat navigasi dengan dbnavigator  
- Mampu membuat sendiri tombol navigator

#### B. Dasar Teori

**Navigasi Record** digunakan untuk mengelola record dalam tabel, diantaranya berpindah kerecord paling awal, berpindah kerecord terakhir, maju persatu record, mundur persatu record.

Dalam Delphi sudah tersedia komponen untuk navigator record seperti gambar ini . Selain komponen navigator yang sudah disediakan oleh Delphi, programmer juga dapat membuat komponen navigasi sendiri, oleh karena itu dalam praktek disampaikan tentang tentang cara pembuatan tombol navigasi.

#### B. Praktek

Siapkan 2 form. Form pertama digunakan untuk tombol navigasi seperti praktek 1 dengan menggunakan database Paradox dan komponen TTable. dan form ke 2 digunakan untuk tombol simpan, batal, baru, tutup dengan database Access dan komponen ADO.

##### 1. Tombol Navigasi

- a. Buat form baru
- b. Tambahkan komponen **TTable**, ubah property **TDataBase** name menjadi **DBDemos**, dan property **TableName** menjadi **Country.db**, aktifkan tabel dengan mengubah property **Active** menjadi **True**.
- c. Tambahkan komponen **TDataSource**, ubah property **DataSet** menjadi **Table1**.

- d. Tambahkan komponen **TDBGrid**, ubah property **DataSource** menjadi **DataSource1**.



- e. Tambahkan lima buah tombol **TButton** dan **satu buah TEdit**, kemudian susun tampilannya.
- f. Lengkapi kode program sebagai berikut :

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    Table1.First;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
    Table1.Prior;
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
    Table1.Next;
end;
procedure TForm1.Button4Click(Sender: TObject);
begin
    Table1.Last;
end;
procedure TForm1.Button5Click(Sender: TObject);
begin
    Table1.MoveBy(StrToInt(Edit1.Text));
end;
```



## 2. Simpan, Batal, Baru, Tutup.

Mula-mula buatlah form baru dengan tampilan seperti berikut :

KODE_BARANG	NAMA_BARANG	SATUAN	BANYAK
A1A01	MONITOR SAMSUNG 14 IN	UNIT	7
A2B01	MONITOR YUNDAI 14 INC	UNIT	2
A2B02	MONITOR YUNDAI 15 INC	UNIT	4
a1b01	ioriowiwo	unit	5

Pedoman dalam pembuatan form dapat dilihat dibawah ini :

KOMPONEN	PROPERTY	NILAI
Form	Caption	Form Data Barang
Group Box	Caption	Input Data Barang
Label1	Caption	Kode Barang
Label2	Caption	Nama Barang
Label3	Caption	Satuan
Label4	Caption	Banyak
Edit1	Name	Edit1
Edit2	Name	Edit2
Edit3	Name	Edit3
Combobox1	Name	Combobox1
Speedbutton1	Caption	Simpan
Speedbutton2	Caption	Batal
Speedbutton3	Caption	Baru
Speedbutton4	Caption	Tutup
TADOTable	ConnectionString	Isi alamat database yang

		akan di hubunngkan. Seperti contoh pertemuan sebelumnya.
	Tablename	Tentukan Nama tabelnya
	Active	True
TdataSource	DataSet	ADOTable1
TDBGrid	DataSource	DataSource1

**Listing program :**

**TOMBOL SIMPAN**

```

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
if (edit1.Text<>")and (edit2.Text<>") and
(edit3.Text<>")and(combobox1.Text<>")then
begin
with ADOTable1 do
begin
ADOTable1.Append;
ADOTable1['KODE_BARANG']:=edit1.Text;
ADOTable1['NAMA_BARANG']:=edit2.Text;
ADOTable1['BANYAK']:=edit3.Text;
ADOTable1['SATUAN']:=combobox1.Text;
ADOTable1.Post;
end;
end
else
showmessage('data harus diisi');
end;

```

**TOMBOL BATAL**

```

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
edit1.Text:="";
edit2.Text:="";

```

```
edit3.Text:="";  
combobox1.Text:="";  
ADOTable1.Cancel;  
edit1.SetFocus;  
end;
```

#### **TOMBOL BARU**

