

PENGENALAN VISUAL BASIC 6.0

Visual Basic 6.0 adalah suatu program aplikasi yang mempermudah programmer untuk membuat suatu program dengan cepat dan lebih mudah diakses oleh kalangan non-programmer dengan meminjam bahasa Object Oriented Project (OOP) serta mempraktikkan secara grafis dan visual. **Visual Basic** terkadang disebut *Rapid Application Development (RAD)* karena memungkinkan programmer membuat aplikasi prototype dengan cepat, RAD (application Development) adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik incremental (bertingkat). Visual basic , bahasa pemrograman berorientasi objek yang berbasis Windows dari Microsoft yang mengizinkan pengguna mengembangkan aplikasi Windows dan Office yaitu dengan:

1. Membuat tombol perintah, kotak teks, jendela dan toolbar,
2. Selanjutnya akan di-link ke program BASIC yang kecil yang melakukan tindakan tertentu.

Visual BASIC merupakan *even-driven*, artinya program menunggu pengguna melakukan sesuatu “**event**”, seperti klik pada ikon, dan kemudian program akan merespons. Misalnya , pada permulaan pengguna dapat menggunakan piranti geser dan turunkan (**drag-and-drop**) untuk mengembangkan antar muka pengguna grafis yang dibuat secara otomatis oleh program. Karena penggunaanya mudah, **Visual Basic** memungkinkan programmer pemula untuk menciptakan aplikasi-aplikasi berbasis windows yang menarik.

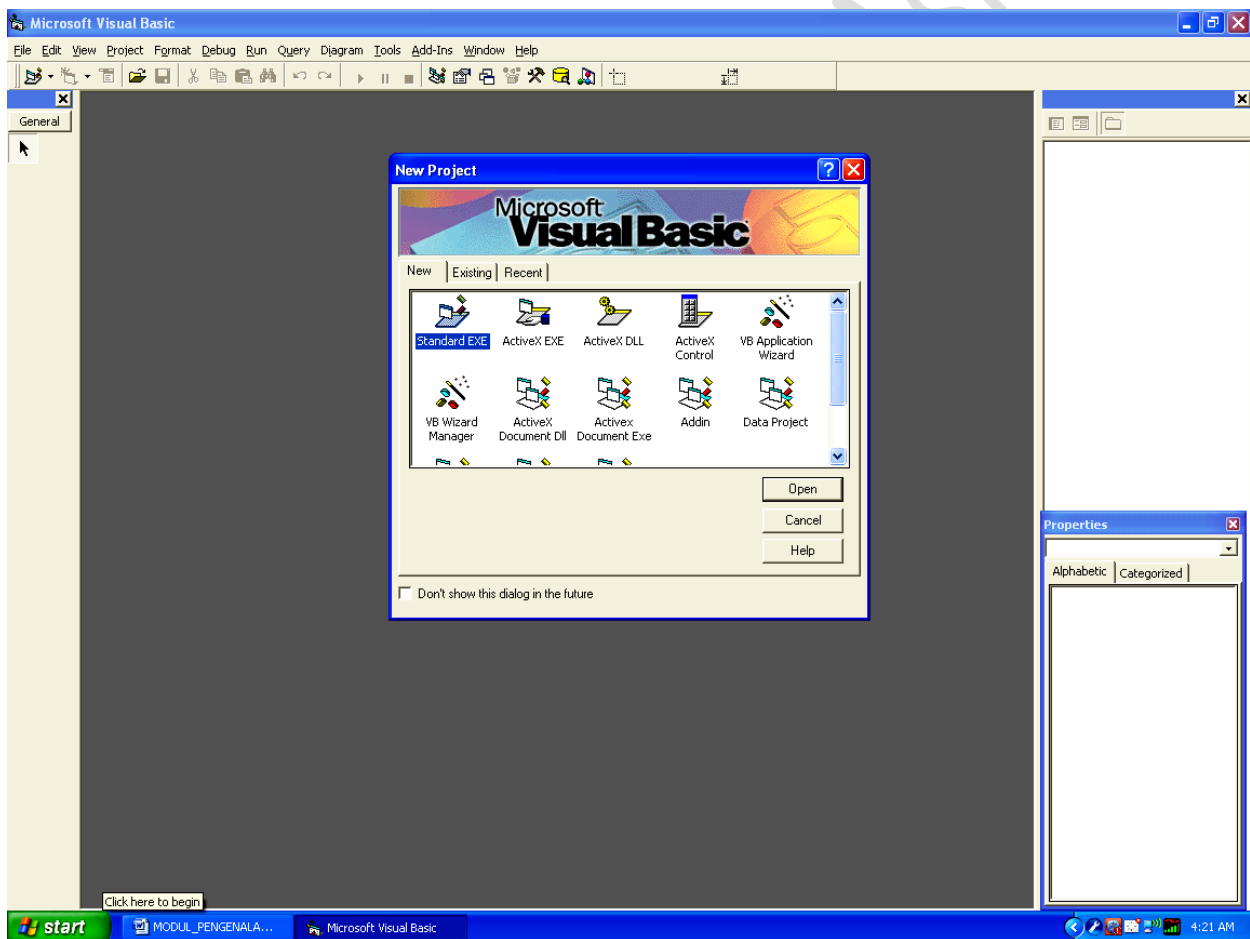
1. Pengenalan IDE Visual Basic

Langkah awal dari belajar Visual Basic adalah mengenal *IDE (Integrated Development Environment)*. *IDE (Integrated Development Environmet)* adalah tampilan muka area kerja **Visual Basic**. Dengan menggunakan *IDE* kita dapat membuat user interface, melakukan koding, melakukan testing dan debugging serta menkompilasi program menjadi executable. Penguasaan yang baik akan IDE akan sangat membantu kita dalam mengefektifkan tugas-tugasnya sehingga

dapat bekerja dengan efisien. Adapun langkah-langkah yang harus dilakukan dalam menjalankan IDE sebagai berikut:

1.1 Menjalankan IDE

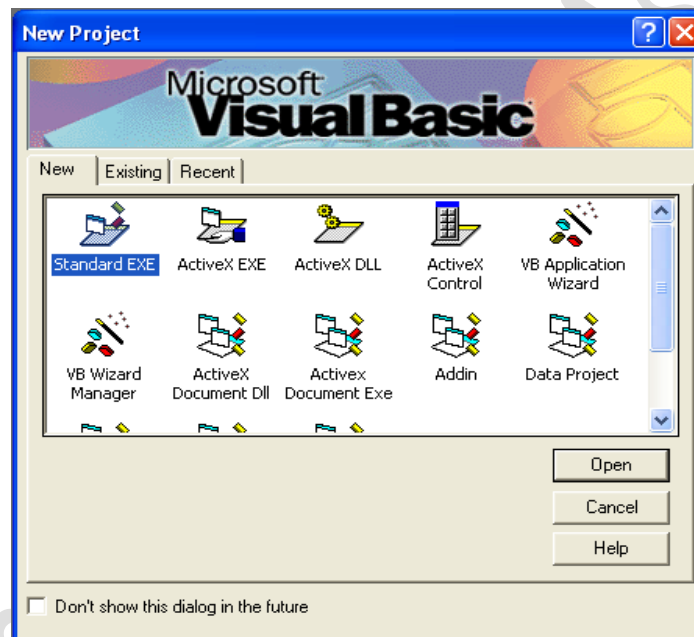
Sebelum kita menjalankan **IDE(Integrated Development Environment)** atau tampilan muka area kerja Visual Basic, kita harus terlebih dahulu membuka program Visual Basic.6 yaitu **Start**, pilih **Microsoft Visual Basic 6.0** dan akhirnya pilih shortcut **Microsoft Visual Basic 6.0**. Maka akan muncul gambar seperti di bawah ini :



Gambar 1.1 Tampilan Visual Basic 6.0

1.2 Memilih jenis Project

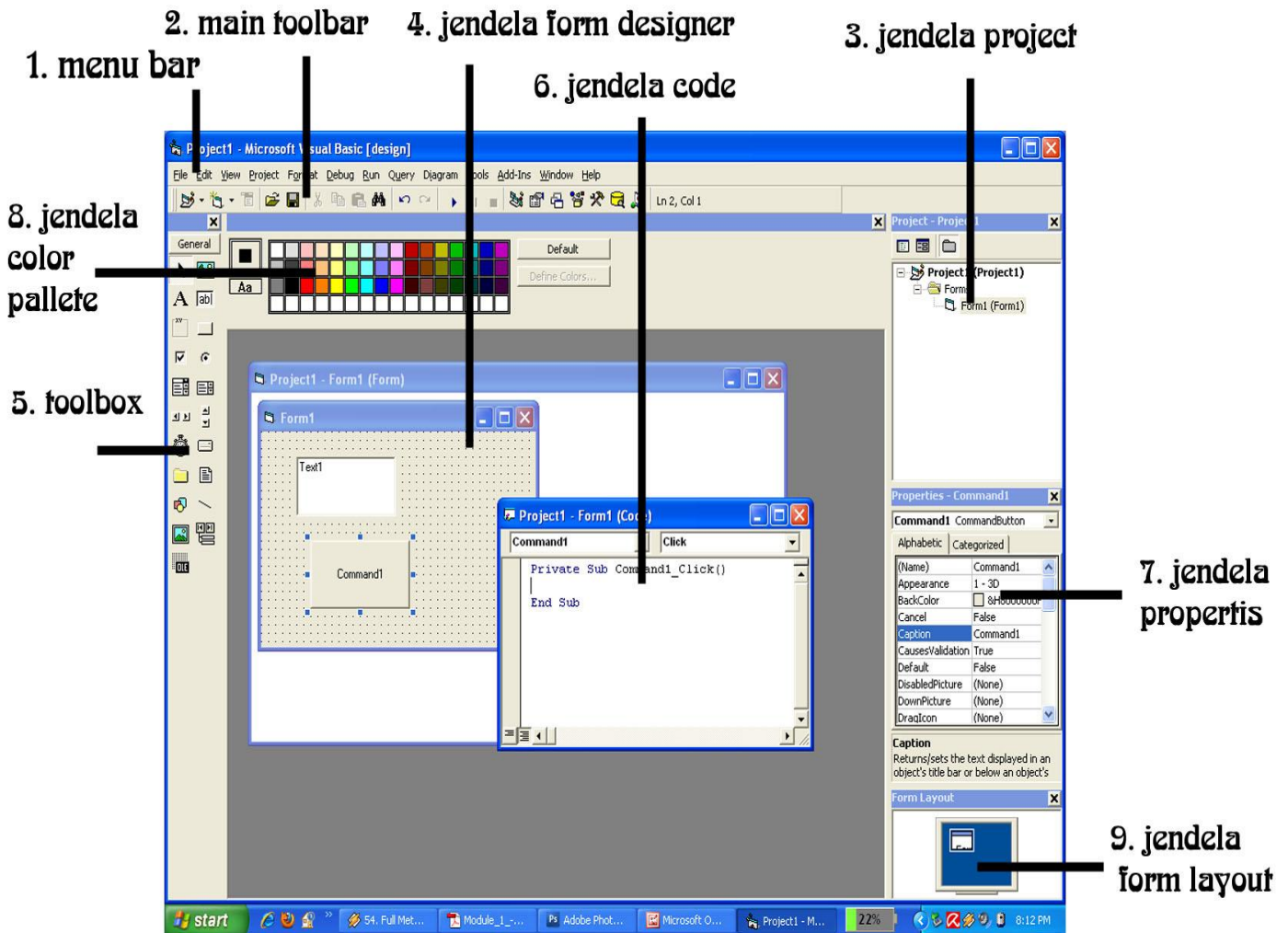
Setelah anda menjalankan Visual Basic, maka IDE Visual Basic pertama kali akan menampilkan kotak dialog "**New Project**" kepada anda, seperti yang bisa anda lihat dibawah ini , Untuk langkah awal ini, mari kita memilih **Standard EXE**, kemudian **klik open**.



