

**JOB SHEET 2**

**PRAKTIKUM AIGORTIMA**

**DAN STRUKTUR DATA**



**Burhnauddin ihsan**

**244107020189**

**TI 1E/06**

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG 2025**

## PERCOBAAN 1

### - Kode program

```
public class Mahasiswa06 {

    String nama;
    String nim;
    double ipk;
    String kelas;
    String nilaiKinerja;

    void tampilkanInformasi(){
        System.out.println("Nama : " + nama);
        System.out.println("NIM : " + nim);
        System.out.println("IPK : " + ipk);
        System.out.println("Kelas : " + kelas);
        System.out.println("Nilai Kinerja : " + nilaiKinerja);
    }
    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";
        }
    }
}
```

### - Pertanyaan

1. Dua karakteristik utama dari class atau object dalam pemrograman adalah atribut dan method:
  - atribut adalah data yang dimiliki sebuah objek dan atribut digunakan untuk menentukan karakteristik dari objek
  - method adalah fungsi yang bisa dilakukan objek dan method digunakan untuk mengoperasikan atribut dari objek
2. Nama, nim, ipk, kelas, nilai kinerja
3. Ada 4 method yaitu
  - tampilkanInformasi()
  - ubahKelas(kelasBaru)

- updateIPK(double ipkBaru)

- nilaiKinerja()

#### 4. Kode program

```
public class Mahasiswa06 {

    String nama;
    String nim;
    double ipk;
    String kelas;

    void tampilkanInformasi(){
        System.out.println("Nama : " + nama);
        System.out.println("NIM : " + nim);
        System.out.println("IPK : " + ipk);
        System.out.println("Kelas : " + kelas);
    }

    public Mahasiswa06(){

    }

    public Mahasiswa06(String nm, String nim, double ipk, String kls) {
        nama = nm;
        this.nim = nim;
        this.ipk = ipk;
        kelas = kls;
    }

    void ubahKelas(String kelasBaru){
        kelas = kelasBaru;
    }

    void updateIPK(double ipkBaru){
        if (ipk >= 0 && ipk <=4) {
            ipk = ipkBaru;
        }else{
            System.out.print("ipk tidak valid. Harus antara 0.0 dan 4.0 ");
        }
    }

    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. Cara kerja method:

- Method ini tidak menerima parameter dan memiliki tipe return String, yang berarti akan mengembalikan sebuah teks berdasarkan nilai IPK.
- Di dalam method, digunakan struktur percabangan if-else untuk menentukan kategori kinerja berdasarkan nilai IPK mahasiswa.
- Method ini akan mengembalikan (return) sebuah teks yang menggambarkan kinerja mahasiswa berdasarkan kategori yang telah ditentukan.

## PERCOBAAN 2

- kode program

```
public class MahasiswaMain06 {  
    public static void main(String[] args) {  
        Mahasiswa06 mhs1 = new Mahasiswa06 ();  
        mhs1.nama = "Muhammad Ali Farhan";  
        mhs1.nim = "244107020189";  
        mhs1.kelas = "ST 2J";  
        mhs1.ipk = 3.55;  
  
        mhs1.tampilkanInformasi();  
        mhs1.ubahKelas("SI 2K");  
        mhs1.updateIPK(3.60);  
        mhs1.tampilkanInformasi();  
    }  
}
```

- Hasil dari kode program

