

## Polimorfisme (Polymorphism): Satu Nama, Banyak Bentuk

Selamat datang di materi terakhir dari empat pilar PBO: **Polimorfisme**. Kata ini berasal dari bahasa Yunani, "poly" yang berarti banyak, dan "morph" yang berarti bentuk. Jadi, **Polimorfisme** adalah kemampuan sebuah objek untuk memiliki banyak bentuk atau, dalam pemrograman, kemampuan untuk memiliki satu nama yang bisa digunakan untuk banyak tujuan.

Ada dua jenis utama dari polimorfisme yang akan kita pelajari: *Method Overriding* dan *Method Overloading*.

---

### 1. Method Overriding

*Method overriding* adalah ketika sebuah kelas turunan (*derived class*) mendefinisikan ulang atau memodifikasi perilaku dari sebuah metode yang diwarisi dari kelas dasar (*base class*). Metode tersebut harus memiliki nama, jumlah parameter, dan tipe data yang sama persis.

Contoh Sehari-hari:

Bayangkan kelas dasar Hewan memiliki metode Bersuara(). Secara umum, kita bisa mendefinisikan "suara" hewan. Namun, ketika kita membuat kelas turunan Kucing dan Anjing, suara mereka berbeda. Kucing bersuara "Meong", dan Anjing bersuara "Guk". Kita perlu "menimpa" (override) metode Bersuara() di setiap kelas turunan agar perilakunya sesuai dengan jenis hewannya.

Struktur Kode:

Untuk memungkinkan overriding, kita harus mendeklarasikan metode di kelas dasar sebagai Overridable. Kemudian, di kelas turunan, kita gunakan kata kunci Overrides untuk menimpa metode tersebut.

**Kode Kelas Dasar Hewan:**

```
VB.Net
```

```
Public Class Hewan
```

```
    ' Metode ini bisa ditimpa (override) oleh kelas turunan
```

```
Public Overridable Sub Bersuara()  
    MsgBox("Hewan ini bersuara...")  
End Sub  
End Class
```

### Kode Kelas Turunan Kucing:

VB.Net

```
Public Class Kucing  
    Inherits Hewan  
  
    ' Menimpa metode Bersuara() dari kelas dasar  
    Public Overrides Sub Bersuara()  
        MsgBox("Meow!")  
    End Sub  
End Class
```

### Kode Kelas Turunan Anjing:

VB.Net

```
Public Class Anjing  
    Inherits Hewan  
  
    ' Menimpa metode Bersuara() dari kelas dasar  
    Public Overrides Sub Bersuara()  
        MsgBox("Guk Guk!")  
    End Sub  
End Class
```

Dengan cara ini, saat kita memanggil metode Bersuara() dari objek Kucing, hasilnya adalah "Meow!". Saat memanggilnya dari objek Anjing, hasilnya adalah "Guk Guk!". Ini adalah **polimorfisme**, satu nama (Bersuara()) menghasilkan banyak bentuk perilaku yang berbeda.

---

## 2. Method Overloading

*Method overloading* adalah kemampuan untuk membuat beberapa metode dengan nama yang sama di dalam satu kelas, tetapi memiliki jumlah atau tipe data parameter yang berbeda.

Contoh Sehari-hari:

Bayangkan kita punya sebuah kalkulator. Tombol tambah (+) bisa digunakan untuk menjumlahkan dua bilangan bulat, tetapi juga bisa menjumlahkan dua bilangan desimal. Ini adalah overloading.

Struktur Kode:

Visual Basic .NET secara otomatis mendukung overloading. Kita hanya perlu mendeklarasikan metode dengan nama yang sama tetapi dengan tanda tangan (signature) yang berbeda (jumlah atau tipe data parameternya).

**Contoh Kode Overloading:**

VB.Net

```
Public Class Kalkulator
```

```
    ' Overloading 1: Menjumlahkan dua Integer
```

```
    Public Function Tambah(ByVal a As Integer, ByVal b As Integer) As Integer
```

```
        Return a + b
```

```
    End Function
```

```
    ' Overloading 2: Menjumlahkan tiga Integer
```

```
    Public Function Tambah(ByVal a As Integer, ByVal b As Integer, ByVal c As Integer) As Integer
```

```
        Return a + b + c
```

```
    End Function
```

```
    ' Overloading 3: Menjumlahkan dua Double
```

```
    Public Function Tambah(ByVal a As Double, ByVal b As Double) As Double
```

```
        Return a + b
```

```
    End Function
```

```
End Class
```

Ketika kita memanggil metode `Tambah()`, Visual Basic .NET akan secara otomatis memilih metode mana yang akan digunakan berdasarkan parameter yang kita berikan.

## Cara Menggunakannya:

VB.Net

```
Dim calc As New Kalkulator()
```

```
' Memanggil overloading 1  
MsgBox(calc.Tambah(5, 10)) ' Output: 15
```

```
' Memanggil overloading 2  
MsgBox(calc.Tambah(5, 10, 15)) ' Output: 30
```

```
' Memanggil overloading 3  
MsgBox(calc.Tambah(3.5, 4.5)) ' Output: 8
```

Meskipun nama metodenya sama (Tambah), program bisa melakukan operasi yang berbeda. Ini membuat kode kita lebih fleksibel dan mudah dibaca.

## Perbedaan Utama: *Overriding* vs. *Overloading*

Fitur	<i>Method Overriding</i>	<i>Method Overloading</i>
<b>Lokasi</b>	Kelas Dasar dan Kelas Turunan	Satu Kelas yang Sama
<b>Tujuan</b>	Mengubah perilaku metode yang diwarisi	Membuat metode yang sama untuk parameter berbeda
<b>Nama Metode</b>	Harus sama	Harus sama
<b>Parameter</b>	Harus sama	Harus berbeda (jumlah atau tipe)

<b>Kata Kunci</b>	Overridable dan Overrides	Tidak ada kata kunci khusus
-------------------	---------------------------	-----------------------------