



**MODUL PRAKTIKUM
PEMROGRAMAN KOMPUTER**

Oleh :

Cahyono Sigit Pramudyo, S.T.,M.T, D.Eng

**LABORATORIUM KOMPUTASI
JURUSAN TEKNIK INDUSTRI
FAKULTAS SAINS DAN TEKNOLOGI
UIN SUNAN KALIJAGA
YOGYAKARTA**

2017

DAFTAR ISI

	DAFTAR ISI	ii
1	PENGENALAN VISUAL BASIC 6	1
	1.1 Pendahuluan	1
	1.2 Struktur Aplikasi	1
	1.3 Aplikasi sederhana	2
	1.4 Aplikasi sederhana perhitungan	4
	1.5 Latihan soal	4
2	PERCABANGAN	5
	2.1 Jenis Percabangan	5
	2.2 Contoh kasus IF	5
	2.3 Contoh kasus Select Case	6
3	PERULANGAN	9
	3.1 Jenis Perulangan	9
	3.2 For ... Next	9
	3.3 While ... Wend	11
	3.4 Do....Loop	11
	3.4.1 Do...Loop...While	12
	3.4.2 Do...Loop...Until	12
4	ARRAY	14
	4.1 Deklarasi Array	14
	4.2 Contoh kasus	14
5	BASIS DATA	16
	5.1 Sistem Basis Data	16
	5.2 Membuat Basis Data Sederhana	16
	5.3 Menghubungkan VB 6 dengan basis data Microsoft Access 2007 (*.accdb)	16
	5.4 Menambahkan fungsi tambah, ubah dan hapus	22
6	PENANGANAN KESALAHAN	25
	6.1 Jenis Kesalahan	25
	6.2 Contoh Kasus	25
	DAFTAR PUSTAKA	28

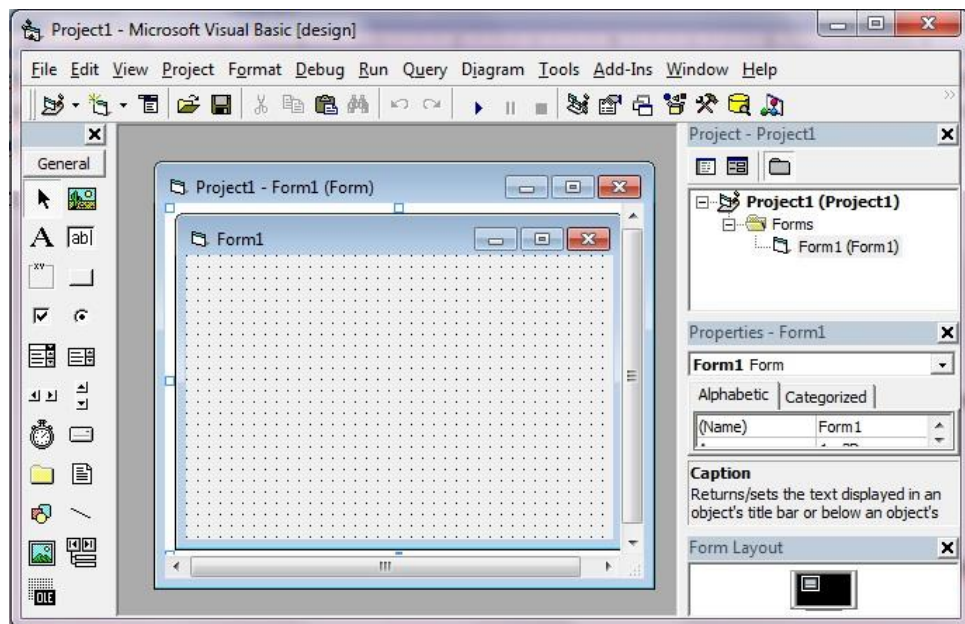
BAB 1

PENGENALAN VISUAL BASIC 6

1.1 Pendahuluan

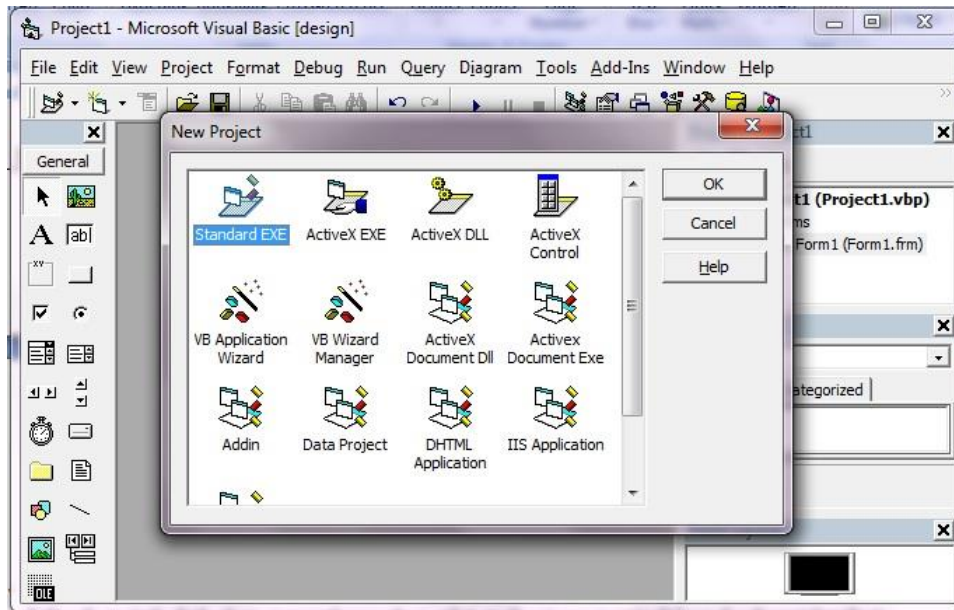
Mata kuliah pemrograman komputer mempelajari pemrograman (programming). Untuk meningkatkan kemampuan pemrograman harus mempelajari bahasa pemrograman (bahasa pengkodean). Praktikum pemrograman komputer lebih menekankan belajar bahasa pemrograman komputer. Pada modul praktikum pemrograman komputer akan digunakan Microsoft Visual Basic 6 sebagai alat untuk melakukan pembelajaran bahasa pemrograman komputer.

Microsoft Visual Basic adalah bahasa pemrograman untuk membuat aplikasi berbasis Microsoft Windows menggunakan GUI (Graphical User Interface). Bahasa pemrograman ini bekerja berdasarkan kejadian, hal ini dikenal dengan *event-driven programming*. *Event-driven programming* berarti program akan bekerja setelah ada *event* (kejadian) yang terdeteksi, misalnya : menekan tombol, memilih menu, dan lain-lain). Berikut adalah gambar halaman awal VB6.



1.2 Struktur Aplikasi

Pada awal pengembangan aplikasi VB, maka kita akan memerlukan lingkungan pengembangan. Sebagai awal kita menggunakan tipe proyek **Standard EXE**.



Sebuah proyek aplikasi VB terdiri dari :

- Form* – adalah jendela yang berfungsi sebagai tempat membuat *user interface*.
- Control* – tampilan berbasis grafis yang dimasukkan pada *form* untuk membuat interaksi dengan pemakai. Bagian ini dapat berupa text box, label, scroll bar, atau tombol command.

