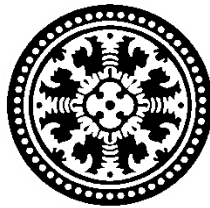


INHERITANCE DAN POLYMORPHISM

Laporan ini bertujuan untuk memenuhi tugas dala perkuliahan Pemrograman Berorientasi

Objek



KETUT YOGI PRASETYA

(1508605028)

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN ILMU KOMPUTER

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS UDAYANA

BUKIT JIMBARAN

2017

1. Pengertian Pewarisan (Inheritance)

Konsep inheritance ini mengadopsi dunia riil dimana suatu entitas/obyek dapat mempunyai entitas/obyek turunan. Dengan konsep inheritance, sebuah class dapat mempunyai class turunan. Suatu class yang mempunyai class turunan dinamakan parent class atau base class. Sedangkan class turunan itu sendiri seringkali disebut subclass atau child class. Suatu subclass dapat mewarisi apa-apa yang dipunyai oleh parent class. Karena suatu subclass dapat mewarisi apa-apa yang dipunyai oleh parent class-nya, maka member dari suatu subclass adalah terdiri dari apa-apa yang ia punyai dan juga apa-apa yang ia warisi dari class parent-nya. Dapat dilihat bahwa, semakin kebawah, class akan semakin bersifat spesifik. Class mamalia memiliki seluruh sifat yang dimiliki oleh binatang, demikian halnya juga macan, kucing, Paus dan Monyet memiliki seluruh sifat yang diturunkan dari class mamalia. Dengan konsep ini, karakteristik yang dimiliki oleh class binatang cukup didefinisikan dalam class binatang saja. Class mamalia tidak perlu mendefinisikan ulang apa yang telah dimiliki oleh class binatang, karena sebagai class turunannya, ia akan mendapatkan karakteristik dari class binatang secara otomatis. Demikian juga dengan class macan, kucing, Paus dan monyet, hanya perlu mendefinisikan karakteristik yang spesifik dimiliki oleh class-nya masing-masing. Dengan memanfaatkan konsep pewarisan ini dalam pemrograman, maka hanya perlu mendefinisikan karakteristik yang lebih umum akan didapatkan dari class darimana ia diturunkan.

2. Deklarasi Pewarisan (Inheritance)

Deklarasi yang digunakan adalah dengan menambahkan kata kunci `extends` setelah deklarasi nama class, kemudian diikuti dengan nama parent class-nya. Kata kunci `extends` tersebut memberitahu kompiler Java bahwa kita ingin melakukan perluasan class.

```
public class B extends A { ... }
```

Pada saat dikompilasi, Kompiler Java akan membacanya sebagai subclass dari class `Object`.

```
public class A extends Object { ... }
```

3. Penerapan Inheritance

Kita baru perlu menerapkan inheritance pada saat kita jumpai ada suatu class yang dapat diperluas dari class lain. Misal terdapat class `Pegawai` public class `Pegawai`

```
{ public String nama; public double gaji; }
```

Misal terdapat class Manager public class Manajer

```
{ public String nama; public double gaji; public String departemen; }.
```

Dari 2 buah class diatas, kita lihat class Manajer mempunyai data member yang identik sama dengan class Pegawai, hanya saja ada tambahan data member departemen. Sebenarnya yang terjadi disana adalah class Manajer merupakan perluasan dari class Pegawai dengan tambahan data member departemen. Disini perlu memakai konsep inheritance, sehingga class Manajer dapat kita tuliskan seperti berikut :

```
public class Manajer extends Pegawai { public String departemen; }
```

4. Keuntungan Inheritance

Subclass menyediakan state/behaviour yang spesifik yang membedakannya dengan superclass, hal ini akan memungkinkan programmer Java untuk menggunakan ulang source code dari superclass yang telah ada. Programmer Java dapat mendefinisikan superclass khusus yang bersifat generik, yang disebut abstract class, untuk mendefinisikan class dengan behaviour dan state secara umum.

