

**LAPORAN PRAKTIKUM**  
**PEMROGRAM BERORIENTASI OBJEK**  
**MODUL 3**



**Oleh :**

Zoe Satrian

2311103141

S1SI-07-C

**PROGRAM STUDI SISTEM INFORMASI**  
**FAKULTAS REKAYASA INDUSTRI**  
**UNIVERSITAS TELKOM PURWOKERTO**  
**2024**

## A. Studi Kasus

Sistem Informasi Akademik Universitas

## B. Deskripsi

Universitas ingin mengembangkan sistem informasi akademik yang memungkinkan pengelolaan data akademik mahasiswa, dosen, matakuliah, dan pengelolaan KRS (Kartu Rencana Studi) mahasiswa. Fitur utama dari sistem ini adalah sebagai berikut:

1. Manajemen Mahasiswa: Setiap mahasiswa memiliki informasi dasar seperti nama, NIM, prodi, dan daftar KRS yang sudah diambil.
2. Manajemen Dosen: Setiap dosen memiliki nama, NIP, dan daftar matakuliah yang diajarkan.
3. Manajemen Matakuliah: Setiap matakuliah memiliki kode, nama, dan jumlah SKS.
4. Pengelolaan KRS (Kartu Rencana Studi): Mahasiswa dapat memilih dan menghapus matakuliah yang ingin mereka ambil setiap semester.

## C. Guided

1. Buat project baru dengan LatihanP7**Nama**. Nama diganti dengan nama kalian, contoh : *LatihanP7SenaWijayanto*
2. Buatlah program struktur program dasar dengan kelas-kelas berikut:
  - **Person**: Kelas abstrak yang menjadi *superclass* bagi **Mahasiswa** dan **Dosen**.
  - **Mahasiswa**: Kelas turunan dari **Person** yang memiliki informasi KRS.
  - **Dosen**: Kelas turunan dari **Person** yang mengelola daftar matakuliah yang diajarkan.
  - **Matakuliah**: Kelas yang berisi informasi matakuliah, dengan metode **aturJadwal** yang di-overload.
  - **KRS**: Interface yang mengatur pengambilan dan penghapusan matakuliah oleh mahasiswa.

### Kode

Person.java

```
abstract class Person {  
    protected String nama;  
    protected String id;  
  
    public Person(String nama, String id) {  
        this.nama = nama;  
        this.id = id;  
    }  
  
    // Abstract Method (Implementasi khusus di subclass)  
    public abstract void showInfo();  
}
```

Penjelasan : Kode diatas menjelaskan sebuah kelas abstrak bernama "Person" yang memiliki dua variabel instance protected yaitu "nama" dan "id" bertipe String, dilengkapi dengan constructor untuk menginisialisasi kedua variabel tersebut menggunakan parameter yang diterima, serta memiliki satu method abstrak bernama "showInfo()" yang wajib diimplementasikan oleh kelas turunannya (subclass).

#### MataKuliah.java

```
public class MataKuliah {
    private String kode;
    private String namaMatakuliah;
    private int sks;
    private String jadwalHari;
    private String jadwalJam;

    public MataKuliah(String kode, String namaMatakuliah, int sks) {
        this.kode = kode;
        this.namaMatakuliah = namaMatakuliah;
        this.sks = sks;
    }

    // Overloading: Metode aturJadwal dengan dua versi
    public void aturJadwal(String hari, String jam) {
        this.jadwalHari = hari;
        this.jadwalJam = jam;
    }

    public void aturJadwal(String hari) {
        this.jadwalHari = hari;
    }

    public void showInfo() {
        System.out.println("Kode MK: " + kode + ", Nama: " + namaMatakuliah + ", SKS: " + sks);
    }
}
```

Penjelasan : Kelas diatas adalah sebuah blueprint untuk objek mata kuliah yang memiliki lima atribut private yaitu kode, namaMatakuliah, sks, jadwalHari, dan jadwalJam. Kelas ini dilengkapi dengan constructor yang menerima tiga parameter (kode, namaMatakuliah, sks) untuk inisialisasi awal objek. Terdapat dua method overloading untuk mengatur jadwal: method pertama menerima parameter hari dan jam, sedangkan method kedua hanya menerima parameter hari.

## KRS.java

```
interface KRS {  
    void tambahMatakuliah(MataKuliah mk);  
    void hapusMatakuliah(MataKuliah mk);  
}
```

Penjelasan : kelas diatas mendefinisikan dua method abstrak untuk mengelola mata kuliah: method tambahMatakuliah() untuk menambahkan mata kuliah baru yang menerima parameter berupa objek MataKuliah, dan method hapusMatakuliah() untuk menghapus mata kuliah yang juga menerima parameter objek MataKuliah.

