

**LAPORAN PERCOBAAN**

***JOBSHEET 2***



**Rafael Dimas Cahyo Laksono**

**254107020051**

**TI-1C / 23**

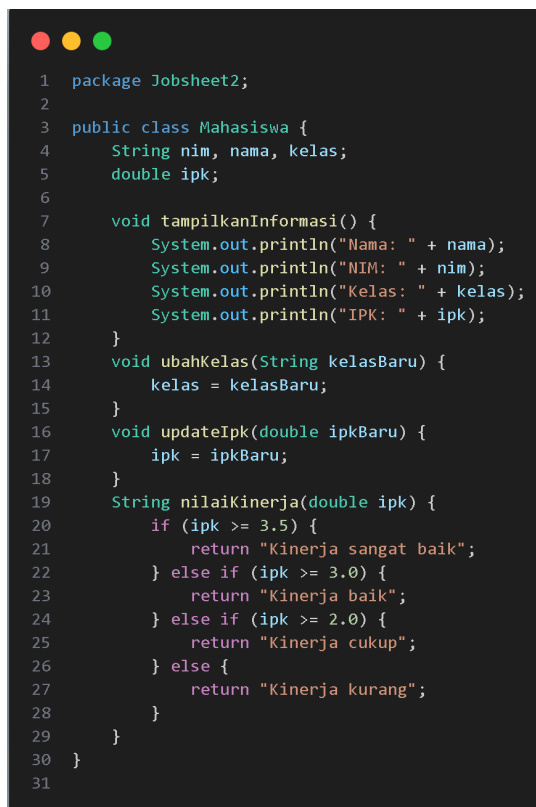
## 1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Mengenal objek dan class sebagai konsep mendasar pada pemrograman berorientasi objek
2. Mendeklarasikan class, atribut dan method
3. Membuat objek (instansiasi)
4. Mengakses atribut dan method dari suatu objek
5. Menerapkan konstruktor

## 2. Praktikum

### 2.1. Percobaan 1



```
1 package Jobsheet2;
2
3 public class Mahasiswa {
4     String nim, nama, kelas;
5     double ipk;
6
7     void tampilkanInformasi() {
8         System.out.println("Nama: " + nama);
9         System.out.println("NIM: " + nim);
10        System.out.println("Kelas: " + kelas);
11        System.out.println("IPK: " + ipk);
12    }
13    void ubahKelas(String kelasBaru) {
14        kelas = kelasBaru;
15    }
16    void updateIpk(double ipkBaru) {
17        ipk = ipkBaru;
18    }
19    String nilaiKinerja(double ipk) {
20        if (ipk >= 3.5) {
21            return "Kinerja sangat baik";
22        } else if (ipk >= 3.0) {
23            return "Kinerja baik";
24        } else if (ipk >= 2.0) {
25            return "Kinerja cukup";
26        } else {
27            return "Kinerja kurang";
28        }
29    }
30 }
31
```


Hasil: Error

### Pertanyaan

1. Karakteristik class bersifat umum atau sebuah template untuk membuat objek. Sedangkan karakteristik objek bersifat lebih spesifik yang dibentuk dari sebuah class.

2. Atribut yang terdapat pada class Mahasiswa ada 4, antara lain: nim, nama, kelas, dan ipk.
3. Method yang terdapat pada class Mahasiswa ada 4, antara lain: tampilkanInformasi, ubahKelas, updateIpk, nilaiKinerja.

```
1 void updateIpk(double ipkBaru) {
2     if (ipkBaru < 0.0 || ipkBaru > 4.0 ) {
3         System.out.println("IPK tidak valid. Harus antara 0.0 dan 4.0");
4     }
5     ipk = ipkBaru;
6 }
```

- 4.
5. Cara kerja dari method nilaiKerja adalah untuk menilai ipk sesuai dengan kondisi yang diberikan. Jika ipk lebih dari sama dengan 3.5 maka method akan mengembalikan / *return* pesan “Kinerja sangat baik”, begitupun seterusnya.
6. 

## 2.2. Percobaan 2

```
1 package Jobsheet2;
2
3 public class MahasiswaMain {
4     public static void main(String[] args) {
5         Mahasiswa mhs1 = new Mahasiswa();
6         mhs1.nama = "Muhammad Ali Farhan";
7         mhs1.nim = "2241720171";
8         mhs1.kelas = "SI 2J";
9         mhs1.ipk = 3.55;
10
11         mhs1.tampilkanInformasi();
12         mhs1.ubahKelas("SI 2K");
13         mhs1.updateIpk(3.60);
14         mhs1.tampilkanInformasi();
15     }
16 }
17
```

Hasil:

```
Nama: Muhammad Ali Farhan
NIM: 2241720171
Kelas: SI 2J
IPK: 3.55
Nama: Muhammad Ali Farhan
NIM: 2241720171
Kelas: SI 2K
IPK: 3.6
```

