

Nama: Arjun Zain Regeng Pangestu

Nim: 254107020144

Kelas: 1C

2.1.3 Pertanyaan

1. Sebutkan dan jelaskan karakteristik class atau object!

Class memiliki karakteristik berikut:

1. Memiliki atribut (properties / variabel)
yaitu data yang menggambarkan keadaan sebuah objek.
2. Memiliki method (perilaku / fungsi)
yaitu tindakan atau operasi yang dapat dilakukan oleh objek.
3. Merupakan blueprint / cetakan objek
class tidak menyimpan data sebenarnya, namun hanya struktur.
4. Objek (instance) terbuat dari class
objek berisi data yang nyata sesuai dengan atribut class.

Object memiliki karakteristik:

1. Memiliki nilai atribut yang nyata (bukan hanya struktur)
2. Dapat menjalankan method yang dimiliki oleh class
3. Merepresentasikan sesuatu yang nyata

2. Pada class Mahasiswa, ada berapa atribut? Sebutkan!

Di jobsheet, class Mahasiswa memiliki 4 atribut, yaitu:

- String nama
- String nim
- double ipk
- String kelas

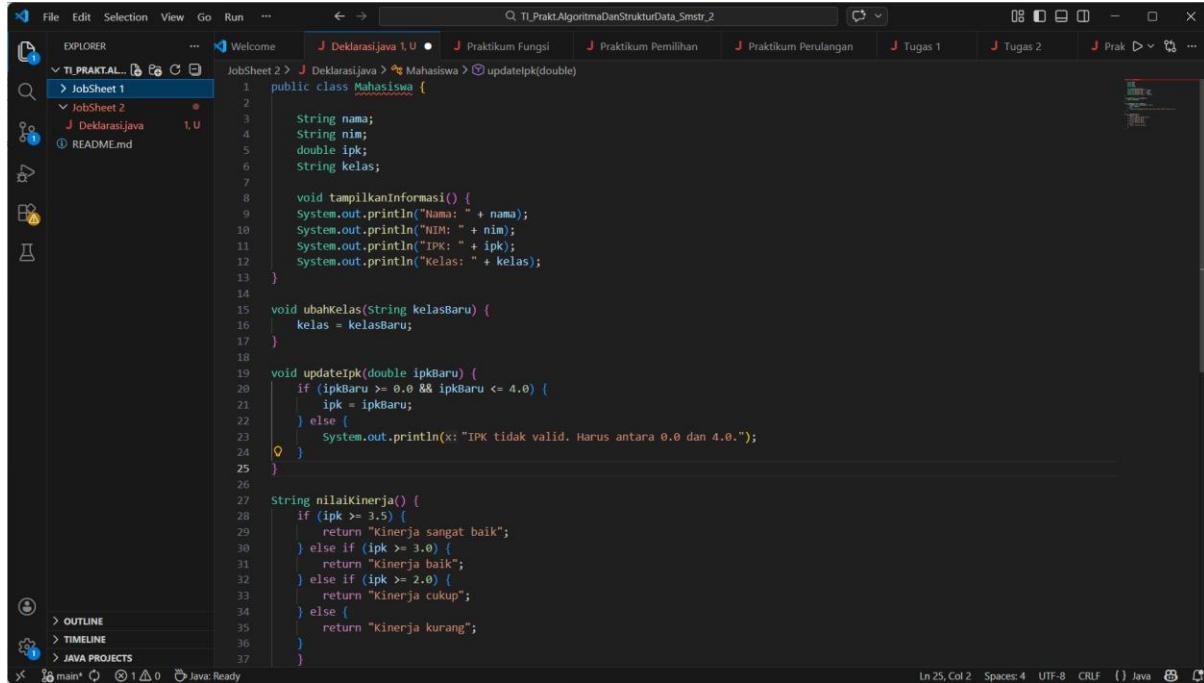
3. Ada berapa method yang dimiliki class tersebut? Sebutkan!

