

Pengenalan Visual Basic 6.0

1. Visual Basic

Visual Basic merupakan salah satu RAD (Rapid Application Development) Tool yang sangat terkenal mudah untuk dipelajari, digunakan dan cepat dalam membangun suatu aplikasi visual berbasis Windows (98, ME maupun Win 2000). Bahkan, dengan adanya motto yang diluncurkan oleh competitor Microsoft, SUN, yaitu " Write One Run Everywhere" untuk produk andalannya SUN, yaitu Java, maka sekarang telah banyak sekali vendor-vendor partner Microsoft yang mempermudah anda untuk mengkonversi aplikasi VB anda yang jalan di Windows ke platform lainnya, seperti Linux, AIX, dan masih banyak lagi. Tetapi untuk bagian ini, saya tidak akan mengajak anda untuk melihat ke arah tersebut, melainkan di sini saya akan memberikan anda materi-materi yang dibuat sesingkat dan semudah mungkin agar anda dapat belajar dan menguasai bahasa pemrograman visual ini dengan cepat.

Visual Basic menawarkan kepada anda untuk membuat User Interface (antar muka pengguna) anda dalam waktu singkat sekali. Bagi anda yang pernah bermain dengan C, Turbo Pascal, Turbo Basic, anda tentu merasakan berapa banyak waktu yang anda habiskan untuk mendesign User Interface anda sebelum anda konsentrasi ke flow program anda (business logic).

Dengan melalui Visual Basic, konsep pengembangan aplikasi akan dimulai dengan pembentukan *user interface*, kemudian *mengatur properti dari objek-objek* yang digunakan dalam user interface (warna tombol, warna layar, judul layar, dll) , dan baru dilakukan *penulisan kode program* untuk menangani kejadian-kejadian (event).

2. Pengenalan IDE Visual Basic

Langkah awal dari belajar Visual Basic adalah mengenal IDE (Integrated Development Environment) Visual Basic yang merupakan Lingkungan Pengembangan Terpadu (Integrated Development Environment) bagi programmer dalam mengembangkan aplikasinya. Dengan menggunakan IDE programmer dapat membuat user interface, melakukan koding, melakukan testing dan debugging serta

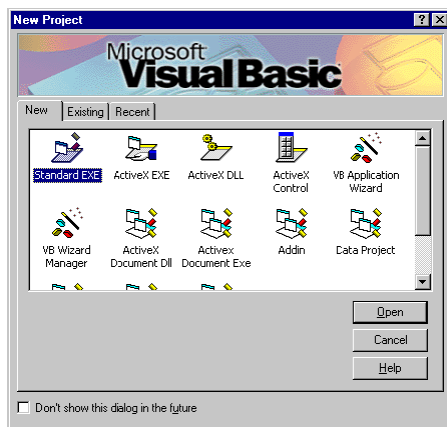
menkompilasi program menjadi executable. Penguasaan yang baik akan IDE akan sangat membantu programmer dalam mengefektifkan tugas-tugasnya sehingga dapat bekerja dengan efisien.

3. Menjalankan IDE

Salah satu cara untuk mengaktifkan IDE Visual Basic adalah menjalankannya dari Menu **Start**, pilih **Microsoft Visual Basic 6.0** dan akhirnya pilih shortcut **Microsoft Visual Basic 6.0**.

