BAB II

LANDASAN TEORI

2.1. Pengenalan Visual Basic 6.0

Visual Basic 6.0 merupakan salah satu *tool* untuk pengembangan aplikasi yang banyak diminati oleh orang. *Visual Basic* merupakan bahasa pemrograman komputer. Di sini *Visual Basic 6.0* menawarkan kemudahan dalam pembuatan aplikasi dan dapat menggunakan komponen-komponen yang telah disediakan. Untuk memulai *Visual Basic 6.0* perlu dipasang *software Visual Basic 6.0* pada komputer.

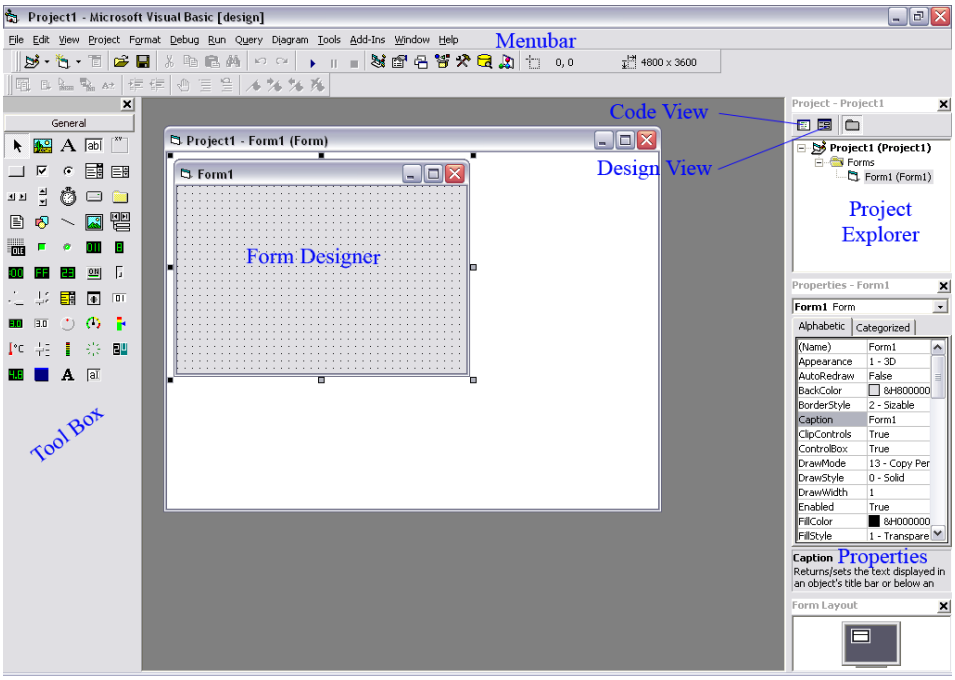
Dengan menggunakan *Visual Basic 6.0* dapat menghasilkan berbagai macam jenis program. Aplikasi yang dibuat dapat diintegrasikan dengan *database*, *hardware* lain (*interface*) dan sebagainya.



Gambar 2.1 Tampilan awal Visual Basic 6.0

Pada layar awal akan muncul tampilan seperti di atas. *Visual Basic 6.0* menyediakan banyak jenis modul aplikasi. Untuk memulai program standar pilihlah **Standar EXE**, kemudian klik open.

Setelah itu akan muncul tampilan seperti berikut ini, yang menunjukan bagian-bagian dari IDE (*Integrated Development Environment*) yang akan digunakan.



Gambar 2.2 Tampilan lembar kerja Visual Basic 6.0

Form Designer

Form design merupakan tempat perancangan *user interface* (antar muka pemakai). Untuk menampilkan layar ini, klik **Design View** atau dengan menekan **shift + F7**. Sedangkan untuk menampilkan layar *coding* dapat menekan **F7** atau klik pada **Coding View**.

Menu

Merupakan menu standar pada *Windows*, dapat digunakan untuk menyimpan *project*, menjalankan *project*, membuka *project* baru dan sebagainya.

Toolbox

Toolbox merupakan tempat komponen-komponen yang disediakan untuk merancang *user interface*. Setiap komponen memiliki ciri dan kegunaan yang berbeda, disesuaikan dengan kebutuhan pengguna.

Project Explorer

Merupakan struktur *project* yang sedang dikerjakan, suatu *project* dapat terdiri dari beberapa *form*.

Properties

Menampilkan bagian-bagian dari komponen yang sedang aktif. Setiap komponen mempunyai karakteristik yang berbeda, bergantung pada kegunaan.

