

Mengakses Database MS. SQL Server 2000 Menggunakan ADO di MS. Visual FoxPro 9.0

Oleh :

Muhammad Miftakhul Amin, S. Kom.

miftakhulamin@plasa.com

<http://www.mafisamin.org>

*Dipublikasikan dan didedikasikan
untuk perkembangan pendidikan di Indonesia melalui*

MateriKuliah.Com

Lisensi Pemakaian Artikel:

*Seluruh artikel di **MateriKuliah.Com** dapat digunakan, dimodifikasi dan disebarakan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut Penulis. Hak Atas Kekayaan Intelektual setiap artikel di **MateriKuliah.Com** adalah milik Penulis masing-masing, dan mereka bersedia membagikan karya mereka semata-mata untuk perkembangan pendidikan di Indonesia. **MateriKuliah.Com** sangat berterima kasih untuk setiap artikel yang sudah Penulis kirimkan.*

1.1 DESAIN DATABASE

Pada bagian ini akan dibuat sebuah database menggunakan MS. SQL Server 2000 sebagai bahan latihan pada pembahasan selanjutnya. Database di SQL Server tidak hanya terdiri dari table-table saja, melainkan terdiri dari beberapa objek lain yang membangun database. Objek-objek tersebut satu sama lain saling terkait untuk menciptakan sebuah database yang tangguh.

Dengan menggunakan Tool Enterprise Manager, buat sebuah database dengan nama DBSiakad, Setelah membuat database langkah berikutnya yang kita lakukan adalah membuat table untuk menyimpan data yang akan kita olah. Untuk membuat table ikuti langkah-langkah berikut :

1. Klik **Node Database** → Pilih **DBSiakad** → Sehingga muncul objek-objek database → Pilih Objek **Tables** → klik kanan → pilih **New Table ...**
Desain table beserta skemanya seperti di bawah ini :

	Column Name	Data Type	Length	Allow Nulls
?	IDFakultas	int	4	
	fakultas	varchar	50	✓
	description	varchar	50	✓

Gambar 1.1 Skema Tabel rFakultas

2. Setelah selesai mendesain table seperti di atas, simpan dengan nama **rFakultas**.
3. Buat satu tabel lagi dengan nama **rJurusan** dengan skema tabel sebagai berikut :

	Column Name	Data Type	Length	Allow Nulls
?	IDJurusan	int	4	
	IDFakultas	int	4	
	Jurusan	nvarchar	50	✓
	Jenjang	char	2	✓

Gambar 1.2 Skema Tabel rJurusan

4. Kemudian Lanjutkan dengan mendesain tabel-tabel dengan desain sebagai berikut :
 1. rMahasiswa

	Column Name	Data Type	Length	Allow Nulls
?	NPM	char	8	
	Nama	varchar	50	✓
	tmpLahir	nvarchar	50	✓
	tglLahir	datetime	8	✓
	sex	char	1	✓
	Alamat	ntext	16	✓
	IDJurusan	int	4	
	NIK	char	9	

Gambar 1.3 Skema Tabel rMahasiswa

2. rDosen

	Column Name	Data Type	Length	Allow Nulls
🔑	NIK	char	9	
	Nama	nvarchar	50	✓
	tmpLahir	nvarchar	50	✓
	tglLahir	datetime	8	✓
	sex	char	1	✓
	alamat	ntext	16	✓
	NoTelp	varchar	14	✓
	Foto	image	16	✓

Gambar 1.4 Skema Tabel rDosen

1.2 SEKILAS ADO

Mengapa anda menggunakan ADO ketika membuat aplikasi menggunakan VFP, bukankah dengan menggunakan RV dan SPT yang disediakan VFP telah cukup untuk membuat aplikasi database client-server. Yang jelas ada beberapa hal yang dapat anda lakukan dengan ADO yang tidak dapat anda lakukan ketika bekerja dengan RV maupun SPT. Dalam beberapa contoh aplikasi di bab ini anda akan menemukan beberapa jawabannya.

Karena ADO merupakan sebuah *Component Object Model* (COM), maka komponen ADO ini dapat dipergunakan di bahasa pemrograman yang berjalan di bawah sistem operasi windows seperti VB, Delphi, VC demikian juga di VFP. Untuk menggunakan ADO dalam aplikasi VFP, anda cukup mengkaitkan object ADO dengan aplikasi anda. Anda dapat menggunakan kontrol berupa ADODC ataupun dengan mengkaitkan Library ADO. Dalam prakteknya penggunaan kontrol ADODC dalam membangun aplikasi sangat diminimalisir. Hal ini cukup beralasan. Dapat dibayangkan ketika anda membangun aplikasi yang melibatkan form yang cukup banyak, maka setiap form akan berisi kontrol ADODC yang menghendaki anda mengkonfigurasi koneksi yang ada di kontrol ADODC tersebut. Jika ternyata suatu ketika dengan beberapa alasan konfigurasi database server anda berubah, maka secara otomatis konfigurasi kontrol ADODC anda juga akan ikut berubah di setiap form yang ada di aplikasi anda, tentunya hal ini kurang efektif dan kurang fleksibel. Penggunaan kontrol ADODC akan tetap dipakai terutama dengan tujuan untuk mengenalkan konsep pemrograman ADO kepada mereka yang baru mengenal pemrograman ADO.

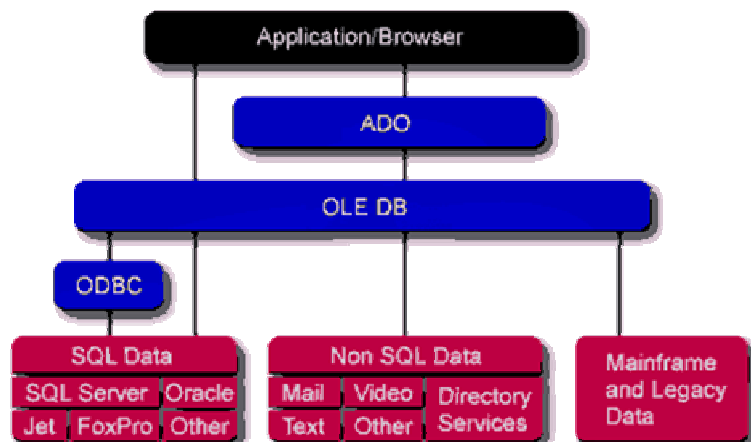
ActiveX Data Object (ADO) merupakan *Application Programming Interface* (API) yang dikhususkan dalam pengaksesan data dan informasi. Kegunaan ADO adalah menyediakan akses aplikasi ke berbagai tipe data, tidak peduli format dan lokasinya. Hal ini dapat dilakukan dengan menggunakan suatu penyedia data **OLE DB** yang memberikan data ke pemakai data (dalam hal ini ADO).

Ketentuan untuk mengakses berbagai sumber data melalui **OLE DB** menimbulkan peningkatan dalam pengembangan penyedia data **OLE DB** pihak-ketiga. Pada saat yang sama perusahaan merilis **Driver ODBC** untuk beragam sumber data, sehingga mereka merilis penyedia data **OLE DB** untuk memenuhi kebutuhan untuk mengakses data via **OLE DB**.

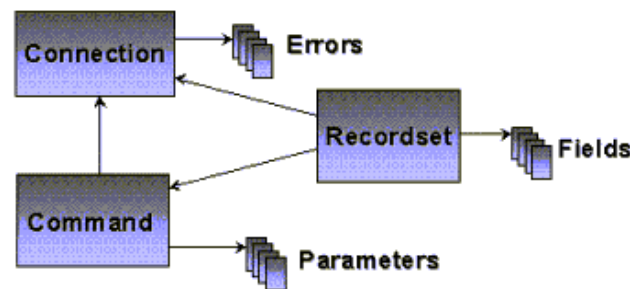
Microsoft juga menyediakan sebuah penyedia data **OLE DB** bagi ODBC sehingga **ADO** bisa mengakses berbagai sumber data via **ODBC**, penyedia data/Provider yang disediakan diantaranya adalah :

- Microsoft OLE DB Provider untuk ODBC (MSDASQL).
- Microsoft OLE DB Provider untuk SQL Server (SQLOLEDB).
- Microsoft OLE DB Provider untuk Access 2000 (Microsoft.Jet.OLEDB.4.0).
- Microsoft OLE DB Provider untuk ORACLE (MSDAORA).
- Dll.

Arsitektur aplikasi ADO dapat anda lihat pada bagan berikut :



Gambar 10.1 Arsitektur Aplikasi Menggunakan ADO



Gambar 10.2 Objek Utama ADO

1.3 OBJEK CONNECTION

ADO mempunyai objek utama yaitu Connection, Command dan Recordset. Sebelum anda dapat menggunakan objek-objek ADO, objek yang pertama kali berperan penting adalah objek Connection. Karena objek inilah yang menjembatani objek-objek lain untuk berhubungan dengan database server. Objek Command dan Recordset memerlukan objek connection untuk menentukan koneksi ke sumber database server.

Untuk menciptakan objek connection di VFP anda cukup mengetikkan kode program sebagai berikut :

```
oConnection =  
CreateObject("ADODB.connection")
```

Selanjutnya anda dapat menambahkan beberapa property penting dari objek connection, seperti berikut :

```
With oConnection  
  
.CursorLocation=adUseClient  
  
.ConnectionString = " Provider = SQLOLEDB.1; Persist  
Security Info=False;User ID=sa;PWD= miftakh;Initial  
Catalog=Northwind;Data Source=SERVER"  
  
.Open  
  
EndWith
```

Penggunaan Microsoft **OLE DB Provider** untuk **SQL Server** sangat dianjurkan karena tidak perlu melewati **ODBC** ataupun **Driver SQL Server**, tetapi langsung berhubungan/koneksi ke Sumber datanya langsung. Untuk menggunakan **Microsoft OLE DB Provider** untuk **SQL Server**, settinglah argument **Provider** pada String Koneksi (ConnectionString) ke **SQLOLEDB** sebagai berikut :

```
Provider=SQLOLEDB.1;  
Data Source=(local);  
User ID=sa;  
Password=miftakh;  
Initial Catalog=DBSiakad
```

String Koneksi memakai beberapa argument sebagai berikut :

Argumen	Penjelasan
Provider	OLE DB Provider untuk SQL Server. Nama penyediannya adalah SQLOLEDB.
Data Source/Server	Nama Server di mana SQL Server berada.
Initial Catalog/Database	Nama database SQL Server
User ID/UID	Nama User SQL Server
Password/PWD	Password untuk User yang login ke SQL Server

1.4 OBJEK RECORDSET

Objek Recordset merupakan inti dari semua aplikasi Client/Server. Objek ini dapat diprogram dimana cursor disimpan. Objek recordset hanya dapat dipergunakan apabila anda telah terhubung dengan sumber data melewati objek Connection.

Untuk menciptakan objek Recordset di VFP anda cukup mengetikkan kode program sebagai berikut :

```
oRecordset = CreateObject("ADODB.Recordset")
```

Ada tiga property penting dari objek recordset, yaitu :

- **CursorLocation**

Sebuah cursor bisa terdapat pada server maupun pada cursor di client, bergantung pada nilai property **CursorLocation**.

Konstanta	Keterangan
<i>adUseClient</i>	Membuka cursor pada client
<i>adUseServer</i>	Membuka cursor pada server

- **CursorType**

ADO mendukung empat jenis cursor, dan anda bisa menentukan jenis cursor yang diinginkan dengan menentukan nilai dari property **CursorType** ke salah satu konstanta di bawah ini :

Konstanta	Keterangan
<i>adOpenForwardOnly</i>	Membuat cursor forward-only
<i>adOpenKeyset</i>	Membuat cursor keyset-driven
<i>adOpenDynamic</i>	Membuat cursor dinamis
<i>adOpenStatic</i>	Membuat cursor statis

- **LockType**

Property LockType mendefinisikan tipe kunci yang digunakan saat update dibuat ke database.

Konstanta	Keterangan
<i>adLockReadOnly</i>	Membuat cursor/recordset yang paling efisien karena , karena jenis cursor ini hanya bisa dibaca.
<i>adLockPessimistic</i>	Record ini sangat mahal karena cursor bertahan selama cursor tersebut dibuka. Ini bisa jadi problematic jika tidak ditangani dengan benar pada client, karena penguncian bisa ditempatkan pada record-record membuatnya terus terbuka menghalangi user lain mengakses data.

<i>adLockOptimistic</i>	Lebih efisien (bergantung pada aplikasinya) karena record-record hanya dikunci saat mereka sedang di-update.
<i>adLockBatchOptimistic</i>	Dibutuhkan untuk recordset terputus. Data dalam SQL Server tidak di-update hingga anda memanggil metode UpdateBatch.

Untuk dapat menggunakan property-property penting dari objek Recordset di atas, di VFP akan mengenali nilai property tersebut sebagai konstanta. Untuk itu anda harus mendefinisikan dulu property-property di atas sebagai konstanta. Bagaimana cara mendefinisikan property-property di atas sebagai konstanta, anda dapat menemukan pembahasan dan contohnya pada bagian sub bab di bab ini.

Berikut ini adalah beberapa metode dan property penting dari objek Recordset :

- 1) Property **State** : menentukan status recordset, apakah recordset terbuka (adStateOpen) atau tertutup (adStateClose). Pengecekan status penting karena recordset hanya bisa dibuat jika status recordset dalam keadaan tertutup (adStateClose).
- 2) Property **BOF** : bernilai True jika posisi record pada record pertama atau table kosong.
- 3) Property **EOF** : bernilai True jika posisi record pada record terakhir.
- 4) Metode **AddNew** : menambahkan sebuah record baru.
- 5) Metode **Update** : memperbaharui perubahan record.
- 6) Metode **Delete** : menghapus sebuah atau sekelompok record.
- 7) Metode **Open** : membuka recordset.
- 8) Metode **Find** : mencari sebuah record dengan kriteria tertentu.
- 9) Metode **Close** : menutup recordset.
- 10) Metode **MoveFirst**, **MoveLast**, **MoveNext**, **MovePrevious** : memindahkan posisi baca pada record.
- 11) Property **RecordCount** : untuk mengetahui jumlah record dari table.

