

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 4
“INHERITENCE”**



Disusun oleh
Bunga Laelatul Muna
NIM : 21102010

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2023**

BAB 1 – TUJUAN PRAKTIKUM

Tujuan dari praktikum instalasi Java (IntelliJ IDEA) adalah untuk memberikan pengetahuan dan keterampilan kepada mahasiswa tentang kelas dan objek yang ada di Java

1. Mahasiswa dapat memahami konsep PBO yaitu abstraction, encapsulation dan inheritance
2. Mahasiswa dapat membuat program menggunakan inheritance

BAB II- DASAR TEORI

Dalam Pemrograman Berorientasi Objek, terdapat empat prinsip utama yang Anda wajib ketahui:

a. Abstraction

Sederhananya, Abstraction adalah menggunakan hal-hal sederhana untuk mewakili hal-hal rumit atau kompleks. Untuk mempermudah memahami prinsip ini,

```
//before
if (number > 0 && number <= angka.length) {
    System.out.println("Error!");
}

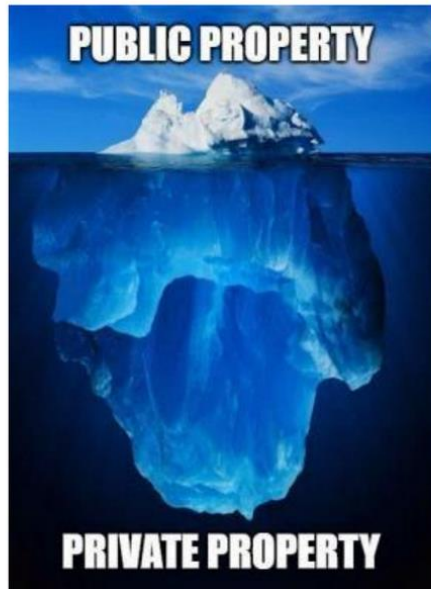
//after
boolean inputInvalid = number > 0 && number <= angka.length;
if (inputInvalid) {
    System.out.println("Error!");
}
```

Pada contoh di atas, pada if pertama, orang lain yang belum paham tentang kode kita mungkin akan bertanya-tanya kondisi apa yang sedang dicek di if tersebut. Sedangkan pada if kedua, jika kita menunjukkan kode kita pada orang awam pun, orang itu akan paham bahwa kode tersebut sedang mengecek apakah input yang diterima program valid atau invalid.

b. Encapsulation

Pada dasarnya, prinsip Encapsulation adalah menyembunyikan data dan proses kompleks dengan memberikan modifier Private dan memberikan modifier Public

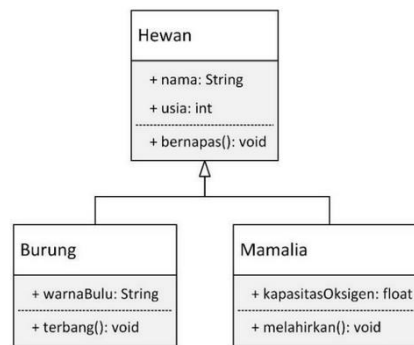
untuk data dan proses-proses sederhana. Alasan mengapa Anda harus memperhatikan prinsip Encapsulation adalah karenabiasanya proses-proses kompleks sangat sensitif dengan yang namanya error dan tentunya Anda tidak ingin sembarang orang dapat mengaksesnya karena mungkin dapat memicu hal-hal tidak diinginkan di program Anda.



Bayangkan sebuah TV. Anggaplah tombol-tombol di depan TV yang dapat kita tekan adalah method Public milik TV tersebut, sedangkan proses kompleks yang terjadi di belakang layar saat kita menekan tombol-tombol tersebut adalah method Private. Sebagai pemilik TV, kita tahu apa fungsi dari masing-masing tombol, tetapi kita tidak tahu dan tidak peduli apa proses kompleks yang terjadi di belakang layar saat kita menekan tombol tersebut.

c. Inheritance

Inheritance (arti: Pewarisan) merupakan konsep menurunkan atribut dan method milik sebuah class ke class lain. Perhatikan contoh Class Diagram berikut:



Pada class diagram di atas, class Hewan disebut sebagai Parent class atau Superclass sedangkan class Burung dan Mamalia disebut sebagai Child class atau Subclass. Semua attribute dan method yang dimiliki oleh Parent class pasti dimiliki oleh semua Child class-nya walaupun Anda tidak menulis attribute dan method tersebut di dalam source code milik Child class-nya. Hal ini tidak berlaku sebaliknya. Attribute dan method yang dimiliki sebuah Child class belum tentu dimiliki oleh Parent class-nya. Dalam merancang class, jika Anda menemukan ada beberapa class dengan attribute dan method yang konteksnya sama, lebih baik menggunakan Inheritance saja. Salah satu keuntungannya adalah jika Anda sewaktu-waktu ingin mengubah kode Anda, Anda hanya perlu mengubah kode Anda satu kali saja, sedangkan jika kita menuliskan class berbeda tanpa ada hubungan Inheritance, jika Anda ingin melakukan perubahan, Anda harus merubah isi dari semua class tersebut satu persatu.