5. Istilah Inheritance

Extends keyword harus kita tambahkan pada definisi class yang menjadi subclass. Superclass Superclass digunakan untuk menunjukkan hirarki class yang berarti class dasar dari subclass/class anak. Subclass Subclass adalah class anak atau turunan secara hirarki dari superclass. Super Keyword ini digunakan untuk memanggil konstruktor dari superclass atau menjadi variabel yang mengacu pada superclass. Methode Overriding Pendefinisian ulang method yang sama pada subclass. Dalam inheritance, method overriding berbeda dengan method overloading. Kalau method overriding adalah mendefinisikan kembali method yang sama, baik nama method maupun signature atau parameter yang diperlukan dalam subclass, kalau method overloading adalah mendefinisikan method yang memiliki nama yang sama, tetapi dengan signature yang berbeda dalam definisi class yang sama.

6. Contoh Latihan Inheritance :

```
class Kotak{  
    double sisil;
```

```

double sisi2;
double sisi3;
Kotak() {
}
Kotak(double satu, double dua, double tiga){
sisi1=satu;
sisi2=dua;
sisi3=tiga;
}
public void CetakKotak(){
System.out.println("Isi Kotak adalah = " + sisi1 + sisi2 + sisi3);}
}
public class Kotakku extends Kotak{
double sisi1;
Kotakku(double pertama, double kedua, double ketiga, double
tambahan)
{
super(pertama, kedua, ketiga);
sisi1=tambahan;
}
public static void main(String[] args) {
Kotakku Kotaksaya= new Kotakku(10,5,15,30);
Kotaksaya.CetakKotak();
System.out.println("Sisi Pertama dari Kotak adalah = " +
Kotaksaya.sisi1);
System.out.println("Sisi Pertama dari Kotak adalah = " +
Kotaksaya.sisi2);
System.out.println("Sisi Pertama dari Kotak adalah = " +
Kotaksaya.sisi3);
}
}

```

7. Polymorphism

Suatu aksi yang memungkinkan pemrogram menyampaikan pesan tertentu keluar dari hirarki obyeknya, dimana obyek yang berbeda memberikan tanggapan/respon terhadap pesan yang sama sesuai dengan sifat masing-masing obyek. Atau Polymorphism dapat berarti banyak bentuk, maksudnya yaitu kita dapat menimpa (override), suatu method, yang berasal dari parent class (super class) dimana object tersebut diturunkan, sehingga memiliki kelakuan yang berbeda. Polymorphism dapat disamakan dengan Method Overloading, dimana di dalam sebuah class terdapat beberapa method dengan nama sama. Sebuah Method bernama CetakData() yang misalnya berfungsi menampilkan data String, tentu tidak bisa menangani masukan berupa data numerik, boolean, ataupun karakter, demikian juga sebaliknya. Solusi yang bisa dilakukan adalah dengan menyediakan beberapa Method CetakData() yang akan menangani setiap tipe data,

sehingga data apapun yang diberikan sebagai parameter tetap bisa diproses.

8. Contoh Latihan Polymorphims

Simpan dengan nama CetakDataTipe.java

```
class CetakDataTipe {  
    // membuat method dengan nama sama tapi berbeda tipe data  
    public void CetakData(String Cetak) {  
        System.out.println(Cetak);  
    }  
    public void CetakData(int Cetak) {  
        System.out.println(Cetak);  
    }  
    public void CetakData(double Cetak) {  
        System.out.println(Cetak);  
    }  
    public void CetakData(char Cetak) {  
        System.out.println(Cetak);  
    }  
}
```

Simpan dengan nama Polymorphism.java

```
public class Polymorphism {  
    public static void main(String[] args) {  
        CetakDataTipe data = new CetakDataTipe();  
        System.out.print("Cetak Data Tipe String : ");  
        data.CetakData("Sumatra");  
        System.out.print("Cetak Data Tipe Integer : ");  
        data.CetakData(1989);  
        System.out.print("Cetak Data Tipe Double : ");  
        data.CetakData(16.5 / 2);  
        System.out.print("Cetak Data Tipe Char : ");  
        data.CetakData('N');  
    }  
}
```