

## **LAPORAN PRAKTIKUM STRUKTUR DATA JOBSHEET 2 - OBJECT**

Nama: Wildan Dwi Triatna Kusuma

Kelas : TI-1C

Semester: 2

Mata Kuliah: Algoritma dan Struktur Data

Program Studi: Teknik Informatika

### **Pendahuluan**

Pada praktikum ini saya mempelajari konsep dasar pemrograman berorientasi objek menggunakan bahasa Java. Materi yang dipelajari meliputi pembuatan class, atribut, method, instansiasi object, serta penggunaan konstruktor. Praktikum ini membantu saya memahami bagaimana struktur program dibuat secara lebih terorganisir.

### **Percobaan 1 - Class Mahasiswa**

Kode Mahasiswa.java:

```

public class Mahasiswa {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa() {
    }

    Mahasiswa(String var1, String var2, String var3, double var4) {
        this.nim = var1;
        this.nama = var2;
        this.kelas = var3;
        this.updateIpk(var4);
    }

    void tampilkanInformasi() {
        System.out.println("NIM : " + this.nim);
        System.out.println("Nama : " + this.nama);
        System.out.println("Kelas : " + this.kelas);
        System.out.println("IPK : " + this.ipk);
        System.out.println("Nilai Kinerja: " + this.nilaiKinerja(this.ipk));
        System.out.println("-----");
    }

    void ubahKelas(String var1) {
        this.kelas = var1;
    }

    void updateIpk(double var1) {
        if (var1 >= 0.0 && var1 <= 4.0) {
            this.ipk = var1;
        } else {
            System.out.println("IPK tidak valid. Harus antara 0.0 dan 4.0");
        }
    }

    String nilaiKinerja(double var1) {
        if (var1 >= 3.5) {
            return "Sangat Baik";
        } else if (var1 >= 3.0) {
            return "Baik";
        } else {
            return var1 >= 2.0 ? "Cukup" : "Kurang";
        }
    }
}

```

Output:

```

wede@wedes-MacBook-Air Semester 2 % /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/wede/Library/Application\ Support/Antigravity/User/workspaceStorage/5ff73163fa54be7c908bca97cdea935a/redhat.java/jdt_ws/jdt.ls-java-project/bin Jobsheet2.MahasiswaMain
NIM : 123
Nama : Wildan
Kelas : TI-1A
IPK : 3.6
Nilai Kinerja: Sangat Baik
-----
NIM : 123
Nama : Wildan
Kelas : TI-1B
IPK : 3.2
Nilai Kinerja: Baik
-----
NIM : 456
Nama : Budi
Kelas : TI-2A
IPK : 3.9
Nilai Kinerja: Sangat Baik
-----
wede@wedes-MacBook-Air Semester 2 %

```

Jawaban Pertanyaan Percobaan 1:

1. Class memiliki atribut dan method sebagai karakteristik utama.
2. Atribut: nim, nama, kelas, ipk (4 atribut).
3. Method: tampilkanInformasi, ubahKelas, updateIpk, nilaiKinerja (4 method).
4. Method updateIpk divalidasi agar hanya menerima nilai 0.0 sampai 4.0.
5. Method nilaiKinerja menggunakan percabangan untuk menentukan kategori kinerja berdasarkan IPK dan mengembalikan nilai dalam bentuk String.

## Percobaan 2 - Instansiasi Object

Kode MahasiswaMain.java:

```
package Jobsheet2;

public class MahasiswaMain {
    public static void main(String[] args) {
        // buat objek pakai konstruktor default, data diisi manual
        Mahasiswa mhs1 = new Mahasiswa();
        mhs1.nim = "123";
        mhs1.nama = "Wildan";
        mhs1.kelas = "TI-1A";
        mhs1.updateIpk(3.6);

        mhs1.tampilkanInformasi();

        // coba ubah kelas dan update IPK
        mhs1.ubahKelas("TI-1B");
        mhs1.updateIpk(3.2);
        mhs1.tampilkanInformasi();

        // buat objek pakai konstruktor berparameter
        Mahasiswa mhs2 = new Mahasiswa("456", "Budi", "TI-2A", 3.9);
        mhs2.tampilkanInformasi();
    }
}
```

Contoh Output:

NIM: 123  
Nama: Wildan  
Kelas: TI-1A  
IPK: 3.6  
Kinerja: Sangat Baik

NIM: 123  
Nama: Wildan  
Kelas: TI-1B  
IPK: 3.2  
Kinerja: Baik

NIM: 456

Nama: Budi

Kelas: TI-2A

IPK: 3.9

Kinerja: Sangat Baik

Jawaban Pertanyaan Percobaan 2:

1. Instansiasi dilakukan dengan syntax: Mahasiswa mhs1 = new Mahasiswa();
2. Atribut dan method diakses menggunakan tanda titik (.).
3. Output berbeda karena data objek sudah diubah sebelum pemanggilan kedua.

Percobaan 3 - Konstruktor

Pada percobaan ini ditambahkan dua konstruktor yaitu konstruktor default dan konstruktor berparameter untuk menginisialisasi objek secara langsung.

Jawaban Pertanyaan Percobaan 3:

1. Konstruktor berparameter ditulis tanpa tipe return, contoh: Mahasiswa(String nim, String nama, String kelas, double ipk).
2. Instansiasi dengan konstruktor parameter langsung mengisi atribut saat objek dibuat.
3. Jika konstruktor default dihapus, maka pemanggilan new Mahasiswa() akan menyebabkan error karena tidak ada konstruktor yang sesuai.
4. Method tidak harus dipanggil berurutan, selama objek sudah dibuat.
5. Contoh object baru: Mahasiswa mhsWildan = new Mahasiswa("789", "Wildan", "TI-1B", 3.8);

Kesimpulan

Dari praktikum ini saya memahami konsep dasar object oriented programming seperti pembuatan class, atribut, method, instansiasi object, dan konstruktor. Konsep ini sangat penting sebagai dasar dalam pengembangan aplikasi berbasis Java.