```
Nama : Muhammad Ali Farhan  
NIM : 244107020189  
IPK : 3.55  
Kelas : ST 2J  
Nama : Muhammad Ali Farhan  
NIM : 244107020189  
IPK : 3.6  
Kelas : SI 2K  
PS C:\Users\Lenovo\OneDrive\file\Praktikum-ASD>
```

- PERTANYAAN

1. Baris kode untuk instansiasi: `Mahasiswa06 mhs1 = new Mahasiswa06();`

nama objek yang dihasilkan adalah mhs1

2. – atribut diakses dengan `namaObjek.namaAtribut`

– Method dipanggil dengan `namaObjek.namaMethod()`.

3. Output pertama dan kedua berbeda karena method `ubahKelas("SI 2K")` dan `updateIPK(3.60)` mengubah nilai atribut kelas dan ipk dalam objek mhs setelah pemanggilan pertama.

### PERCOBAAN 3

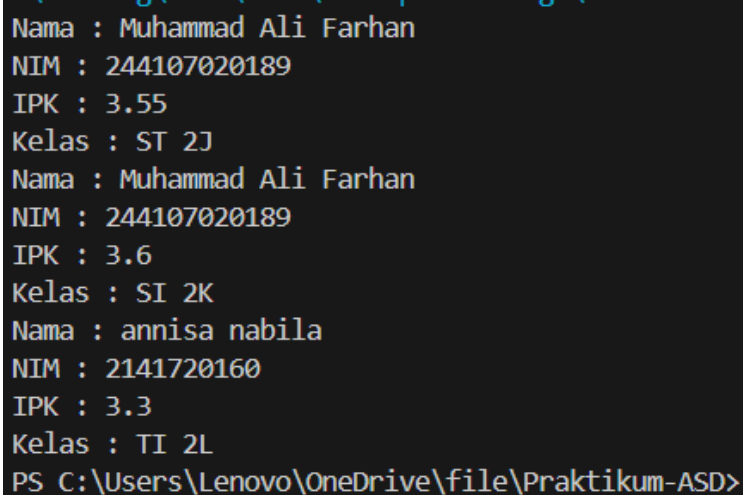
- kode program

```
public class MahasiswaMain06 {
    public static void main(String[] args) {
        Mahasiswa06 mhs1 = new Mahasiswa06 ();
        mhs1.nama = "Muhammad Ali Farhan";
        mhs1.nim = "244107020189";
        mhs1.kelas = "ST 2J";
        mhs1.ipk = 3.55;

        mhs1.tampilkanInformasi();
        mhs1.ubahKelas("SI 2K");
        mhs1.updateIPK(3.60);
        mhs1.tampilkanInformasi();

        Mahasiswa06 mhs2 = new Mahasiswa06("annisa nabila", "2141720160", 3.25, "TI
2L");
        mhs2.updateIPK(3.30);
        mhs2.tampilkanInformasi();
    }
}
```

- Hasil dari kode program



```
Nama : Muhammad Ali Farhan
NIM : 244107020189
IPK : 3.55
Kelas : ST 2J
Nama : Muhammad Ali Farhan
NIM : 244107020189
IPK : 3.6
Kelas : SI 2K
Nama : annisa nabila
NIM : 2141720160
IPK : 3.3
Kelas : TI 2L
PS C:\Users\Lenovo\OneDrive\file\Praktikum-ASD>
```

- PERTANYAAN

1. baris kode program yang digunakan untuk mendeklarasikan konstruktor berparameter adalah

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

2. - Membuat objek mhs dari class Mahasiswa06
  - membuat konstruktor berparameter untuk langsung mengisi atribut objek
  - mempermudah inisialisasi objek tanpa perlu mengisi atribut satu per satu

3. Kompilator akan menghasilkan error

Karena Kode `new Mahasiswa06();` Tidak Valid

Kita mencoba membuat objek mhs1 tanpa memberikan parameter.

tetapi, karena satu-satunya konstruktor yang tersedia membutuhkan empat parameter (nama, nim, ipk, kelas), maka Java tidak tahu cara memproses `new Mahasiswa06();` dan menampilkan error.

Juga karena Java Tidak Secara Otomatis Membuat Konstruktor Default

4. Tidak, method dalam class Mahasiswa tidak harus diakses secara berurutan.

- Setiap Method Bisa Dipanggil Secara Mandiri

Kita bisa memanggil method dalam urutan apa pun tergantung pada kebutuhan program.

- Tergantung pada Efek yang Dihasilkan

Jika ada method yang mengubah atribut, maka method lain yang bergantung pada atribut tersebut akan terpengaruh.

## 5. - kode program

```
public class MahasiswaMain06 {
    public static void main(String[] args) {
        Mahasiswa06 mhs1 = new Mahasiswa06 ();
        mhs1.nama = "Muhammad Ali Farhan";
        mhs1.nim = "244107020189";
        mhs1.kelas = "ST 2J";
        mhs1.ipk = 3.55;

        mhs1.tampilkanInformasi();
        mhs1.ubahKelas("SI 2K");
        mhs1.updateIPK(3.60);
        mhs1.tampilkanInformasi();

