

BAB VI KONTROL ALIR

Semua bahasa pemrograman harus menyediakan satu atau lebih jalan untuk mengeksekusi beberapa pernyataan keluar dari suatu runtutan/rangkaian dimana mereka nampak dalam *listing* program. Terlepas dari panggilan ke prosedur **Sub** atau **Function**, anda dapat mengelompokkan semua pernyataan kontrol alir dasar ke dalam dua grup: pernyataan percabangan dan pernyataan pengulangan.

STRUKTUR PERCABANGAN (BRANCH)

Pernyataan utama percabangan adalah blok *If...Else...Else If...End*. Visual Basic mendukung beberapa cara dalam penulisan pernyataan ini, termasuk versi satu baris dan banyak baris :

```
' Versi satu baris, tanpa argumen Else
If x > 0 Then y = x

' Versi satu baris, dengan argumen Else
If x > 0 Then y = x Else y = 0

' Satu baris, tapi dengan banyak pernyataan yang dipisahkan oleh colon (:)
If x > 0 Then y = x: x = 0 Else y = 0

' Versi banyak baris dari kode di atas (lebih terbaca)
If x > 0 Then
    y = x
    x = 0
Else
    y = 0
End If

' Contoh dari blok If..ElseIf..Else
If x > 0 Then
    y = x
ElseIf x < 0 Then
    y = x * x
Else
    ' x di sini tentunya 0, tidak perlu lagi untuk mengetesnya
    x = -1
End If
```

Beberapa teknik optimisasi tingkat lanjut juga memungkinkan anda untuk mengkombinasikan banyak kondisi menggunakan operator **AND** dan **OR**. Contoh berikut ini menunjukkan bagaimana anda dapat menuliskan kode lebih ringkas dan efisien dengan menulis ulang sebuah ekspresi *Boolean* :

```
' Jika kedua angka adalah nol, anda dapat menggunakan operator OR
' ke bit-bit dan anda masih memiliki nol.
If x = 0 And y = 0 Then ...
If (x Or y) = 0 Then ...

' Jika salah satu nilainya adalah <> 0, anda dapat menggunakan operator OR
' ke bit-bit mereka dan anda tentunya memiliki nilai bukan nol.
If x <> 0 Or y <> 0 Then ...
If (x Or y) Then ...
```

```
' Jika dua bilangan integer memiliki tanda yang berlawanan, penggunaan
' operator XOR ke mereka menghasilkan sebuah hasil yang memiliki tetapan bit
' tanda. (Dengan kata lain, sebuah nilai negatif.)
If (x < 0 And y >= 0) Or (x >= 0 And y < 0) Then ...
If (x Xor y) < 0 Then ...
```

Sumber ambiguitas lain adalah operator **NOT**. Dalam Visual Basic, operator ini mengembalikan nilai *False* hanya jika argumennya bernilai *True* (-1), jadi anda jangan pernah menggunakannya dengan apapun kecuali hasil *Boolean* dari suatu perbandingan atau dengan sebuah variabel *Boolean* :

```
If Not (x = y) Then ... ' Sama sebagaimana x<>y
If Not x Then ...      ' Sama sebagaimana x<>-1, jangan menggunakan x=0
```

Struktur **Select Case** bekerja dengan ekspresi pengujian tunggal yang bisa diakses saat itu yaitu ada di bagian atas struktur. Setelah Visual Basic mengeksekusi ekspresi tersebut, selanjutnya membandingkan hasil ekspresi dengan nilai pada setiap **case** dalam struktur. Jika ditemukan, Visual Basic melanjutkan mengeksekusi blok pernyataan yang dihubungkan dengan **case**.

```
Select Case Jabatan ' Jangan lupa bahwa Jabatan dalam hal ini adalah numerik
    Case 1
        Nama = "Manajer"
        Tunjangan = 5000000
    Case 2
        Nama = "Sekretaris"
        Tunjangan = 3000000
    Case 3
        Nama = "Personalia"
        Tunjangan = 4000000
    Case Else
        Nama = "Karyawan Lain"
        Tunjangan = 1000000
End Select
```

STRUKTUR PENGULANGAN (LOOP)

Struktur pengulangan memberi kesempatan pada anda untuk mengulang proses eksekusi terhadap satu atau beberapa baris kode.

