

A. PENGENALAN GUI VISUAL BASIC

A. Pengenalan

Visual Basic adalah bahasa pemrograman windows yang berbasis grafis (GUI-Graphical User Interface). Sifat bahasa pemrogramannya adalah *event-driven*, artinya program akan terjadi jika ada respon dari pemakai berupa event/kejadian tertentu (tombol diklik, mouse ditekan dan lain-lain). Saat event terjadi maka kode yang berhubungan dengan event akan dijalankan.

Dalam Visual Basic, pembuatan aplikasi dimulai dengan memperkirakan kebutuhan, merancang tampilan dan selanjutnya diikuti dengan pembuatan kode untuk program tersebut.

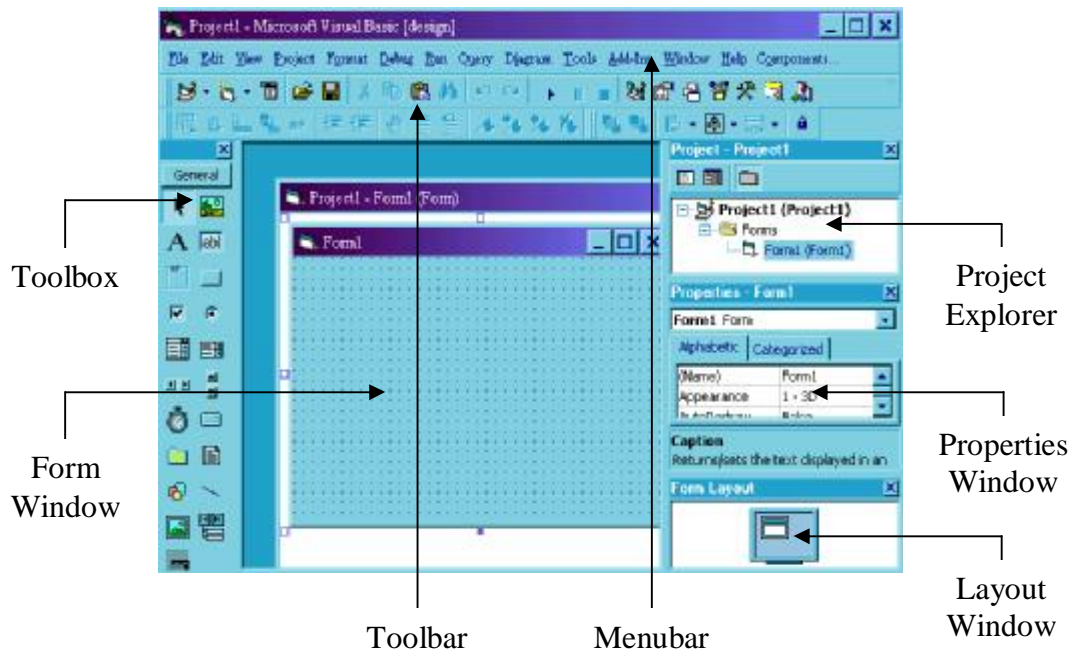
Pada bagian ini akan dijelaskan tentang pengenalan program Visual Basic dalam pembuatan aplikasi Windows.

Menjalankan program Visual Basic sama dengan menjalankan program windows lainnya, yaitu dengan mengklik ganda icon yang digunakan untuk menjalankan program. Berikut adalah tampilan saat pertama kali Visual Basic dijalankan.