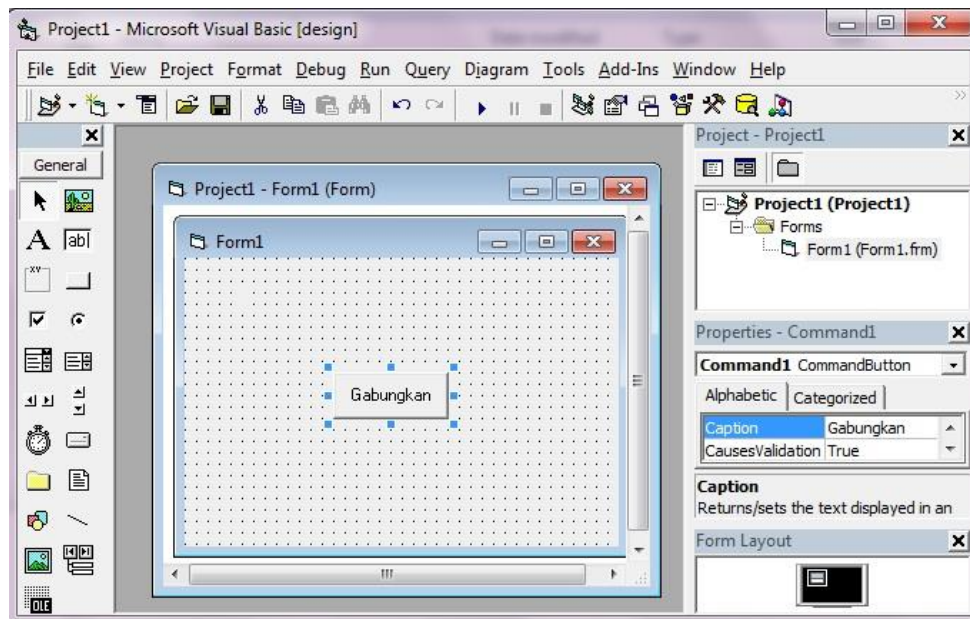
Form dan *control* dikelompok sebagai objek.

- Properties* adalah nilai atau karakter yang dimiliki oleh sebuah objek visual basic. Contohnya, Name, Captions, Size, Font, Position, dan lain-lain
- Methods* adalah serangkaian perintah yang tersedia pada suatu objek yang dapat diminta untuk mengerjakan suatu tugas. Misalnya : OnClick, dan lain-lain
- Event Procedure* adalah kode yang terhubung dengan suatu objek
- Module* adalah kumpulan prosedur umum, deklarasi variabel dan definisi konstanta yang digunakan oleh aplikasi..

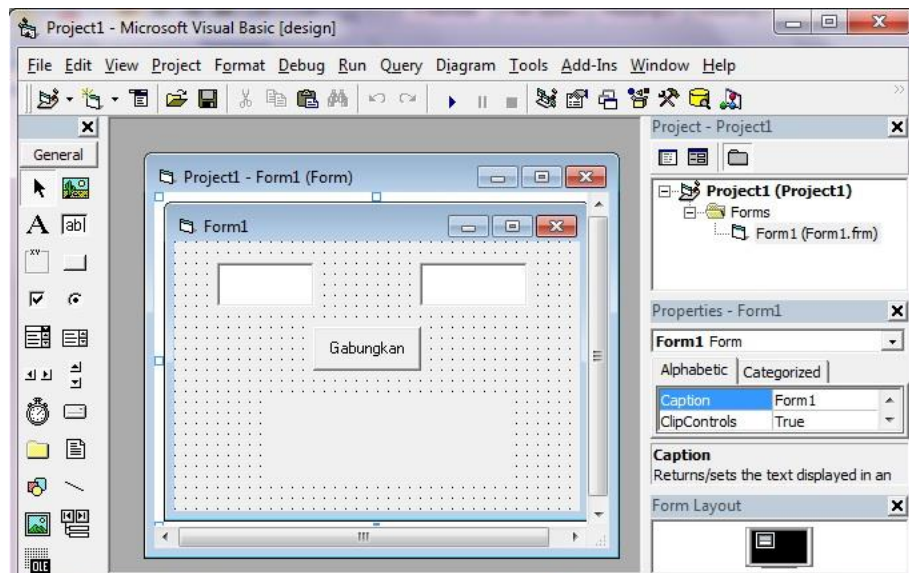
1.3 Aplikasi sederhana

Berikut ini adalah aplikasi sederhana dalam pemrograman visual basic. Kita akan membuat kita akan membuat aplikasi menggabungkan 2 kata.

- Kita memerlukan 1 control berupa tombol command yang akan melakukan proses perhitungan. Maka lakukan **Click and drag** command button dan ganti nama captions pada *properties* menjadi **Gabungkan**.



- b. Kita memerlukan 2 inputan dan 1 keluaran, maka click and drag 2 buah text box dan 1 label.



- c. Kemudian isi perintah pada tombol **Gabungkan** dengan cara mengkliknya. Tuliskan perintah berikut.

$$Label2.Caption = Text1.Text + Text2.Text$$

- d. Kemudian lakukan tekan F5 atau klik tombol start pada menu.



- e. Tulisan pada contoh tampak menempel. Modifikan agar bisa memiliki spasi dengan menambahkan tanda " " .

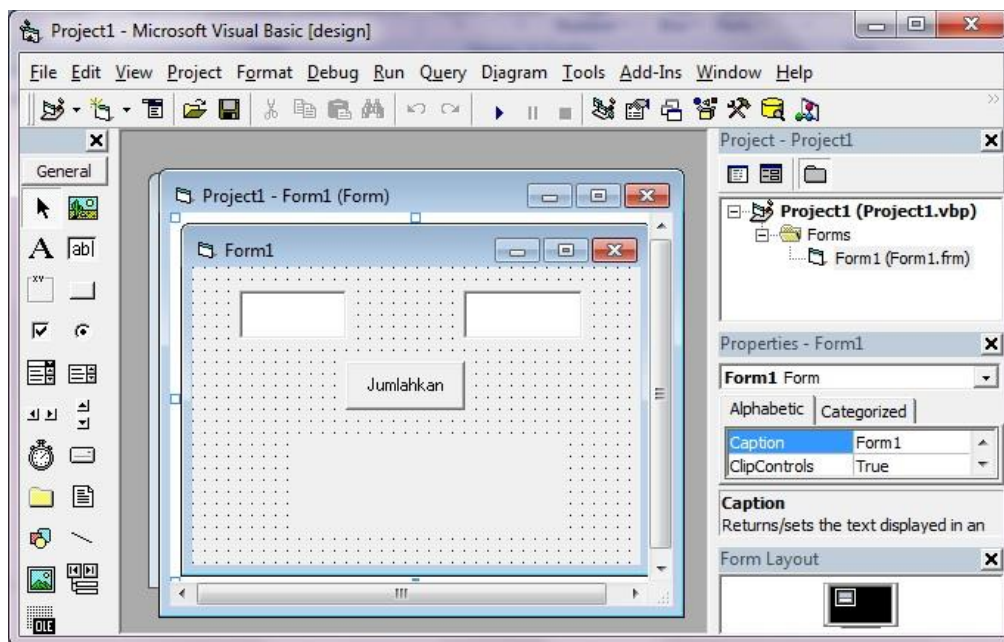
```
Label2.Caption = Text1.Text + " " + Text2.Text
```



- f. Demikian contoh sederhana program VB6.

1.4 Aplikasi sederhana perhitungan

Masih menggunakan aplikasi yang sama, tetapi nama tombol dari **Gabungkan** menjadi **Jumlahkan**.



Langkah berikutnya mengganti perintah pada tombol diganti menjadi sebagai berikut

```
Label2.Caption = Val(Text1.Text) + Val(Text2.Text)
```

Val berfungsi mengubah suatu tulisan menjadi angka

1.5 Latihan soal

Buatlah aplikasi kalkulator!

BAB 2

PERCABANGAN

2.1 Jenis Percabangan

Struktur percabangan berfungsi memerintahkan satu atau lebih perintah. Misalnya **If...Then...Else** atau **Select Case**. Struktur percabangan adalah sebagai berikut

```
If Kondisi 1 Then
Perintah1
Elseif kondisi2 Then:
Else
Perintah terakhir
End If
```

Atau

```
Select Case Kondisi
Case Ekspresi1
    Perintah1
Case Ekspresi2
    Perintah2
.....
Case Else
    PerintahTerakhir
End Case
```

2.2 Contoh kasus IF

Pengelompokan Nilai Angka menjadi Huruf.

Sebuah matakuliah akan mengelompokan nilai mahasiswa menjadi huruf sebagai berikut :

Nilai (X)	Nilai angka
$X \geq 80$	A
$70 \leq X < 80$	B
$60 \leq X < 70$	C
$X < 60$	D

Langkah – langkah :

- Buatlah **Standar EXE** baru
- Kita akan menggunakan 1 TextBox, 3 Label dan 1 command.