BAB III – GUIDED

Guided 1

- **Source Code (Kendaraan.Java)**

```

//Bunga Laelatul Muna
// 21102010

package com.Bunga_Laelatul_Muna.PBO.Pertemuan4.Guided;

public class Kendaraan {
    String nama;
    int jmlRoda;

    public void nyalakanMesin() {
        System.out.println("Mesin " + nama + " telah
  
```

```

        dinyalakan");
    }
    public void showInfo() {
        System.out.println("Nama Kendaraan : " + nama);
        System.out.println("Jumlah Roda      : " + jmlRoda);
    }
    private void extraInfo(){

    }

}

```

- **Source Code (Mobil.java)**

```

//Bunga Laelatul Muna
//21102010
package com.Bunga_Laelatul_Muna.PBO.Pertemuan4.Guided;

public class Mobil extends Kendaraan {
    public void belok(String arah){
        System.out.println("Mobil " + nama + " belok ke "
+ arah + "!");
    }
}

```

- **Source Code (Main.Java)**

```

//Bunga Laelatul Muna
// 21102010
package com.Bunga_Laelatul_Muna.PBO.Pertemuan4.Guided;

public class Main {
    public static void main(String[] args) {
        //Membuat Objek
        Mobil mb = new Mobil();
        mb.nama = "Lamborghini";
        mb.jmlRoda = 4;
        mb.showInfo();
        mb.nyalakanMesin();
        mb.belok("kanan");

    }
}

```

- **SS Output**

```
Nama Kendaraan : Lamborgini
Jumlah Roda    : 4
Mesin Lamborgini telah dinyalakan
Mobil Lamborgini belok ke kanan!

Process finished with exit code 0
```

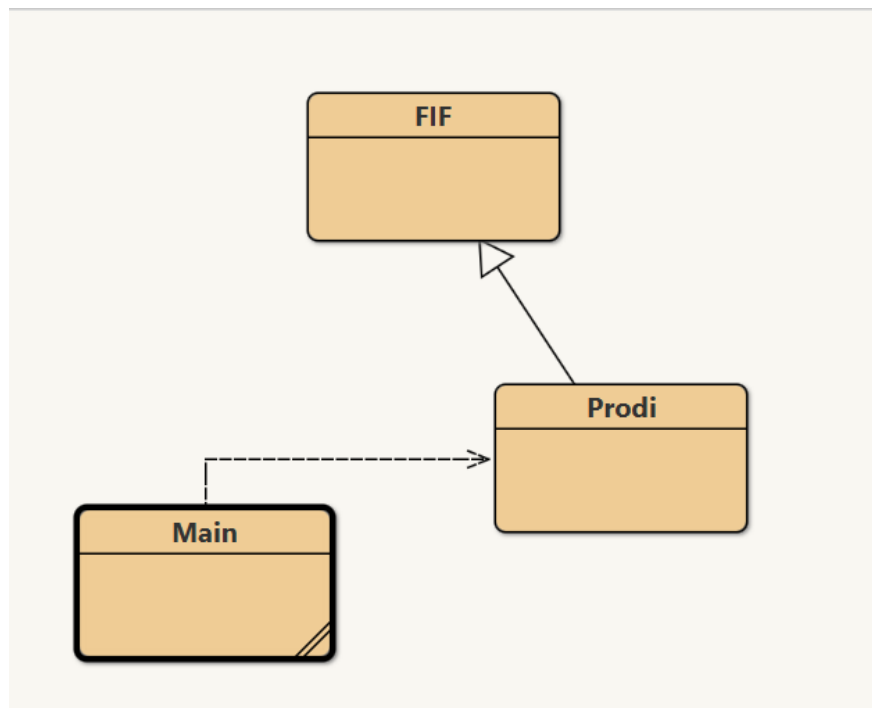
- **Penjelasan**

Ini Program menggunakan inheritance atau pewarisan. Disini kita membuat 3 class java, class kendaraan untuk inisialisais prosedurnya dan juga inialisasi data classnya. Class mobil adalah sebuah class anakan dari class Kendaraan(Extend). Di dalam class ini memuat sebuah inialisasi prosedur untuk arah mobil. Sedangkan class main digunakan untuk melakukan fungsi pemanggilan output dari program yang sudah kita buat.

