

Nama : Achmad Pradita Dwi Firmansyah  
NIM : 254107020130  
Kelas / Absen : TI – 1G / 01

## JOBSHEET II

### OBJECT

#### 1. Praktikum

##### 1.1 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 updatelpk(ipkBaru: double): void nilaiKinerja(ipk: double): String

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

##### 1.1.1 Kode Program

```
public class Mahasiswa01 {  
    String nama;  
    String nim;  
    String kelas;  
    double ipk;  
    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 updatelpk(double ipkBaru) {  
        ipk = ipkBaru;  
    }  
    String nilaiKerja() {  
        if (ipk >= 3.5) {  
            return "Kinerja sangat baik";  
        }else if (ipk >= 3.0) {  
            return "Kinerja baik";  
        }  
    }  
}
```

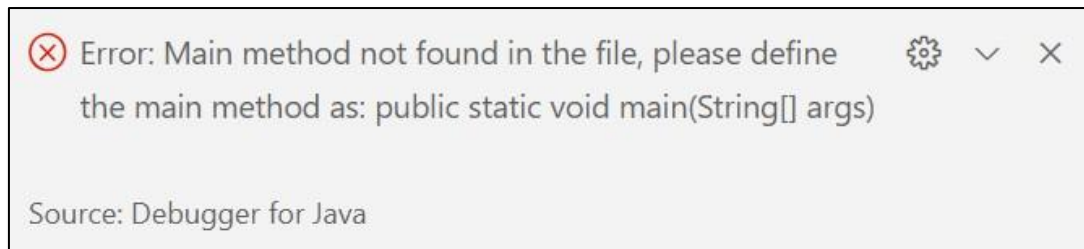
```

    }else if (ipk >= 2.0) {
        return "Kinerja cukup";
    }else {
        return "Kinerja kurang";
    }
}
}
}

```

### 1.1.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.



```

PS C:\PENYIMPANAN\Documents\G\PraktikumAlgoritmaDanStrukturData> java Mahasiswa01.java
Error: Could not find or load main class Mahasiswa01.java
Caused by: java.lang.ClassNotFoundException: Mahasiswa01.java
PS C:\PENYIMPANAN\Documents\G\PraktikumAlgoritmaDanStrukturData>

```

### 1.1.3 Pertanyaan

1. Sebutkan dua karakteristik class atau object!
  - a) Atribut yang merupakan data yang dimiliki oleh object atau class.
  - b) Method yang merupakan tingkah laku yang bisa dilakukan oleh object.
2. Perhatikan class **Mahasiswa** pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Mahasiswa? Sebutkan apa saja atributnya!

Pada class **Mahasiswa**, terdapat 4 macam atribut yaitu, nama, nim, kelas, dan ipk.

3. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya!  
Class **Mahasiswa** memiliki 4 macam method, yaitu method untuk tampilkanInformasi, ubahKelas, updateIpk, dan nilai kerja
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".

```

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");
    }
}

```

```
}  
  
}
```

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?

Pada method **nilaiKerja()**, program membutuhkan input nilai dari ipk terlebih dahulu, kemudian jika nilai ipk 3.5 atau di atasnya, maka method akan mengembalikan nilai “Kinerja sangat baik”. Jika nilai ipk antara dari 3.5 hingga 3.0 maka nilai yang dikembalikan “Kinerja baik”. Jika nilai kurang dari 3.0 sampai sama dengan 2.0 maka nilai yang dikembalikan “Kinerja cukup”. Dan jika nilai di bawah itu maka nilai yang dikembalikan “Kinerja Kurang”.

6. **Commit dan push kode program ke Github**

## 1.2 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.

#### 1.2.1 Kode Program

```
public class MahasiswaMain01 {  
    public static void main(String[] args) {  
        Mahasiswa01 mhs1 = new Mahasiswa01();  
        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();  
    }  
}
```

#### 1.2.2 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
```

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

```
PS C:\PENYIMPANAN\Documents\G\PraktikumAlgoritmaDanStrukturData>
```

### 1.2.3 Pertanyaan

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

```
Mahasiswa01 mhs1 = new Mahasiswa01();
object yang dihasilkan mhs1
```

2. Bagaimana cara mengakses atribut dan method dari suatu objek?  
Untuk mengakses atribut dan method dari fungsi main, kita perlu untuk menuliskan perintah nama object di sambung dengan titik dan nama atribut atau methodnya. Contoh : **mhs1.nama & mhs1.tampilkanInformasi();**
3. Mengapa hasil output pemanggilan method **tampilkanInformasi()** pertama dan kedua berbeda?  
Karena setelah pemanggilan method **tampilkanInformasi()** pertama terdapat perintah untuk mengubah kelas dan nilai ipk.

## 1.3 Percobaan 3: Membuat Konstruktor

### Waktu Percobaan: 60 Menit

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

#### 1.3.1 Kode Program

