

Pertanyaan 2.1.3

1. Karakteristik Class : Sebuah template dan pembentuk struktur
Karakteristik Object : Memiliki nilai dan memiliki wujud nyata
2. Ada 4 atribut, yakni Nama, NIM, Kelas dan IPK.
3. Ada 4 Method, yakni tampilkanInformasi, ubahKelas, updateIPK, dan nilaiKinerja.
- 4.

```
String nilaiKinerja(){  
    if(ipk < 0.00 || ipk > 4.00){  
        return "IPK tidak valid. Harus antara 0.0 dan 4.0";  
    }  
    else 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";  
    }  
}
```

5. Cara Kerja Method

Method ini bekerja menggunakan struktur percabangan bersarang (**if-else if**). Proses evaluasinya berjalan secara sekuensial (dari atas ke bawah) dengan alur berikut:

1. **Validasi Input:** Hal pertama yang dicek adalah batasan logis nilai IPK. Program memastikan bahwa nilai tidak mungkin negatif atau lebih dari 4.0 menggunakan operator logika OR (**||**).
2. **Pengecekan Berurutan:** Jika IPK valid, program akan turun ke blok **else if** dan mengecek batas bawah nilai secara berurutan, dimulai dari yang paling tinggi (3.5), lalu 3.0, lalu 2.0.
3. **Eksekusi Langsung Berhenti:** Karena menggunakan struktur **else if** dan **return**, begitu program menemukan satu kondisi yang bernilai **true** (benar), program akan langsung mengirimkan teks tersebut keluar dari method dan mengabaikan sisa kondisi di bawahnya.

Kriteria Penilaian

Kriteria satu-satunya yang dievaluasi dalam method ini adalah besaran nilai pada variabel **ipk**. Berdasarkan struktur kodenya, rentang kriterianya terbagi menjadi:

- **Tidak Valid:** Jika **ipk** di bawah 0.0 atau di atas 4.0.

- **Sangat Baik:** Jika **ipk** bernilai 3.5 hingga batas maksimum 4.0.
- **Baik:** Jika **ipk** bernilai 3.0 hingga 3.49.
- **Cukup:** Jika **ipk** bernilai 2.0 hingga 2.99.
- **Kurang:** Jika **ipk** bernilai 0.0 hingga 1.99 (ditangkap oleh blok **else** terakhir karena semua kondisi di atasnya tidak terpenuhi).

Nilai yang Dikembalikan (Return Value)

Sesuai dengan deklarasi tipe data pada awal method (**String**), method ini mengembalikan **sebuah data berupa teks (String)**. Teks spesifik yang dikembalikan bergantung sepenuhnya pada kondisi **ipk** mana yang terpenuhi, entah itu berupa deskripsi kinerja seperti "**Kinerja Sangat Baik**" atau pesan peringatan "**IPK tidak valid. Harus antara 0.0 dan 4.0**".

Pertanyaan 2.2.3

1. Nama Objectnya adalah mhs1

```
public static void main(String[] args) {
    Mahasiswa19 mhs1 = new Mahasiswa19();
}
```

2. **Cara mengakses atribut:** **namaObjek.namaAtribut**
 - Contoh memberikan nilai: **mhs1.nama = "Naufal";**
 - Contoh mengambil nilai: **System.out.println(mhs1.nama);**

Cara mengakses method: **namaObjek.namaMethod()**

- Contoh memanggil method tanpa parameter:
mhs1.tampilkanInformasi();
- Contoh memanggil method dengan parameter:
mhs1.ubahKelas("TI-2B");

3. Hasil outputnya berbeda karena terjadi perubahan nilai atribut pada objek tersebut di antara pemanggilan pertama dan kedua.

Pertanyaan 2.3.3

1.

```
public Mahasiswa19(String nama, String nim, String kelas, double ipk){  
    this.nama = nama;  
    this.nim = nim;  
    this.kelas = kelas;  
    this.ipk = ipk;  
}
```

2. Memberikan nilai pada Atribut di suatu Objek.

3. Hasilnya program tidak dapat dieksekusi, karena hak pembuatan constructor otomatis dibelakang layar dicabut, karena sudah ada constructor parameter. Agar program dapat berjalan normal, antara nulis constructor default lagi atau membuat format instansiasi untuk constructor parameter.

4. Tidak, method di dalam sebuah class tidak harus diakses secara berurutan, karena dalam Pemrograman Berorientasi Objek (OOP), objek yang sudah diinstansiasi bersifat independen dan menyimpan *state* (nilai atributnya) sendiri di dalam memori.

5.

```
Mahasiswa19 mhsNopal = new Mahasiswa19(nama: "Nopal", nim: "2241720172", kelas: "SI-2J", ipk: 3.00);  
mhsNopal.updateIPK(ipkBaru: 3.10);  
mhsNopal.tampilkanInformasi();
```