

## Pemrograman Berorientasi Objek (Object Oriented Programming – OOP)

### 1. Polimorphism (Banyak Bentuk)

Sub atau function didalam VB.Net dapat memiliki banyak bentuk (Namanya sama tetapi parameternya berbeda dengan tipe data yang berbeda)

Contoh :

```
Public Class ClsPemrosesan
```

```
    Public Function Hitung(ByVal Harga As Double, ByVal Jumlah As Integer) As Double  
        Return Harga * Jumlah  
    End Function
```

```
    Public Function Hitung(ByVal Harga As Double, ByVal Jumlah As Integer, ByVal  
        Diskon As Double) As Double  
        Return Harga * Jumlah - Diskon  
    End Function
```

```
End Class
```

**Ket** : Dalam OOP baik nama Sub atau Nama fungsi (Function) bisa saja sama tetapi parameter inputannya harus berbeda baik jumlah maupun tipe datanya, dimana pada function Hitung yang pertama membutuhkan 2 parameter (Harga dan Jumlah) sedangkan function hitung yang kedua membutuhkan 3 parameter (Harga, Jumlah dan Diskon). Selain perbedaan jumlah parameter untuk function yang sama dapat juga dibedakan hanya dengan membedakan tipe datanya, misalnya jika di Hitung yang pertama Harga As Double maka di function Hitung yang kedua mungkin dapat dibuat Harga As Integer untuk membedakannya.

### 2. Overloading

Polimorphism di Class yang berbeda.

Contoh :

```
Public Class Class1
```

```
    Public Function Hitung(ByVal Harga As Double, ByVal Jumlah As Integer) As Double  
        Return Harga * Jumlah  
    End Function
```

```
End Class
```

```
Public Class Class1a
```

```
    Inherits Class1
```

```
    Public Overloads Function Hitung(ByVal Harga As Integer) As Integer  
        Return Harga  
    End Function
```

```
    Public Sub Test1()
```

```
        Hitung(50000)
```

```
    End Sub
```

```
End Class
```

**Ket** : Function Hitung di atas berada di Class1, ketika ada Class1a yang mewarisi (inherits Class1) semua yang dimiliki oleh class1 (baik function, sub atau property), maka sebenarnya Class1a sudah memiliki function Hitung, jikalau ingin menambah function hitung yang baru maka harus ditambahkan kata Overloads pada saat deklarasinya ( Public Overloads Function Hitung(...) As Integer ).

### 3. Overrides

Yaitu menimpa Sub atau Function yang lama dengan bentuk Sub atau Function yang baru, tetapi Sub atau Function yang lama hanya dapat ditimpa jika diberi kata kunci 'Overridable' pada saat pembuatan pertama kalinya, lalu untuk Sub atau Function yang baru jika ingin menimpa Sub atau Function yang lama harus menggunakan kata kunci 'Overrides'.

Contoh :

```
Public Class Class1
```

```
    Public Overridable Function HitungGaji(ByVal x As Integer, ByVal y As Double) As Double
        Return x * y
    End Function
```

```
End Class
```

```
Public Class Class1a
```

```
    Inherits Class1
```

```
    Public Overrides Function HitungGaji(ByVal x As Integer, ByVal y As Double) As Double
```

```
        Return MyBase.HitungGaji(x, y) + 10000
    End Function
```

```
End Class
```

**Ket:** Pada Class1a mendapat warisan semua yang dimiliki oleh Class1 hal ini karena menggunakan kata 'Inherits Class1', maka sebenarnya Class1a telah memiliki Function Hitung, ketika Class1a ingin membuat Function Hitung yang baru yang akan menimpa Function Hitung yang lama (hasil dari warisan Class1) maka wajib menggunakan kata 'Overrides' dan inipun bisa ditimpa jika Function Hitung yang lama dideklarasikan dengan kata 'Overridable' pada saat pertama kali dibuat.