

LAPORAN PRATIKUM
PERTEMUAN 6
PEWARISAN JAMAK (MULTIPLE INHERITANCE)



Oleh

Nama:

Alif Alpian Sahrul Muharom

(20102007)

Dosen:

Agus Priyanto, M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
PURWOKERTO

2022

I. Tujuan

- a. Mengerti dan memahami prinsip pewarisan jamak dan pemakaiannya dalam membentuk suatu kelas
- b. Mampu mengerti tentang interface dan implementasinya

II. Tool

1. Apache NetBeans IDE 13
2. Java SE Development Kit 18

III. Dasar teori

- a. Multiple inheritance adalah pewarisan class berjenjang dari satu class ke class lain. Misalkan kita memiliki class A yang diturunkan kepada class B. Kemudian class B ini juga diturunkan kepada class C, sehingga sudah terjadi sebuah multilevel inheritance. Dalam contoh ini class A adalah kakek dari class C, maka semua property dan method dari class A juga bisa diakses dari class C.
- b. Interface adalah kumpulan method yang hanya memuat deklarasi dan struktur method, tanpa detail implementasinya. Interface digunakan bila ingin mengaplikasikan suatu method yang spesifik, yaitu tidak diperoleh dari proses pewarisan kelas.

IV. Guided Buku.java

```
package main.java.com.Alpian.Pertemuan_6.Guided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Buku {
    String judul, pengarang;
    long hargaBuku;

    public Buku(String judul, String pengarang, long hargaBuku) {
        this.judul = judul;
        this.pengarang = pengarang;
        this.hargaBuku = hargaBuku;
    }

    public void cetakBuku() {
        System.out.println("\nJudul: " + judul);
        System.out.println("Pengarang: " + pengarang);
        System.out.println("Harga Buku: Rp" + hargaBuku);
        System.out.println();
    }
}
```

Penjelasan:

Pada kelas buku sebagai kelas induk dari kelas paket. Pada kelas buku terdapat judul dan pengarang dengan tipe string dan harga buku dengan tipe long

ChildCD.java

```
package main.java.com.Alpian.Pertemuan_6.Guided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class ChildCD extends CD implements InterfaceCD {
    public ChildCD(String ukuran, long hargaCD) {
        super(ukuran, hargaCD);
    }
}
```

Penjelasan:

ChildCD merupakan anak dari kelas CD dan mengimplementasikan kelas InterfaceCD.

CD.java

```
package main.java.com.Alpian.Pertemuan_6.Guided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class CD {
    String ukuran;
    long hargaCD;

    public CD(String ukuran, long hargaCD) {
        this.ukuran = ukuran;
        this.hargaCD = hargaCD;
    }
    public long getHargaCD() {
        return hargaCD;
    }

    public void cetakCD() {
        System.out.println("Ukuran CD : "+ukuran);
        System.out.println("Harga CD : Rp "+hargaCD);
        System.out.println();
    }
}
```

Penjelasan:

Pada kelas CD terdapat atribut ukuran dengan tipe string dan hargaCD dengan tipe long. Fungsi getter pada CD bernama getHargaCD.

InterfaceCD.java

```
package main.java.com.Alpian.Pertemuan_6.Guided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public interface InterfaceCD {
    void cetakCD();
    long getHargaCD();
}
```

Penjelasan:

Merupakan interface dari kelas CD

Paket.java

```
package main.java.com.Alpian.Pertemuan_6.Guided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Paket extends Buku implements InterfaceCD {
    long hargaPaket;
    InterfaceCD interfaceCD;

    public Paket(String judul, String pengarang, long hargaBuku, String
ukuran, long hargaCD) {
        super(judul, pengarang, hargaBuku);
        interfaceCD = new ChildCD(ukuran, hargaCD);
    }

    public void hitungHargaPaket() {
        hargaPaket = super.hargaBuku + getHargaCD();
    }

    @Override
    public void cetakCD() {
        interfaceCD.cetakCD();
    }

    @Override
    public long getHargaCD() {
        return interfaceCD.getHargaCD();
    }

    public void cetakPaket() {
        super.cetakBuku();
        cetakCD();
        System.out.println("Harga paket Buku dan CD: Rp" + hargaPaket +
"\n");
    }
}
```

Penjelasan:

Paket merupakan anak dari kelas Buku dan mengimplementasikan kelas InterfaceCD. Jadi, paket bisa memanggil fungsi-fungsi dari kedua kelas tersebut. Terdapat 2 method yang di override di kelas ini.