Class Mahasiswa memiliki 4 method, yaitu:

- tampilanInformasi()
- ubahKelas(String kelasBaru)
- updateIpk(double ipkBaru)

- nilaiKinerja()

4. Perintah: “Modifikasi method updateIpk() sehingga sebelum IPK diubah, dilakukan pengecekan apakah IPK valid (0.0 – 4.0). Jika tidak valid, tampilkan pesan error.”



```

File Edit Selection View Go Run ... ← → Q TI_Prakt.AlgoritmaDanStrukturData_Smstr_2
EXPLORER Welcome J DeklarasiJava 1, U ● J Praktikum Fungsi | J Praktikum Pemilihan | J Praktikum Perulangan | J Tugas 1 | J Tugas 2 | J Prak ...
JobSheet 2 > J DeklarasiJava > Mahasiswa > updateIpk(double)
1 public class Mahasiswa {
2
3     String nama;
4     String nim;
5     double ipk;
6     String kelas;
7
8     void tampilkanInformasi() {
9         System.out.println("Nama: " + nama);
10    System.out.println("NIM: " + nim);
11    System.out.println("IPK: " + ipk);
12    System.out.println("Kelas: " + kelas);
13 }
14
15 void ubahKelas(String kelasBaru) {
16     kelas = kelasBaru;
17 }
18
19 void updateIpk(double ipkBaru) {
20     if (ipkBaru >= 0.0 && ipkBaru <= 4.0) {
21         ipk = ipkBaru;
22     } else {
23         System.out.println("IPK tidak valid. Harus antara 0.0 dan 4.0.");
24     }
25 }
26
27 String nilaiKinerja() {
28     if (ipk >= 3.5) {
29         return "Kinerja sangat baik";
30     } else if (ipk >= 3.0) {
31         return "Kinerja baik";
32     } else if (ipk >= 2.0) {
33         return "Kinerja cukup";
34     } else {
35         return "Kinerja kurang";
36     }
37 }

```

Ln 25, Col 2 Spaces: 4 UTF-8 CRLF {} Java

5. Jelaskan cara kerja method nilaiKinerja()

Method ini digunakan untuk mengevaluasi kinerja mahasiswa berdasarkan IPK.

Cara kerjanya:

1. Method membaca nilai ipk pada objek mahasiswa.

2. Method membandingkan IPK dengan batas nilai:

- $IPK \geq 3.5 \rightarrow$ kinerja sangat baik
- $IPK \geq 3.0 \rightarrow$ kinerja baik
- $IPK \geq 2.0 \rightarrow$ kinerja cukup
- $IPK < 2.0 \rightarrow$ kinerja kurang

3. Method mengembalikan (return) String sesuai dengan kategori.

Method tersebut tidak menampilkan output, tetapi mengembalikan nilai agar bisa digunakan di tempat lain.

2.2.3 Pertanyaan

1. Pada class MahasiswaMain, tunjukkan baris kode program yang digunakan untuk proses instansiasi! Apa nama object yang dihasilkan?

- Baris instansiasi:

Mahasiswa mhs1 = new Mahasiswa();

- Nama object yang dihasilkan:

mhs1

2. Bagaimana cara mengakses atribut dan method dari suatu objek?

- Cara mengakses atribut:

Menggunakan: NamaObjek.namaAtribut

- Cara mengakses method:

Menggunakan: NamaObjek.namaMethod()

3. Mengapa hasil output pemanggilan method tampilanInformasi() pertama dan kedua berbeda?

Karena nilai atribut objek mhs1 berubah sebelum pemanggilan yang kedua dilakukan.

Perubahan yang terjadi:

1. Kelas diubah:

mhs1.ubahKelas("SI 2K");

2. IPK diperbarui:

mhs1.updateIpk(3.60);

Akibatnya:

Pemanggilan tampilanInformasi() pertama akan menampilkan data awal sedangkan pemanggilan tampilanInformasi() kedua menampilkan data setelah diubah.

