

LAPORAN PRAKTIKUM ALGORITMA & STRUKTUR DATA

JOBSHEET 2: OBJECT



Disusun oleh:

Naura Fadhilla Aditya Putri (254107020007)

Kelas : TI -1C

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

TAHUN AJARAN

2026

2.1 Percobaan 1: Deklarasi Class, Atribut dan Method

- a. Berikut Adalah kode dari praktikum percobaan 1

The screenshot shows a Java development environment with the following details:

- Title Bar:** Minggu2
- Project Explorer:** Shows a file named "Mahasiswa20.java" under a package "Java > Mahasiswa20".
- Code Editor:** Displays the following Java code:

```
1 public class Mahasiswa20 {
2     String nama;
3     String nim;
4     String kelas;
5     double ipk;
6
7     void tampilanInfo() {
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
14     void ubahKelas(String KelasBaru) {
15         kelas = KelasBaru;
16     }
17
18     void updateIPK(double IPKBaru) {
19         ipk = IPKBaru;
20     }
21
22     String nilaiKinerja() {
23         if (ipk >= 3.5) {
24             return "Kinerja sangat baik";
25         } else if (ipk >= 3.0) {
26             return "Kinerja baik";
27         } else if (ipk >= 2.5) {
28             return "Kinerja cukup";
29         } else {
30             return "Kinerja kurang";
31         }
32     }
33 }
34 }
```
- Debugger Window:** A modal dialog box displays an error message: "Error: Main method not found in the file, please define the main method as: public static void main(String[] args)".
- Source:** Debugger for Java

- b. Terdapat hasil compile kode program beru eror yang muncul pada gambar diatas, dikarenakan file Mahasiswa20.java tidak memiliki method main, sehingga program tidak tahu titik awal untuk dijalankan.
- c. Berikut ialah jawaban dari pertanyaan:
1. Dua karakteristik class atau object yaitu Mempunyai Sesuatu (data, properti, atribut) dan dapat melakukan sesuatu (Tingkah laku, fungsi, main)
 2. Berikut ialah atribut yang dimiliki pada class Mahasiswa20.java:
String nama, String NIM, String kelas, dan double ipk

```
✓ public class Mahasiswa20 {  
    String nama;  
    String nim;  
    String kelas;  
    double ipk;
```

3. Ada 4 method yang ada di class:

- `tampilkanInfo()`

Menampilkan informasi mahasiswa (nama, NIM, kelas, IPK).

- `ubahKelas(String kelasBaru)`

Mengubah atribut kelas sesuai nilai baru yang diberikan.

- `updateIPK(double ipkBaru)`

Mengubah atribut ipk sesuai nilai baru yang diberikan.

- `nilaiKinerja()`

Mengembalikan string penilaian kinerja berdasarkan IPK:

$IPK \geq 3.5 \rightarrow \text{"Kinerja sangat baik"}$

$IPK \geq 3.0 \rightarrow \text{"Kinerja baik"}$

$IPK \geq 2.5 \rightarrow \text{"Kinerja cukup"}$

$IPK < 2.5 \rightarrow \text{"Kinerja kurang"}$

4. Berikut hasil dari modifikasi IPK yang dimasukkan di dalam rentang 0.0 sampai dengan 4.0 ($0.0 \leq IPK \leq 4.0$). Jika IPK tidak pada rentang tersebut maka dikeluarkan pesan: "IPK tidak valid. Harus antara 0.0 dan 4.0".

```
void updateIPK(double IPKBaru) {  
    if (ipk >= 0.0 && ipk >= 4.00 ) {  
        ipk = IPKBaru;  
    } else {  
        System.out.println("IPK tidak valid. Harus antara 0.0 dan 4.0.");  
    }  
}
```