Gambar 1. Kotak dialog New Project

Jika kita pilih **Standard Exe** akan ditampilkan sebuah ruang kerja dengan sebuah form seperti terlihat pada gambar berikut :






Gambar 2. Bidang Kerja Standar Exe Visual Basic

Keterangan :


- Toolbox** : Kumpulan dari objek user interface dan kontrol untuk program aplikasi
- Menubar** : Menu drop-down yang merupakan kumpulan perintah yang dikelompokkan berdasarkan kriteria operasi
- Toolbar** : Kumpulan gambar yang mewakili perintah tertentu pada Visual Basic
- Form Window** : Tempat merancang program aplikasi Visual Basic
- Project Explorer** : Tempat daftar form dan modul proyek aplikasi
- Properties Window** : Tempat setting propertis dari objek yang dipilih, misalnya warna, caption dan lain-lain dari objek yang bersangkutan
- Layout Window** : Tempat mengatur posisi relatif form terhadap layar monitor

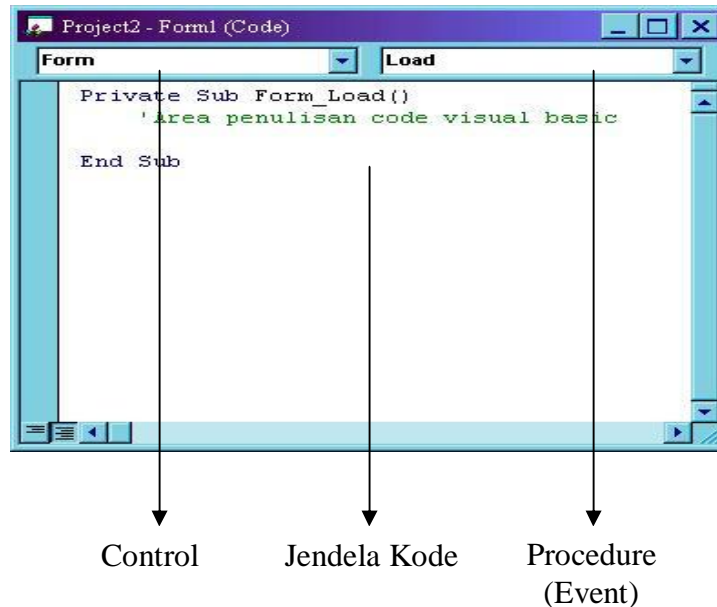
Berikut ini adalah penjelasan kontrol yang ada di toolbox

| Bentuk | Nama | Penjelasan |
|---|------------|---|
|  | Pointer | Bukan kontrol, tetapi penunjuk kontrol yang berfungsi memindahkan dan mengubah ukuran kontrol pada form |
|  | PictureBox | Kontrol untuk menampilkan file gambar (*.bmp, *.jpg, *.gif, *.wmf, dan *.ico) |
|  | Label | Untuk menampilkan teks pada form, tetapi pemakai tidak bisa berinteraksi dengannya (saat aplikasi dijalankan) |

| | | |
|---|---------------|---|
|  | TextBox | Untuk menampilkan teks dan pemakai dapat berinteraksi denganya |
|  | Frame | Untuk mengelompokkan beberapa kontrol |
|  | CommandButton | Untuk membuat tombol pelaksana suatu perintah |
|  | CheckBox | Untuk memberikan perintah pilihan benar/salah atau ya/tidak |
|  | OptionButton | Untuk memilih dari banyak pilihan yang ada |
|  | ComboBox | Sebagai tempat mengetikkan pilihan atau memilih suatu pilihan lewat Drop-Down-List |
|  | ListBox | Untuk menampilkan daftar pilihan yang dapat digulung secara horizontal maupun vertikal |
|  | HScrollBar | Untuk menggulung areal kerja secara horizontal |
|  | VScrollBar | Untuk menggulung areal kerja secara vertikal |
|  | Timer | Untuk mengoperasikan waktu kejadian pada rutin program dalam selang waktu yang ditentukan |
|  | DriveListBox | Untuk menampilkan daftar drive |
|  | DirListBox | Untuk menampilkan daftar Direktori (folder) pada drive yang aktif |
|  | FileListBox | Untuk menampilkan daftar file pada Direktori yang aktif |
|  | Shape | Untuk membuat lingkaran, segiempat dan elips |
|  | Line | Untuk membuat garis |
|  | Image | Kontrol untuk menampilkan file gambar (*.bmp, *.jpg, *.gif, *.wmf, dan *.ico) dengan perbedaan dari PictureBox adalah lebih sedikit memakai memory komputer |
|  | Data | Sebagai sarana akses data dalam suatu database |
|  | OLE | Untuk menghasilkan proses <i>Link</i> dan <i>Embed</i> objek antar aplikasi |