```
procedure TForm1.SpeedButton3Click(Sender: TObject);  
begin  
edit1.Text:="";  
edit2.Text:="";  
edit3.Text:="";  
combobox1.Text:="";  
edit1.Enabled:=true;  
edit2.Enabled:=false;  
edit1.SetFocus;  
speedbutton1.Enabled:=true;  
end;
```

#### **TOMBOL TUTUP**

```
procedure TForm1.SpeedButton4Click(Sender: TObject);  
begin  
Application.Terminate;  
end;
```

#### **D. Latihan.**

1. Buat program dengan menggunakan tombol navigasi untuk salah satu tabel dalam database DBstok.mdb (gunakan komponen ADO)
2. Gunakan tabel lain dalam DBstok.mdb untuk membuat form dengan navigasi dengan menggunakan komponen DBNavigator (gunakan komponen ADO)
3. Tambahkan tombol untuk proses pengeditan data dan penghapusan data.



# MODUL V

## MEMFORMAT DAN MEMVALIDASI DATA

### A. Kompetensi dan Indikator

Kompetensi Dasar: Memahami tentang Format dan Validasi Data

Indikator : - Mampu membuat Validasi terhadap suatu data

### B. Dasar Teori

Validasi data adalah data yang bukan seharusnya diinputkan tidak dapat diinputkan/tidak diterima oleh komponen, semisal data yang **bertipe Number** tidak dapat diinputkan dengan yang **bertipe Text**.

### C. Praktek

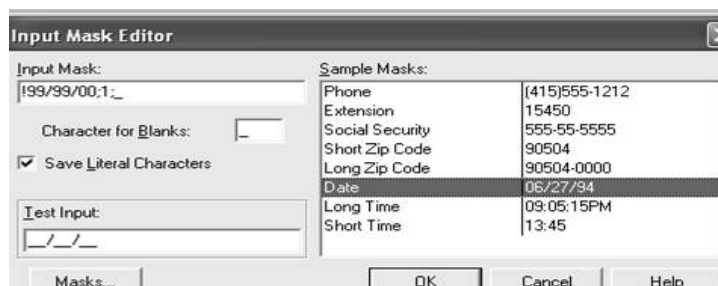
Praktekkan pengisian format pada **TMaskEdit** agar karakter seperti A tidak bisa di masukkan. Ikuti langkah berikut :

- a. Buat form dengan tampilan sebagai berikut :



The image shows a form with three input fields. The first field is labeled 'Nomor Retur' and is empty. The second field is labeled 'Tanggal Retur' and contains the date '7/27/2005'. The third field is labeled 'Banyak' and is empty. Below the fields are three buttons: 'Simpan', 'Batal', and 'Selesai'.

- b. klik pada ... yang terletak disebelah **kanan MaskEdit**, dan pilih Date hasilnya sebagai berikut .



- c. Modifikasikan Input Masks menjadi **!99/99/0000;1;\_**
- d. Klik OK untuk menutup **input mask editor**.

### **C. Latihan**

Gunakan validasi data ini terhadap aplikasi yang telah dibuat sebelumnya!



## MODUL VI

### DATA MODUL

#### A. Kompetensi dan Indikator

Kompetensi Dasar: Memahami tentang Data Modul

Indikator : - Mampu membuat Data Modul terhadap suatu Aplikasi

#### B. Dasar Teori

Data Modul adalah suatu

Untuk membuat data modul pilihlah File, New Data Modul. Selanjutnya tempatkan komponen lain seperti TADOTable, TQuery dan sebagainya.

Langkah pembuatan datamodul.

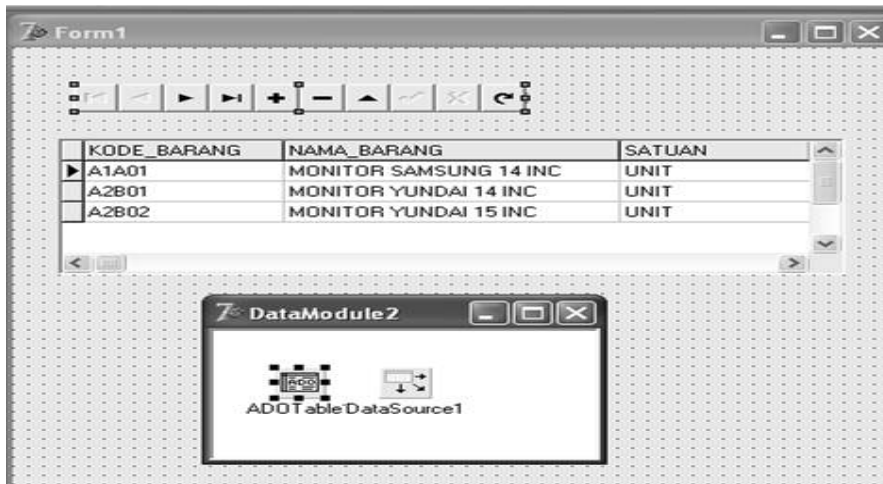
1. Buat aplikasi baru dengan memilih **File, New Application**.
2. Buat sebuah **datamodul** dengan memilih **File, New Data Modul**.
3. Klik **page ADO** pada komponen **pallette**, lalu tempatkan komponen ADOTable di dalam **data modul**.
4. Tempatkan pula **DataSource** didalam **data modul**.
5. Klik page Data Control pada **Component Palette**, lalu tempatkan **DBGrid** pada form.
6. Ubah **Property komponen** Adotable seperti.

Property
ConnectionString
TableName
Active

7. Klik komponen **DataSource1**, kemudian ubah property **DataSet** dengan **Table1**.
8. Tempatkan **DBNavigator** pada form.



9. Untuk mengakses dataset yang ada pada Datamodul, form harus menggunakan **Datamodul**. Caranya **Klik menu file, use unit**. Pilih **unitnya lalu OK**.
10. Klik **DBGrid**, ubah property **DataSource** pada Object Inspector menjadi **DataModul2.DataSource1**.
11. Lakukan langkah 10 pada **DBNavigator**.



**Gambar** Menggunakan DataModul

### **Latihan.**

Buat data modul dengan beberapa **TADOTable** dan **TDataSource** yang akan menampilkan beberapa database yang ada.



## MODUL VII

### **DCL (Data Control Language)**

#### **A. MENAMBAHKAN AKSESORIS KALENDER.**

Menambahkan komponen MontCalender kedalam form.

Langkah yang harus di lakukan :

1. Klik pada halaman **Win32** pada Component palette.
2. Klik icon **MontCalender**.
3. Pada **objek inspector** lakukan perubahan pada .

<b>Komponen</b>	<b>Property</b>	<b>Isi</b>
MounthCalender	ShowToday	False
MountCalender	Name	MC
Combo box	Name	CB
TADOTable	ConnectionString	Isi alamat database yang akan di hubunngkan. Seperti contoh pertemuan sebelumnya.
	Tablename	Tentukan Nama tabelnya
	Active	True
TdataSource	DataSet	ADOTable1
TDBGrid	DataSource	DataSource1

4. Tambahkan tiga buah Button pada form. Button ( simpan, batal, keluar ).
5. Tampilan form sebagai berikut.



**Gambar** Aplikasi dengan MontCalendar.

Listing Program.

