



STATEMEN KONTROL

- ❑ Fitri Ayuning Tyas, S. Kom
- ❑ yas.0373@gmail.com
- ❑ 081804767700
- ❑ tyas-tamimy.com

Statemen Kontrol



- ❑ Statemen kontrol disediakan oleh Visual Basic .NET yang berfungsi untuk mengontrol alur program.
- ❑ Melalui statemen kontrol, jalannya program dapat diatur sesuai dengan kebutuhan sehingga menghasilkan *output* yang benar.

Statemen Pemilihan

- `If...Then...Else`
- `Select...Case`
- `Choose ()`
- `Switch ()`

Statemen Pengulangan

- `While...End While`
- `Do...Loop`
- `For...Next`
- `For Each...Next`

Statemen Loncat

- `Exit`
- `Continue`
- `GoTo`
- `End`
- `Return`
- `Stop`
- `OnError`
- `Throw`

Statemen Pengulangan

Pengulangan/ *Looping*



- ❑ *Loop* merupakan salah satu struktur penting yang digunakan untuk mengulangi eksekusi pernyataan berulang kali. Selama tiap repetisi/perulangan, pernyataan-pernyataan tersebut berperan dalam mengubah nilai-nilai variabel.
- ❑ Dalam menangani proses pengulangan statemen, VB .NET memiliki beberapa jenis kontruksi. Masing-masing kontruksi memiliki karakteristik yang berbeda dan dirancang untuk menangani permasalahan yang berbeda pula.

Statemen While...End While



- ❑ Kontruksi ini digunakan untuk melakukan eksekusi statemen secara berulang selama kondisi bernilai `True`.
- ❑ Harus ada aksi yang menyebabkan kondisi bernilai `False`, agar pengulangan berhenti.
- ❑ Cocok diterapkan untuk kasus dimana kita belum tahu secara pasti berapa kali proses pengulangan akan dilakukan.
- ❑ Bentuk umum:

```
While (kondisi)  
    Statemen  
End While
```

Contoh Program Statemen While...End While



Module PengulanganWhile

```
Sub Main()  
  Dim a As Integer  
  a = 0  
  While (a < 5)  
    Console.Write("Langkah ke -{0}: ", a)  
    Console.WriteLine("Tyas Tamimy")  
    a = a + 1  
    Console.ReadLine()  
  End While  
End Sub  
  
End Module
```

□ Hasil:

```
Langkah ke -0: Tyas Tamimy  
Langkah ke -1: Tyas Tamimy  
Langkah ke -2: Tyas Tamimy  
Langkah ke -3: Tyas Tamimy  
Langkah ke -4: Tyas Tamimy
```

Contoh Program Statemen While...End While



Module Faktorial

```
Sub Main()
    Dim n, i As Integer
    Dim faktorial As Long = 1
    Dim s As String

    Console.WriteLine("Masukkan bilangan bulat positif: ")
    s = Console.ReadLine()
    n = Int32.Parse(s)

    Console.WriteLine("{0}! = ", n)

    If (n > 1) Then
        i = n
        While (i >= 1)
            If (i <> 1) Then
                Console.WriteLine("{0} x ", i)
            Else
                Console.WriteLine("{0} = ", i)
            End If
            faktorial = faktorial * i
            i -= 1
        End While
    End If

    Console.WriteLine(faktorial)

    Console.ReadLine()
End Sub
End Module
```

□ Hasil:

```
Masukkan bilangan bulat positif: 7
7! = 7 x 6 x 5 x 4 x 3 x 2 x 1 = 5040
```

Statemen Do . . . Loop



□ Bentuk umum:

```
Do{While} (kondisi)
    Statemen
Loop
```

```
Do{Until} (kondisi)
    Statemen
Loop
```

```
Do
    Statemen
Loop{While} (kondisi)
```

```
Do
    Statemen
Loop{Until} (kondisi)
```

- Cara kerja kata kunci `Until` (sampai), proses pengulangan akan terus dilakukan selama kondisi bernilai `False`. Ketika kondisi bernilai `True`, pengulangan dihentikan. (Kebalikan dari `While`)

Contoh Program Statemen Do . . . Loop



Module DoLoop

```
Sub Main()  
    Dim i As Integer  
  
    Console.WriteLine("Do While...Loop:")  
    i = 1  
  
    'pengulangan akan terus dilakukan  
    'selama nilai i lebih kecil atau sama dengan 5  
    Do While (i <= 5)  
        Console.Write("{0} ", i * 100)  
        i += 1  
    Loop  
  
    Console.WriteLine(Chr(10))  
  
    Console.WriteLine("Do Until...Loop:")  
    i = 1  
  
    'pengulangan akan terus dilakukan  
    'selama nilai i lebih besar dari 5  
    Do Until (i > 5)  
        Console.Write("{0} ", i * 100)  
        i += 1  
    Loop  
  
    Console.ReadLine()  
End Sub  
End Module
```

□ Hasil:

```
Do While...Loop:  
100 200 300 400 500  
  
Do Until...Loop:  
100 200 300 400 500 _
```