### Pertanyaan

1. Mahasiswa mhs1 = new Mahasiswa(); adalah baris kode yang digunakan untuk instansiasi. Objek yang dihasilkan dari proses instansiasi adalah mhs1.
2. Cara mengakses atribut dan method dari suatu objek dapat menggunakan format kode sebagai berikut: namaObjek.namaAtribut = nilai; dan namaObjek.namaMethod();
3. Hasil output pemanggilan method tampilkanInformasi() pertama dan kedua berbeda karena setelah pemanggilan method tampilkanInformasi() pertama kita melakukan update kelas dan IPK menggunakan method ubahKelas() dan method updateIpk() sehingga output dari pemanggilan method tampilkanInformasi() yang pertama dan kedua berbeda.

### 2.3. Percobaan 3

#### Kode main:

```
1 package Jobsheet2;
2
3 public class MahasiswaMain {
4     public static void main(String[] args) {
5         Mahasiswa mhs1 = new Mahasiswa();
6         mhs1.nama = "Muhammad Ali Farhan";
7         mhs1.nim = "2241720171";
8         mhs1.kelas = "SI 2J";
9         mhs1.ipk = 3.55;
10
11         mhs1.tampilkanInformasi();
12         mhs1.ubahKelas("SI 2K");
13         mhs1.updateIpk(3.60);
14         mhs1.tampilkanInformasi();
15
16         Mahasiswa mhs2 = new Mahasiswa("Annisa Nabila", "21417120160", 3.25, "TI 2L");
17         mhs2.updateIpk(3.30);
18         mhs2.tampilkanInformasi();
19     }
20 }
21
```

## Kode Class:

```
1 package Jobsheet2;
2
3 public class Mahasiswa {
4     String nim, nama, kelas;
5     double ipk;
6
7     Mahasiswa() {
8
9     }
10
11     Mahasiswa(String nm, String nim, double ipk, String kls) {
12         nama = nm;
13         this.nim = nim;
14         this.ipk = ipk;
15         kelas = kls;
16     }
17
18     void tampilkanInformasi() {
19         System.out.println("Nama: " + nama);
20         System.out.println("NIM: " + nim);
21         System.out.println("Kelas: " + kelas);
22         System.out.println("IPK: " + ipk);
23     }
24
25     void ubahKelas(String kelasBaru) {
26         kelas = kelasBaru;
27     }
28
29     void updateIpk(double ipkBaru) {
30         if (ipkBaru < 0.0 || ipkBaru > 4.0 ) {
31             System.out.println("IPK tidak valid. Harus antara 0.0 dan 4.0");
32         }
33         ipk = ipkBaru;
34     }
35
36     String nilaiKinerja(double ipk) {
37         if (ipk >= 3.5) {
38             return "Kinerja sangat baik";
39         } else if (ipk >= 3.0) {
40             return "Kinerja baik";
41         } else if (ipk >= 2.0) {
42             return "Kinerja cukup";
43         } else {
44             return "Kinerja kurang";
45         }
46     }
47 }
48
```

## Hasil:

Nama: Muhammad Ali Farhan

NIM: 2241720171

Kelas: SI 2J

IPK: 3.55

Nama: Muhammad Ali Farhan

NIM: 2241720171

Kelas: SI 2K

IPK: 3.6

Nama: Annisa Nabila

NIM: 21417120160

Kelas: TI 2L

IPK: 3.3

## Pertanyaan

1. Baris kode program yang digunakan untuk mendeklarasikan konstruktor berparameter adalah

```
1 Mahasiswa(String nm, String nim, double ipk, String kls) {  
2     nama = nm;  
3     this.nim = nim;  
4     this.ipk = ipk;  
5     kelas = kls;  
6 }
```

2. Pada baris program tersebut adalah instansiasi dengan menambahkan nilai pada parameter konstruktor sehingga kita tidak perlu menambahkan value satu-satu seperti pada konstruktor yang tidak memiliki parameter.
3. Program akan mengalami error. Umumnya, jika kita menuliskan konstruktor default pada proses instansiasi tanpa menuliskannya di class dapat terjadi karena compiler java otomatis menyediakan konstruktor default. Namun jika kita mendeklarasikan konstruktor dengan parameter di class, maka proses instansiasi dengan konstruktor default tanpa kita deklarasikan di class tidak dapat terjadi karena compiler java berhenti menyediakan konstruktor default
4. Tidak, karena masing-masing method memiliki fungsinya masing-masing dan tidak saling terikat sehingga dalam mengakses method tidak perlu berurutan.

5. 

```
1 Mahasiswa mhsRafaelDimas = new Mahasiswa("Rafael Dimas", "254107020051", 3.75, "TI 1C");
```