BAB IV – UNGUIDED

UnGuided 1

- **Diagram**



- **Source Code(FIF.Java)**

```
//Bunga Laelatul Muna
// 21102010

package com.Bunga_Laelatul_Muna.PBO.Pertemuan4.Tugas;

public class FIF {
    private String namaFakultas;
    private String namaDekan;

    public void Fakultas(String namaFakultas, String
namaDekan) {
        this.namaFakultas = namaFakultas;
        this.namaDekan = namaDekan;
    }

    public void infoFakultas() {
        System.out.println("Fakultas " + namaFakultas);
        System.out.println("Dekan " + namaDekan);
    }
}
```

- **Source Code (Prodi.java)**

```
//Bunga Laelatul Muna
// 21102010
package com.Bunga_Laelatul_Muna.PBO.Pertemuan4.Tugas;

public class Prodi extends FIF{
    private String namaProdi;
    private int jumlahMahasiswa;

    public Prodi(String namaFakultas, String namaDekan,
String namaProdi, int jumlahMahasiswa) {
        Fakultas(namaFakultas, namaDekan);
        this.namaProdi = namaProdi;
        this.jumlahMahasiswa = jumlahMahasiswa;
    }

    public void infoProdi() {
        System.out.println("Program Studi " + namaProdi);
        System.out.println("Jumlah Mahasiswa " +
jumlahMahasiswa);
    }
}
```

- **Source Code (Main.Java)**

```
// 21102010
// Bunga Laelatul Muna

package com.Bunga_Laelatul_Muna.PBO.Pertemuan4.Tugas;

import java.util.Scanner;
```

```

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int pilihan;

        Prodi prodi1 = new Prodi("FIF", "Auliya
Burhanudin", "Teknik Informatika", 500);
        Prodi prodi2 = new Prodi("FTTE", "Iwan Setiawan",
"Teknik Elektro", 150);

        do {
            System.out.println("Pilih program studi yang
ingin dilihat informasinya:");
            System.out.println("1. FIF");
            System.out.println("2. FTTE");
            System.out.println("0. Keluar");
            System.out.print("Pilihan: ");
            pilihan = sc.nextInt();

            switch (pilihan) {
                case 1:
                    System.out.println();
                    prodi1.infoFakultas();
                    prodi1.infoProdi();
                    System.out.println();
                    break;
                case 2:
                    System.out.println();
                    prodi2.infoFakultas();
                    prodi2.infoProdi();
                    System.out.println();
                    break;
                case 0:
                    System.out.println("Program selesai");
                    break;
                default:
                    System.out.println("Pilihan tidak
tersedia");
                    break;
            }
        } while (pilihan != 0);
    }
}

```

- **SS Output**

```

Pilih program studi yang ingin dilihat informasinya:
1. FIF
2. FTTE
0. Keluar
Pilihan: 1

Fakultas FIF
Dekan Auliya Burhanudin
Program Studi Teknik Informatika
Jumlah Mahasiswa 500

```



```
Pilih program studi yang ingin dilihat informasinya:
1. FIF
2. FTTE
0. Keluar
Pilihan: 2

Fakultas FTTE
Dekan Iwan Setiawan
Program Studi Teknik Elektro
Jumlah Mahasiswa 150

Pilih program studi yang ingin dilihat informasinya:
1. FIF
2. FTTE
0. Keluar
Pilihan: 0
Program selesai

Process finished with exit code 0
```

- **Penjelasan**

Ini program Fakultas yang ingin mengaksesterkait informasi prodi. Disini dibuat sebuah Kode tersebut merupakan implementasi dari sebuah program yang menampilkan informasi program studi di suatu fakultas. Dalam program tersebut terdapat dua objek Prodi yang mewakili program studi Teknik Informatika dan Teknik Elektro. Setiap objek memiliki atribut seperti kode program studi, nama kaprodi, nama program studi, dan jumlah mahasiswa. Program ini memungkinkan pengguna untuk memilih program studi yang ingin dilihat informasinya melalui penggunaan perintah switch case pada blok do-while loop. Program akan terus berjalan hingga pengguna memilih opsi keluar dengan memilih angka 0.

A. REFERENSI

1. PetaniKode
2. Java Documentation
3. Modul 4