Contoh Program Statemen Do . . . Loop



Module Login

```
Sub Main()  
  Const CUSERNAME As String = "Tyas"  
  Const CPASSWORD As String = ".net54321"  
  
  Dim username, password As String  
  Dim ok As Boolean = False  
  
  Console.WriteLine("LOGIN")  
  Do  
    Console.Write("Username: ")  
    username = Console.ReadLine()  
    Console.Write("Password: ")  
    password = Console.ReadLine()  
    Console.WriteLine()  
    ok = (username = CUSERNAME) AndAlso (password = CPASSWORD)  
  Loop While (Not ok)  
  
  Console.WriteLine("Login sukses...")  
  
  Console.ReadLine()  
End Sub  
End Module
```

□ Hasil:

```
LOGIN  
Username: Tya  
Password: 123  
  
Username: Tyas  
Password: .net  
  
Username: Tyas.net4321  
Password: 1  
  
Username: Tyas  
Password: .net4321  
  
Username: Tyas  
Password: .net54321  
  
Login sukses...
```

Statemen For...Next



- ❑ Statemen For...Next digunakan untuk melakukan pengulangan yang jumlahnya sudah diketahui dengan pasti.
- ❑ Bentuk umum:

```
For indeks [As TipeData]=NilaiAwal To NilaiAkhir [Step stepke-]  
    Statemen  
Next[indeks]
```

- ❑ Contoh:

```
For i As Integer = 1 To 10 Step 1  
    Console.write("{0}", i)  
Next i
```

```
For i = 1 To 10 Step 1  
    Console.write("{0}", i)  
Next i
```

Contoh Program Statemen For . . . Next



Module ProgramFor

```
Sub Main()  
    Console.WriteLine("Pengulangan ke-1:")  
    For i As Integer = 1 To 10 Step 1  
        Console.Write("{0} ", i)  
    Next i  
  
    Console.WriteLine(Chr(10))  
  
    Console.WriteLine("Pengulangan ke-2:")  
    For i As Integer = 1 To 10  
        Console.Write("{0} ", i)  
    Next i  
  
    Console.WriteLine(Chr(10))  
  
    Console.WriteLine("Pengulangan ke-3:")  
    For i = 1 To 10  
        Console.Write("{0} ", i)  
    Next  
  
    Console.WriteLine(Chr(10))  
  
    Console.WriteLine("Pengulangan ke-4:")  
    For i = 1 To 10 Step 2  
        Console.Write("{0} ", i)  
    Next  
  
    Console.ReadLine()  
End Sub  
End Module
```

□ Hasil:

```
Pengulangan ke-1:  
1 2 3 4 5 6 7 8 9 10  
  
Pengulangan ke-2:  
1 2 3 4 5 6 7 8 9 10  
  
Pengulangan ke-3:  
1 2 3 4 5 6 7 8 9 10  
  
Pengulangan ke-4:  
1 3 5 7 9 _
```

Statemen For Each...Next



- ❑ Statemen pengulangan yang khusus digunakan untuk mengakses setiap elemen yang terdapat di dalam sekumpulan nilai tertentu (`array` maupun koleksi)
- ❑ Proses pengulangan dilakukan dengan cara menelusuri setiap elemen yang terdapat di dalam koleksi, mulai dari elemen pertama sampai elemen terakhir. Ketika elemen sudah tidak ada lagi, pengulangan dihentikan.
- ❑ Bentuk umum:

```
For Each elemen[As TipeData] In Koleksi  
    Statemen  
Next [elemen]
```

Contoh Program Statemen For Each...Next



Module ProgramNilaiMaksimum

```
Sub Main()
```

```
'array dengan 10 elemen bertipe Integer
```

```
Dim data() As Integer = {8, 21, 7, 19, 6, 50, 9, 10, 17, 45}
```

```
Dim maks As Integer
```

```
maks = data(0) 'mula-mula elemen pertama dianggap maksimum
```

```
'mengakses seluruh elemen array
```

```
Console.WriteLine("Data: ")
```

```
For Each elemen As Integer In data
```

```
    Console.WriteLine("{0} ", elemen)
```

```
    If elemen > maks Then maks = elemen
```

```
Next
```

```
Console.WriteLine(Chr(10))
```

```
Console.WriteLine("Nilai maksimum " & "adalah: {0}", maks)
```

```
Console.ReadLine()
```

```
End Sub
```

```
End Module
```

□ Hasil:

```
Data: 8 21 7 19 6 50 9 10 17 45
Nilai maksimum adalah: 50
```

Pengulangan Bersarang



- ❑ Nested Loop adalah blok pengulangan yang terdapat di dalam blok pengulangan lain.
- ❑ Anda dapat menggunakan statemen While...End While, Do...Loop, For...Next, For Each...Next, maupun kombinasi dari jua jenis statemen tersebut.

Contoh Program Pengulangan Bersarang



Module ProgramPengulanganBersarang

```
Sub Main()
    'pengulangan bagian luar
    For i = 1 To 10
        'pengulangan bagian dalam
        For j = 1 To i
            Console.WriteLine("{0} ", i * j)
        Next
        'baris baru di setiap nilai i
        Console.WriteLine()
    Next

    Console.ReadLine()
End Sub
End Module
```

Hasil:

```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49
8 16 24 32 40 48 56 64
9 18 27 36 45 54 63 72 81
10 20 30 40 50 60 70 80 90 100
```