Untuk membuka objek Recordset anda dapat melakukannya dengan notasi program sebagai berikut :

```
With oRecordset
    .ActiveConnection = oConnection
    .Source = "Select * From Customer"
    .CursorType = adOpenStatic
    .LockType = adLockReadOnly
```

```
.CursorLocation = adUseClient  
  
.Open  
EndWith
```

Atau

```
oRecordset.Open("Select * From Customer",;  
                oConnection,;  
                adUseClient,;  
                adLockBatchOptimistic)
```

Atau

```
With oRecordset  
  
.ActiveConnection = oConnection  
.Source = "Customer"  
.CursorType = adOpenStatic  
.LockType = adLockReadOnly  
.CursorLocation = adUseClient  
.Open(,,,adCmdtable)  
EndWith
```

Metode lain yang penting dari objek Recordset adalah pencarian data. Dalam hal ini objek Recordset menyediakan metode **Find**, mekanisme pencariannya dilakukan sama persis dengan pencarian yang dilakukan di VFP ketika anda menggunakan perintah LOCATE dan SEEK, yaitu pencarian secara sekuensial dari satu record ke record berikutnya. Mekanisme penggunaan metode ini adalah sebagai berikut :

```
oRecordset.Find(StringPencarian)
```

Misalnya anda mencari nama mahasiswa dengan NPM 01050203, maka anda dapat menjalankan perintah sebagai berikut :


```
oRecordset.Find("NPM='01050203'")
```

Karena pencariannya dilakukan secara sekuensial, maka hal yang harus anda perhatikan adalah posisi pointer/cursor berada di mana saat proses pencarian dilakukan. Untuk itu anda dapat meletakkan pununjuk cursor ke record pertama, baru kemudian melakukan proses pencarian berdasarkan kriteria yang telah ditentukan.

```
oRecordset.MoveFirst
```

```
oRecordset.Find("NPM='01050203'")
```

Untuk mengakses nilai yang ada di recordset yang berasal dari field/kolom table anda dapat mengakses dengan memberikan notasi sebagai berikut :

```
oRecordset.Fields("NamaField").Value
```

Atau

```
oRecordset.Fields(NomorIndexField).Value
```

Sebagai contoh jika anda akan megakses field NPM dari table rMahasiswa maka anda dapat memperolehnya dengan notasi sebagai berikut :

```
rsMahasiswa.Fields("NPM").Value
```

Atau

```
rsMahasiswa.Fields(0).Value
```

Jika anda menggunakan notasi berupa indeks, maka untuk menentukan field mana yang akan anda akses, nomor indeks dimulai dari nilai indeks 0 (NOL). Kedua notasi program di atas akan menghasilkan nilai kembalian yang sama, yaitu data yang ada di kolom NPM dari table rMahasiswa.

1.5 OBJEK COMMAND

Seperti namanya objek Command dipergunakan untuk menjalankan perintah. Objek Command merupakan objek yang sangat fleksibel, karena memiliki fungsionalitas yang cukup banyak. Objek ini dipergunakan untuk mengeksekusi perintah SQL, Stored Procedure dan beberapa perintah yang dijalankan di sisi database server. Untuk bekerja dengan objek Command, objek ini memerlukan objek Connection sebagai jembatan untuk koneksi ke database server.

Dengan menggunakan objek Command anda dapat menjalankan perintah-perintah sebagai berikut :

1. Stored Procedure di Database Server.
2. Sebuah Tabel di Database Server.
3. Sebuah Recordset.
4. Action Query untuk menjalankan perintah INSERT, UPDATE dan DELETE.
5. Selection Query untuk mengambil data dari Database Server, yang dapat disimpan dalam objek Recordset.

Untuk menciptakan objek Command di VFP anda dapat menciptakannya dengan perintah sebagai berikut :

```
oCommand = CreateObject("ADODB.command")
```

Objek Command memiliki 3 buah metode, yaitu :

1. Cancel
Untuk membatalkan proses dari pemanggilan metode Execute.
2. Create Parameter
Untuk membuat parameter.
3. Execute
Untuk mengeksekusi perintah yang terdapat di property CommandText.
Metode utama dari objek Command adalah metode Execute, yang akan mengeksekusi kode yang ada di property CommandText. Beberapa property penting dari objek Command adalah :
 1. State
Untuk menandai apakah sebuah Command berada dalam kondisi terbuka atau tertutup. Property ini sangat penting diketahui sebelum objek Command dipergunakan untuk menjalankan suatu operasi. Karena objek Command dapat dipergunakan jika kondisinya terbuka.
 2. ActiveConnection
Property ini akan memberi tahu objek Command, database server mana yang akan dikenai perintah dari objek Command.
 3. CommandText
Property ini akan memberi tahu objek Command, perintah apakah yang akan dijalankan.
 4. CommandType
Property ini akan memberi tahu objek Command, type perintah yang telah diisikan di property CommandText.

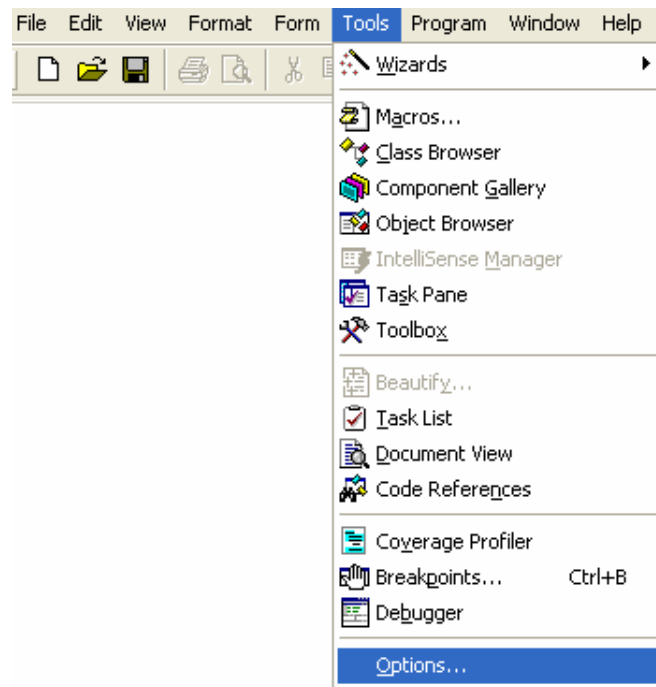
1.6 APLIKASI ADO MENGGUNAKAN ADO-DATA CONTROL

Untuk dapat menggunakan ADO Data Control, terlebih dahulu anda harus mendaftarkan kontrol ADODC ke dalam Form Controls Toolbar Visual FoxPro. Selain kontrol ADODC kontrol lain yang perlu anda tambahkan adalah Microsoft Data Grid Control untuk menampilkan/browsing data yang berasal dari tabel.

Aplikasi berikut memberikan contoh bagaimana menggunakan ADODC sebagai kontrol untuk koneksi ke database server, dan untuk memanipulasi data

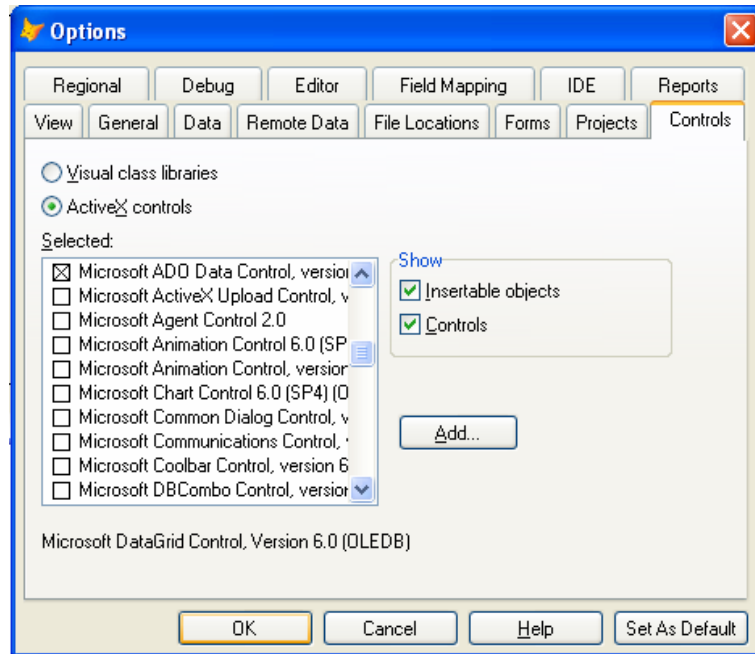
dengan menggunakan metode-metode yang disediakan oleh kontrol ADODC. Untuk itu ikuti langkah-langkah sebagai berikut :

1. Buat sebuah form baru, simpan dengan nama frmADODC.SCX. Kemudian dari form controls toolbar VFP tambahkan dua buah kontrol ADODC dan Microsoft Data Grid Control. Caranya adalah dari menu **Tolls→Pilih Options...**



Gambar 10.3 Pilihan Menu Optios

Kemudian akan ditampilkan windows dialog Options. Pilih tab **Controls**, pilih option **ActiveX controls**. Maka akan ditampilkan control-kontrol .OCX yang dapat anda tambahkan ke dalam aplikasi anda. Selanjutnya pada daftar pilihan **Selected**, centang kontrol **Microsoft ADO Data Control, Version 6.0 (OLEDB)**, dan **Microsoft Data Grid Control, Version 6.0 (OLEDB)**.



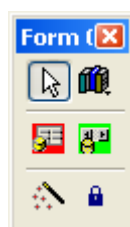
Gambar 10.3 Window Options

Kemudian klik tombol OK. Selanjutnya dari Form Controls Toolbar, pilih icon View Classes→Pilih ActiveX Controls.



Gambar 10.4 Menampilkan kontrol ActiveX

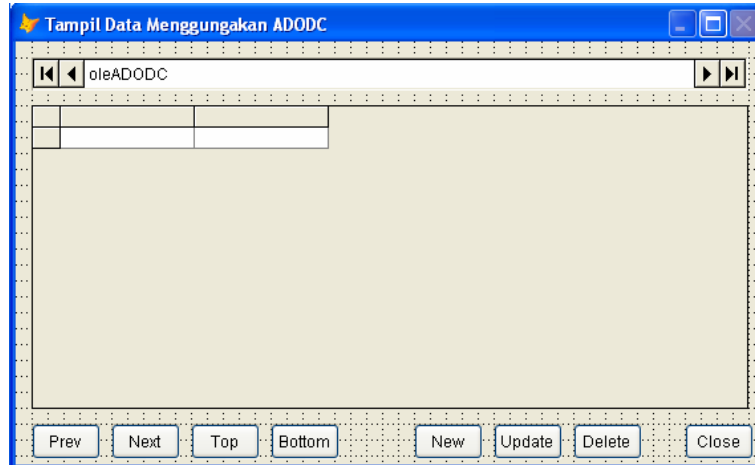
Maka sekarang Form Controls Toolbar anda akan berubah menjadi tampilan sebagai berikut :



Gambar 10.5 Form Control Toolbars

Kontrol Microsoft ADO Data Control, Version 6.0 (OLEDB), dan Microsoft Data Grid Control, Version 6.0 (OLEDB) telah ditambahkan pada Toolbar dan siap digunakan untuk mendesain form.

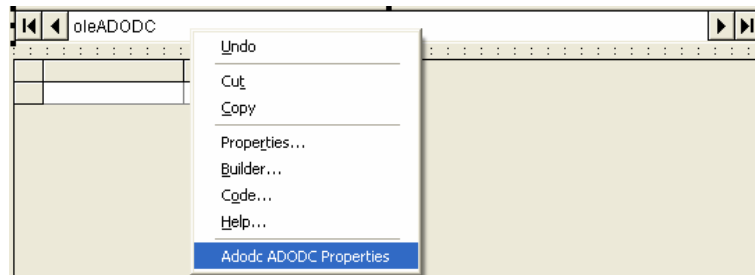
2. Desain Form sebagai berikut :



Gambar 10.6 Desain Form Menggunakan ADODC

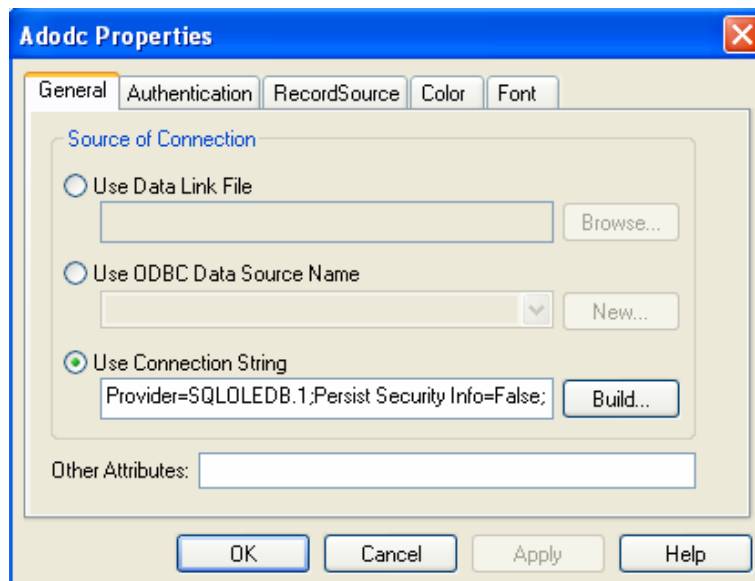
Nama Objek	Properties	Value
Form	Name	frmADODC
	AutoCenter	.T.
	MaxButton	.F.
ODODC	Name	OleADODC
DATAGRID	Name	OleDataGrid
Command1	Name	cmdPrev
	Caption	Prev
Command2	Name	cmdNext
	Caption	Next
Command3	Name	cmdTop
	Caption	Top
Command4	Name	cmdBottom
	Caption	Bottom
Command5	Name	cmdNew
	Caption	New
Command6	Name	cmdUpdate
	Caption	Update
Command7	Name	cmdDelete
	Caption	Delete
Command8	Name	cmdClose
	Caption	Close

3. Kemudian klik kanan objek ADODC → Pilih ADODC Properties.

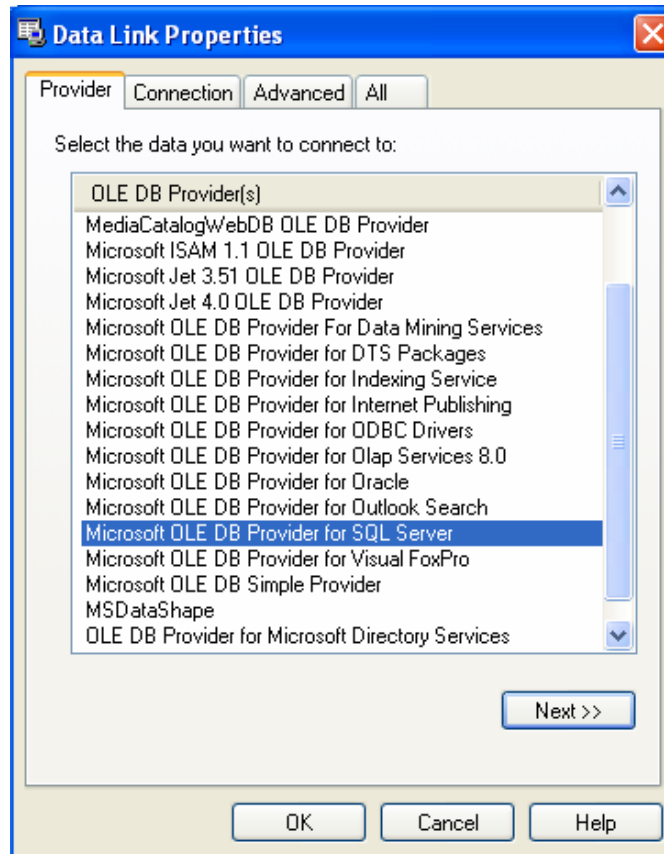


Gambar 10.7 Menentukan Property ADODC

4. Maka akan ditampilkan window ADODC Properties, untuk melakukan koneksi dengan server database SQL Server 2000. Pilih Tab General, akan ditampilkan bagaimana cara anda untuk berkoneksi dengan Database Server. Ada 3 pilihan yang disediakan. Untuk aplikasi kali ini kita akan menggunakan Connection String untuk melakukan koneksi ke database server. Klik tombol **Build...** Maka akan ditampilkan window Data Link Properties. Pilih Tab Provider untuk memilih nama provider penyedia data SQL Server. Dalam hal ini pilih **Microsoft OLE DB Provider for SQL Server**. Selanjutnya klik tombol Next.

