

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

PERTEMUAN KE-7



Disusun Oleh :

NAMA : Raden Isnawan Argi Aryasatya

NIM : 195410257

JURUSAN : Teknik Informatika

JENJANG : S1

KELAS : TI-5

Laboratorium Terpadu
Sekolah Tinggi Manajemen Informatika Komputer
AKAKOM
YOGYAKARTA

2021

PERTEMUAN KE-7

(PEWARISAN (INHERITANCE))

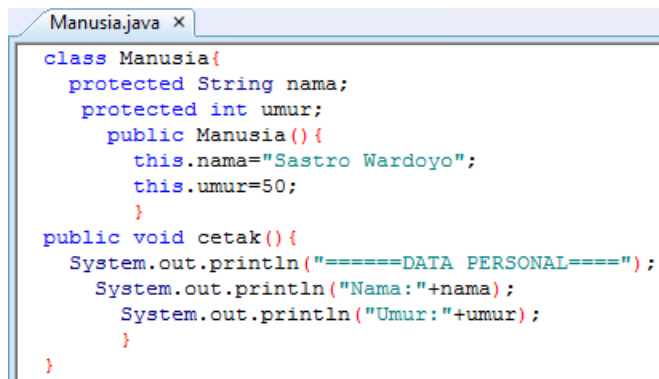
TUJUAN

Dapat memahami konsep inheritance dan dapat membuat program yang menggunakan konsep inheritance.

DASAR TEORI

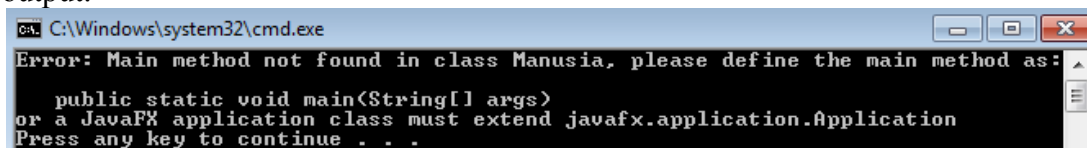
Pewarisan (Inheritance) merupakan sifat dalam bahasa berorientasi objek yang memungkinkan sifat-sifat dari suatu kelas diturunkan ke kelas lain. Sistem OOP memungkinkan kita untuk mendefinisikan suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada. Penurunan sifat ini bisa dilakukan dilakukan secara bertingkat-tingkat, sehingga semakin ke bawah maka kelas tersebut semakin spesifik, contoh, apabila kelas B adalah turunan dari kelas A, dalam proses pembuatan kelas turunan tersebut, Anda dapat menambahkan sifat dan perilaku baru ke dalam kelas B, yang sebelumnya tidak dimiliki dalam kelas A.

PRAKTIK 1



```
Manusia.java x
class Manusia{
    protected String nama;
    protected int umur;
    public Manusia(){
        this.nama="Sastro Wardoyo";
        this.umur=50;
    }
    public void cetak(){
        System.out.println("====DATA PERSONAL====");
        System.out.println("Nama:"+nama);
        System.out.println("Umur:"+umur);
    }
}
```

output:



```
C:\Windows\system32\cmd.exe
Error: Main method not found in class Manusia, please define the main method as:
    public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
Press any key to continue . . .
```

penjelasan:

Pada program di atas, kita membuat sebuah bagian program Inheritance bernama superclass. Konsep inheritance membuat sebuah struktur atau 'hierarchy' class dalam kode program. class Manusia yang akan diturunkan tersebut berperan sebagai class induk (parent class), super class, atau bisa juga disebut dengan base class. Superclass adalah bagian dari inheritance yang paling general (luas/umum), lalu nantinya semakin spesifik saat kita membuat subclass.