Properti :

Objek	Properti	Pengaturan
Form1	Caption	Konversi Nilai
Label1	Caption	Masukan Nilai Angka
Label2	Caption	Nilai Huruf Anda adalah
Label3	Caption	
	Alignment	Center
Command1	Caption	Konversikan Nilai
TextBox	Text	

- c. Klik **Command1**, kemudian tuliskan perintah sebagai berikut :

```
Private Sub Command1_Click()  
    bil = Val(Text1)  
    If bil >= 80 Then  
        Label3.Caption = "A"  
    ElseIf bil >= 70 Then  
        Label3.Caption = "B"  
    ElseIf bil >= 60 Then  
        Label3.Caption = "C"  
    Else  
        Label3.Caption = "D"  
    End If
```

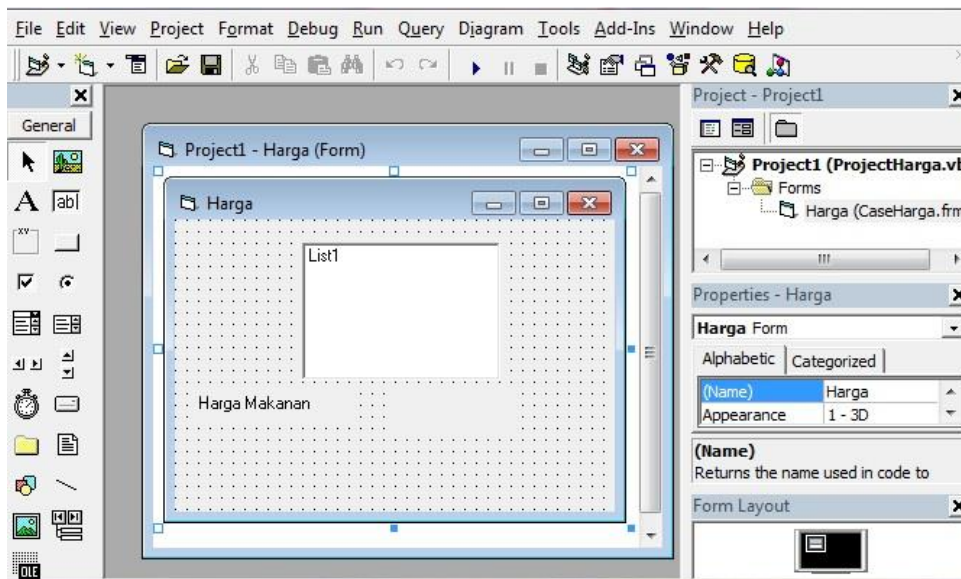
- d. Simpan dalam **ProjectKonversi**.
e. Ketika dijalankan, aplikasi akan bekerja sebagai berikut :



2.3 Contoh kasus Select Case

Langkah – langkah :

- Buatlah **Standar EXE** baru
- Kita akan menggunakan 1 ListBox, dan 2 Label sebagaimana tampilan berikut :



Properti

Objek	Properti	Pengaturan
Form1	(Name)	Harga
	Caption	Harga
Label1	Caption	Harga Makanan
Label2	Caption	

- c. Klik pada **Form**, kemudian tuliskan perintah sebagai berikut:

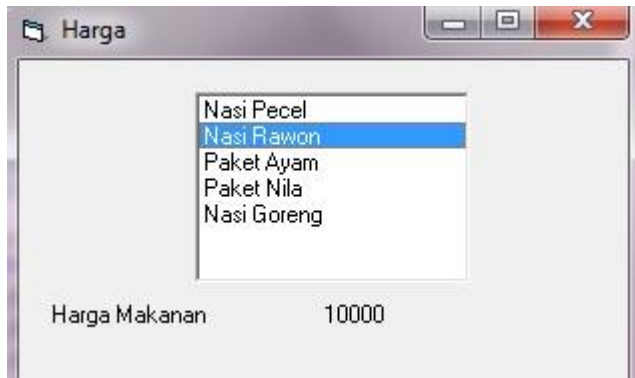
```
List1.AddItem "Nasi Pecel"
List1.AddItem "Nasi Rawon"
List1.AddItem "Paket Ayam"
List1.AddItem "Paket Nila"
List1.AddItem "Nasi Goreng"
```

- d. Klik **ListBox** kemudian tuliskan perintah sebagai berikut :

```
Dim Harga As Long
Select Case List1.ListIndex
    Case 0
        Harga = 5000
    Case 1
        Harga = 10000
    Case 2
        Harga = 12000
    Case 3
        Harga = 13000
    Case 4
        Harga = 10000
End Select
Label2.Caption = Harga
```

- e. Simpan dalam **ProjectSelect** dan form sebagai **CaseHarga**.

- f. Ketika dijalankan, aplikasi akan bekerja sebagai berikut :



BAB 3

PERULANGAN

3.1 Jenis Perulangan

Struktur perulangan berfungsi melakukan perintah berulang sampai jumlah yang ditentukan. Penghentian perulangan digunakan suatu kondisi tertentu.

3.2 For ... Next

Struktur perulangan ini digunakan bila kita sudah mengetahui jumlah perulangannya. Struktur perulangan ini menggunakan pola sebagai berikut :

For variabel_yang_diulang **To** nilai_akhir
 Perintah
Next

Contoh aplikasinya adalah menulis suatu kalimat dalam jumlah tertentu.



Langkah-langkah :

- a. Buatlah **Standar EXE** baru
- b. Kita menggunakan 1 ListBox, 2 label, dan 2 TextBox.

Properti :

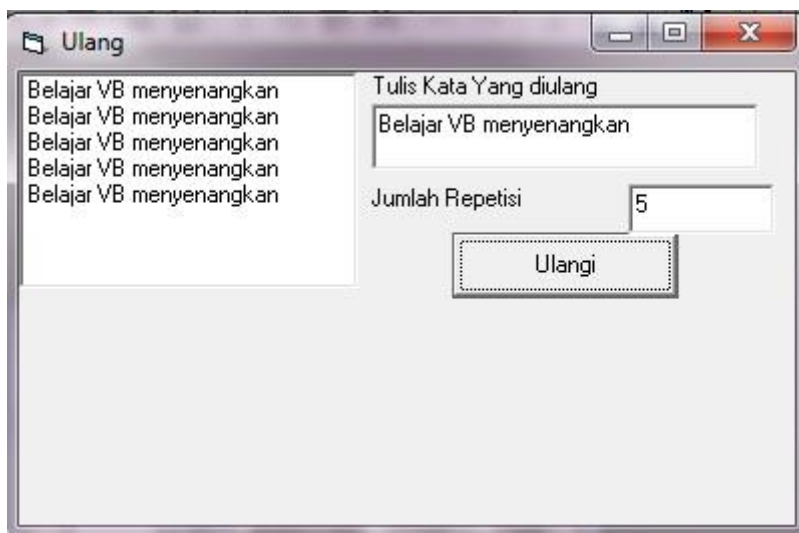
Objek	Properti	Pengaturan
Form1	Caption	Ulang
ListBox1	(Name)	List1
Label1	Caption	Tulis Kata Yang diulang
Label2	Caption	Jumlah Repetisi
TextBox	(Name)	Text1
TextBox	(Name)	Text2
CommandButton	(Name)	Command1

c. Klik tombol Command, tuliskan perintah sebagai berikut :

```
Private Sub Command1_Click()  
  
y = Val(Text2.Text)  
For i = 1 To y  
    List1.AddItem (Text1.Text)  
  
    Next  
  
End Sub
```

d. Simpan dalam **ProjectUlang** dan form sebagai **FrmUlang**

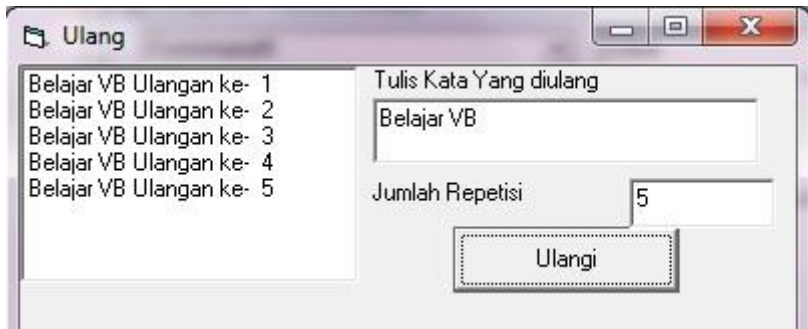
e. Ketika dijalankan, aplikasi akan bekerja sebagai berikut :



f. Dengan sedikit mengganti perintah pada tombol command kita dapat mengetahui urutan pengulangan. Perintah pada tombol command sebagai berikut :

```
Private Sub Command1_Click()  
    y = Val(Text2.Text)  
    For i = 1 To y  
        List1.AddItem Text1.Text + " Ulangan ke- " + Str(i)  
    Next  
End Sub
```

g. Hasil yang diperoleh



3.3 While ... Wend

Pengulangan menggunakan **While...Wend** digunakan untuk pengulangan terkendali selama kondisi yang diberikan masih memenuhi. Struktur perulangan ini menggunakan pola sebagai berikut :