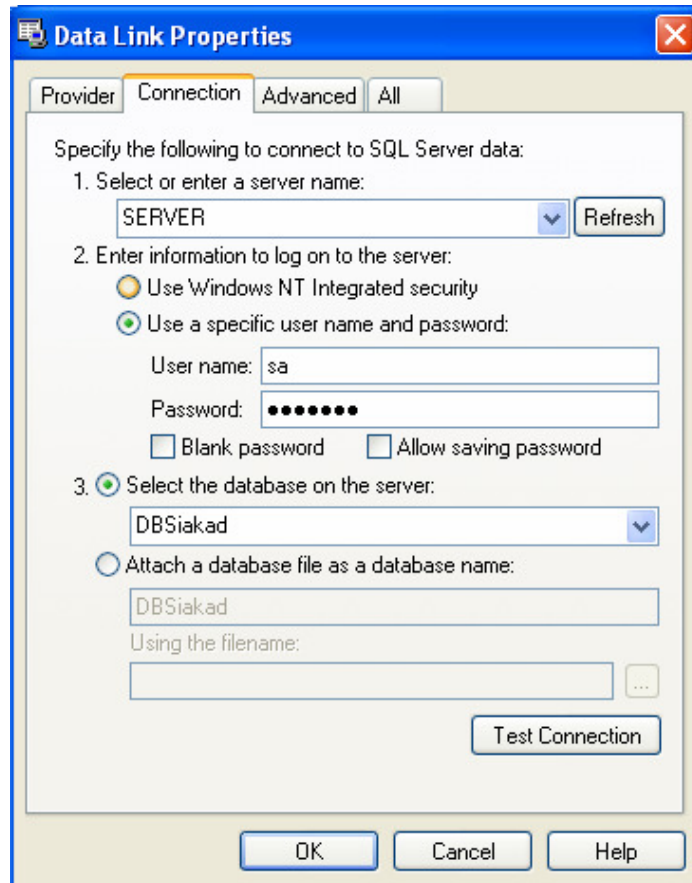


Gambar 10.8 Window ADODC Properties



Gambar 10.9 Pilihan Provider OLEDB

5. Selanjutnya akan ditampilkan Tab Connection yang meminta anda untuk memasukkan alamat server, bagaimana cara anda koneksi ke server, serta nama database yang anda pilih. Seperti tampilan berikut :

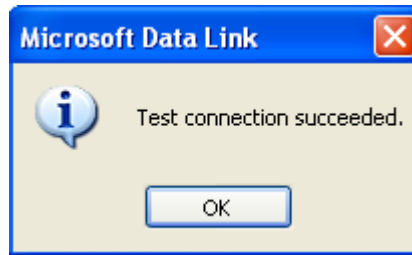


Gambar 10.10 Window Data Link Properties

Catatan penting :

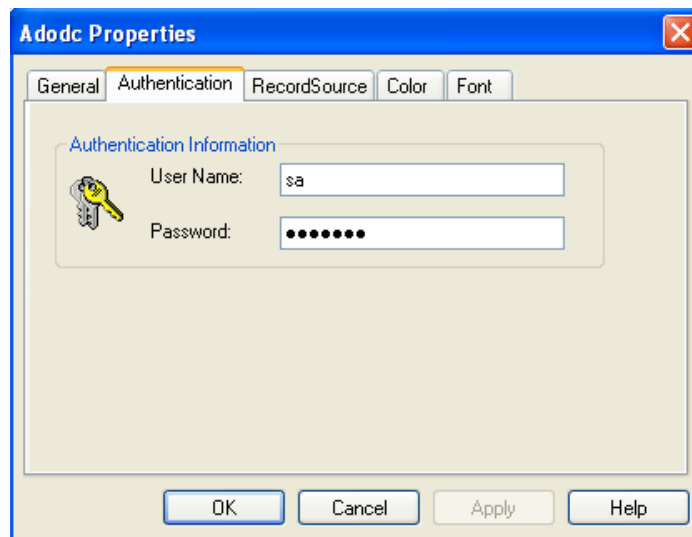
Dalam tahap ini sesuaikan dengan konfigurasi server database yang anda pergunakan. Untuk pilihan cara koneksi ke SQL Server anda dapat juga memilih Use Windows NT Integrated Security.

6. Selanjutnya klik tombol Test Connection, untuk memastikan apakah koneksi yang anda bangun telah sukses. Jika koneksi telah berhasil dilakukan, maka akan dimunculkan window sebagai berikut :



Gambar 10.11 Informasi Koneksi

7. Selanjutnya klik tombol OK, untuk kembali ke window ADODC Properties. Selanjutnya pada window ADODC Properties pilih Tab Authentication untuk melakukan otentikasi user dan password untuk mengakses data.

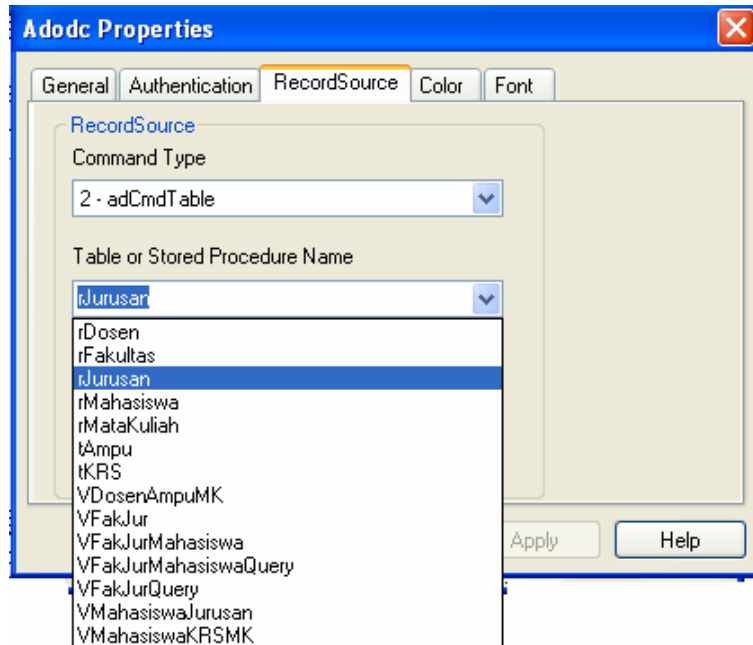


Gambar 10.12 Window Authentication

Catatan penting :

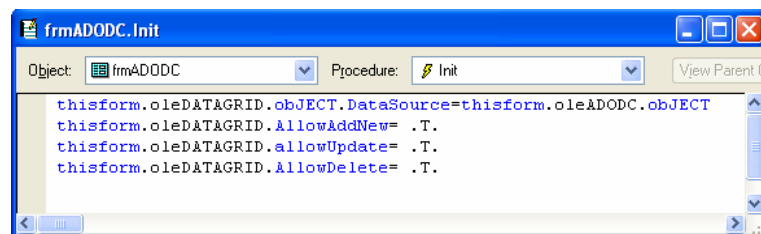
Jika anda memilih Use Windows NT Integrated Security, sebagai pilihan untuk koneksi ke database server, anda dapat meninggalkan tahap ini, karena proses otentikasi telah dibuat pada saat anda masuk ke sistem operasi.

8. Selanjutnya pilih Tab RecordSource untuk memilih table yang akan anda pergunakan untuk aplikasi. Pada Command Type pilih 2-adCmdTable. Kemudian pada pilihan **Table or Stored Procedure Name** pilih table **rJurusan** sebagai sumber data table untuk aplikasi. Kemudian klik tombol OK.



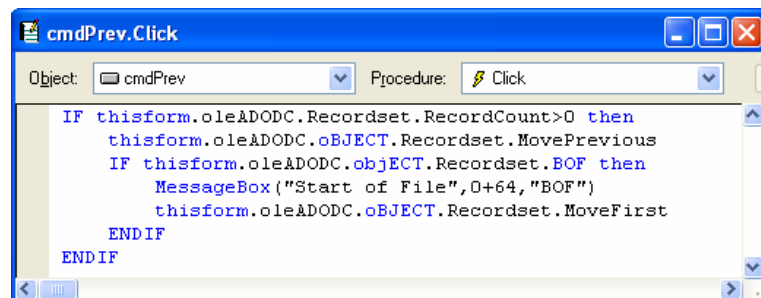
Gambar 10.13 Menentukan Sumber Data Tabel

9. Sampai di sini anda telah mengkonfigurasi kontrol ADODC, sehingga kontrol ADODC tersebut telah terbinding/terkait dengan database dan table yang akan anda gunakan untuk bekerja.
10. Selanjutnya pada metode frmADODC_Init tambahkan kode sebagai berikut :



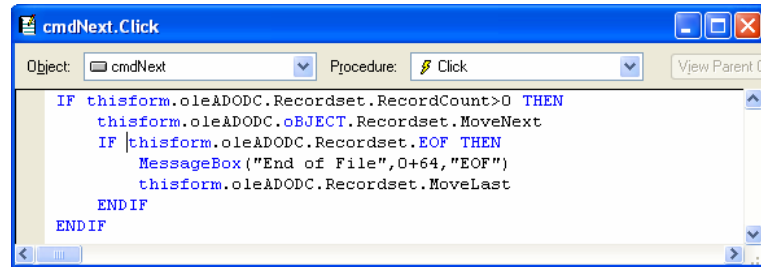
Gambar 10.14 Kode Program di Metode Form.Init

11. Pada even cmdPrev_Click tambahkan kode sebagai berikut :



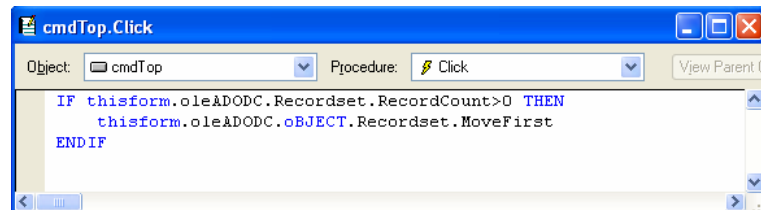
Gambar 10.15 Kode Program Even cmdPrev.Click

12. Prosedur pada cmdNext_Click, isikan kode program berikut :



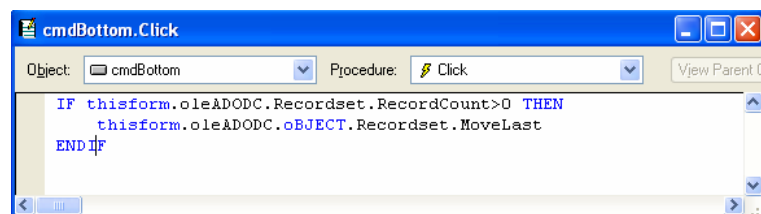
Gambar 10.16 Kode Program Even cmdNext.Click

13. Pada prosedur even cmdTop_Click, tulis kode program berikut :



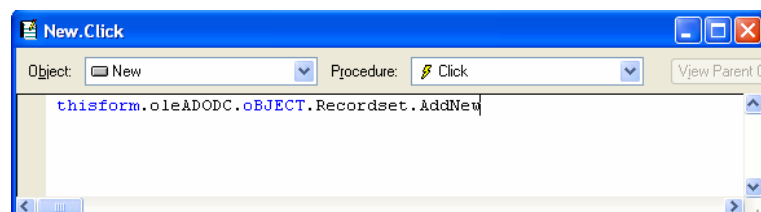
Gambar 10.17 Kode Program Even cmdTop.Click

14. Prosedur pada even cmdBottom_Click, tulis kode program berikut :



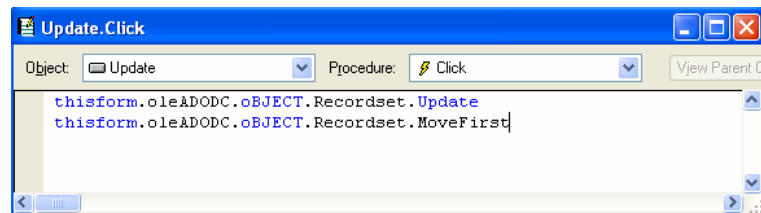
Gambar 10.18 Kode Program Even cmdBottom.Click

15. Pada prosedur cmdNew_Click, tulis kode program berikut :



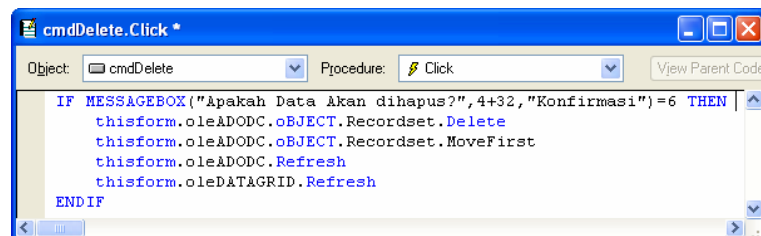
Gambar 10.19 Kode Program Even cmdNew.Click

16. Pada even cmdUpdate_Click, tulis kode program berikut :



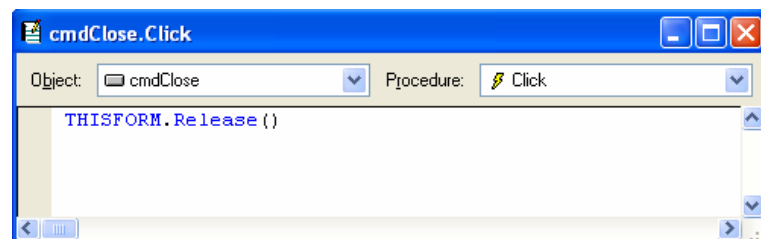
Gambar 10.20 Kode Program Even cmdUpdate.Click

17. Pada even cmdDelete_Click, tulis kode program berikut :



Gambar 10.21 Kode Program Even cmdDelete.Click

18. Selanjutnya pada even cmdSelesai_Click, tulis kode program berikut :



Gambar 10.22 Kode Program Even cmdClose.Click

19. Selanjutnya anda dapat menjalankan form yang telah didesain di atas, anda dapat melakukan navigasi berpindah record dengan memilih pilihan Prev, Next, Top dan Bottom. Anda juga dapat menambahkan record dengan memilih tombol New, menyimpan perubahan dengan menekan tombol Update, serta menghapus record dengan menekan tombol delete. Untuk mengisi data baru dan melakukan perubahan data anda dapat langsung mengetikkan perubahan pada objek datagrid. Pada saat form dijalankan objek ADODC tidak akan tampil di form. Sehingga diperoleh hasil sebagai berikut :



Gambar 10.23 Running Form Menggunakan ADODC

20. Metode-metode dari objek recordset yang dipergunakan untuk melakukan navigasi adalah :

Untuk berpindah ke record pertama:

ObjekADODC.Recordset.MoveFirst

Untuk berpindah ke record berikutnya :

ObjekADODC.Recordset.MoveNext

Untuk berpindah ke record sebelumnya:

ObjekADODC.Recordset.MovePrevious

Untuk berpindah ke record terakhir :

ObjekADODC.Recordset.MoveLast

1.7 APLIKASI ADO MENGGUNAKAN ADODC-RECORDSET

Setelah anda berhasil membuat aplikasi menggunakan kontrol ADODC selanjutnya dalam aplikasi ini anda akan diajak untuk membangun aplikasi menggunakan objek Recordset ADODC, dengan terlebih dahulu membuat koneksi menggunakan objek Connection ADODC. Bedanya dengan aplikasi sebelumnya adalah aplikasi ini menggunakan objek dari form untuk mengetry data.