        Mahasiswa06 mhs2 = new Mahasiswa06("annisa nabila", "2141720160", 3.25, "TI
2L");
        mhs2.updateIPK(3.30);
        mhs2.tampilkanInformasi();

        Mahasiswa06 mhsihsan = new Mahasiswa06 ("Burhanuddin Ihsan", "244107020189",
3.88, "ST 2J");
        mhsihsan.tampilkanInformasi();
    }
}
```

### - Hasil dari kode program

```
Nama : Muhammad Ali Farhan
NIM : 244107020189
IPK : 3.55
Kelas : ST 2J
Nama : Muhammad Ali Farhan
NIM : 244107020189
IPK : 3.6
Kelas : SI 2K
Nama : annisa nabila
NIM : 2141720160
IPK : 3.3
Kelas : TI 2L
Nama : Burhanuddin Ihsan
NIM : 244107020189
IPK : 3.88
Kelas : ST 2J
PS C:\Users\Lenovo\OneDrive\file\Praktikum-ASD>
```

## - LATIHAN

### 1. Mata kuliah

#### - Kode program

```
public class MataKuliah06 {
    String kodeMK;
    String nama;
    int sks;
    int jumlahJam;

    public MataKuliah06() {
        this.kodeMK = "-";
        this.nama = "-";
        this.sks = 0;
        this.jumlahJam = 0;
    }

    public MataKuliah06(String kodeMK, String nama, int sks, int jumlahJam) {
        this.kodeMK = kodeMK;
        this.nama = nama;
        this.sks = sks;
        this.jumlahJam = jumlahJam;
    }

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

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

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

    public void kurangiJam(int jam) {
        if (this.jumlahJam >= jam) {
            this.jumlahJam -= jam;
            System.out.println("Jumlah jam telah dikurangi. Sekarang: " + jumlahJam);
        } else {
            System.out.println("Pengurangan tidak dapat dilakukan. Jumlah jam tidak mencukupi.");
        }
        System.out.println();
    }
}
```



- Main MataKuliah

```
public class MataKuliahMain06 {
    public static void main(String[] args) {
        MataKuliah06 mk1 = new MataKuliah06();
        System.out.println("Informasi Mata Kuliah 1:");
        mk1.tampilInformasi();
        System.out.println();

        MataKuliah06 mk2 = new MataKuliah06("ASD", "Struktur data dan algoritma", 3,
4);
        System.out.println("Informasi Mata Kuliah 2:");
        mk2.tampilInformasi();
        System.out.println();

        mk2.ubahSKS(2);
        mk2.tambahJam(2);
        mk2.kurangiJam(3);

        MataKuliah06 mk3 = new MataKuliah06("AGM", "Agama", 2, 3);
        System.out.println("Informasi Mata Kuliah 3:");
        mk3.tampilInformasi();
        System.out.println();

        mk3.ubahSKS(5);
        mk3.tambahJam(4);
        mk3.kurangiJam(2);

    }
}
```

- Hasil kode program

```
Informasi Mata Kuliah 1:
Kode Mata Kuliah: -
Nama Mata Kuliah: -
SKS: 0
Jumlah Jam: 0

Informasi Mata Kuliah 2:
Kode Mata Kuliah: ASD
Nama Mata Kuliah: Struktur data dan algoritma
SKS: 3
Jumlah Jam: 4

SKS telah diubah menjadi: 2
Jumlah jam telah ditambah. Sekarang: 6
Jumlah jam telah dikurangi. Sekarang: 3

Informasi Mata Kuliah 3:
Kode Mata Kuliah: AGM
Nama Mata Kuliah: Agama
SKS: 2
Jumlah Jam: 3

SKS telah diubah menjadi: 5
Jumlah jam telah ditambah. Sekarang: 7
Jumlah jam telah dikurangi. Sekarang: 5
```

## 2. Daftar nama dosen

### - Kode program

