

Pada praktikum kali ini kita akan melanjutkan pembahasan tentang salah satu konsep Object Oriented Programming (OOP) yaitu: Polymorphism

Polymorphism

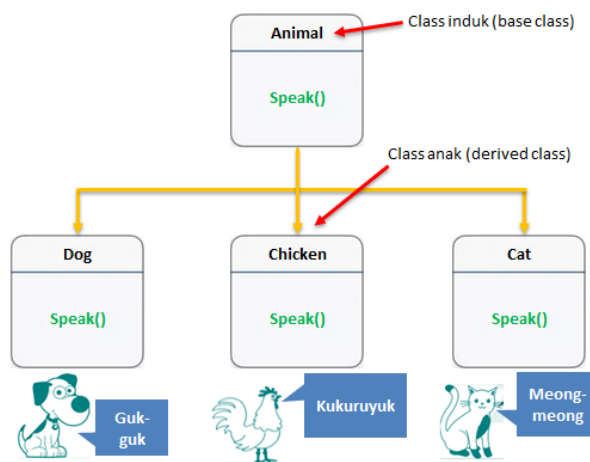
Ada dua aspek yang paling penting pada inheritance yang pertama adalah code reuse atau penggunaan kembali kode yang sudah pernah ditulis, jadi programmer dapat mengembangkan class baru dari class lama yang sudah pernah dibuat, tidak perlu membuat dari awal lagi dengan menuliskan kode yang sama

Aspek kedua yang tidak kalah penting dari inheritance adalah polymorphism, poly berarti banyak dan morph berarti bentuk. Jadi polymorphism adalah kemampuan sebuah objek untuk mengimplementasikan sesuatu hal yang berbeda dengan cara sama.

Dari istilahnya Polymorphism kesannya sulit dan kompleks, tetapi sebenarnya simple, ilustrasinya begini ketika sebuah perusahaan provide telpon mengirimkan signal ke anda (dering telpon), mereka tidak tahu telpon apa yang anda gunakan, bisa model lama dengan dering kuno, bisa HP dengan dering music mp3, TV Phone, PDA Phone dll, yang provider tahu bahwa masing-masing telepon mempunyai 'base type' telepon dan mempunyai method untuk berdering. Ketika provider mengirimkan signal mereka hanya berharap bahwa telepon akan berdering tidak peduli jenis telponnya, nah provider telepon memperlakukan telepon anda secara polymorphism.

Latihan 1 Penggunaan Inheritance (pewarisan)

Ada dua cara untuk mengimplementasikan konsep polymorphism yang pertama yaitu menggunakan *inheritance* (pewarisan) seperti yang sudah kita pelajari di pertemuan sebelumnya dan yang kedua menggunakan *interface* (akan di bahas di pertemuan-pertemuan berikutnya). Untuk latihan yang pertama ini kita masih menggunakan *inheritance* (pewarisan) dengan contoh kasus hubungan antar class Hewan dan turunannya.



1. Buat project Console Application baru dengan nama 'LatihanPolymorphism'.
2. Tambahkan class dengan nama 'Animal.cs' kedalam project. Kemudian tulis kode berikut:

```
1 public class Animal
2 {
3     public string Name { get; set; }
4
5     public void Speak()
6     {
7         Console.WriteLine("The animal speaks");
8     }
9 }
```

3. Tambahkan class baru dengan nama 'Cat.cs' yang diturunkan dari class Animal.

```
1 public class Cat : Animal
2 {
3
4 }
```

4. Tambahkan juga class 'Dog.cs' yang diturunkan dari class Animal.

```
1 public class Dog : Animal
2 {
3
4 }
```

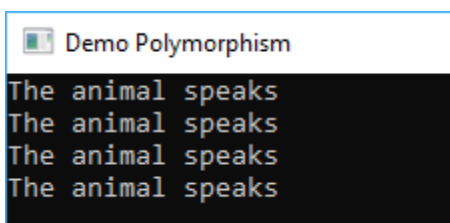
5. Terakhir, tambahkan juga class 'Chicken.cs' yang juga diturunkandari class Animal.

```
1 public class Chicken : Animal
2 {
3
4 }
```