Untuk pembuatan/penulisan program di atas, kita awali dengan membuat class Manusia yang di dalamnya ada atribut/variabel String nama dan int umur yang keduanya memiliki modifier protected. Modifier protected akan membuat member dan class hanya bisa diakses dari: class itu sendiri; sub class atau class anak; dan package (class yang berada satu package dengannya). Kemudian kita buat konstruktor yang tidak memiliki argumen dan memiliki nama sama dengan class nya yaitu Public Manusia(), yang di dalamnya digunakan untuk menampung baris-baris kode untuk inisialisasi variabel yaitu:

```
this.nama="Sastro Wardoyo";
    this.umur=50;
```

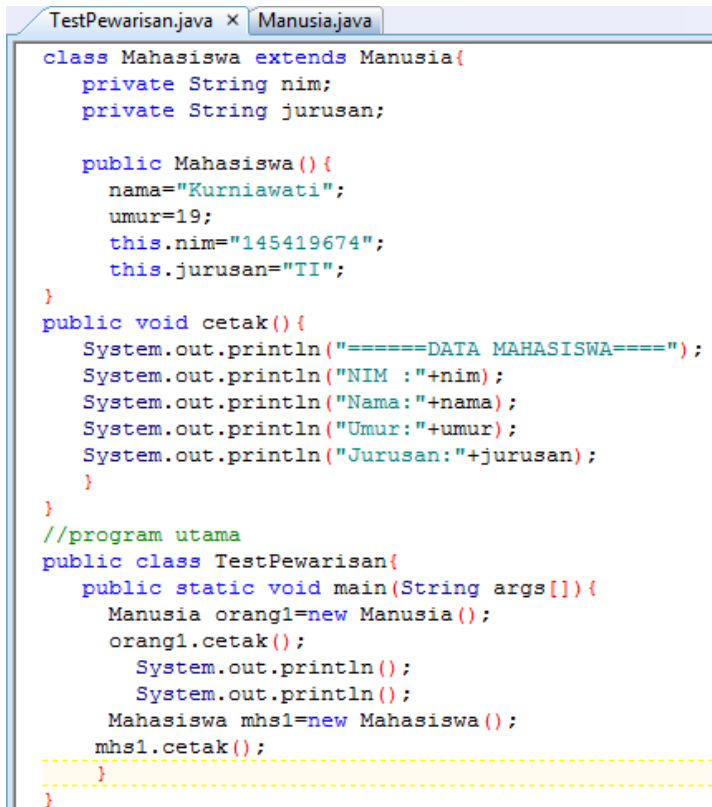
Di bawahnya kita buat method bernama cetak() yang akan menuliskan dan menampilkan output seperti berikut:

```
System.out.println("====DATA PERSONAL====");
```

```
System.out.println("Nama:"+nama);
```

```
System.out.println("Umur:"+umur);
```

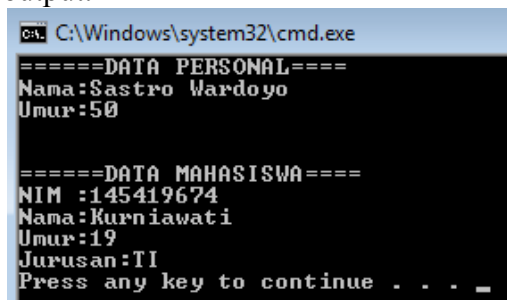
PRAKTIK 2



```
TestPewarisan.java x Manusia.java
class Mahasiswa extends Manusia{
    private String nim;
    private String jurusan;

    public Mahasiswa(){
        nama="Kurniawati";
        umur=19;
        this.nim="145419674";
        this.jurusan="TI";
    }
    public void cetak(){
        System.out.println("====DATA MAHASISWA====");
        System.out.println("NIM :"+nim);
        System.out.println("Nama:"+nama);
        System.out.println("Umur:"+umur);
        System.out.println("Jurusan:"+jurusan);
    }
}
//program utama
public class TestPewarisan{
    public static void main(String args[]){
        Manusia orang1=new Manusia();
        orang1.cetak();
        System.out.println();
        System.out.println();
        Mahasiswa mhs1=new Mahasiswa();
        mhs1.cetak();
    }
}
```

output:



```
C:\Windows\system32\cmd.exe
====DATA PERSONAL====
Nama:Sastro Wardoyo
Umur:50

====DATA MAHASISWA====
NIM :145419674
Nama:Kurniawati
Umur:19
Jurusan:TI
Press any key to continue . . . _
```

penjelasan:

Kemudian kita buat subclass Mahasiswa yang inherit dari class Manusia yaitu dengan baris kode **class Mahasiswa extends Manusia**. Fungsi dari extends tersebut adalah memanggil fungsi atau method dari class lain, sehingga kita tidak perlu lagi membuat script yang sama pada class yang akan kita buat dengan class yang kita buat sebelumnya. Class Mahasiswa adalah class yang 'menerima penurunan' yang disebut sebagai class anak (child class), sub class, derived class atau heir class. Di dalamnya kita deklarasi dua variabel bermodifer private yaitu String nim dan String jurusan.