DemoPaket.java

```
package main.java.com.Alpian.Pertemuan_6.Guided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
class DemoPaket {
    public static void main(String[] args) {
        Paket a=new Paket("Pemrograman Berorientasi Objek","Benyamin
Langgu Sinaga",60000,"700 MB",50000);
        a.hitungHargaPaket();
        a.cetakPaket();
    }
}
```

Penjelasan:

DemoPaket merupakan kelas Main. Objek Paket digunakan untuk memanggil kelas paket dan akan dijalankan pada kelas Main yaitu DemoPaket.

V. Unguided

Bus.java

```
package main.java.com.Alpian.Pertemuan_6.Unguided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Bus extends Mobil {
    int kapasitasPenumpang, kapasitasBagasi;

    public Bus() {

    }

    public Bus(String noPlat, String merk, float pajak, int
    kapasitasPenumpang, int kapasitasBagasi){
        super(noPlat, merk, pajak);
        this.kapasitasPenumpang = kapasitasPenumpang;
        this.kapasitasBagasi = kapasitasBagasi;
    }

    public void infoBus() {
        System.out.println("Kapasitas Penumpang : " +
this.kapasitasPenumpang);
        System.out.println("Kapasitas Bagasi : " +
this.kapasitasBagasi);
        System.out.println("Total Pajak : Rp. " +
hitungPajak());
    }

    @Override
    public void tampilInfo() {
        System.out.println("\nInfo Kendaraan:");
        super.tampilInfo();
        infoBus();
    }

    public float hitungPajak() {
        return (this.pajak + (this.pajak * this.kapasitasPenumpang *
this.kapasitasBagasi * 0.00005f));
    }
}
```

ChildSedan.java

```
package main.java.com.Alpian.Pertemuan_6.Unguided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class ChildSedan extends Sedan implements InterfaceSedan{

    public ChildSedan(String noPlat, String merk, float pajak, String
    fasilitasKeamanan, String fasilitasKenyamanan, int kapasitasCC){
        super(noPlat, merk, pajak, fasilitasKeamanan,
        fasilitasKenyamanan, kapasitasCC);
    }
}
```

Interface.java

```
package main.java.com.Alpian.Pertemuan_6.Unguided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public interface InterfaceSedan {
    void tampilInfo();
    float hitungPajak();
}
```

Main.java

```
package main.java.com.Alpian.Pertemuan_6.Unguided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Main {
    public static void main(String[] args) {
        Mobil m = new Mobil("DD 1909 EA", "Avanza", 10000);
        Bus b = new Bus("R 123 A", "Toyota", 5000, 4, 7);
        Sedan s = new Sedan("R 456 B", "Honda Jazz", 3000, "Anti air",
        "Full AC", 6);
        MiniBus mb = new MiniBus("R 789 C", "Pajero", 15000, 9, 10,
        "Wagon", "Anti Air dan Api", "Full AC Full Senyum", 9);

        m.tampilInfo();
    }
}
```



```

        b.tampilInfo();
        b.hitungPajak();

        s.tampilInfo();

        mb.tampilInfo();

    }
}