Gambar 1.2 Membuka Standar Exe

1.3 Mengenal Jendela IDE

Setelah anda memilih **Standard Exe** untuk proyek yang akan kita coba bahas di sini, maka **Visual Basic 6.0** akan menampilkan **IDE** dimana tempat kita akan memulai membuat aplikasi sederhana kita. Kita akan belajar mengenali semua window-window (jendela) yang ada di **IDE Visual Basic 6.0** seperti yang terlihat di gambar berikut ini :



Gambar 1.3 Bagian Jendela IDE

Untuk proses tutorial perdana ini mengenai pengenalan terhadap Visual Basic, maka kita akan fokus pada beberapa jendela yang penting terlebih dahulu sehingga konsentrasi tidak menjadi pecah, dan kita tidak menjadi bingung. Adapun jendela-jendela yang perlu anda perhatikan adalah sebagai berikut :

1. **Menu Bar**, digunakan untuk memilih tugas-tugas tertentu seperti menyimpan project, membuka project, dll.
2. **Main Toolbar**, digunakan untuk melakukan tugas-tugas tertentu dengan cepat.
3. **Jendela Project**, jendela ini berisi gambaran dari semua modul yang terdapat dalam aplikasi anda. Anda dapat menggunakan icon Toggle Folders untuk menampilkan modul-modul dalam jendela tersebut secara di group atau berurut berdasarkan nama.

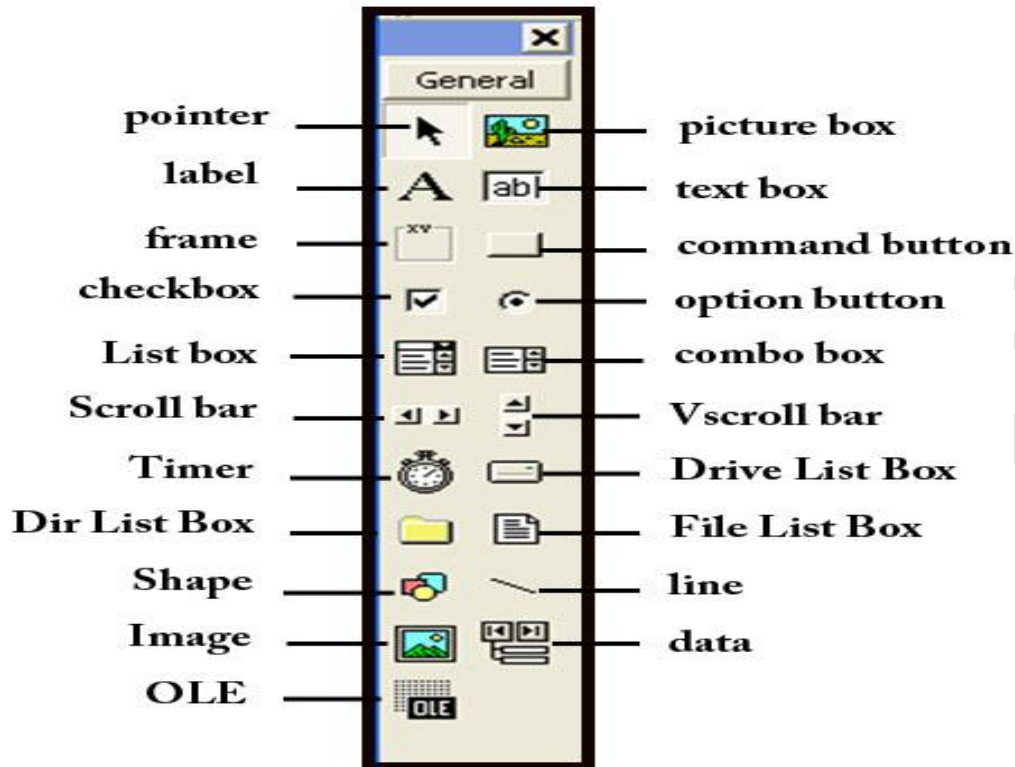
Anda dapat menggunakan Ctrl+R untuk menampilkan jendela project, ataupun menggunakan icon Project Explorer.

4. **Jendela Form Designer**, jendela ini merupakan tempat anda untuk merancang user interface dari aplikasi anda. Jadi jendela ini menyerupai kanvas bagi seorang pelukis.
5. **Jendela Toolbox**, jendela ini berisi komponen-komponen yang dapat anda gunakan untuk mengembangkan user interface.
6. **Jendela Code**, merupakan tempat bagi anda untuk menulis koding. Anda dapat menampilkan jendela ini dengan menggunakan kombinasi Shift-F7.
7. **Jendela Properties**, merupakan daftar properti-properti object yang sedang terpilih. Sebagai contohnya anda dapat mengubah warna tulisan (foreground) dan warna latarbelakang (background). Anda dapat menggunakan F4 untuk menampilkan jendela properti.
8. **Jendela Color Palette**, adalah fasilitas cepat untuk mengubah warna suatu object.
9. **Jendela Form Layout**, akan menunjukkan bagaimana form bersangkutan ditampilkan ketika runtime.

Kemudian, Anda bisa mencoba menutup jendela-jendela tersebut dengan klik di **Close button** dimasing-masing jendele (window) dan dengan mudah anda bisa menampilkan lagi melalui **Menu View**.

1.4 Mengenali Bagian-Bagian Toolbox

Jendela Toolbox merupakan jendela yang sangat penting bagi anda. Dari jendela inilah anda akan memilih control-control (object , komponent) yang nantinya anda perlukan untuk diintegrasikan ke dalam aplikasi anda. Setiap control mewakili fungsi masing-masing . Di bawah ini tampilan gambarnya yaitu:



Gambar 1.4 Bagian-bagian Toolbox

Adapun secara garis besar fungsi dari masing-masing kontrol-kontrol tersebut adalah sebagai berikut :

1. **Pointer** bukan merupakan suatu kontrol; gunakan icon ini ketika anda ingin memilih kontrol yang sudah berada pada form.
2. **PictureBox** adalah kontrol yang digunakan untuk menampilkan image dengan format: BMP, DIB (bitmap), ICO (icon), CUR (cursor), WMF.
3. **Label** adalah kontrol yang digunakan untuk menampilkan teks yang tidak dapat diperbaiki oleh pemakai.
4. **TextBox** adalah kontrol yang mengandung string yang dapat diperbaiki oleh pemakai, dapat berupa satu baris tunggal, atau banyak baris.
5. **Frame** adalah kontrol yang digunakan sebagai kontainer bagi kontrol lainnya.
6. **CommandButton** merupakan kontrol hampir ditemukan pada setiap form, dan digunakan untuk membangkitkan event proses tertentu ketika pemakai melakukan klik padanya.

7. **CheckBox** digunakan untuk pilihan yang isinya bernilai yes/no, true/false.
8. **OptionButton** sering digunakan lebih dari satu sebagai pilihan terhadap beberapa option yang hanya dapat dipilih satu.
9. **ListBox** mengandung sejumlah item, dan user dapat memilih lebih dari satu (bergantung pada property *MultiSelect*).
10. **ComboBox** merupakan kombinasi dari TextBox dan suatu ListBox dimana pemasukkan data dapat dilakukan dengan pengetikkan maupun pemilihan.
11. **HScrollBar** dan **VScrollBar** digunakan untuk membentuk scrollbar berdiri sendiri.
12. **Timer** digunakan untuk proses background yang diaktifkan berdasarkan interval waktu tertentu. Merupakan kontrol non-visual.
13. **DriveListBox**, **DirListBox**, dan **FileListBox** sering digunakan untuk membentuk dialog box yang berkaitan dengan file.
14. **Shape** dan **Line** digunakan untuk menampilkan bentuk seperti garis, persegi, bulatan, oval.
15. **Image** berfungsi menyerupai image box, tetapi tidak dapat digunakan sebagai kontainer bagi kontrol lainnya. Sesuatu yang perlu diketahui bahwa kontrol image menggunakan resource yang lebih kecil dibandingkan dengan PictureBox
16. **Data** digunakan untuk *data binding*
17. **OLE** dapat digunakan sebagai tempat bagi program eksternal seperti Microsoft Excel, Word, dll.

