

# VBA - Visual Basic for Application

Bahasa pemrograman untuk aplikasi Windows

The background is a solid blue color. On the left side, there is a faint, light-colored grid pattern that resembles a globe or a map projection. A small, dark blue pushpin graphic is pinned to the grid, with its point resting on the text 'Bagian 2'.

# Bagian 2

## Variabel dan konstanta

# Apa yang dibicarakan?

- ◆ Mendeklarasi variabel
- ◆ Memahami lingkup variabel
- ◆ Mengirim variabel sbg referensi dan sbg argumen
- ◆ Variabel statik
- ◆ Menghindari kesalahan variabel
- ◆ Jenis data variabel
- ◆ Array
- ◆ Konstanta

# Deklarasi variabel

- ◆ Variabel dalam VBA sebaiknya dideklarasikan untuk menyatakan nama dan jenis data (data type)
- ◆ Pendeklrasian variabel menghindari kesalahan dan mengefisienkan program
- ◆ Caranya: dengan pernyataan Dim  
*Dim namavar As jenisdata*

*Namavar* harus dimulai dengan huruf, max 255 karakter, tidak mengandung spasi dan !  
@  
#  
\$  
%  
&

# contoh

```
Function GrossMargin()  
  ` deklarasi variabel  
  Dim totalSales  
  Dim totalExpenses  
  `program  
  totalSales = Application.Sum(Range("Sales"))  
  totalExpenses =  
  Application.Sum(Range("Expenses"))  
  GrossMargin=(totalSales -  
  totalExpenses)/totalSales  
End Function
```

# Lingkup/scope variabel

- ◆ Dalam VBA lingkup variabel
  - Lingkup prosedur
  - Lingkup modul
  - Lingkup publik

# Lingkup prosedur

- ◆ Variabel hanya bisa diakses oleh pernyataan-pernyataan dalam prosedur yang sama

```
Sub Procedure1()
```

```
    Dim pesan
```

```
    ` semua pernyataan di dalam prosedur ini
```

```
    ` dapat mengakses variabel `pesan`
```

```
    pesan = " Saya dalam lingkup"
```

```
    MsgBox pesan
```

```
End Sub
```

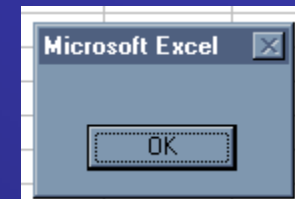
```
Sub Procedure2()
```

```
    ` pernyataan di dalam prosedur ini
```

```
    ` tak dapat mengakses variabel `pesan`
```

```
    MsgBox pesan ` variabel pesan tak dikenal dalam  
    prosedur ini
```

```
End Sub
```



# Deklarasi implisit dan eksplisit

- ◆ Pada contoh 2, variabel pesan sebetulnya tak dikenal, tetapi ketika dijalankan tetap memberi hasil, dalam hal ini <blank>. Ini dikarenakan VBA mengenal deklarasi implisit (semua variabel otomatis diberi harga <blank>, meski tak dikenal)
- ◆ Kondisi semacam ini kurang baik dalam pemrograman, sebaiknya dinyatakan bahwa semua variabel harus dideklarasikan agar tak terjadi kesalahan
- ◆ Untuk menghindari
  - Option Explicit



# Lingkup modul

- ◆ Agar variabel dapat diakses oleh beberapa prosedur harus dideklarasikan sebelum/di luar prosedur-prosedur yang menggunakan variabel tersebut

