

JOBSHEET II OBJECT

1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Mengenal objek dan class sebagai konsep mendasar pada pemrograman berorientasi objek
2. Mendeklarasikan class, atribut dan method
3. Membuat objek (instansiasi)
4. Mengakses atribut dan method dari suatu objek
5. Menerapkan konstruktor

2. Praktikum

2.1 Percobaan 1: Deklarasi Class, Atribut dan Method

Waktu Percobaan : 50 Menit

Pada Percobaan 1 ini dilakukan pembuatan class beserta atribut dan method yang dimilikinya.

Perhatikan Class Diagram berikut ini:

Mahasiswa
nim: String nama: String kelas: String ipk: double
tampilkanInformasi(): void ubahKelas(kelasBaru: String): void updateIpk(ipkBaru: double): void nilaiKinerja(ipk: double): String

Berdasarkan class diagram tersebut, akan dibuat program menggunakan bahasa Java.

2.1.1 Langkah-langkah Percobaan

1. Buat folder baru bernama Jobsheet2 di dalam repository Praktikum ASD
2. Buka text editor. Buat file baru, beri nama Mahasiswa<NoAbsen>.java
3. Lengkapi class Mahasiswa dengan atribut yang telah digambarkan di dalam class diagram tersebut

```
public class Mahasiswa {  
    String nama;  
    String nim;  
    String kelas;  
    double ipk;  
}
```

4. Lengkapi class Mahasiswa dengan method yang telah digambarkan di dalam class diagram tersebut.

```
void tampilkanInformasi() {  
    System.out.println("Nama: " + nama);  
    System.out.println("NIM: " + nim);  
}
```

```

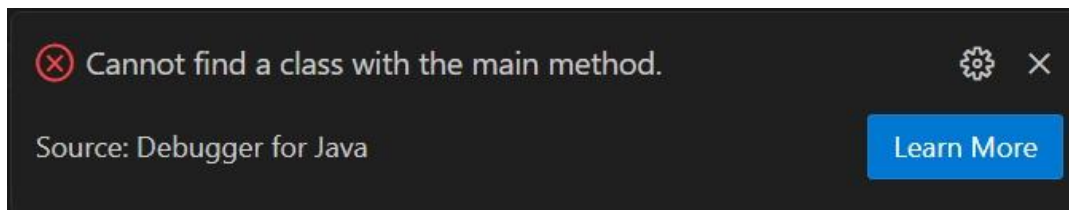
        System.out.println("IPK: " + ipk);
        System.out.println("Kelas: " + kelas);
    }
    void ubahKelas(String kelasBaru) {
        kelas = kelasBaru;
    }
    void updateIpk(double ipkBaru) {
        ipk = ipkBaru;
    }
    String nilaiKinerja() {
        if (ipk >= 3.5) {
            return "Kinerja sangat baik";
        } else if (ipk >= 3.0) {
            return "Kinerja baik";
        } else if (ipk >= 2.0) {
            return "Kinerja cukup";
        } else {
            return "Kinerja kurang";
        }
    }
}

```

5. Compile dan run program.

2.1.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.



2.1.3 Pertanyaan

1. Sebutkan dua karakteristik class atau object!
2. Perhatikan class Mahasiswa pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Mahasiswa? Sebutkan apa saja atributnya!
3. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya!
4. Perhatikan method updateIpk() yang terdapat di dalam class Mahasiswa. Modifikasi isi method tersebut sehingga IPK yang dimasukkan valid yaitu terlebih dahulu dilakukan pengecekan apakah IPK yang dimasukkan di dalam rentang 0.0 sampai dengan 4.0 ($0.0 \leq \text{IPK} \leq 4.0$). Jika IPK tidak pada rentang tersebut maka dikeluarkan pesan: "IPK tidak valid. Harus antara 0.0 dan 4.0".