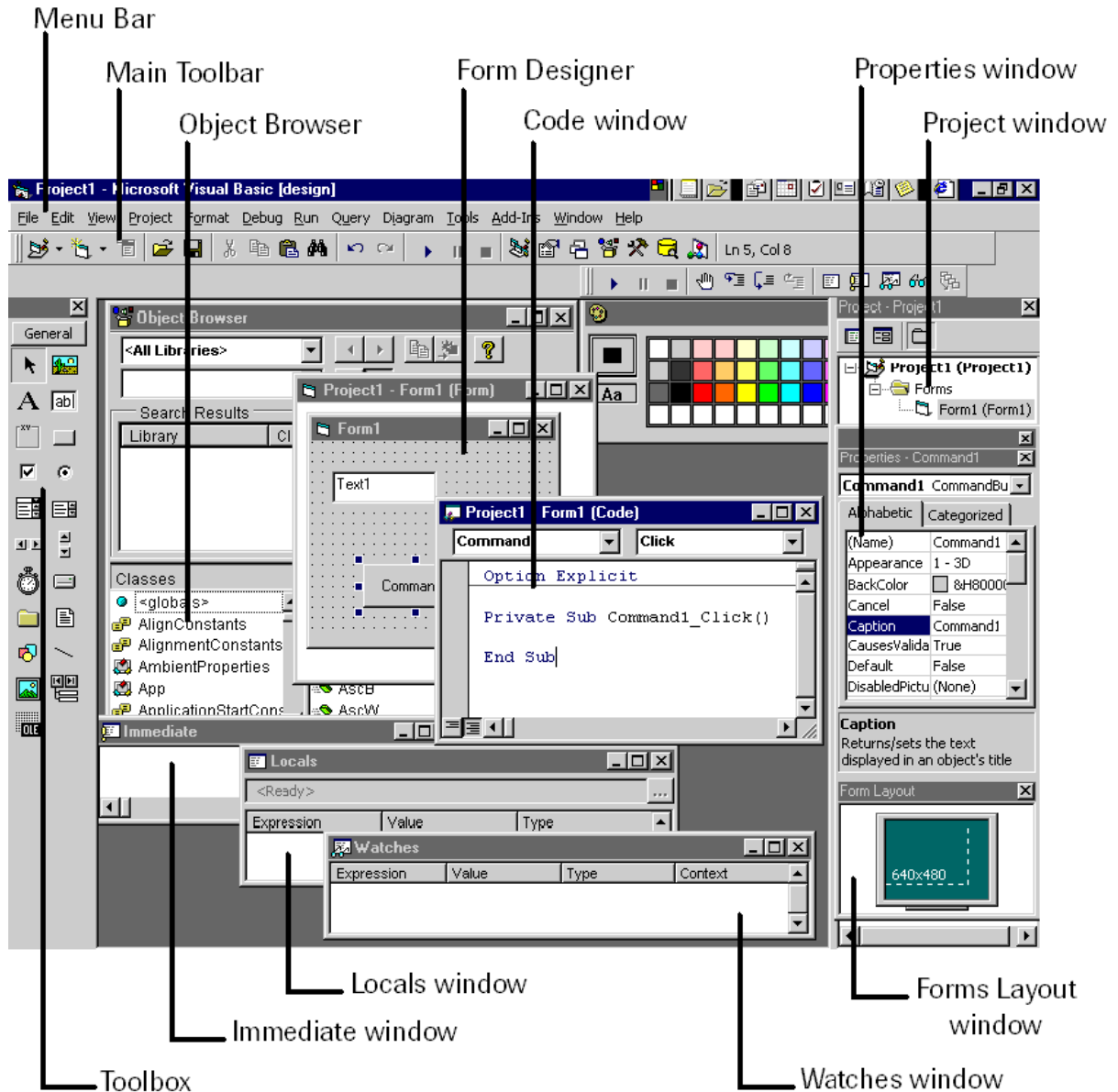
4. Memilih jenis Project

Setelah anda menjalankan VB, maka IDE VB pertama kali akan menampilkan kotak dialog "New Project" kepada anda, seperti yang bisa anda lihat dibawah ini : Untuk tutorial pertama ini, mari kita memilih **Standard EXE**.