```
public class Mahasiswa01 {
    String nama;
    String nim;
    String kelas;
    double ipk;

    public Mahasiswa01() {

    }
}
```

```

public Mahasiswa01(String nm, String nim, double ipk, String kls) {
    nama = nm;
    this.nim = nim;
    this.ipk = ipk;
    kelas = kls;
}
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) {
    if (ipk >= 0.0 && ipk <= 4.00) {
        ipk = ipkBaru;
    }else {
        System.out.println("IPK tidak valid. Harus antara 0.0 dan 4,0");
    }
}
String nilaiKerja() {
    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";
    }
}
}
public class MahasiswaMain01 {
    public static void main(String[] args) {
        Mahasiswa01 mhs1 = new Mahasiswa01();
        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();

        Mahasiswa01 mhs2 = new Mahasiswa01("Annisa Nabila", "2141720160", 3.25, "TI 2L");
        mhs2.updateIpk(3.30);
        mhs2.tampilkanInformasi();
    }
}

```

### 1.3.2 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
```

```
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
PS C:\PENYIMPANAN\Documents\G\PraktikumAlgoritmaDanStrukturData>
```

### 1.3.3 Pertanyaan

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

```
public Mahasiswa01(String nm, String nim, double ipk, String kls) {
```

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

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

Baris program berikut berisi perintah untuk pembuatan object baru yaitu mhs2 dengan nama Annisa Nabila, nim 2141720160, ipk 3.25, dan kelas TI 2L

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

Jika konstruktor default dihapus, maka program harus membutuhkan input nilai dari atribut, sehingga program tidak bisa membaca baris **Mahasiswa01 mhs1 = new Mahasiswa01();** yang menyebabkan program error.

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

Untuk urutan mengakses method dapat dilakukan secara acak, asalkan object tersebut sudah di instansiasi terlebih dahulu, jika salah satu atribut object belum di instansiasi maka atribut tersebut akan kosong.

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

```
Mahasiswa01 mhsAdit = new Mahasiswa01("Achmad Pradita Dwi Firmansyah", "254107020130",  
3.5, "TI 1G");  
mhsAdit.tampilkanInformasi();
```

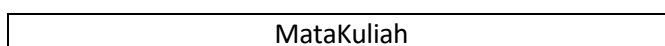
6. Commit dan push kode program ke Github

```
public class MahasiswaMain01 {  
    public static void main(String[] args) {  
        Mahasiswa01 mhs1 = new Mahasiswa01();  
        mhs1.nama = "Muhammad Ali Farhan";  
        mhs1.nim = "2241720171";  
        mhs1.kelas = "SI 2J";  
        mhs1.ipk = 3.55;  
  
        mhs1.tampilkanInformasi();  
        mhs1.ubahKelas("SI 2K");  
        mhs1.updatelpk(3.60);  
        mhs1.tampilkanInformasi();  
  
        Mahasiswa01 mhs2 = new Mahasiswa01("Annisa Nabila", "2141720160", 3.25, "TI 2L");  
        mhs2.updatelpk(3.30);  
        mhs2.tampilkanInformasi();  
  
        Mahasiswa01 mhsAdit = new Mahasiswa01("Achmad Pradita Dwi Firmansyah",  
"254107020130", 3.5, "TI 1G");  
        mhsAdit.tampilkanInformasi();  
    }  
}
```

## 1.4 Latihan Praktikum

Waktu : 150 Menit

1. Diberikan class diagram dari class MataKuliah sebagai berikut:



kodeMK: String nama: String sks: int jumlahJam: int
tampilInformasi(): 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.
- ubahSKS(int sksBaru): method ini memungkinkan pengubahan nilai SKS untuk mata kuliah. Setelah mengubah nilai, method ini memberi tahu pengguna bahwa SKS telah diubah.
- tambahJam(int jam): method ini menambahkan jumlah jam tambahan ke jumlah jam yang sudah ada untuk mata kuliah.
- kurangiJam(int jam): method ini berfungsi untuk mengurangi jumlah jam dari mata kuliah. Sebelum mengurangi, method ini melakukan pengecekan untuk memastikan bahwa jumlah jam yang tersisa cukup untuk dikurangi. Jika jumlah jam tidak mencukupi (jumlah jam awal lebih kecil dari jam pengurang), method ini akan memberi tahu pengguna bahwa pengurangan tidak dapat dilakukan. Jika pengurangan berhasil, method ini mengupdate jumlah jam dan mencetak nilai jumlah jam yang baru.