```
public class Dosen06 {
    String idDosen;
    String nama;
    boolean statusAktif;
    int tahunBergabung;
    String bidangKeahlian;

    public Dosen06() {
        this.idDosen = "";
        this.nama = "";
        this.statusAktif = false;
        this.tahunBergabung = 0;
        this.bidangKeahlian = "";
    }

    public Dosen06(String idDosen, String nama, boolean statusAktif, int tahunBergabung,
String bidangKeahlian) {
        System.out.println();
        System.out.println("----- Daftar Nama Dosen -----");
        this.idDosen = idDosen;
        this.nama = nama;
        this.statusAktif = statusAktif;
        this.tahunBergabung = tahunBergabung;
        this.bidangKeahlian = bidangKeahlian;
    }

    public void tampilInformasi() {
        System.out.println();
        System.out.println("ID Dosen: " + idDosen);
        System.out.println("Nama: " + nama);
        System.out.println("Status Aktif: " + (statusAktif ? "Aktif" : "Tidak Aktif"));
        System.out.println("Tahun Bergabung: " + tahunBergabung);
        System.out.println("Bidang Keahlian: " + bidangKeahlian);
    }

    public void setStatusAktif(boolean status) {
        this.statusAktif = status;
    }

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

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

- Main Dosen

```
public class DosenMain06 {

    public static void main(String[] args) {
        Dosen06 dosen1 = new Dosen06();
        dosen1.tampilInformasi();

        Dosen06 dosen2 = new Dosen06("12839123", "VIT ZURAIDA", true, 2010, "Basis
data");
        dosen2.tampilInformasi();

        dosen2.setStatusAktif(false);
        dosen2.ubahKeahlian("Data Science");
        dosen2.tampilInformasi();

        System.out.println("Masa kerja dosen kedua: " + dosen2.hitungMasaKerja(2024)
+ " tahun");

        Dosen06 dosen3 = new Dosen06("192932", "wilda imama", true, 2005, "Rekayasa
perangkat lunak");
        dosen3.tampilInformasi();

        dosen3.setStatusAktif(false);
        dosen3.ubahKeahlian("Agama");
        dosen3.tampilInformasi();

        System.out.println("Masa kerja dosen kedua: " + dosen3.hitungMasaKerja(2024)
+ " tahun");

        Dosen06 dosen4 = new Dosen06("823745", "Yan watequlis", true, 2008,
"critical thingking");
        dosen4.tampilInformasi();

        dosen4.setStatusAktif(false);
        dosen4.ubahKeahlian("Dasar pemrograman");
        dosen4.tampilInformasi();

        System.out.println("Masa kerja dosen kedua: " + dosen4.hitungMasaKerja(2024)
+ " tahun");
    }
}
```

- Hasil dari kode program

```
----- Daftar Nama Dosen -----  
  
ID Dosen: 12839123  
Nama: VIT ZURAIDA  
Status Aktif: Aktif  
Tahun Bergabung: 2010  
Bidang Keahlian: Basis data  
  
ID Dosen: 12839123  
Nama: VIT ZURAIDA  
Status Aktif: Tidak Aktif  
Tahun Bergabung: 2010  
Bidang Keahlian: Data Science  
Masa kerja dosen kedua: 14 tahun  
  
----- Daftar Nama Dosen -----  
  
ID Dosen: 192932  
Nama: wilda imama  
Status Aktif: Aktif  
Tahun Bergabung: 2005  
Bidang Keahlian: Rekayasa perangkat lunak  
  
ID Dosen: 192932  
Nama: wilda imama  
Status Aktif: Tidak Aktif  
Tahun Bergabung: 2005  
Bidang Keahlian: Agama  
Masa kerja dosen kedua: 19 tahun  
  
----- Daftar Nama Dosen -----  
  
ID Dosen: 823745  
Nama: Yan watequlis  
Status Aktif: Aktif  
Tahun Bergabung: 2008  
Bidang Keahlian: critical thingking  
  
ID Dosen: 823745  
Nama: Yan watequlis  
Status Aktif: Tidak Aktif  
Tahun Bergabung: 2008  
Bidang Keahlian: Dasar pemrograman  
Masa kerja dosen kedua: 16 tahun  
PS C:\Users\Lenovo\OneDrive\file\Praktikum-ASD>
```