Struktur Do...Loop

Gunakan struktur **Do...loop** untuk mengeksekusi sebuah blok pernyataan sebanyak angka yang anda definisikan di dalamnya. Dalam pernyataan **Do...loop** ini terdapat beberapa variasi pernyataan, tetapi setiap mengevaluasi kondisi angka Visual Basic melanjutkan eksekusi. Seperti dalam **If...Then**, *condition* harus berupa nilai atau ekspresi *False* (nol) atau *True* (selain nol).

```
Dim i As Integer ' Mencetak angka 1 sampai 10
i = 1
Do While i <= 10
    Print i
    i = i + 1
Loop
```

Variasi lain dari pernyataan **Do...loop**, mengeksekusi pernyataan pertama dan selanjutnya mengetes *condition* setelah proses eksekusi pertama. Dalam variasi ini minimal akan mengeksekusi pernyataan sekali, untuk selanjutnya akan tergantung dari hasil pengujian kondisi.

```
Dim i As Integer ' Mencetak angka 1 sampai 10
i = 1
Do
    Print i
    i = i + 1
Loop While i <= 10
```

Variasi lain adalah dengan menggunakan pernyataan berikut :

DO UNTIL <kondisi>
<blok kode program>
LOOP

Artinya :

<blok kode program> akan diulang **sampai** <kondisi> bernilai TRUE. Pengulangan berhenti bila <kondisi> sudah bernilai TRUE.

Sedangkan untuk pernyataan yang sudah kita bahas beserta contohnya :

DO WHILE <kondisi>
<blok kode program>
LOOP

Artinya :

<blok kode program> akan diulang **selama** <kondisi> bernilai TRUE. Pengulangan berhenti bila <kondisi> sudah bernilai FALSE.

Struktur For...Next

Dalam penggunaan struktur pengulangan **Do**, anda tidak mengetahui seberapa banyak anda akan mengulang pernyataan dalam *loop*. Saat anda mengetahui harus mengeksekusi pernyataan sekian kali, akan lebih baik jika anda menggunakan struktur pengulangan **For...Next**.

```
Dim i As Integer ' Mencetak angka 1 sampai 10
i = 1
For i = 1 To 10
    Print i
Next
```

Berikut ini contoh mencetak bilangan bulat genap tanpa menggunakan operator matematika. Yaitu hanya dengan menggunakan argumen tambahan **STEP**.

```
Dim i As Integer ' Mencetak angka genap 2,4,6,8,10
For i = 0 To 10 Step 2
    Print i
Next
```

Sedangkan berikut ini contoh penggunaan argumen **STEP** untuk hitungan mundur atau menurun.

```
Dim i As Integer ' Mencetak angka genap 10,8,6,4,2
For i = 10 To 1 Step -2
    Print i
Next
```

Struktur For Each...Next

Struktur ini mirip dengan struktur pengulangan **For...Next**, tetapi struktur pengulangan ini mengulang sebuah grup pernyataan dalam setiap elemen dalam sekelompok obyek atau dalam sebuah *array* (larik), yang diharapkan mengulang pernyataan sebanyak angka yang dispesifikasikan. Hal ini akan membantu anda jika anda tidak mengetahui seberapa elemen yang akan diulang.

```
Dim larik() As Variant, angka As Variant
larik = Array(1, 21, 4, 41, 3, 25, 5, 75, 53, 24, 5) ' Contoh kumpulan angka dalam array
For Each angka In larik
    Print angka ' Mencetak semua isi dalam array
Next
```