2.2. Pengkodean pada Visual Basic 6.0

2.2.1. Tipe Data

Tipe data memiliki ciri – ciri tersendiri. Berikut bentuk dan ukuran dari tipe data:

Table 2.1 Ukuran dari tipe data

Tipe Data	Ukuran Storage	Jangkauan
Byte	1 Byte	0 s/d 255
Boolean	2 Byte	True atau False
Integer	2 Byte	-32768 s/d 32767
Long	4 Byte	-2.147.483.648 s/d 2.147.483.647
Single	4 Byte	-3,40282e38 s/d -1,401296e-45 (-)
		1,401296e-45 s/d 3,402823e38 (+)
Double	8 Byte	-1,797691348623e308 s/d -4,9406564844127
Currency	8 Byte	-922.337.203.685.477,5808 s/d 922.337.203.685.477,5807
Decimal	14 Byte	7,92E+028
Date	8 Byte	1 Januari 100 s/d 31 desember 9999
Object	4 Byte	Mangacu pada objek tertentu
String	Panjang dari string	1 sampai ± 65400
Variant	16 Byte	Sembarang angka sampai jangkauan jenis double atau string

2.2.2. Variabel

Variabel digunakan untuk menampung nilai sementara di memori. Untuk membuat sebuah variabel terdapat ketentuan sebagai berikut :

a. Harus dimulai dengan suatu huruf
b. Tidak dapat mengandung titik atau spesial karakter
c. Tidak dapat lebih dari 255 huruf
d. Tidak dapat sama dengan *keyword* dari visual basic
e. Tidak membedakan huruf besar dan huruf kecil (*no case sensitive*)

Cara mendeklarasi variabel adalah sebagai berikut :

```
Dim [nama variabel] As [tipe data] atau
Public [nama variabel] As [tipe data] atau
Private [nama variabel] As [tipe data]
```

Public akan membuat suatu variabel dapat diakses dari segala tempat di dalam *project*, sedangkan **Dim** dan **Private** akan membuat suatu variabel yang hanya dapat diakses di dalam module dimana variabel tersebut dideklarasikan.

2.2.3. Operator

Visual basic menyediakan operator aritmatika, komparasi dan logika, salah satu hal yang harus dipahami adalah tata urut dari masing–masing operator, sehingga mampu membuat ekspresi yang akan menghasilkan nilai yang benar.

a. Operator aritmatika

Table 2.2 Operator Aritmatika

Nama Operator	Tanda Operator
Pangkat	^
Negatif	-
Kali dan Bagi	* , /
Pembagian Bulat	\
Sisa Bagi	Mod
Tambah dan Kurang	+ , -
Penggabungan string	&

b. Operator Komparasi

Table 2.3 Operator Komparasi

Nama Operator	Tanda Operator
Sama	=
Tidak Sama	<>
Kurang dari	<
Lebih dari	>
Kurang dari sama	<=
Lebih dari sama	>=
Like	Like

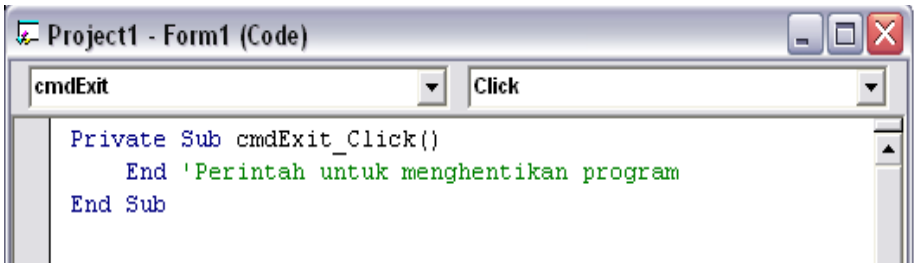
c. Operator Logika

Table 2.4 Operator Logika

Nama Operator	Tanda Operator
Not	Not
And	And
Or	Or
Xor	Xor

2.2.4. Event-driven

Visual Basic 6.0 bersifat *event-driven*, artinya program bekerja berdasarkan *event* yang terjadi pada objek di dalam program tersebut. Misalkan *event* klik pada tombol *exit*, pada saat tombol *exit* diklik maka program akan berhenti/keluar. Caranya klik dua kali pada tombol *exit*, maka akan muncul layar *coding* sebagai berikut.