## Dosen.java

```
class Dosen extends Person {  
    private String nip;  
    private List<MataKuliah> daftarMatakuliah = new ArrayList<>();  
  
    public Dosen(String nama, String id, String nip) {  
        super(nama, id);  
        this.nip = nip;  
    }  
  
    // Overriding showInfo (Implementasi khusus untuk Dosen)  
    @Override  
    public void showInfo() {  
        System.out.println("Nama Dosen: " + nama + ", NIP: " + nip);  
    }  
  
    // Tambah Matakuliah untuk Dosen  
    public void tambahMatakuliah(MataKuliah mk) {  
        daftarMatakuliah.add(mk);  
    }  
}
```

Penjelasan : Kelas Dosen adalah kelas turunan (subclass) dari kelas abstrak Person yang mengimplementasikan detail khusus untuk objek dosen. Kelas ini memiliki atribut tambahan berupa nip (Nomor Induk Pegawai) dan daftarMatakuliah yang menggunakan ArrayList untuk menyimpan kumpulan mata kuliah yang diampu.

Mahasiswa.java

```
class Mahasiswa extends Person implements KRS {
    private String nim;
    private String prodi;
    private List<MataKuliah> daftarKRS = new ArrayList<>();

    public Mahasiswa(String nama, String id, String nim, String prodi) {
        super(nama, id);
        this.nim = nim;
        this.prodi = prodi;
    }

    // Overriding showInfo (Implementasi khusus untuk Mahasiswa)
    @Override
    public void showInfo() {
        System.out.println("Nama Mahasiswa: " + nama + ", NIM: " + nim + ", Prodi: " + prodi);
    }

    // Implementasi metode interface untuk tambah dan hapus matakuliah
    @Override
    public void tambahMatakuliah(MataKuliah mk) {
        daftarKRS.add(mk);
    }

    @Override
    public void hapusMatakuliah(MataKuliah mk) {
        daftarKRS.remove(mk);
    }
}
```

Penjelasan : Kelas Mahasiswa merupakan turunan dari kelas Person dan mengimplementasikan interface KRS. Kelas ini memiliki atribut tambahan berupa nim, prodi, dan daftarKRS yang menggunakan ArrayList untuk menyimpan mata kuliah yang diambil.

## LatihanP7ZoeSatrian.java

```
public class Main {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // Membuat Objek Mahasiswa dan Dosen  
        Mahasiswa mhs = new Mahasiswa("Zoe Satrian", "123", "2311103141", "Sistem Informasi");  
        Dosen dosen = new Dosen("Sena Wijayanto", "456", "D001");  
  
        // Membuat Objek Matakuliah  
        MataKuliah mk1 = new MataKuliah("SI101", "PBO", 3);  
        MataKuliah mk2 = new MataKuliah("SI102", "Basis Data", 3);  
  
        // Mengatur Jadwal Matakuliah (Overloading)  
        mk1.aturJadwal("Senin", "08:00");  
        mk2.aturJadwal("Selasa");  
  
        // Menambahkan Matakuliah ke KRS Mahasiswa  
        mhs.tambahMatakuliah(mk1);  
        mhs.tambahMatakuliah(mk2);  
  
        // Menampilkan Informasi  
        mhs.showInfo(); // Menampilkan informasi Mahasiswa (Polimorfisme)  
        dosen.showInfo(); // Menampilkan informasi Dosen (Polimorfisme)  
        mk1.showInfo(); // Informasi Matakuliah 1  
        mk2.showInfo(); // Informasi Matakuliah 2  
    }  
}
```

Penjelasan : kode diatas merupakan implementasi utama dari sistem akademik sederhana yang dimulai dengan pembuatan objek Mahasiswa bernama Zoe Satrian dan Dosen bernama Sena Wijayanto. Program kemudian membuat dua objek MataKuliah yaitu PBO dan Basis Data, masing-masing dengan kode dan jumlah SKS yang telah ditentukan. Jadwal untuk kedua mata kuliah diatur menggunakan method overloading, dimana PBO dijadwalkan pada Senin pukul 08:00 dan Basis Data dijadwalkan pada hari Selasa. Kedua mata kuliah tersebut kemudian ditambahkan ke dalam KRS mahasiswa menggunakan method tambahMatakuliah(). Program diakhiri dengan menampilkan informasi menggunakan method showInfo() untuk masing-masing objek yang mendemonstrasikan konsep polimorfisme, dimana method yang sama menghasilkan output berbeda sesuai dengan tipe objeknya (mahasiswa menampilkan NIM dan prodi, dosen menampilkan NIP, dan mata kuliah menampilkan kode dan SKS).

