



## KELAS DAN OBJEK

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

# Pendahuluan



- ❑ Visual Basic .NET merupakan bahasa yang mendukung pemrograman berorientasi objek.
- ❑ Kita dapat merumuskan suatu permasalahan pemrograman ke dalam sebuah model abstrak yang disebut kelas (*Class*).
- ❑ Kelas dapat didefinisikan sebagai kumpulan data dan prosedur/fungsi (metode) yang menggambarkan suatu objek tertentu.

# Konsep Dasar Pemrograman Berorientasi Objek



- ❑ Disingkat OOP (*Object Oriented Programming*) bukan bahasa pemrograman melainkan suatu paradigma pemrograman yang berorientasikan objek atau *object-oriented* (berfikir dan memandang suatu permasalahan sebagai objek).
- ❑ Contoh, kita akan membuat program menghitung Luas Segitiga dan Lingkaran. Dalam OOP kita perlu berfikir membuat *object* Segitiga dan Lingkaran dengan memodelkan *object* tersebut ke dalam tipe data bentukan terlebih dahulu. Tipe data bentukan seperti inilah yang disebut Kelas.
- ❑ Konsep dasar pada OOP: Pembungkusan (*Encapsulation*), Pewarisan (*Inheritance*) dan Polimorfisme (*Polymorphism*)

# Perbedaan Class dan Object



- ❑ **Class**: konsep dan deskripsi dari sesuatu

Class mendeklarasikan method yang dapat digunakan (dipanggil) oleh object. **Class** merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object

- ❑ **Object**: instance dari class, bentuk (contoh) nyata dari class

Object memiliki sifat independen dan dapat digunakan untuk memanggil method

# Perbedaan Class dan Object



**Object: Manusia**

## **Behavior:**

- Berfikir
- Berbicara
- Berjalan
- dsb



## **Attribute:**

- Nama
- Tanggal lahir
- Jenis kelamin
- Berat badan
- Tinggi Badan
- dsb

# Perbedaan Class dan Object



**Object:** Mobil

## **Behavior:**

- Maju
- Mundur
- Berhenti
- dsb



## **Attribute:**

- Merk
- Tahun Produksi
- Type
- Warna
- dsb

# Encapsulation

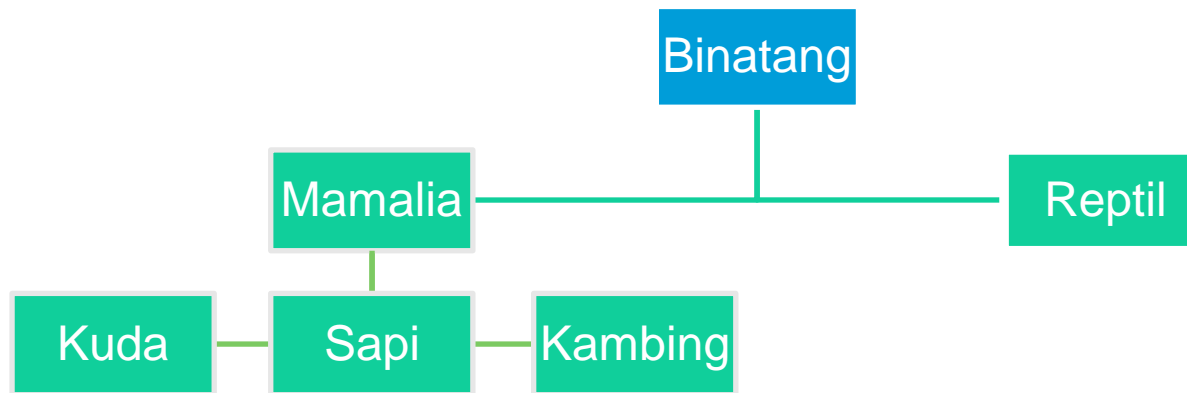


- ❑ *Encapsulation* pada OOP adalah proses membungkus data dan kemampuan dari suatu objek ke dalam suatu kelas.
- ❑ Atau dapat diartikan sebagai bungkus atau pelindung data yang sedang diolah, pembungkus ini mendefinisikan perilaku dan melindungi program dan data yang sedang diolah agar tidak diakses oleh objek lain.
- ❑ Dalam melakukan proses *encapsulation* terdapat tiga tingkat akses yaitu:
  1. `Private`. Digunakan untuk menjadikan anggota kelas (data atau metode) tidak dapat diakses oleh kelas lain
  2. `Protected`. Digunakan untuk menjadikan anggota kelas hanya boleh diakses oleh kelas itu sendiri dan kelas-kelas lain yang merupakan turunan dari kelas bersangkutan
  3. `Public`. Digunakan untuk menjadikan anggota kelas bebas diakses oleh kelas lain, meskipun kelas lain tersebut bukan merupakan kelas turunan dari kelas bersangkutan

# Inheritance



- ❑ *Inheritance* atau pewarisan adalah proses pembuatan kelas baru yang diturunkan dari kelas yang sudah ada.
- ❑ Hasilnya adalah kelas baru akan mewarisi sifat yang dimiliki oleh kelas induknya.



# Polymorphism



- ❑ *Polymorphism* dapat diartikan sebagai banyak bentuk (banyak rupa) atau suatu objek yang memiliki banyak bentuk yang berbeda. Dalam OOP *Polymorphism* mengizinkan kita untuk membuat satu metode (fungsi atau prosedur) yang dapat memberikan hasil yang berbeda. Satu cara namun dapat memberikan banyak hasil. Atau kemampuan suatu variabel referensi objek untuk memiliki aksi yang berbeda jika method yang sama dipanggil.

# Contoh Program



Lanjutan

```
'mendefinisikan kelas Segiempat
```

```
Class Segiempat
```

```
Public panjang As Double
```

```
Public lebar As Double
```

```
End Class
```

```
'program utama
```

```
Module Program
```

```
Sub Main()
```

```
'mendeklarasikan variabel lokal
```

```
Dim luas As Double
```

```
'mendeklarasikan referensi ke kelas Segiempat
```

```
Dim se As Segiempat
```

```
'membuat objek dari kelas Segiempat
```

```
se = New Segiempat()
```

```
'mengisi nilai objek
```

```
se.panjang = 8
```

```
se.lebar = 6
```

```
'menghitung luas
```

```
luas = se.panjang * se.lebar
```

```
'menampilkan luas
```

```
Console.WriteLine("Luas segiempat = {0}", luas)
```

```
Console.ReadLine()
```

```
End Sub
```

```
End Module
```

# Contoh Program



Lanjutan

```
'mendefinisikan kelas Segiempat
Class Segiempat
  Public panjang As Double
  Public lebar As Double
End Class

'program utama
Module Program
  Sub Main()
    'mendeklarasikan variabel lokal
    Dim luas1, luas2 As Double

    'mendeklarasikan dua referensi ke kelas Segiempat
    Dim se1, se2 As Segiempat

    'membuat objek dari kelas Segiempat
    se1 = New Segiempat()
    se2 = New Segiempat()

    'mengisi nilai objek pertama
    se1.panjang = 8
    se1.lebar = 6
```

```
'mengisi nilai objek kedua
se2.panjang = 10
se2.lebar = 5

'menghitung luas
luas1 = se1.panjang * se1.lebar
luas2 = se2.panjang * se2.lebar

'menampilkan luas
Console.WriteLine("Luas objek pertama {0}= {1}", _
  Chr(9), luas1)
Console.WriteLine("Luas objek kedua {0}= {1}", _
  Chr(9), luas2)

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