Keluar dari Looping

Anda dapat keluar dari kontrol struktur sebelum proses tersebut selesai dikerjakan. Anda dapat menggunakan perintah **Exit For** dan **Exit Do**. Format kedua argumen ini adalah sebagai berikut :

```
For pencacah = awal To akhir [Step increment]
    [blok pernyataan]
    [Exit For]
    [blok pernyataan]
Next

Do [{While | Until} Condotion]
    [blok pernyataan]
    [Exit Do]
    [blok pernyataan]
Loop
```

Contoh :

```
Dim i As integer ' Mencetak angka 1,2,3,4,5
For i = 1 To 10
    Print i
    If i = 5 Then Exit For ' Jika i=5 maka keluar dari looping for.
Next
```

Latihan 1

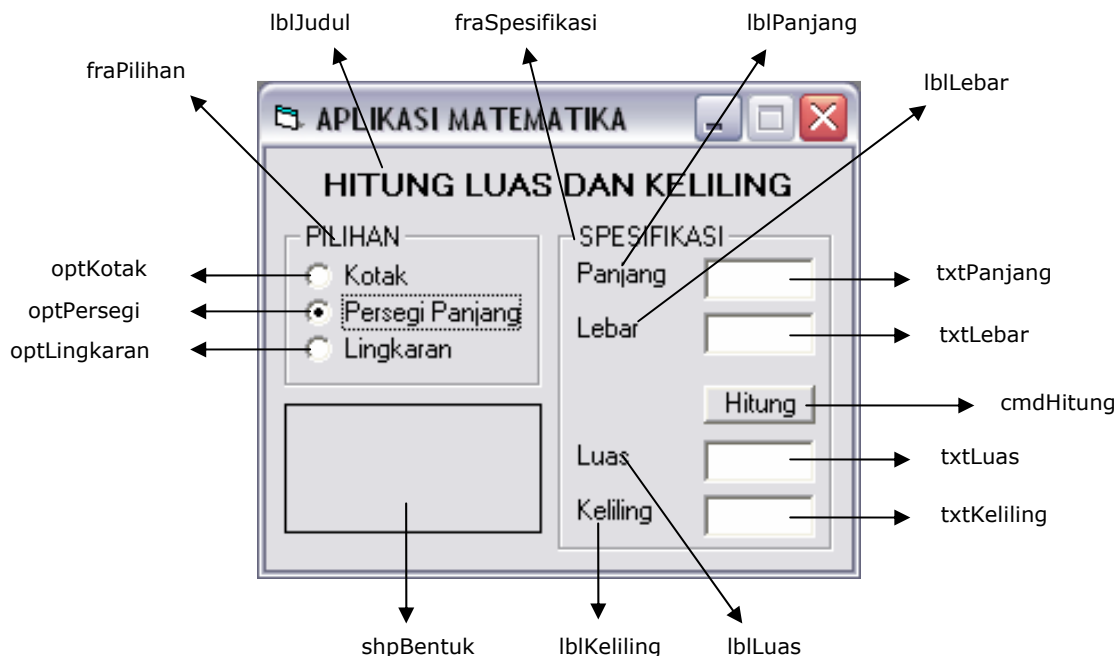
Menghitung luas dan keliling dari kotak, persegi panjang, dan lingkaran dengan menggunakan prosedur dan fungsi.

Komponen :

Komponen	Properti
CommandButton	Name = cmdHitung Caption = Hitung
Dua Frame	Name = fraPilihan ; Caption = PILIHAN Name = fraSpesifikasi ; Caption = SPESIFIKASI
Empat TextBox	1. Name = txtPanjang 2. Name = txtLebar 3. Name = txtLuas 4. Name = txtKeliling
Tiga OptionButton	1. Name = optKotak 2. Name = optPersegi ; Value = True 3. Name = optLingkaran
Lima Label	1. Name = lblJudul 2. Name = lblPanjang 3. Name = lblLebar 4. Name = lblLuas 5. Name = lblKeliling
Shape	Name = shpBentuk ; Shape = 0 – Rectangle

KET: Nama untuk tiap label sesuaikan dengan kebutuhan.