Gambar 2.3 Tampilan layar pengkodean

2.2.5. Struktur Pemilihan

Struktur pemilihan biasanya menggunakan *if-else*. Contoh penggunaan perintah *if-else*, sebagai berikut :

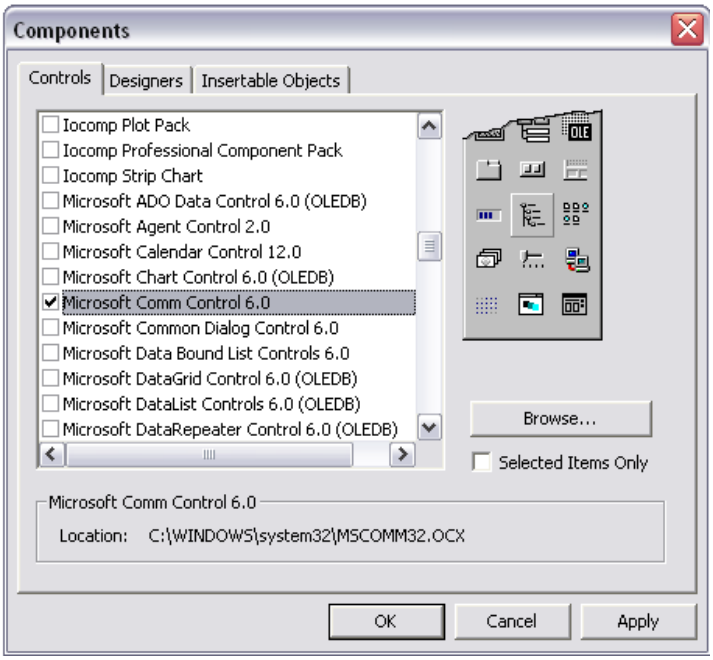
```
If kondisi Then
        ….
        ….
End If
```

```
If kondisi Then
        ….
        ….
Else
        ….
        ….
End If
```

```
If kondisi Then
        ….
        ….
ElseIf kondisi Then
        ….
        ….
End If
```

2.3. Komunikasi serial Visual Basic 6.0

Dalam berkomunikasi dengan "dunia luar" atau perangkat lain di luar komputer, *Visual Basic 6.0* menyediakan komponen **MS Comm Control 6.0**, sebagai media komunikasi. Untuk menambahkan komponen ini pada *Visual Basic*, pilih **Project** -> **Components**. Setelah itu akan muncul tampilan sebagai berikut :



Gambar 2.4 Penambahan komponen pada Visual Basic 6.0

Fungsi dari komponen ini adalah sebagai berikut :

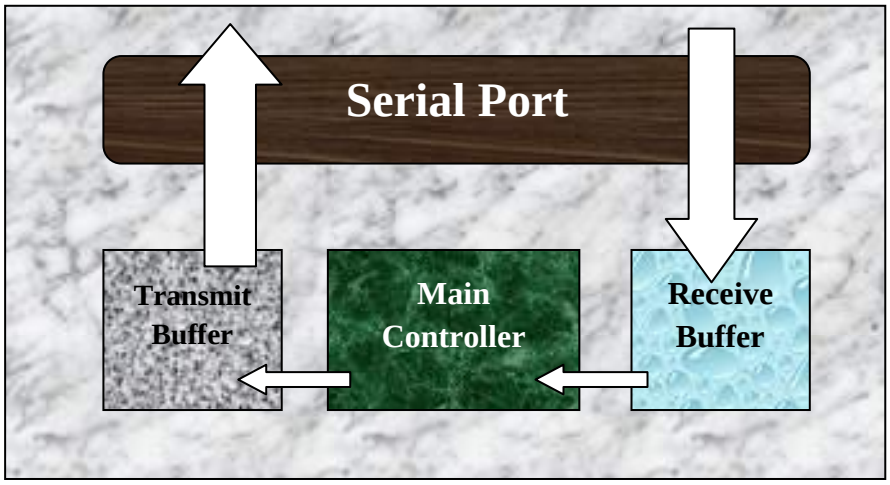
a. Mengadakan hubungan dengan *serial port* PC
b. Berhubungan dengan alat komunikasi lain (contoh: *modem*)
c. Melakukan pertukaran data
d. Memonitor dan merespon *event* dan *error* yang terjadi pada hubungan *serial*

Untuk mengadakan suatu komunikasi serial antara 2 peralatan, kita harus melakukan beberapa langkah.

1. Membuka *serial port*
2. Mengatur *serial device*
3. *Setting Receive and Transmit Buffer Properties*
4. *Managing Receive and Transmit Buffer*

2.3.1. Membuka Serial Port

Pada komunikasi serial, bit-bit data yang masuk dari dunia luar ke dalam komputer melalui *serial port* akan ditampung dulu di *receive buffer* sebelum akan dieksekusi oleh *main controller*. Demikian pula sebelum dikirimkan ke luar, data akan ditampung dulu di *transmit buffer*. Gambar skema lengkapnya dapat dilihat pada gambar di bawah ini.



