

The background is a solid blue color. On the left side, there is a faint, light blue grid pattern that resembles a wireframe sphere or a dome. A small, white pushpin is pinned to the grid, with its head pointing towards the top right and its sharp point directed downwards.

# Excel in a nutshell

# Excel as object

- ◆ Dalam graphical user interface, aplikasi dan bagian-bagiannya dikenal sebagai object
- ◆ Object excel
  - Excel
  - Workbook
  - Worksheet
  - Range dalam worksheet
  - Listbox
  - Chart sheet
  - Chart
  - Dsb.

# Object utama

- ◆ **Workbooks** merupakan “folder” yang berisi:
  - Worksheets
  - Chart sheets
  - Macro sheets (.xlm → obsolete)
  - Dialog sheets (obsolete)
- ◆ **Worksheets** merupakan lembar kerja
  - 1 worksheet terdiri dari 256 kolom dan 65,536 baris (16,777,216 sel!!) → setara dengan 36,000 halaman cetak!!!

# Excel user interface

- ◆ Antarmuka excel bisa dilakukan dengan beberapa cara:
  - Menu
  - Dialog box
  - Toolbars
  - Drag-and-drop
  - Keyboard shortcuts

# Entry data

- ◆ Nilai (angka, tanggal, waktu)
- ◆ Text
- ◆ Rumus
- ◆ Nilai logika (boolean – true-false)
- ◆ Beberapa tips
  - Mengisi data pada range: isikan data pada active cell, tekan ctrl+Enter
  - Mengcopy ke bawah: drag pojok selection
  - Mengcopy ke bawah dgn increment: drag pojok selection sambil tekan ctrl
  - Untuk penulisan teks yang panjang, ganti baris dengan alt+enter
  - Untuk menulis pecahan tulis 0 <spasi> pecahan
  - Tanggal hari ini: ctrl+;
  - Jam saat ini: ctrl+shift+;



# Beberapa kemampuan excel

- ◆ Formatting
- ◆ Formulas
- ◆ Names
- ◆ Functions
- ◆ Shapes
- ◆ Charts
- ◆ Macros
- ◆ Database access → filter

# Analysis tools

- ◆ Outline
- ◆ Automatic subtotals
- ◆ Scenario management
- ◆ Analysis toolpak
- ◆ Pivot table
- ◆ Auditing
- ◆ Solver
- ◆ Add-ins

# cell

- ◆ Relative A1
- ◆ Absolute \$A\$1
- ◆ row absolute A\$1
- ◆ Column absolute \$A1
- ◆ Referencing other sheets/workbooks  
=[data.xls]Sheet2[A2]+1
- ◆ Referensi dengan nama : insert→  
name→ create



# 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, light-colored pushpin is pinned to the grid, with its point resting on the text 'Bagian 1'.

# Bagian 1

## Dasar-dasar pemrograman VBA

# Apa yang dibicarakan?

- ◆ Mengenal VBA
- ◆ IDE
- ◆ Pemrograman berbasis object
- ◆ Cara membuat program VBA
- ◆ Makro
- ◆ Editor program
- ◆ Menjalankan program
- ◆ Membuat fungsion sendiri
- ◆ Membuat prosedur
- ◆ Mengatur editor
- ◆ Melindungi program

# Apa dan kenapa VBA

- ◆ Merupakan bahasa pemrograman bagi semua aplikasi Microsoft Office (Excel, Word, Access, Powerpoint) dan beberapa yang lain (Outlook, Visio, AutoCAD, MapInfo, dsb.)
- ◆ Dapat berfungsi sebagai macro atau bahasa pemrograman
- ◆ Merupakan bahasa program beorientasi object, semua object office dapat dengan mudah dimanipulasi dengan VBA
- ◆ Belajar satu bahasa dapat diaplikasikan dalam banyak aplikasi Windows

# kenapa

- ◆ Mendukung aplikasi internet dan intranet
- ◆ Mendukung ActiveX controls
- ◆ Code editor dengan bantuan yang lengkap dan debugger
- ◆ Form building tools
- ◆ Object browser
- ◆ Dll yang memudahkan pembuatan program



# IDE – integrated development environment

- ◆ Untuk menulis/edit program VBA digunakan IDE

# Module window

Object list

Procedure list

Project explorer

Properties window

The screenshot displays the Microsoft Visual Basic for Applications (VBA) editor interface. The main window is titled "Microsoft Visual Basic - komputer1.xls - [ThisWorkbook (Code)]". The interface is divided into several panes:

- Project Explorer:** Located on the left, it shows a tree view of the project structure. Under "YBAPProject (komputer1.xls)", there are "Microsoft Excel Objects" containing "Sheet1 (Sheet1)", "Sheet2 (Sheet2)", "Sheet3 (Sheet3)", and "ThisWorkbook".
- Properties Window:** Located below the Project Explorer, it shows the properties for the selected "ThisWorkbook" object. The "Alphabetic" tab is active, displaying a list of properties such as "AcceptLabelsInFormulas", "AutoUpdateFrequency", "ChangeHistoryDuration", "ConflictResolution", "Date1904", "DisplayDrawingObjects", "EnableAutoRecover", "EnvelopeVisible", "HasRoutingSlip", "HighlightChangesOnScroll", "IsAddin", "KeepChangeHistory", "ListChangesOnNewSheet", and "Password".
- Module Window:** The main area on the right, titled "Workbook" and "Open", contains VBA code for the "Workbook\_Open" event. The code is as follows:

```
' prosedur pemberian salam saat workbook dibuka
' meminta nama pengguna
Private Sub Workbook_Open()
    Dim pesan, judul, Default, MyValue
    pesan = "Hi, siapa nama anda? " ' minta input nama.
    judul = "Salam pembuka" ' Set judul.
    Default = "tanpa nama" ' Set default.
' Display pesan, judul, dan nilai default.
    MyValue = InputBox(pesan, judul, Default)
    MsgBox ("Assalamu'alaikum " + MyValue)
End Sub
```

Red arrows point from the labels "Object list" and "Procedure list" to the "ThisWorkbook" object in the Project Explorer and the "Workbook\_Open" procedure in the Module Window, respectively.