5. Jendela IDE

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



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 peserta belajar menjadi bingung. Adapun jendela-jendela yang perlu anda perhatikan adalah sebagai berikut :

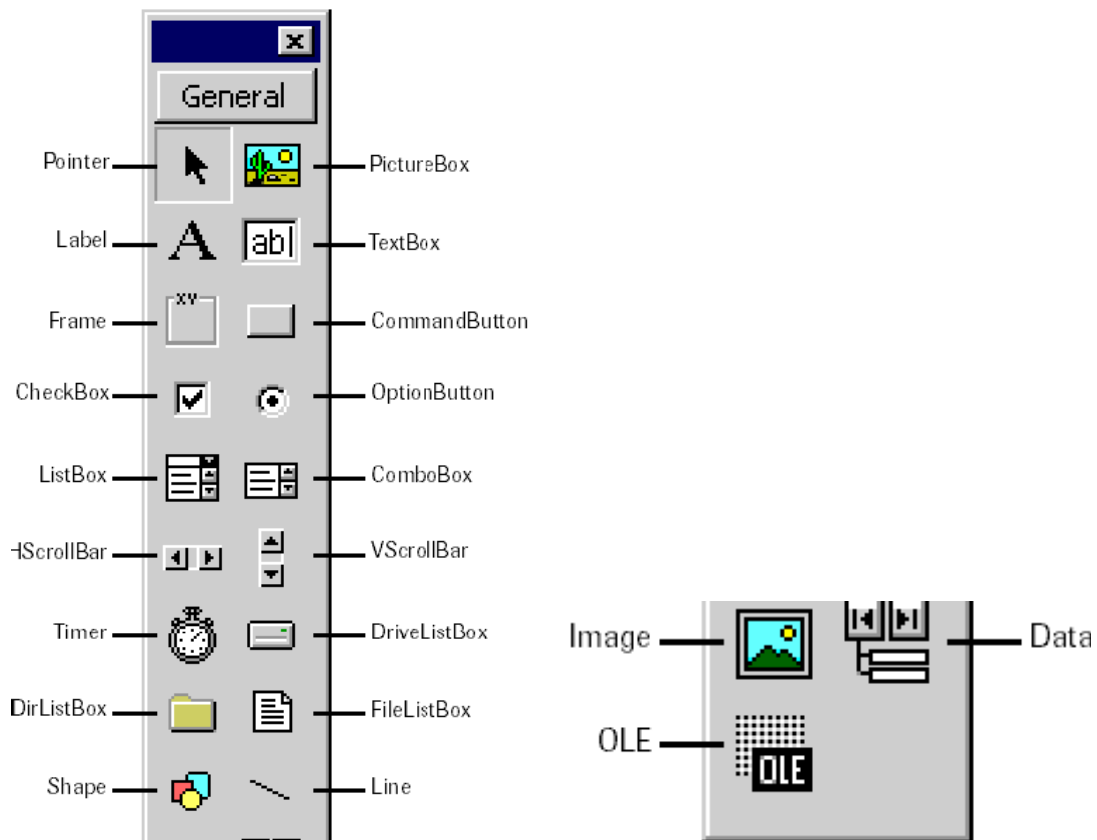
1. *Menu Bar*, digunakan untuk memilih tugas-tugas tertentu seperti menyimpan project, membuka project, dll.
2. *Main Toolbar*, digunakan untuk melakukan tugas-tugas tertentu dengan cepat.

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.

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**.

6. 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. Kita akan belajar semua control-control ini dalam module-module berikutnya. Pada module pengenalan ini, kita akan membahas secara singkat kegunaan dari control-control (object) yang ada pada IDE VB 6.0 ini.



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 (metafile), EMF (enhanced metafile), GIF, dan JPEG.
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.

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

7. Bahasa Visual Basic

Dasar dari pemrograman pada Visual Basic adalah bahasa BASICA yang sempat populer di sekitar tahun 1990 an, dimana bahasa ini pada awalnya langsung built-in di dalam komputer PC IBM. Jika ada diantara pembaca yang pernah bermain dengan Basica, Quick Basic, Turbo Basic maka belajar bahasa VB akan terasa mudah sekali. SEbaliknya bagi para pemula di dunia Visual Programming, jangan berkecil hati, karena bahasa BASICA / VB merupakan bahasa yang sangat gampang untuk di pelajari.

8. Type Variabel

Dibandingkan dengan type data yang terdapat pada bahasa basic, maka pada VB, type data yang disediakan lebih banyak, seperti type Currency, Decimal, Object, dan Variant. Variant merupakan type variabel yang istimewa, karena dapat berubah dari satu type ke type yang lain, sesuai dengan evaluasi ekspresi oleh Visual Basic.