6. Kemudian pada Program.cs lengkapi kode untuk method Main:

```
1 static void Main(string[] args)
2 {
3     Animal animal = new Animal();
4     animal.Speak();
5
6     Cat cat = new Cat();
7     cat.Speak();
8
9     Chicken chicken = new Chicken();
10    chicken.Speak();
11
12    Dog dog = new Dog();
13    dog.Speak();
14
15    Console.ReadKey();
16 }
```

7. Jalankan programnya dengan menekan tombol F5, maka akan ditampilkan hasil:



```
Demo Polymorphism
The animal speaks
The animal speaks
The animal speaks
The animal speaks
```

Dapat dilihat pada program diatas bahwa class Cat, Chicken, dan class Dog yang merupakan turunan dari class Animal dapat mengakses method yang mempunyai access modifier public atau protected pada class Animal. Class Cat, Chicken, dan class Dog juga bisa mempunyai property dan method yang lebih spesifik dari Animal.

Latihan 2 Menambahkan Keyword Virtual dan Override

1. Buka kembali class 'Animal.cs', kemudian tambahkan kode seperti berikut:

```
1 public class Animal
2 {
3     public string Name { get; set; }
4
5     public virtual void Speak()
6     {
7         Console.WriteLine("The animal speaks");
8     }
9 }
```

Pada kode di atas kita menambahkan keyword *virtual* di method `Speak()`. Keyword ini digunakan untuk memberi tanda bahwa method pada base class (class induk) bisa di override (timpa) oleh sub class-nya (class anak).

2. Kemudian edit class 'Cat.cs', 'Dog.cs', dan 'Chicken.cs' seperti kode berikut:

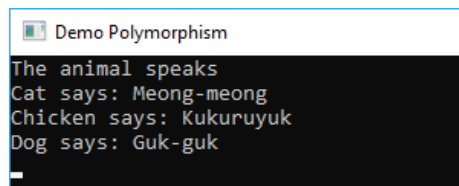
```
public class Cat : Animal
{
    public override void Speak()
    {
        Console.WriteLine("Cat says: Meong-meong");
    }
}

public class Dog : Animal
{
    public override void Speak()
    {
        Console.WriteLine("Dog says: Guk-guk");
    }
}

public class Chicken : Animal
{
    public override void Speak()
    {
        Console.WriteLine("Chicken says: Kukuruyuk");
    }
}
```

Pada kode di atas kita mengoverride (menimpa) method `Speak()` di masing-masing class anak sesuai dengan kebutuhan.

3. Jika sudah, jalankan aplikasi sekali lagi, dan pastikan hasilnya seperti berikut:



```
Demo Polymorphism
The animal speaks
Cat says: Meong-meong
Chicken says: Kukuruyuk
Dog says: Guk-guk
```

Latihan 3 Penggunaan Polymorphism

1. Masih di project yang sama, kemudian edit kode di method `Main` seperti kode berikut:

```
1 static void Main(string[] args)
2 {
3     Animal animal = new Animal();
4     animal.Speak();
5
6     Cat cat = new Cat();
7     animal = cat;
8     //cat.Speak();
9     animal.Speak();
10
11     Chicken chicken = new Chicken();
12     animal = chicken;
13     //chicken.Speak();
14     animal.Speak();
15
16     Dog dog = new Dog();
17     animal = dog;
18     //dog.Speak();
19     animal.Speak();
20
21     Console.ReadKey();
22 }
```

Polymorphism

Polymorphism

Polymorphism

Kita dapat melihat bahwa class 'Cat', 'Chicken' dan 'Dog' dapat di generalisasi tipenya menjadi class 'Animal', karena class 'Cat', 'Chicken' dan 'Dog' merupakan turunan dari class 'Animal'. Ini yang disebut sebagai polymorphism yaitu kemampuan sebuah objek untuk mengimplementasikan sesuatu hal yang berbeda dengan cara sama.