1.5 Menggunakan Project Explorer

Didalam project explorer ini dapat terdiri dari beberapa form. Form digunakan untuk mendesain program-program yang bakal kita buat. Didalam project tersebut kita menampilkan properties yang sedang aktif. Property adalah data-data atau karakteristik yang dimiliki suatu objek komponen.

Setelah kita mengenali berbagai fitur-fitur yang ada dalam program Visual basic 6.0 tersebut maka kita akan membuat suatu studi kasus yang berupa latihan membuat program yang sangat sederhana agar kita dapat memahami program Visual Basic 6.0 ini lebih mendalam.

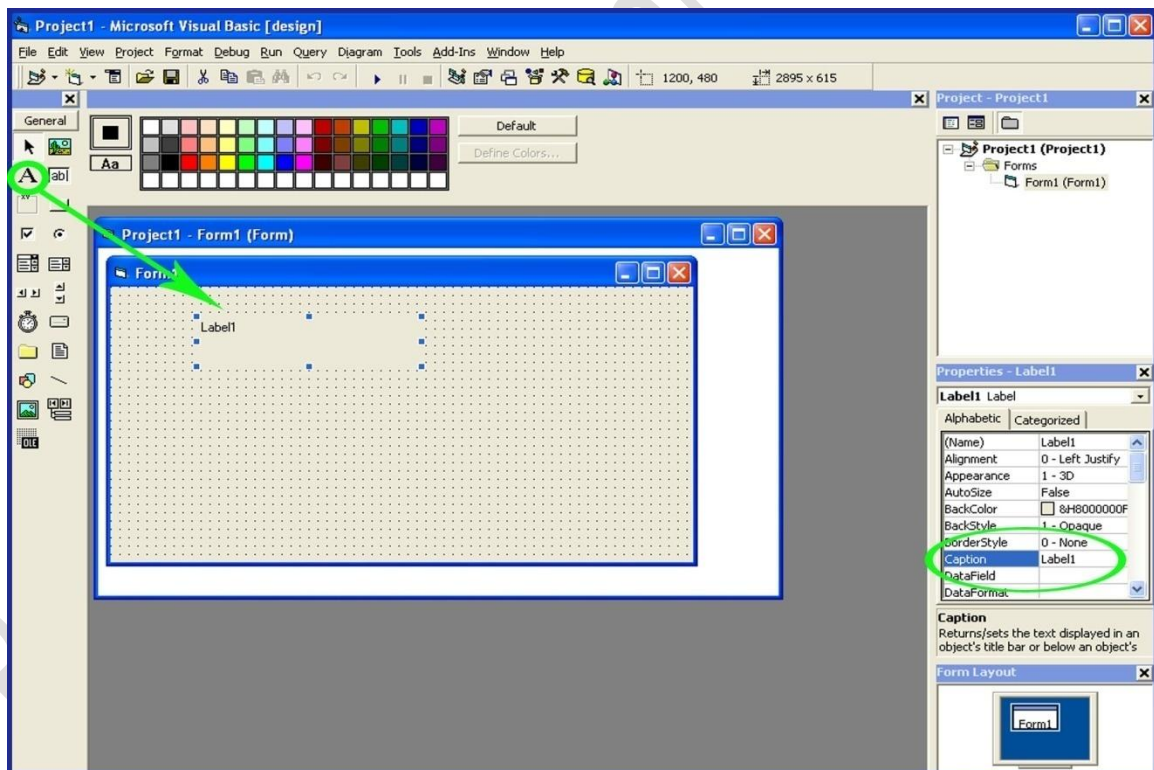
Dalam modul ini akan ditampilkan tiga latihan yang disertai langkah-langkah membuatnya. Adapun latihan pertama yaitu sebagai berikut:

1. Latihan Pertama

Membuat Program “*Welcome UPI Guyss!!*”

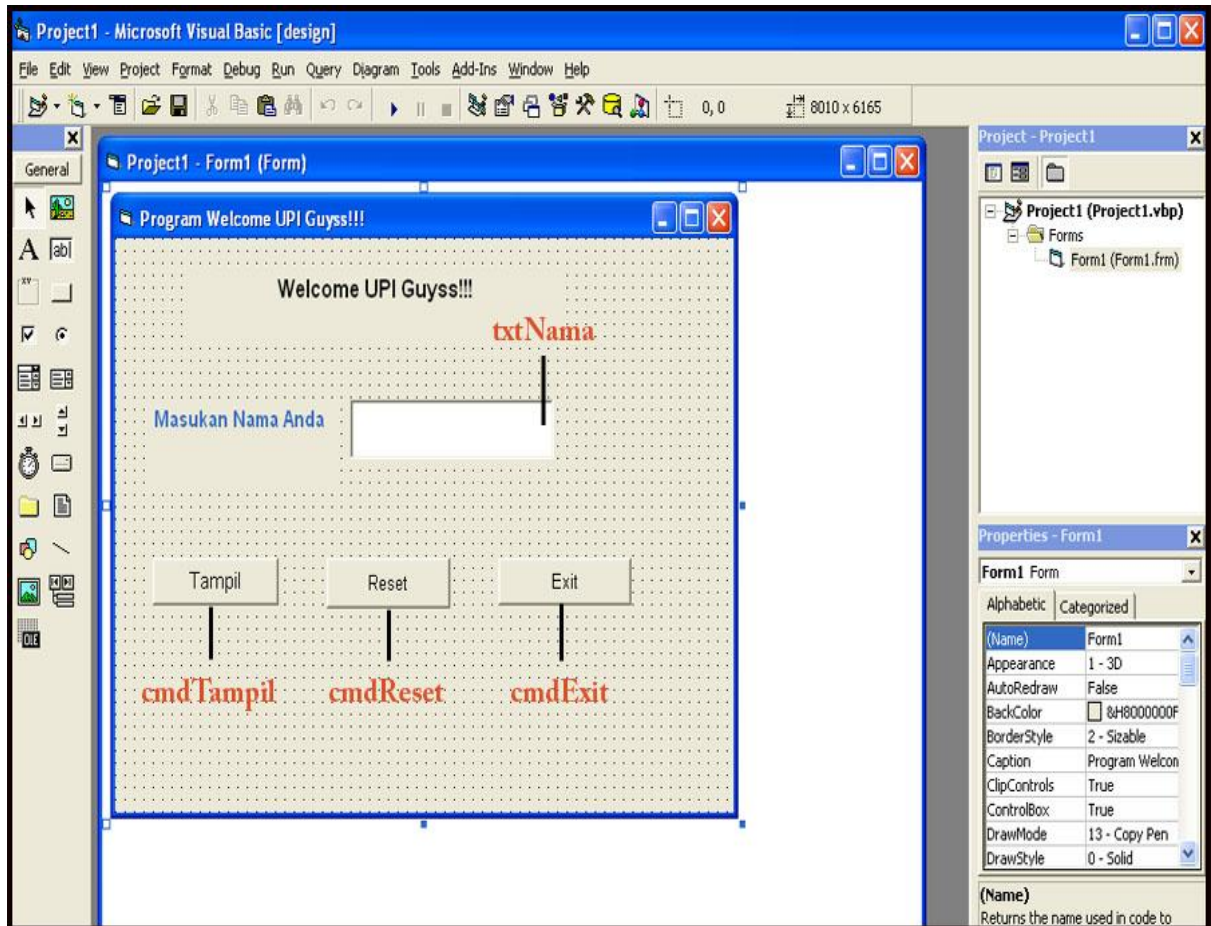
Adapun Langkah-Langkah membuatnya yaitu:

1. Buka Program Visual Basic 6.0 , yaitu **Start**, pilih **Microsoft Visual Basic 6.0** dan akhirnya pilih shortcut **Microsoft Visual Basic 6.0** seperti contoh **Gambar 1.1 Tampilan Visual Basic 6.0**.
2. Kemudian Pilih **Standard EXE**, klik **open** seperti contoh pada **Gambar 1.2 Membuka Standar Exe**
3. Kemudian akan muncul Tampilan Window Project seperti gambar di bawah ini: Buatlah label seperti contoh gambar di bawah ini, Pilihlah **Label** pada Toolbox yang menggunakan simbol **A**. Kemudian drag ke form design.



Gambar 1.5 Langkah awal membuat program “*Welcome UPI Guyss!!*”

4. Kemudian pada kolom Properties ubah property **Caption** untuk mengubah teks yang tertulis pada layar seperti gambar di bawah ini;



Gambar 1.6

Berikut ini adalah property yang akan kita setting pada form diatas:

objek	Property	Nilai
Form1	Caption	Program Welcome UPI Guyss!!
Label 1	Caption	Welcome UPI Guyss!!
Label 2	Caption	Masukan Nama Anda
Text Box	Caption Name	- txtnama
Comand Button 1	Caption Name	Tampil Cmdtampil
Comand Button 2	Caption Name	Reset Cmdreset
Comand Button 3	Caption Name Name	Exit Cmdexit

Telah dijelaskan sebelumnya bahwa pemrograman pada Visual Basic menggunakan konsep event-driven, di mana suatu prosedur dijalankan pada saat terjadi suatu kejadian tertentu. Misalkan kita ingin membuat pada saat tombol Exit diklik maka langsung keluar dari program. Caranya:

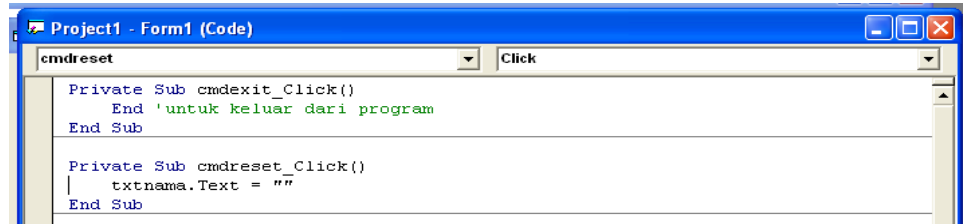
5. **Kliklah dua kali pada tombol EXIT** dan akan membawa anda ke layar coding. Didalam layar coding tersebut kita harus memasukan syntax-syntax yang dapat menjalankan program tersebut. Adapaun penulisannya dapat dilihat pada gambar di bawah ini:

```

Project1 - Form1 (Code)
cmdexit Click
Private Sub cmdexit_Click()
    End 'untuk keluar dari program
End Sub
    
```