# contoh

```
Sub CalcMargins1()  
  Range("GrossMarg").Value = GrossMarginCalc  
  Range("NetMarg").Value =  
    NetMarginCalc(Range("FixedCosts").Value)  
End Sub  
Function GrossMarginCalc()  
  Dim totSales  
  Dim totExpenses  
  totSales = Application.Sum(Range("Sales"))  
  totExpenses = Application.Sum(Range("Expenses"))  
  GrossMarginCalc = (totSales - totExpenses) /  
    totSales  
End Function  
Function NetMarginCalc(fixedCosts)  
  Dim totSales  
  Dim totExpenses  
  totSales = Application.Sum(Range("Sales"))  
  totExpenses = Application.Sum(Range("Expenses"))  
  NetMarginCalc = (totSales - totExpenses -  
    fixedCosts) / totSales  
End Function
```

```
Dim totSales2  
Dim totExpenses2  
Sub CalcMargins2()  
  Range("GrossMarg").Value = GrossMarginCalc2  
  Range("NetMarg").Value =  
    NetMarginCalc2(Range("FixedCosts").Value)  
End Sub  
Function GrossMarginCalc2()  
  totSales2 = Application.Sum(Range("Sales"))  
  totExpenses2 =  
    Application.Sum(Range("Expenses"))  
  GrossMarginCalc2 = (totSales2 - totExpenses2) /  
    totSales2  
End Function  
Function NetMarginCalc2(fixedCosts)  
  NetMarginCalc2 = (totSales2 - totExpenses2 -  
    - fixedCosts2) / totSales2  
End Function
```

# Lingkup publik

- ◆ Kadang kita menginginkan semua modul dalam proyek kita dapat mengakses variabel → lingkup publik
- ◆ Caranya: deklarasikan dengan `public namavariabel`

Untuk menggantikan `dim namavariabel`

# Mengirim argumen variabel

- ◆ Bila program terdiri dari banyak prosedur, mendeklarasikan variabel dalam lingkup modul dapat berbahaya, karena nilai variabel dapat diubah oleh setiap prosedur yang mengakses variabel tersebut atau variabel yang sama dideklarasikan berbeda dalam prosedur lain
- ◆ Untuk menghindari kesalahan semacam itu, variabel dapat dikirim dari satu prosedur ke yang lain sebagai argumen
- ◆ Pengiriman argumen variabel dapat berdasar referensi atau berdasar nilai variabel

# Mengirim variabel berdasar referensi

- ◆ Contoh:

```
Sub procedure3()
```

```
  Dim pesan
```

```
  pesan = "Pesan asli!"
```

```
  procedure4 pesan
```

```
  MsgBox pesan
```

```
End Sub
```

```
Sub procedure4(kiriman)
```

```
  MsgBox kiriman
```

```
  kiriman = "pesan telah sampai!"
```

```
End Sub
```

- ◆ Bila procedure3 dijalankan, mula-mula pesan berisi "Pesan asli!" kemudian berisi "pesan telah sampai!"

# Mengirim variabel berdasar nilai

- ◆ Contoh:

```
Sub procedure5()
```

```
  Dim pesan
```

```
  pesan = "Pesan asli!"
```

```
  procedure6 pesan
```

```
  MsgBox pesan
```

```
End Sub
```

```
Sub procedure6(ByVal pesan)
```

```
  MsgBox pesan
```

```
  pesan = "terimakasih atas kirimannya!"
```

```
End Sub
```

- ◆ Bila procedure5 dijalankan, variabel pesan akan **tetap** berisi "Pesan asli!" karena yang dikirim hanya nilainya (**ByVal**)

# Variabel statis

- ◆ Dalam VBA, variabel dalam prosedur setelah digunakan nilainya akan dihapus; ketika prosedur dipanggil lagi, akan diisi/ hitung lagi
- ◆ Bila ingin nilai variabel tak diubah bisa dipakai variabel statis (static variable)

# Contoh variabel statik

```
Sub tesStatik()  
  ProsedurStatik  
  ProsedurStatik  
End Sub  
Sub ProsedurStatik()  
  Static varStatik  
  Dim varReguler  
  
  varStatik = varStatik + 5  
  varReguler = varReguler + 5  
  MsgBox "varStatik = " & varStatik & " dan varReguler = " &  
  varReguler  
End Sub
```

- ◆ Saat dijalankan nilai varStatik mula-mula 5, lalu 10, sedangkan varReguler tetap 5



# Menghindari kesalahan variabel

- ◆ Kadang kita salah mengetik nama variabel, kesalahan ini paling sering terjadi pada pembuatan program yang besar
- ◆ Untuk menghindarinya, gunakan perintah explicit, maka VBA akan memberi pesan error bila ketemu variabel baru yang belum dideklarasikan
- ◆ Caranya:
  - Dengan mendeklarasikan di bagian atas modul  
Option Explicit
  - Pilih Tools|Option → modul general, pilih/contreng Require Variable Declaration