While kondisi

Perintah selama kondisi terpenuhi

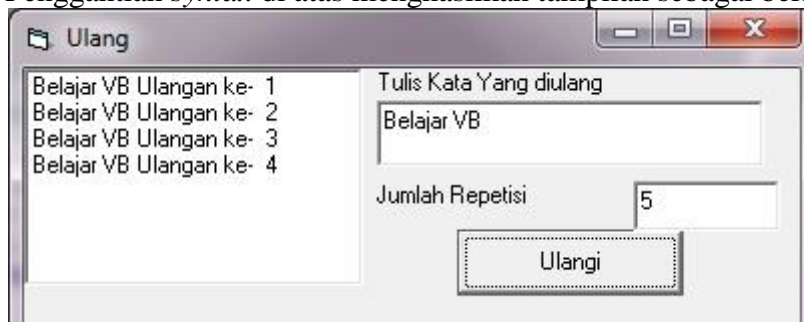
Incremental

Wend

Kita akan menggunakan file yang sama dengan mengganti syntax-nya pada **command** menjadi sebagai berikut :

```
Private Sub Command1_Click()
    y = Val(Text2.Text)
    i = 1
    While i < y
        List1.AddItem Text1.Text + " Ulangan ke- " + Str(i)
        i = i + 1
    Wend
End Sub
```

Penggantian *syntax* di atas menghasilkan tampilan sebagai berikut :



Jika kita perhatian, pada perulangan **While...Wend** terdapat fungsi incremental ($i = i + 1$), sedangkan pada perulangan **For... Next** fungsi incremental dilakukan oleh statement **next**.

3.4 Do....Loop

Pernyataan ini mengulang blok pernyataan bila kondisi benar.

3.4.1 Do...Loop...While

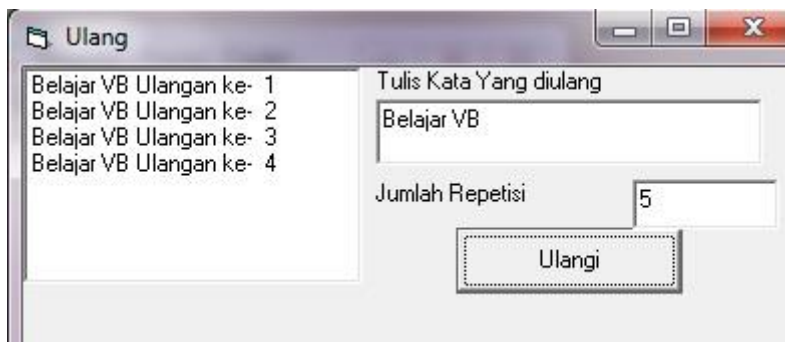
Struktur perulangan **Do...Loop...While** menggunakan pola sebagai berikut :

Do While kondisi Pernyataan/Perintah Incremental Loop
--

Mari kita modifikasi perintah command pada kasus For...Next untuk memberikan gambaran berjalannya perintah **Do...Loop...While**. Syntax pada tombol command adalah sebagai berikut:

<pre>Private Sub Command1_Click() y = Val(Text2.Text) i = 1 Do While i < y List1.AddItem Text1.Text + " Ulangan ke- " + Str(i) i = i + 1 Loop End Sub</pre>
--

Hasilnya :



Hasilnya sama dengan perintah menggunakan While...Wend

3.4.2 Do...Loop...Until

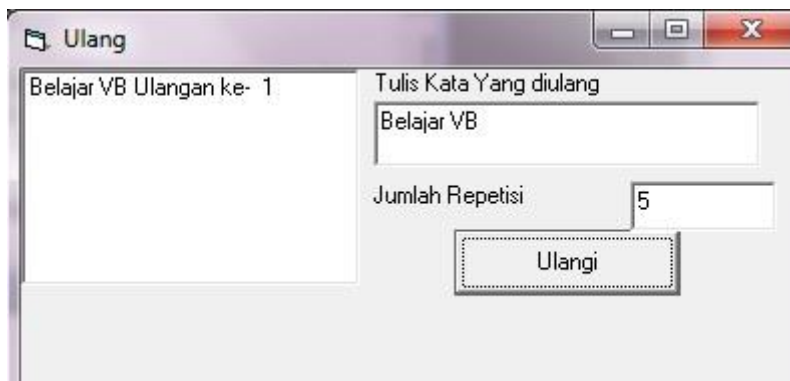
Struktur perulangan **Do...Loop...While** menggunakan pola sebagai berikut :

Do Pernyataan/Perintah Incremental Loop until kondisi
--

kita modifikasi perintah command pada kasus For...Next untuk memberikan gambaran berjalannya perintah **Do...Loop...Until**. Syntax pada tombol command adalah sebagai berikut:

```
Private Sub Command1_Click()  
y = Val(Text2.Text)  
i = 1  
Do  
    List1.AddItem Text1.Text + " Ulangan ke- " + Str(i)  
    i = i + 1  
Loop Until i < y  
  
End Sub
```

Hasilnya :



Mengapa hanya terjadi sekali perulangan?

Karena perintah dalam Do dilakukan sekali, setelahnya kondisi tidak terpenuhi (e.g. $1 < 5$), maka loop dihentikan.

BAB 4

ARRAY

4.1 Deklarasi Array

Deklarasi array untuk kumpulan data yang bertipe sama. Array digunakan untuk menyimpan kumpulan data yang bertipe sama. Data dalam tipe array disebut sebagai elemen. Indeks elemen menyatakan urutan data yang menjadi anggota array. Secara default selalu mulai dari 0 (nol).

4.2 Contoh kasus

Mari kita coba menerapkan aplikasi array untuk **Menampilkan data harga rumah;**

Langkah-langkah :

- Buatlah **Standar EXE** baru
- Kita menggunakan 1 Combo1, 4 Label, 3 TextBox dan 1 CommandButton.

Objek	Properti	Pengaturan
Form1	Caption	Harga Rumah
Combo1	(Name)	Combo1
Label1	Caption	Harga
Label2	Caption	Diskon
Label3	Caption	Harga Jual
Label4	Caption	Diskon
TextBox	(Name)	Text1
	Caption	
TextBox	(Name)	Text2
	Caption	0.2
TextBox	(Name)	Text3
	Caption	
CommanButton	Caption	Keluar

- Ketik pernyataan dengan meng-klik **Form1** sebagai berikut :


```

Private Sub Form_Load()
Combo1.List(0) = "Tipe 36"
Combo1.List(1) = "Tipe 45"
Combo1.List(2) = "Tipe 60"
Combo1.List(3) = "Tipe 72"
Combo1.List(4) = "Tipe 100"
Combo1.ItemData(0) = "1500000000"
Combo1.ItemData(1) = "1750000000"
Combo1.ItemData(2) = "2000000000"
Combo1.ItemData(3) = "2250000000"
Combo1.ItemData(4) = "2500000000"
End Sub

```

d. Ketik pernyataan dengan meng-klik **CommandButton** sebagai berikut :

```

Private Sub Command1_Click()
End
End Sub

```

e. Ketik pernyataan dengan meng-klik **Combo1** sebagai berikut :

```

Private Sub Combo1_Click()
Dim Harga As Single
Harga = Combo1.ItemData(Combo1.ListIndex)
Text1.Text = Val(Harga)
Text3.Text = Harga * (1 - (Val(Text2.Text)/100))
End Sub

```

h. Simpan dalam **ProjectArray** dan form sebagai **HargaRumah**

f. Ketika dijalankan, aplikasi akan bekerja sebagai berikut :



BAB 5

BASIS DATA

5.1 Sistem Basis Data

Sistem basis data adalah system penyimpanan data secara elektronik. Pada bagian ini akan disampaikan pemrograman VB dengan menggunakan basis data.