Ketepatan pemilihan type variabel akan sangat menentukan pemakaian resources oleh aplikasi yang dihasilkan (performance, size daripada hasil kompilasi aplikasi, dll) dan ini merupakan tugas anda sebagai programmer untuk memilih type yang sesuai untuk menghasilkan program yang efisien dan berperformance tinggi.

Dibawah ini, saya sediakan tabel yang merupakan daftar type-type variable yang ada di VB :

Type Data	Ukuran Storage	Jangkuan
Byte	1	byte 0 s/d 255
Boolean	2	byte True atau False
Integer	2 byte	-32,768 s/d 32767
Long	4 byte	-2,147,483,648 s/d 2,147,483,647
Single	4 byte	-3.402823E38 s/d -1.401298E-45 (-) 1.401298E-45 s/d 3.402823E38 (+)
Double	8 byte	-1.79769313486232E308 s/d -4.94065645841247E-324
Currency	8 byte	-922,337,203,685,477.5808 s/d 922,337,203,685,477.5807
Decimal	14 byte	+/-79,228,162,514,264,337,593,543,950,335
Date	8 byte	1 Januari 100 s/d 31 Desember 9999
Object	4 byte	Mengacu pada objek tertentu
String	10 byte + panjang	string 0 sampai lebih kurang 2 milyar
Variant(angka)	16 byte	Sembarang angka sampai jangkauan jenis Double
Variant(karakter)	22 byte + panjang	String Sama dengan jangkauan variabel String

9. Cara Menuliskan Komentar di Visual Basic

Pada saat anda membuat program di VB, anda dianjurkan untuk sering memberikan komentar kepada program yang anda buat. Seperti jika di C, anda mengenal tanda "//" untuk memberikan komentar kepada source code C anda, maka di VB anda bisa menggunakan tanda " ' " untuk menambahkan komentar kepada program yang sedang anda buat.

Contoh : Print A 'untuk memprint nilai dari A ke layar

10. Operator Aritmatika dan Logika di Visual Basic

Visual basic menyediakan operator aritmatika, komparasi dan logika, salah satu hal yang harus dipahami oleh programmer adalah tata urutan operasi dari masing-masing operator tersebut sehingga mampu membuat ekspresi yang akan menghasilkan nilai yang benar. Tabel berikut mencoba memberikan anda gambaran symbol-symbol operator yang akan kita gunakan dalam pemrograman di Visual Basic .

Contoh :

Jumlah = 5 - 2 * 3 'Akan menghasilkan -1

Jumlah = (5 - 2) * 3 'Akan menghasilkan 9

Aritmatika	Komparasi	Logika
Pangkat (^)	Sama (=)	Not
Negatif (-)	Tidak sama (<>)	And
Kali dan Bagi	(*, /) Kurang dari (<)	Or
Pembagian bulat	(\)	Xor
Sisa Bagi (Mod)	(<=)	Eqv
Tambah dan Kurang	(+,-) Lebih dari atau sama (>=)	Imp
Pengabungan String (&)	Like	

11. Deklarasi Variabel

Visual Basic memungkinkan kita untuk menggunakan variabel tanpa deklarasi. Tetapi hal ini adalah kurang baik untuk program yang terstruktur dan menghindari kesalahan pengolahan yang diakibatkan oleh kesalahan dalam pengetikkan nama variabel. Agar setiap variabel yang digunakan harus dideklarasikan, dapat digunakan perintah :

Option Explicit

Cara pengdeklarasian variabel di VB adalah :

<keyword> NamaVariabel [**As** TypeVariabel]

Dimana keyword disini bisa berupa : DIM, GLOBAL, PUBLIC, PRIVATE dan STATIC Untuk tutorial-tutorial perdana ini kita akan cukup mengenal keyword DIM, yang sebetulnya merupakan singakatan dari DIMENSION.