Gambar 1.7 Tampilan membuat syntax cmdExit

6. Begitu pula seterusnya pada **tombol RESET**, klik dua kali pada tombol yang kita buat. Dan tuliskan syntax-syntax seperti pada gambar di bawah ini:

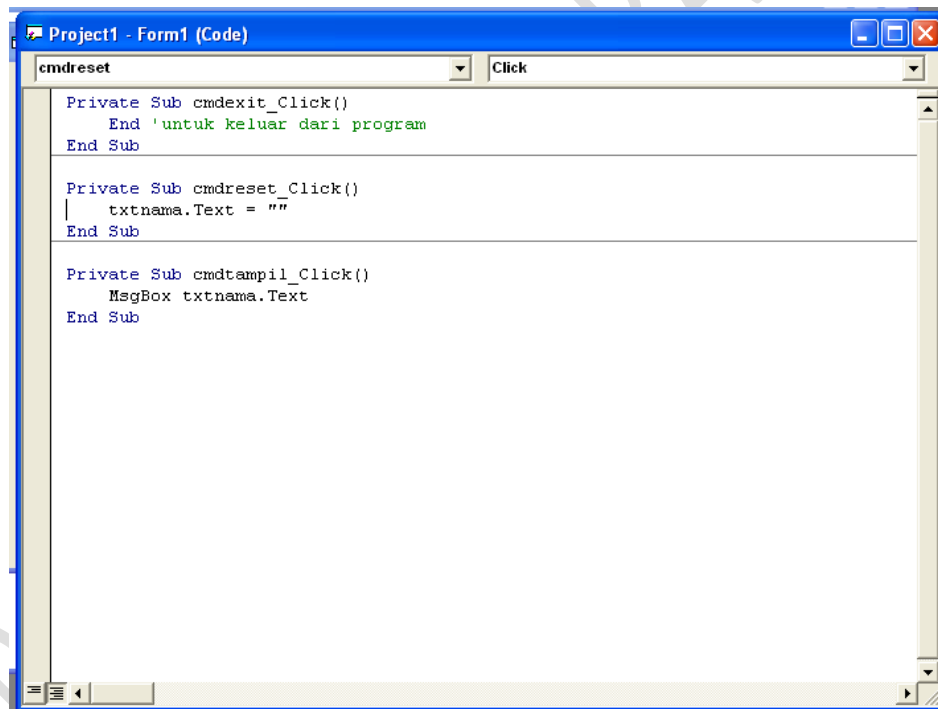


```
Project1 - Form1 (Code)
cmdreset Click
Private Sub cmdexit_Click()
    End 'untuk keluar dari program
End Sub

Private Sub cmdreset_Click()
    txtnama.Text = ""
End Sub
```

Gambar 1.8 Tampilan membuat syntak cmdReset

7. Dan terakhir **tombol TAMPIL**, klik dua kali pada tombol yang telah kita buat. Kemudian tuliskan syntax-syntax seperti gambar di bawah ini:




```
Project1 - Form1 (Code)
cmdreset Click
Private Sub cmdexit_Click()
    End 'untuk keluar dari program
End Sub

Private Sub cmdreset_Click()
    txtnama.Text = ""
End Sub

Private Sub cmdtampil_Click()
    MsgBox txtnama.Text
End Sub
```

Gambar 1.9 Tampilan membuat syntak cmdTampil

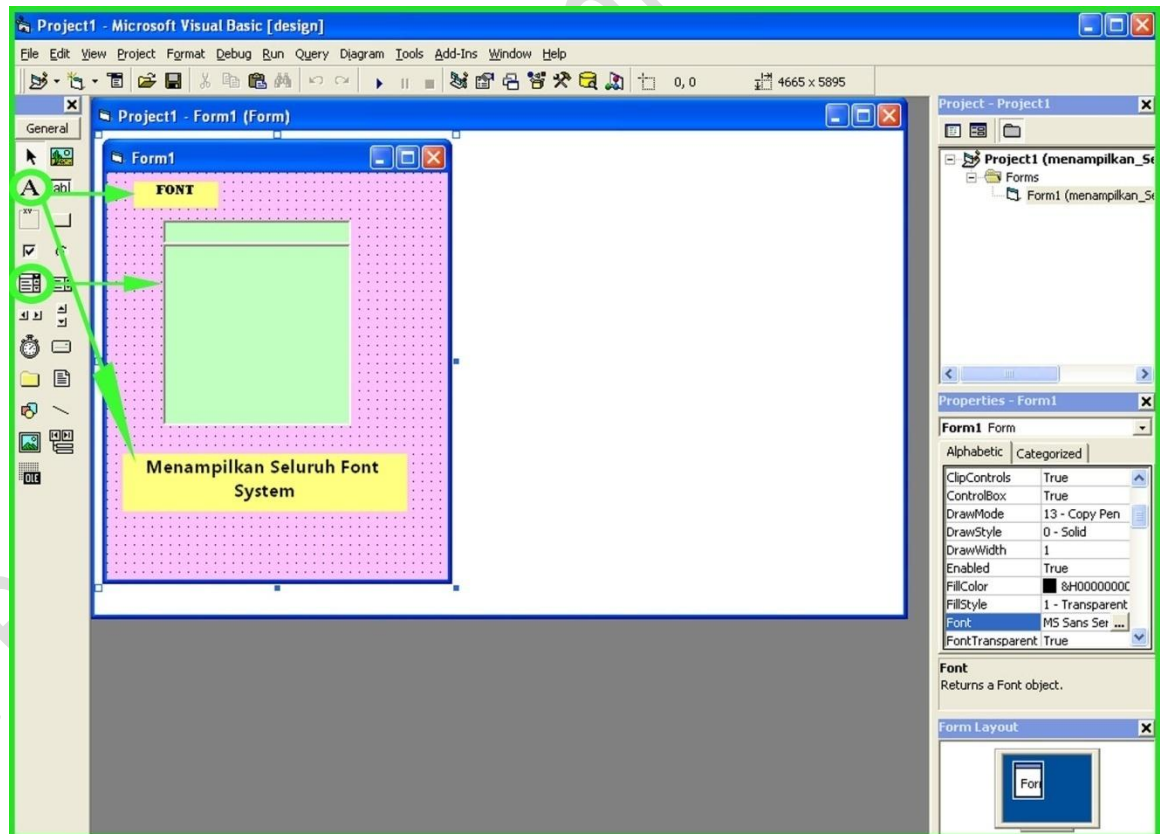
8. Setelah menuliskan syntax-syntax pada program tersebut kemudian jalankan program tersebut, dengan mengklik  pada **main toolbar**.

2. Latihan Kedua

Membuat Program “Menampilkan Seluruh Font System”

Adapun Langkah-Langkah membuatnya yaitu:

1. Buka Program Visual Basic 6.0 dan mulailah dengan form baru, yaitu **Start**, pilih **Microsoft Visual Basic 6.0** dan akhirnya pilih shortcut **Microsoft Visual Basic 6.0** seperti contoh **Gambar 1.1 Tampilan Visual Basic 6.0**.
2. Kemudian Pilih **Standard EXE**, klik **open** seperti contoh pada **Gambar 1.2 Membuka Standar Exe** pada awal materi.
3. Setelah membuka form , masukkan dua komponen **Label** dan sebuah komponen **ComboBox** dari jendela **Toolbox** ke dalam area form, kemudian aturlah properties **Style** komponen **Combo 1** menjadi **1-Simple** sehingga tampak seperti gambar di bawah ini:

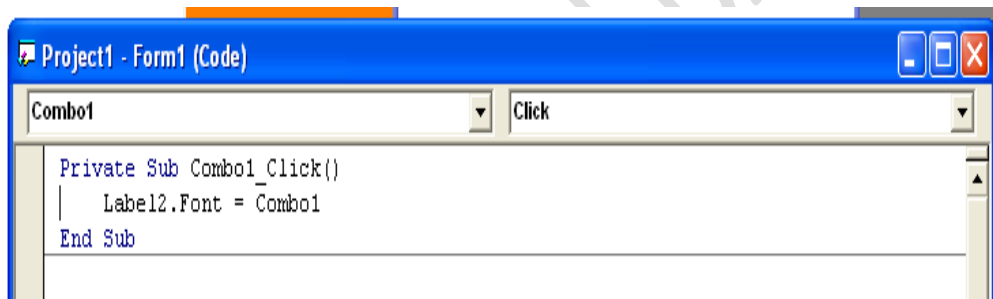


Gambar 1.11

Berikut ini adalah property yang akan kita setting pada form diatas:

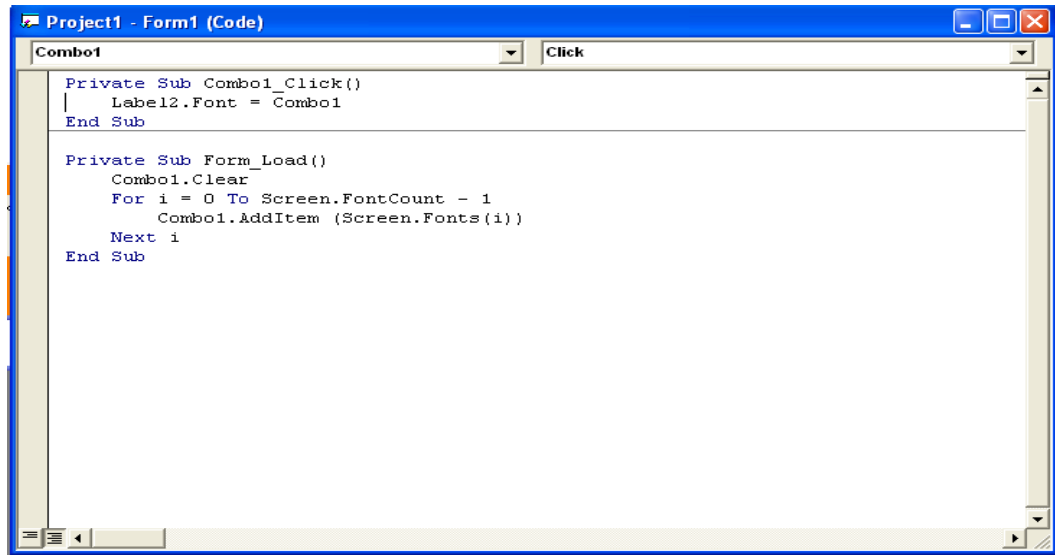
Objek	Property	Nilai
Form 1	Caption	Form 1
Label 1	Caption	Font
Label 2	Caption	Menampilkan seluruh Font System
	Name	Label 2
ComboBox	Name	Combo1s

- Selanjutnya **klik ganda komponen Combo 1**. Ubahlah **Event Change** menjadi **event Click**, dan ketiklah perintah pada gambar di bawah ini untuk mengubah font pada komponen label 2 sesuai dengan font yang Anda pilih pada komponen Combo 1.




