

# JOBSHEET 2 OBJECT

Nama:Daffa Dwi Nugroho

Kelas:TI 1C

## 2.1

```
PS C:\Users\Daffa Dwi\OneDrive\Documents\SMT 2\PraktikumASD\MINGGU 2> java mahasiswa
Error: Main method not found in class mahasiswa, please define the main method as:
    public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
PS C:\Users\Daffa Dwi\OneDrive\Documents\SMT 2\PraktikumASD\MINGGU 2>
```

```
1 class mahasiswa{
2     String nama;
3     String nim;
4     String 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(){
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
```

### 1. Jawab:

- Untuk karakteristik class bersifat umum dan berupa rancangan/template
- Untuk karakteristik object berifat spesifik dan berupa objek nyata

### 2. Ada 4 atribut yaitu nama, nim, kelas, ipk

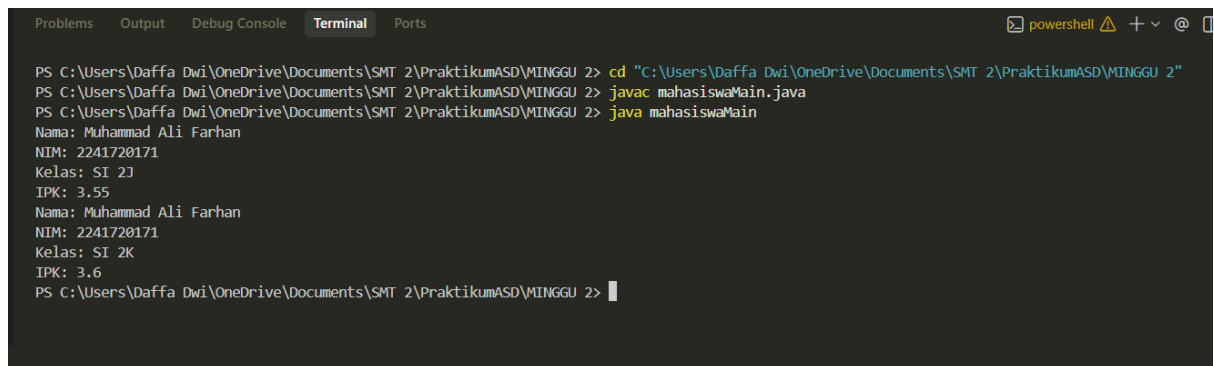
### 3. Ada 3 Method yaitu tampilkanInformasi,ubahKelas,updateIpk

### 4.

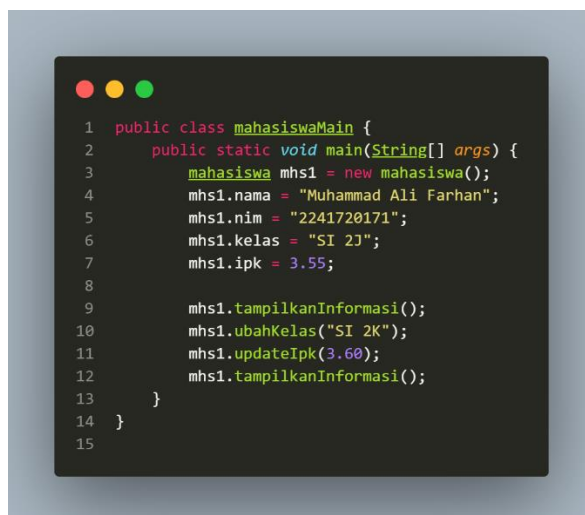
```
1 class mahasiswa{
2     String nama;
3     String nim;
4     String kelas;
5     double ipk;
6
7     void tampilkanInformasi(){
8         System.out.println("Nama: " + nama);
9         System.out.println("NIM: " + nim);
10        System.out.println("Kelas: " + kelas);
11        System.out.println("IPK: " + ipk);
12    }
13    void ubahKelas(String kelasBaru){
14        kelas = kelasBaru;
15    }
16    void updateIpk(double ipkBaru){
17        if (ipkBaru >= 0.0 && ipkBaru <= 4.0) {
18            ipk = ipkBaru;
19        } else {
20            System.out.println("IPK tidak valid harus antara 0.0 dan 4.0");
21        }
22    }
23    String nilaiKinerja(){
24        if (ipk >= 3.5) {
25            return "Kinerja Sangat Baik";
26        } else if (ipk >= 3.0) {
27            return "Kinerja Baik";
28        } else if (ipk >= 2.0) {
29            return "Kinerja Cukup";
30        } else {
31            return "Kinerja Kurang";
32        }
33    }
34 }
35
```