# Prinsip pemrograman berbasis object

- ◆ Object
  - Workbook, worksheet, chart, cell, dsb.
- ◆ Properties
  - Setiap object memiliki properties: color, font, italic, dsb.
- ◆ Method & procedure
  - Cara/perintah untuk melakukan sesuatu: pindah worksheet, pindah baris, dsb.

# VBA programming

- ◆ Program dalam VBA disebut prosedur (dan function)
- ◆ Tiga jenis prosedur
  - Command macros → mengotomasi prosedur-prosedur, fungsi-fungsi yang ada (built-in) dalam aplikasi, memformat, menyimpan, dsb.
  - User-defined function (function procedure) → program yang berisi rumus dan perintah yang kita susun sendiri
  - Property procedure menunjukkan atau mengubah properti dari suatu object (class module)

# Cara membuat program

- ◆ Dengan merekam macro (macro recording)
- ◆ Dengan menulis program dalam IDE/VBA editor



# Membuat dan menjalankan macro

- ◆ Tools → macro → record macro
- ◆ Lakukan prosedur-prosedur dan perubahan properties pada worksheet
- ◆ Stop macro recording
- ◆ Edit macro untuk memperbaiki dan mengubah hasil rekaman
- ◆ Menjalankan:
  - Dengan shortcut, atau
  - Tools → macro → run → pilih nama macro

# Menulis program

- ◆ Tools → macro → visual basic editor atau
- ◆ Click tombol VBE
- ◆ Pindah dari VBE ke aplikasi → alt+F11
- ◆ Tuliskan program (function atau procedure)
  - Sub
    - ◆ Baris program
  - End sub

# Fungsi

- ◆ Fungsi adalah rumus yang mengolah argumen dan mengembalikan hasil

- ◆ Struktur fungsi

Function *NamaFungsi*(*argumen1*, *argumen2*, ...)

[Pernyataan/statement VBA]

*NamaFungsi* = *hasil yang dikembalikan*

End Function

- ◆ Bagianbagian:

- Statement Function
- Nama fungsi
- Argumen fungsi (parameter)
- Statement VBA yang melakukan hitungan
- Hasil yang dikembalikan → nama fungsi =
- Statement End Function

# Membuat fungsi

- ◆ Contoh: menghitung sisi miring segitiga

- $\text{Sisimiring} = \text{sqr}(x^2 + y^2)$

Function sisimiring(x, y)     ` nama fungsi

    sisimiring = sqr(x ^ 2 + y ^ 2) ` nilai  
    hasil

End Function

- ◆ Pemakaian: isikan pada sel dengan nama formula, misal =sisimiring(3, 4) akan menghasilkan 5

# Membuat fungsi

```
Function akar(A, B, C)
If A <> 0 Then
    diskrim = B ^ 2 - 4 * A * C
    If diskrim > 0 Then
        akar = "dua akar riil"
    ElseIf diskrim = 0 Then
        akar = -B / (2 * A)
    Else
        akar = "akar imajiner"
    End If
Else
    akar = "bukan persamaan kuadrat"
End If
End Function
```



# prosedur

- ◆ Prosedur adalah unit program dalam VBA yang merupakan blok kode dalam satu modul

- ◆ Struktur sebuah prosedur

*Sub NamaProsedur (argumen1,  
argumen2,...)*

[pernyataan/statement VBA]

End Sub

# Membuat prosedur

- ◆ Contoh: perhitungan bunga

- ◆ Sub EnterLoanData()

```
Sub InputDataPinjaman()
```

```
    Range("bunga").Value = 0.08
```

```
    Range("lama").Value = 10
```

```
    Range("pinjaman").Value = 100000000
```

```
    Range("bayar").Value =
```

```
    "=PMT(bunga/12,lama*12,pinjaman)"
```

```
End Sub
```

- ◆ Penggunaan: tools → macro → pilih nama sub InputDataPinjaman → run

# Memfaatkan intellisense

- ◆ Setiap object dalam excel memiliki properties dan methods,
- ◆ Untuk memanipulasi obyek dalam program VBA, kita bisa memanfaatkan intellisense
- ◆ Caranya:
  - Tuliskan nama obyek dilanjutkan dengan titik (.) → akan keluar properties dari obyek tersebut
  - Contoh: application.
  - Kalau kita tekan <esc> intellisense tak akan muncul lagi untuk obyek yang sama, untuk memunculkan lagi → tekan Ctrl+J

# Informasi parameter

- ◆ Setiap fungsi memiliki satu/beberapa parameter, parameter tersebut dapat dimunculkan dengan menuliskan nama fungsi dan tanda kurung buka
  - `activecell.Formula=pmt(` → akan muncul argumen untuk fungsi `pmt`

# Menyimpan/export modul

- ◆ Dari project explorer, pilih modul yang akan disimpan
- ◆ Pilih File|Export File
- ◆ Pilih lokasi tempat menyimpan file
- ◆ Beri nama file.BAS
- ◆ Tekan Save