Gambar 1.12

- Kemudian ke **tampilan desain form** selanjutnya klik ganda di dalam area form, kemudian ketiklah perintah di bawah ini ke dalam prosedur **Form_Load**.



Gambar 1.13

6. Untuk melihat hasilnya tekan tombol F5 pada keyboard anda atau dengan mengklik  pada *main toolbar*. Kemudian akan muncul program seperti gambar di bawah ini:




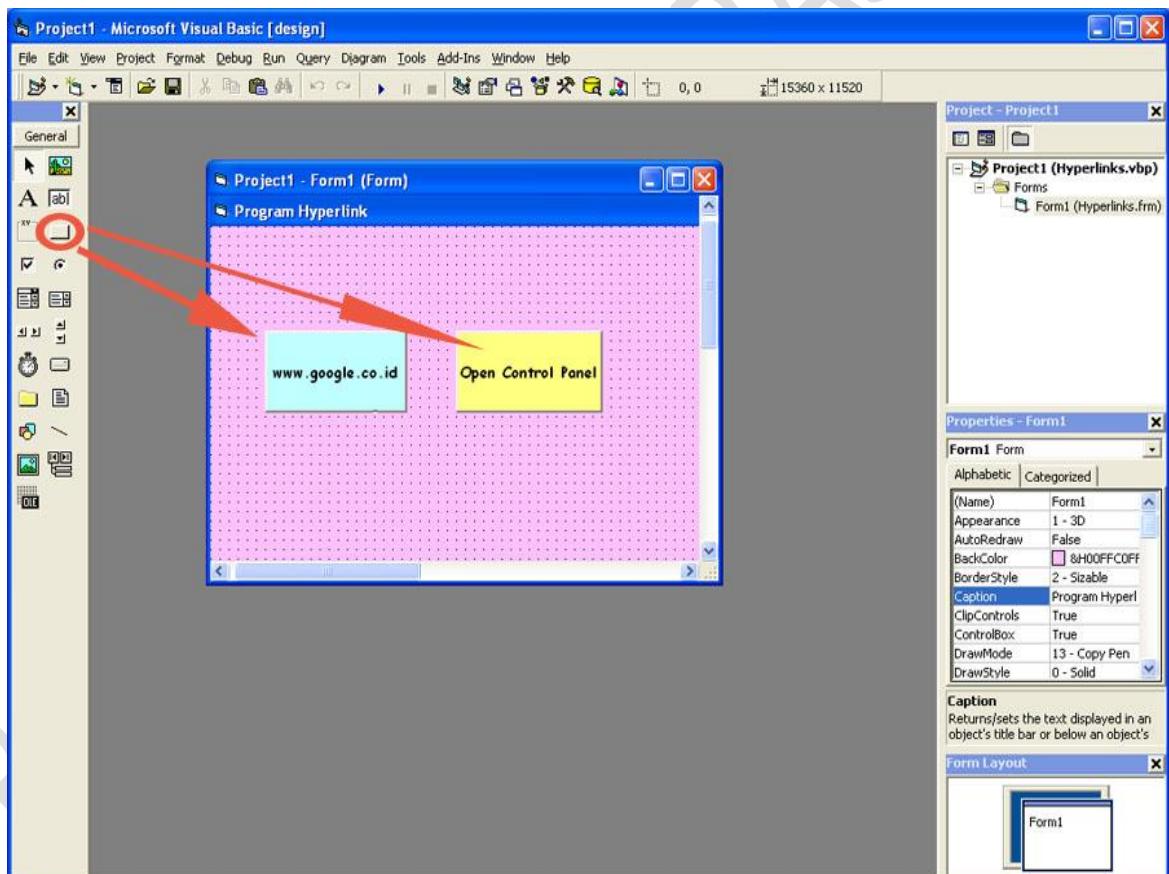
Gambar 1.14

3. Latihan Ketiga

Membuat Program “Hyperlinks”

Adapun Langkah-Langkah membuatnya yaitu:

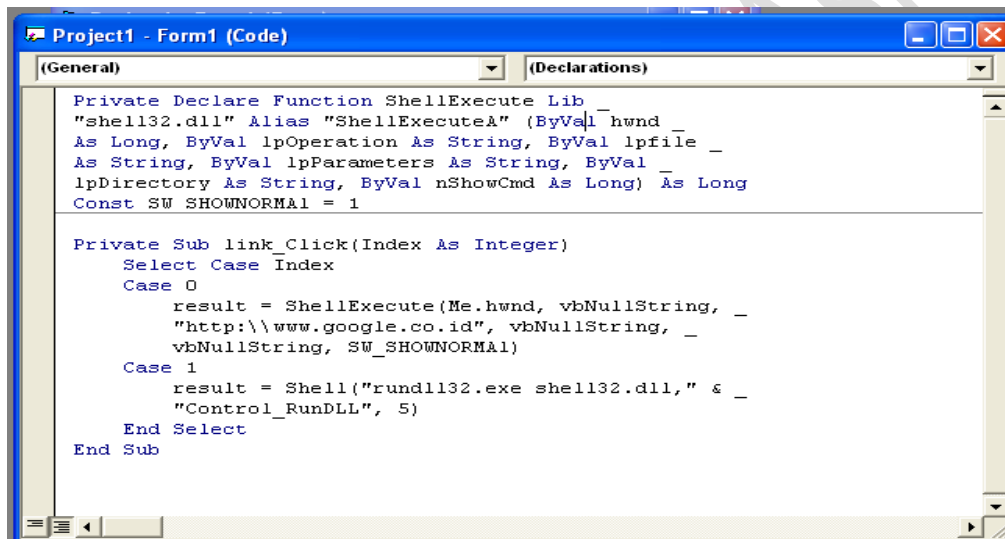
1. Buka Program Visual Basic 6.0 dan mulailah dengan form baru, yaitu **Start**, pilih **Microsoft Visual Basic 6.0** dan akhirnya pilih shortcut **Microsoft Visual Basic 6.0** seperti contoh **Gambar 1.1 Tampilan Visual Basic 6.0**.
2. Kemudian Pilih **Standard EXE**, klik **open** seperti contoh pada **Gambar 1.2 Membuka Standar Exe** pada awal materi.
3. Selanjutnya , **masukan dua komponen Comand Button**  ke dalam area form dan ubah properties nama pada masing-masing komponen menjadi **link**. Adapun cara membuat tampilan dua komponen Comand Button tersebut, dapat dilihat pada contoh gambar di bawah ini



Ubahlah property sesuai pada tabel di bawah ini:


Objek	Property	Nilai
Form 1	Caption	Program Hyperlink
Comand Button1	Caption	www.google.co.id
	Name	link
Comand Button 2	Caption	Open Control Panel
	Name	link

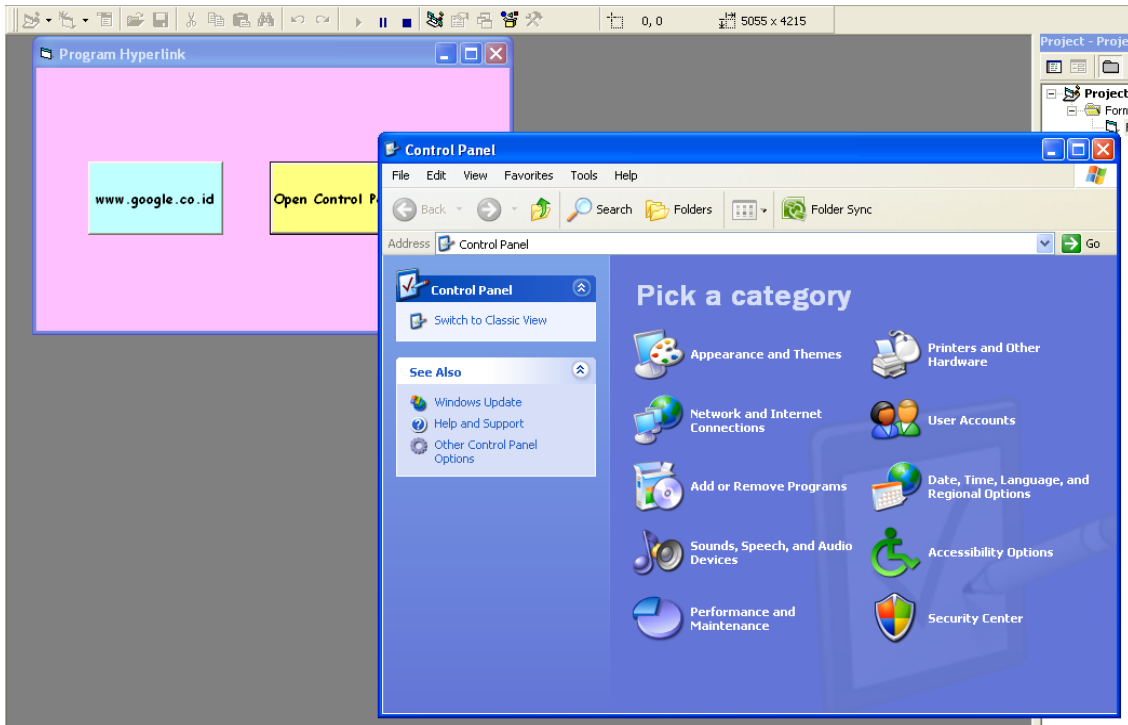
4. Kemudian, klik ganda pada salah satu tombol dan ketiklah perintah di bawah ini pada bagian *General* dan prosedur *link_click*.



```
Project1 - Form1 (Code)
(General) (Declarations)
Private Declare Function ShellExecute Lib _
    "shell32.dll" Alias "ShellExecuteA" (ByVal hwnd _
    As Long, ByVal lpOperation As String, ByVal lpfile _
    As String, ByVal lpParameters As String, ByVal _
    lpDirectory As String, ByVal nShowCmd As Long) As Long
Const SW_SHOWNORMAL = 1

Private Sub link_Click(Index As Integer)
    Select Case Index
        Case 0
            result = ShellExecute(Me.hwnd, vbNullString, _
                "http:\\www.google.co.id", vbNullString, _
                vbNullString, SW_SHOWNORMAL)
        Case 1
            result = Shell("rundll32.exe shell32.dll," & _
                "Control_RunDLL", 5)
    End Select
End Sub
```