#### 1.4.1 Kode Program

```
public class MataKuliah01 {
```



```

String kodeMK, nama;
int sks, jumlahJam;

public MataKuliah01() {

}

public MataKuliah01(String mk, String nm, int sks, int jmlJam) {
    kodeMK = mk;
    nama = nm;
    this.sks = sks;
    jumlahJam = jmlJam;
}

void tampilInformasi() {
    System.out.println("Kode Mata Kuliah: " + kodeMK);
    System.out.println("Nama Mata Kuliah: " + nama);
    System.out.println("Jumlah SKS\t: " + sks);
    System.out.println("Jumlah Jam\t: " + jumlahJam);
}

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

void tambahJam(int jam) {
    jumlahJam += jam;
    System.out.println("Jumlah jam telah ditambah menjadi: " + jumlahJam);
}

void kurangJam(int jam) {
    if (jumlahJam > jam) {
        jumlahJam -= jam;
        System.out.println("Jumlah jam telah dikurangi menjadi: " + jumlahJam);
    } else {
        System.out.println("Jam tidak mencukupi untuk dikurangi");
    }
}

}

public class MataKuliahMain01 {
    public static void main(String[] args) {
        MataKuliah01 mk1 = new MataKuliah01();
        mk1.kodeMK = "PASD_TI";
        mk1.nama = "Praktikum Algoritma dan Struktur Data";
        mk1.sks = 2;
        mk1.jumlahJam = 4;

        mk1.tampilInformasi();
        mk1.ubahSKS(3);
        mk1.tambahJam(3);
        mk1.kurangJam(1);
        mk1.tampilInformasi();

        MataKuliah01 mk2 = new MataKuliah01("ASD_TI", "Algoritma dan Struktur Data", 2, 4);
        mk2.tampilInformasi();
    }
}

```

```
}  
}
```

```
Kode Mata Kuliah: PASD_TI  
Nama Mata Kuliah: Praktikum Algoritma dan Struktur Data  
Jumlah SKS      : 2  
Jumlah Jam      : 4  
SKS telah diubah menjadi: 3  
Jumlah jam telah ditambah menjadi: 7  
Jumlah jam telah dikurangi menjadi: 6  
Kode Mata Kuliah: PASD_TI  
Nama Mata Kuliah: Praktikum Algoritma dan Struktur Data  
Jumlah SKS      : 3  
Jumlah Jam      : 6  
Kode Mata Kuliah: ASD_TI  
Nama Mata Kuliah: Algoritma dan Struktur Data  
Jumlah SKS      : 2  
Jumlah Jam      : 4  
PS C:\PENYIMPANAN\Documents\G\PraktikumAlgoritmaDanStrukturData>
```

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

- bidangKeahlian (String): bidang keahlian dosen, yang menjelaskan spesialisasi atau fokus akademik dosen

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.

```
ID Dosen: ACH
Nama: Achmad
Status: Tidak Aktif
Tahun Bergabung: 2006
Bidang Keahlian: Teknologi Informasi
Masa Kerja: 20 tahun

Status telah diubah menjadi: Aktif
Bidang Keahlian telah diubah menjadi : Sistem Informasi

ID Dosen: ACH
Nama: Achmad
Status: Aktif
Tahun Bergabung: 2006
Bidang Keahlian: Sistem Informasi

ID Dosen: PA
Nama: Pradita
Status: Tidak Aktif
Tahun Bergabung: 2015
Bidang Keahlian: Teknologi Informasi
PS C:\PENYIMPANAN\Documents\G\PraktikumAlgoritmaDanStrukturData>
```

#### 1.4.2 Kode Program

```

public class Dosen01 {
    String idDosen, nama, bidangKeahlian;
    boolean statusAktif;
    int tahunBergabung;

    public Dosen01() {

    }

    public Dosen01 (String id, String nm, String bidang, boolean status, int tahunBergabung) {
        idDosen = id;
        nama = nm;
        statusAktif = status;
        this.tahunBergabung = tahunBergabung;
        bidangKeahlian = bidang;
    }

    void tampilInformasi() {
        System.out.println("\nID Dosen: " + idDosen);
        System.out.println("Nama: " + nama);
        if (statusAktif == true) {
            System.out.println("Status: Aktif");
        } else {
            System.out.println("Status: Tidak Aktif");
        }
        System.out.println("Tahun Bergabung: " + tahunBergabung);
        System.out.println("Bidang Keahlian: " + bidangKeahlian);
    }

    void setStatusAktif(boolean status) {
        statusAktif = status;
        if (statusAktif == true) {
            System.out.println("\nStatus telah diubah menjadi: Aktif");
        } else {
            System.out.println("\nStatus telah diubah menjadi: Tidak Aktif");
        }
    }

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

    void ubahKeahlian(String bidang) {
        bidangKeahlian = bidang;
        System.out.println("Bidang Keahlian telah diubah menjadi : "+ bidangKeahlian);
    }
}

public class DosenMain01 {
    public static void main(String[] args) {
        Dosen01 ds1 = new Dosen01();
        ds1.idDosen = "ACH";
        ds1.nama = "Achmad";
        ds1.statusAktif = false;
    }
}

```

```
ds1.tahunBergabung = 2006;
ds1.bidangKeahlian = "Teknologi Informasi";

ds1.tampilInformasi();
int masaKerja = ds1.hitungMasaKerja(2026);
System.out.println("Masa Kerja : " + masaKerja + " Tahun");
ds1.setStatusAktif(true);
ds1.ubahKeahlian("Sistem Informasi");
ds1.tampilInformasi();

Dosen01 ds2 = new Dosen01("PA", "Pradita", "Teknologi Informasi", false, 2015);
ds2.tampilInformasi();
    }
}
```