```

MiniBus.java

```

package main.java.com.Alpian.Pertemuan_6.Unguided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class MiniBus extends Bus implements InterfaceSedan {
    String tipe;

    InterfaceSedan interfaceSedan;

    public MiniBus(){

    }

    public MiniBus(String noPlat, String merk, float pajak, int
    kapasitasPenumpang, int kapasitasBagasi, String tipe, String
    fasilitasKeamanan, String fasilitasKenyamanan, int kapasitasCC){
        super(noPlat, merk, pajak, kapasitasPenumpang, kapasitasBagasi);
        this.tipe = tipe;
        interfaceSedan = new ChildSedan(noPlat, merk, pajak,
        fasilitasKeamanan, fasilitasKenyamanan, kapasitasCC);
    }

    public void infoMiniBus() {
        if (tipe == "Pribadi"){
            System.out.println("Tipe MiniBus          : Pribadi,
            digunakan sebagai kendaraan pribadi.");
        } else if (tipe == "Wagon"){
            System.out.println("Tipe MiniBus          : Wagon, digunakan
            sebagai kendaraan angkut/travel.");
        } else{
            System.out.println("Inputan salah!");
        }

        System.out.println("Total Pajak          : Rp. " +
        hitungPajak());
    }
}

```

```

    public void tampilInfo() {
        interfaceSedan.tampilInfo();
        super.infoBus();
        infoMiniBus();
    }

    public float hitungPajak() {
        if (tipe == "Pribadi"){
            return ((interfaceSedan.hitungPajak()* 0.06f) +
(super.hitungPajak() * 0.03f));
        } else if (tipe == "Wagon"){
            return ((interfaceSedan.hitungPajak() * 0.03f) +
(super.hitungPajak() * 0.05f));
        } else{
            return 0;
        }
    }
}

```

Mobil.java

```

package main.java.com.Alpian.Pertemuan_6.Unguided;
/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Mobil {
    String noPlat;
    String merk;
    float pajak;

    public Mobil(){

    }

    public Mobil(String noPlat, String merk, float pajak){
        this.noPlat = noPlat;
        this.merk = merk;
        this.pajak = pajak;
    }

    public void tampilInfo() {
        System.out.println("Nomor PLat           : " + this.noPlat);
        System.out.println("Merk Mobil           : " + this.merk);
        System.out.println("Pajak                : Rp. " + this.pajak);
    }
}

```

Sedan.java

```
package main.java.com.Alpian.Pertemuan_6.Unguided;

/**
 *
 * @author
 * Alif Alpian Sahrul Muharom
 * 20102007
 * IF-08-0
 */
public class Sedan extends Mobil {
    String fasilitasKeamanan, fasilitasKenyamanan;
    int kapasitasCC;

    public Sedan(){

    }

    public Sedan(String noPlat, String merk, float pajak, String
fasilitasKeamanan, String fasilitasKenyamanan, int kapasitasCC){
        super(noPlat, merk, pajak);
        this.fasilitasKeamanan = fasilitasKeamanan;
        this.fasilitasKenyamanan = fasilitasKenyamanan;
        this.kapasitasCC = kapasitasCC;
    }

    public void infoSedan() {
        System.out.println("Fasilitas Keamanan : " +
this.fasilitasKeamanan);
        System.out.println("Fasilitas Kenyamanan : " +
this.fasilitasKenyamanan);
        System.out.println("Kapasitas CC : " +
this.kapasitasCC);
        System.out.println("Total Pajak : Rp. " +
hitungPajak());
    }

    @Override
    public void tampilInfo() {
        System.out.println("\nInformasi Kendaraan:");
        super.tampilInfo();
        infoSedan();
    }

    public float hitungPajak() {
        return (this.pajak + (this.pajak * this.kapasitasCC *
0.00005f));
    }
}
```

VI. Kesimpulan

Multilevel atau Multiple Inheritance adalah pewarisan class berjenjang dari satu class ke class lain. Misalkan kita memiliki class A yang diturunkan kepada class B. Kemudian class B ini juga diturunkan kepada class C, sehingga sudah terjadi sebuah multilevel inheritance. Dalam contoh ini class A adalah kakek atau grandparent dari class C, maka semua property dan method dari class A juga bisa diakses dari class C. Interface adalah kumpulan method yang hanya memuat deklarasi dan struktur method, tanpa detail implementasinya. interface digunakan bila ingin mengaplikasikan suatu method yang spesifik, yaitu tidak diperoleh dari proses pewarisan kelas. Interface bersifat disisipkan (embeded) pada program, dan programmer diberi keleluasaan untuk merancang dan mendefinisikan sendiri detail prosesnya.