Jendela kode (*Code window*)

Jendela kode (*Code window*) adalah tempat menuliskan kode-kode program visual basic. Secara default code window tidak ditampilkan di ruang kerja, untuk mengaktifkannya klik icon  view code pada project explorer atau dengan cara double klik pada objek tertentu di form window.



Gambar 3. Jendela Kode

Event adalah peristiwa atau kejadian yang diterima oleh suatu objek, misalnya mouse diklik kanan, doubleklik, penekanan tombol (CommandButton) dan sebagainya.

Event itu sendiri ada yang bersifat spesifik terhadap suatu kontrol (objek), dan ada yang bersifat umum. Contohnya event umum adalah kejadian mouse diklik atas objek, sedangkan event resize hanya dimiliki oleh objek form, sedangkan objek yang lain tidak.

Untuk mengetahui kejadian apa saja yang dimiliki oleh sebuah objek, dapat dilihat dengan fasilitas help, dengan terlebih dahulu memilih objek dan menekan tombol F1.

Modul

Modul dapat disejajarkan dengan form, tetapi tidak terdapat objek. Modul hanya berisi kode-kode program berupa prosedur dan deklarasi variabel yang sifatnya global, artinya kode-kode yang terdapat pada modul dapat digunakan oleh seluruh bagian program, sedangkan kode-kode yang diletakkan pada form hanya dapat digunakan oleh form yang mengandung kode tersebut. Karena hanya menggunakan satu buah form didalam buku ini tidak digunakan modul. Untuk membuat modul adalah pada menubar **Project** pilih **Add Modul**.

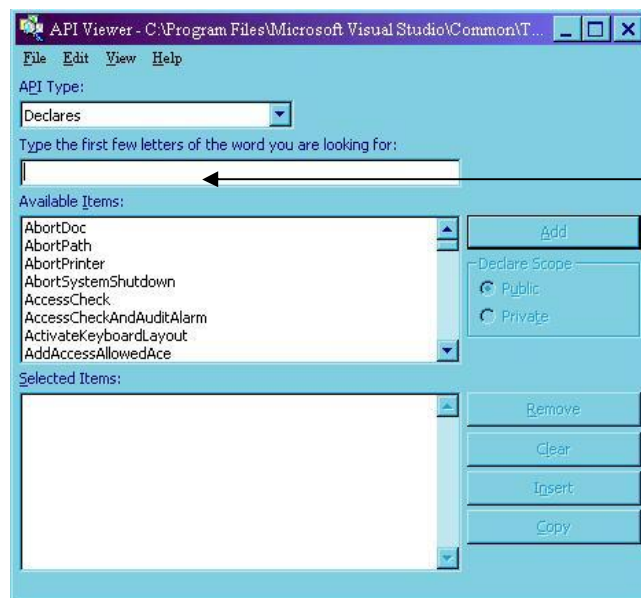


Gambar 4. Jendela Modul

B. Pengenalan Windows API (*Application Programming Interface*)

Window API merupakan kumpulan fungsi-fungsi eksternal yang disediakan library windows untuk mengatur kemampuan dan tingkah laku setiap element di Windows (dari tampilan di desktop hingga alokasi memory) sehingga dapat dimanfaatkan suatu program untuk meningkatkan kemampuan program tersebut, contohnya saja untuk pengaksesan registry windows dengan fasilitas built in visual basic sangat sukar dilakukan, tetapi dengan adanya fasilitas API untuk registry dari library advapi32.dll, pengaksesan tersebut menjadi lebih mudah.