5.2 Membuat Basis Data Sederhana

Pada bagian ini, mari kita membuat basis data sederhana menggunakan Microsoft Access 2010.

Langkah-langkah :

- a. Buka Micosoft Access
- b. Klik **Blank Database**
- c. **Create** file **DataSiswa.accdb** dan simpan dalam format Ms Access 2007 (*.accdb) di desktop
- d. Klik kanan pada Table1, klik **Design View** dan berikan nama pada tabel dengan nama **Table1**
- e. Buatlah **Table1**, **Field** terdiri dari ID, Nama, Alamat.

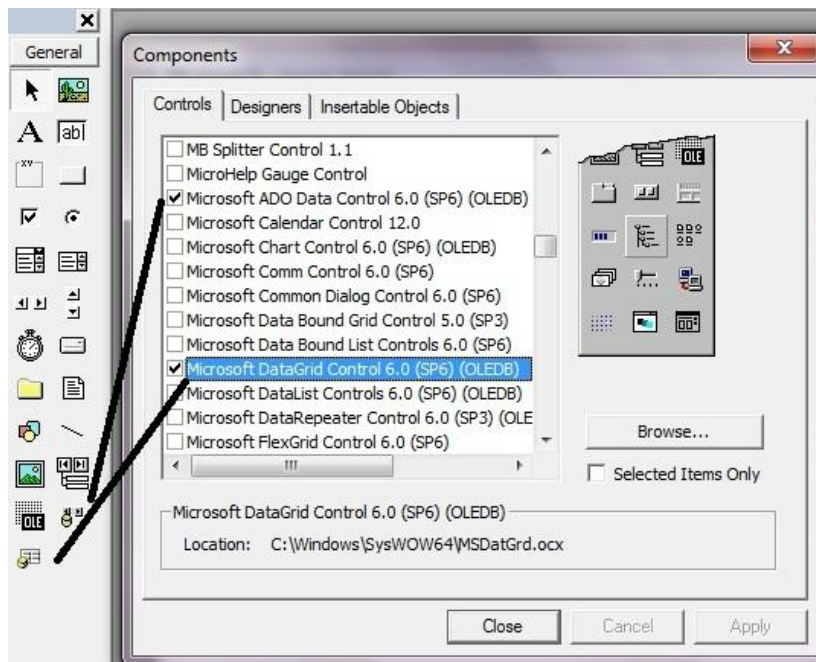
ID	AutoNumber
Nama	Text
Alamat	Text

- f. Isi Data
- g. Ingatlah lokasi anda menyimpan file **DataSiswa**, karena file akan digunakan pada kegiatan berikutnya.

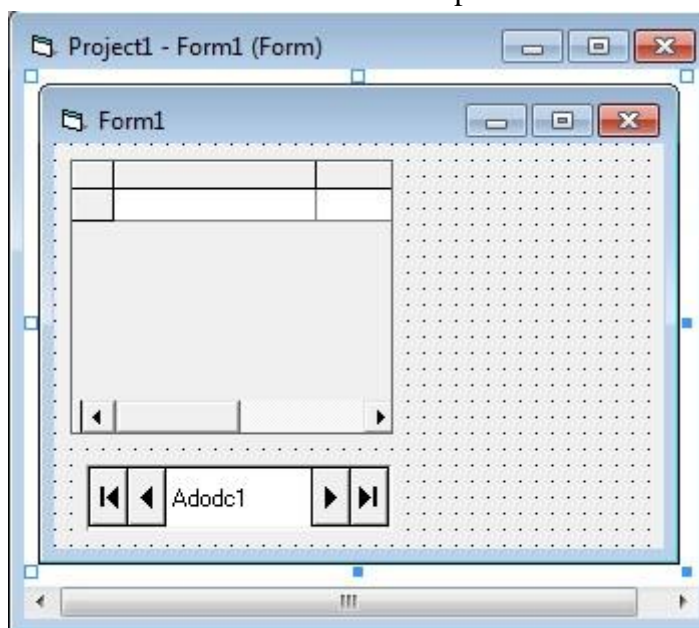
5.3 Menghubungkan VB 6 dengan basis data Microsoft Access 2007 (*.accdb)

Langkah – Langkah :

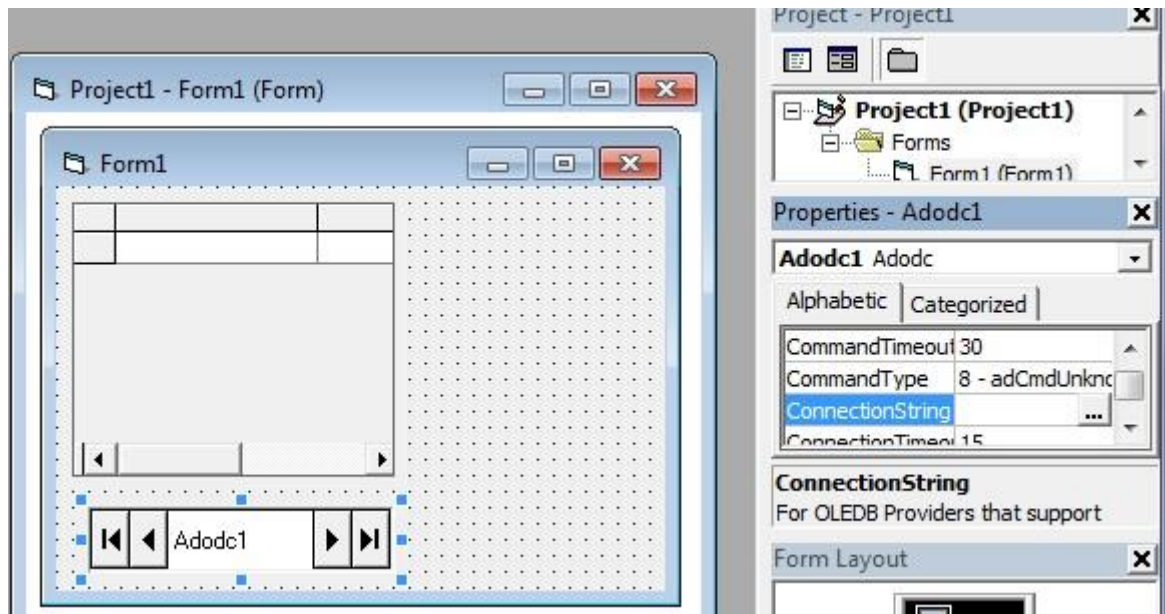
- a. Buatlah **Standar EXE** baru
- b. Tambahkan komponen **Microsoft ADO Data Control 6.0 (OLEDB)** dan **Microsoft DataGrid Control 6.0 (OLEDB)**. Klik :
 - Project > Components > **Microsoft ADO Data Control 6.0 (OLEDB)**, Klik Apply
 - Project > Components > **Microsoft DataGrid Control 6.0 (OLEDB)**, Klik Apply