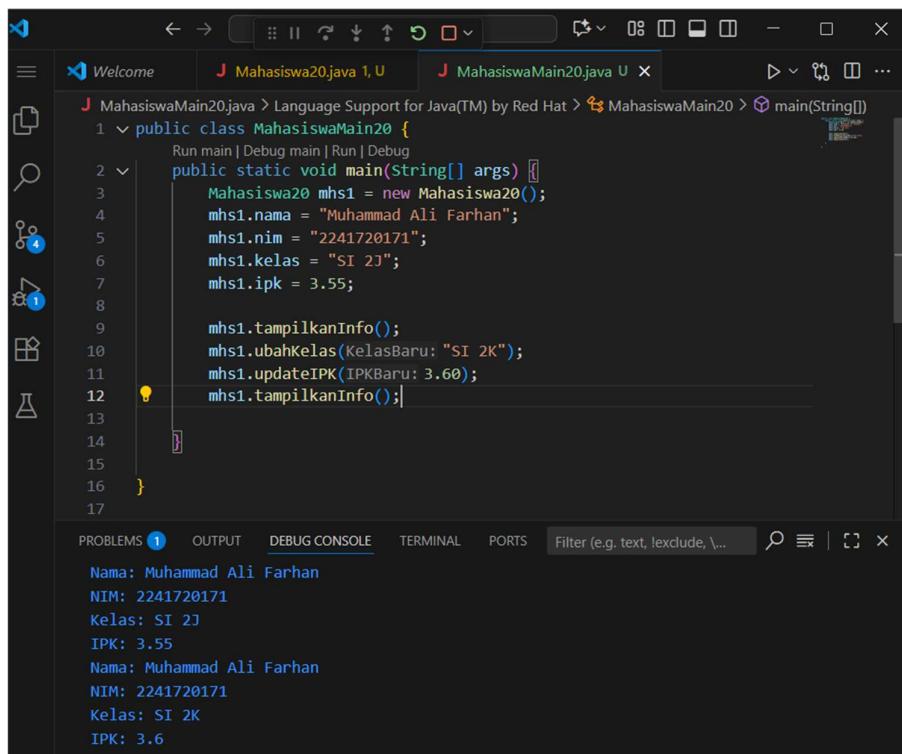
```
IPK tidak valid. Harus antara 0.0 dan 4.0.  
Nama: Annisa Nabila  
NIM: 2141720160  
Kelas: TI 2L  
IPK: 3.25
```

Keterangan tidak valid dikarenakan saya mengganti angka ipk sebesar 6.0 pada mhs2.updateIpk sehingga mengeluarkan output seperti pada gambar.

5. Method nilaiKinerja() berfungsi untuk mengevaluasi kinerja akademik mahasiswa berdasarkan nilai IPK yang dimilikinya. Pertama, method membaca nilai IPK dari objek mahasiswa. Kemudian dilakukan pengecekan berurutan: jika IPK berada pada rentang 3.5–4.0 maka kinerja dinilai sangat baik, jika berada pada rentang 3.0–3.49 maka kinerjanya baik, jika berada pada rentang 2.5–2.99 maka kinerjanya cukup, dan jika kurang dari 2.5 maka kinerjanya dianggap kurang. Dengan demikian, kriteria yang digunakan adalah rentang IPK yang dibagi menjadi empat kategori utama: sangat baik, baik, cukup, dan kurang. String tersebut dikembalikan (return) oleh method nilaiKinerja() sehingga dapat ditampilkan pada program.

2.2 Percobaan 2: Instansiasi Object, serta Mengakses Atribut dan Method

- a. Berikut Adalah hasil program dari percobaan 2:



The screenshot shows a Java development environment with two tabs open: 'Mahasiswa20.java' and 'MahasiswaMain20.java'. The 'MahasiswaMain20.java' tab contains the following code:

```
1 public class MahasiswaMain20 {
2     public static void main(String[] args) {
3         Mahasiswa20 mhs1 = new Mahasiswa20();
4         mhs1.nama = "Muhammad Ali Farhan";
5         mhs1.nim = "2241720171";
6         mhs1.kelas = "SI 2J";
7         mhs1.ipk = 3.55;
8
9         mhs1.tampilanInfo();
10        mhs1.ubahKelas(KelasBaru: "SI 2K");
11        mhs1.updateIPK(IPKBaru: 3.60);
12        mhs1.tampilanInfo();
13    }
14 }
```

The 'OUTPUT' tab shows the following console output:

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

- b. Berikut ialah jawaban dari pertanyaan:

1. Baris kode program yang digunakan untuk proses instansiasi

```
Mahasiswa20 mhs1 = new Mahasiswa20();
```

Objek yang dihasilkan yaitu mhs1.