Lalu kita buat konstruktor tanpa argumen bernama Public Mahasiswa(). Di dalamnya kita memberi value kepada nama dan umur yaitu dengan:

```
nama="Kurniawati";
umur=19;
```

kita tidak perlu menggunakan this lagi pada kedua variabel tersebut karena di class Manusia, kedua variabel tersebut sudah diinisialisasi. Kemudian di bawahnya persis, kita buat baris kode untuk inisialisasi dan memberi nilai kepada variabel nim dan jurusan yaitu dengan baris kode berikut:

```
this.nim="145419674";
this.jurusan="TI";
```

Setelah itu kita buat method cetak() untuk menuliskan dan mencetak beberapa kalimat seperti berikut:

```
public void cetak(){
    System.out.println("=====DATA MAHASISWA=====");
    System.out.println("NIM :"+nim);
    System.out.println("Nama:"+nama);
    System.out.println("Umur:"+umur);
    System.out.println("Jurusan:"+jurusan);
}
```

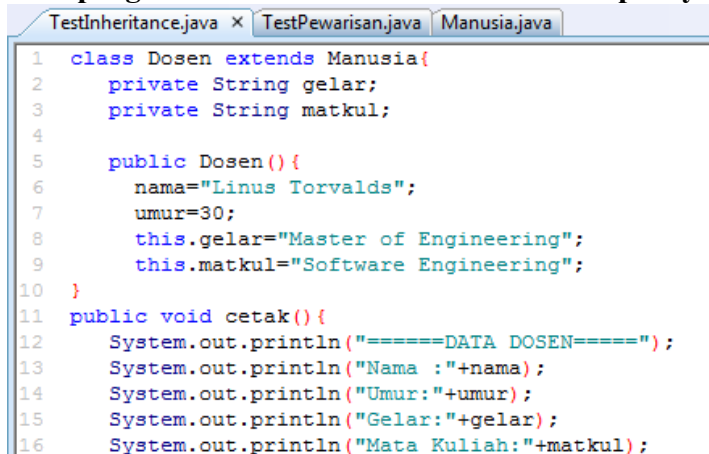
Terakhir, kita buat program utama bernama public class TestPewarisan. Kita awali dengan menuliskan **public static void main (String args[])** sebagai method utama dalam menjalankan program. Public, berarti metode ini dapat dipanggil dari luar class. Static, menunjukkan metode ini bersifat sama untuk semua class. Void, berarti metode ini tidak mengembalikan nilai. Argument args[] adalah array objek string argument baris-baris perintah.

Di dalamnya kita lakukan instansiasi yaitu pembuatan objek. Objek pertama yang kita buat adalah objek dari class Manusia yaitu objek yang bernama orang1 dengan baris kode **Manusia orang1=new Manusia()**. Kemudian kita buat objek itu berinisialisasi dengan method cetak supaya memunculkan kalimat-kalimat dan nilai yang tadi sudah kita isi di dalam method cetak(), yaitu dengan baris kode **orang1.cetak()**. Di bawahnya kita tulis System.out.println(); dua kali untuk memberi jarak dengan output objek selanjutnya.

