



MATERI 4 – PEWARISAN

PEMROGRAMAN LANJUT S1 TI

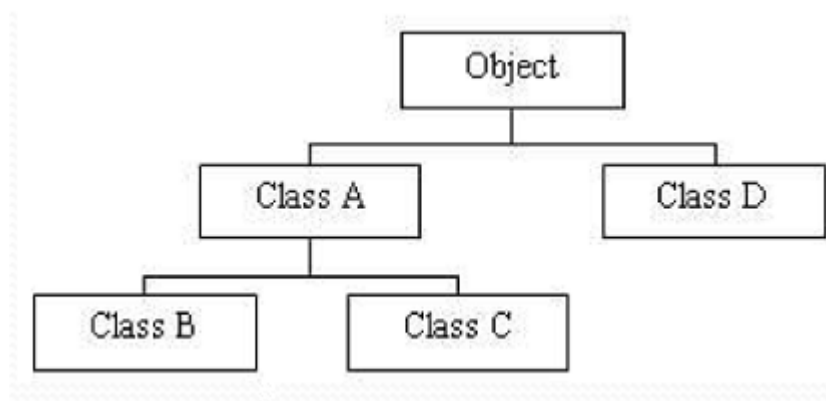
STEVI EMA WIJAYANTI



PEWARISAN (INHERITANCE)

DEFINISI PEWARISAN (INHERITANCE)

Inheritance merupakan proses pewarisan data dan method dari suatu class yang telah ada kepada suatu class baru. Class yang mewariskan disebut dengan **superclass/parent class/base class**, sedangkan class yang mewarisi (class yang baru) disebut dengan **subclass/child class/derived class**. Subclass tidak dapat mewarisi anggota private dari superclass-nya.



Dengan inheritance (pewarisan), class yang baru (subclass) akan mirip dengan class yang lama (superclass) namun memiliki karakteristik yang baru. Dalam Java, subclass hanya bisa memiliki satu superclass (*single inheritance*) sedangkan superclass bisa memiliki satu atau lebih subclass.

Contoh 1 : Program Pewarisan

KelasBase.cs

```
public class KelasBase {  
    protected string kode;  
    protected string nama;  
}
```

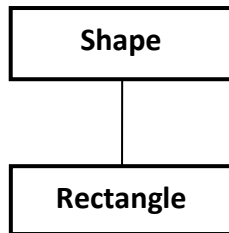
KelasDerived.cs

```
public class KelasDerived : KelasBase {  
    public string alamat;  
    public string telp;  
}
```

Penjelasan :

Sesuai dengan penjelasan sebelumnya bahwa **derived class** akan mewarisi sifat dari **base class**. Sesuai dengan contoh program 1, maka **KelasDerived** mewarisi variable **kode** dan **nama** milik **KelasBase** karena kode dan nama tidak dideklarasikan menggunakan hak akses **private**.

CONTOH DAN IMPLEMENTASI PEWARISAN (PART. 1)



Berdasarkan gambar disamping dapat diartikan bahwa class Shape merupakan class induk (**Base Class**) dan class Rectangle merupakan class turunan (**Derived Class**). Dalam artian bahwa kelas **Rectangle** akan mewarisi sifat dari kelas **Shape**.

Apabila diimplementasikan dalam sebuah program, hasilnya adalah sebagai berikut :

Shape.cs (Base Class)

```

public class Shape {
    protected int length;
    protected int width;

    public int Length {
        set {
            length = value;
        }
    }

    public int Width {
        set {
            width = value;
        }
    }
}

```

Rectangle.cs (Derived Class)

```

public class Rectangle : Shape {
    public int GetArea() {
        return (length * width);
    }
}