Contoh pengdeklarasian varibel di VB :

```
DIM myArray(10) as INTEGER
```

```
DIM Jumlah as LONG
```

```
DIM Gaji as CURRENCY
```

12. Penggunaan Struktur Kontrol

Mengenal Struktur Kontrol

Struktur kontrol di dalam bahasa pemrograman adalah perintah dengan bentuk (struktur) tertentu yang digunakan untuk mengatur (mengontrol) jalannya program.

Visual Basic 6 mengenal dua jenis struktur kontrol, yaitu :

1. **Struktur kontrol keputusan** - digunakan untuk memutuskan kode program mana yang akan dikerjakan berdasarkan suatu kondisi.

1.1 Penggunaan If

```
If <pilihan> then
```

```
<pernyataan>
```

```
End if
```

Atau

```
If <pilihan> then
```

```
<pernyataan>
```

```
Elseif <pilihan> then  
Else  
<pernyataan>  
End If
```

1.2 Penggunaan CASE

```
SELECT CASE <pilihan>  
CASE <pilihan 1>  
<blok kode program 1>  
CASE <pilihan 2>  
<blok kode program 2>  
CASE <pilihan n>  
<blok kode program n>  
[CASE ELSE  
<blok kode program x>]  
END SELECT
```

2. Struktur kontrol pengulangan - digunakan untuk melakukan pengulangan kode program.

2.1 For...Next

```
FOR <pencacah> = <awal> TO <akhir> [STEP <langkah>]  
<blok kode program>  
NEXT <pencacah>
```

- <pencacah> adalah variabel (tipe: integer) yang digunakan untuk menyimpan angka pengulangan.
- <awal> adalah nilai awal dari <pencacah>.
- <akhir> adalah nilai akhir dari <pencacah>.
- <langkah> adalah perubahan nilai <pencacah> setiap pengulangan. Sifatnya optional (boleh ditulis ataupun tidak). Bila tidak ditulis maka nilai <langkah> adalah 1.

2.2 Do...Loop

```
DO WHILE <kondisi>
```

<blok kode program>

LOOP

<blok kode program> akan diulang **selama** <kondisi> bernilai TRUE.

Pengulangan berhenti bila <kondisi> sudah bernilai FALSE.

Atau

DO UNTIL <kondisi>

<blok kode program>

LOOP

<blok kode program> akan diulang **sampai** <kondisi> bernilai TRUE.

Pengulangan berhenti bila <kondisi> sudah bernilai TRUE.

13. Penggunaan Array

Array merupakan sekumpulan nilai data yang “dikelompokkan” dalam sebuah variabel. Array digunakan bila ada beberapa nilai data yang tipe datanya sama dan akan mendapat perlakuan yang sama pula. Misalnya, ada 10 nilai data dengan tipe string dan akan diolah dengan cara yang sama, maka akan lebih mudah jika menggunakan sebuah array dibandingkan bila menggunakan 10 variabel yang berbeda. Setiap nilai data di dalam sebuah array disebut **elemen array** dan masing-masing dibedakan dengan nomer indeksinya.

Dim nama_array(jumlah_elemen - 1) As tipe_data

Contoh :

Dim NamaSiswa(99) As String array NamaSiswa akan mempunyai elemen sebanyak 100 dengan nomer indeks mulai dari 0 s/d 99. Selanjutnya untuk mengisi nilai data ke dalam array :

nama_array(no_indeks) = nilai_data

Catatan :

- Agar nomer indeks array dimulai dari 1 (bukan 0), bisa menggunakan perintah **Option Base 1** sebelum perintah Dim-nya, contoh :

Option Base 1

Dim NamaSiswa(100) As String

- Atau bisa juga menggunakan kata **To** di dalam penentuan jumlah elemen array, contoh :

Dim NamaSiswa(1 To 100) As String

- Untuk mengetahui berapa jumlah elemen di dalam sebuah array, bisa menggunakan perintah **Ubound(nama_array)**, contoh :

Dim NamaSiswa(1 To 100) As String

Dim JmlElemen As Integer

JmlElemen = Ubound(NamaSiswa) hasilnya = 100

- Bila jumlah elemen array yang dibutuhkan tidak diketahui atau ingin bisa diubah-ubah, maka bagian jumlah_elemen pada perintah Dim tidak perlu diisi, contoh :

Dim NamaSiswa() As String

Teknik ini biasa disebut sebagai array dinamis, biasanya untuk mengisi data yang berkembang terus, dan belum bisa ditentukan di awal jumlah elemennya.