10.6.1 APLIKASI INSERT

Aplikasi INSERT adalah aplikasi yang digunakan untuk memasukkan data ke dalam table yang ada di dalam database. Pada proses entry data bentuk penulisan perintah INSERT adalah sebagai berikut :

ObjekADODC.Recordset.AddNew

. . .
. . .
. . .

ObjekADODC.Recordset.Update

Dalam membuat aplikasi Entry data biasanya dilakukan proses pencarian terhadap data tertentu. Hal ini dimaksudkan agar tidak terjadi duplikasi data yang nantinya akan disimpan di dalam tabel. Untuk melakukan proses pencarian objek Recordset ADODC menyediakan metode **Find** untuk melakukan pencarian terhadap record data dengan kriteria tertentu. Perintahnya adalah sebagai berikut :

ObjekADODC.Recordset.Find ("Kriteria")

Perlu diingat bahwa pada saat melakukan pencarian, posisi kursor penunjuk record sangat berpengaruh terhadap proses pencarian. Sehingga sebelum dilakukan proses pencarian sebaiknya posisi pointer penunjuk record dipindahkan ke record pertama. Sehingga perintahnya adalah sebagai berikut :

ObjekADODC.Recordset.MoveFirst

ObjekADODC.Recordset.Find ("Kriteria")

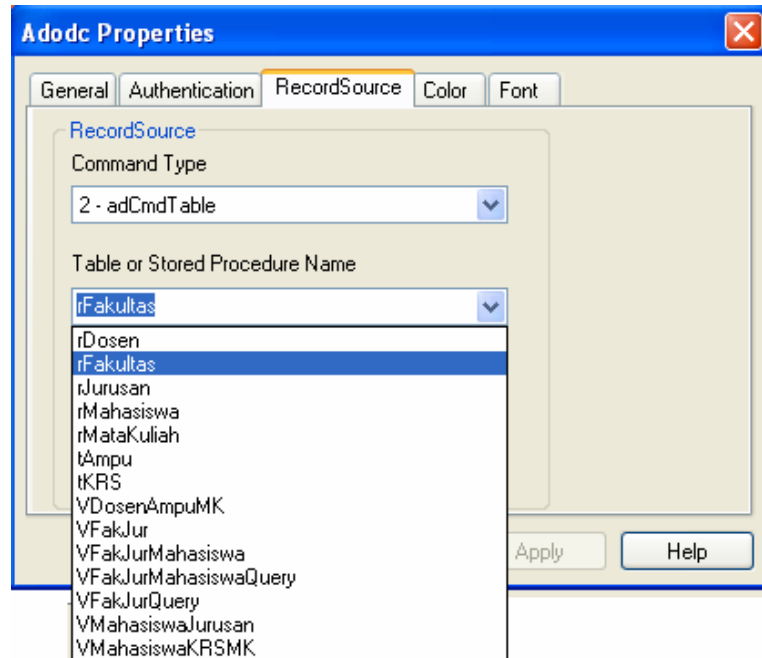
Untuk membangun aplikasi INSERT ikuti langkah-langkah pembuatannya sebagai berikut :

1. Tambahkan dua buah kontrol ActiveX yaitu kontrol ADODC dan DATAGRID ke dalam Form Controls Toolbars, seperti pada pembahasan sebelumnya.
2. Desain form sebagai berikut :

Gambar 10.24 Desain Form Entry Data

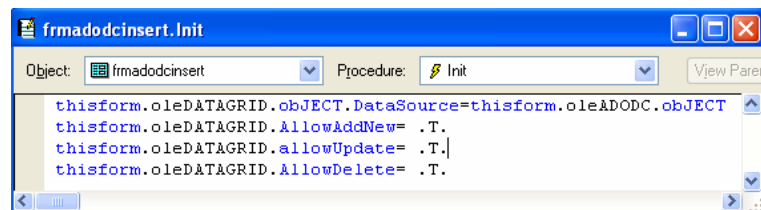
Nama Objek	Properties	Value
Form1	Name Caption	FrmADODCInsert :: Entry Data fakultas dengan ADODC
	MaxButton AutoCenter	.F. .T.
Textbox1	Name	NIK
Textbox2	Name	Nama
Textbox2	Name	Description
Command1	Name Caption	cmdSimpan Simpan
Command2	Name Caption	cmdKosong Kosong
Command3	Name Caption	cmdSelesai Selesai
ADODC	Name	oleADODC
DataGrid	Name	oleDataGrid

- Kemudian pada objek ADODC buatkan koneksinya, serta memilih rFakultas sebagai sumber recordnya, seperti tampak pada gambar berikut ini :



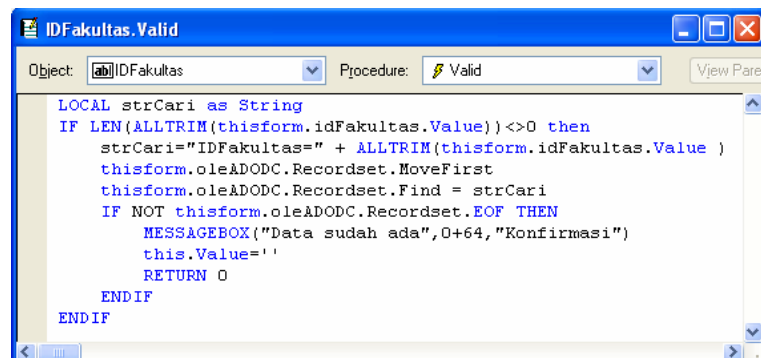
Gambar 10.25 Menentukan Sumber Tabel ADODC

4. Setelah itu pada metode Form_Init, tulis kode program sebagai berikut :



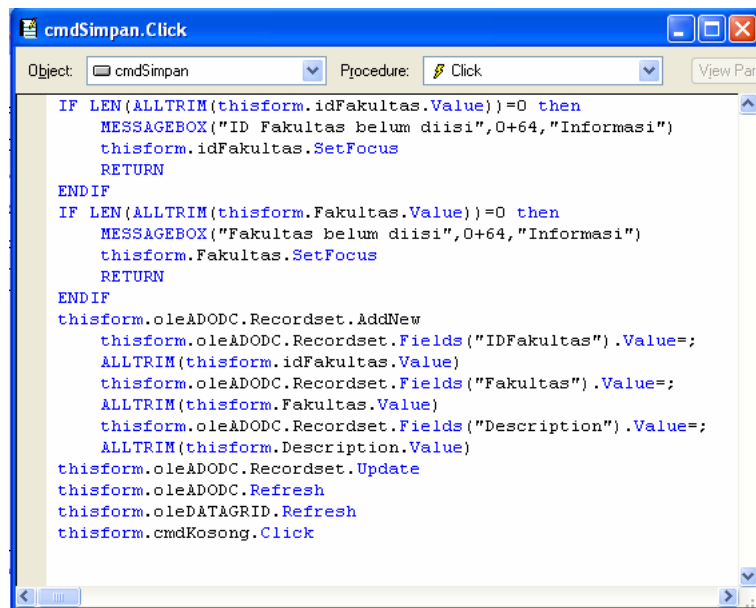
Gambar 10.26 Kode Program Pada Metode Form_Init

5. Selanjutnya pada even idFakultas_Valid tulis kode program sebagai berikut :



Gambar 10.27 Kode Program Pada Even IDFakultas.Valid

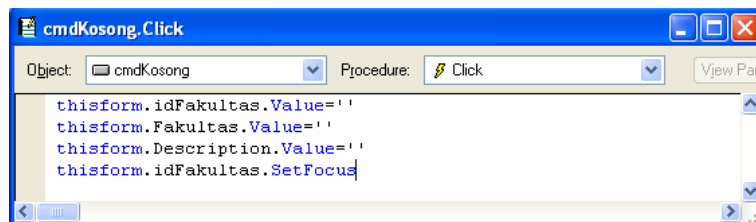
6. Selanjutnya pada even tombol cmdSimpan_Click, tulis kode program sebagai berikut :



```
IF LEN(ALLTRIM(thisform.idFakultas.Value))=0 then
    MESSAGEBOX("ID Fakultas belum diisi",0+64,"Informasi")
    thisform.idFakultas.SetFocus
    RETURN
ENDIF
IF LEN(ALLTRIM(thisform.Fakultas.Value))=0 then
    MESSAGEBOX("Fakultas belum diisi",0+64,"Informasi")
    thisform.Fakultas.SetFocus
    RETURN
ENDIF
thisform.oleADODC.Recordset.AddNew
thisform.oleADODC.Recordset.Fields("IDFakultas").Value=;
ALLTRIM(thisform.idFakultas.Value)
thisform.oleADODC.Recordset.Fields("Fakultas").Value=;
ALLTRIM(thisform.Fakultas.Value)
thisform.oleADODC.Recordset.Fields("Description").Value=;
ALLTRIM(thisform.Description.Value)
thisform.oleADODC.Recordset.Update
thisform.oleADODC.Refresh
thisform.oleDATAGRID.Refresh
thisform.cmdKosong.Click
```

Gambar 10.28 Kode Program Pada Even cmdSimpan.Click

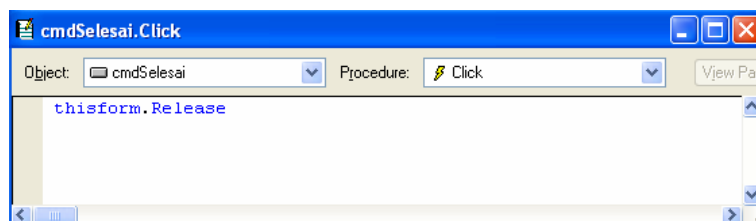
7. Pada even cmdKosong_Click, tuliskan kode program sebagai berikut :



```
thisform.idFakultas.Value=''
thisform.Fakultas.Value=''
thisform.Description.Value=''
thisform.idFakultas.SetFocus
```

Gambar 10.29 Kode Program Pada Even cmdKosong.Click

8. Dan terakhir pada even cmdSelesai_Click, tuliskan kode program sebagai berikut :



```
thisform.Release
```

Gambar 10.30 Kode Program Pada Even cmdSelesai.Click

9. Selanjutnya anda dapat menjalankan form yang telah anda bangun ini, sehingga diperoleh hasil sebagai berikut :

idfakultas	fakultas	description	
1	FE	Fakultas Ekonomi	
2	FTI	Fakultas Teknologi In	
3	FBSB	Fakultas Bahasa Sa	
4	FMIPA	Fakultas MIPA	
5	FTSP	Fakultas Teknik Sipil	
6	FKIP	Fakultas KIP	
*			

Gambar 10.31 Running Form Aplikasi INSERT

10. Jika anda mengetikkan ID Fakultas yang sudah terdaftar dalam table di database, maka akan muncul informasi Data sudah ada. Kejadian ini dipicu oleh metode FIND dari objek recordset, pada even idFakultas_Valid, yang direpresentasikan dalam notasi program berikut :

```
LOCAL strCari as String
IF LEN(ALLTRIM(thisform.idFakultas.Value)) <> 0 then
    strCari = "IDFakultas=" +
        ALLTRIM(thisform.idFakultas.Value)
    thisform.oleADODC.Recordset.MoveFirst
    thisform.oleADODC.Recordset.Find = strCari
    IF NOT thisform.oleADODC.Recordset.EOF THEN
        MESSAGEBOX("Data sudah
ada", 0+64, "Konfirmasi")
        this.Value = ''
        RETURN 0
    ENDIF
ENDIF
```

Objek recordset akan melakukan pencarian dengan terlebih dahulu berpindah dari record pertama sampai record terakhir. Jika ternyata data ditemukan, akan ditampilkan pesan bahwa data sudah ada. Hal ini diketahui dengan nilai kembalian objek recordset .EOF.

10.6.2 APLIKASI UPDATE DAN DELETE

Aplikasi Update adalah aplikasi yang digunakan untuk memperbaharui data yang ada di dalam database. Untuk aplikasi Update dan Delete biasanya dilakukan proses pencarian terhadap data yang akan dikenai proses Update dan Delete, setelah data yang dicari ditemukan maka proses selanjutnya dapat dilaksanakan. Pada proses Update data bentuk penulisan perintah Update hanya ditutup dengan perintah :

...
...
...