2. Cara mengakses atribut dan method yaitu menggunakan operator titik(.) setelah nama objek. Seperti namaObjek.namaMethod(paramenter);

```
mhs1.tampilkanInfo();
mhs1.ubahKelas(kelasBaru: "SI 2K");
mhs1.updateIPK(IPKBaru: 3.60);
mhs1.tampilkanInfo();|
```

Objek memanggil method untuk menampilkan info, mengubah kelas, dan memperbarui IPK.

3. Hasil Output pemanggilan method tampilkanInfo() pertama dan kedua berbeda karena atribut objek telah diubah di antara dua pemanggilan tersebut. Jadi, perbedaan output terjadi karena objek yang sama (mhs1) mengalami perubahan nilai atribut sebelum pemanggilan kedua, sehingga informasi yang ditampilkan sudah sesuai dengan kondisi terbaru.

2.3 Percobaan 3: Membuat Konstruktor

- a. Berikut Adalah hasil program dari percobaan 3:

The screenshot shows a Java code editor with a code snippet and its terminal output. The code defines a class Mahasiswa20 with a constructor that initializes attributes: nama, nim, kelas, and ipk. It also contains methods tampilkanInfo(), ubahKelas(), and updateIPK(). The terminal output shows two instances of the class being created and their details being printed.

```
1 MahasiswaMain20.java > ...
  public class MahasiswaMain20 {
    Run main | Debug main | Run | Debug
  2   public static void main(String[] args) {
  3     Mahasiswa20 mhs1 = new Mahasiswa20();
  4     mhs1.nama = "Muhammad Ali Farhan";
  5     mhs1.nim = "2241720171";
  6     mhs1.kelas = "SI 2J";
  7     mhs1.ipk = 3.55;
  8
  9     mhs1.tampilkanInfo();
 10    mhs1.ubahKelas(kelasBaru: "SI 2K");
 11    mhs1.updateIPK(IPKBaru: 3.60);
 12    mhs1.tampilkanInfo();
 13
 14    Mahasiswa20 mhs2 = new Mahasiswa20(nama: "Annisa Nabila", nim: "2141720160", kelas: "TI 2L", ipk: 3.25 );
 15    mhs2.updateIPK(IPKBaru: 3.30);
 16    mhs2.tampilkanInfo();
 17  }
 18
 19 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
dhat.java\jdt_ws\Minggu2_98c2118c\bin' 'MahasiswaMain20'
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\hp\Documents\TUGAS KUYEAH\Praktikum ASD\PraktikumASD\Minggu2>
```

b. Berikut Adalah jawaban dari pertanyaan:

1. Baris kode program yang mendeklarasi konstruktor berparameter

```
public Mahasiswa20(String nama, String nim, String kelas, double ipk) {  
    this.nama = nama; // Konstruktor Parameter  
    this.nim = nim; // menggunakan this untuk menyimpan nilai parameter ke atribut kelas  
    this.kelas = kelas;  
    this.ipk = ipk;  
}
```

2. Pada baris program tersebut berfungsi untuk mengakses konstruktor berparamenter dengan object mhs2, untuk mengisi atau menginput ke konstruktor.
3. Berikut hasil dari Hapus konstruktor default pada class Mahasiswa yang menghasilkan eror, karena konstruktor tanpa parameter sudah tidak ada. Sehingga, semua instansiasi objek wajib menggunakan constructor berparameter.

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:  
  The constructor Mahasiswa20() is undefined  
  
  at MahasiswaMain20.main(MahasiswaMain20.java:3)  
PS C:\Users\hp\Documents\TUGAS KULYEAH\Praktikum ASD\PraktikumASD\Minggu2>
```

4. Method dalam class Mahasiswa tidak harus diakses berurutan, tetapi urutan pemanggilan akan memengaruhi hasil output karena setiap method bekerja berdasarkan kondisi atribut objek pada saat dipanggil.
5. Berikut hasil membuat object nama mhs<NamaMahasiswa> menggunakan konstruktor berparameter dari class Mahasiswa

```
Mahasiswa20 mhsNaura = new Mahasiswa20(nama: "Naura Fadhillah", nim: "254107020007", kelas: "TI 1C", ipk: 3.8);  
mhsNaura.tampilkanInfo();  
  
Nama: Naura Fadhillah  
NIM: 254107020007  
Kelas: TI 1C  
IPK: 3.8
```