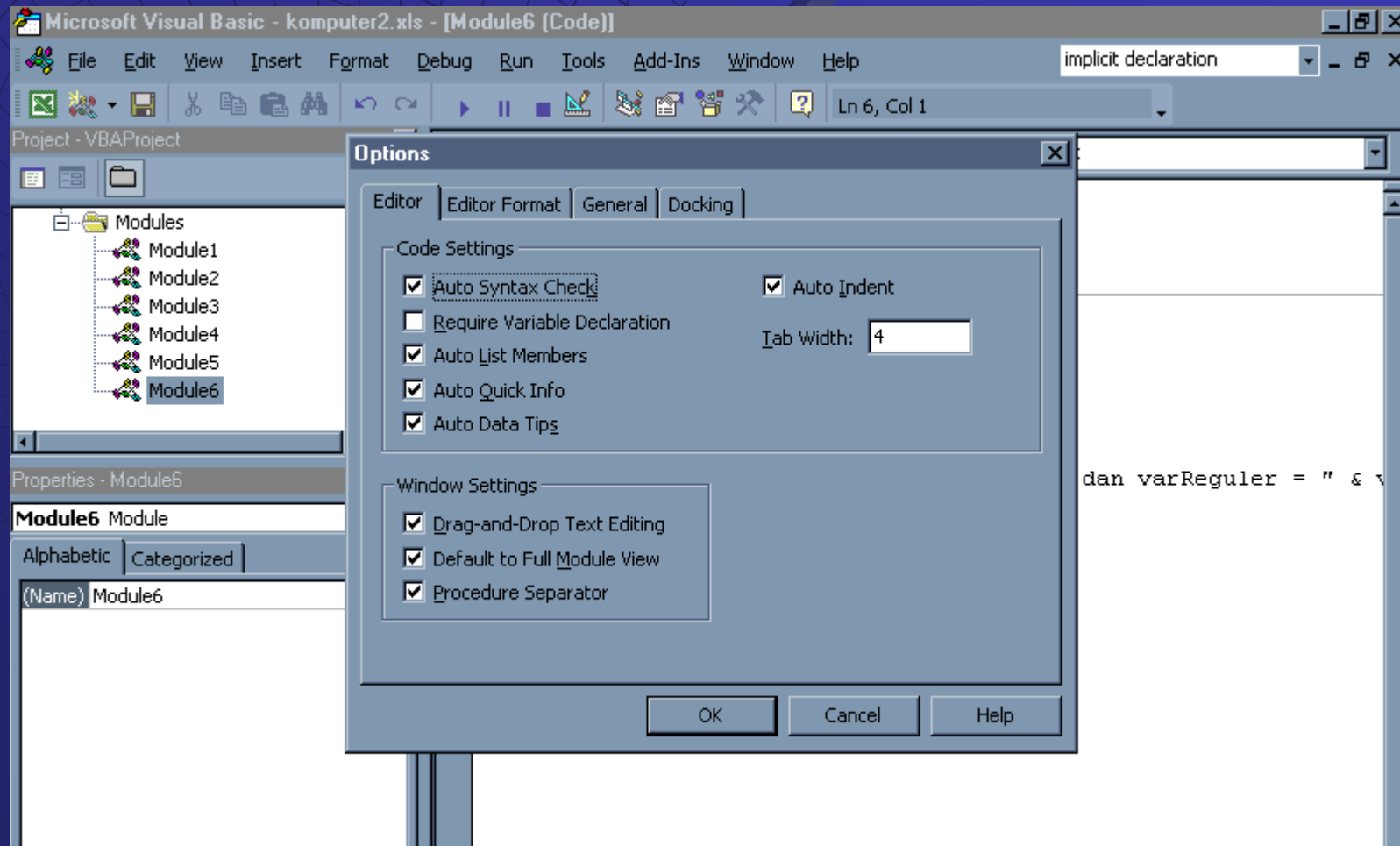


# Mengimpor/ambil modul

- ◆ Dari project explorer
- ◆ Pilih File|Import File
- ◆ Pilih file dari lokasi tempat menyimpan file.BAS
- ◆ Tekan Open

# Mengatur editor

- ◆ Cara kerja dan tampilan editor dapat diatur dengan menu Tools|Options



# Melindungi program

- ◆ Program dapat dilindungi dengan password
- ◆ Tools|ProjectName Properties → dialog box, pilih Protection
- ◆ Isikan password untuk melindungi program
- ◆ OK

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
```

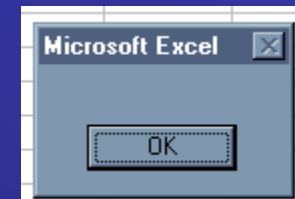
```
Sub Procedure2()
```

```
    ` 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	#	Pressi 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*



# Bagian 3

## Ekspresi VBA

# Apa yang dibicarakan?

- ◆ Struktur ekspresi
- ◆ Operator VBA
- ◆ Bekerja dengan ekspresi numerik
- ◆ Bekerja dengan ekspresi tex/string
- ◆ Bekerja dengan ekspresi logika
- ◆ Bekerja dengan ekspresi tanggal
- ◆ format

# Struktur ekspresi VBA

- ◆ Variabel = statemen tentang variabel

contoh:

```
energy = mass * (speedOfLight ^ 2)
```

operand <assignment> operand &  
operator

Operand harus menggunakan jenis variabel yang kompatibel

Operator harus sesuai dengan jenis data



# Operator VBA

## ◆ Aritmetik

+	tambah/plus
-	kurang/minus
-	negasi
*	kali
/	bagi
\	pembagian integer
^	pangkat
Mod	modulus $10 \text{ Mod } 5 = 0$

## ◆ Konkatenasi untuk jenis data string "Soft" & "ware"

# Operator VBA

- ◆ Pembanding menghasilkan nilai True atau False
  - = sama dengan  $10=5 \rightarrow \text{false}$
  - > Lebih besar dari
  - < Lebih kecil dari
  - >= Lebih besar atau sama dengan
  - <= Lebih kecil atau sama dengan
  - <> Tidak sama dengan
  - Like mirip Jo?n Like John  $\rightarrow \text{True}$

# Operator logika

- ◆ Operator logika menghasilkan nilai True atau False

And	$e1 \text{ And } e2$	T if both true
Eqv	$e1 \text{ Eqv } e2$	T if both T or both F
Imp	$e1 \text{ Imp } e2$	F if $e1$ T and $e2$ F
Or	$e1 \text{ Or } e2$	T if at least one is T
Xor	$e1 \text{ Xor } e2$	F if both T or both F
Not	Not expr	T if expr F

Hierarki: ^; (negasi) ; \* dan /; \; Mod; + dan -;  
&; comparison; Logical

# Fungsi matematik

- ◆ Fungsi matematika dasar:

- Abs(x)
- Atn(x)
- Cos(x)
- Exp(x)
- Fix(x)
- Hex(x)
- Hex\$(x)
- Int(x)
- Log(x)
- Oct(x)
- Oct\$(x)
- Rnd(x)
- Sgn(x)
- Sin(x)
- Sqr(x)
- Tan(x)