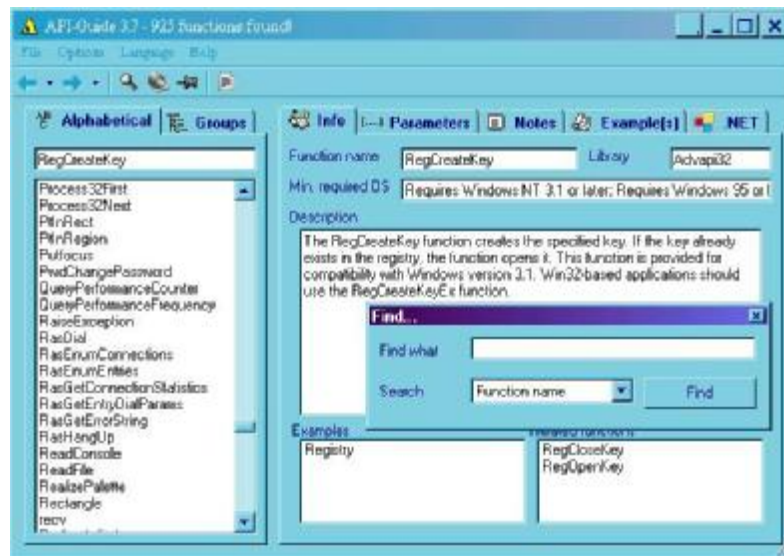
Untuk memasukkan Window API ke dalam proyek Visual Basic, cara yang paling mudah adalah dengan menggunakan program API Viewer yang terdapat di menubar Add-Ins yaitu Add-In Manager. Pilih Loaded API Viewer, lalu buka text file darinya, pilih Win32api.txt



Tempat
pencarian
fungsi API

Gambar 5. Tampilan program API Viewer

Atau dengan menggunakan program API-GUIDE dari www.allapi.net yang juga dapat ditemukan dalam CD buku ini.



Gambar 6. Tampilan program API-Guide

Program tersebut menyediakan contoh-contoh source code visual basic yang menggunakan API. Sehingga kita dapat mempelajarinya.

B. PENGENALAN PERINTAH

A. Variabel

Dalam bekerja dengan kode program, termasuk dengan program Visual Basic, kita tidak akan pernah lepas dari yang namanya Variabel. Variabel adalah nama penyimpanan data di memori komputer yang digunakan selama eksekusi program dan nilainya datanya bisa berubah-ubah.

Nama variabel sebaiknya disesuaikan dengan nilai yang disimpan ke dalamnya, tetapi ada aturan dalam pembuatan nama variabel, yaitu :

- Dimulai dengan huruf, setelahnya baru diperbolehkan diikuti angka
- Jumlah karakter tidak boleh lebih dari 255 karakter
- Tidak boleh disisipi dengan karakter-karakter tertentu, seperti karakter . , ? | \ / " () { } ! @ # \$ % ^ & *
- Unik, tidak boleh sama dengan yang lain.

Variabel terdiri atas beberapa jenis. Setiap jenis variabel memiliki nilai rentang yang diperbolehkan, selain itu pemilihan jenis variabel dan jumlah variabel akan berpengaruh terhadap jumlah kebutuhan memory komputer (RAM), berikut jenis variabel yang digunakan dalam Visual Basic :

| Jenis Variabel | Rentang nilai | Banyak RAM |
|----------------|--|------------------------------|
| Boolean | 2 nilai, yaitu True dan False | 2 byte |
| Byte | 0 - 255 | 1 byte |
| Currency | -922.337.203.685.477,5808 - 922.337.203.685.477,5807 | 8 byte |
| Date | Tanggal dan waktu dari 1 januari 100 hingga 31 Desember 9999 | 8 byte |
| Double | Untuk bilangan negatif : -1,79769313486232E308 hingga -4,94065645841247E-324 dan Untuk bilangan positif : 4,94065645841247E-324 hingga 1,79769313486232E308 | 8 byte |
| Integer | -32.768 hingga 32.767 | 2 byte |
| Long | -2.147.483.648 hingga 2.147.483.647 | 4 byte |
| Object | Digunakan untuk mewakili suatu object. Menggunakan statement SET, contohnya : Set Ctrl=Command1 Ctrl.Caption="Tombolku" | 4 byte |
| Single | Untuk bilangan negatif : -3,402823E38 hingga -1,401298E-45 Untuk bilangan positif : 1,401298E-45 hingga 3,402823E38 | 4 byte |
| String | Berisi karakter ASCII | 2 byte + 1 byte per karakter |
| Variant | Bisa mewakili semua jenis data di atas | 16 byte |