5. Jelaskan bagaimana cara kerja method nilaiKinerja() dalam mengevaluasi kinerja mahasiswa, kriteria apa saja yang digunakan untuk menentukan nilai kinerja tersebut, dan apa yang dikembalikan (di-return-kan) oleh method nilaiKinerja() tersebut?
6. Commit dan push kode program ke Github

Jawaban Pertanyaan

1. memiliki 4 atribut dan memiliki 4 method
2. 4 atribut yaitu:
 1. nama, 2. nim, 3. kelas, 4. ipk
3. 4 method yaitu:
 1. tampilkan informasi, 2. ubah kelas, 3. update ipk, 4. nilai kinerja
- 4.

```
5.     void updateIpk(double ipkBaru) {  
6.         if (ipkBaru < 0.0 || ipkBaru > 4.0) {  
7.             System.out.println("IPK tidak valid. Harus antara 0.0  
dan 4.0");  
8.         } else {  
9.             ipk = ipkBaru;  
10.        }  
11.    }
```

5. dialam fugsi nilai kerja
Fungsi akan membaca data ipk yang sudah diisi di fungsi updatelpk() kemudian lanjut ke pemilihan:
 - > jika lebih atau sama dengan 3.5 maka akan mengembalikan nilai yang berupa String yaitu "Kinerja sangat baik"
 - > jika lebih atau sama dengan 3.0 maka akan mengembalikan nilai yang berupa String yaitu "Kinerja baik"
 - > jika lebih atau sama dengan 2.0 maka akan mengembalikan nilai yang berupa String yaitu "Kinerja cukup"
 - > jika tidak memenuhi semua kriteria diatas maka akan mengembalikan nilai yang berupa String yaitu "Kinerja kurang"
- 6.

Mahasiswa14.java

Add files via upload

now

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

Waktu Percobaan: 50 Menit

Sampai tahap ini, class Mahasiswa telah berhasil dibuat pada Percobaan 1. Selanjutnya, apabila class Mahasiswa tersebut ingin digunakan dan diakses atribut serta method-nya, maka perlu dibuat object/instance dari class Mahasiswa terlebih dahulu melalui proses instansiasi.

2.2.1 Langkah-langkah Percobaan

1. Buat file baru, beri nama MahasiswaMain<NoAbsen>.java
2. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi main()

3. Di dalam fungsi main(), lakukan instansiasi, kemudian lanjutkan dengan mengakses atribut dan method dari objek yang telah terbentuk.

```
public class MahasiswaMain14 {  
  
    public static void main(String[] args) {  
        Mahasiswa14 mhs1 = new Mahasiswa14();  
        mhs1.nama = "Muhammad Ali Farhan";  
        mhs1.nim = "2241720171";  
        mhs1.kelas = "SI 2J";  
        mhs1.ipk = 3.55;  
  
        mhs1.tampilkanInformasi();  
        mhs1.ubahKelas("SI 2K");  
        mhs1.updateIpk(3.60);  
        mhs1.tampilkanInformasi();  
    }  
}
```

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

4.

5.

MahasiswaMain14.java

Commit percobaan2

now

2.2.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.

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

2.2.3 Pertanyaan

1. Pada class MahasiswaMain, tunjukkan baris kode program yang digunakan untuk proses instansiasi! Apa nama object yang dihasilkan?
2. Bagaimana cara mengakses atribut dan method dari suatu objek?
3. Mengapa hasil output pemanggilan method tampilkanInformasi() pertama dan kedua berbeda?

Jawaban Pertanyaan

1. `Mahasiswa14 mhs1 = new Mahasiswa14();`
nama object: mhs1
2. atribut diakses dengan cara memberi nilai atau data pada object tersebut, method diakses dengan mengetik fungsi setelah object
3. karena pada pemanggilan method tampilkanInformasi() yang pertama dilakukan sebelum nilai pada class mahasiswa dirubah, sedangkan pada method tampilkaninformasi() yang kedua dilakukan saat nilai pada class sudah dirubah

2.3 Percobaan 3: Membuat Konstruktor

Waktu Percobaan: 60 Menit

Pada percobaan ini, dilakukan pembuatan kode program untuk mengimplementasikan berbagai macam konstruktor berdasarkan parameternya.