# Jenis data variabel

- ◆ VBA secara otomatis memberi jenis data variant (22 byte) untuk setiap variabel yang dideklarasikan tanpa diberi jenis data
- ◆ Untuk menghemat memori dan menambah efisiensi program jenis data sebaiknya dinyatakan pada saat deklarasi variabel
- ◆ Cara:
  - Dim *namaVariabel* As *JenisData*
- ◆ Contoh:
  - Dim kata As String
  - Dim noMhs As Integer

# Jenis data variabel

Jenis	Ukuran	lambang	Catatan
Boolean	2 bytes		True & false
Byte	1 byte		Nilai 0 sampai 255
Currency	8 bytes	@	Untuk mata uang, nilai antara -922,337,203,685,477.5808 hingga 922,337,203,685,477.5808
Date	8 bytes	#date#	Untuk tanggal, antara 1 Jan 1000 hingga 31 Desember 9999
Double	8 bytes	#	Presisi ganda, bilangan negatif -1.79769313486232E308 hingga -4.9406564581247E-324; bilangan positif dari 4.9406564581247E-324 hingga 1.79769313486232E308
Integer	2 bytes	%	Untuk bil cacah antara -32,768 sampai 32,767
Long	4 bytes	&	Bilangan cacah besar -2,147,483,648 sampai 2,147,483,648
Object	4 bytes		Referensi obyek
Single	4 bytes	!	Bil pecahan presisi tunggal
String	1 byte per char	\$	Untuk teks, dapat sampai 64KByte
Variant	16 byte		Dapat untuk sembarang jenis data

# default jenis data

- ◆ Kita dapat menyatakan jenis data default
- ◆ Contoh:
  - DefInt I-N
  - Berarti semua variabel yang mulai dengan huruf I hingga N (kapital) merupakan integer
- ◆ Keywords:
  - DefBool
  - DefByte
  - DefInt
  - DefLng
  - DefSng
  - DefDbf
  - DefDate
  - DefStr
  - DefObj
  - DefVar

# User-defined data types

- ◆ Kita dapat mendefinisikan jenis data sendiri
- ◆ Contoh:

Type Mahasiswa

nama As String

nomhs As Integer

angkatan As Integer

tglLhr As Date

End Type

Mahasiswa.nama = "Ahmad Musa"

Mahasiswa.nomhs = 22718

Mahasiswa.angkatan = 2002

Mahasiswa.tglLhr = #01/01/1985#

# Variabel array

- ◆ Variabel matriks dan vektor dapat dideklarasikan dengan variabel array
- ◆ Cara:

Dim mhs(200) As String

mhs(0) = "Bambang Yuwono"

mhs(100) = "Banowati Ipit"

Catatan: bisa Option Base 1 → mulai dari larik 1

Atau Dim vektor(50 to 100) As Double

Dim matriks(1 to 50, 1 to 20)

# Array dinamis

- ◆ Bila ukuran array belum diketahui, bisa dideklarasikan sebagai array dinamis
- ◆ Bila dimensinya diketahui bisa dinyatakan kemudian

```
Dim myArray() As Double
```

...

```
ReDim myArray(105)
```

- ◆ ReDim akan me-reinisialisasi nilai variabel dalam array tersebut, bila nilai variabel ingin tak di-reinisialisasi:

```
ReDim Preserve myArray(105)
```

- ◆ Dimensi array bisa diketahui dengan  
LBound(namaArray) → indeks bawah  
UBound(namaArray) → indeks atas ©

# contoh

```
Sub PerformCalculations()
```

```
Dim calcVal() As Double, totVal as Integer
```

- 
- 
- 

```
totVal = GetTotalValue()
```

```
ReDim calcVal(totVal)
```

- 
- 

```
End Sub
```



# konstanta

- ◆ Konstanta built-in → ada di dalam Excel untuk properties dan methods berbagai objects
- ◆ Contoh:  
ActiveWindow.WindowState=xlMaximized
- ◆ User-defined constants

*Public|Private Const NamaKonstanta [ As Type] = expression*

Contoh:

*Public Const gravitasi As Single = 9.81*