Desain :



Kode :

```

Option Explicit
'*****
' Program Aplikasi Matematika
' Programmer >> Someone. ©Copyright. 2006
' Jurusan Teknik Informatika - UNIKOM
'*****
Dim benda As String

Private Sub cmdHitung_Click()
    Dim p As Double, l As Double

    p = Val(txtPanjang)
    l = Val(txtLebar)

    txtLuas = LUAS(benda, p, l)
    txtKeliling = KELILING(benda, p, l)
End Sub

Private Sub optKotak_Click()
    ' Bentuk Square (kotak)
    shpBentuk.Shape = vbShapeSquare ' Atau shpBentuk.Shape = 1

    ' Menentukan spesifikasi
    benda = "kotak"
    Call Spesifikasi(benda)
End Sub

Private Sub optLingkaran_Click()
    ' Bentuk Circle (lingkaran)
    shpBentuk.Shape = vbShapeCircle ' Atau shpBentuk.Shape = 3

    ' Menentukan spesifikasi
    benda = "lingkaran"
    Call Spesifikasi(benda)
End Sub

Private Sub optPersegi_Click()
    ' Bentuk Rectangle (persegi panjang)
    shpBentuk.Shape = vbShapeRectangle ' Atau shpBentuk.Shape = 0

    ' Menentukan spesifikasi
    benda = "persegi"
    Call Spesifikasi(benda)
End Sub

Private Sub Spesifikasi(Optional benda As String = "persegi")
    If benda = "kotak" Then
        lblPanjang.Caption = "Sisi"
        lblLebar.Visible = False
        txtLebar.Enabled = False
    ElseIf benda = "lingkaran" Then
        lblPanjang.Caption = "Jari-jari"
        lblLebar.Visible = False
        txtLebar.Enabled = False
    Else
        lblPanjang.Caption = "Panjang"
        lblLebar.Visible = True
        txtLebar.Enabled = True
    End If
End Sub

```

```

Private Function LUAS(Optional benda As String = "persegi", _
    Optional p As Double = 0, _
    Optional l As Double = 0) As Double
    If benda = "kotak" Then
        LUAS = p * p ' luas = sisi x sisi
    ElseIf benda = "lingkaran" Then
        LUAS = 3.14 * p * p ' luas = pi x jari x jari
    Else
        LUAS = p * l ' luas = panjang x lebar
    End If
End Function

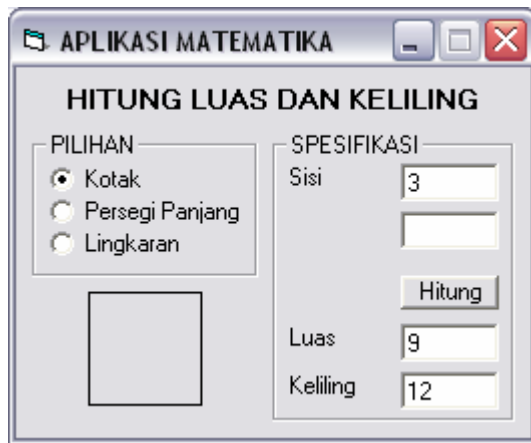
Private Function KELILING(Optional benda As String = "persegi", _
    Optional p As Double = 0, _
    Optional l As Double = 0) As Double
    If benda = "kotak" Then
        KELILING = 4 * p ' keliling = 4 x sisi
    ElseIf benda = "lingkaran" Then
        KELILING = 2 * 3.14 * p ' keliling = 2 x pi x jari
    Else
        KELILING = (2 * p) + (2 * l) ' keliling = (2 x panjang) + (2 x lebar)
    End If
End Function

```

Tampilan.

Pemilihan bentuk persegi panjang dengan lebar dikosongkan (berarti lebar=0).