Objek kedua yang kita buat bernama mhs1 yang merupakan objek milik class Mahasiswa. Kita tuliskan **Mahasiswa mhs1=new Mahasiswa()**; kemudian kita cetak hasil dengan **mhs1.cetak()**;

LATIHAN

Buatlah kelas Dosen turunan dari kelas Manusia lengkap dengan atribut dan method serta buat program utama untuk melihat hasil outputnya!



```
TestInheritance.java x TestPewarisan.java Manusia.java
1 class Dosen extends Manusia{
2     private String gelar;
3     private String matkul;
4
5     public Dosen(){
6         nama="Linus Torvalds";
7         umur=30;
8         this.gelar="Master of Engineering";
9         this.matkul="Software Engineering";
10    }
11    public void cetak(){
12        System.out.println("=====DATA DOSEN=====");
13        System.out.println("Nama :"+nama);
14        System.out.println("Umur:"+umur);
15        System.out.println("Gelar:"+gelar);
16        System.out.println("Mata Kuliah:"+matkul);
```

```

17     }
18 }
19
20 public class TestInheritance{
21     public static void main(String args[]){
22
23         Manusia orang1=new Manusia();
24         orang1.cetak();
25         System.out.println();
26         System.out.println();
27         Mahasiswa mhs1=new Mahasiswa();
28         mhs1.cetak();
29         System.out.println();
30         System.out.println();
31         Dosen dosen1=new Dosen();
32         dosen1.cetak();
33         -----
34     }
35 }

```

Output:

```

C:\Windows\system32\cmd.exe
=====DATA PERSONAL=====
Nama:Sastro Wardoyo
Umur:50

=====DATA MAHASISWA=====
NIM :145419674
Nama:Kurniawati
Umur:19
Jurusan:TI

=====DATA DOSEN=====
Nama :Linus Torvalds
Umur:30
Gelar:Master of Engineering
Mata Kuliah:Software Engineering
Press any key to continue . . .

```

penjelasan:

kita buat subclass Dosen yang inherit dari class Manusia yaitu dengan baris kode **class Dosen extends Manusia**. Fungsi dari extends tersebut adalah memanggil fungsi atau method dari class lain, sehingga kita tidak perlu lagi membuat script yang sama pada class yang akan kita buat dengan class yang kita buat sebelumnya. Class Dosen adalah class yang ‘menerima penurunan’ yang disebut sebagai class anak (child class), sub class, derived class atau heir class. Di dalamnya kita deklarasikan dua variabel bermodifler private yaitu String gelar dan String matkul.

Lalu kita buat konstruktor tanpa argumen bernama Public Dosen(). Di dalamnya kita memberi value kepada nama dan umur yaitu dengan:

nama="Linus Torvalds";

umur=30;

kita tidak perlu menggunakan this lagi pada kedua variabel tersebut karena di class Manusia, kedua variabel tersebut sudah diinisialisasi. Kemudian di bawahnya persis, kita buat baris kode untuk inisialisasi dan memberi nilai kepada variabel gelar dan matkul yaitu dengan baris kode berikut:

this.gelar="Master of Engineering";

this.matkul="Software Engineering";

Setelah itu kita buat method cetak() untuk menuliskan dan mencetak beberapa kalimat seperti berikut:

```

public void cetak(){
    System.out.println("=====DATA DOSEN=====");
    System.out.println("Nama :"+nama);

```

```
System.out.println("Umur:"+umur);
System.out.println("Gelar:"+gelar);
System.out.println("Mata Kuliah:"+matkul);
}
```

Terakhir di dalam program utama, kita menambahkan objek baru bernama dosen1 yaitu dengan **baris kode Dosen dosen1=new Dosen();** yang akan menjalankan method cetak() yang akan menampilkan output yaitu dengan menulis **dosen1.cetak();**. Sehingga tampilan output program adalah hasil dari method cetak() milik 3 kelas yaitu Manusia, Mahasiswa, dan Dosen.

KESIMPULAN

Dari pembelajaran tentang inheritance ini, kita jadi mengerti tentang istilah-istilah dalam inheritance, seperti:

- Extends = Keyword ini harus kita tambahkan pada definisi class yang menjadi subclass.
- Superclass = digunakan untuk menunjukkan hirarki class yang berarti class dasar dari subclass/class anak.
- 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.

Pembelajaran tentang inheritance ini juga memberi kita wawasan tentang keuntungan menggunakan inheritance yaitu:

- 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.
- Kita dapat mendefinisikan superclass khusus yang bersifat generik, yang disebut abstract class, untuk mendefinisikan class dengan behaviour dan state secara umum.

Terima Kasih.