Gambar 2.5 Skema jalur penerimaan dan pengiriman data

Sebelum membuka *serial port*, dilakukan pengaturan protokol komunikasi serial dengan *property* MSComm. Menentukan nomor *port* komunikasi menggunakan *CommPort* dan menentukan *baud rate*, *parity*, data *bits*, *stop bits* mengguankan *property setting*. Sedangkan untuk membuka *serial port* cukup menggunakan perintah *PortOpen*. Sehingga kode program akan tertulis sebagai berikut :

```
MSComm1.ComPort = 2
MSComm1.Settings = "9600,N,8,1"
MSComm1.PortOpen = True
```

2.3.2. Mengatur Serial Device

Pada tahap ini kita perlu memastikan bahwa pengaturan protokol komunikasi serial yang digunakan pada peralatan lain yang kita akses, sesuai dengan pengaturan pada komputer yang kita pakai.

2.3.3. Setting Receive dan Transmit Buffer Properties

Ada beberapa *property* dari *receive buffer* dan *transmit buffer* (*porperty* dari **MSComm**) yang perlu kita atur.

a. *InBufferSize* : mengatur ukuran *receive buffer*
b. *OutBuffer Size* : mengatur ukuran *transmit buffer*
c. *Rthreshold* : menentukan jumlah karakter yang diterima oleh *receive buffer* sebelum *OnComm event* dipicu
d. *Sthreshold* : menentukan jumlah karakter yang diterima oleh *transmit buffer* sebelum *OnComm event* dipicu. Nilai 0 berarti tidak pernah dipicu, sedangkan nilai 1 berarti dipicu setiap satu karakter.
e. *InputLen* : menentukan jumlah karakter yang dibaca CPU dari *receive buffer*
f. *InputMode* : menentukan tipe data input yang akan dibaca CPU. *comInputModeText* untuk data *string*/teks dan *comInputModeBinary* untuk data biner.

2.3.4. Managing Receive dan Transmit Buffer

Untuk menampilkan data dari peralatan lain ke dalam aplikasi VB, digunakan properti **Input**, sedangkan untuk mengirim data dari aplikasi VB ke peralatan lain digunakan poperti **Output**. Contoh struktur kode untuk **Input** dan **Output**.

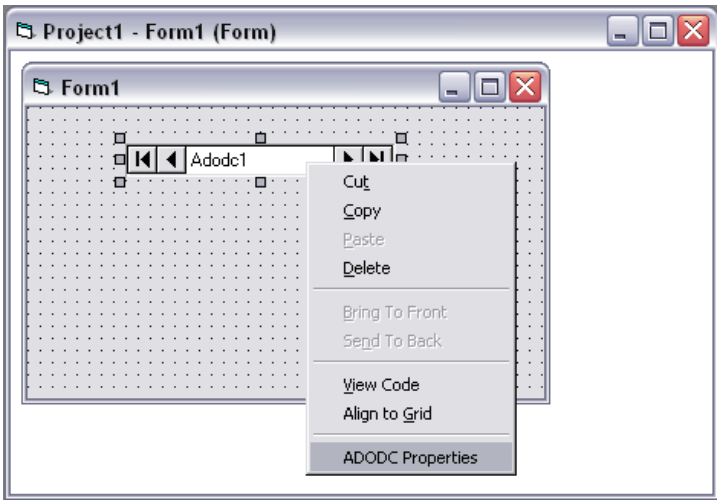
```
TxtDisplay.Text = MSComm1.Input   ( Contoh Input )
MSComm1.Output = "Data String"  ( Contoh Output )
```

2.4. Koneksi Visual Basic 6.0 dengan Database Microsoft Access

Visual Basic 6.0 memberikan antarmuka yang lebih menarik pada *user* dalam mengakses suatu sistem informasi *database*, sehingga dapat mempermudah *user* dalam mengakses suatu data yang diperlukan. Komponen yang diperlukan Visual Basic untuk mengakses *database* adalah **Microsoft ADO Data Control 6.0** (**ADODC**). Untuk melihat isi *database* bisa