## D. Unguided

Kembangkan Sistem Informasi Akademik dengan menambahkan fitur berikut:

1. Menampilkan Daftar Mata Kuliah yang Diambil oleh Mahasiswa.  
Tambahkan metode **showKRS()** dalam kelas **Mahasiswa** yang menampilkan seluruh mata kuliah yang ada dalam daftar KRS mahasiswa tersebut.

Mahasiswa.java

```
package latihanp7zoestrian;

/**
 * @author Zoe Satrian / 2311103141
 */
import java.util.ArrayList;
import java.util.List;

class Mahasiswa extends Person implements KRS {

    private String nim;
    private String prodi;
    private List<MataKuliah> daftarKRS = new ArrayList<>();

    public Mahasiswa(String nama, String id, String nim, String prodi) {
        super(nama, id);
        this.nim = nim;
        this.prodi = prodi;
    }

    // Overriding showInfo (Implementasi khusus untuk Mahasiswa)
    @Override
    public void showInfo() {
        System.out.println("Nama Mahasiswa: " + nama + ", NIM: " + nim + ", Prodi: " +
prodi);
    }

    // Implementasi metode interface untuk tambah dan hapus matakuliah
    @Override
    public void tambahMatakuliah(MataKuliah mk) {
```

```
        daftarKRS.remove(mk);
    }

    public void showKRS() {
        System.out.println("");
        System.out.println("===== Daftar KRS Mahasiswa " + nama + "=====");
        if (daftarKRS.isEmpty()) {
            System.out.println("Tidak ada mata kuliah yang diambil.");
        } else {
            for (MataKuliah mk : daftarKRS) {
                mk.showInfo(); // Memanggil showInfo() dari kelas MataKuliah
            }
        }
        System.out.println("Total SKS: " + hitungTotalSKS()); // Menampilkan total SKS
    }
}
}
```

Penjelasan : kodingan diatas menambahkan metode showKRS(), metode ini mencetak judul yang menunjukkan nama mahasiswa. berikutnya, metode melakukan pengecekan apakah daftar mata kuliah (daftarKRS) kosong. Jika kosong, program akan mencetak pesan yang menyatakan bahwa tidak ada mata kuliah yang diambil, memberi tahu pengguna bahwa mahasiswa tersebut belum mendaftar untuk mata kuliah apa pun. Sebaliknya, jika daftar tidak kosong, metode akan menggunakan loop for-each untuk iterasi melalui setiap objek MataKuliah dalam daftar tersebut.



## LatihanP7ZoeSatrian.java

```
package latihanp7zoesatrian;

/**
 * @author Zoe Satrian / 2311103141
 */
public class LatihanP7ZoeSatrian {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Membuat Objek Mahasiswa dan Dosen
        System.out.println("===== Informasi Mahasiswa =====");
        Mahasiswa mhs = new Mahasiswa("Zoe Satrian", "123", "2311103141", "Sistem
Informasi");
        Dosen dosen = new Dosen("Sena Wijayanto", "456", "D001");

        // Membuat Objek Matakuliah
        MataKuliah mk1 = new MataKuliah("SI101", "PBO", 3);
        MataKuliah mk2 = new MataKuliah("SI102", "Basis Data", 3);
        MataKuliah mk3 = new MataKuliah("SI103", "Jaringan Komputer", 3);

        // Mengatur Jadwal Matakuliah (Overloading)
        mk1.aturJadwal("Senin", "08:00");
        mk2.aturJadwal("Selasa");

        // Menambahkan Matakuliah ke KRS Mahasiswa
        mhs.tambahMatakuliah(mk1);
        mhs.tambahMatakuliah(mk2);
        mhs.tambahMatakuliah(mk3);

        // Menampilkan Informasi
        mhs.showInfo(); // Menampilkan informasi Mahasiswa (Polimorfisme)
        dosen.showInfo(); // Menampilkan informasi Dosen (Polimorfisme)
        mhs.showKRS(); // Menampilkan daftar mata kuliah
    }
}
```

Penjelasan : Pada file LatihanP7ZoeSatrian hanya memanggil showKRS() untuk menampilkan daftarmata kuliah yang diambil mahasiswa.

