

Nama : Fadila Mayamasita

Kelas : 1C_TI

Jawaban percobaan 1 : Deklarasi Class, Atribut dan Method

1. Karakteristik object : terdiri dari atribut, tipe data, variabel dan method
2. Atribut yang ada pada file Mahasiswa.13 yaitu Nama,Nim,Ipk dan kelas
3. Methodnya ada 4 yaitu tampilkan informasi, ubah kelas, update ipk dan menampilkan hasil nilai kinerja
4. Tampilannya akan seperti ini :

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

5. Cara kerja method Nilai kerja adalah :

Dengan membaca nilai variabel ipk. Yang kemudian dilakukan pengecekan menggunakan if- else if – else, lalu kondisi dicek dari nilai ipk yang tertinggi ke yang terendah. Jika salah satu kondisi terpenuhi, method lanngsung mereturn hasilnya dan berhenti. Kriteria yang di gunakan seperti ini

IPK	Kategori kinerja
≥ 3.5	Kinerja sangat baik
≥ 3.0	Kinerja baik
≥ 2.0	Kinerja cukup
Else	Kinerja kurang

Yang dikembalikan oleh Return adalah teks berupa kategori kinerja, yaitu salah satu dari ini "kinerja sangat baik", "kinerja baik", "kinerja cukup" dan "kinerja kurang"

JAWABAN PERCOBAAN 2 : Instansiasi Object, serta Mengakses Atribut dan Method

Jawaban 2.2.2

```
Nama: Muhammad Ali Farhan  
NIM: 2241720171  
IPK: 3.55  
Kelas: S1 2KJ  
Nama: Muhammad Ali Farhan  
NIM: 2241720171  
IPK: 3.6  
Kelas: S1 2K
```

Jawaban 2.2.3

1. Baris kode yang di instansiasi :

```
package MINGGU2;  
  
public class MahasiswaMain13 {  
    Run | Debug | Run main | Debug main  
    public static void main(String[] args) {  
        Mahasiswa13 mhs1 = new Mahasiswa13();  
        mhs1.nama = "Muhammad Ali Farhan";  
        mhs1.nim = "2241720171";  
        mhs1.kelas = "S1 2KJ";  
        mhs1.ipk = 3.55;  
  
        mhs1.tampilkanInformasi();  
        mhs1.ubahKelas(kelasBaru: "S1 2K");  
        mhs1.updateIpk(ipkBaru: 3.60);  
        mhs1.tampilkanInformasi();  
    }  
}
```

Pada class MahasiswaMain13, proses **instansiasi** terjadi pada baris

```
Mahasiswa13 mhs1 = new Mahasiswa13();
```

Nama object yang dihasilkan adalah **mhs1**.

2. Cara mengakses atribut dan method dari suatu objek adalah:

- Menggunakan operator titik (.)
- Format: namaObject.anggotaClass
- Bisa untuk membaca/mengubah atribut dan memanggil method

```
mhs1.nama = "Muhammad Ali Farhan";  
mhs1.nim = "2241720171";  
mhs1.kelas = "S1 2KJ";  
mhs1.ipk = 3.55;
```

3. Karena pemanggilan method `tampilkanInformasi()` pertama dan kedua berbeda karena terjadinya perubahan nilai atribut object sebelum pemanggilan kedua dilakukan. Pada method `ubahKelas` mengubah nilai atribut kelas method `updateIpk` yang dimana mengubah nilai atribut ipk. Artinya kondisi data dari object sudah berubah.

JAWABAN PERCOBAAN 3 : Membuat Konstruktor

Jawaban 2.3.2

```
Main13
Nama: Muhammad Ali Farhan
NIM: 2241720171
IPK: 3.55
Kelas: S1 2KJ
Nama: Muhammad Ali Farhan
NIM: 2241720171
IPK: 3.6
Kelas: S1 2K
Nama: Annisa Nabila
NIM: 2141720160
IPK: 3.3
Kelas: TI 2L
```

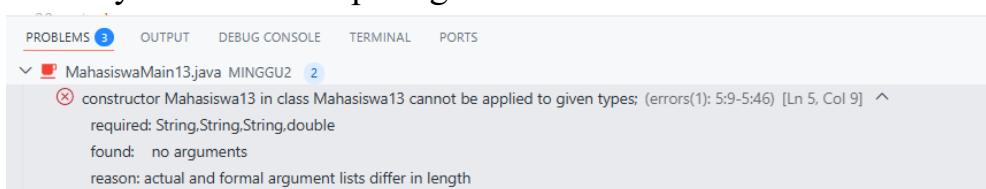
Jawaban 2.3.3

1. baris kode program yang digunakan untuk mendeklarasikan konstruktor berparameter

```
public Mahasiswa13(String nm, String nim, String kelas, double ipk) {
    this.nama = nm;
    this.nim = nim;
    this.kelas = kelas;
    this.ipk = ipk;
}
```

2. Baris Mahasiswa13 mhs2 = new Mahasiswa13, digunakan untuk membuat objek baru bernama **mhs2** dari class Mahasiswa13 sekaligus langsung mengisi nilai atributnya melalui constructor berparameter. Artinya, saat objek dibuat, data seperti nama, nim, kelas, dan ipk langsung diinisialisasi tanpa perlu diisi satu per satu seperti pada mhs1. Jadi, baris tersebut melakukan proses instansiasi dan inisialisasi data dalam satu langkah.

3. Hasilnya akan error seperti gambar di bawah ini :



Karena setelah konstruktor default dihapus, class Mahasiswa13 hanya memiliki konstruktor berparameter

4. Setelah dilakukannya instansi object, method mahasiswa di dalam class tidak harus diakses secara urut. Method dapat dipanggil dalam urutan apa pun sesuai kebutuhan program, karena setiap method berdiri sendiri dan akan dijalankan hanya saat dipanggil. Java tidak mewajibkan pemanggilan method mengikuti urutan tertentu seperti saat method ditulis di dalam class.

5.

```
Mahasiswa13 mhsFadilaMayamasita =  
    new Mahasiswa13(nm: "Fadila Mayamasita", nim: "254107020012", kelas: "TI 1C", ipk: 3.23);  
  
    mhsFadilaMayamasita.tampilkanInformasi();  
    System.out.println("Nilai Kinerja: " + mhsFadilaMayamasita.nilaiKinerja());
```