# Fungsi finansial

DDB(cost, salvage, life, period, factor)	Menghitung depresiasi aset dengan metode double declining balance
FV(rate, nper, pmt, pv, type)	Menghitung nilai masa depan dari investasi atau pinjaman
IPmt(rate,per,nper, pv, fv, type)	Pembayaran bunga untuk masa pinjaman tertentu
IRR(values, guess)	Nilai internal rate of return serangkaian cash-flows
MIRR(values, finance_rate, reinvest_rate)	IRR yang dimodifikasi untuk cash flow periodik
NPer(rate, pmt, p, fv, type)	Jumlah periode investasi atau pinjaman
NPV(rate, value1, value2,...)	Nett present value serangkaian investasi
Pmt(rate,nper, pv, fv, type)	Pembayaran periodik sejumlah pinjaman atau investasi
PPmt(rate, per, nper, pv, fv, type)	Pembayaran pokok pinjaman untuk masa tertentu
PV(rate, nper, pmt, fv, type)	Nilai sekarang sejumlah investasi
Rate(nper, pmt, pv, fv, type, guess)	Bunga periodik pinjaman atau investasi
SLN(cost, salvage, life)	Depresiasi garis lurus suatu aset
SYD(cost, salvage, life, period)	Depresiasi sum-of-year-digits suatu aset



# Fungsi string

ASC(string)	Karakter ANSI huruf pertama <i>string</i>
Chr(charcode)	Karakter (sbg varian) untuk kode ANSI charcode
Chr\$(charcode)	Karakter (sebagai string) untuk kode ANSI charcode
CStr(expression)	Konversi expression ke nilai string
InStr(start, string1, string2)	Posisi karakter kejadian string1 dalam string2 dihitung dari start
InStrB(start, string1, string2)	Posisi byte kejadian string1 dalam string2 dihitung dari start
LCase(string)	Mengubah ke huruf kecil sbg varian
LCase\$(string)	Mengubah ke huruf kecil sbg string
Left(string, length)	Posisi paling kiri dari string sbg varian
Left\$(string, length)	Posisi paling kiri dari string sbg string
Len(string)	Panjang karakter
LTrim(string)	Menghilangkan spasi
Mid(string, start, length)	karakter sepanjang length dari string mulai start
Dst.	

# Bekerja dengan ekspresi logika

- ◆ Operator And
  - $\text{Expr1 And Expr2} \rightarrow T$  bila keduanya T
- ◆ Operator Or
  - $\text{Expr1 Or Expr2} \rightarrow T$  bila salah satu T
- ◆ Operator XOr
  - $\text{Expr1 XOr Expr2} \rightarrow T$  bila satu T dan lainnya F
- ◆ Operator Eqv
  - $\text{Expr1 Eqv Expr2} \rightarrow T$  bila keduanya T atau keduanya F
- ◆ Operator Imp (Implikasi)
  - $\text{Expr1 Imp Expr2} \rightarrow T$  bila Expr2 T atau keduanya F

# Bekerja dengan tanggal

- ◆ Tanggal dalam VBA disimpan dalam bentuk rangkaian angka, dengan 31 Desember 1899 sebagai awal (sembarang) sehingga 1 Januari 1900 adalah 1; 366 adalah 31 Desember 1900; 37,980 adalah 25 Desember 2003; dst.
- ◆ Saat (jam, menit, detik) dinyatakan dalam pecahan, dengan 0 adalah tengah malam; 0.5 adalah tengah hari

# Format

- ◆ Untuk memformat angka/string yang muncul bisa digunakan perintah

`Format(expresi, format)`

Contoh:

`Format(Pemasukan –  
Pengeluaran, "currency")`

`Format(Now, "medium time")`

`Format(NoMhs, "00000/TS")`



# Bagian 4

## Bekerja dengan Objects



# kajian

- ◆ Pengertian Object
- ◆ Hirarki object
- ◆ Bekerja dengan properti object
- ◆ Bekerja dengan metode object
- ◆ Menangani kejadian object
- ◆ Bekerja dengan koleksi object
- ◆ Browser object
- ◆ Mereferensi pustaka object tambahan
- ◆ Memberikan object pada variabel
- ◆ Operator Is
- ◆ Bekerja dengan beberapa properties dan methods
- ◆ Object aplikasi
- ◆ Object windows

# Pengertian object

- ◆ Object adalah segala sesuatu dalam aplikasi Windows yang dapat dilihat dan dimanipulasi/diubah
- ◆ Object bisa berupa satu object atau kumpulan (*collection*) object
- ◆ Contoh: workbook, worksheet, window, cell, range, gambar, grafik, dsb.

# cara memanipulasi oject

- ◆ Cara memanipulasi object:
  - Mengubah *properties*
  - Mengaktivasi *method* yang terkait dengan object
  - Mendefinisi prosedur yang dijalankan saat suatu *event* terjadi
- ◆ Ilustrasi/analogi
  - Komputer: ***object***
  - Merek, jenis prosesor: ***property***
  - Menulis surat, main game: ***method***
  - Dihidupkan, dibuka, dimatikan: ***event***

# Hirarki object

- ◆ Object dalam aplikasi Office memiliki hirarki dari yang paling umum hingga yang lebih spesifik;
- ◆ Object paling umum: Application object → program itu sendiri. Dalam Excel Application object memiliki > 15 objects

# Objects di bawah Application object

<b>Object</b>	<b>Collection</b>	<b>Description</b>
AddIn	AddIns	Semua file add-in/tambahan yang ada dalam Excel
Dialog	Dialogs	Kotak dialog yang built-in dalam Excel
Name	Names	Koleksi nama-nama yang ada dalam semua workbooks yang terbuka
Window	Windows	Kumpulan windows yang terbuka
Workbook	Workbooks	Koleksi dari semua workbooks yang terbuka
WorksheetFunction		Container untuk fungsi-fungsi worksheet Excel



# Bekerja dengan object properties

- ◆ Sintaks: *Object.Property*
- ◆ Contoh: *Application.ActiveWindow*
- ◆ Masalah:
  - *Ada beberapa propertes yang juga merupakan object*

## Application Object

Properties .  
ActiveMenuBar  
ActiveWindow  
Calculation  
Etc.

## Window Object

Properties .  
ActiveCell  
Caption  
DisplayGridlines  
Etc.

## Range Object

Properties .  
ColumnWidth  
Font  
Formula  
Etc.

## Font Object

Properties .  
Italic  
Name  
Size  
Etc.