5. Method nilai kinerja digunakan untuk mengevaluasi kinerja mahasiswa berdasarkan ipk nya menggunakan struktur percabangan if else if else if else, diawali dengan method mengecek apakah nilai ipk sama dengan 3.5 atau lebih jika iya maka akan menghasilkan kinerja sangat baik, jika tidak mencapai 3.5 maka akan dilanjutkan ke percabangan selanjutnya hingga selesai, percabangan ini diawali dari terbesar dikarenakan jika dari terkecil semua nilai ipk akan masuk ke kondisi yang pertama yang mengakibatkan kategori yang lebih tinggi tidak akan tercapai

## 2.2



```
PS C:\Users\Daffa Dwi\OneDrive\Documents\SMT 2\PraktikumASD\MINGGU 2> cd "C:\Users\Daffa Dwi\OneDrive\Documents\SMT 2\PraktikumASD\MINGGU 2"
PS C:\Users\Daffa Dwi\OneDrive\Documents\SMT 2\PraktikumASD\MINGGU 2> javac mahasiswaMain.java
PS C:\Users\Daffa Dwi\OneDrive\Documents\SMT 2\PraktikumASD\MINGGU 2> java mahasiswaMain
Nama: Muhammad Ali Farhan
NIM: 2241720171
Kelas: SI 2J
IPK: 3.55
Nama: Muhammad Ali Farhan
NIM: 2241720171
Kelas: SI 2K
IPK: 3.6
PS C:\Users\Daffa Dwi\OneDrive\Documents\SMT 2\PraktikumASD\MINGGU 2>
```



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

1. Instansiasi nya di baris mahasiswa mhs1 = new mahasiswa(); dan nama objek yang dihasilkan ialah mhs1
2. Untuk mengakses atribut menggunakan format namaObjek.namaAtribut seperti di code ialah mhs1.ipk untuk mengakses method menggunakan format namaObjek.namaMethod seperti di code ialah mhs1.updatelpk
3. Karena pada pemanggilan tampilkan informasi kedua terdapat perubahan data yaitu pada kelas dan ipk nya

## 2.3

```

PS C:\Users\Daffa Dwi\OneDrive\Documents\SMT 2\PraktikumASD\MINGGU 2> & 'C:\Program Files\Java\jdk-24\bin\java.exe'
  -cp 'C:\Users\Daffa Dwi\AppData\Roaming\Windsurf\User\workspaceStorage\c2c9e996c8d68f486e598ee4e
  GU 2_e2399dc3\bin' 'mahasiswaMain'
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: 2141720160
Kelas: TI 2L
IPK: 3.3
PS C:\Users\Daffa Dwi\OneDrive\Documents\SMT 2\PraktikumASD\MINGGU 2>

```

```

1 public class mahasiswaMain {
2     public static void main(String[] args) {
3         mahasiswa mhs1 = new mahasiswa("Muhammad Ali Farhan", "2241720171", 3.55, "SI 2J");
4
5         mhs1.tampilkanInformasi();
6         mhs1.ubahKelas("SI 2K");
7         mhs1.updateIpk(3.60);
8         mhs1.tampilkanInformasi();
9         mahasiswa mhs2 = new mahasiswa("Annisa Nabila", "2141720160", 3.25, "TI 2L");
10        mhs2.updateIpk(3.30);
11        mhs2.tampilkanInformasi();
12    }
13 }
14

```

1. Pada barisan `public mahasiswa(String nm, String nim, double ipk, String kls){`  
`nama = nm;`  
`this.nim = nim;`  
`this.ipk = ipk;`  
`kelas = kls;`
2. Barisan tersebut melakukan instansiasi object dan menginialisasi atribut secara langsung
3. Jika konstruktor default dihapus dari class Mahasiswa, maka program tetap bisa dijalankan selama di dalam main menggunakan konstruktor berparameter. Namun, jika di dalam main masih menggunakan `new mahasiswa()` tanpa parameter, maka akan terjadi error saat compile karena Java tidak menemukan konstruktor yang sesuai.
4. Tidak, Setelah melakukan instansiasi objek, method di dalam class Mahasiswa tidak harus diakses secara berurutan sesuai urutan penulisannya di dalam class. Method dapat dipanggil dalam urutan apa pun sesuai kebutuhan program, selama objeknya sudah dibuat. Urutan penulisan method di dalam class tidak mempengaruhi urutan pemanggilan saat program dijalankan.