

Nama : Reyhandhika Zikri Prijadi
Nim : 254107020219
Kelas : TI_1G

Code Percobaan 1:

```
public class Mahasiswa26 {  
    String nama;  
    String nim;  
    String kelas;  
    double ipk;  
    public Mahasiswa26() {  
    }  
    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";
    }
}
}

```

2.1.3 Pertanyaan

1. Sebutkan dua karakteristik class atau object!

Dalam paradigma Pbject-Oriented Programming(OOP) terdapat Dua karakteristik utama:

1. Atribut (Property/Stace)
 - Menyimpan data atau Keadaan dari object.
 - Contoh : nama, nim, ipk.
2. Method (Behavior)
 - Menentukan perilaku atau aksi yang dapat dilakukan object
 - Contoh : tampilkanInformasi(), ubahKelas().

2. Perhatikan class Mahasiswa pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class

Mahasiswa? Sebutkan apa saja atributnya!

Pada Class Mahasiswa26 memiliki 4 atribut yaitu:

1. String nama
2. String nim
3. String kelas
4. Double ipk

3. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya! pada Class Mahasiswa26 terdapat 4 Method yaitu:

1. tampilkanInformasi() : void
2. ubahKelas(StringKelasBaru) : void
3. Updateipk(double ipkBaru) : void
4. nilaiKinerja() : void

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

memodifikasi code :

```

void updateipk(double ipkBaru){

    if (ipkBaru >= 0.0 && ipkBaru <= 4.0) {

```

```

        ipk = ipkBaru;
    } else {
        System.out.println("Nilai IPK tidak valid. Harus antara 0.0 dan 4.0.");
    }
}

```

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?

→ Method ini menggunakan Struktur control if-else untuk mengevaluasi nilai ipk

1. Jika ipk >= 3.5

→ return "Kinerja sangat baik"

2. Jika ipk >= 3.0

→ return "Kinerja baik"

3. Jika ipk >= 2.0

→ return "Kinerja cukup"

4. Selain itu

→ return "Kinerja kurang"

6. Commit dan push kode program ke Github

Code Percobaan 2

```

public class MahasiswaMain26 {

    public static void main(String[] args) {

        Mahasiswa26 mhs1 = new Mahasiswa26();

        mhs1.nama = "Reyhandhika Zikri Prijadi";
        mhs1.nim = "254107020219";
        mhs1.kelas = "TI 1G";
        mhs1.ipk = 3.7;

        mhs1.tampilkanInformasi();
        mhs1.ubahKelas("TI 1H");
    }
}

```

```

    mhs1.Updatelpk(3.8);
    mhs1.tampilkanInformasi();
}
}

```

2.2.3 Pertanyaan

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

Jawaban :

Proses instansiasi terdapat pada barisan :

```

Mahasiswa26 mhs1 = new Mahasiswa26();

```

barisan ini digunakan untuk membuat objek baru dari class Mahasiswa26.

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

Jawaban :

Untuk mengakses atribut dan method dari suatu objek digunakan operator titik (.).

Caranya adalah dengan menuliskan nama objek, kemudian tanda titik, lalu nama atribut atau method yang ingin digunakan. Jika mengakses atribut, kita bisa membaca atau mengubah nilainya. Jika mengakses method, kita memanggil fungsi tersebut agar dijalankan

3. Mengapa hasil output pemanggilan method tampilkanInformasi() pertama dan kedua

Jawaban :

Karena sebelum pemanggilan kedua, terjadi perubahan data pada objek. Awalnya kelas = TI 1G dan IPK = 3.7 setelah itu berubah Kelas menjadi TI 1H dan IPK Menjadi 3.8 karena data didalam objek sudah berubah, maka saat tampilkanInformasi() dipanggil lagi yang ditampilkan Adalah data Terbaru. Jadi perbedaan output ini terjadi karena nilai atribut pada objek diperbarui sebelum panggilan kedua.

Code Percobaan 3

2.3.3 Pertanyaan

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

mendeklarasikan konstruktor berparameter!