ObjekADODC.Recordset.Update

Sedangkan untuk perintah Delete, perintahnya adalah sebagai berikut :

ObjekADODC.Recordset.Delete

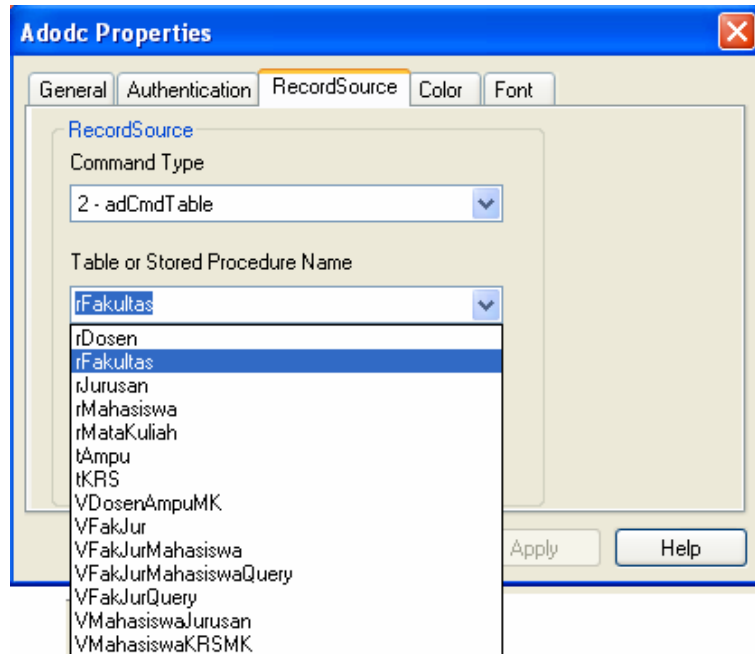
Untuk membangun aplikasi Update dan Delete, ikuti langkah-langkah pembuatannya sebagai berikut :

1. Tambahkan dua buah kontrol ActiveX yaitu kontrol ADODC dan DATAGRID ke dalam Form Controls Toolbars, seperti pada bahasan sebelumnya.
2. Kemudian desain form sebagai berikut :

Gambar 10.32 Desain Form Aplikasi UPDATE dan DELETE

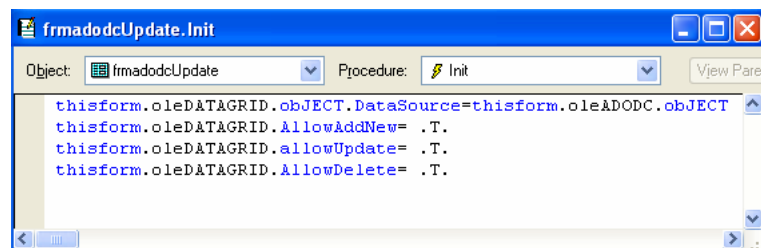
Nama Objek	Properties	Value
Form1	Name Caption MaxButton AutoCenter	FrmADODCInsertEntry Data Fakultas degan ADODC .F. .T.
Textbox1	Name	NIK
Textbox2	Name	Nama
Textbox2	Name	Description
Command1	Name Caption	cmdSimpan Simpan
Command2	Name Caption	cmdHapus Hapus
Command3	Name Caption	cmdKosong Kosong
Command4	Name Caption	cmdSelesai Selesai
ADODC	Name	oleADODC
DataGrid	Name	oleDataGrid

- Kemudian pada objek ADODC buatkan koneksinya, serta memilih rFakultas sebagai sumber recordnya, seperti tampak pada gambar berikut ini :



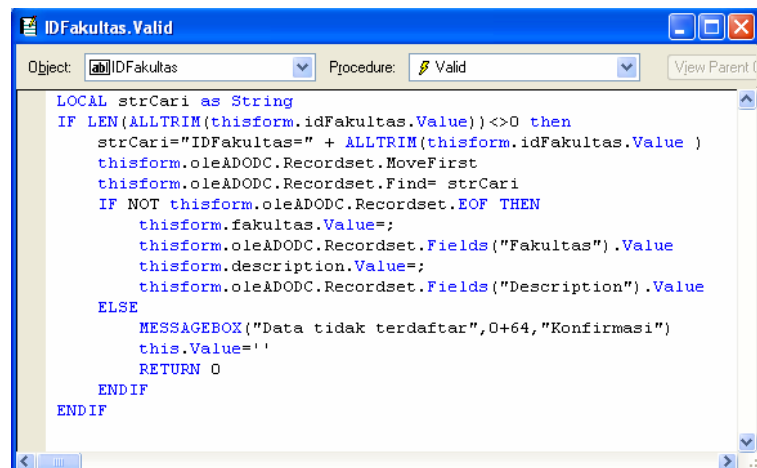
Gambar 10.33 Menentukan Sumber Data ADODC

4. Kemudian pada metode Form_Init, tambahkan kode program sebagai berikut :



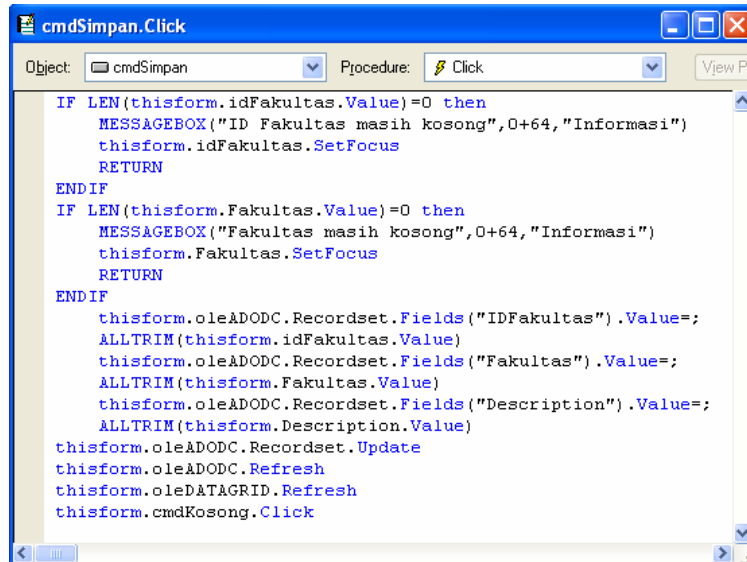
Gambar 10.34 Kode Program Pada Metode Form.Init

5. Selanjutnya pada event idFakultas_Valid, ketikkan kode program berikut ini :



Gambar 10.35 Kode Program Pada Even IDFakultas.Valid

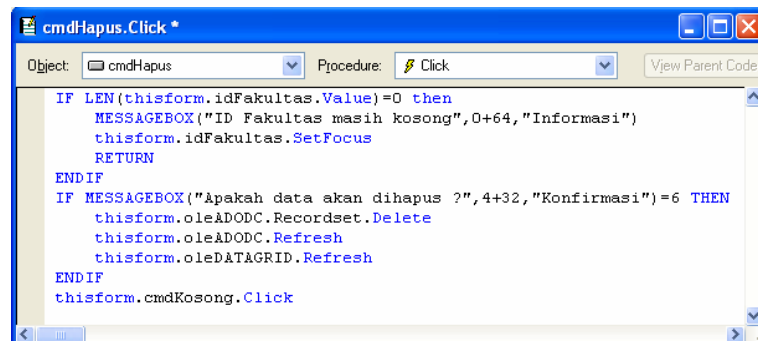
6. Kemudian pada even tombol cmdSimpan_Click, tuliskan kode program berikut :



```
IF LEN(thisform.idFakultas.Value)=0 then
    MESSAGEBOX("ID Fakultas masih kosong",0+64,"Informasi")
    thisform.idFakultas.SetFocus
    RETURN
ENDIF
IF LEN(thisform.Fakultas.Value)=0 then
    MESSAGEBOX("Fakultas masih kosong",0+64,"Informasi")
    thisform.Fakultas.SetFocus
    RETURN
ENDIF
thisform.oleADODC.Recordset.Fields("IDFakultas").Value=;
ALLTRIM(thisform.idFakultas.Value)
thisform.oleADODC.Recordset.Fields("Fakultas").Value=;
ALLTRIM(thisform.Fakultas.Value)
thisform.oleADODC.Recordset.Fields("Description").Value=;
ALLTRIM(thisform.Description.Value)
thisform.oleADODC.Recordset.Update
thisform.oleADODC.Refresh
thisform.oleDATAGRID.Refresh
thisform.cmdKosong.Click
```

Gambar 10.36 Kode Program Pada Even cmdSimpan.Click

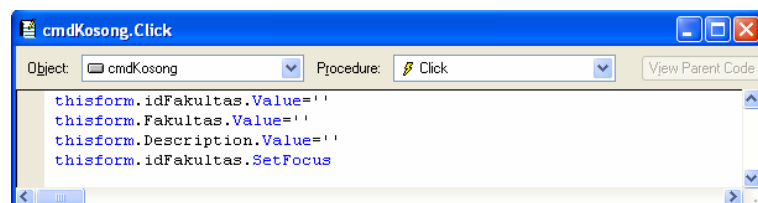
7. Pada cmdHapus_Click, tambahkan kode program sebagai berikut :



```
IF LEN(thisform.idFakultas.Value)=0 then
    MESSAGEBOX("ID Fakultas masih kosong",0+64,"Informasi")
    thisform.idFakultas.SetFocus
    RETURN
ENDIF
IF MESSAGEBOX("Apakah data akan dihapus ?",4+32,"Konfirmasi")=6 THEN
    thisform.oleADODC.Recordset.Delete
    thisform.oleADODC.Refresh
    thisform.oleDATAGRID.Refresh
ENDIF
thisform.cmdKosong.Click
```

Gambar 10.37 Kode Program Pada Even cmdHapus.Click

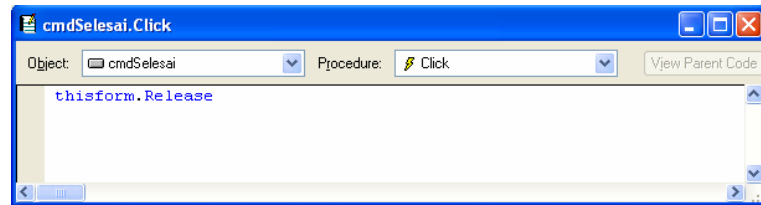
8. Selanjutnya pada cmdKosong_Click, tambahkan kode program sebagai berikut :



```
thisform.idFakultas.Value=''
thisform.Fakultas.Value=''
thisform.Description.Value=''
thisform.idFakultas.SetFocus
```

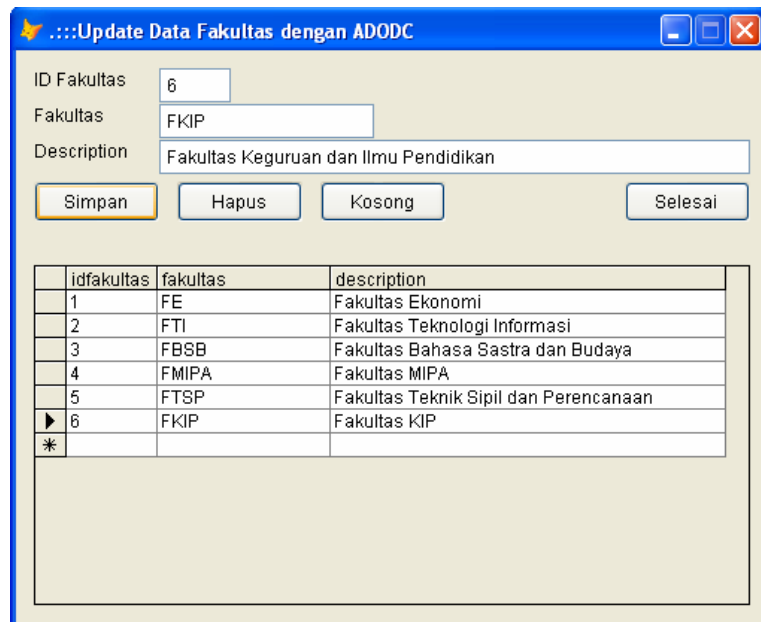
Gambar 10.38 Kode Program Pada Even cmdKosong.Click

9. Terakhir, pada cmdSelesai_Click, tambahkan kode program sebagai berikut :



Gambar 10.39 Kode Program Pada Even cmdSelesai.Click

10. Selanjutnya anda dapat menjalankan programnya sehingga diperoleh tampilan sebagai berikut :



Gambar 10.40 Running Form Aplikasi UPDATE dan DELETE

11. Untuk melakukan proses update, anda cukup menekan tombol simpan. Dan untuk melakukan proses hapus anda dapat menekan tombol Hapus, dengan terlebih dahulu anda diminta konfirmasi apakah akan betul-betul menghapus data yang dimaksud.

1.8 APLIKASI MENGGUNAKAN ADO-RECORDSET

Pada bab sebelumnya anda telah mampu membuat aplikasi menggunakan metode-metode yang disediakan oleh objek recordset menggunakan ADO Data Control.

Pada bab ini anda akan diajak bagaimana membangun sebuah aplikasi yang serupa tetapi langsung menggunakan objek ADO-Recordset, tanpa menggunakan kontrol ADODC.

Koneksi yang dibuat menggunakan ConnectionString dan menggunakan objek-objek yang diturunkan langsung dari objek ADO Library. Untuk membuatnya ikuti langkah-langkah pembuatannya sebagai berikut :

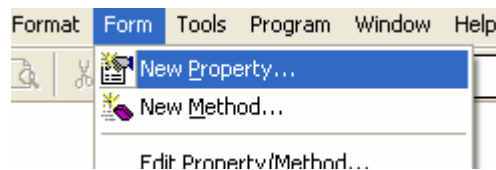
1. Tambahkan sebuah form dalam Project anda, desain formnya sebagai berikut :

Gambar 10.41 Desain Form Aplikasi Menggunakan ADO-Recordset

Nama Objek	Properties	Value
Form1	Name Caption MaxButton AutoCenter	frmMahasiswa Entri Data Mahasiswa .F. .T.
Textbox1	Name MaxLength Enabled	NPM 8 .F.
Textbox2	Name Enabled	Nama .F.
Textbox3	Name Enabled	tmpLahir .F.
Textbox4	Name Enabled	tglLahir .F.
Combo1	Name Enabled	Sex .F.
Textbox5	Name Enabled	Alamat .F.
Combo2	Name Enabled	IDJurusan .F.