Untuk mendeklarasikan suatu variabel yang bukan jenis variant harus menggunakan perintah Private, Dim, Public, atau Static.

Private Digunakan membuat variabel lokal, hanya dapat diakses dalam prosedur atau modul dimana ia dideklarasikan

Contoh Private hari as Date

Dim Digunakan untuk mendaftarkan variabel tipe tertentu

Contoh : Dim i as integer

Public Digunakan agar variabel tersebut dapat diakses dari prosedur lain

Contoh : Public i as long

Static Digunakan agar isi datanya tidak tetap ada di memory setiap kali digunakan. Contoh : Static i as Integer

B. Konstanta

Konstanta adalah variabel yang nilainya tetap, cara penulisannya adalah sebagai berikut :

[Public/Private] Const *nama_konstanta* [As Type]= Ekspresi

Keterangan :

[..] Diperbolehkan tidak ada

Public Dapat diakses dari semua form/modul.

Private Hanya dapat diakses dari form/modul dimana dideklarasikan

Type Jenis data, misalnya integer, string dan sebagainya

Ekspresi Nilai yang cara penulisannya tergantung pada type data, misalnya :

- Konstanta Numerik (Byte, Integer, Single, Double dan lainnya) ditulis apa adanya. Contoh : 250, -75, 567E-2
- Konstanta String ditulis dengan diapit tanda petik (“ ”). Contoh : “007”, “Tulisanku”, “I love you”
- Konstanta Date ditulis dengan diapit tanda pagar (# #). Contoh : #30/06/1976#, #4:15 PM#
- Konstanta Boolean, hanya berisi dua nilai, yaitu True atau False.

C. Fungsi

Fungsi adalah suatu prosedur yang dapat menerima berbagai nilai untuk kemudian diolah dan memberikan hasil sebagai umpan baliknya. Nilai yang dimasukkan ke dalam suatu fungsi disebut Parameter, yang dapat berupa berbagai jenis angka, string, dan sebagainya. Terdiri atas :

a. Fungsi string

Berguna untuk memanipulasi data string, contohnya

- & Berguna untuk menggabungkan nilai string, contohnya : “Rumah” & “Sakit” menghasilkan “Rumah Sakit”
- Asc() Berguna untuk mengubah nilai string menjadi nilai ASCII, Contohnya : Asc(“A”) menghasilkan 65
- Chr() Berguna untuk mengubah nilai ASCII menjadi nilai String, Contohnya : Chr(65) menghasilkan A
- Left() Berguna mengambil n karakter di sebelah kiri string

Contoh : Left("Komputer", 2) menghasilkan "Ko"

- Right() Berguna mengambil n karakter di sebelah kanan string
Contoh : Right("Komputer", 2) menghasilkan "er"
- Mid() Berguna mengambil n karakter string dari suatu posisi yang Ditetapkan. Contoh : Mid("Komputer", 2, 3) menghasilkan "ompu"
- Len() Berguna mendapatkan jumlah karakter yang membentuk string
Contoh : Len("Komputer") menghasilkan 8
- Str() Berguna mengubah nilai numerik menjadi nilai string
Contoh : Str(2+3) menghasilkan "5"
- Val() Berguna mengubah nilai string menjadi nilai numerik
Contoh : Val("5") menghasilkan 5
- Ucase Berguna mengubah nilai string menjadi huruf besar
Contoh : Ucase("Rumah sakit") menjadi "RUMAH SAKIT"
- Lcase Berguna mengubah nilai string menjadi huruf kecil
Contoh : Lcase("Rumah Sakit") menjadi "rumah sakit"

b. Fungsi Matematik

Berguna untuk melakukan perhitungan ataupun manipulasi tertentu terhadap data numerik, contohnya :