Application.ActiveWindow.ActiveCell.Font.Italic  
sama dengan  
ActiveCell.Font.Italic

# Menetapkan nilai property

- ◆ Sintax: *Object.Property = value*
- ◆ *Value* berupa nilai pernyataan untuk menset property, bisa berupa :
  - Numerik: `ActiveCell.Font.Size = 14`
  - String: `ActiveCell.Font.Name = "Arial"`
  - Logika: `ActiveCell.Font.Italic = True`
- ◆ Nilai property dapat dicari dengan
  - *variabel = Object.Property*

# Bekerja dengan methods

- ◆ Method menggambarkan apa yang dilakukan object; contoh: metode sort, pivot table, simpan, buka, dsb.
- ◆ Sintax: *Object.Method(arg1, arg2,...)*  
contoh: `ActiveWorkbook.Save`  
`ActiveWorkbook.Close(SaveChanges, fileName, RouteWB)`

# Mengelola object events

- ◆ *Event* adalah segala sesuatu yang terjadi pada object, seperti: membuka workbook adalah event pada workbook
- ◆ Respons terhadap *event* yang terjadi disebut *event handlers*
- ◆ Event handlers akan dijalankan bila event terjadi pada suatu object



# Contoh event handler

' prosedur pemberian salam saat workbook dibuka

' meminta nama pengguna

```
Private Sub Workbook_Open() ' event membuka  
Workbook
```

```
Dim pesan, judul, Default, MyValue
```

```
pesan = "Hi, siapa nama anda? " ' minta input  
nama.
```

```
judul = "Salam pembuka" ' Set judul.
```

```
Default = "tanpa nama" ' Set default.
```

' Display pesan, judul, dan nilai default.

```
MyValue = InputBox(pesan, judul, Default)
```

```
MsgBox ("Assalamu'alaikum " + MyValue)
```

```
End Sub
```

# Mengelola event

Contoh:

*' prosedur pemberian salam saat workbook dibuka*

*' meminta nama pengguna*

Private Sub Workbook\_Open() *'event*

*membuka wb*

Dim pesan, judul, Default, MyValue

pesan = "Hi, siapa nama anda? " *' minta*

*input nama.*

judul = "Salam pembuka" *' Set*

*judul.*

Default = "tanpa nama" *' Set default.*

*' Display pesan, judul, dan nilai default.*

MyValue = InputBox(pesan, judul, Default)

MsgBox ("Assalamu'alaikum " + MyValue) *Nizam ©*

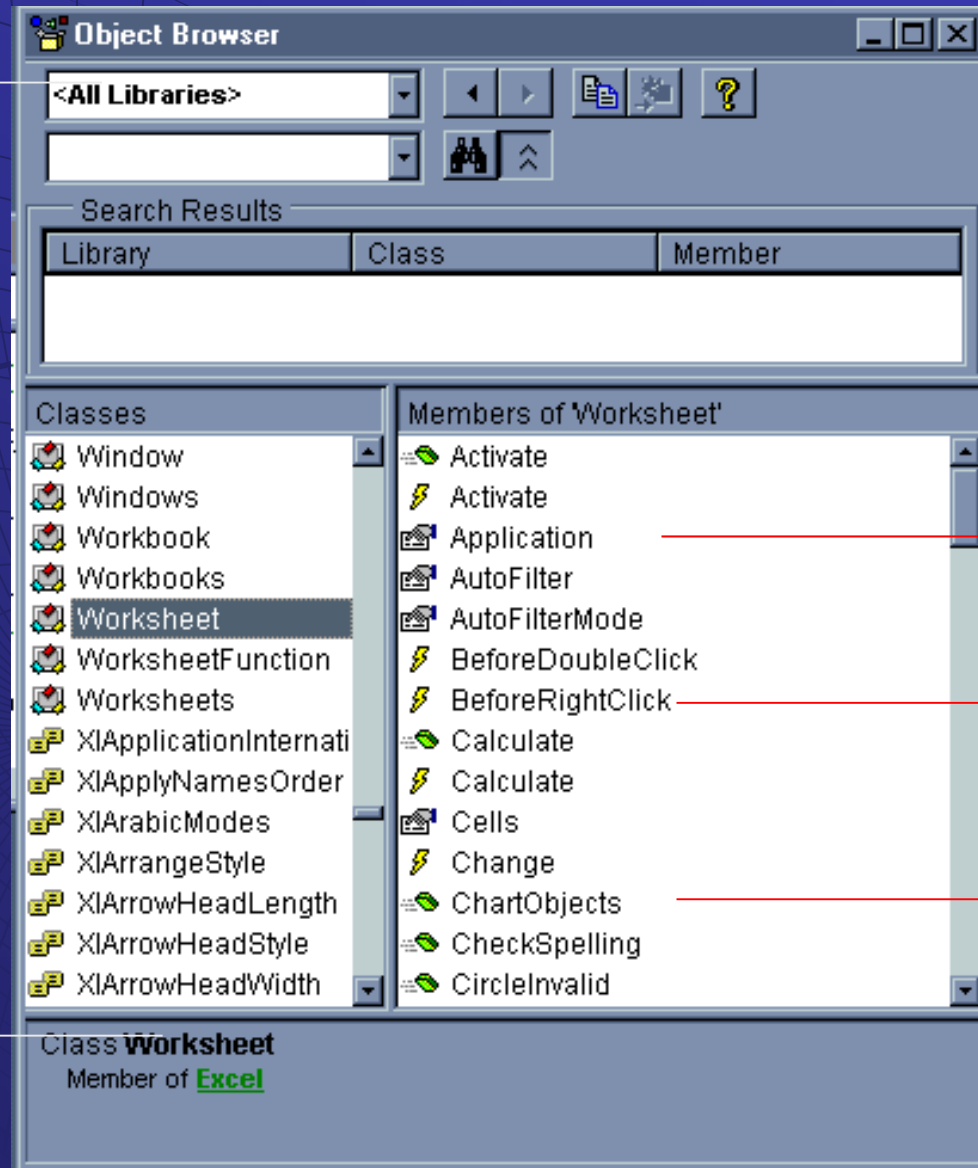
End Sub

# Bekerja dengan koleksi object

- ◆ Collection adalah himpunan objects yang serupa; seperti WorkBooks Collection merupakan koleksi workbook yang terbuka
- ◆ *Collection* juga merupakan *object* sehingga memiliki *properties* dan *methods* yang bisa dimanipulasi
- ◆ Anggota *collection* disebut *element* yang bisa diakses dengan namanya atau dengan index
- ◆ Contoh:  
`Workbooks("Budget.xls").Close`  
`ActiveSheet.Picture(1).Copy`
- ◆ Kalau tak diberi index, diasumsikan semua koleksi diperlakukan dengan method yang sama