2.3.1 Langkah-langkah Percobaan

1. Buka kembali class Mahasiswa. Tambahkan dua buah konstruktor di dalam class Mahasiswa tersebut, yang terdiri dari satu konstruktor default dan satu konstruktor berparameter. Konstruktor merupakan method istimewa, penempatan kode program untuk konstruktor dapat diperlakukan sama seperti method yang lain (setelah atribut).

```
public Mahasiswa14() {  
  
}  
  
public Mahasiswa14(String nm, String nim, double ipk, String  
kls) {  
    nama = nm;  
    this.nim = nim;  
    this.ipk = ipk;  
    kelas = kls;  
}
```


2. Buka kembali class MahasiswaMain. Buat sebuah object lagi bernama mhs2 dengan menggunakan konstruktor berparameter.

```
Mahasiswa14 mhs2 = new Mahasiswa14("Annisa  
Nabila", "2141720160", 3.25, "TI 2L");  
mhs2.updateIpk(3.30);  
mhs2.tampilkanInformasi();
```

3. compile dan run program

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

4. Commit dan push kode program ke Github

 Mahasiswa14.java	Commit Percobaan3
 MahasiswaMain14.java	Commit Percobaan3

2.3.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.

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

2.3.3 Pertanyaan

1. Pada class Mahasiswa di Percobaan 3, tunjukkan baris kode program yang digunakan untuk mendeklarasikan konstruktor berparameter!
2. Perhatikan class MahasiswaMain. Apa sebenarnya yang dilakukan pada baris program berikut?

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

3. Hapus konstruktor default pada class Mahasiswa, kemudian compile dan run program.

Bagaimana hasilnya? Jelaskan mengapa hasilnya demikian!

4. Setelah melakukan instansiasi object, apakah method di dalam class Mahasiswa harus diakses secara berurutan? Jelaskan alasannya!
5. Buat object baru dengan nama mhs<NamaMahasiswa> menggunakan konstruktor berparameter dari class Mahasiswa!
6. Commit dan push kode program ke Github

Jawaban Pertanyaan

1.

```
public Mahasiswa14(String nm, String nim, double ipk, String kls) {
    nama = nm;
    this.nim = nim;
    this.ipk = ipk;
    kelas = kls;
}
```

2. Mengisi data yang ada di parameter dari konstruktor **Mahasiswa14** yang ada di class Mahasiswa14

3. error dikarenakan Karena saat kita menghapus konstruktor default, Java tidak lagi menyediakan konstruktor tanpa parameter secara otomatis. Karena ada konstruktor berparameter, Java tidak menyediakan konstruktor default secara otomatis.

4. Tidak harus berurutan dikarenakan : setiap metode bisa dipanggil kapan saja setelah object dibuat

5.

```
Mahasiswa14 mhsAdam = new Mahasiswa14("Moch. Adam Arsyad
Faizin", "244107020164", 3.64, "TI 1B");
mhsAdam.tampilkanInformasi();
```

OUTPUT:

```
Nama: Moch. Adam Arsyad Faizin
NIM: 244107020164
IPK: 3.64
Kelas: TI 1B
```

2.4 Latihan Praktikum

Waktu : 150 Menit

1. Diberikan class diagram dari class MataKuliah sebagai berikut:

MataKuliah
kodeMK: String nama: String sks: int jumlahJam: int
tampilkanInformasi(): void ubahSKS(sksBaru: int): void tambahJam(jam: int): void kurangiJam(jam: int): void

Buat program untuk mengimplementasikan class MataKuliah berdasarkan class diagram di atas,

yang terdiri dari:

- Class MataKuliah (MataKuliah<NoAbsen>.java)
- Class MataKuliahMain (MataKuliahMain<NoAbsen>.java)

Pada class MataKuliahMain buatlah minimal 2 objek. Gunakan konstruktor default dan

konstruktor berparameter saat mengintansiasi objek. Lalu panggil semua method yang sudah

dibuat pada class MataKuliah.