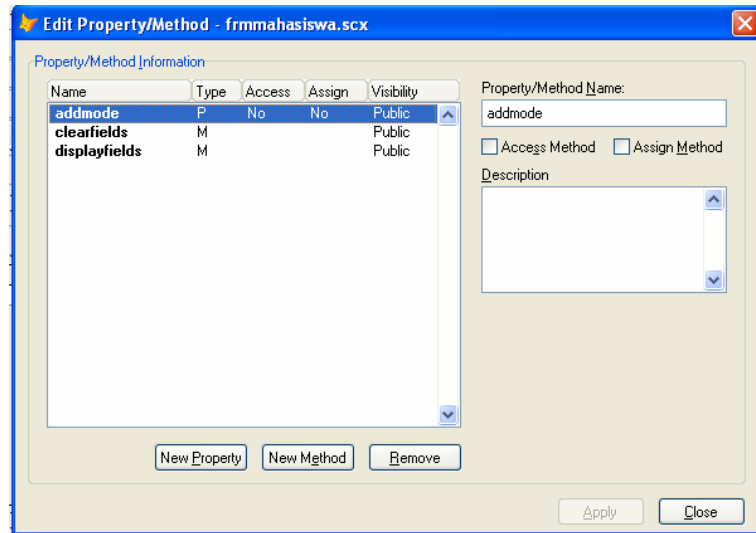
Combo3	Name Enabled	NIK .F.
Textbox6	Name Enabled	txtRecCount .F.
Command1	Name Caption	cmdPrev Preview
Command2	Name Caption	cmdNext Next
Command3	Name Caption	cmdTop Top
Command4	Name Caption	cmdBottom Bottom
Command4	Name Caption	cmdAdd Add
Command4	Name Caption	cmdEdit Edit
Command4	Name Caption	cmdDelete Delete
Command4	Name Caption	cmdClose Close
DataGrid1	Name	GridMahasiswa

2. Selanjutnya tambahkan 1 buah property dan 2 buah metode buatan sendiri, property yang dibuat adalah **addmode**, sedangkan metode yang dibuat adalah **clearfields** dan **displayfields**. Untuk membuat property dari menu Form→Pilih New Property... dan untuk metode pilih menu Form→New Method... Seperti gambar berikut :



Gambar 10.42 Menu Property dan Method baru

3. Pastikan anda telah memiliki beberapa metode dan property yang dibuat sendiri, dengan memilih Form→Edit Property/Method... seperti gambar berikut :



Gambar 10.43 Daftar Metode dan Property yang Dibuak Sendiri

4. Selanjutnya tambahkan kode program ke beberapa prosedur sebagai berikut :

Prosedur Form.Init

```
*---- CursorTypeEnum Values ----
#define ADOPENFORWARDONLY          0
#define ADOPENKEYSET                1
#define ADOPENDYNAMIC              2
#define ADOPENSTATIC               3

*---- LockTypeEnum Values ----
#define ADLOCKREADONLY             1
#define ADLOCKPESSIMISTIC          2
#define ADLOCKOPTIMISTIC           3
#define ADLOCKBATCHOPTIMISTIC     4

*---- ObjectStateEnum Values ----
#define ADSTATECLOSED              0x00000000
#define ADSTATEOPEN                0x00000001
#define ADSTATECONNECTING          0x00000002
#define ADSTATEEXECUTING           0x00000004
```

```

*----- CursorLocationEnum Values -----
#define ADUSESERVER          2
#define ADUSECLIENT          3

SET DATE AMERICAN
SET CENTURY ON
WITH THISFORM

    *-- Membuat Objek-Objek ADO Sebagai property Form.
    .AddProperty('oKoneksi',CREATEOBJECT('ADODB.Connection'))
    .AddProperty('oRSMahasiswa',CREATEOBJECT('ADODB.RecordSet'))
    .AddProperty('oRSJurusan',CREATEOBJECT('ADODB.RecordSet'))
    .AddProperty('oRSPA',CREATEOBJECT('ADODB.RecordSet'))

    *-- Membuat String Koneksi ke SQL Server
    *-- dengan Mode Trusted Connection
    strKoneksi="DRIVER={SQL Server}; ";
        + "SERVER=SERVER; ";
        + "DATABASE=DBSiakad; ";
        + "Trusted_Connection=Yes"
    .oKoneksi.ConnectionString = strKoneksi

    .oKoneksi.Open()
    IF .oKoneksi.State = ADSTATEOPEN

        .oRSMahasiswa.ActiveConnection = .oKoneksi
        .oRSMahasiswa.CursorType = adOpenKeySet

```

```

        .oRSMahasiswa.LockType      = adLockOptimistic

        .oRSMahasiswa.Open("rMahasiswa")
        .txtRecCount.Value =
.oRSMahasiswa.RecordCount
        .oRSMahasiswa.MoveFirst

        .oRSJurusan.ActiveConnection = .oKoneksi
        .oRSJurusan.CursorType = adOpenKeySet

        .oRSJurusan.LockType      = adLockOptimistic

        .oRSJurusan.Open("rJurusan")

        .oRSPA.ActiveConnection = .oKoneksi
        .oRSPA.CursorType = adOpenKeySet

        .oRSPA.LockType      = adLockOptimistic

        .oRSPA.Open("rDosen")

*-- Memanggil Metod
THISFORM.displayfields(.oRSMahasiswa )

.sex.AddItem("L")
.sex.AddItem("P")
.sex.listindex=1
*-- Menambahkan Recordset ke Kontrol Combo
DO WHILE .NOT. thisform.oRSJurusan.EOF

```

```

        thisform.idjurusan.AddItem;

        (STR(thisform.orsJurusan.fields("IDJurusan").value
) ;

        + "|" +
thisform.orsJurusan.fields("Jurusan").value)
        thisform.oRSJurusan.movenext
    ENDDO
    thisform.idjurusan.ListIndex=1
    DO WHILE .NOT. thisform.oRSPA.EOF
        thisform.NIK.AddItem;
        (thisform.orsPA.fields("NIK").value;
        + "|" +
thisform.orsPA.fields("Nama").value)
        thisform.oRSPA.movenext
    ENDDO
    thisform.NIK.ListIndex=1
ENDIF
ENDWITH

```

Prosedur Form.Destroy

```

WITH THISFORM
    .oRSMahasiswa.Close() && Close the Record Set.
    .oRSJurusan.Close() && Close the Record Set.
    .oRSPA.Close() && Close the Record Set.
    .oKoneksi.Close() && Close the Connection
ENDWITH

```

Prosedur Form.displayfields

```

LPARAMETERS toRecordSet
WITH toRecordSet

```

```

THISFORM.npm.Value      = .Fields("npm").Value
THISFORM.nama.Value     = .Fields("nama").Value
THISFORM.tmplahir.Value = .Fields("tmplahir").Value
THISFORM.tglLahir.Value =
DTC(TOD(.Fields("tglLahir").Value))
THISFORM.sex.Value      = .Fields("sex").Value
THISFORM.alamat.Value   = .Fields("alamat").Value
THISFORM.idjurusan.Value = .Fields("idJurusan").Value
THISFORM.NIK.Value      = .Fields("NIK").Value
THISFORM.txtRecCount.Value = .RecordCount
thisform.gridMahasiswa.OBJECT.DataSource=toRecordSet
thisform.gridMahasiswa.OBJECT.Refresh
ENDWITH

```

Prosedur Form.clearfields

```

WITH THISFORM
    .npm.Value      = ''
    .nama.Value     = ''
    .tmplahir.Value = ''
    .tglLahir.Value = ''
    .sex.Value      = ''
    .alamat.Value   = ''
    .sex.Value      = ''
    .idjurusan.Value = ''
    .NIK.Value      = ''
ENDWITH

```

Prosedur NPM.Valid

```

WITH thisform

```

```

IF .addmode=.T. THEN
.oRSMahasiswa.MoveFirst
.oRSMahasiswa.Find("NPM=" +
ALLTRIM(thisform.NPM.Value)+ "'")
    IF .NOT. EMPTY(This.Value) THEN
        IF .NOT. .oRSMahasiswa.EOF THEN
            =MESSAGEBOX("NPM Tersebut sudah
dientry",0+64,"Info")
            RETURN 0
        ENDIF
    ENDIF
ENDIF
ENDWITH

```

Prosedur cmdPrev.Click

```

WITH THISFORM
    IF .oRSMahasiswa.RecordCount > 0

        .oRSMahasiswa.MovePrevious
        IF .oRSMahasiswa.BOF
            MessageBox("Start of File",0+64,"BOF")
            .oRSMahasiswa.MoveFirst
        ELSE
            =THISFORM.DisplayFields( .oRSMahasiswa)
        ENDIF
    ENDIF
ENDWITH

```

Prosedur cmdNext.Click

```

WITH THISFORM

```

```

        IF .oRSMahasiswa.RecordCount > 0

            .oRSMahasiswa.MoveNext
            IF .oRSMahasiswa.EOF
                MessageBox("End of File",0,"EOF")
                .oRSMahasiswa.MoveLast
            ELSE
                =THISFORM.DisplayFields( .oRSMahasiswa )
            ENDIF
        ENDIF
    ENDWITH

```

Prosedur cmdTop.Click

```

WITH THISFORM
    IF .oRSMahasiswa.RecordCount > 0

        .oRSMahasiswa.MoveFirst
        =THISFORM.DisplayFields( .oRSMahasiswa )
    ENDIF
ENDWITH

```

Prosedur cmdBottom.Click

```

WITH THISFORM
    IF .oRSMahasiswa.RecordCount > 0

        .oRSMahasiswa.MoveLast
        =THISFORM.DisplayFields( .oRSMahasiswa )
    ENDIF
ENDWITH

```


Prosedur cmdAdd.Click

```
WITH THISFORM
    IF THIS.Caption = "Add"
        *-- Enable Fields.
        STORE .T. TO .npm.enabled, ;
            .nama.enabled, ;
            .tmplahir.enabled, ;
            .tgllahir.enabled, ;
            .sex.enabled, ;
            .alamat.enabled, ;
            .idjurusan.enabled, ;
            .NIK.enabled
        STORE .F. TO .cmdPrev.Enabled , ;
            .cmdNext.Enabled, ;
            .cmdDelete.Enabled, ;
            .cmdTop.Enabled, ;
            .cmdBottom.Enabled

        THIS.Caption      = "Save"
        .cmdEdit.Caption = "Undo"
        .AddMode = .T.

        .ClearFields()
        .npm.SetFocus()

    ELSE && --- Save Mode.....
        IF EMPTY(.NPM.Value)
            =MessageBox("NPM Masih
Kosong", 0+64, "Info")
            .NPM.SetFocus()
        RETURN
```

```

ENDIF
IF EMPTY(.Nama.Value)
    =MessageBox("Nama Masih
Kosong",0+64,"Info")
    .Nama.SetFocus()
    RETURN
ENDIF
IF EMPTY(.IDJurusan.Value)
    =MessageBox("Jurusan Masih
Kosong",0+64,"Info")
    .IDJurusan.SetFocus()
    RETURN
ENDIF
IF EMPTY(.NIK.Value)
    =MessageBox("NIK Pembimbing Akademik
Masih Kosong",0+64,"Info")
    .NIK.SetFocus()
    RETURN
ENDIF

IF .AddMode
    .oRSMahasiswa.AddNew
ENDIF

    .oRSMahasiswa.Fields("npm").Value      = ;
ALLTRIM(PROPER(.npm.Value))
    .oRSMahasiswa.Fields("nama").Value      = ;
ALLTRIM(.nama.Value)
    .oRSMahasiswa.Fields("tmplahir").Value = ;
ALLTRIM(.tmplahir.Value)

```

```

        .oRSMahasiswa.Fields("tglLahir").Value = ;
ALLTRIM(.tglLahir.Value)
        .oRSMahasiswa.Fields("sex").Value      = ;
ALLTRIM(.sex.Value)
        .oRSMahasiswa.Fields("alamat").Value    = ;
ALLTRIM(.alamat.Value)
        .oRSMahasiswa.Fields("idjurusan").Value= ;
LEFT(ALLTRIM(.idjurusan.Text),2)
        .oRSMahasiswa.Fields("NIK").Value= ;
LEFT(ALLTRIM(.NIK.Text),9)

.oRSMahasiswa.Update
thisform.oRSMahasiswa.REQUERY
thisform.gridMahasiswa.Refresh
*-- Disable Fields.
STORE .F. TO .npm.enabled,;
        .nama.enabled,;
        .tmplahir.enabled,;
        .tglLahir.enabled,;
        .sex.enabled,;
        .alamat.enabled,;
        .idjurusan.enabled,;
        .NIK.enabled
STORE .T. TO .cmdPrev.Enabled ,;
        .cmdNext.Enabled,;
        .cmdDelete.Enabled,;
        .cmdTop.Enabled,;
        .cmdBottom.Enabled

.DisplayFields( .oRSMahasiswa )
THIS.Caption      = "Add"

```

```

        .cmdEdit.Caption    = "Edit"
        .AddMode             = .F.
    ENDIF
ENDWITH

```

Prosedur cmdEdit.Click

```

WITH THISFORM
    IF THIS.Caption = "Edit"
        *-- Enable Fields.
        STORE .T. TO .npm.enabled, ;
                                .nama.enabled, ;
                                .tmplahir.enabled, ;
                                .tglLahir.enabled, ;
                                .sex.enabled, ;
                                .alamat.enabled, ;
                                .NIK.enabled, ;
                                .idjurusan.enabled
        STORE .F. TO .cmdPrev.Enabled , ;
                                .cmdNext.Enabled, ;
                                .cmdDelete.Enabled, ;
                                .cmdTop.Enabled, ;
                                .cmdBottom.Enabled

        THIS.Caption    = "Undo"
        .cmdAdd.Caption = "Save"
        .AddMode        = .F.
        .npm.SetFocus()

    ELSE && --- Undo Mode
        .DisplayFields( .oRSMahasiswa )
    ENDIF
ENDWITH

```

```

        *-- Disable Fields.
        STORE .F. TO .npm.enabled,;
                .nama.enabled,;
                .tmplahir.enabled,;
                .tglLahir.enabled,;
                .sex.enabled,;
                .alamat.enabled,;
                .NIK.enabled,;
                .idjurusan.enabled
        STORE .T. TO .cmdPrev.Enabled ,;
                .cmdNext.Enabled,;
                .cmdDelete.Enabled,;
                .cmdTop.Enabled,;
                .cmdBottom.Enabled
        .DisplayFields( .oRSMahasiswa )
        THIS.Caption      = "Edit"
        .cmdAdd.Caption    = "Add"
        .AddMode           = .F.
    ENDIF
ENDWITH

```

Prosedur cmdDelete.Click

```

LOCAL lcMess
WITH THISFORM
    IF .oRSMahasiswa.RecordCount > 0

        lcMess = "Akan menghapus data?" + CHR(13) + ;

        ALLTRIM(.oRSMahasiswa.Fields("NPM").Value)
        IF MESSAGEBOX(lcMess,36+256,"Delete
Record")=6
    
```

```

        .oRSMahasiswa.Delete
        .oRSMahasiswa.Requery
        .DisplayFields(.oRSMahasiswa)
    ENDIF
ENDIF
ENDWITH

```

Prosedur cmdClose.Click

```
THISFORM.Release()
```

5. Jalankan form yang sudah didesain, sehingga terbentuk aplikasi sebagai berikut :

NPM	Nama	tmpLahir	tglLahir	sex	Alamat
03030303	Siti Hardiyanti	Lampung	2/17/1980	L	Metro Lampung Tengah
03030304	Miftakh	Lampung	2/11/2006	P	Lampung
03030080	Budiyanto Sinaga	Medan	5/15/1985	L	Jl. Tongkol 27 Me
03030097	Uswatun Hasanah	Liwa	12/17/1984	P	Jl. Linsum 48 Lar
03030158	Dian Novitasari	PringSewu	8/26/1984	P	Jl. Antasari 28 A F
03030281	Endah Langgeng	Metro	8/24/1985	P	Jl. Mujair 25 Metro
03030333	Andri Andayani	Kalirejo	5/18/1983	L	Jl. Lembayung N

Gambar 10.44 Hasil Running Form.