# Object browser

Libraries  
dan Project



Properties

Events

Methods

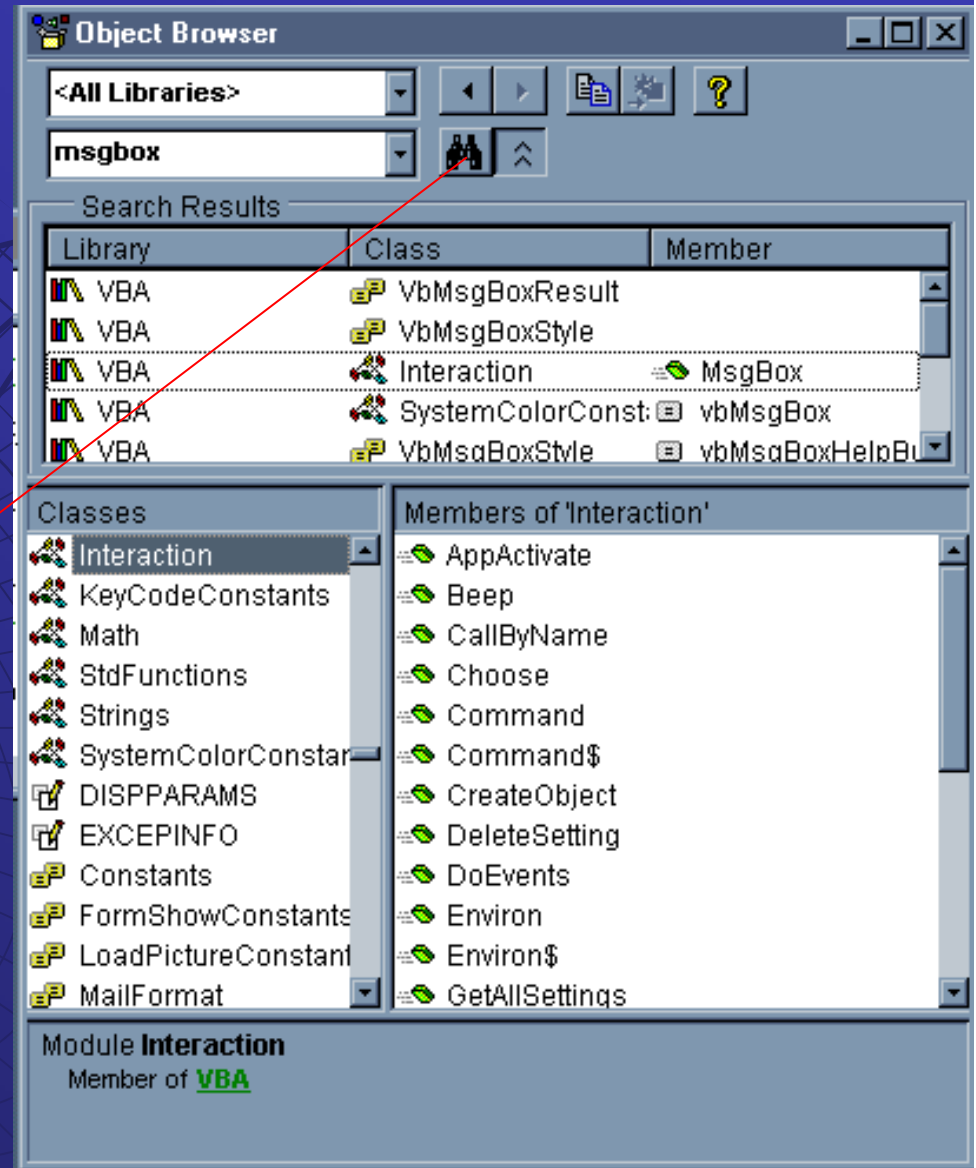
Template  
Code

Nizam ©

# Bekerja dengan object browser

- ◆ Menunjukkan *objects* yang ada pada satu library, *properties*, *methods* dan *events* yang terkait dengan object

Pencari object





# Memasukkan object dalam variabel

- ◆ Sintax: `variableName = ObjectName`

Dim budgetSheet As Object

Set budgetSheet = \_  
    Workbooks("Budget.xls").Worksheets("2003  
    Budget)

Lebih baik:

Dim budgetSheet As Worksheet

Set budgetSheet = \_  
    Workbooks("Budget.xls").Worksheets("2003  
    Budget)

# Operator Is

- ◆ Kita kadang perlu membandingkan beberapa objects, VBA menyediakan operator Is untuk membandingkan dua objects
- ◆ *Hasil = Object1 Is Object2*
  - *Object1* dan *Object2* adalah object atau variabel, bila sama, *hasil* akan *True*, bila beda akan *False*

# Bekerja dengan properties dan methods yang banyak (1/3)

## Contoh:

```
Sub FormatRange()
```

```
Worksheets("Sheet1").Range("B2:B5").Style = "Currency"
```

```
Worksheets("Sheet1").Range("B2:B5").WrapText= True
```

```
Worksheets("Sheet1").Range("B2:B5").Font.Size = 16
```

```
Worksheets("Sheet1").Range("B2:B5").Font.Bold = True
```

```
Worksheets("Sheet1").Range("B2:B5").Font.Color = RGB(255,  
0, 0)
```

```
Worksheets("Sheet1").Range("B2:B5").Font.Name =  
"Arial"
```

```
End Sub
```

# Bekerja dengan properties dan methods yang banyak (2/3)

## Contoh yang lebih baik:

```
Sub FormatRange2()
```

```
    With Worksheets("Sheet1").Range("B2:B5")
```

```
        .Style = "Currency"
```

```
        .WrapText = True
```

```
        .Font.Size = 16
```

```
        .Font.Bold = True
```

```
        .Font.Color = RGB(255, 0, 0)
```

```
        .Font.Name = "Arial"
```

```
    End With
```

```
End Sub
```