5. Untuk melihat hasilnya tekan tombol F5 pada keyboard anda atau dengan mengklik  pada *main toolbar*. Kemudian akan muncul program seperti gambar di bawah ini:



BAB II

TIPE DATA

Saat kita menggunakan suatu program komputer, maka komputer akan meminta informasi yang kemudian akan disimpan dan diolah. Informasi tersebut dinamakan data. Tipe data merupakan pola representasi suatu data di komputer dan berfungsi untuk mendefinisikan objek yang akan diprogram. Tipe data menentukan representasi internal data atau suatu nilai, selain itu tipe data digunakan untuk menyimpan dan merepresentasikan data di dalam aplikasi.

Data memiliki tipe yang berbeda-beda dan biasanya data dikelompokkan pada kelompok yang sejenis agar tidak terjadi operasi matematika diantara data yang berbeda jenis tersebut, contohnya kita tidak bisa menjumlahkan nilai suatu jarak (Km) dengan Massa (Kg) karena keduanya memiliki tipe data yang berbeda. Begitu juga dalam suatu program, setiap nilai harus dikelompokkan pada jenis-jenis tertentu yang disebut dengan tipe data.

Tipe data Variant akan menyajikan semua jenis data yang didefinisikan pada visual basic. Jika sebuah variabel dideklarasikan tanpa perintah As, maka standar dari tipe data yang digunakan adalah variant.

Dim Jumlah, Contoh1, Contoh2

Macam-macam tipe data yang terdapat dalam visual basic

Tipe Data	Ukuran Storage	Range
Byte	1 byte	0 s.d 225
boolean	2 bytes	Benar atau salah
integer	2 bytes	-32,768 s.d 32,767
Long integer	4 bytes	-2,147,483,648 s.d 2,147,483,647
Single negatif	4 bytes	-3,402823E+38 s.d 1.401298E-45
Single positif	4 bytes	1.401298E-45 s.d 3,402823E+38
Double negatif	8 bytes	- 1.79769313486232E+308 s.d - 4.94065645841247E-324
Double positif	8 bytes	4.94065645841247E-324 s.d 1.79769313486232E+308
currency	8 bytes	- 922,337,203,685,477.5808 s.d 922,337,203,685,477.5807
Date	8 bytes	1 January 100 s.d 31 Desember 9999
object	4 bytes	Mengacu pada objek tertentu
String	10 bytes	0 s/d 2 milyar karakter
Variant null	16 bytes	Error dan tipe seluruh tipe data yang lain

Tipe-tipe data diatas terbagi ke dalam 3 kategori tipe data, yaitu:

1. Tipe Data Numerik

Visual Basic menyediakan beberapa tipe data numerik — Integer, Long (long integer), Single (single-precision floating point), Double (double-precision floating point), dan Currency. Dengan menggunakan tipe data numerik secara umum menggunakan ruang penyimpanan yang lebih kecil dibandingkan jika Anda menggunakan tipe data Variant.

Jika Anda mengetahui bahwa sebuah variabel akan selalu menyimpan seluruh angka (seperti 12) daripada nomor dengan desimal (seperti 3.57), deklarasikan dia sebagai **Integer** atau tipe **Long**. Operasi lebih cepat dengan Integer, dan tipe ini menggunakan sedikit memory dibandingkan dengan tipe data lainnya. Mereka khususnya berguna sebagai variabel counter di dalam blok perulangan (loop) seperti **For...Next**.

Jika sebuah variabel mengandung data yang memiliki nilai desimal, deklarasikan variabel tersebut sebagai **Single**, **Double**, atau **Currency**. Tipe data **Currency** mendukung sampai 4 digit ke kanan setelah tanda pemisah desimal dan 15 digit ke kiri; tipe ini merupakan tipe data yang akurat dan cocok untuk perhitungan uang. Angka Floating-point (Single dan Double) memiliki batas bawah dan akhir (range) yang lebih besar dibandingkan dengan **Currency**, tapi dapat mengikuti kesalahan pembulatan yang kecil.

Nilai Floating-point dapat dinyatakan sebagai **mmmEeee** atau **mmmDeee**, di mana **mmm** adalah mantissa dan **eee** adalah eksponen (pangkat 10). Nilai positif tertinggi dari sebuah tipe data Single adalah 3,402823E+38, atau 3,4 kali 10 pangkat 38; nilai positif tertinggi dari sebuah tipe data Double adalah 1,79769313486232D+308, atau sekitar 1,8 kali 10 pangkat 308. Penggunaan **D** untuk memisahkan mantissa dan exponent dalam terjemahan angka menyebabkan nilai diperlakukan sebagai tipe data Double. Demikian juga, penggunaan **E** dalam bentuk yang sama memperlakukan nilai sebagai tipe data Single.

2. Tipe Data Byte

Jika variabel terdiri dari data binary, deklarasikanlah sebagai sebuah array dari tipe data **Byte**. Penggunaan variabel Byte untuk menyimpan data binary dan menjaganya selama konversi format. Ketika variabel String dikonversi di antara format ANSI dan Unicode, data binary apapun di variabel tersebut akan berkurang. Visual Basic dapat secara otomatis mengkonversi antara ANSI dan Unicode ketika:

- a. Pembacaan dari file
- b. Menulis ke file
- c. Pemanggilan DLL
- d. Pemanggilan method dan properties pada object

Semua operator yang bekerja pada **Integer** bekerja dengan tipe data **Byte** kecuali minus. Karena Byte merupakan sebuah tipe yang *unsigned* dengan range 0-255, dia tidak dapat mewakili angka negatif. Jadi, untuk angka minus, Visual Basic memaksa terlebih dulu Byte ke sebuah Integer yang ditandai.

Semua variabel numerik dapat saling diisi ke variabel yang lainnya dan ke variabel yang bertipe Variant. Visual Basic menonaktifkan pembulatan daripada memotong sebagian kecil dari angka floating-point sebelum menugaskannya ke sebuah Integer.

3. Tipe Data String

Jika Anda memiliki sebuah variabel yang selalu mengandung sebuah String dan tidak pernah terdiri dari angka, Anda dapat mendeklarasikannya sebagai tipe String:

```
Private S As String
```

Anda kemudian dapat menugaskan String ke variabel ini dan memanipulasinya dengan menggunakan fungsi-fungsi yang terkait dengan String:

```
S = "Database"
```

```
S = Trim(S)
```

Secara default, sebuah variabel String atau argumen adalah String dengan panjang yang tidak tetap. String dapat berkembang atau menyusut sesuai dengan data baru yang Anda tugaskan kepadanya. Anda juga dapat mendeklarasikan String dengan panjang yang tetap. Anda dapat menentukan String dengan panjang yang tetap menggunakan sintaks berikut:

```
String * size
```

Sebagai contoh, untuk mendeklarasikan sebuah String yang selalu panjangnya 50 karakter, gunakanlah kode seperti ini:

```
Dim EmpName As String * 50
```

Jika Anda menugaskan sebuah String yang lebih kecil dari 50 karakter, EmpName diisi dengan ruang spasi yang cukup sehingga mencapai 50 karakter. Jika Anda menugaskan sebuah String yang lebih besar daripada panjang String, maka Visual Basic akan otomatis memotong karakter yang lebih tersebut. Karena String yang panjangnya tetap diisi dengan tanda spasi, Anda dapat menggunakan fungsi **Trim** dan **RTrim**, yang akan menghapus spasi. Sangat berguna ketika dibutuhkan.

String yang panjangnya tetap di dalam standard module dapat dideklarasikan sebagai **Public** atau **Private**. Di form dan class module, string yang panjangnya tetap harus dideklarasikan sebagai **Private**.

VARIABEL

Variabel adalah tempat dalam memori komputer yang diberi nama (sebagai pengenalan) dan dialokasikan untuk menampung data. Sesuai data yang ditampung maka variabel harus mempunyai tipe data yang sesuai dengan isinya. Untuk mendeklarasikan suatu variabel digunakan dua perintah, yaitu:

- 1) Dengan menggunakan perintah-perintah Dim, Private, Static, dan Public yaitu mendeklarasikan nama variabel beserta tipe-tipe datanya pada awal procedure.

Contoh:

```
Dim Nama as String
```

```
Dim Alamat as String * 30
```

```
Dim Gaji as Long
```

- 2) Dengan menggunakan deklarasi Implisit, Contoh:

```
Nama$ = "Ivananda"
```

```
Alamat$ = "Jl. Skarda N II No. 13"
```

```
Gaji& = 500000
```

Ruang Lingkup Variabel (variable scope) adalah ruang atau daerah dimana variabel yang dibuat dikenal. Ada variabel yang dapat dikenal di seluruh bagian program dan hidup selama program berjalan, contohnya Static dan Public, tetapi ada juga variabel yang hanya dikenal di procedure atau fungsi tempat variabel tersebut dibuat dan hidup hanya selama procedure atau fungsi tersebut berjalan, contohnya Dim dan Private.