- Abs() Berguna menghasilkan nilai absolut, contoh : Abs(-30) = 30
- Atn() Menghasilkan nilai arc tangen dalam radian,
contoh : Atn(3.14) = 1.2626
- Cos() Menghasilkan nilai cosinus dalam radian, contoh : Cos(3.14) = -1
- Sin() Menghasilkan nilai sinus dalam radian
- Tan() Menghasilkan nilai tangen dalam radian
- Mod() Menghasilkan nilai sisa pembagian, contoh : 8 mod 3 = 2
- Sqr() Menghasilkan nilai akar bilangan, contoh : Sqr(16) = 4
- Round() Membulatkan bilangan cacah dalam jumlah desimal tertentu,
Contoh : Round(33.3457, 2) = 33.34
- Hex() Menghasilkan nilai hexadesimal dari suatu bilangan
Contoh : Hex(459) = 1CB
- Oct() Menghasilkan nilai oktal dari suatu bilangan
Contoh : Oct(8) = 10

c. Fungsi Tanggal dan Jam

Beberapa fungsi yang diperuntukkan khusus bagi variabel type Date adalah :

- Now Menghasilkan tanggal dan jam sistem komputer.
- Date Menghasilkan tanggal sistem komputer, contoh : Date menghasilkan tanggal sekarang, misalnya 17/01/2005
- Day() Menghasilkan angka tanggal dari tanggal,
contoh : Day(Date) = 13
- Month() Menghasilkan angka bulan dari tanggal,
contoh : Month(Date) = 1
- Year() Menghasilkan angka tahun dari tanggal,
contoh : Year(#17/01/2005#) = 2005
- Time Menghasilkan jam sistem komputer, contohnya : Time

menghasilkan 11:30:56

- Hour() Menghasilkan angka jam, contoh : Hour(#11:30:56#) = 8
- Minute() Menghasilkan angka menit, contoh : Minute(Time) = 30
- Second() Menghasilkan angka detik, contoh : Second(Time) = 56

D. Operator

Operator adalah tanda yang digunakan untuk menghubungkan suatu variabel atau konstanta dengan variabel atau konstanta lain untuk diolah atau dimanipulasi.

Ada beberapa type operator, yaitu :

a. Operator Aritmatik

| Operator | Operasi | Contoh |
|----------|----------------------------|------------------|
| ^ | Pemangkatan | $4^2 = 8$ |
| - | Minus | -8 |
| *, / | Perkalian dan pembagian | $4*8/2$ |
| \ | Pembagian integer | $10 \setminus 3$ |
| Mod | Sisa hasil bagi | 10 Mod 4 |
| +, - | Penambahan dan pengurangan | $5 + 2 - 1$ |

b. Operator Perbandingan

| Operator | Operasi | Contoh |
|----------|------------------------------|-------------|
| = | Sama dengan | Nilai = 12 |
| <> | Tidak sama dengan | Nilai <> 12 |
| < | Lebih kecil | Nilai < 15 |
| > | Lebih besar | Nilai > 15 |
| <= | Lebih kecil atau sama dengan | Nilai <= 15 |
| >= | Lebih besar atau sama dengan | Nilai >= 15 |

c. Operator Logika

| Operator | Operasi |
|----------|-------------|
| Not | Tidak |
| And | Dan |
| Or | Atau |
| Xor | Exlusive Or |
| Eqv | Ekivalen |
| Imp | Implikasi |

E. Penyeleksian kondisi

Ada beberapa penyeleksian kondisi yang sering digunakan dalam pengontrolan jalannya suatu program, yaitu :

a. If (kondisi) then (ekspresi)