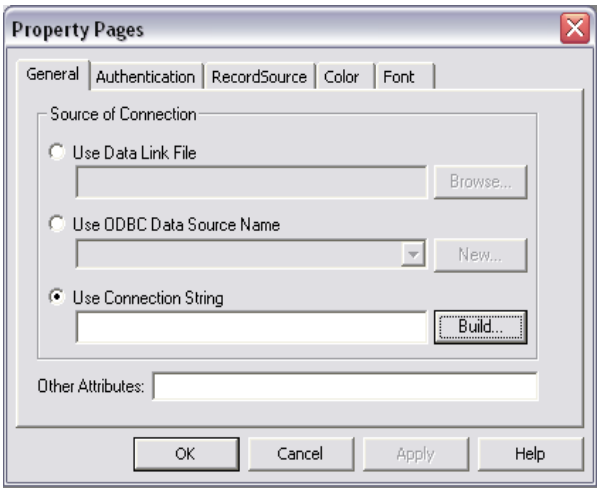
menggunakan **Microsoft Datagrid control**. Komponen *adodc* dan *datagrid* dapat ditambah pada *toolbox* dengan mengambil dari list komponen.

Untuk menghubungkan *adodc* dengan *database*, klik kanan pada *adodc* lalu pilih *ADODC properties*.



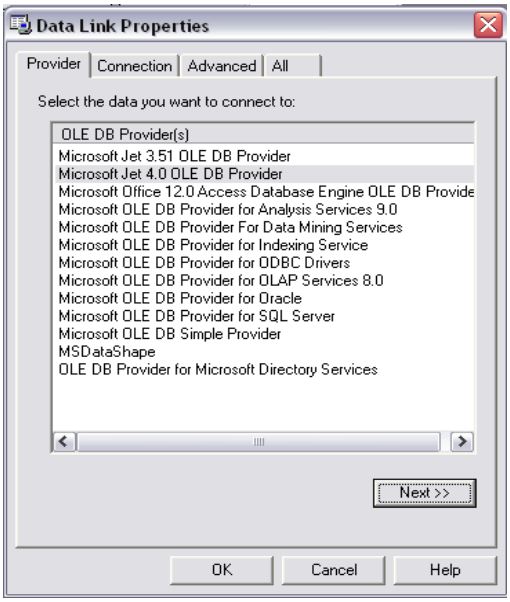
Gambar 2.6 Properties pada ADODC

Setelah dipilih maka akan muncul jendela **Property Page**. Kemudian pilih **Build**.



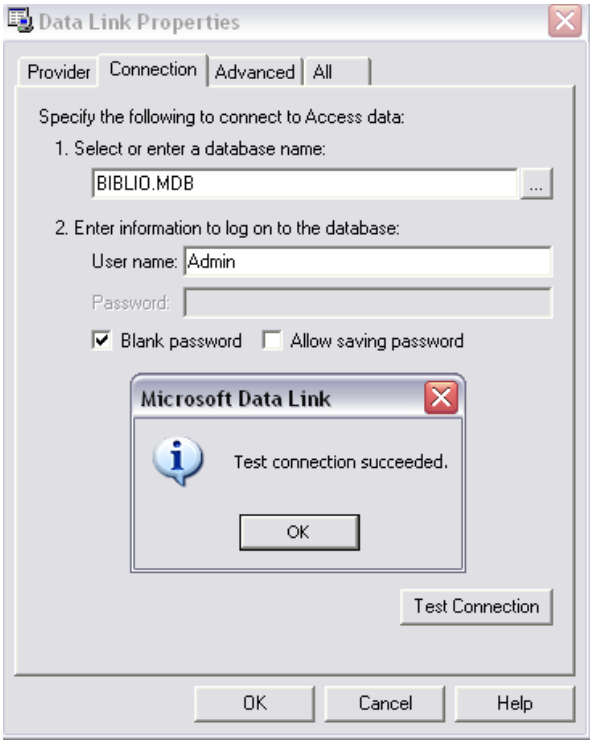
Gambar 2.7 Properties Page pada ADODC

Muncul jendela **Data Link Properties**. Lalu pilih **Microsoft Jet 4.0 OLE** dan klik **Next**.



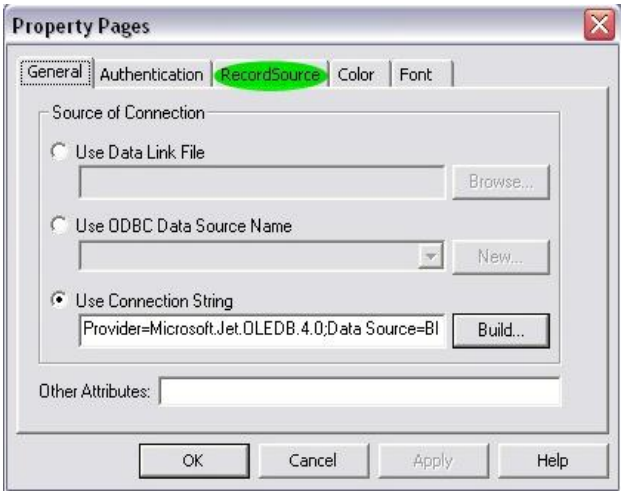
Gambar 2.8 Data Link Properties ADODC

Setelah itu ambil file *database* dengan format (***.mdb**). kemudian klik tombol **Test Connection**, setelah sukses klik tombol **OK**.