Jawaban :

Konstruktor Berparameter terdapat pada barisan:

```

public Mahasiswa26(String nm, String nim, String kls, double ipk) {

```

Barisan ini disebut konstruktor Berparameter karena menerima nilai saat objek di buat. Nilai tersebut langsung digunakan untuk mengisi atribut nama, nim, kelas, dan ipk

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

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

Jawaban :

Barisan program tersebut digunakan untuk membuat objek baru Bernama mhs2 dari class Mahasiswa26 dengan menggunakan Konstruktor Berparameter saat objek digunakan dibuat, data seperti nama, nim, kelas, dan lpg langsung diisi sesuai nilai yang diberikan didalam tanda kurung. Didalam satu baris tersebut terjadi proses pembuatan objek sekaligus pengisian data awal mahasiswa tanpa perlu mengatur atribut satu per satu lagi.

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

Jawaban :

Jika konstruktor ini dihapus, maka saat program dijalankan maka terjadi Error Compile.

4. Setelah melakukan instansiasi object, apakah method di dalam class Mahasiswa harus diakses

secara berurutan? Jelaskan alasannya!

Jawaban :

Tidak Harus karena bisa dipanggil sesuai kebutuhan program. Tidak ada aturan bahwa method harus dipanggil sesuai urutan tertentu.

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

```
Mahasiswa26 mhs3 = new Mahasiswa26("Dimas", "254107020221", "TI 1G", 2.5);  
mhs3.tampilkanInformasi();
```

6. Commit dan push kode program ke Github

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
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.

code Latihan MataKuliah26

```
public class Matakuliah26 {  
  
    String kodeMatakuliah;  
  
    String namaMatakuliah;  
  
    int sks;  
  
    int jumlahjam;  
  
  
    public Matakuliah26() {  
  
    }  
  
    public Matakuliah26(String kodeMk, String namaMk, int sks, int jumlahjam) {  
  
        kodeMatakuliah = kodeMk;  
  
        namaMatakuliah = namaMk;  
  
        this.sks = sks;  
  
        this.jumlahjam = jumlahjam;  
  
    }  
  
    void tampilkanInformasi() {  
  
        System.out.println("Kode Matakuliah: " + kodeMatakuliah);  
  
        System.out.println("Nama Matakuliah: " + namaMatakuliah);  
  
        System.out.println("SKS: " + sks);  
  
        System.out.println("Jumlah Jam: " + jumlahjam);  
  
    }  
  
    void ubahSks(int sksBaru){  
  
        sks = sksBaru;  
  
        System.out.println("SKS berhasil diubah menjadi: " + sks);  
  
    }  
  
    public void tambahjam(int jam){  
  
        jumlahjam += jam;  
  
        System.out.println("Jumlah jam berhasil ditambah menjadi: " + jumlahjam);  
  
    }  
  
}
```

```

    }

    public void kurangjam(int jam){
        if (jumlahjam - jam >= 0) {
            jumlahjam -= jam;

            System.out.println("Jumlah jam berhasil dikurangi menjadi: " + jumlahjam);
        } else {
            System.out.println("Jumlah jam tidak bisa dikurangi. Jumlah jam saat ini: " +
jumlahjam);
        }
    }
}