```

procedure TForm1.Button1Click(Sender: TObject);
begin
begin
with ADOTable1 do
begin
ADOTable1.Append;
ADOTable1['nama']:=edit1.Text;
ADOTable1['Tglahir']:=mc.Date;
ADOTable1['Klamin']:=cb.Text;
ADOTable1.Post;
end;
end
end;

```

### **Latihan.**

Coba dengan menggunakan DateTimePicker.

### **B. MENAMPILKAN RECORD MENURUT INDEKS.**

Indeks mempunyai peran penting sebagai:

1. *Alternatif* untuk melakukan pencarian data
2. Menghubung *ketabel* lain.

Untuk memahami lebih lanjut buat form data di bawah ini :



Langkah yang harus di lakukan :

1. Tambahkan **2 Items MainMenu** didalam form.
2. Inputkan 2 buah component **TEdit** di dalam form.
3. Tambahkan komponen **TADOTable**. Ubah property **ConnectionString** menjadi **ADOConnection1** yang dijadikan tempat penyimpanan database. ubah property TableName menjadi table yang diinginkan, dan aktifkan tabel dengan mengubah property **Active** menjadi **True**.
4. Tambahkan komponen **TDataSource** dan ubah propertiy **DataSet** menjadi **ADOTable1**.
5. Tambahkan komponen **TDBGrid** dan ubah property **DataSource** dari komponen tersebut menjadi **DataSource1**.
6. Double klik komponen **Items MainMenu/hapus record**.

Tuliskan listing programnya, sebagai berikut:

```
procedure TForm1.HAPUSRECORD1Click(Sender: TObject);
begin
if messagedlg('data ini akan dihapus',
mtinformation,mbokcancel,0)=mrok then
begin
```

```
ADOTable1.Delete;  
ADOTable1.Refresh;  
end;  
end;
```

7. Pilih komponen TEdit1, klik Event di Object Inspector pilih **On key Press**. Double klik. Isikan listing sebagai berikut.

```
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);  
begin  
Adotable1.IndexName:='kode_barang';  
adotable1.IndexName:='';  
Adotable1.FindNearest([edit1.Text+key]);  
dbgrid1.Refresh;  
Adotable1.FindNearest([edit1.Text+key]);  
edit2.Text:=Adotable1['nama_barang'];  
end;
```

8. Pilih komponen TEdit2, klik Event di Object Inspector pilih **On key Press**. Double klik. Isikan listing sebagai berikut.

```
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);  
begin  
Adotable1.IndexName:='nama_barang';  
Adotable1.FindNearest([edit2.Text+key]);  
Adotable1.FindNearest([Adotable1['kode_barang']]);  
edit1.Text:=Adotable1['kode_barang'];  
end;  
end;
```

Database yang digunakan adalah database yang sudah tersimpan di access seperti pertemuan sebelumnya.

### **Latihan.**

Periksa kesalahan yang ada dalam program edit1 dan edit2.



## **MODUL VIII**

### **Aplikasi Sederhana pengakses MySQL**

#### **Kompetensi dan Indikator**

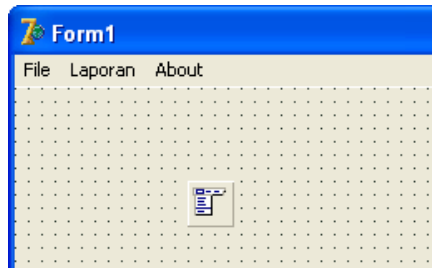
Kompetensi Dasar: Memahami tentang gambaran aplikasi sederhana

Indikator : - Mampu membuat aplikasi sederhana yang terhubung dengan database MySQL

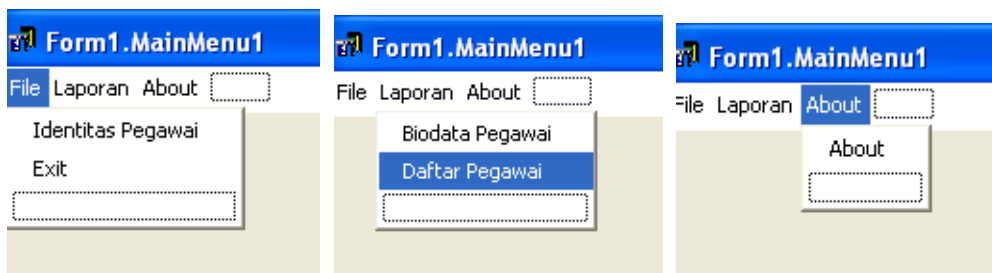
#### **Praktek**

Langkah-langkah yang dikerjakan

1. Buat project baru, dan pada form1 tambahkan komponen mainmenu, seperti gambar:

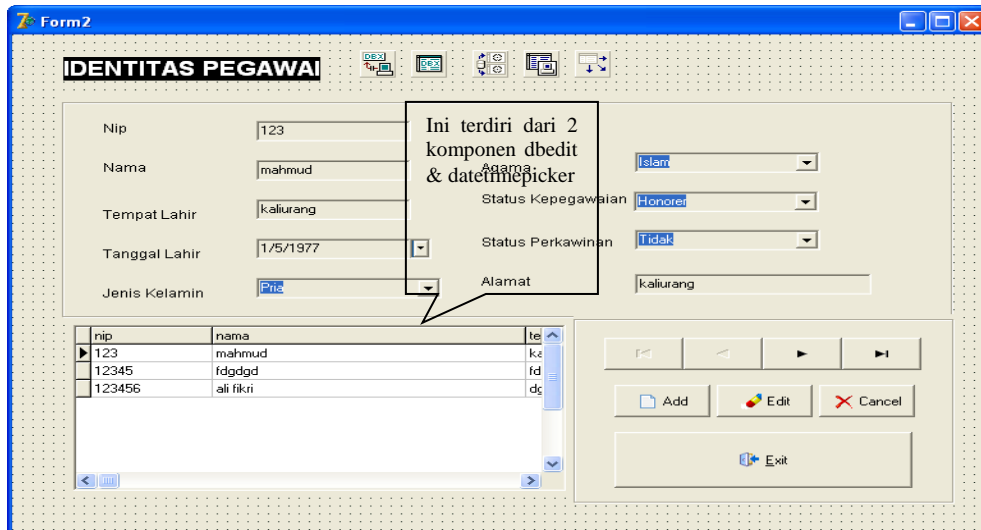


2. Untuk menset main menu, pilih komponen mainmanu, klik kanan maka akan muncul menu designer, kemudian tuliskan menu-menu seperti contoh dalam gambar



3. Buka form2, dan tambahkan komponen label, dbedit, dbcombo, dbgrid, dbnavigator, groupbox, datettimepicker, SQLConnetion, SQLTable, DatasetProvider, ClientDataset, Datasource dan button, seperti gambar





4. Dan untuk setiap komponen yang memiliki koneksi, koneksikan dengan database pegawai tabel data yang pernah dibuat dan hubungkan setiap komponen sebagaimana biasa.

5. Untuk DBEdit dan DBCombo set datasource dan data fieldnya seperti pada dbgrid

6. Pada DBCombo ubah property items isikan sesuai dengan keterangan label, untuk status kepegawaian isikan Honorer, Kontrak, CPNS, dan PNS. Untuk Status isikan Menikah, Belum menikah, Janda dan Duda.

7. Pilih Datetimepicker kemudian pada property events pilih Onchange, klik ganda dan kemudian tuliskan

