



unindra
universitas indraprasta PGRI

Banyak Bentuk

Polymorphism

PERTEMUAN 11

POLYMORPHISM

- Polimorfisme terbagi menjadi dua suku kata yaitu, *Poly* yang berarti banyak dan *Morfisme* yang berarti bentuk. Dalam ilmu sains, Polimorfisme (*polymorphism*) adalah sebuah prinsip dalam biologi di mana organisme atau spesies memiliki banyak bentuk serta tahapan (*stages*). Prinsip tersebut diterapkan juga pada bahasa Java.
- Polimorfisme dalam OOP merupakan sebuah konsep OOP di mana *class* memiliki banyak “bentuk” *method* yang berbeda, meskipun namanya sama. Maksud dari “bentuk” adalah isinya yang berbeda, namun tipe data dan parameternya berbeda.

Polimorfisme juga dapat diartikan sebagai teknik *programming* yang mengarahkan kamu untuk memprogram secara general daripada secara spesifik.

Contohnya :

1. Kita memiliki tiga class yang berbeda yaitu: “Kelinci”, “Kucing”, dan “Sapi”. Di mana ketiga *class* tersebut merupakan turunan dari *class* “Hewan”.
2. Kita memiliki tiga class yang berbeda yaitu: “Makan”, “Minum”, dan “Tidur”. Di mana ketiga *class* tersebut merupakan turunan dari *class* “Kegiatan”.

Polimorfisme pada Java memiliki 2 macam yaitu diantaranya:

- Static Polymorphism (Polimorfisme statis).
- Dynamic Polymorphism (Polimorfisme dinamis).

Perbedaan keduanya terletak pada cara membuat polimorfisme. Polimorfisme statis menggunakan *method overloading*, sedangkan polimorfisme dinamis menggunakan *method overriding*.

Penggunaan metode dengan nama sama dapat melalui *method overloading* dan *method overriding*. Peran polymorphism tidak terbatas hanya pada hal-hal tersebut. Ada kaitan erat antara polymorphism dan inheritance (turunan).

Contoh Program polymorphism

The image displays a screenshot of an IDE with several windows open, illustrating a Java program that demonstrates polymorphism. The windows are:

- hewan_poly**: Contains the base class `hewan_poly` with a `jawab ()` method that returns "Macam-macam Nama dan Makanan Hewan : \n".
- sapi_poly**: Contains the class `sapi_poly` which extends `hewan_poly` and overrides the `jawab ()` method to return "Sapi : Rumput".
- kucing_poly**: Contains the class `kucing_poly` which extends `hewan_poly` and overrides the `jawab ()` method to return "Kucing : Ikan".
- kelinci_poly**: Contains the class `kelinci_poly` which extends `hewan_poly` and overrides the `jawab ()` method to return "Kelinci : Wortel".
- panggilpoly**: Contains the `main` method of a class that creates instances of `hewan_poly`, `kelinci_poly`, `kucing_poly`, and `sapi_poly`. It then calls the `jawab ()` method on each instance and prints the results.
- Blue: Terminal Window - P...**: Shows the output of the program execution, displaying the results of the `jawab ()` calls for each animal.

```
class hewan_poly{
    public String jawab ()
    {
        return ("Macam-macam Nama dan Makanan Hewan : \n" ) ;
    }
}

class sapi_poly extends hewan_poly {
    public String jawab ()
    {
        return ("Sapi : Rumput " ) ;
    }
}

class kelinci_poly extends hewan_poly {
    public String jawab ()
    {
        return ("Kelinci : Wortel " ) ;
    }
}

class kucing_poly extends hewan_poly {
    public String jawab ()
    {
        return ("Kucing : Ikan " ) ;
    }
}

public class panggilpoly {
    public static void main (String [] xx){
        hewan_poly hwn = new hewan_poly ();
        kelinci_poly kln = new kelinci_poly ();
        kucing_poly kc = new kucing_poly ();
        sapi_poly sp = new sapi_poly ();

        String panggil = " " ;

        panggil = hwn.jawab ();
        System.out.println (panggil);
        panggil = kln.jawab ();
        System.out.println ("Pertama > " +panggil);
        panggil = kc.jawab ();
        System.out.println ("Kedua > " +panggil);
        panggil = sp.jawab ();
        System.out.println ("Ketiga > " +panggil);
    }
}
```

Macam-macam Nama dan Makanan Hewan :
Pertama > Kelinci : Wortel
Kedua > Kucing : Ikan
Ketiga > Sapi : Rumput

```
class kendaraan{
    String nama;
    int jumlahroda;
    public kendaraan(String nama, int jumlahroda){
        this.nama=nama;
        this.jumlahroda=jumlahroda;
    }
    public void info(){
        System.out.println("Nama : "+this.nama);
        System.out.println("Jumlah Roda : "+this.jumlahroda);
    }
}
```

```
class mobil extends kendaraan{
    String merk;
    public mobil(String nama,int jumlahroda,String merk){
        super(nama,jumlahroda);
        this.merk=merk;
    }
    public void info(){
        super.info();
        System.out.println("Merk : "+this.merk);
    }
}
```

```
class bmw extends mobil{
    String jenismobil;
    public bmw(String nama,int jumlahroda,String merk,String jenismobil){
        super(nama,jumlahroda,merk);
        this.jenismobil=jenismobil;
    }
    public void info(){
        super.info();
        System.out.println("Jenis Mobil : "+this.jenismobil);
    }
    public void info2(){
        System.out.println("ini contoh polimorfisme");
    }
}
```

```
public class poly{
    public static void main(String[]args){
        bmw cetak2=new bmw("Mobil",4,"BMW","Manual");
        cetak2.info();
    }
}
```

Latihan

1. Buatlah program Inputan Scanner dengan konsep PBO (inheritance, encapsulation, polymorphism) dengan ketentuan :
 - a. NPM Ganjil membuat tema Karyawan
 - b. NPM Genap membuat tema Penjualan
 - c. Tentukan atribut dan accessnya yang di buat dalam sebuah method (atribut minimal 3 untuk tiap methodnya)
2. Berikan simpulan dan perbedaan dari 3 konsep PBO yang sudah kalian pelajari !