Pemilihan bentuk lingkaran secara otomatis mengubah bentuk **shape** menjadi sebuah *circle* (lingkaran). Efek lainnya yaitu **lblLebar** tidak ditampilkan dan **txtLebar** di-*disable*-kan. Juga tulisan atau *caption* pada **lblPanjang** berubah menjadi "Jari-jari"



Pemilihan bentuk kotak secara otomatis mengubah bentuk **shape** menjadi sebuah *squre* (kotak). Efek lainnya yaitu **lblLebar** tidak ditampilkan dan **txtLebar** di-*disable*-kan sebagaimana pada lingkaran. Juga tulisan atau *caption* pada **lblPanjang** berubah menjadi "Sisi"

Latihan 2.

Penggunaan **select...case** untuk mengetahui kelompok dari sebuah karakter.

Komponen :

Komponen	Properti
CommandButton	Name = cmdTes Caption = TES
TextBox	Name = txtKarakter
Label	Name = lblTulisan ; Caption = Masukkan sebuah karakter

Desain :



Tampilan :

Contoh bila teks diisi dengan bilangan atau numerik 2. Dan hasil penekanan tombol TES adalah sebuah kotak pesan di sebelah kanannya.



Contoh bila teks diisi dengan bilangan atau numerik Z dan hasil penekanan tombol TES.



Untuk selanjutnya silakan anda mencoba mengisi karakter di teks dengan karakter lain.

Kode :

Option Explicit

' Program Aplikasi Karakter

' Programmer >> SomeOne. ©Copyright. 2006

' Jurusan Teknik Informatika - UNIKOM

Private Sub cmdTes_Click()

Dim kata As String

Select Case txtKarakter

Case "0" To "9"

 kata = "Sebuah bilangan integer"

Case "A" To "Z", "a" To "z"

 kata = "Sebuah huruf abjad"

Case ".", ",", " ", ":", ";", "?"

 kata = "Sebuah tanda baca atau spasi"

Case Else

 kata = "Karakter lainnya"

End Select

MsgBox kata, vbInformation, "Pemberitahuan"

End Sub

Private Sub txtKarakter_GotFocus()

 txtKarakter = vbNullChar

' Bila textbox mendapat fokus maka isi teks akan

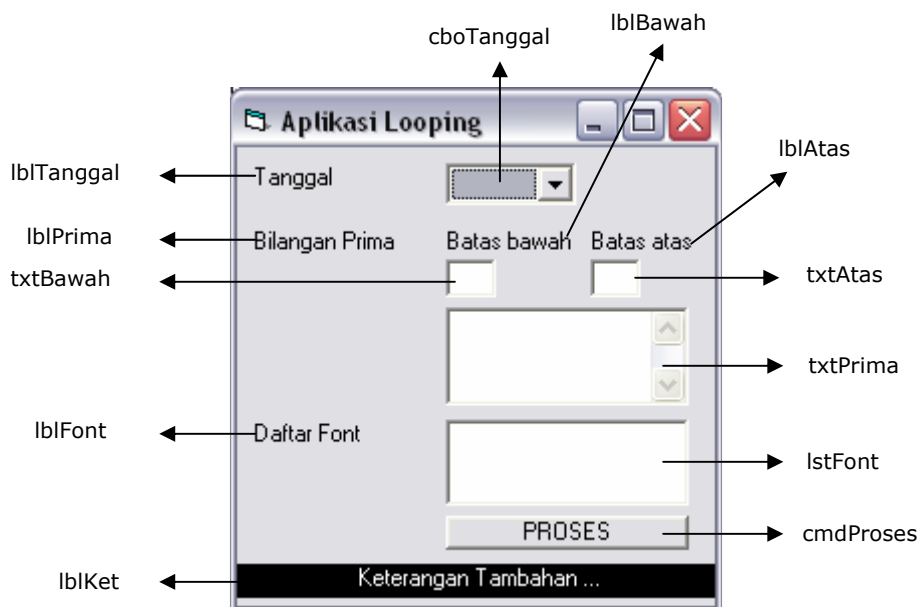
End Sub

' dikosongkan

Latihan 3

1. Membangkitkan tanggal dari 1 sampai 31 ke dalam **ComboBox**.
2. Membangkitkan bilangan prima dari batas bawah dan atas yang telah ditentukan pengguna ke dalam **TextBox**.
3. Mendaftar jenis-jenis font dalam sistem dalam sebuah **ListBox**