## Output

```
Output - LatihanP7ZoeSatrian (run)

run:
===== Informasi Mahasiswa =====
Nama Mahasiswa: Zoe Satrian, NIM: 2311103141, Prodi: Sistem Informasi

===== Informasi Dosen =====
Nama Dosen: Sena Wijayanto, NIP: D001

===== Daftar KRS Mahasiswa Zoe Satrian=====
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3
Kode MK: SI103, Nama: Jaringan Komputer, SKS: 3
BUILD SUCCESSFUL (total time: 0 seconds)
```

## 2. Menghitung Total SKS yang Diambil oleh Mahasiswa.

Tambahkan metode **hitungTotalSKS()** dalam kelas **Mahasiswa** untuk menghitung total SKS dari seluruh mata kuliah yang telah diambil.

Mahasiswa.java

```
package LatihanP7ZoeSatrian;

import java.util.ArrayList;
import java.util.List;

class Mahasiswa extends Person implements KRS {

    private String nim;
    private String prodi;
    private List<MataKuliah> daftarKRS = new ArrayList<>();

    public Mahasiswa(String nama, String id, String nim, String prodi) {
        super(nama, id);
        this.nim = nim;
        this.prodi = prodi;
    }

    // Overriding showInfo (Implementasi khusus untuk Mahasiswa)
    @Override
    public void showInfo() {
```

```
        System.out.println("Nama Mahasiswa: " + nama + ", NIM: " + nim + ", Prodi: " + prodi);
    }

    // Implementasi metode interface untuk tambah dan hapus matakuliah
    @Override
    public void tambahMatakuliah(MataKuliah mk) {
        daftarKRS.add(mk);
    }

    @Override
    public void hapusMatakuliah(MataKuliah mk) {
        daftarKRS.remove(mk);
    }

    // Metode untuk menghitung total SKS
    public int hitungTotalSKS() {
        int totalSks = 0; // Inisialisasi total SKS
        for (MataKuliah mk : daftarKRS) {
            totalSks += mk.getSks(); // Menggunakan getter untuk menambahkan SKS
        }
        return totalSks; // Mengembalikan total SKS
    }

    public void showKRS() {
        System.out.println("");
        System.out.println("===== Daftar KRS Mahasiswa " + nama + "=====");
        if (daftarKRS.isEmpty()) {
            System.out.println("Tidak ada mata kuliah yang diambil.");
        } else {
            for (MataKuliah mk : daftarKRS) {
                mk.showInfo(); // Memanggil showInfo() dari kelas MataKuliah
            }
            System.out.println("Total SKS: " + hitungTotalSKS()); // Menampilkan total SKS
        }
    }
}
```

Penjelasan : Dalam kodingan diatas menambahkan metode `hitungTotalSKS()` metode ini untuk menghitung total SKS dari seluruh mata kuliah yang diambil oleh mahasiswa.

## MataKuliah.java

```
package latihanp7zoesatrian;

/**
 * @author Zoe Satrian / 2311103141
 */
public class MataKuliah {

    private String kode;
    private String namaMatakuliah;
    private int sks;
    private String jadwalHari;
    private String jadwalJam;

    public MataKuliah(String kode, String namaMatakuliah, int sks) {
        this.kode = kode;
        this.namaMatakuliah = namaMatakuliah;
        this.sks = sks;
    }

    // Getter untuk sks
    public int getSks() {
        return sks;
    }

    // Overloading: Metode aturJadwal dengan dua versi
    public void aturJadwal(String hari, String jam) {
        this.jadwalHari = hari;
        this.jadwalJam = jam;
    }

    public void aturJadwal(String hari) {
        this.jadwalHari = hari;
    }

    public void showInfo() {
        System.out.println("Kode MK: " + kode + ", Nama: " + namaMatakuliah + ", SKS: " +
sks);
    }
}
```

Penjelasan : Kodingan diatas menambahkan metode `getSks()`, metode `getSks()` sebuah *getter* yang dirancang untuk memberikan akses ke nilai atribut `sks` di dalam kelas `MataKuliah`. Atribut `sks`, yang menyimpan jumlah sks dari suatu mata kuliah, dideklarasikan sebagai `private`, yang berarti bahwa nilai ini tidak dapat diakses langsung dari luar kelas tersebut.

## Output

Output - LatihanP7ZoeSatrian (run)

```
run:
===== Informasi Mahasiswa =====
Nama Mahasiswa: Zoe Satrian, NIM: 2311103141, Prodi: Sistem Informasi

===== Informasi Dosen =====
Nama Dosen: Sena Wijayanto, NIP: D001

===== Daftar KRS Mahasiswa Zoe Satrian=====
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3
Kode MK: SI103, Nama: Jaringan Komputer, SKS: 3
Total SKS: 9
BUILD SUCCESSFUL (total time: 0 seconds)
```