# Bekerja dengan properties dan methods yang banyak (3/3)

## Contoh yang lebih baik lagi:

```
Sub FormatRange3()  
  With Worksheets("Sheet1").Range("B2:B5")  
    .Style = "Currency"  
    .WrapText = True  
    With .Font  
      .Size = 16  
      .Bold = True  
      .Color = RGB(255, 0, 0)  
      .Name = "Arial"  
    End With  
  End With  
End Sub
```



# Object aplikasi

- ◆ *Application Object* merupakan *container* yang berisi semua object dalam aplikasi yang dijalankan; object ini memiliki beberapa *properties* dan *methods* yang banyak gunanya
- ◆ Beberapa contoh
  - `Application.ActivePrinter`: mengembalikan himpunan printer driver yang tersedia
  - `Application.ActiveWindow`: mengembalikan window yang sedang aktif
  - `Application.Caption`: mengembalikan judul aplikasi
  - `Application.Dialogs`: built-in dialog yang ada dlm aplikasi
  - `Application.Visible`: mengatur aplikasi terlihat/tidak
  - `Application.Width`: lebar window aplikasi
  - `Application.Windows`: kumpulan window aplikasi yang aktif

# Methods pada Application Object

- ◆ Application object memiliki beberapa methods, diantaranya
  - Application.Help: mendisplay sistem Help aplikasi
  - Application.Quit: keluar dari aplikasi
  - Application.Repeat: mengulang kegiatan yang baru saja dilakukan
  - Application.EnableCancelKey

# Object Window

- ◆ Menspesifikasi object window
  - Dengan indeks, atau dengan menuliskan namanya
    - ◆ Windows(1)
    - ◆ Windows("Nilai.xls"), atau
    - ◆ ActiveWindow → window yang saat ini sedang aktif
- ◆ Membuka window baru
  - Window.NewWindow

# Object Window

- ◆ Beberapa properties object window
  - Window.Caption
  - Window.Height
  - Window.Left
  - Window.Top
  - Window.UsableHeight
  - Window.UsableWidth
  - Window.Visible
  - Window.Width
  - Window.WindowNumber
  - Window.WindowState
- ◆ Beberapa methods object window
  - Window.Activate →  
Windows("Nilai.xls").Activate
  - Window.Close
  - Window.LargeScroll(Down,Up,ToRight,ToLeft)
  - Window.SmallScroll(Down,Up,ToRight,ToLeft)



# Bagian 5

## Mengendalikan alur program



# Apa yang akan dibahas?

- ◆ Pengendalian alur program
- ◆ If...Then
- ◆ If...Then...Else
- ◆ Membuat keputusan ganda
- ◆ Fungsi yang mengambil keputusan
- ◆ Loops
- ◆ Do...Loop
- ◆ For...Next
- ◆ For Each...Next
- ◆ Exit

# Pengendalian alur program

- ◆ Alur program dikendalikan dengan logika pengambilan keputusan (True atau False)
- ◆ If..Then
  - Langsung/sebaris  
*If condition Then statement*
  - Dengan blok  
*If condition Then*  
*[statements]*  
*End If*

# Pengendalian alur program

- ◆ If...Then...Else

- Sintaks

```
If condition Then  
    [TrueStatements]  
Else  
    [FalseStatements]  
End If
```

Contoh:

```
Function FutureValue(Rate, Nper, Pmt, Frequency)  
    If Frequency = "Monthly" Then  
        FutureValue = FV(Rate / 12, Nper * 12, Pmt / 12) ` freq  
        monthly  
    Else  
        FutureValue = FV(Rate / 4, NPer * 4, Pmt / 4) ` freq  
        quarterly  
    End If  
End Function
```

# Keputusan ganda

- ◆ Penggunaan operator And dan Or

```
If condition1 And condition2 Then  
    [TrueStatements]
```

```
Else
```

```
    [FalseStatements]
```

```
End If
```

Contoh:

```
Function FutureValue2(Rate, Nper, Pmt, Frequency)
```

```
    If Frequency <> "Monthly" And Frequency <> "Quarterly" Then
```

```
        MsgBox "The Frequency argument must be either " & ""Monthly""
```

```
or_
```

```
        ""Quarterly""!"
```

```
        Exit Function
```

```
    End If
```

```
    If Frequency = "Monthly" Then
```

```
        FutureValue2 = FV(Rate / 12, Nper * 12, Pmt / 12) ` freq monthly
```

```
    Else
```

```
        FutureValue2 = FV(Rate / 4, NPer * 4, Pmt / 4) ` freq quarterly
```

```
    End If
```

```
End Function
```

Nizam ©

# Keputusan ganda

- ◆ Penggunaan statement multi If...Then...Else

```
If condition1 Then
    [condition1 TrueStatements]
ElseIf condition2
    [condition2 TrueStatements]
<etc.>
Else
    [FalseStatements]
End If
```

## Contoh:

```
Function FutureValue3(Rate, Nper, Pmt, Frequency)
    If Frequency = "Monthly" Then
        FutureValue3 = FV(Rate / 12, Nper * 12, Pmt / 12) ` freq monthly
    ElseIf Frequency = "Quarterly" Then
        FutureValue3 = FV(Rate / 4, NPer * 4, Pmt / 4) ` freq quarterly
    Else
        MsgBox "The Frequency argument must be either " & _
            ""Monthly"" or ""Quarterly""!"
    End If
End Function
```



# Keputusan ganda

- ◆ Penggunaan Select Case

```
Select Case TestExpression
Case FirstCaseList
  [FirstStatements]
Case SecondCaseList
  [SecondStatements]
<etc.>
Case Else
  [ElseStatements]
End Select
```

## Contoh:

```
Function FutureValue4(Rate, Nper, Pmt, Frequency)
  Select Case Frequency
    Case "Monthly"
      FutureValue4 = FV(Rate / 12, Nper * 12, Pmt / 12) ` freq
monthly
    Case "Quarterly"
      FutureValue3 = FV(Rate / 4, NPer * 4, Pmt / 4) ` freq
quarterly
    Case Else
      MsgBox "The Frequency argument must be either " & _
        ""Monthly"" or ""Quarterly""!"
  End Select
End Function
```