```

Code MataKuliahMain26

```

public class MahasiswaMain26 {

    public static void main(String[] args) {

        Mahasiswa26 mhs1 = new Mahasiswa26();
        mhs1.nama = "Reyhandhika Zikri Prijadi";
        mhs1.nim = "254107020219";
        mhs1.kelas = "TI 1G";
        mhs1.ipk = 3.7;

        mhs1.tampilkanInformasi();
        mhs1.ubahKelas("TI 1H");
        mhs1.UpdateIpk(3.8);
        mhs1.tampilkanInformasi();

        Mahasiswa26 mhs2= new Mahasiswa26("Yudistira", "254107020220", "TI 1G", 3.8);
        mhs2.updateipk(3.30);
        mhs2.tampilkanInformasi();

        Mahasiswa26 mhs3 = new Mahasiswa26("Dimas", "254107020221", "TI 1G", 2.5);
    }
}

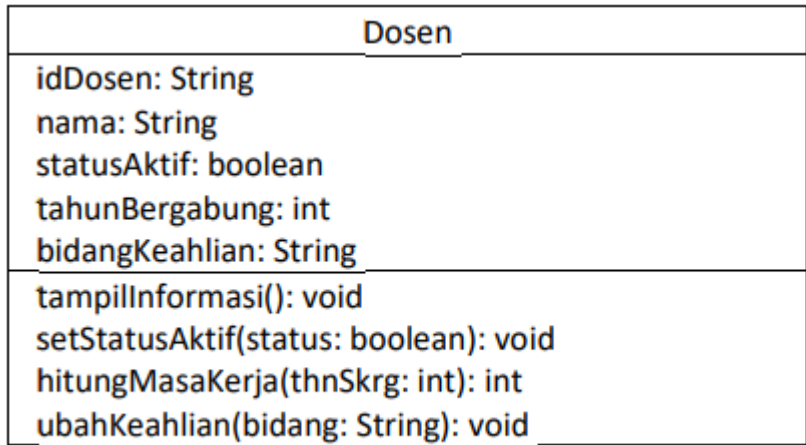
```

```

        mhs3.tampilkanInformasi();
    }
}

```

2. Diberikan class diagram dari class Dosen sebagai berikut



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(thnSkr: int)`: method ini menghitung dan mengembalikan (me-returnkan) masa kerja dosen dalam tahun, berdasarkan tahun bergabung dan tahun saat ini (`thnSkr`) 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.

Code Dosen26

```
public class Dosen26 {

    String idDosen;

    String nama;

    boolean statusAktif;

    int tahunBergabung;

    String bidangKeahlian;

    public Dosen26() {

    }

    public Dosen26(String idDosen, String nm, boolean statusAktif, int tahunBergabung, String bidangKeahlian) {

        this.idDosen = idDosen;

        nama = nm;

        this.statusAktif = statusAktif;

        this.tahunBergabung = tahunBergabung;

        this.bidangKeahlian = bidangKeahlian;

    }

    public void tampilInformasi() {
```

```

        System.out.println("ID Dosen      : " + idDosen);
        System.out.println("Nama Dosen   : " + 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) {
        statusAktif = status;

        System.out.println("Status aktif dosen telah diubah menjadi: " + (statusAktif ? "Aktif" :
        "Tidak Aktif"));
    }

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

    public void ubahKeahlian(String bidang) {
        bidangKeahlian = bidang;

        System.out.println("Bidang keahlian telah diubah menjadi: " + bidangKeahlian);
    }
}

```

Code DosenMain26

```

public class DosenMain26 {

    public static void main(String[] args) {

        Dosen26 dsn1 = new Dosen26();
        dsn1.idDosen = "D001";
        dsn1.nama = "Dr. Budi Santoso";
        dsn1.statusAktif = true;
        dsn1.tahunBergabung = 2010;
        dsn1.bidangKeahlian = "Kecerdasan Buatan";
    }
}

```

```
System.out.println("=== Informasi Dosen 1 (default) ===");  
dsn1.tampilInformasi();  
dsn1.setStatusAktif(false);  
int masaKerja1 = dsn1.hitungMasaKerja(2026);  
System.out.println("Masa kerja: " + masaKerja1 + " tahun");  
dsn1.ubahKeahlian("Machine Learning");  
System.out.println();
```

```
Dosen26 dsn2 = new Dosen26("D002", "Prof. Siti Aminah", true, 2005, "Rekayasa  
Perangkat Lunak");
```

```
System.out.println("=== Informasi Dosen 2 (berparameter) ===");  
dsn2.tampilInformasi();  
dsn2.setStatusAktif(true);  
int masaKerja2 = dsn2.hitungMasaKerja(2026);  
System.out.println("Masa kerja: " + masaKerja2 + " tahun");  
dsn2.ubahKeahlian("DevOps");
```

```
}
```

```
}
```