```
procedure TForm1.DateTimePicker1Change(Sender: TObject);
begin
clientdataset1.fieldbyname('tglahir').AsDateTime
:=datetimepicker1.Date ;
end;
```

8. untuk lebih lengkapnya, lihat pada listing berikut

```
unit Unit2;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
```

```

    Dialogs, ComCtrls, StdCtrls, DBCtrls, Mask, Grids, DBGrids,
ExtCtrls, DB,
    DBClient, SimpleDS, FMTBcd, Provider, SqlExpr,
DBXpress, xpman, Buttons;
type
    TForm2 = class(TForm)
        GroupBox1: TGroupBox;
        Label9: TLabel;
        Label8: TLabel;
        Label7: TLabel;
        Label6: TLabel;
        Label4: TLabel;
        Label3: TLabel;
        Label11: TLabel;
        Label2: TLabel;
        Label5: TLabel;
        DBEdit5: TDBEdit;
        DBComboBox4: TDBComboBox;
        DBComboBox3: TDBComboBox;
        DBComboBox2: TDBComboBox;
        DBComboBox1: TDBComboBox;
        DBEdit4: TDBEdit;
        DateTimePicker1: TDateTimePicker;
        DBEdit3: TDBEdit;
        DBEdit1: TDBEdit;
        DBEdit2: TDBEdit;
        GroupBox2: TGroupBox;
        DBNavigator1: TDBNavigator;
        DBGrid1: TDBGrid;
        Label10: TLabel;
        DataSource1: TDataSource;
        SQLTable1: TSQLTable;
        DataSetProvider1: TDataSetProvider;
        ClientDataSet1: TClientDataSet;
        SQLConnection1: TSQLConnection;
    end;

```

