Pemrograman Komputer

oleh
Djoko Luknanto
Departemen Teknik Sipil dan Lingkungan
Fakultas Teknik Universitas Gadjah Mada

Jenis data dalam VBA (1/2)

- Boolean, depends on implementing platform: True or False
- Byte, 1 byte: 0 through 255 (unsigned)
- Char (single character), 2 bytes: 0 through 65535 (unsigned)
- Date, 8 bytes: 0:00:00 (midnight) on January 1, 0001 through 11:59:59 PM on December 31, 9999
- Decimal, 16 bytes: 0 through +/-79,228,162,514,264,337,593,543,950,335 (+/-7.9...E+28) † with no decimal point; 0 through +/-7.9228162514264337593543950335 with 28 places to the right of the decimal; smallest nonzero number is +/-0.0000000000000000000000000001 (+/-1E-28) †
- Double (double-precision floating-point), 8 bytes: -1.7976931348623157OE+308 through -4.94065645841246544E-324 † for negative values; 4.94065645841246544E-324 through 1.7976931348623157OE+308 † for positive values
- Integer, 4 bytes: -2,147,483,648 through 2,147,483,647 (signed)
- Long (long integer), 8 bytes: -9,223,372,036,854,775,808 through 9,223,372,036,854,775,807 (9.2...E+18 †) (signed)

Jenis data dalam VBA (2/2)

- Object (class), 4 bytes on 32-bit platform; 8 bytes on 64-bit platform: Any type can be stored in a variable of type Object
- Sbyte, 1 byte: -128 through 127 (signed)
- Short (short integer), 2 bytes: -32,768 through 32,767 (signed)
- Single (single-precision floating-point), 4 bytes: -3.4028235E+38 through -1.401298E-45 † for negative values; 1.401298E-45 through 3.4028235E+38 † for positive values
- String (variable-length), Depends on implementing platform: 0 to approximately 2 billion Unicode characters
- Uinteger, 4 bytes: 0 through 4,294,967,295 (unsigned)
- Ulong, 8 bytes: 0 through 18,446,744,073,709,551,615 (1.8...E+19 †) (unsigned)
- User-Defined (structure), Depends on implementing platform: Each member of the structure has a range determined by its data type and independent of the ranges of the other members
- Ushort, 2 bytes: 0 through 65,535 (unsigned)
- Sumber: https://msdn.microsoft.com/en-us/library/47zceaw7.aspx

Jenis Data Khusus

- Cara Pertama: Privat, berlaku pada sub () yang mendefiniskan

```
Private Type SystemInfo
    CPU As Variant
    Memory As Long
    VideoColors As Integer
    Cost As Currency
    PurchaseDate As Variant
End Type
```

- Cara menggunakan

```
Dim MySystem As SystemInfo,
YourSystem As SystemInfo
```

- Cara Kedua: Publik, berlaku umum

```
Public Type DataMhs
    Nama As String
    TanggalLahir As Variant
    TempatLahir As String
    NoMhs As Integer
    GolDarah As Char
End Type
```

- Cara menggunakan

```
Dim Datamu As DataMhs,
Dataku As DataMhs
```

Struktur Program VBA

```
Sub ProsesAll()
    BersihkanSemuaSheets
    HapusSemuaNama
    CheckData
    If NError > 0 Then Exit Sub 'Kondisi Error
    Application.StatusBar = "Proses: Data Masukan ..."
    ProsesDataMasukan
    Application.StatusBar = "Proses: Statistik Dasar ..."
    StatistikDasar
    Application.StatusBar = "Proses: Statistik Dasar ..."
    UjiSmirnovKolmogorov
    Application.StatusBar = "Proses: Chi Kuadrat ..."
    UjiChiKuadrat
    Application.StatusBar = "Proses: Hitungan Akhir ..."
    HitungKalaUlang
    Application.StatusBar = "Proses: Pembuatan Kurva ..."
    ProsesSemuaKurva
    Sheets("Kala Ulang").Activate
    Application.StatusBar = HelloString
End Sub
```

Rekursi

```
Function Faktorial(ByVal n As Integer) As Integer
   If n <= 1 Then
      Faktorial = 1
   Else
      Faktorial = n*Faktorial(n - 1)
   End If
End Function
```

Versi rekursif

$n! = n*(n-1)!$

```
Function Faktorial(ByVal n As Integer) As Integer
Dim I, Hasil As Integer
    Hasil = 1
    For I = 2 To n
        Hasil = Hasil*I
    Next I
    Faktorial = Hasil
 End Function
```

Versi looping

$n! = n*(n-1)*(n-2)* \ldots *1$

Contoh implementasi dalam Excel

Hitungan Faktorial			
n	n!	Rekursif	Loop
	=FACT(A5)	=RFaktorial(A5)	=Faktorial(A5)
0	1	1	1
1	1	1	1
2	2	2	2
3	6	6	6
4	24	24	24
5	120	120	120
6	720	720	720
7	5.040	5.040	5.040
8	40.320	40.320	40.320
9	362.880	362.880	362.880
10	3.628.800	3.628.800	3.628.800
11	39.916.800	39.916.800	39.916.800
12	479.001.600	479.001.600	479.001.600
13	6.227.020.800	6.227.020.800	6.227.020.800
14	87.178.291.200	87.178.291.200	87.178.291.200
15	1.307.674.368.000	1.307.674.368.000	1.307.674.368.000
16	20.922.789.888.000	20.922.789.888.000	20.922.789.888.000
17	355.687.428.096.000	355.687.428.096.000	355.687.428.096.000
18	6.402.373.705.728.000	6.402.373.705.728.000	6.402.373.705.728.000
19	121.645.100.408.832.000	121.645.100.408.832.000	121.645.100.408.832.000
20	2.432.902.008.176.640.000	2.432.902.008.176.640.000	2.432.902.008.176.640.000

Pertimbangan memilih rekursif

- **Lebih sederhana**. Biasanya rekursif digunakan karena pemrogaman jauh lebih sederhana.
- **Kondisi akhir**. Fungsi rekursif harus dirancang agar dapat berhenti pada satu kondisi agar tidak berputar terus menerus.
- **Penggunaan Memori**. Setiap aplikasi mempunyai RAM untuk variabel lokal yang terbatas. Setiap pemanggilan rekursif akan menambah kebutuhan RAM untuk duplikasi variabel lokal. Jika rekursif dilakukan terlalu panjang dapat terjadi error StackOverflowException.
- **Efisiensi**. Rekursi selalu dapat digantikan dengan loop biasa yang biasanya lebih efisien dalam penggunaan RAM.
- Sumber: https://msdn.microsoft.com/en-us/library/81tad23s.aspx