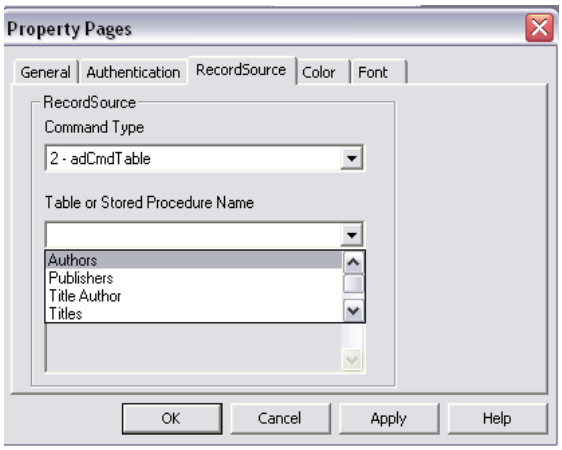
Gambar 2.9 Test connection dari database

Kemudian pilih tab **RecordSource**.



Gambar 2.10 Tab RecordSource

Pada **Command Type** pilih **2-adCmdTable**, sedangkan pada **Tabel or Stored** pilih tabel yang telah dibuat didatabase. Kemudian tekan **OK**.



Gambar 2.11 Pemilihan Tabel Database

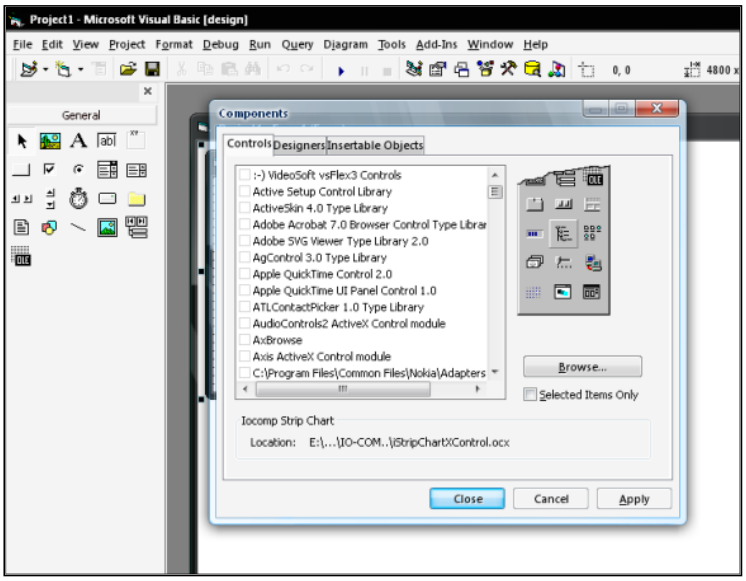
Database telah terhubung dengan Visual Basic. Untuk menampilkan isi *database* pada *Datagrid*, klik *Datagrid*-nya lalu di properti pilih *Data Source* untuk memilih *ADODC* yang telah terhubung dengan *database*.

2.5. Komponen Tambahan (ActiveX)

ActiveX adalah pengaturan terhadap komponen-komponen dan aplikasi pada antarmuka agar dapat saling berinteraksi dan berbagi informasi.

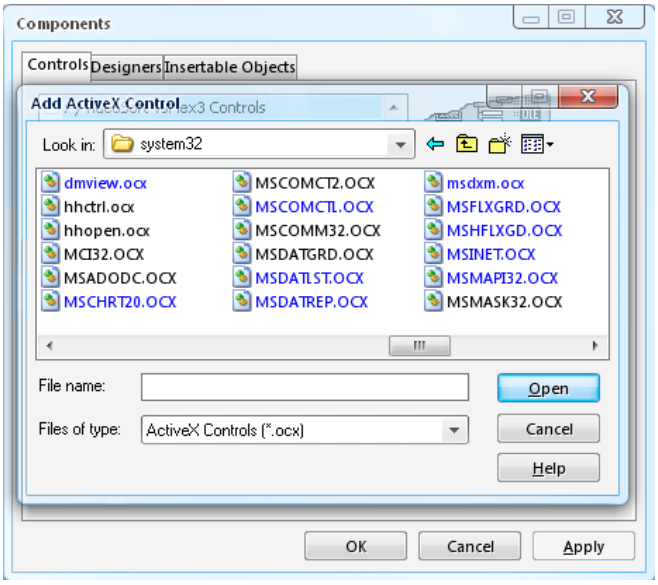
Berikut contoh penambahan *ActiveX*:

Pilih **Project->Component** sehingga muncul tampilan berikut.