2.3.3 Pertanyaan

1. Pada class Mahasiswa di Percobaan 3, tunjukkan baris kode program yang digunakan untuk mendeklarasikan konstruktor berparameter!

```
public Mahasiswa05(String namaBaru, String nimBaru, double ipkBaru, String kelasBaru) {
    nama = namaBaru;
    nim = nimBaru;
    ipk = ipkBaru;
    kelas = kelasBaru;}
```

2. Perhatikan class MahasiswaMain. Apa sebenarnya yang dilakukan pada baris program berikut?

```
Mahasiswa mhs2 = new Mahasiswa("Annisa Nabila", "2141720160", 3.25, "TI 2I");
```

Kegunaan baris program tersebut adalah:

- Membuat (instansiasi) objek baru bernama mhs2
- Mengisi atribut objek itu berdasarkan konstruktor berparameter:
 1. nama = "Annisa Nabila"
 2. nim = "2141720160"
 3. ipk = 3.25
 4. kelas = "TI 2I"
- Semua nilai akan langsung tersimpan di dalam object mhs2

3. Hapus konstruktor default pada class Mahasiswa, kemudian compile dan run program. Bagaimana hasilnya? Jelaskan mengapa hasilnya demikian!

Error, karena Java tidak secara otomatis membuat konstruktor default apabila kita sudah menuliskan konstruktor berparameter. Akibatnya, objek yang dibuat tanpa parameter akan menyebabkan error karena konstruktor default tersebut tidak tersedia.

4. Setelah melakukan instansiasi object, apakah method di dalam class Mahasiswa harus diakses secara berurutan? Jelaskan alasannya!

Pemanggilan method pada sebuah objek tidak harus dilakukan secara berurutan, karena setiap method bekerja secara mandiri. Hal yang terpenting adalah objek tersebut sudah dibuat sebelum method dipanggil.

5. Buat object baru dengan nama mhs<NamaMahasiswa> menggunakan konstruktor berparameter dari class Mahasiswa!

The screenshot shows a Java IDE interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows projects like "JI_PRAK_ALGORITMADA_05", "JobSheet2", "JobSheet1", "JobSheet2", "Mahasiswa05.java", "README.md", "MahasiswaMain05.java", "Praktikum Fungsi", "Praktikum Pemilihan", "Praktikum Perulangan", "Tugas 1", and "Tugas 2".
- Code Editor:** Displays the "MahasiswaMain05.java" file with the following code:

```
public class MahasiswaMain05 {
    public static void main(String[] args) {
        Mahasiswa05 mhs1 = new Mahasiswa05();
        mhs1.nama = "Muhammad Ali Farhan";
        mhs1.nim = "2141720173";
        mhs1.kelas = "SI 21";
        mhs1.ipk = 3.55;

        mhs1.tampilkanInformasi();
        mhs1.setLulusKelas("kelasBaru: SI 2K");
        mhs1.updateIpk(3.60);
        mhs1.tampilkanInformasi();

        Mahasiswa05 mhs2 = new Mahasiswa05(namaBaru: "Annisa Habila", nimBaru: "2141720160", ipkBaru: 3.25, kelasBaru: "TI 2L");
        mhs2.updateIpk(3.30);
        mhs2.tampilkanInformasi();

        Mahasiswa05 mhs3 = new Mahasiswa05(namaBaru: "Bagus Saputra", nimBaru: "254107021155", ipkBaru: 3.67, kelasBaru: "TI 1C");
        mhs3.tampilkanInformasi();
    }
}
```
- Bottom Bar:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS.
- Terminal:** PowerShell window showing command "Run: MahasiswaMain05".
- Status Bar:** Line 16, Col 35, Spaces: 4, UTF-8, CR/LF, Java.