```

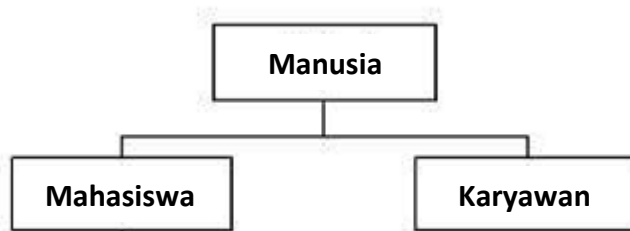
ExecuteRectangle.cs (Main Class)

```

public class ExecuteRectangle {
    static void main (string [] coba) {
        //inisialisasi
        Rectangle rcObj = new Rectangle();
        //input
        rcObj.Length = 10;
        rcObj.Width = 8;
        //output
        Console.WriteLine("Luas = {0}", rcObj.GetArea());
    }
}

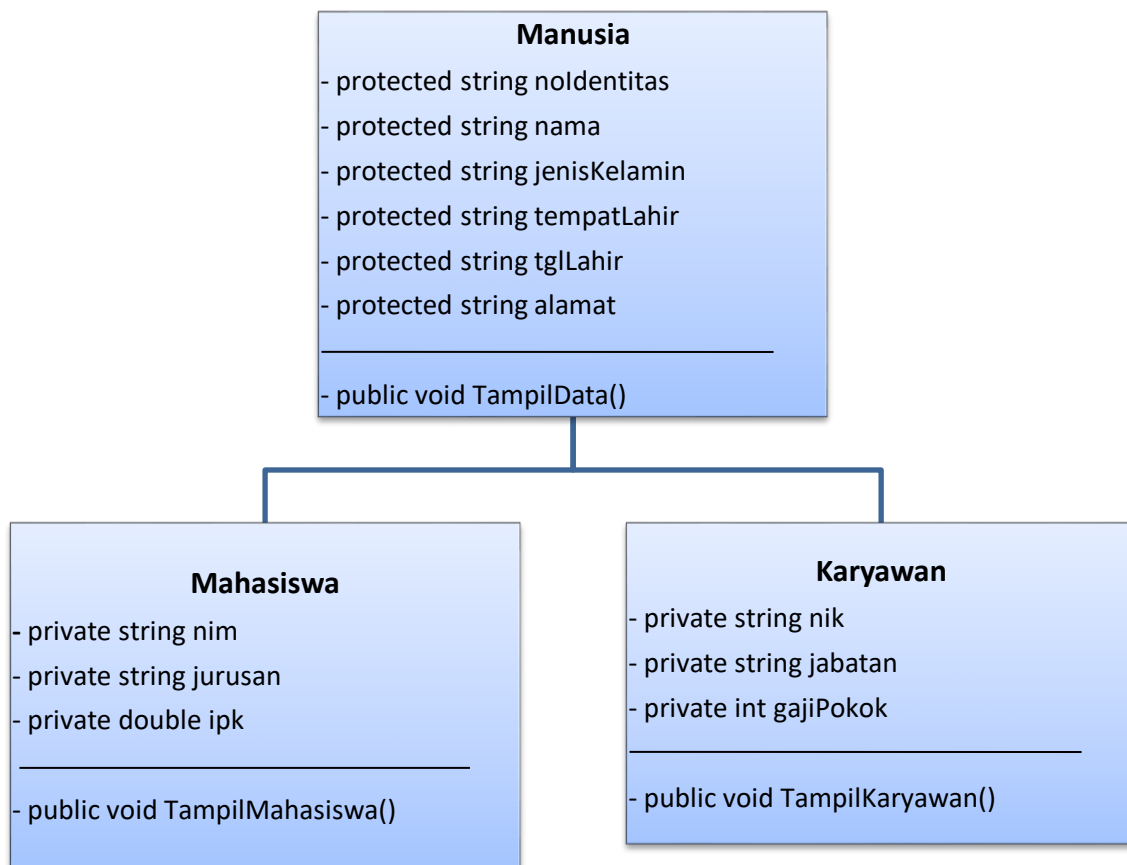
```

CONTOH DAN IMPLEMENTASI PEWARISAN (PART. 2)



Berdasarkan gambar diatas dapat diartikan bahwa class Manusia merupakan class induk (**Base Class**) dan class Mahasiswa serta Karyawan merupakan class turunan (**Derived Class**). Dalam artian bahwa kelas **Mahasiswa** dan **Karyawan** akan mewarisi sifat (dalam pemrograman dapat berupa variable atau fungsi) dari kelas **Manusia** asalkan tidak dideklarasikan menggunakan hak akses **private**.