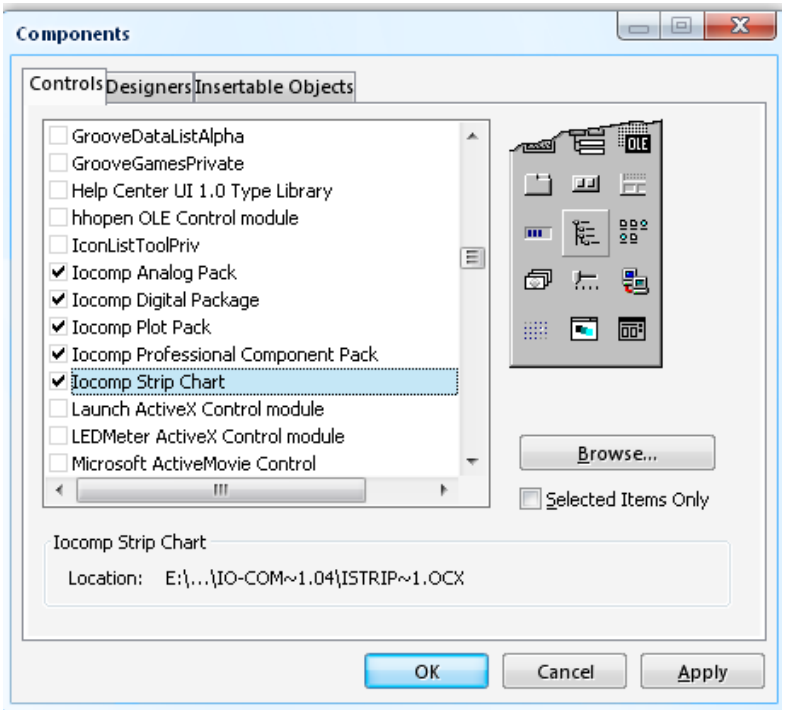
Gambar 2.12 Penambahan komponen ActiveX

Lalu tekan tombol **Browse** untuk mengambil komponen *ActiveX* (.ocx). Setelah memilih tekan **Open** untuk memasukannya kedalam daftar.



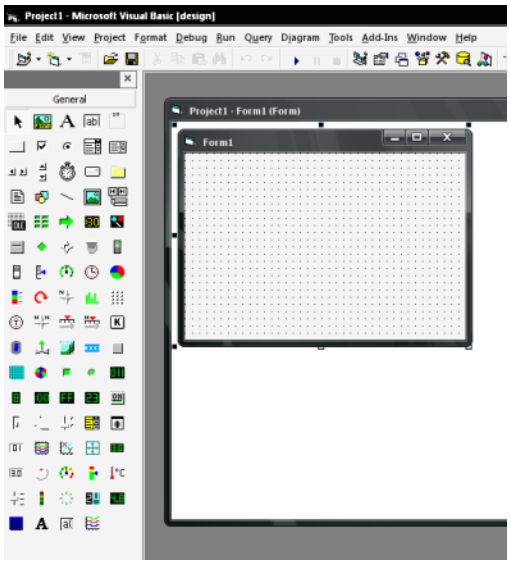
Gambar 2.13 Pengambilan komponen ActiveX

Cek list komponen yang baru dimasukan kemudian tekan OK.



Gambar 2.14 Pemilihan komponen ActiveX

Pada tool box akan muncul komponen yang baru di masukan.



Gambar 2.15 komponen yang berhasil ditambahkan

2.6. Pengiriman Data Lewat Radio

Radio yang dipakai dalam pengiriman data ini adalah YS-1020L.



Gambar 2.16 Modem Radio YS-1020L

Modem radio ini mempunyai spesifikasi sebagai berikut :

1. Mempunyai 8 channel untuk pengiriman/penerimaan data
2. Tipe modulasi yang dipakai adalah GFSK (*Gaussian Frequensy Shift Keying*). Menggunakan *Gaussian filter* untuk memperhalus penyimpangan frekuensi yang terjadi.
3. Dapat menggunakan *level* TTL (*Transistor-Transistor Logic*).
4. Integrasi antara *receiver* dan *transmiter* memerlukan waktu 10 ms antara pengiriman dan penerima.