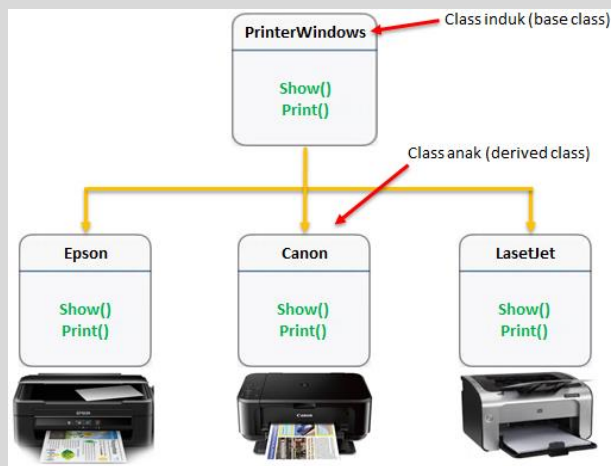
2. Jika sudah, jalankan aplikasi sekali lagi, dan pastikan hasilnya harus sama seperti sebelumnya:

```
Demo Polymorphism
The animal speaks
Cat says: Meong-meong
Chicken says: Kukuruyuk
Dog says: Guk-guk
```

Tugas

Buatkan project lengkap untuk implementasi konsep polymorphism berdasarkan gambar dan kode berikut:

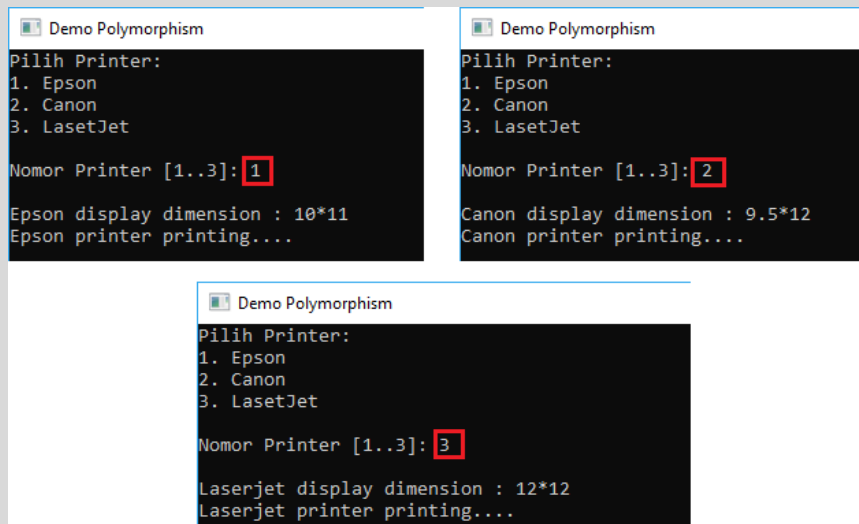
1. Gambar hirarki inheritance (pewarisan) class Printer



2. Implementasi kode di method Main

```
1 static void Main(string[] args)
2 {
3     PrinterWindows printer;
4
5     Console.WriteLine("Pilih Printer:");
6     Console.WriteLine("1. Epson");
7     Console.WriteLine("2. Canon");
8     Console.WriteLine("3. LasetJet\n");
9
10    Console.Write("Nomor Printer [1..3]: ");
11    int nomorPrinter = Convert.ToInt32(Console.ReadLine());
12
13    if (nomorPrinter == 1)
14        printer = new Epson();
15    else if (nomorPrinter == 2)
16        printer = new Canon();
17    else
18        printer = new LaserJet();
19
20    printer.Show();
21    printer.Print();
22
23    Console.ReadKey();
24 }
```

3. Output



The image shows three screenshots of a console application titled "Demo Polymorphism". Each screenshot displays the same menu: "Pilih Printer:", "1. Epson", "2. Canon", and "3. LasetJet".

- The first screenshot shows the user inputting "1" for "Nomor Printer [1..3]:". The output is "Epson display dimension : 10*11" and "Epson printer printing....".
- The second screenshot shows the user inputting "2" for "Nomor Printer [1..3]:". The output is "Canon display dimension : 9.5*12" and "Canon printer printing....".
- The third screenshot shows the user inputting "3" for "Nomor Printer [1..3]:". The output is "Laserjet display dimension : 12*12" and "Laserjet printer printing....".

Selesai ☺

Kamarudin, M.Kom
<http://coding4ever.net/>
<https://github.com/rudi-krsoftware/open-retail>