- Untuk mengubah jumlah elemen array digunakan perintah Redim :

Redim [Preserve] nama_array(jumlah_elemen)

Contoh :

Redim NamaSiswa(1 To 150) jumlah elemen array NamaSiswa menjadi 150 dengan nomer indeks dari 1 s/d 150.

14. Menangani Runtime Error

Untuk menangani runtime error, di dalam kode program dibuatkan bagian khusus untuk menangani error yang terjadi. Bagian khusus ini diberi tanda berupa <label>. Agar setiap error yang terjadi dapat ditangani oleh bagian khusus ini, pada awal kode program ditulis perintah berikut ini :

1. On Error GoTo <label>

Contoh:

On Error Go To sala

<Code Program>

MsgBox "Error:" & Err.Description, vbInformation

2. Resume

Ada 3 bentuk perintah Resume, yaitu :

- **Resume** – kembali ke kode program yang menimbulkan error. Biasanya digunakan untuk *retry* (mencoba lagi).
- **Resume <label>** – kembali ke bagian tertentu pada kode program yang diberi tanda <label>.
- **Resume Next** – kembali ke kode program berikutnya (kode program setelah kode program yang menimbulkan error). Biasanya digunakan untuk *abort* atau *cancel* (mengabaikan atau membatalkan).

Contoh:

On Error Resume Next

<Code Program>

End

15. Penggunaan Procedure

Procedure adalah blok kode program yang berisi perintah-perintah untuk mengerjakan tugas tertentu. Bila di dalam kode program yang kita buat ada perintah-perintah untuk melakukan tugas yang sama di beberapa tempat, maka akan lebih baik perintah-perintah tersebut dibuat dalam sebuah procedure. Kemudian, procedure itu bisa di-‘panggil’ bila diperlukan.

Pada VB6 ada 4 jenis procedure, yaitu :

- **Procedure Sub** – procedure yang tidak mengembalikan nilai setelah ‘tugas’-nya selesai.

[Public | Private] Sub <nama_sub> ([<argumen>])

...

<isi procedure>

...

End Sub

- **Procedure Function** – procedure yang mengembalikan nilai setelah ‘tugas’-nya selesai.

**[Public | Private] Function <nama_function> ([<argumen>]) As
<tipe_data>**

...

<isi procedure>

...

End Function

- **Procedure Event** – procedure untuk suatu event pada sebuah object. Digunakan di dalam class module.
- **Procedure Property** – procedure untuk mengubah (*let*) atau mengambil (*get*) nilai property pada sebuah object. Digunakan di dalam class module.

16. Event Pada Keyboard

Event keyboard merupakan salah satu elemen utama dari interaksi antara *user* dengan program yang kita buat. Event keyboard terjadi saat *user* menekan (*pressed*) ataupun melepas (*released*) tombol pada keyboard.

Menangani event keyboard dapat dilakukan pada 2 (dua) level, yaitu :

- Pada **level kontrol** (*low-level*) – menangani event keyboard yang terjadi pada sebuah kontrol, misalnya ketika *user* mengetik pada sebuah TextBox. Tidak semua kontrol mempunyai event keyboard. Hanya kontrol yang bisa mendapatkan focus (dicirikan dengan property TabIndex dan TabStop) saja yang mempunyai event keyboard.
- Pada **level form** (*hight-level*) – menangani event keyboard yang terjadi pada lingkup sebuah form. Artinya form akan lebih dulu merespon event keyboard daripada kontrol-kontrol yang ada di dalamnya. Agar form selalu lebih dulu merespon event keyboard maka property KeyPreview pada form tersebut harus diset menjadi True.

