

**LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**  
**JOBSHEET II : OBJECT**



Aylafada Syakira

TI-1C

254107020116

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG 2026**

## 2.1 Percobaan 1: Deklarasi Class, Atribut dan Method

### 2.1.1 Langkah-langkah Percobaan

Berikut hasil program percobaan 1:

```
package minggu2;

//class: template untuk membuat object
public class Mahasiswa06 {
    //atribut: data yang memiliki object
    String nama;
    String nim;
    String kelas;
    double ipk;

    //method/fungsi: digunakan object untuk berinteraksi dengan object yang lain
    void tampilkanInformasi () {
        System.out.println("Nama: " +nama);
        System.out.println("NIM: " +nim);
        System.out.println("IPK: " +ipk);
        System.out.println("Kelas: " +kelas);
    }

    void ubahKelas (String kelasBaru) {
        kelas = kelasBaru;
    }

    void updateIpk (double ipkBaru) {
        ipk = ipkBaru;
    }

    String nilaiKinerja() {
        if (ipk >= 3.5) {
            return "Kinerja sangat baik";
        } else if (ipk >= 3.0) {
            return "Kinerja baik";
        } else if (ipk >= 2.0) {
            return "Kinerja cukup";
        } else {
            return "Kinerja kurang";
        }
    }
}
```

### 2.1.3 Pertanyaan

1. Karakteristik Object: Mempunyai sesuatu (seperti data, properti, variabel, state, atribut) dan melakukan sesuatu
2. Atribut yang dimiliki oleh class Mahasiswa adalah nama, nim, kelas, dan ipk
3. Method yang dimiliki oleh class tersebut adalah tampilkanInformasi, ubahKelas, updateIpk, nilaiKinerja

4. Berikut hasil modifikasi updateIpk():

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

5. Kriteria yang digunakan untuk menentukan nilai kinerja adalah

- a. IPK > 3.5 "Kinerja sangat baik"  
IPK >= 3.0 "Kinerja baik"  
IPK >= 2.0 "Kinerja cukup"  
Selain itu "Kinerja kurang"
- b. Yang dikembalikan oleh method nilaiKinerja() adalah  
"Kinerja sangat baik"  
"Kinerja baik"  
"Kinerja cukup"  
"Kinerja kurang".

## 2.2 Percobaan 2: Instansiasi Object, serta Mengakses Atribut dan Method

### 2.2.1 Langkah-langkah Percobaan

Berikut hasil program percobaan 2:

```
//object  
Mahasiswa06 mhs1 = new Mahasiswa06();  
mhs1.nama = "Muhammad Ali Farhan";  
mhs1.nim = "2241720171";  
mhs1.kelas = "SI 2J";  
mhs1.ipk = 3.55;  
  
//data sebelum diubah  
mhs1.tampilkanInformasi();  
//memberikan nilai atribut dari object mhs1  
mhs1.ubahKelas(kelasBaru: "SI 2K");  
mhs1.updateIpk(ipkBaru: 3.60);  
//data sesudah diubah (menjalankan method dari object mhs1)  
mhs1.tampilkanInformasi();
```

### 2.2.3 Pertanyaan

- Berikut baris program yang digunakan untuk proses instansiasi:

```
Mahasiswa06 mhs2 = new Mahasiswa06 (nm: "Annisa Nabila", nim: "2141720160", ipk: 3.25, kls: "TI 2L");
mhs2.updateIpk (ipkBaru: 3.30);
mhs2.tampilkanInformasi();
```

Nama object yang dihasilkan adalah mhs2

- Cara mengakses atribut dan method dari suatu objek adalah dengan menggunakan operator titik. Berikut baris program yang digunakan untuk mengakses atribut dan method dari suatu objek:

```
mhs1.nama = "Muhammad Ali Farhan";
mhs1.nim = "2241720171";
mhs1.kelas = "SI 2J";
mhs1.ipk = 3.55;
```

- Hasil output pemanggilan method `tampilkanInformasi()` pertama dan kedua berbeda karena sebelum pemanggilan data kedua ada perubahan data object, karena jika state berubah maka output method yang membaca state itu juga ikut berubah

## 2.3 Percobaan 3: Membuat Konstruktor

### 2.3.1 Langkah-langkah Percobaan

- Berikut 2 buah konstruktor yang telah ditambahkan ke percobaan 1:

```
//konstruktor default
public Mahasiswa06(){
}

//konstruktor berparameter
public Mahasiswa06 (String nm, String nim, double ipk, String kls) {
    //digunakan untuk mengakses atribut
    nama = nm;
    this.nim = nim;
    this.ipk = ipk;
    kelas = kls;
}
```

- Berikut object yang telah ditambahkan dengan menggunakan konstruktor berparameter:

```
//instansiasi: pembuatan object dari class (new)
//konstruktor (istimewa): nama method sama dengan nama class, tdk ada tipe data
Mahasiswa06 mhs2 = new Mahasiswa06 (nm: "Annisa Nabila", nim: "2141720160", ipk: 3.25, kls: "TI 2L");
mhs2.updateIpk (ipkBaru: 3.30);
mhs2.tampilkanInformasi();
```

### 2.3.3 Pertanyaan

1. Berikut baris program yang digunakan untuk mendeklarasikan konstruktor berparameter:

```
public Mahasiswa06 (String nm, String nim, double ipk, String kls) {  
    nama = nm;  
    this.nim = nim;  
    this.ipk = ipk;  
    kelas = kls;  
}
```

2. Membuat object baru dari class mahasiswa dan menginisialisasi data
3. Maka akan terjadi error, object harus dibuat dengan parameter
4. Tidak, karena method dapat dipanggil kapan saja dan urutan pemanggilan hanya berpengaruh jika method tersebut mengubah data
5. Berikut object baru menggunakan konstruktor berparameter dari class Mahasiswa:

```
Mahasiswa06 mhsAyla = new Mahasiswa06 (nm: "Aylafada Syakira", nim: "2541070116", ipk: 3.95, kls: "TI 1C");  
mhsAyla.updateIpk (ipkBaru: 4.00);  
mhsAyla.tampilkanInformasi();
```