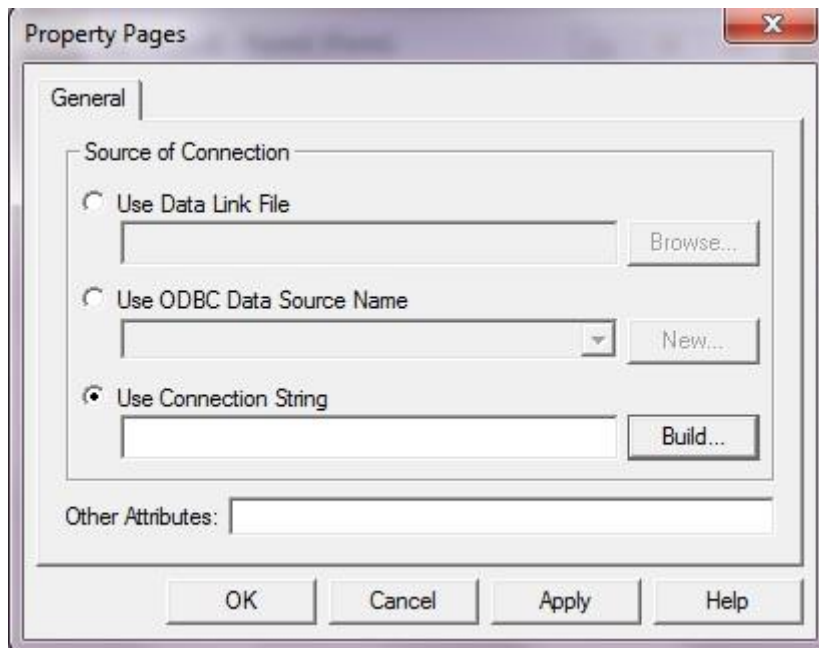
- c. Buatlah interface untuk kedua komponen dalam 1 form.



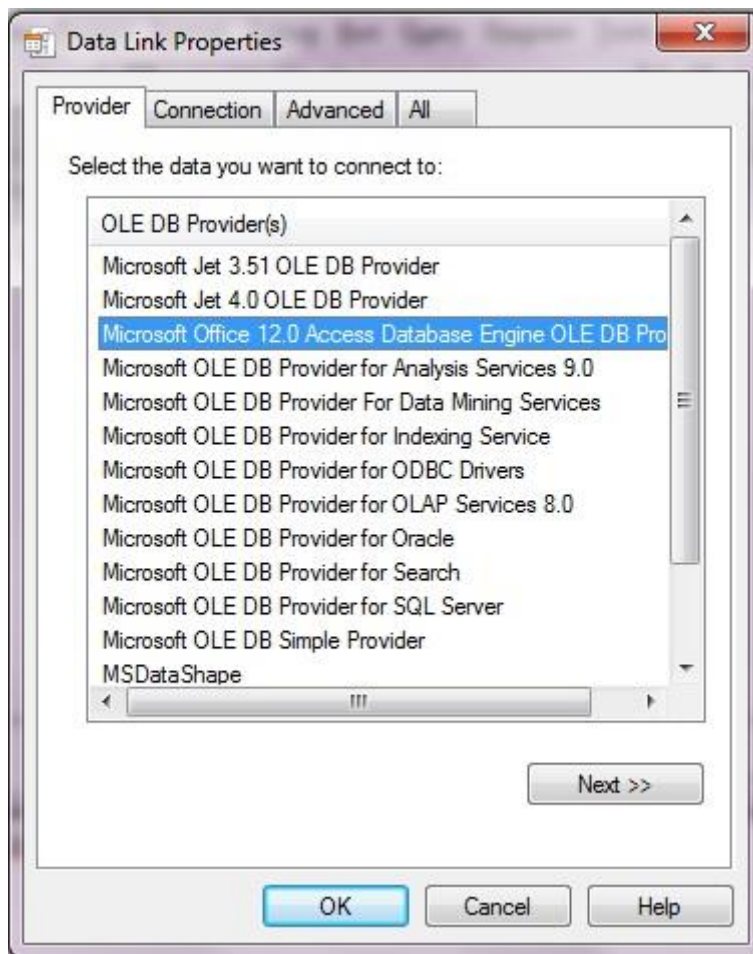
- d. Simpan Form dan Project.
e. Hubungkan Adodc1 dengan basis data yang sudah dibuat dengan mengisi Properties Adodc1 bagian **ConnectionString (klik)**.



f. Klik **Build**

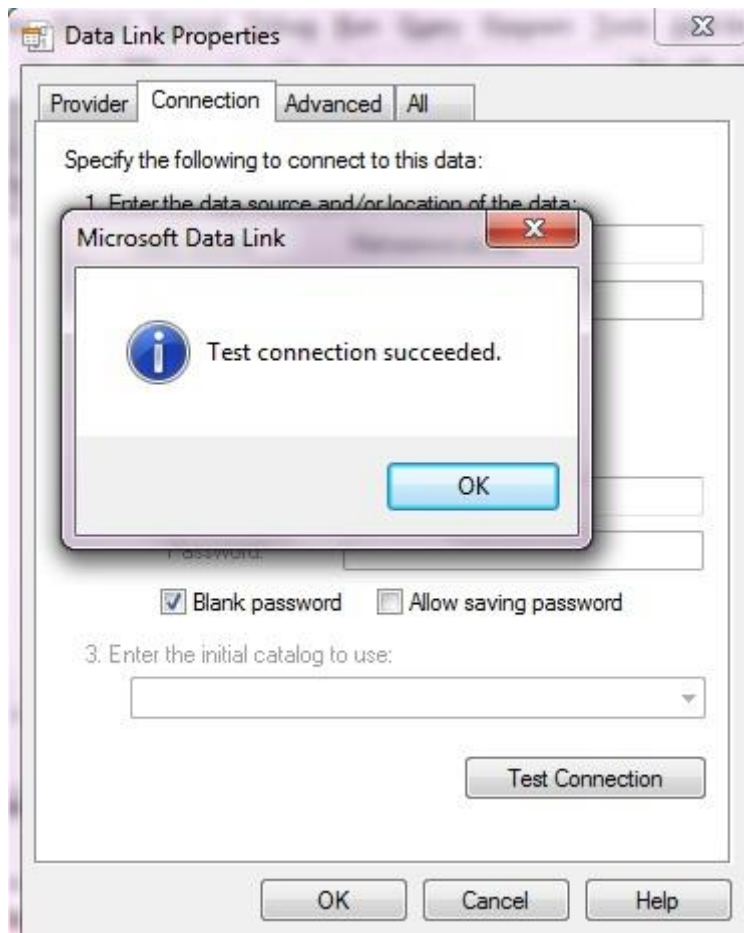


g. Pilih Microsoft Office 12.0 Access Database Engine OLE DB Provider

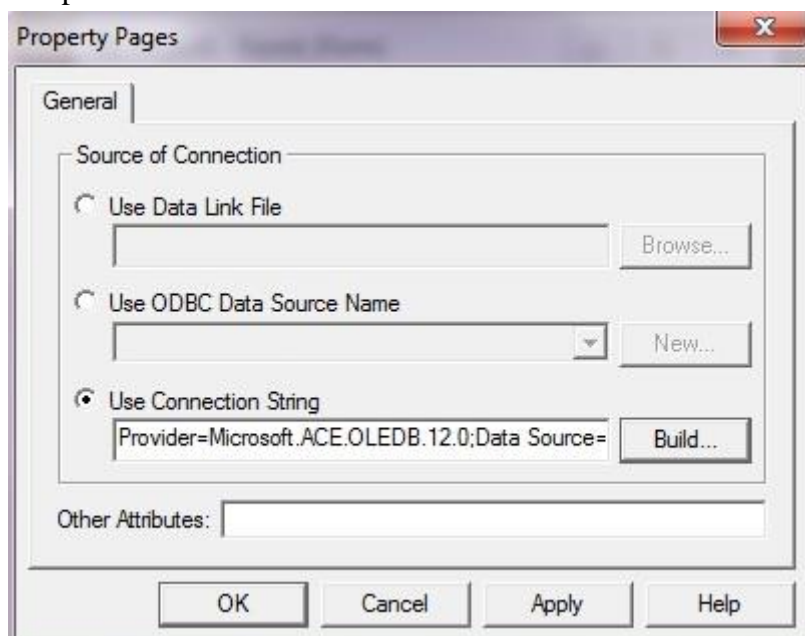


- h. Klik Next dan pilih **Data Source**: Isi dengan alamat lokasi file basis data yang sudah dibuat. Misalnya :C:\Users\UU\Desktop\DataSiswa.accdb , kemudian **test connection**.