2.7. Komunikasi radio

Sistem komunikasi radio tidak menggunakan kawat dalam proses perambatannya, melainkan menggunakan udara atau ruang angkasa sebagai bahan penghantar. Secara garis besar sistem ini adalah sebuah pemancar Tx yang memancarkan dayanya menggunakan antena ke arah tujuan, sinyal yang dipancarkan berbentuk gelombang elektromagnetis. Pada penerima gelombang elektromagnetik ini diterima oleh sebuah antena yang sesuai. Sinyal yang diterima kemudian diteruskan ke sebuah pesawat penerima Rx.

Komunikasi yang akan digunakan bersifat *half duplex*. *Half duplex* adalah komunikasi dua arah yang dilakukan secara bergantian. Jarak yang dapat dijangkau oleh radio ini sekitar 800 m.

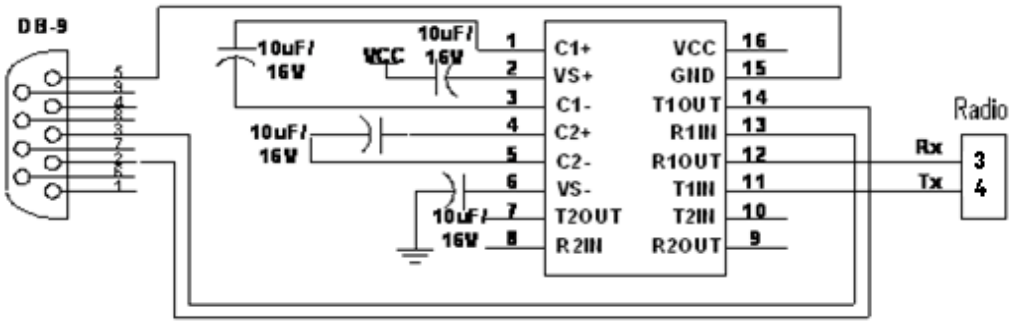
2.8. Max232

Fungsi dari IC MAX232 adalah untuk mengubah tegangan dari *level* TTL ke *level* RS232, sebab radio yang dipakai menggunakan komunikasi serial pada *level* TTL sedangkan komputer pada *level* RS232.



Gambar 2.17 Max232

Berikut rangkaian Max232 :

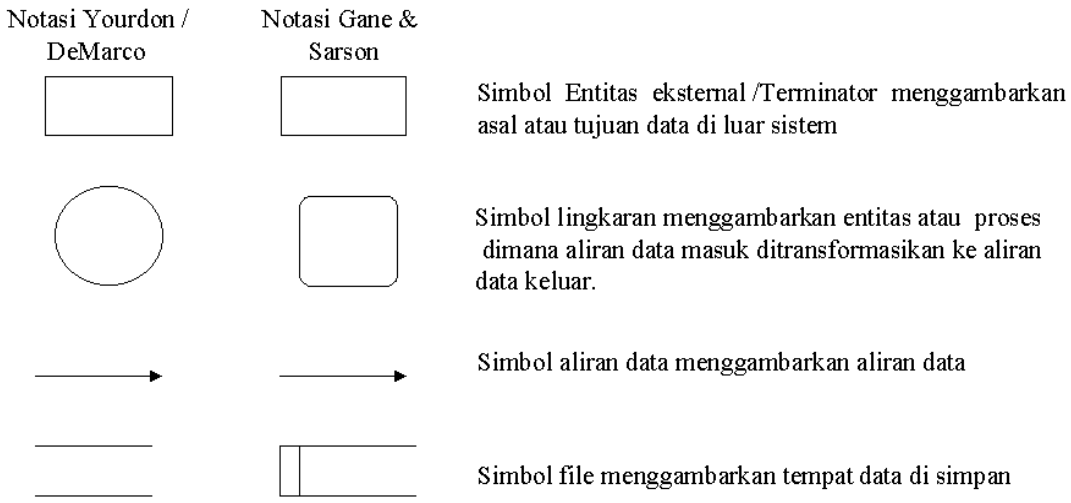


Gambar 2.18 Rangkaian Max232

2.9. Data Flow Diagram

Data flow diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut, dan asal, tujuan, dan penyimpanan dari data tersebut.

Empat simbol yang digunakan :



Gambar 2.19 Simbol yang digunakan pada DFD