Kelebihan dari visual basic adalah bahwa setiap obyek baik berupa form, report (laporan), data atau komponen dapat menjadi variabel. Variabel di dalam visual basic dibedakan menjadi 3 macam variabel yaitu variabel lokal, variabel global terhadap form dan variabel global terhadap aplikasi (project).

- 1) Variabel lokal: adalah variabel yang hanya aktif dalam suatu fungsi atau subroutine di dalam visual basic. Caranya dituliskan secara langsung di dalam suatu fungsi atau subroutine tanpa pendefinisian atau dengan definisi variabel di dalam fungsi.

- 2) Variabel global dalam form: adalah variabel yang aktif selama satu form berjalan, bila pindah ke form yang lain maka variabel ini tidak aktif. Ini dapat dilakukan dengan mendefinisikan variabel di luar subroutine, biasanya diletakkan pada baris paling atas dari suatu form. Contohnya **Dim a as integer** diletakkan pada baris paling atas, maka variabel a selalu aktif selama form aktif.
- 3) Variabel global dalam aplikasi: adalah variabel yang aktif selama aplikasi masih aktif. Variabel ini masih aktif meskipun form yang berjalan sudah berganti. Untuk mendefinisikan variabel ini dilakukan dengan menambahkan modul dan pendefinisian dengan **global**.

Syarat-syarat pemberian nama sebuah variabel dalam Visual Basic adalah:

1. Tidak lebih dari 40 karakter.
2. Terdiri atas huruf, angka, dan underscore (_).
3. Karakter pertama harus sebuah huruf.
4. Tidak boleh menggunakan kata yang sudah didefinisikan oleh VB (*reserved word*).

Tingkatan Variabel

- 1) Tingkatan Prosedur/Event

Di dalam prosedur, variabel dideklarasikan dengan pernyataan “Dim”. Variabel ini hanya berlaku dalam prosedur tersebut dan di luarnya tidak berlaku lagi.

Contoh:

Dim usia as Integer

- 2) Tingkatan Form

Dideklarasikan dalam objek General dengan event declaration dengan pernyataan “Dim”.

Variabel ini akan dikenal oleh seluruh prosedur/event yang terdapat dalam form.

Contoh:

Dim tanggal as Date

3) Tingkatan

Aplikasi

Dideklarasikan menggunakan module dan diletakkan pada bagian declaration dengan sub general pada module tersebut. Variabel ini akan dikenal oleh seluruh prosedur dalam aplikasi yang dibuat. Variabel ini dibuat menggunakan pernyataan “Public”.

KONSTANTA

Konstanta adalah variabel yang memiliki nilai yang tetap. Variabel konstanta cukup diperlukan dalam beberapa situasi tertentu karena terkadang kita lupa kode-kode dalam Visual Basic. Misalnya, kode warna hitam adalah “&H0000F”. Dengan variabel konstanta ini, kita dapat memasukkan kode ini ke dalamnya, sehingga untuk memanggil warna hitam, kita hanya perlu mengetikkan variabel konstantanya.

Tingkatan variabel ini tergantung pada pernyataan yang digunakan dan lokasi pendeklarasiannya. Untuk mendeklarasikan variabel konstanta, kita menggunakan pernyataan “Const”.

Contoh:

```
Public Const hitam = "&H0000F"
```

Dengan begitu, untuk memanggil warna hitam, kita cukup mengetikkan “hitam”. Contoh:

```
Private Sub Form_Load()  
Me.BackColor = hitam  
End Sub
```

Sumber untuk sebuah konstanta ada 2 yaitu:

1. Konstanta Intrinsik adalah konstanta yang telah disediakan oleh aplikasi dan kontrol yang ada dalam Visual Basic. Misalnya vbRed, vbTileVertical dll.
2. Konstanta Symbolic atau User-Defined, yaitu konstanta yang dideklarasikan

sendiri oleh user dengan menggunakan statemen Const. Misalnya:

```
Const conPi = 3.14159
```

```
Public conHarga As Integer = 10000
```

Konstanta mirip dengan variabel. Bedanya nilai konstanta tidak pernah berubah sejak dideklarasikan. Untuk mendeklarasikan konstanta dibutuhkan fungsi define()
Sintak : define("nama konstanta",nilai)

Sekarang Anda sudah memahami penggunaan variabel dan konstanta. Di pelajaran Matematika, variabel dan konstanta juga ada, hanya saja penggunaannya yang berbeda. Di Visual Basic, variabel bisa saja ditulis seperti berikut.

$a = a + 1$

Artinya adalah mengubah isi dari variabel a menjadi isi variabel a yang lama ditambah 1. Dalam Matematika, yaitu persamaan linear satu variabel, pernyataan ini tidak dibenarkan karena berapa pun nilai a, persamaan tersebut tidak akan menjadi benar. Misalnya nilai a adalah 5. Bila $a = a + 1$, maka itu artinya $5 = 5 + 1$, yaitu $5 = 6$. Tidak benar, bukan? Kalau begitu, apakah Visual Basic menyalahi aturan dalam Matematika? Jawabannya adalah tidak. Tanda sama dengan (=) di Visual Basic dalam kasus ini adalah operator penunjukkan (assignment), yaitu operator untuk mengisi nilai di sebelah kanan operator ini, ke variabel di sebelah kiri operator ini.

Pengenalan Visual Basic 6.0

OPERATOR

Operator adalah suatu tanda yang digunakan untuk menghubungkan satu variabel atau konstanta dengan variabel atau konstanta lain dengan tujuan melakukan berbagai manipulasi dan pengolahan data. Pada Visual Basic 6.0 terdapat bermacam-macam operator, yaitu:

1) Operator Penugasan (Assignment)

Operator penugasan disimbolkan dengan tanda sama dengan (=) dan berfungsi memasukkan suatu data ke dalam suatu variabel.

Contoh:

```
Harga = 2500  
Nama = "Anita"
```

2) Operator Aritmatika

Operator aritmatika digunakan untuk melakukan operasi aritmatika. Operator aritmatika mempunyai hirarki paling tinggi dibanding operator pembandingan dan operator logika.

Penulisan operator aritmatika dengan hirarki dari paling tinggi ke paling rendah, sebagai berikut:

Operator	Operasi
^	Pemangkatan
-	Tanda Negatif
*	Perkalian
/	Pembagian
\	Pembagian Integer
Mod	Modulus
+	Penambahan
-	Pengurangan
+,&	Penggabungan string

Contoh:

$$A = (5 + 5) * 7, B = (100 \text{ Mod } 5)$$

3) Operator Pembandingan (Relasi)

Operator pembandingan digunakan untuk membandingkan suatu data (ekspresi) dengan data (ekspresi) lain dan menghasilkan nilai logika (boolean) Benar tau Salah. Bentuk dari operator pembandingan seperti yang terpampang pada tabel dibawah ini:

Operator	Operasi
=	Sama dengan
<>	Tidak sama dengan
<	Kurang dari
>	Lebih dari
<=	Kurang dari atau sama dengan
>=	Lebih dari atau sama dengan
Like	Mempunyai cirri yang sesuai
Is	Sama referensi objek

Contoh:

$5 > 3$, $5 + 7 = 12$, $"abc" = "ab" + "c"$

4) Operator Logika

Operator logika digunakan untuk mengekspresikan satu atau lebih data (ekspresi) logika (boolean) yang menghasilkan data logika baru. Tabel operator logika dengan hirarki dari atas ke bawah adalah sebagai berikut:

Operator	Keterangan
Not	Tidak
And	Dan
Or	Atau
XOr	Exclusive Or
Eqv	Ekivalen
Imp	Implikasi

EKSPRESI

Ekspresi adalah suatu cara penulisan untuk memberikan atau memasukkan nilai kedalam variabel. Ekspresi secara umum dalam *computer statement* dituliskan sebagai:

Variabel = Nilai

Di dalam Visual Basic ekspresi menggunakan tanda sama dengan (=). Dengan aturan sebelah kiri adalah variabel penampung (hasil) dan sebelah kanan adalah nilai yang dimasukkan ke variable. Sebagai contoh untuk memasukkan nilai 5 ke dalam variabel a dapat dilakukan dengan **a=5**. Atau memasukkan nama 'widya' ke dalam variabel nama dapat dituliskan dengan **nama="widya"** (Pada tipe data string, penulisan diberi tanda petik dua). Atau untuk memasukkan nilai keputusan benar dapat dituliskan dengan **keputusan=True**, Pada tipe data boolean nilainya hanya True dan False. Sehingga dapat dikatakan bahwa cara memasukkan nilai ini sangat tergantung dari jenis nilai apakah itu boolean, numerik, string atau date. Sedangkan format bilangan dan tanggal secara default menggunakan format yang ada pada sistem operasi, meskipun tidak menutup kemungkinan diubah menggunakan format tersendiri.

Berikut ini contoh-contoh penulisan ekspresi untuk memasukkan nilai ke dalam variabel:

Ekspresi	Arti dari Ekspresi
<code>a= 10.5</code>	Nilai 10.5 dimasukan kedalam variable a, tanda titik berarti nilai pecahan dalam desimal.
<code>Tg="10-02-2009"</code>	Memasukan tanggal 10 februari 2009 ke dalam varibel tg. Format tanggal secara umum menggunakan format tanggal system operasi yaitu bulan- tanggal- tahun, sebelum melakukan ekspresi ini sebaiknya deidefinisikan varibel tg dengan cara Dim tg as date
<code>Nrp= "74030010"</code>	Memasukan format string "7403030010" dalam variable Nrp
<code>Ketemu= True</code>	Memasukan nilai True dalam varibel Boolean Ketemu

Ekspresi merupakan suatu proses yang bersifat *sequential*, yang artinya bahwa proses dilakukan dari baris paling atas sampai baris terakhir. Sebagai contoh bila dituliskan:

```
a=10
a=5
```


Maka artinya pada baris pertama a bernilai 10, dan pada baris kedua a bernilai 5, sehingga nilai 10 diganti dengan nilai 5. Sehingga hasilnya a bernilai 5.

Ekspresi bukan hanya seperti diatas, tetapi dapat juga merupakan *penulisan* suatu formula dengan melibatkan variabel-variabel yang sudah ada sebelumnya.