Penjelasan dari atribut dan method pada class MataKuliah tersebut adalah sebagai berikut:

a. Atribut

- kodeMK (String): kode unik untuk mata kuliah.
- nama (String): nama lengkap dari mata kuliah
- sks (int): SKS (Satuan Kredit Semester)
- jumlahJam (int): jumlah total jam pertemuan per minggu untuk mata kuliah

b. Method

- tampilInformasi(): method ini digunakan untuk menampilkan semua informasi yang berkaitan dengan mata kuliah.

2. Diberikan class diagram dari class Dosen sebagai berikut:

Dosen
idDosen: String nama: String statusAktif: boolean tahunBergabung: int bidangKeahlian: String
tampilInformasi(): void setStatusAktif(status: boolean): void hitungMasaKerja(thnSkr: int): int ubahKeahlian(bidang: String): void

Buat program untuk mengimplementasikan class Dosen berdasarkan class diagram di atas, yang

terdiri dari:

- Class Dosen (Dosen<NoAbsen>.java)
- Class DosenMain (DosenMain<NoAbsen>.java)

Pada class DosenMain buatlah minimal 2 objek. Gunakan konstruktor default dan konstruktor

berparameter saat mengintansiasi objek. Lalu panggil semua method yang sudah dibuat pada

class Dosen

Penjelasan dari atribut dan method pada class Dosen tersebut adalah sebagai berikut:

a. Atribut

- idDosen (String): id unik untuk setiap dosen.
- nama (String): nama lengkap dari dosen.

- statusAktif (boolean): menunjukkan apakah dosen tersebut aktif (true) atau tidak aktif (false) dalam menjalankan tugasnya.
 - tahunBergabung (int): tahun ketika dosen mulai bergabung dengan perguruan tinggi
- b. Method
- tampilInformasi(): method ini digunakan untuk menampilkan informasi lengkap tentang dosen
 - setStatusAktif(status: boolean): method ini digunakan untuk mengatur status aktif dosen. Jika parameter status diatur ke true, berarti dosen tersebut menjadi aktif. Sebaliknya, jika false, dosen tersebut dinyatakan tidak aktif.
 - hitungMasaKerja(thnSkrng: int): method ini menghitung dan mengembalikan (me-return-kan) masa kerja dosen dalam tahun, berdasarkan tahun bergabung dan tahun saat ini (thnSkrng) yang menjadi parameter input method ini. Hasil perhitungan memberikan informasi tentang lamanya dosen bekerja di perguruan tinggi.
 - ubahKeahlian(bidang: String): method ini digunakan untuk mengubah bidang keahlian dosen.

Jawaban

1.

//FILE MATAKULIAH14

```
public class MataKuliah14 {
    String kodeMK;
    String nama;
    int sks;
    int jumlahJam;
    public MataKuliah14() {

    }
    public MataKuliah14(String kodeMK, String nama, int sks, int
jumlahJam) {
        this.kodeMK = kodeMK;
        this.nama = nama;
        this.sks = sks;
        this.jumlahJam = jumlahJam;
    }

    void tampilInformasi() {
        System.out.println("Kode MK : " + kodeMK);
        System.out.println("Nama : " + nama);
        System.out.println("sks : " + sks);
        System.out.println("Jumlah jam : " + jumlahJam);
    }
}
```

```

    }

    void ubahSKS(int sksBaru) {
        sks = sksBaru;
        System.out.println("SKS Telah diubah menjadi " + sksBaru + "
SKS");
    }

    void tambahJam(int jam) {
        jumlahJam += jam;
    }

    void kurangiJam(int jam) {
        if (jumlahJam < jam) {
            System.out.println("Pengurangan tidak dapat dilakukan");
        } else {
            jumlahJam -= jam;
        }
    }
}

