

Praktikum 6

Pemrograman Berbasis Obyek (PBO)

(Inheritance)

A. Tujuan

1. Mahasiswa menerapkan konsep inheritance (Pewarisan)

B. Dasar Teori

Konsep inheritance pada umumnya digunakan untuk menghindari terjadinya duplikasi kode program. Yaitu dengan menurunkan **data member** dan **member function** yang dimiliki oleh suatu **class** kepada **class** lain. Istilah untuk Class yang akan diturunkan disebut sebagai **class induk (parent class)** atau **super class**, sedangkan class penurunan atau pewaris bisa disebut sebagai **class anak (child class)** atau **sub class**.

Pada inheritance tidak semua **data member** dan **member function** class induk dapat diturunkan, untuk data dengan hak akses **protected** dan **public** saja yang bisa diakses dari child class, tetapi untuk data dan function dengan hak akses **private** tidak akan diturunkan kepada child class. Untuk melakukan inheritance pada sebuah class dilakukan dengan menuliskan nama parent class pada saat pendefinisian child class, Berikut contoh pendefinisian child class :

```
#include <iostream>

using namespace std;

class Kendaraan {
    // kode untuk class Kendaraan
};

class SepedaMotor: public Kendaraan {
    // kode untuk class Motor
};

int main()
{
    Kendaraan KendaraanKantor;
    SepedaMotor SepedaMotorKartawan;

    return 0;
}
```

Program tersebut merupakan contoh menurunkan class Kendaraan kepada class SepedaMotor Terdapat dua class yaitu Kendaraan dan SepedaMotor,, cara

mendefinisikan class turunan yaitu dengan cara menulis tanda titik dua, diikuti hak akses class yang dalam hal ini **public**, beserta nama **parent class** yang ingin diturunkan. class **SepedaMotor** adalah anak (child) dari class **Kendaraan**. Jika class **Kendaraan** memiliki data member dan member function yang di-set sebagai **public** atau **protected**, juga akan bisa diakses dari class **SepedaMotor**. Berikut contoh program untuk melakukan akses dapat data pada parent class dan child class.

```
#include <iostream>

using namespace std;

class Kendaraan {
    // kode untuk class Kendaraan
public:
    string penggerak = "4 Tak";

    string cekKendaraan() {
        return "Ini berasal dari class Kendaraan";
    }
};

class SepedaMotor: public Kendaraan {
    // kode untuk class Motor
public:
    string merk = "Yammahaa";

    string cekSepedaMotor() {
        return "Ini berasal dari class Motor";
    }
};

int main()
{
    SepedaMotor SepedaMotorKaryawan;
    cout << SepedaMotorKaryawan.penggerak << endl;
    cout << SepedaMotorKaryawan.merk << endl;
    cout << SepedaMotorKaryawan.cekKendaraan() << endl;
    cout << SepedaMotorKaryawan.cekSepedaMotor() << endl;

    return 0;
}
```

Hasil kode program sebagai berikut :

```
4 Tak
Yammahaa
Ini berasal dari class Kendaraan
Ini berasal dari class Motor
Program ended with exit code: 0
```

Pada class **Kendaraan** memiliki satu data member **penggerak** dan satu member function (**cekKendaraan**). Sedangkan class **SepedaMotor** memiliki data member **merk** dan member function (**cekSepedaMotor**).

Karena class **SepedaMotor** adalah turunan dari class **Kendaraan**, maka dilakukan akses pada semua isi class **Kendaraan** dari object class **SepedaMotor**. Pada function main() terdapat object **SepedaMotorKaryawan** yang di instansiasi dari class **SepedaMotor**. Object **SepedaMotorKaryawan** bisa mengakses semua isi class **Kendaraan**, yakni **SepedaMotorKaryawan.penggerak** dan **SepedaMotorKaryawan.cekKendaraan()**. Seolah-olah data dan function ini di definisikan dari dalam class **SepedaMotor**. Contoh Program hak akses protected pada inheritance sebagai berikut.

```
#include <iostream>

using namespace std;

class Kendaraan {
    // kode untuk class Kendaraan

    protected:
        string tahun = "2010";
        string tipe = "Bensin";

    public:
        string penggerak = "4 Tak";

        string cekKendaraan() {
            return "Ini berasal dari class Kendaraan";
        }
};

class SepedaMotor: public Kendaraan {
    // kode untuk class Motor
    public:
        string data() {
            return "Tahun: " + tahun + ", tipe: " + tipe;
        }
};

int main()
{
    SepedaMotor SepedaMotorKaryawan;
    cout << SepedaMotorKaryawan.data() << endl;
    return 0;
}
```