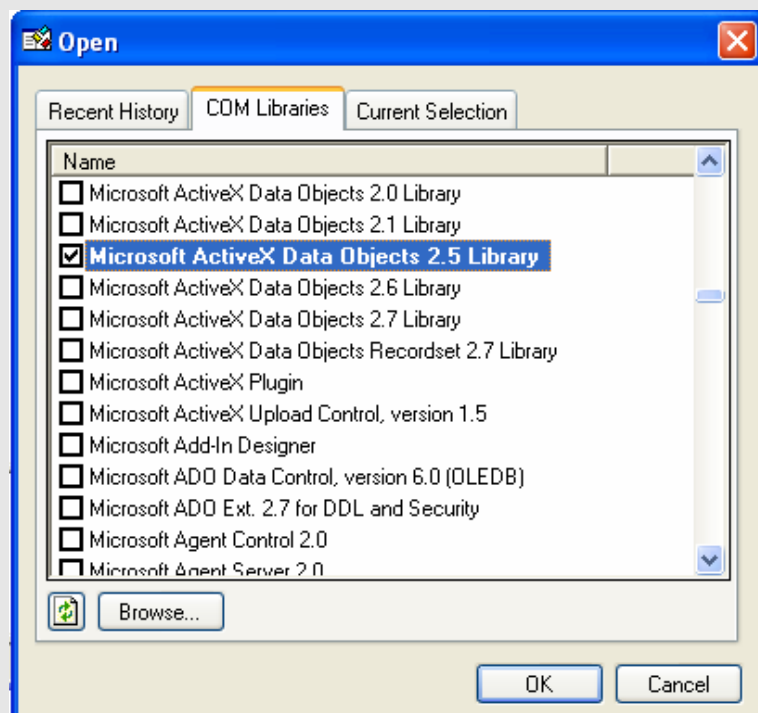
Penjelasan :

Di awal program ditambahkan beberapa konstanta dari objek ADO, dengan nilai yang diberikan kepada masing-masing konstanta. Informasi mengenai nama konstanta objek ADO beserta nilainya diperoleh dari objek browser objek ADO. Anda dapat memilih menu **Tools**→pilih **Object Browser**→kemudian akan ditampilkan window kosong **Object Browser**. Selanjutnya pilih icon **Open Type Library** yang terletak tepat di sebelah pojok kiri window **Object Browser**. Yang dapat anda lihat seperti tampilan berikut :



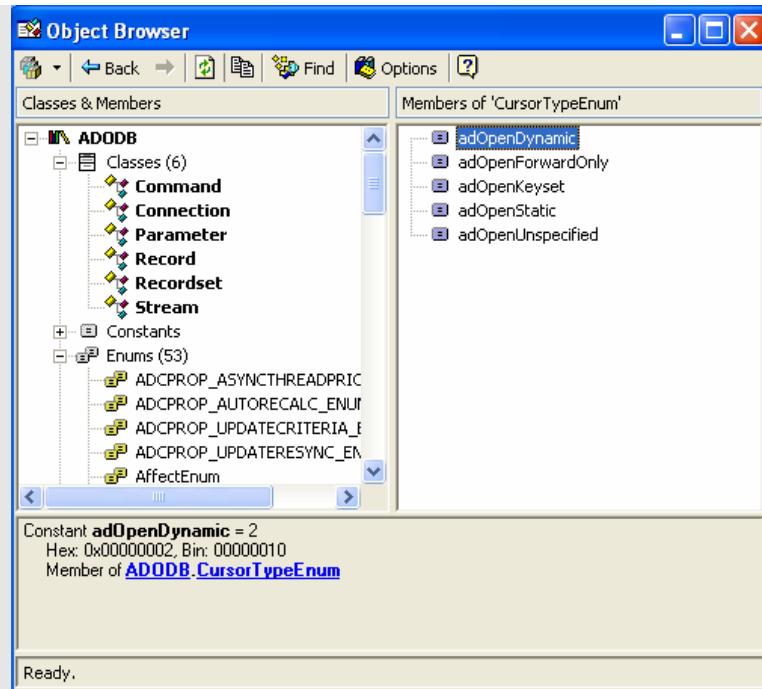
Gambar 10.45 Icon Open Type Library

Selanjutnya pilih Tab **Com Libraries**, di sini akan ditampilkan daftar *Component Object Model* (COM) yang dapat anda tambahkan ke aplikasi anda. Selanjutnya karena kita menggunakan Objek Library ADO, maka anda dapat memilih pilihan **Microsoft ActiveX Data Objects 2.5 Library**. Selanjutnya klik OK.



Gambar 10.46 Menambahkan Library ADO

Maka dalam Object Browser anda kini telah terdapat sebuah COM ADODB, berikut objek apa saja yang terdapat didalamnya. Di bagian bawah anda akan menemukan informasi tentang detail konstanta berikut nilainya dari objek-objek ADO. Seperti gambar di berikut nampak bahwa objek Recordset mempunyai tipe cursor dengan nama konstanta **adOpenDynamic** dengan nilai konstantanya adalah 2. Selanjutnya anda dapat melihat objek-objek ADO yang lain.



Gambar 10.47 Objek ADO beserta nilai-nilai konstantanya

1.9 APLIKASI MENGGUNAKAN ADO-SQL

Dalam aplikasi berikut akan disajikan bagaimana cara membangun Aplikasi menggunakan ADO dan perintah SQL.

Untuk aplikasi yang sederhana mungkin anda cukup membuatnya dengan metode-metode yang disediakan objek ADO-Recordset, seperti pada pembahasan sebelumnya. Namun bukan pilihan yang tepat, jika anda membangun aplikasi yang mengelola data cukup besar dan memerlukan pemrosesan data yang cukup kompleks. Di sini anda dapat menggunakan perintah-perintah SQL untuk dikolaborasikan bekerja dengan objek ADO. Untuk membuatnya ikuti langkah pembuatannya sebagai berikut :

1. Desain Form sebagai berikut :

Gambar 10.48 Desain Form Aplikasi ADO-SQL

Nama Objek	Properties	Value
Form1	Name Caption MaxButton AutoCenter	FrmADO :::Entry Data Dosen .F. .T.
Textbox1	Name	NIK
Textbox2	Name	Nama
Textbox3	Name	Nama
Textbox4	Name	Nama
Combo1	Name	Nama
Textbox5	Name	Nama
Textbox6	Name	Nama
Command1	Name Caption	CmdSimpan Simpan
Command2	Name Caption	cmdHapus Hapus
Command3	Name Caption	cmdKosong Kosong
Command4	Name Caption	cmdSelesai Selesai
DataGrid1	Name	oleDataGrid

- Selanjutnya tambahkan kode program untuk Event dan Metode sebagai berikut :

Isi Procedure Form.Init

```

LOCAL strKoneksi as String
PUBLIC ModeEditing, strSQL, strCari As String

thisform.addproperty('Koneksi',CREATEOBJECT('ADODB.C
onnection'))
thisform.addproperty('rsDosen',CREATEOBJECT('ADODB.R
ecordset'))

strKoneksi = "PROVIDER=SQLOLEDB.1; " ;
+"SERVER=SERVER; ";
+"DATABASE=DBSiakad; ";
+"UID=sa; ";
+"PWD=miftakh"

thisform.koneksi.ConnectionString=strKoneksi
thisform.koneksi.open()
if thisform.koneksi.state=1 then
    thisform.rsDosen.CursorLocation=3
ELSE
    MESSAGEBOX("Koneksi ke Server
Gagal",0+64,"informasi")
    RETURN
ENDIF

thisform.tampildata
thisform.sex.AddItem ("L")
thisform.sex.AddItem ("P")
thisform.sex.ListIndex=0

```

Isi Procedure Form.Destroy

```
RELEASE strKoneksi, ModeEditing, strSQL, strCari
IF thisform.Koneksi.State=1 THEN
    thisform.Koneksi.Close
ENDIF
IF thisform.rsDosen.State=1 THEN
    thisform.rsDosen.Close
ENDIF
```

Isi Procedure Form.TampilData

```
IF thisform.rsDosen.State=1 THEN
    thisform.rsDosen.Close
ENDIF

thisform.rsDosen.open("SELECT * FROM
rDosen",thisform.Koneksi,3,3,1)
thisform.oleDatagrid.DataSource=thisform.rsDosen.data
source
thisform.oleDataGrid.Refresh
```

Isi Procedure NIK.Valid

```
IF .NOT. EMPTY(this.Value) THEN
    strCari="NIK=" + ALLTRIM(thisform.NIK.Value)

    thisform.rsDosen.MoveFirst
    thisform.rsDosen.Find = strCari

    IF thisform.rsDosen.EOF() THEN
        ModeEditing="INSERT"
    ELSE
        ModeEditing="UPDATE"
        thisform.nama.Value=;
```

```

        thisform.rsDosen.Fields("nama").value + ""
        thisform.tmpLahir.Value=;
        thisform.rsDosen.Fields("tmpLahir").value +
""

        thisform.tglLahir.Value=;

        TRANSFORM(thisform.rsDosen.Fields("tglLahir").valu
e, "99/99/9999")

        thisform.Sex.value=;
        thisform.rsDosen.Fields("sex").value + ""
        thisform.alamat.Value=;
        thisform.rsDosen.Fields("alamat").value + ""
        thisform.noTelp.Value=;
        thisform.rsDosen.Fields("noTelp").value + ""

    ENDIF
ENDIF

```

Isi Procedure cmdSimpan.Click

```

IF EMPTY(thisform.nik.Value) THEN
    MESSAGEBOX("NIK masih Kosong",0+64,"Konfirmasi")
    thisform.nik.SetFocus
    RETURN
ENDIF

IF EMPTY(thisform.nama.Value) THEN
    MESSAGEBOX("Nama Dosen masih
Kosong",0+64,"Konfirmasi")
    thisform.nama.SetFocus
    RETURN
ENDIF

IF ModeEditing="INSERT" THEN
    strSQL="INSERT INTO rDosen " ;

```

```

+ "(NIK,Nama,TmpLahir,TglLahir,Sex,Alamat,NoTelp) "
;

+ "VALUES(" + ALLTRIM(thisform.nik.Value) ;
+ ", '" + ALLTRIM(thisform.nama.Value) ;
+ "', '" + ALLTRIM(thisform.TmpLahir.Value) ;
+ "', '" + ALLTRIM(thisform.TglLahir.Value) ;
+ "', '" + ALLTRIM(thisform.Sex.Value) ;
+ "', '" + ALLTRIM(thisform.Alatamat.Value) ;
+ "', '" + ALLTRIM(thisform.NoTelp.Value) + "'"
thisform.koneksi.Execute(strSQL)
thisform.tampildata
thisform.cmdKosong.Click
ELSE
    IF ModeEditing="UPDATE" then
        strSQL="UPDATE rDosen " ;
        + "SET Nama='" + ALLTRIM(thisform>Nama.Value)
;
        + "', TmpLahir='" +
ALLTRIM(thisform.TmpLahir.Value) ;
        + "', TglLahir='" +
ALLTRIM(thisform.TglLahir.Value) ;
        + "', Sex='" + ALLTRIM(thisform.sex.Value) ;
        + "', Alamat='" +
ALLTRIM(thisform.Alatamat.Value) ;
        + "', NoTelp='" +
ALLTRIM(thisform.NoTelp.Value) ;
        + "' WHERE NIK=" +
ALLTRIM(thisform.NIK.Value)
        thisform.koneksi.Execute(strSQL)
        thisform.tampildata
        thisform.cmdKosong.Click

```

```
ELSE
    RETURN
ENDIF
ENDIF
```

Isi Procedure cmdHapus.Click

```
IF EMPTY(thisform.nik.Value) THEN
    MESSAGEBOX("Masukkan NIK yang akan
dihapus",0+64,"Info")
    thisform.nik.SetFocus
    RETURN
ENDIF
IF MESSAGEBOX("Data Aakan
dihapus?",4+32,"Konfirmasi")=6 THEN
    strSQL="DELETE FROM rDosen WHERE NIK=" +
ALLTRIM(thisform.NIK.Value)
    thisform.Koneksi.Execute(strSQL)
    thisform.tampildata
    thisform.cmdKosong.Click
ENDIF
```

Isi Procedure cmdKosong.Click

```
thisform.nik.Value=''
thisform.nama.Value=''
thisform.tmpLahir.Value=''
thisform.tglLahir.Value=''
thisform.sex.Value=''
thisform.alamat.Value=''
thisform.noTelp.Value=''
thisform.nik.SetFocus
```

Isi Procedure cmdSelesai.Click

thisform.Release

3. Jalankan aplikasi sehingga akan menghasilkan tampilan program sebagai berikut :

NIK	Nama	tmpLahir	tglLahir	sex	alamat	NoTel
080505051	M. Anwar Sanusi, S. Metro		1/5/1975	L	Jl. Yulius Usma No.	08157
080505052	Siti Anun Nafsiyah, Dr Tanjung Karang		5/23/1977	P	Jl. Pringgodani 65	08132
080505053	Kristin Anunsiyah, Dr Solo		8/19/1980	P	Jl. Tongkol 28 A Solo	08562
080505054	M. Miftakul Amin, S. K Metro		12/17/1980	L	Jl. Yos Sudarso 28 A	08157
080505055	Masnur Analisa, S. S Palembang		2/5/1976	L	Jl. Intan 27 A BDL	08125
080505056	Kuala Siregar, Dr., M Medan		2/26/1973	L	Jl. Antasari 25 B BDL	08884

Gambar 10.46 Hasil Running Aplikasi ADO-SQL

1. 10 MENGOLAH FIELD IMAGE DENGAN ADO-STREAM

Salah satu kelebihan dari MS. SQL Server adalah disediakannya field dengan tipe Image. Mungkin di beberapa database server yang lain anda pernah menemukan field yang serupa seperti di MySQL misalnya menyediakan tipe data Binary Large Object (BLOB). Tipe data ini dipergunakan untuk menyimpan file binary seperti misalnya image atau teks yang berukuran besar yang dapat anda manipulasi seperti halnya tipe data yang lain.

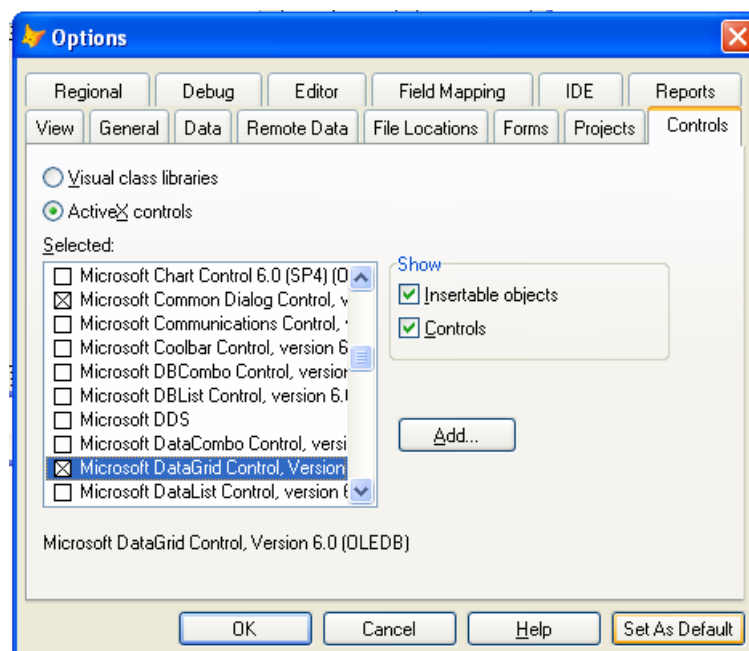
Salah satu alasan mengapa anda menggunakan ADO yang dibahas dalam bab ini sebagai alternatif akses data, selain Remote View dan SPT, ini adalah juga salah satu jawabannya. Karena RV dan SPT tidak menyediakan mekanisme secara khusus penanganan terhadap field dengan tipe data Image. Jika anda pernah membuat aplikasi menggunakan VFP untuk mengolah data dengan tipe general, aplikasi yang akan anda bangun ini juga sama seperti anda mengolah tipe data general tersebut. Jika anda mengolah data di VFP dengan perintah APPEND GENERAL FROM, dengan menggunakan ADO anda dapat menggunakan objek Stream. Sejak dikeluarnya ADO 2.5, objek ADO ditambah satu lagi yaitu objek Stream.