```

//FILEMAIN

```

import java.util.Scanner;

public class MataKuliahMain14 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MataKuliah14 objek1 = new MataKuliah14();
        objek1.kodeMK = "123";
        objek1.nama = "Adam";
        objek1.sks = 3;
        objek1.jumlahJam = 5;
        System.out.print("Ubah sks: ");
        int sksbru = sc.nextInt();
        objek1.ubahSKS(sksbru);
        System.out.print("Tambah jam: ");
        int tmbh = sc.nextInt();
        objek1.tambahJam(tmbh);
        System.out.print("Kurangi jam : ");
        int krng = sc.nextInt();
        objek1.kurangiJam(krng);
        objek1.tampilInformasi();

        System.out.println("\n");
        MataKuliah14 objek2 = new MataKuliah14("321","Arsyad",3,5);
    }
}

```

```

        objek2.tampilInformasi();
    }
}

```

OUTPUT

```

Ubah sks: 2
SKS Telah diubah menjadi 2 SKS
Tambah jam: 3
Kurangi jam : 2
Kode MK : 123
Nama : Adam
sks : 2
Jumlah jam : 6

Kode MK : 321
Nama : Arsyad
sks : 3
Jumlah jam : 5

```

2.

//FILE CLASS

```

public class Dosen14 {

    String idDosen;
    String nama;
    boolean statusAktif;
    int tahunBergabung;
    String bidangKeahlian;
    int masaKerja;

    public Dosen14() {

    }

    public Dosen14(String id, String nm, boolean sttus, int
    thnbergabung, String bidng, int mskerja) {
        idDosen = id;
        nama = nm;
        this.statusAktif = sttus;
        tahunBergabung = thnbergabung;
    }
}

```

```

        bidng = bidangKeahlian;
        mskerja = masaKerja;
    }

    void tampilInformasi() {
        System.out.println("id dosen : " + idDosen);
        System.out.println("Nama : " + nama);
        System.out.println("Status : " + statusAktif);
        System.out.println("tahun bergabung : " + tahunBergabung);
        System.out.println("bidang keahlian : " + bidangKeahlian);
        System.out.println("Lamanya masa kerja dosen " + masaKerja + "
thn");
    }

    void setStatusAktif(boolean status) {
        this.statusAktif = status;
        if (status == true) {
            System.out.println("Dosen tersebut aktif");
        } else {
            System.out.println("Dosen tersebut tidak aktif");
        }
    }

    int hitungMasaKerja(int thnSkrng) {
        masaKerja = thnSkrng - tahunBergabung;
        return masaKerja;
    }

    void ubahKeahlian(String bidang) {
        this.bidangKeahlian = bidang;
    }
}

```

//FILE MAIN

```

import java.util.Scanner;

public class DosenMain14 {

    public static void main(String[] args) {
        Dosen14 dsn1 = new Dosen14();
        dsn1.idDosen = "1234";
        dsn1.nama = "adam";
        dsn1.statusAktif = false;
        dsn1.tahunBergabung = 2013;
        dsn1.bidangKeahlian = "Daspro";
        dsn1.setStatusAktif(true);
    }
}

```

```

        dsn1.hitungMasaKerja(2025);
        dsn1.ubahKeahlian("CTPS");
        dsn1.tampilInformasi();

        System.out.println("\n");
        Dosen14 dsn2 = new
Dosen14("4321", "mada", true, 2020, "BasisData", 0);
        dsn2.setStatusAktif(false);
        dsn2.hitungMasaKerja(2025);
        dsn2.ubahKeahlian("Matematika Dasar");
        dsn2.tampilInformasi();
    }
}

```

OUTPUT

```

Dosen tersebut aktif
id dosen : 1234
Nama : adam
Status : true
tahun bergabung : 2013
bidang keahlian : CTPS
Lamanya masa kerja dosen 12 thn

Dosen tersebut tidak aktif
id dosen : 4321
Nama : mada
Status : false
tahun bergabung : 2020
bidang keahlian : Matematika Dasar
Lamanya masa kerja dosen 5 thn

```