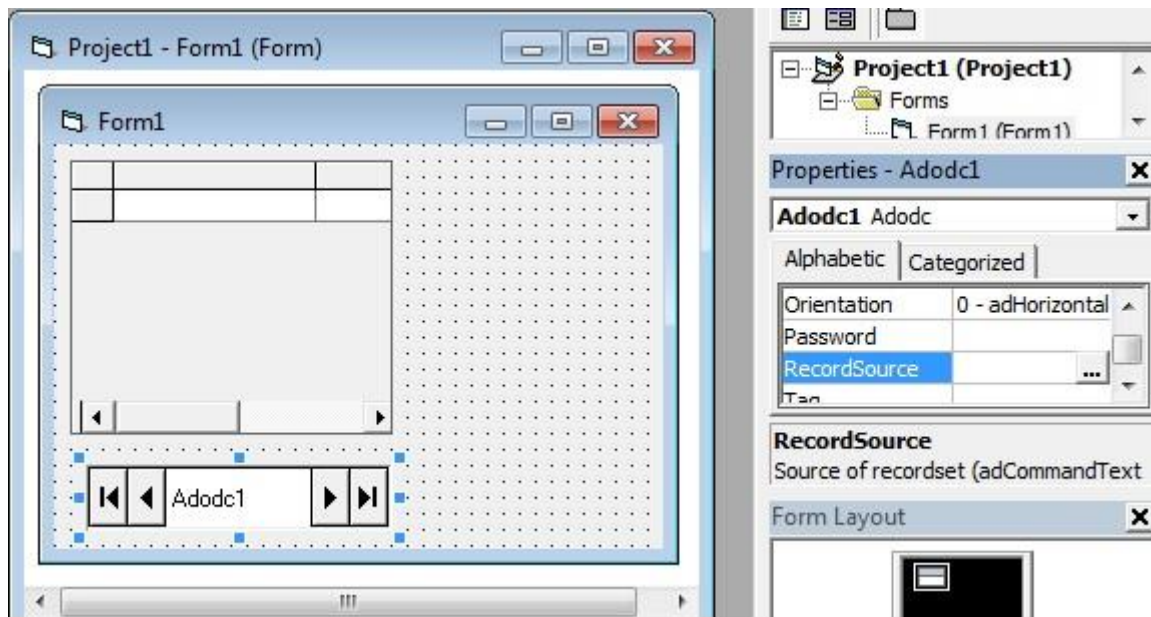
Bila koneksi berhasil akan muncul seperti berikut



- i. Kemudian Klik OK pada test koneksi di atas
- j. Sampai muncul halaman berikut ini.

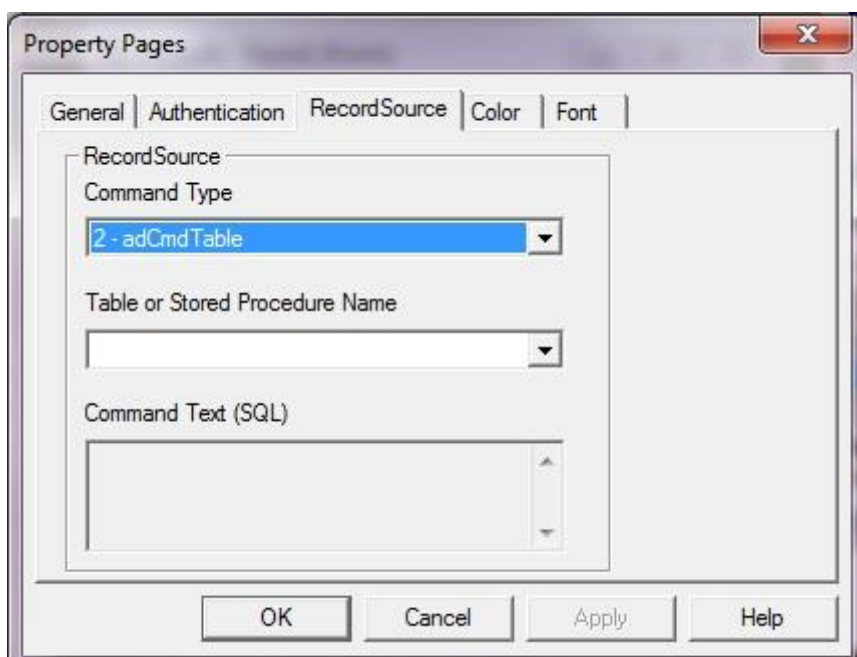


- k. Klik Apply dan OK pada interface di atas.
- l. Isi menu Properties untuk **Adodc1** bagian **Record Source**



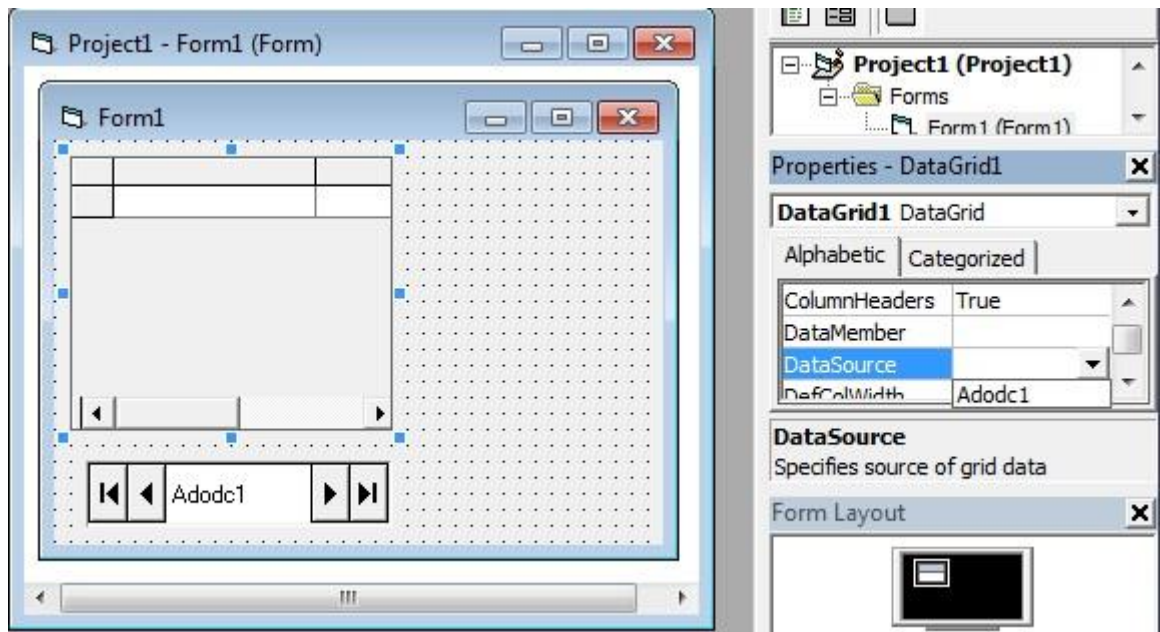
Isi **RecordSource**

- pada bagian **Command Type** dengan **2-adCmdTable**
- pada bagian **Table or Stored Procedure Name** dengan **Table 1**
- klik **apply** sebagai berikut :

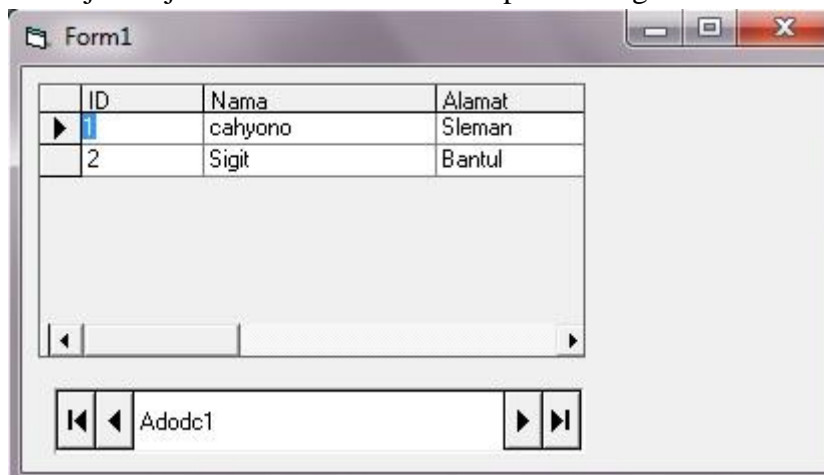


- Klik **OK**

m. Untuk **DataGrid1**. Isi **Properties**-nya pada bagian **DataSource** dengan **Adodc1** seperti pada gambar di bawah ini.