Misalkan diberikan sifat pada masing-masing kelas seperti berikut ini :



Dari hirarki di atas apabila diimplementasikan dalam sebuah contoh program, hasilnya adalah sebagai berikut :

Manusia.cs

```

class Manusia {
    protected string noIdentitas;
    protected string nama;
    protected string jenisKelamin;
    protected string tempatLahir;
    protected string tglLahir;
    protected string alamat;

    public void TampilData() {
        Console.WriteLine("No Identitas : {0}", noIdentitas);
        Console.WriteLine("Nama : {0}", nama);
        Console.WriteLine("Jenis Kelamin : {0}", jenisKelamin);
        Console.WriteLine("Tempat, Tanggal Lahir : {0}, {1}",
            tempatLahir, tglLahir );
        Console.WriteLine("Alamat : {0}", alamat);
    }
}

```

Mahasiswa.cs

```

class Mahasiswa : Manusia {
    private string nim;
    private string jurusan;
    private double ipk;

    public void TampilMahasiswa() {
        Console.WriteLine("NIM : {0}", nim);
        Console.WriteLine("Jurusan : {0}", jurusan);
        Console.WriteLine("IPK : {0}", ipk);
    }
}

```

Karyawan.cs

```

class Karyawan : Manusia {
    private string nik;
    private string jabatan;
    private int gajiPokok;

    public void TampilKaryawan() {
        Console.WriteLine("NIK : {0}", nik);
        Console.WriteLine("Jabatan : {0}", jabatan);
        Console.WriteLine("Gaji Pokok : {0}", gajiPokok);
    }
}

```

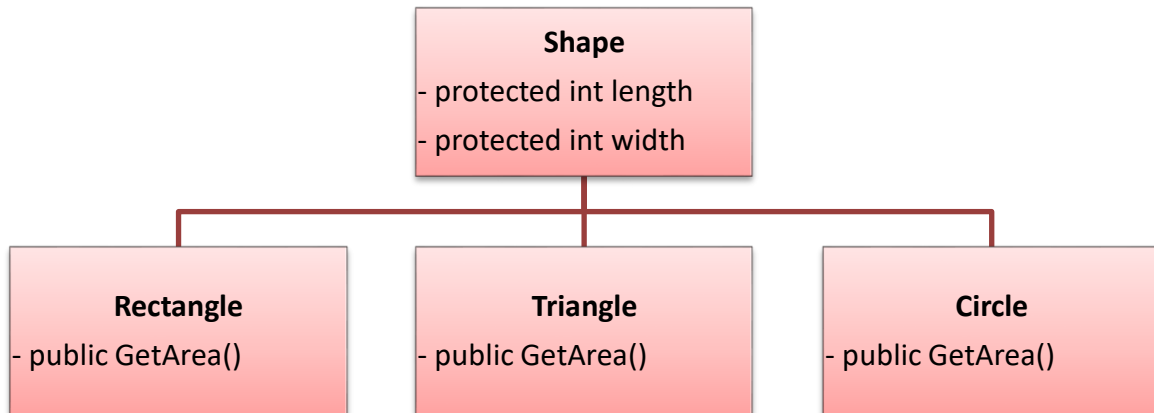
Latihan 1 :

Buat Kelas Main (**Main Class**) untuk contoh program 3. Tambahkan fungsi Get / Set apabila dibutuhkan.

LATIHAN

Latihan 2 :

Dengan memodifikasi contoh program 2, tambahkan kelas turunan yang lain seperti gambar hirarki berikut ini :



Jika sebelumnya program hanya digunakan untuk menghitung luas dari persegi panjang, maka tambahkan derived kelas lain yaitu segitiga dan lingkaran dan hitung masing-masing luas bangunnya seperti pada kelas persegi panjang. Modifikasi pada bagian **Main Class**.