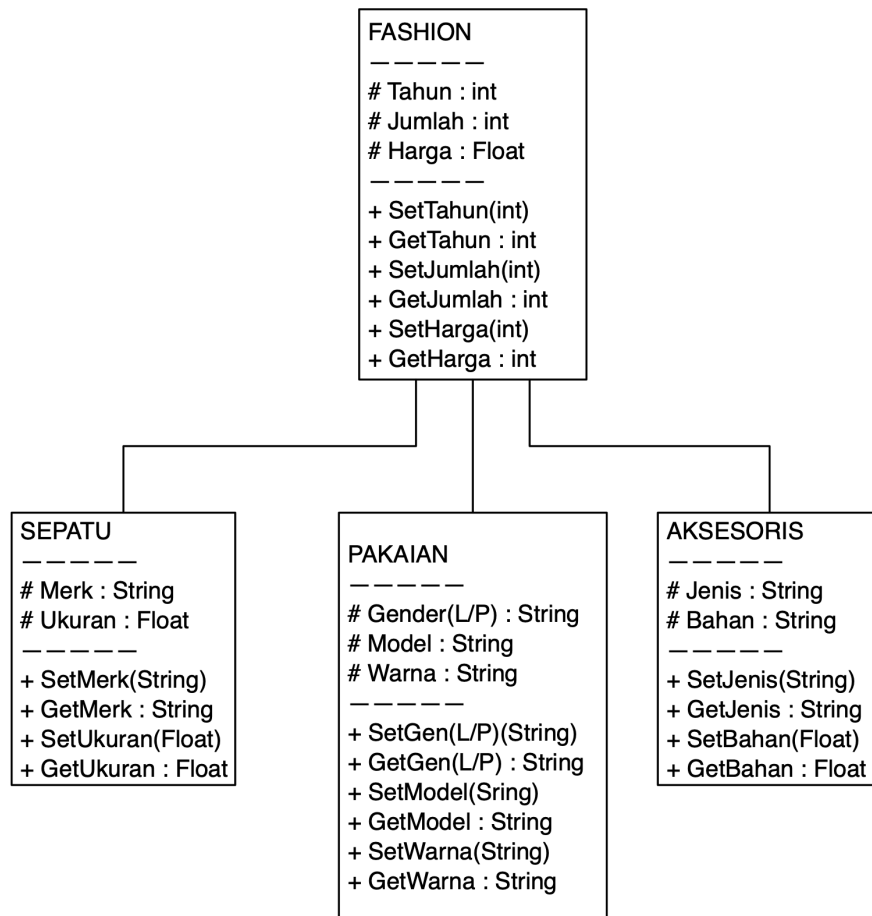
Hasil kode program sebagai berikut :

Tahun: 2010, tipe: Bensin
Program ended with exit code: 0

Pada program diatas terdapat dua data member dengan hak akses **protected**, yakni **tahun**, **tipe**. Class **Kendaraan** kemudian diturunkan kepada class **SepedaMotor**. Pada class **SepedaMotor** terdapat satu member function **data()**, yang akan mengakses data member milik class **Kendaraan**. Ini bisa dilakukan karena protected tetap **bisa diakses** dari class turunan. Namun data protected **tidak bisa diakses dari luar class**.

C. Percobaan

Buatlah program berdasarkan class diagram dibawah ini!



Selanjutnya inputkan data berdasarkan tabel berikut :

No.	Tahun	Jumlah	Harga	Gen (L/P)	Model	Warna	Merk	Jenis	Bahan	Ukuran
1	2020	123	IDR 120.000	L	Jaket	Hitam				
2	2021	330	IDR 320.000	P	Dress	Merah				
3	2022	10	IDR 1.400.000				Nike			38
4	2020	24	IDR 120.000	L	Kemaja	Biru				
5	2021	33	IDR 120.000	L	Kaos	Putih				
6	2022	56	IDR 120.000					Sabuk	Kain	
7	2019	34	IDR 80.000					Topi	Kulit	
8	2017	8	IDR 3.000.000				Adidas			42