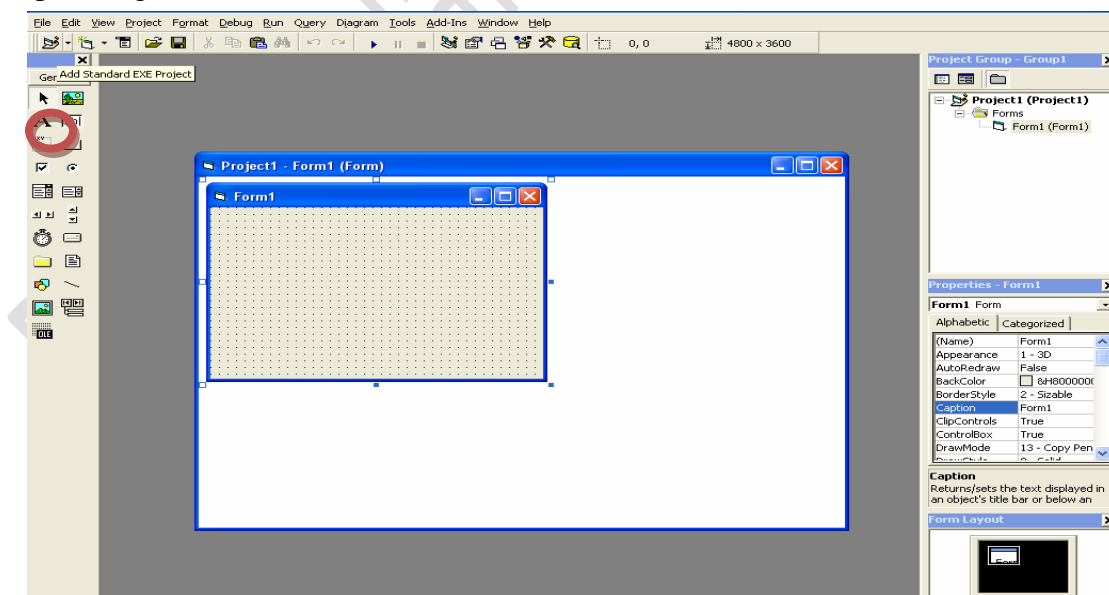
Contoh 1:

```
a=5
b=10
c=a+b
```

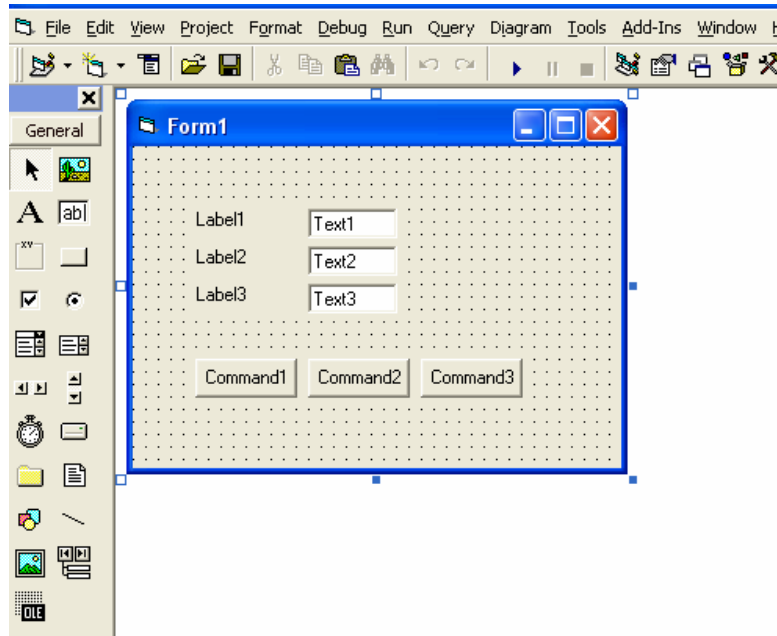
Hasilnya variabel a bernilai 5, b bernilai 10 dan c bernilai 15 sebagai hasil dari $a+b=5+10=15$.

Membuat Penjumlahan, Pengurangan dan Perkalian Untuk membuat project yang menyajikan operasi aritmatika sederhana seperti penjumlahan, pengurangan dan perkalian dapat dilakukan dengan langkah-langkah sebagai berikut:

- (1) Membuat project baru, dengan click ikon add standard EXE pada Visual Basic yang adalah pada bagian sebelah kiri toolbar



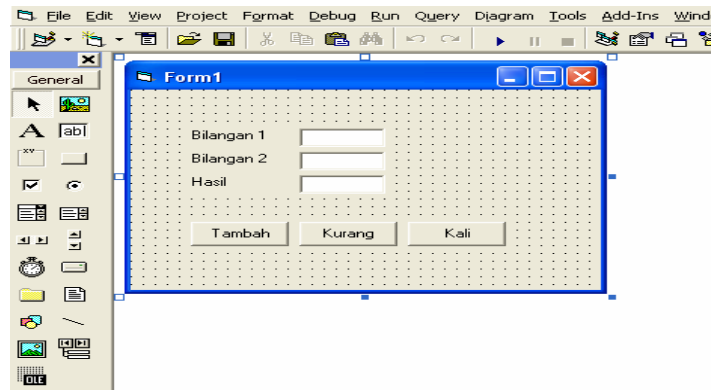
(2) Tambahkan komponen-komponen: 3 komponen label, 3 komponen textbox dan 3 komponen commandButton yang ada di [toolbox]. Atur tempatnya pada form seperti gambar



(3) Pada komponen label1, isikan property [Caption] dengan Bilangan 1. Pada label2, isikan property [Caption] dengan Bilangan 2. Dan pada label 3 isikan property [Caption] dengan Hasil.

(4) Pada komponen textbox1, isikan property [Name] dengan bil1 dan property [Text] dikosongkan. Pada textbox2, isikan property [Name] dengan bil2 dan property [Text] dikosongkan. Dan pada textbox3, isikan property [Name] dengan bil3 dan property [Text] dikosongkan.

- (5) Pada komponen command1, isikan property [Caption] dengan Tambah. Pada command2, isikan property [Caption] dengan Kurang. Dan pada command3, isikan property [Caption] dengan Kali. Sehingga hasil tampilan form menjadi



- (6) Click pada command1 (Tambah), sehingga muncul tampilan script dengan disediakan perintah dari event command_click

```
Private Sub Command1_Click()
```

```
End Sub
```

Isikan program penjumlahan dari bil1 dan bil2, hasilnya ditampung dalam bil3 sebagai berikut

```
Private Sub Command1_Click()
```

```
a=val(bil1)
```

```
b=val(bil2)
```

```
c=a+b
```

```
bil3 = c
```

```
End Sub
```

Val merupakan suatu fungsi yang digunakan untuk mengkonversi tipe data string atau variant seperti pada textbox menjadi tipe data numerik sehingga bisa dihitung dengan operasi aritmatika.

- (7) Click pada command1 (kurang), sehingga muncul tampilan script dengan disediakan perintah dari event command_click

```
Private Sub Command2_Click()
```

```
End Sub
```

Isikan program pengurangan dari bil1 dan bil2, hasilnya ditampung dalam bil3 sebagai berikut

```
Private Sub Command2_Click()
```

```
a=val(bil1)
```

```
b=val(bil2)
```

```
c=a-b
```

```
bil3 = c
```

```
End Sub
```

- (8) Click pada command3 (kali), sehingga muncul tampilan script dengan disediakan perintah dari event command_click

```
Private Sub Command3_Click()
```

```
End Sub
```

Isikan program perkalian dari bil1 dan bil2, hasilnya ditampung dalam bil3 sebagai berikut

```
Private Sub Command1_Click()
```

```
a=val(bil1)
```

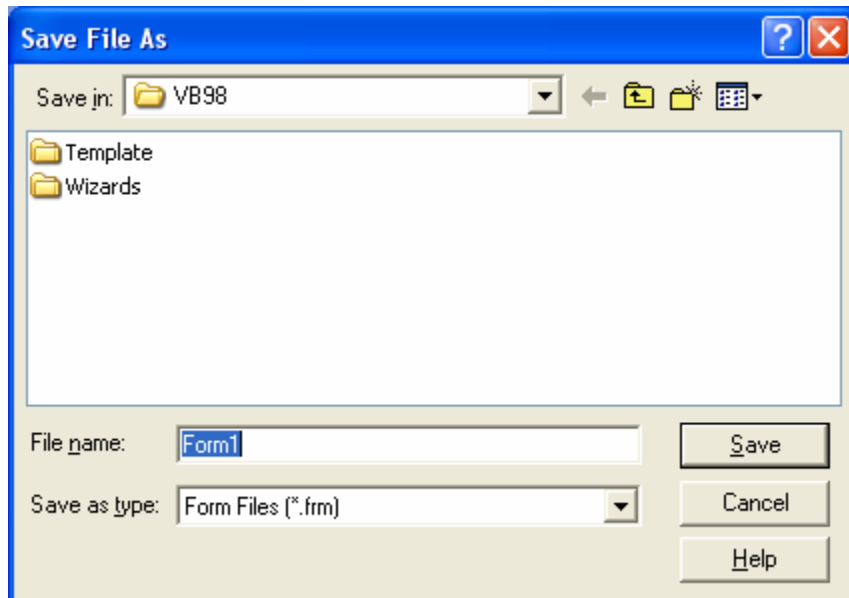
```
b=val(bil2)
```

```
c=a*b
```

```
bil3 = c
```

```
End Sub
```

- (9) Project sudah selesai, untuk menjalankan tekan F5. Cobalah untuk mengisi Bilangan 1 dan Bilangan 2 dengan sembarang nilai numerik, lalu click tombol-tombol tambah, kurang dan kali. Bila hasilnya benar maka project aplikasi dikatakan benar. Untuk menyimpan project pilih menu [File] >> [Save Project]



Pilih Folder mana untuk menyimpan file, sebaiknya buat folder baru di My Documents. Isikan nama file dengan FormLatihan31 untuk form lalu tekan [Save]. Setelah itu muncul jendela yang sama untuk project, isikan nama project dengan ProjectLatihan31 dan tekan [Save].