Ketika anda membangun aplikasi untuk menyimpan biodata mahasiswa, karyawan, nasabah, dan data lain yang membutuhkan informasi untuk menyimpan data gambar/foto misalnya, maka anda dapat langsung menyimpannya ke dalam database. Data yang disimpan nantinya akan berupa file binary.

Aplikasi berikut memberikan contoh bagaimana mengolah data foto dosen yang disimpan di table dengan tipe data Image. Aplikasi berikut dipergunakan untuk memasukkan data foto dosen ke dalam table, kemudian menampilkannya lewat laporan yang dibuat dengan menggunakan Crystal Report. Program ini adalah aplikasi Edit data yang hanya khusus dibuat untuk memasukkan foto saja.

Untuk membuatnya terlebih dahulu tambahkan 3 buah control .OCX ke dalam aplikasi anda yaitu **Microsoft DataGrid Control, Version 6.0 (OLEDB)**, **Microsoft Common Dialog Control Version 6.0**, serta **Crystal Report Control**. Selanjutnya ikuti langkah-langkah sebagai berikut :

1. Tambahkan 3 buah komponen .OCX ke dalam aplikasi anda, yaitu Microsoft DataGrid Control, Version 6.0(OLEDB), Microsoft Common Dialog Control version 6.0, serta Crystal Report Control. Dari menu Tolls→pilih Options...→pada window Options→pilih Tab Controls. Kemudian pada List ActiveX Controls pilih .OCX yang akan ditambahkan. Kemudian klik OK. Selanjutnya pastikan Form Controls Tollbars anda kini telah ditambahkan 3 kontrol .OCX di atas. Kontrol DataGrid diperlukan untuk menampilkan data table yang diambil objek ADO-recordset. Kemudian objek Common Dialog dipergunakan untuk membuka file foto yang tersimpan dalam folder di komputer anda. Sedangkan Crystal Report Control dipergunakan untuk memanggil laporan yang telah didesain menggunakan Seagate Crystal Report.



Gambar 10.47 Menambahkan ActiveX Control



Gambar 10.48 Form Control Toolbars Setelah ditambahkan File .OCx

2. Desain form seperti berikut :

Gambar 10.49 Desain Form Mengolah Data Gambar

Nama Objek	Properties	Value
Form1	Name Caption	FrmADO ...:ImageProcessing
Combo1	Name	NIK
Textbox2	Name	Nama
Command1	Name Caption	cmdEntryFoto Entry Foto
Command2	Name Caption	cmdCetak Cetak
Command3	Name Caption	cmdHapus Hapus Foto
Command4	Name Caption	cmdSelesai Selesai
DataGrid1	Name	oleDataGrid
Dialog1	Name	oleDialog
CrasytalReport1	Name	oleCR
Image1	Name	Foto

3. Selanjutnya tambahkan kode program berikut sesuai dengan event dan metode yang ada.

Isi Procedure Form.Init

```
LOCAL strKoneksi as String

*---- CursorTypeEnum Values ----
#define ADOPENFORWARDONLY      0
#define ADOPENKEYSET           1
#define ADOPENDYNAMIC          2
#define ADOPENSTATIC           3

*---- LockTypeEnum Values ----
#define ADLOCKREADONLY         1
#define ADLOCKPESSIMISTIC      2
#define ADLOCKOPTIMISTIC       3
#define ADLOCKBATCHOPTIMISTIC  4

*---- CursorLocationEnum Values ----
#define ADUSESERVER            2
#define ADUSECLIENT            3

thisform.addproperty('Koneksi',CREATEOBJECT('ADODB.C
onnection'))
thisform.addproperty('rsDosen',CREATEOBJECT('ADODB.R
ecordset'))
thisform.addproperty('StreamIn',CREATEOBJECT('ADODB.
Stream'))
thisform.addproperty('StreamOut',CREATEOBJECT('ADODB
.Stream'))
```

```

strKoneksi = "DATABASE=DBSiakad; " ;
+"SERVER=SERVER; ";
+"UID=sa; ";
+"PWD=miftakh; ";
+"PROVIDER=SQLOLEDB.1"

thisform.koneksi.ConnectionString=strKoneksi
thisform.koneksi.open()
if thisform.koneksi.state=1 then
    thisform.rsDosen.ActiveConnection=thisform.koneksi
    thisform.rsDosen.CursorType=adOpenStatic
    thisform.rsDosen.LockType=adLockOptimistic
    thisform.rsDosen.CursorLocation=adUseClient
ELSE
    MESSAGEBOX("Koneksi ke Server
Gagal",0+64,"informasi")
ENDIF

thisform.rsDosen.open("SELECT * FROM rDosen")
thisform.oleDatagrid.DataSource=thisform.rsDosen.data
source

DO WHILE .NOT. thisform.rsDosen.eof
    thisform.nik.AddItem
(thisform.rsDosen.fields("NIK").value)
    thisform.rsDosen.movenext
ENDDO

```

Isi Procedure NIK.Click

```

*---- StreamTypeEnum Values ----
#define adTypeBinary 1

```

```

#DEFINE adTypeText          2

*---- SaveOptionsEnum Values ----
#DEFINE adSaveNotExist      1
#DEFINE adSaveOverWrite     2

LOCAL strCari as String
strCari="NIK=" + ALLTRIM(thisform.nik.Value)

thisform.rsdozen.movefirst
thisform.rsdozen.find = strCari

IF .NOT. thisform.rsdozen.eof then
    thisform.nama.Value=thisform.rsdozen.fields("Nama"
).value
    thisform.streamOut.Type=adTypeBinary
    thisform.streamOut.Open
    IF ISNULL(thisform.rsDosen.fields("Foto").value)
THEN
        thisform.foto.Picture=""
    ELSE
        thisform.streamOut.write(thisform.rsdozen.fields("
Foto").value)
        thisform.streamOut.SaveToFile ("C:\tmp.bmp",2)
        thisform.foto.Picture="C:\tmp.bmp"
    ENDIF
    thisform.streamOut.Close
ELSE
    MESSAGEBOX("Data tidak ada",0+64,"Konfirmasi")
ENDIF

```

Penjelasan :

Prosedur untuk menampilkan data foto cukup sederhana, anda hanya perlu melakukan hal-hal sebagai berikut :

- Membuka objek ADO-Stream

```
thisform.StreamIn.Open
```

- Mengambil data yang berasal dari field dan menuliskannya ke dalam sebuah file di computer anda.

```
thisform.streamOut.write(thisform.rsdosen.fields("Foto").value)
thisform.streamOut.SaveToFile ("C:\tmp.bmp",2)
```

- Dan terakhir adalah menampilkan field foto tersebut.

```
thisform.foto.Picture="C:\tmp.bmp"
```

Isi Procedure cmdEntryFoto.Click

```
*---- StreamTypeEnum Values ----
#define adTypeBinary 1
#define adTypeText 2

*---- SaveOptionsEnum Values ----
#define adSaveNotExist 1
#define adSaveOverWrite 2
LOCAL varian as Variant
IF EMPTY(thisform.nik.Value) THEN
    MESSAGEBOX("Pilih NIK yang akan dientry
Fotonya",0+64,"Info")
ELSE
    thisform.oleDialog.Filter = "Bitmaps
(*.bmp)|*.bmp|Jpegs(*.jpg)|*.jpg|Gif(*.gif)|*.gif"
    thisform.oleDialog.showOpen
    varian=thisform.oleDialog.FileName
    IF .NOT. EMPTY(varian) THEN
        thisform.StreamIn.Type=adTypeBinary
        thisform.StreamIn.Open
        thisform.StreamIn.LoadFromFile(varian)
```

```

        thisform.rsDosen.fields("Foto").value =
thisform.StreamIn.read
        thisform.rsDosen.update
        thisform.StreamIn.Write(thisform.rsDosen.fields("F
oto").value)
        thisform.StreamIn.savetofile("C:\tmp.bmp",2)
        thisform.foto.Picture="C:\tmp.bmp"
        thisform.StreamIn.Close
    ENDIF
ENDIF

```

Penjelasan :

Prosedur untuk memasukkan data foto cukup sederhana, anda hanya perlu melakukan hal-hal sebagai berikut :

- Membuka objek ADO-Stream

```

        thisform.StreamIn.Open

```
- Mengambil data yang berasal dari file

```

        thisform.StreamIn.LoadFromFile(varian)

```
- Mengisi data field foto dengan membaca isi file ADO-stream

```

        thisform.rsDosen.fields("Foto").value =
thisform.StreamIn.read

```
- Dan terakhir adalah mengupdate hasil pengisian field foto tersebut.

```

        thisform.rsDosen.update

```

Isi Procedure cmdCetak.Click

```

thisform.oleCR.ReportFileName="D:\Siakad\Laporan\rpt
InfoDosen.rpt"
thisform.oleCR.windowState= 2  && crptMaximized
thisform.oleCR.destination= 0  && crptToWindow
thisform.oleCR.Action=1

```

Penjelasan :

Untuk property ReportFileName, sesuaikan dengan alamat file dimana anda menyimpan file laporan anda. Selanjutnya property windowstate akan membuat laporan anda memenuhi layar monitor ketika laporan dijalankan. Property destination menunjukkan tujuan laporan, dalam hal ini kita memilih crptToWindow agar laporan cukup ditampilkan di layar monitor. Dan property Action mengisntruksikan kepada control Crystal Report untuk menjalankan laporan.

Isi Procedure cmdHapus.Click

```
IF MESSAGEBOX("Kosongkan Foto ?",4+32,"Konfirmasi")
=6 THEN
    thisform.rsDosen.fields("Foto").value=NULL
    thisform.rsDosen.update
    thisform.foto.Picture=""
ENDIF
```

Penjelasan :

Prosedur ini hanya dipergunakan untuk mengosongkan isi field Foto. Program akan meminta konfirmasi user apakah akan mengosongkan field foto, field akan dikosongkan jika user memilih tombol yes pada messagebox.

Isi Procedure Form.Destroy

```
IF thisform.Koneksi.State=1 THEN
    thisform.Koneksi.Close
ENDIF

IF thisform.rsDosen.State=1 THEN
    thisform.rsDosen.Close
ENDIF

IF thisform.StreamIn.State=1 THEN
    thisform.StreamIn.Close
ENDIF

IF thisform.StreamOut.State=1 THEN
    thisform.StreamOut.Close
ENDIF
```

Penjelasan :

Prosedur ini dipergunakan untuk menutup semua variable yang dibuat menggunakan objek ADO, ketika variable-variabel tersebut tidak dipergunakan lagi di aplikasi anda.

Isi Procedure cmdSelesai.Click

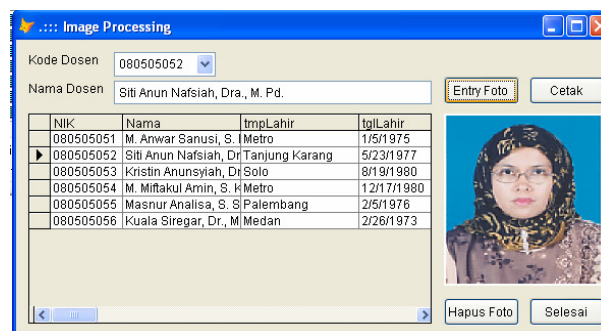
```
thisform.Release
```

4. Selanjutnya desain Laporan menggunakan Cryustal Report sebagai berikut, simpan nama laporan dengan rptInfoDosen.rpt :



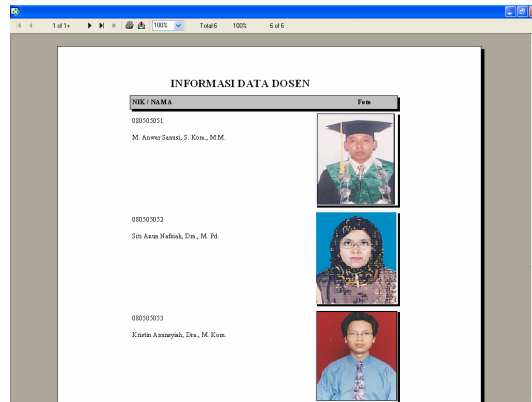
Gambar 10.50 Desain Laporan

5. Jalankan aplikasi sehingga diperoleh tampilan sebagai berikut, untuk mengganti isi field foto anda cukup memilih NIK dosen, kemudian pilih tombol Entry Foto untuk mengganti foto dosen sesuai NIK yang dipilih :



Gambar 10.51 Hasil Running Aplikasi

6. Selanjutnya untuk menampilkan data berupa laporan anda dapat mengklik tombol Cetak, sehingga akan ditampilkan hasil laporan sebagai berikut :



Gambar 10.52 Hasil Laporan

Sampai di sini anda telah mampu membuat aplikasi menggunakan ADO yang dapat anda jadikan alternatif akses data untuk aplikasi database anda selain teknik akses data yang disediakan oleh VFP yaitu Remote View dan SPT. Kemudahan yang ditawarkan ADO serta binding data yang kuat, merupakan daya tarik tersendiri sehingga ADO cukup terkenal untuk membangun aplikasi database.

BIOGRAFI PENULIS



Muhammad Miftakhul Amin dilahirkan di Metro, pada tanggal 17 desember 1980. Pendidikan Sekolah Dasar (SD. N. 1 Raman Aji) , Madrasah Tsanawiyah (MTs. N. 1 Raman Utara) sampai Madrasah Aliyah (MAN. 1 Metro) ditempuh di tanah kelahirannya. Kemudian penulis menyelesaikan pendidikan Strata-1 di jurusan Teknik Informatika, Fakultas Teknologi Informasi, Universitas Teknologi Yogyakarta (UTY), lulus tahun 2004. Penulis sekarang adalah Dosen Tetap di Program Studi Manajemen Informatika Perguruan Tinggi “**Darmajaya**” Bandar Lampung dan juga

menjabat sebagai Kepala UPT-Pengembangan Sistem Informasi (PSI) di perguruan tinggi yang sama.