# Keputusan ganda

- ◆ Contoh lain: mengubah nilai angka ke huruf  
Function nilHuruf(nilAngka As Integer) As String  
Select Case nilAngka  
Case Is < 0  
nilHuruf = "Salah data nilai kurang dari 0!"  
Case Is < 50  
nilHuruf = "F"  
Case Is < 60  
nilHuruf = "D"  
Case Is < 70  
nilHuruf = "C"  
Case Is < 80  
nilHuruf = "B"  
Case Is <= 100  
nilHuruf = "A"  
Case Else  
nilHuruf = "Salah! Nilai lebih dari 100!"  
End Select  
End Function

# Fungsi pengambil keputusan

## ◆ Statement IIf (Inline If)

- Sintaks: `IIf (condition, TrueResult, FalseResult)`

```
Function ProesorCacat() As Boolean
```

```
    ProesorCacat = IIf((4195835 - (4195835/3145727) * 3145727),  
        True, False)
```

```
End Function
```

## ◆ Fungsi Choose

- Sintaks: `Choose(index, value1, value2, ...)`

```
Function NamaHari(noHari As Integer) As String
```

```
    NamaHari = Choose(noHari, "Ahad", "Senin", "Selasa", "Rabu", "Kamis",  
        "Jumat", _ "Sabtu")
```

```
End Function
```

# Fungsi pengambil keputusan

## ◆ Fungsi Switch

- Sintaks: `Switch(expr1, value1, value2, ...)`

```
Function NilHuruf2(nilAngka As Integer) As String
```

```
NilHuruf2 = Switch(nilAngka < 0 "Error <0", _
```

```
    nilAngka < 50, "F", _
```

```
    nilAngka < 60, "D", _
```

```
    nilAngka < 70, "C", _
```

```
    nilAngka < 80, "B", _
```

```
    nilAngka <= 100, "A", _
```

```
    nilAngka > 100, "Error > 100")
```

```
End Function
```

# Pengulangan - loop

## ◆ Struktur Do...Loop

## ◆ Sintaks:

- Do While *condition*  
  [*statement*]  
  Loop
  - ' check dulu baru loop
  - ' dilakukan selama kondisi T
- Do  
  [*statement*]  
  Loop While *condition*
  - ' lakukan dulu baru check
  - ' diulang bila kondisi T
- Do Until *condition*  
  [*statemnt*]  
  Loop
  - ' check dulu baru loop
  - ' dilakukan selama kondisi F
- Do  
  [*statement*]  
  Loop Until *condition*
  - ' dilakukan dulu baru check
  - ' diulang bila kondisi masih F



# Pengulangan - loop

- ◆ Contoh

```
Sub BigNumbers()
```

```
Dim rowNum As Integer, colNum As Integer, currCell As Range
```

```
rowNum = ActiveCell.Row
```

```
colNum = ActiveCell.Column
```

```
Set currCell = ActiveSheet.Cells(rowNum, ColNum)
```

```
Do While currCell.Value <> ""
```

```
    If IsNumeric(currCell.Value) Then
```

```
        If currCell.Value >= 1000 Then
```

```
            currCell.Font.Color = VBAColor("magenta")
```

```
        End If
```

```
        rowNum = rowNum + 1
```

```
        Set currCell = ActiveSheet.Cells(rowNum, colNum)
```

```
    Loop
```

```
End Sub
```

# For ... Next

- ◆ Loop dengan For ... Next
- ◆ Sintaks: For *counter* = *start* To *end* [*inkremen*]  
    [statement]  
    Next [*counter*]

```
Sub LoopTest()  
    Dim counter As Integer  
    For counter = 1 To 10  
        Application.StatusBar = "Counter value: " &  
        counter  
        Application.Wait Now +  
        TimeValue("00:00:01")  
    Next counter  
    Application.StatusBar = False  
End Sub
```

# For Each ... Next

- ◆ Loop dengan For Each ... Next
- ◆ Sintaks: For Each *element* In *group*  
    [statement]  
Next [*element*]

```
Sub convertProper()  
    Dim cellObject As Object  
    For Each cellObject In Selection  
        cellObject.Formula =  
        Application.Proper(cellObject)  
    Next  
End Sub
```

# Penggunaan Exit For atau Exit Do

- ◆ Kadang kita perlu keluar dari Loop
- ◆ Caranya: gunakan Exit For atau Exit Do

```
Sub BigNumbers2()
```

```
Dim rowNum As Integer, colNum As Integer, currCell As Range
```

```
rowNum = ActiveCell.Row
```

```
colNum = ActiveCell.Column
```

```
Set currCell = ActiveSheet.Cells(rowNum, ColNum)
```

```
Do While currCell.Value <> ""
```

```
    If IsNumeric(currCell.Value) Then
```

```
        If currCell.Value >= 1000 Then
```

```
            currCell.Font.Color = VBAColor("magenta")
```

```
        End If
```

```
    Else
```

```
        Exit Do
```

```
    Do loop
```

```
    End If
```

```
    rowNum = rowNum + 1
```

```
    Set currCell = ActiveSheet.Cells(rowNum, colNum)
```

```
Loop
```

```
End Sub
```

' bila bukan angka keluar dari

# Tugas - 1

- ◆ Salah satu cara untuk mencari akar persamaan  $f$  adalah dengan metode Newton
  - Bila  $f(x) = 0$  adalah fungsi yang akan dicari akarnya, dan  $x_n$  adalah perkiraan akar  $f(x)$  maka pendekatan nilai akar persamaan berikutnya dapat dicari dengan rumus

$$x_{n+1} = x_n - f(x_n)/f'(x_n)$$

Di mana  $f'(x_n)$  adalah nilai turunan fungsi di  $x_n$   
Tuliskan prosedur dalam VBA untuk mencari akar fungsi  $f(x) = 0 \rightarrow$  yang merupakan fungsi yang dapat didefinisikan dalam program (user-defined)



# Tugas - 2

- ◆ Buat sekumpulan program dan fungsi untuk melakukan operasi bilangan kompleks: penjumlahan, pengurangan, perkalian, pembagian

# Tugas 3

- ◆ Buat satu program aplikasi VBA excel untuk bidang teknik sipil/fisika/mekanika