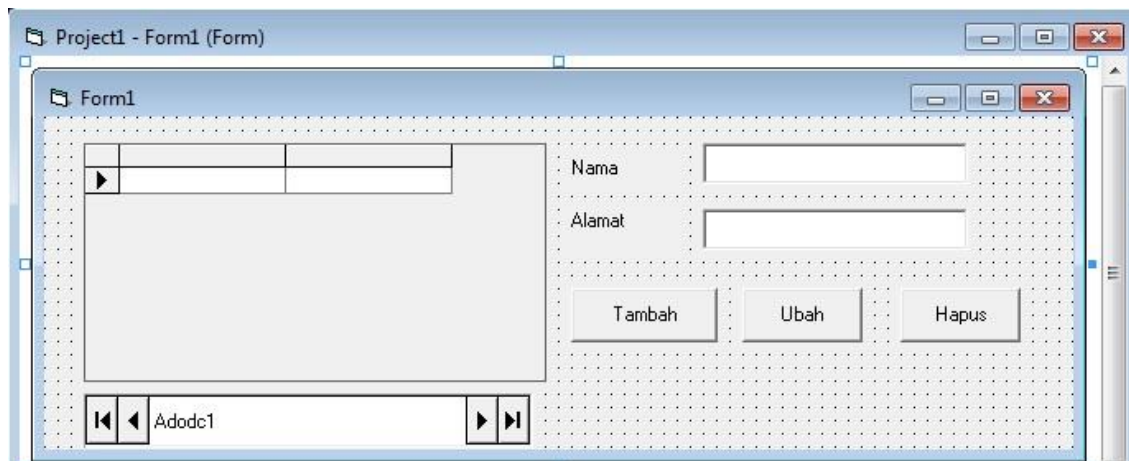
- n. Maka jika dijalankan akan muncul tampilan sebagai berikut :



5.4 Menambahkan fungsi tambah, ubah dan hapus

Langkah – langkah :

- Tambahkan 2 Label (untuk Nama dan Alamat), 2 TextBox (Text1 dan Text2 untuk mengisi data Nama dan Alamat) , dan tambahkan 3 tombol (Tambah, Ubah, dan Hapus). Tampilan akan seperti berikut :



- b. Klik tombol Tambah, kemudian

```
Private Sub Command1_Click()
    Adodc1.Recordset.AddNew
    Adodc1.Recordset!Nama = Text1.Text
    Adodc1.Recordset!Alamat = Text2.Text
End Sub
```

Ketika melakukan pengisian bagian ID akan terisi sendiri karena setting basis data sebagai **autonumber**.

- c. Untuk mengubah data, lakukan klik pada DataGrid kemudian tuliskan

```
Private Sub DataGrid1_Click()
    Text1.Text = Adodc1.Recordset!Nama
    Text2.Text = Adodc1.Recordset!Alamat
End Sub
```

Kemudian, klik tombol Ubah, kemudian tuliskan

```
Private Sub Command2_Click()
    Adodc1.Recordset!Nama = Text1.Text
    Adodc1.Recordset!Alamat = Text2.Text
    Adodc1.Recordset.Update
    MsgBox "Data Sudah Diubah"
End Sub
```

- d. Klik tombol Hapus, kemudian tuliskan

```
Private Sub Command3_Click()
    Adodc1.Recordset.Delete
    Text1.Text = ""
    Text2.Text = ""
    MsgBox "Data Sudah Dihapus"
    Adodc1.Refresh
End Sub
```

- e. Maka jika dijalankan akan muncul tampilan sebagai berikut :

	ID	Nama	Alamat
	1	Cahyono	Sleman
▶	2	Sigit	Bantul

Nama:

Alamat:

Navigation:

BAB 6

PENANGANAN KESALAHAN

6.1 Jenis Kesalahan

Terdapat 3 jenis kesalahan :

- Kesalahan penulisan kode /syntax. Kesalahan ini melanggar aturan penulisan pada visual basic
- Kesalahan Run Time, yaitu kesalahan yang menyebabkan program berhenti. Misalnya, masukan yang seharusnya angka tetapi diisi dengan huruf.
- Kesalahan logika, yaitu kesalahan hasil program.

6.2 Contoh Kasus

Untuk memberikan gambaran, kita akan menggunakan aplikasi konversi nilai :

Contoh kasus IF : Pengelompokan Nilai Angka menjadi Huruf.

Sebuah matakuliah akan mengelompokkan nilai mahasiswa menjadi huruf sebagai berikut :

Nilai (X)	Nilai angka
$X \geq 80$	A
$70 \leq X < 80$	B
$60 \leq X < 70$	C
$X < 60$	D

Langkah – langkah :

- Buatlah **Standar EXE** baru
- Kita akan menggunakan 1 TextBox, 3 Label dan 1 command.

Properti :

Objek	Properti	Pengaturan
Form1	Caption	Konversi Nilai
Label1	Caption	Masukan Nilai Angka
Label2	Caption	Nilai Huruf Anda adalah
Label3	Caption	
	Allignment	Center
Command1	Caption	Konversikan Nilai
TextBox	Text	

- Klik **Command1**, kemudian tuliskan perintah sebagai berikut :

```

Private Sub Command1_Click()
Dim bil As Integer

If Not IsNumeric(Text1.Text) Then
MsgBox "Bukan Numerik"
Text1.Text = ""
Exit Sub
End If

bil = Text1.Text
If bil >= 80 Then
Label3.Caption = "A"
ElseIf bil >= 70 Then
Label3.Caption = "B"
ElseIf bil >= 60 Then
Label3.Caption = "C"
Else
Label3.Caption = "D"
End If

```

- d. Simpan aplikasi.
- e. Ketika dijalankan, aplikasi akan bekerja sebagai berikut :



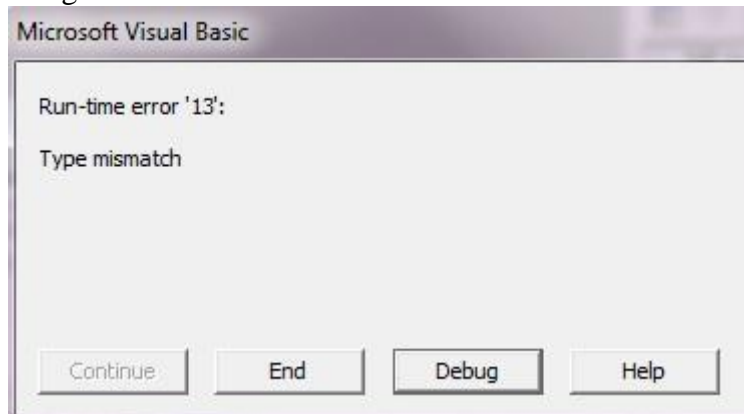
- f. Bila kita mengisikan angka dengan huruf, maka akan muncul peringatan



g. Apabila kita hilangkan syntx berikut :

```
If Not IsNumeric(Text1.Text) Then  
MsgBox "Bukan Numerik"  
Text1.Text = ""  
Exit Sub  
End If
```

Ketika kita memasukkan huruf maka aplikasi akan menampilkan kesalahan runtime sebagai berikut :



DAFTAR PUSTAKA

Kusumo, A. S. (2001). *Buku Latihan Microsoft Visual Basic 6*. Jakarta: PT. Elex media Komputindo.

Kusumo, A. S. (2003). *Buku Latihan Pemrograman Database Dengan Visual Basic 6.0*. Jakarta: PT Elex Media Komputindo.

Subekti, M. (2004). *Sistem Manajemen Basis Data*. Bogor: Ghalia Indonesia.

Wahid, F. (2004). *Dasar-dasar algoritma dan pemrograman*. Yogyakarta: Penerbit Andi.