```

    badd: TBitBtn;
    bedit: TBitBtn;
    bcancel: TBitBtn;
    bexit: TBitBtn;
    procedure DateTimePicker1Change(Sender: TObject);
    procedure baddClick(Sender: TObject);
    procedure bcancelClick(Sender: TObject);
    procedure beditClick(Sender: TObject);
    procedure bexitClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    Form2: TForm2;
implementation
uses Unit1;
{$R *.dfm}
//mengisi tgl lahir melalui datetimesticker
procedure TForm2.DateTimePicker1Change(Sender: TObject);
begin
    clientdataset1.fieldbyname('tglahir').AsDateTime
:=datetimesticker1.Date ;
end;
//button ADD diberi name BADD yang apabila diklik akan
otomatis berubah menjadi
//Save pada captionnya untuk menyimpan data
procedure TForm2.baddClick(Sender: TObject);
begin
    if badd.Caption = 'Add' then
    begin
        groupbox1.Enabled :=true;
        badd.Caption :='Save';
    end;
end;

```

```

bedit.Enabled :=false;
bcancel.Caption :='Cancel';
bcancel.Enabled :=true;
bexit.Enabled :=false;
clientdataset1.Append ;
end
else
begin
groupbox1.Enabled :=false;
badd.Caption :='Add';
bedit.Enabled :=true;
bcancel.Caption :='Delete';
bexit.Enabled :=true;
clientdataset1.post;
clientdataset1.ApplyUpdates(0);
end;
end;
//bcancel untuk membatalkan proses dan apabila di klik akan
otomatis captionnya
//berubah menjadi delete yang digunakan untuk menghapus
procedure TForm2.bcancelClick(Sender: TObject);
begin
if bcancel.Caption='Cancel' then
begin
clientdataset1.Cancel ;
bcancel.Caption :='Delete';
badd.Caption :='Add';
bedit.Caption :='Edit';
bedit.Enabled :=true;
bexit.Enabled :=true;
groupbox1.Enabled :=false;
end
else
begin

```

```

if messagedlg('Anda mau menghapus data ini ?',
mtconfirmation,[mbYes, mbNo],0) = mrYes then
begin
clientdataset1.Delete ;
clientdataset1.ApplyUpdates(0);
messagedlg('Data sudah dihapus',mtinformation,[mbOk],0);
end;
groupbox1.Enabled :=false;
bcancel.Caption :='Delete';
bcancel.Enabled :=true;
badd.Enabled :=true;
badd.Caption :='Add';
end;
end;
//bedit untuk mengubah data
procedure TForm2.beditClick(Sender: TObject);
begin
clientdataset1.Edit ;
groupbox1.Enabled :=true;
badd.Caption :='Save';
bcancel.Caption :='Cancel';
badd.Enabled :=true;
bedit.enabled :=false;
bcancel.Enabled :=true;
bexit.Enabled :=false;
end;
//untuk keluar dari form2
procedure TForm2.bexitClick(Sender: TObject);
begin
close;
//ini bisa dieksekusi ketika form1 sudah diisi listing nya
pada langkah 11 dan
//form1 yang dijadikan mainform
form1.Enabled :=true;
end;

```

```

procedure TForm2.FormCreate(Sender: TObject);
begin
groupbox1.Enabled :=false;
bcancel.Caption :='Delete';
end;

end.

```

9. setelah semua teks perintah dimasukkan, kemudian di eksekusi, pastika program yang anda buat berjalan sesuai dengan yang diharapkan

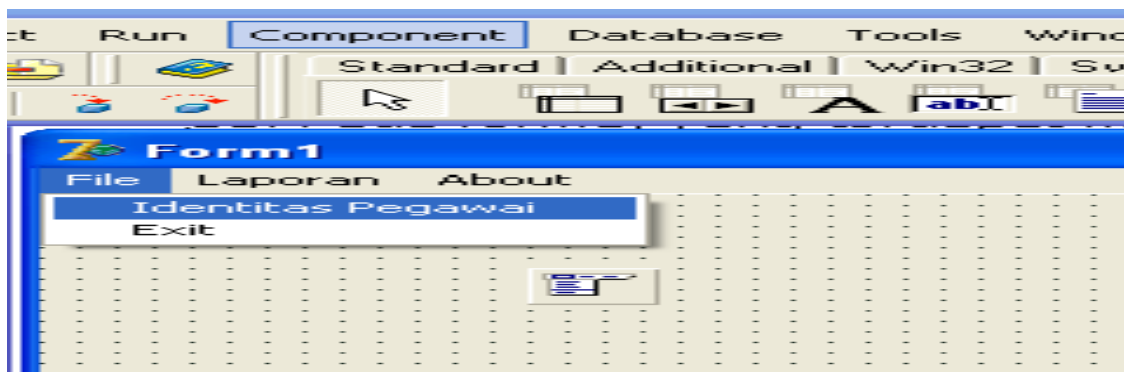
10. untuk berpindah dari satu record kerecord yang lain, gunakan dbnavigator

11. Pada form1, yang terdapat mainmenu, maka lakukan langkah-langkah sebagai berikut: klik menu identitas pegawai dan tuliskan program sebagai berikut:

```

procedure TForm1.IdentitasPegawai1Click(Sender: TObject);
begin
form2.ShowModal; end;

```



12. pada menu file exit, klik ganda dan berikan perintah application.terminate;

### C. Latihan

Lakukan hal yang sama pada tabel riwayat pendidikan dengan menggunakan form selanjutnya (harus satu project dengan form/unit

yang telah dibahas dalam praktek) dan tambahkan pada mainmenu, menu untuk inputkan riwayat pendidikan!

## **MEMBUAT REPORT DENGAN QREPORT**

### **A. Kompetensi dan Indikator**

Kompetensi Dasar: Memahami tentang pembuatan laporan dengan QReport

Indikator : - Mampu membuat aplikasi sederhana khususnya laporan dengan menggunakan QuickRep yang terhubung dengan database MySQL

### **B. Praktek**

Langkah-langkah yang dikerjakan

1. siapkan form baru (seperti gambar), dan tambahkan beberapa komponen QReport seperti : QuickRep, Qrband, QRLabel,QRDBText, label, edit, button, dan komponen-komponen untuk koneksi
2. koneksikan komponen-komponen yang ada, termasuk QuickRep dan QRDBText yang dikoneksikan dengan komponen-komponen yang telah terkoneksi pada database dan tabel.
3. tambahkan perintah pada button1click, sbb:

```
procedure TForm3.Button1Click(Sender: TObject);
begin
form1.enabled:=true;

sqldataset1.Close ;
sqlconnection1.Close ;

sqldataset1.CommandText :='select* from tdata where
nip=:nip';
sqldataset1.ParamByName('nip').AsString:=edit1.Text;

sqldataset1.Open ;
sqlconnection1.Open ;
quickrepl.Preview ;
```

CLOSE;

end;

4. eksekusi form ini, untuk memastikan koneksi berjalan dengan baik, tambahkan terlebih dahulu sebuah DBGrid yang digunakan untuk memastikan koneksi antara database/tabel berjalan baik dengan konektor yang ada

5. Kalau sudah normal berjalannya, hubungkan form laporan ini dengan mainmenu sebelumnya.

The screenshot shows a Delphi form window titled "Form3". At the top left, there is a text label "Inputkan Nip:" followed by an empty text input field. To the right of the input field is a "Preview" button. Further right is a toolbar with several icons, including one labeled "DBX". Below the toolbar is a data grid with a grid background. The grid has 5 columns and 3 rows. The first row is a header row with the text "BIODATA" in the first column. The second row contains the following data: "[NIP]" in column 1, "[nip]" in column 2, "[NAMA]" in column 3, "[nama]" in column 4, and "[TEMPAT/TANGGAL LAHIR]" in column 5. The third row contains the following data: "[JENIS KELAMIN]" in column 1, "[kelamin]" in column 2, "[AGAMA]" in column 3, "[agama]" in column 4, and "[ALAMAT]" in column 5. The text "Column Header" is visible at the bottom left of the grid area.

### C. Latihan

1. Buat Laporan lainnya, yang dapat dibentuk dari kedua tabel yang ada !
2. Buat Aplikasi Sederhana lainnya !
3. Pastikan bahwa Anda sudah bisa membuat Program Aplikasi Sederhana dengan Database !