Desain :



Komponen :

Komponen	Properti
CommandButton	Name = cmdProses ; Caption = PROSES
Tiga TextBox	1. Name = txtPrima ; Multiline = True ; ScrollBars = 2 - Vertical 2. Name = txtBawah 3. Name = txtAtas
Enam Label	1. Name = lblTanggal ; Caption = Tanggal 2. Name = lblPrima ; Caption = Bilangan Prima 3. Name = lblBawah ; Caption = Batas bawah 4. Name = lblAtas ; Caption = Batas atas 5. Name = lblFont ; Caption = Daftar Font 6. Name = lblKet ; Caption = Keterangan Tambahan ... BackColor = vbBlack ; ForeColor = vbWhite ; Alignment = 2 - Center
ComboBox	Name = cboTanggal ; Style = 2 - Dropdown List
ListBox	Name = lstFont ; Sorted = True

Kode :

Option Explicit

```
*****
```

```
' Program Aplikasi Looping
```

```
' Programmer >> Someone. ©Copyright. 2006
```

```
' Jurusan Teknik Informatika - UNIKOM
```

```
*****
```

```
Dim i As Integer
```

```
Dim hitung As Integer
```

```
Private Sub cmdProses_Click()
```

```
    ' Memasukkan nilai tanggal 1-31 ke ComboBox
```

```
    Call PANGGIL_TANGGAL
```

```
    ' Memproses pemanggilan bilangan prima
```

```
    txtPrima = vbNullString ' Kosongkan teks
```

```
    Call PRIMA(Val(txtBawah), Val(txtAtas))
```

```
    ' Memproses pemanggilan font yang ada dalam sistem
```

```
    lstFont.Clear
```

```
    Call DAFTAR_FONT
```

```
    ' Memanggil jumlah bil. prima dan jumlah font
```

```
    lblKet.Caption = "Jumlah bil. prima = " & hitung & " dan Jumlah Font = " & Screen.FontCount
```

```
End Sub
```

```
Private Sub PANGGIL_TANGGAL()
```

```
    Do Until i >= 31
```

```
        i = i + 1
```

```
        cboTanggal.AddItem i
```

```
    Loop
```

```
    cboTanggal.ListIndex = 0
```

```
End Sub
```

```
Private Sub PRIMA(Optional bawah As Integer = 2, _
```

```
    Optional atas As Integer = 100)
```

```
    hitung = 0
```

```
    For i = bawah To atas
```

```
        If i = 2 Then txtPrima = 2: hitung = 1
```

```
        If i = 3 Then txtPrima = txtPrima & " " & 3: hitung = hitung + 1
```

```
        If i Mod 2 <> 0 And i Mod 3 <> 0 Then
```

```
            txtPrima = txtPrima & " " & i
```

```
            hitung = hitung + 1
```

```
        End If
```

```
    Next
```

```
End Sub
```

```
Private Sub DAFTAR_FONT()
```

```
    For i = 0 To Screen.FontCount
```

```
        lstFont.AddItem Screen.Fonts(i) ' Memasukkan setiap font ke listbox
```

```
    Next
```

```
    lstFont.ListIndex = 1 ' Menampilkan font pada indeks ke-1
```

```
End Sub
```

Tampilan :

Daftar tanggal dari 1-31



Bilangan prima dari 1-100



Bilangan prima dari 5-20