VB6 menyediakan 3 (tiga) jenis event pada form dan pada beberapa kontrol yang bisa menerima input dari *user* melalui keyboard, yaitu :

- Event **KeyPress** – terjadi ketika tombol-tombol yang mempunyai kode ASCII pada keyboard ditekan. ASCII (*American Standard Code for Information Interchange*) adalah kode dari sekumpulan karakter pada tombol keyboard yang terdiri dari : abjad, angka dan beberapa karakter khusus (Enter, Escape, Tab, Backspace).

Contoh:

```
Private Sub Form1_KeyPress (KeyAscii As Integer)
MsgBox "Kode ASCII tombol yang ditekan : " & KeyAscii
End Sub
```

- Event **KeyDown** – terjadi ketika setiap tombol pada keyboard ditekan.

Contoh:

```
Private Sub Form1_KeyDown(KeyCode As Integer, Shift As Integer)
MsgBox "Kode tombol yang ditekan : " & KeyCode
End Sub
```

- Event **KeyUp** – terjadi ketika setiap tombol pada keyboard dilepas.

Contoh:

```
Private Sub Form1_KeyUp(KeyCode As Integer, Shift As Integer)
MsgBox "Kode tombol yang dilepas : " & KeyCode
End if
```

17. Operasi File

Visual Basic menyediakan suatu fasilitas untuk membaca, menyimpan dan memperbaiki file data, sehingga memungkinkan untuk memisahkan data dengan program, dan program dapat menjadi lebih baik karena bisa tidak bergantung pada data (independent terhadap data), yang artinya datanya bisa diubah-ubah dengan mudah tanpa harus mengubah-ubah file programnya.

Operasi File di dalam Visual Basic terdapat bermacam-macam cara akses file ke dalam program aplikasinya, yaitu:

1. Membuka File

Untuk membuka file baik untuk keperluan penyimpanan maupun untuk keperluan pembacaan menggunakan perintah umum sebagai berikut::

Open "Nama_File" For {Status} As #{n}

Dimana: Nama_File : nama file lengkap dengan nama foldernya.

Status : adalah untuk keperluan apa file tersebut baca/simpan.

Status = INPUT untuk membaca File

Status = OUTPUT untuk menyimpan File

n : adalah nomor buffer (integer 1, 2, 3, ...) yang digunakan untuk keperluan pemilihan file yang mana yang akan diproses.

Setiap mengakses file baik untuk membaca maupun untuk menyimpan, maka file tersebut harus dibuka terlebih dahulu dengan status akses yang diberikan. Setelah file terbuka maka langkah berikutnya adalah bagaimana cara membaca dan menyimpan file.

File yang sudah dibaca, seharusnya ditutup saat program sudah selesai sehingga tidak ada data yang tertinggal di memory menyebabkan memory menjadi tidak efektif. Untuk menutup file dapat dilakukan dengan:

Close #n

2. Menyimpan Data Ke File

Untuk menyimpan data ke file di dalam program Visual Basic dapat dilakukan dengan membuka file untuk keperluan menyimpan atau menulis ke file:

Open Nama_File For Output As #1

Saat file dibuka dengan cara semacam ini, maka file hanya bisa ditulisi dan bila file tersebut sudah ada isinya sebelumnya, maka dianggap hilang. Atau dengan kata lain file selalu dianggap baru. Cara menyimpan file semacam ini menghasilkan file yang bersifat file text, sehingga bisa dibaca oleh *text editor* seperti notepad, wordpad atau microsoft word.

Setelah file dibuka, langkah berikutnya untuk menyimpan data dalam variabel ke file dapat dilakukan dengan:

Print #1, nama_variabel

3. Membaca Data Dari File

Untuk membaca data dari file di dalam program Visual Basic dapat dilakukan dengan terlebih dahulu membuka file untuk keperluan membaca dari file:

Open Nama_File For Input As #1

Saat file dibuka dengan cara semacam ini, maka file hanya bisa dibaca dan tidak bisa ditulisi. File yang bisa dibaca adalah file teks.

Untuk membaca data variabel ke dalam file, dapat dilakukan menggunakan perintah:

Input #1, nama_variabel