Berfungsi jika sesuai dengan kondisi, maka ekspresi akan dijalankan, contoh :
 If data = 20 then msgbox("OKE")
 Artinya jika nilai variabel data = 2 maka dimunculkan kotak bertulisan OKE.

b. If (kondisi) Then (Ekspresi1) Else (Ekspresi2)

Berfungsi jika sesuai dengan kondisi maka ekspresi1 dijalankan, jika tidak sesuai maka ekspresi2 di jalankan, contoh :

```
If data = 20 then
  MsgBox("OKE")
Else
  MsgBox("Coba lagi")
End if
```

Artinya jika data isinya 20 maka kotak pesan bertulisan "OKE" jika tidak bertulisan "Coba lagi".

c. Select Case

Digunakan untuk emnjalankan satu blok perintah yang jumlahnya banyak. Bentuk penulisannya adalah sebagai berikut :

```
Select Case (variabel)
  Case (nilai1)
    (blok perintah1)
  Case (nilai2)
    (blok perintah2)
  Case (nilai3)
    (blok perintah3)
  (dst)
End select
```

Contoh :

```
Dim dat as integer
Select case dat
  Case 12
    MsgBox("Dua belas")
  Case 15
    MsgBox("Lima belas")
  Case 20
    MsgBox("Dua puluh")
End Select
```

Artinya jika variabel integer dat nilainya 12 kotak pesan dimunculkan dengan tulisan “Dua belas”, jika nilainya 15 kotak pesan bertulisan Lima belas dan seterusnya.

F. Struktur Pengulangan (Looping)

Dalam pengontrolan jalannya suatu program looping digunakan untuk mengulang suatu blok perintah sampai kondisi tertentu. Proses ini akan terus bekerja selama kondisi yang dibandingkan bernilai benar (True), jika salah maka looping dihentikan dan dilanjutkan oleh perintah selanjutnya.

Struktur pengulangan yang sering digunakan antara lain adalah sebagai berikut :

a. For – Next

Digunakan untuk mengulang blok perintah dalam jumlah yang telah ditentukan, di dalamnya hanya perlu menuliskan nilai awal dan nilai akhir variabel penghitung.

Bentuk penulisannya adalah sebagai berikut :

```
For (Variabel integer) = (awal) to (akhir) [step (pertambahan)]
    (blok perintah)
Next
```

Keterangan :

[step (pertambahan)] tidak selalu harus ada

Contoh :

```
Dim ulang as integer
For ulang = 0 to 30
    Print ulang
Next
```

Akan menghasilkan tulisan dari “0” hingga “30” di form.

b. Do While – Loop

Digunakan untuk melakukan pengulangan terus menerus selama kondisi memenuhi syarat. Jika bernilai False akan berhenti.

Cara penulisannya adalah sebagai berikut :

```
Do While (kondisi)
    (Ekspresi)
Loop
```

Contoh :

```
Dat = 1
Do while Dat < 20
    Print Dat
    Dat = Dat + 1
Loop
```

Jika nilai Dat sudah mencapai 19 maka perulangan akan berhenti.

c. For Each – Next

Digunakan untuk mengulang satu blok perintah bagi tiap elemen pada suatu koleksi objek. Contoh koleksi berupa kumpulan kontrol pada sebuah form, dimana masing-masing kontrol tersebut menjadi komponennya. Pemakaian struktur ini sangat cocok jika kita tidak mengetahui berapa jumlah komponen pada form atau lembar kerja.

Bentuk penulisannya adalah sebagai berikut :

```
For (komponen) in (group)
    (ekspresi)
Next
```

Contoh penulisannya :

```
Private Sub Command1_Click
    Dim ctrl as Object
    For Each ctrl In Form1.Controls
        ctrl.Enabled = False
    Next
End Sub
```

Artinya saat event Command1 diklik maka semua kontrol yang berjenis objek di dalam form1, propertis enablednya diset false.