

**LAPORAN PRAKTIKUM**  
**PEMROGRAMAN BERORIENTASI OBJEK**  
**MODUL III**  
**Konsep Inheritance, Abstract, dan Interface**



Oleh :  
FACHRULLAH ZHAFRAN LISTIYONO  
2211103115  
SI-07-C

**PROGRAM STUDI S1 SISTEM INFORMASI**  
**FAKULTAS REKAYASA INDUSTRI**  
**UNIVERSITAS TELKOM PURWOKERTO**  
**2024**

## **Studi Kasus : Sistem Informasi Akademik Universitas**

### **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.

## **I. GUIDED**

### **Person.java**

#### **Source Code:**

```
package LatihanP7FachrullahZhafranListiyono;

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:**

Kelas Person adalah kelas abstrak yang berfungsi sebagai cetakan dasar untuk entitas dengan properti nama dan id. Kelas ini memiliki constructor untuk menginisialisasi nama dan ID dari objek yang dibuat. Kelas ini juga memiliki metode abstrak showInfo() yang akan diimplementasikan di subclass (kelas turunan), sehingga setiap kelas turunan dapat menunjukkan informasi dengan cara yang khusus sesuai kebutuhan, seperti mahasiswa dan dosen.

### **MataKuliah.java**

#### **Source Code:**

```
package LatihanP7FachrullahZhafranListiyono;

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 MataKuliah merepresentasikan informasi tentang mata kuliah, dengan properti seperti kode (kode mata kuliah), namaMatakuliah (nama mata kuliah), dan sks (jumlah SKS mata kuliah). Metode aturJadwal diimplementasikan dengan overload, sehingga bisa diisi dengan dua parameter (hari dan jam) atau hanya satu parameter (hari). Selain itu, metode showInfo menampilkan informasi mata kuliah yang mencakup kode, nama, dan SKS untuk membantu melihat informasi terkait setiap mata kuliah.

### KRS.java

#### Source Code:

```

package LatihanP7FachrullahZhafranListiyono;

interface KRS {

    void tambahMatakuliah(MataKuliah mk);

    void hapusMatakuliah(MataKuliah mk);
}

```

### Penjelasan:

KRS adalah interface yang mengatur kontrak untuk mengelola KRS mahasiswa, dengan dua metode, yaitu tambahMatakuliah() dan hapusMatakuliah(). Interface ini memastikan bahwa kelas yang mengimplementasinya harus menyediakan cara untuk menambahkan dan menghapus mata kuliah dari daftar KRS mahasiswa, sehingga setiap kelas yang mengikuti kontrak ini bisa mendukung fitur pengelolaan KRS.

### Dosen.java

#### Source Code:

```

package LatihanP7FachrullahZhafranListiyono;

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

```

```

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 subclass dari Person yang menyimpan informasi khusus untuk dosen, seperti nip dan daftar mata kuliah yang diajarkan. Constructor kelas ini menggunakan super untuk menginisialisasi nama dan id dari Person, lalu menyimpan NIP dosen. Kelas ini juga mengimplementasikan metode showInfo() dengan overriding untuk menampilkan informasi khusus dosen, dan menyediakan metode tambahMatakuliah yang berfungsi untuk menambah mata kuliah yang diajarkan dosen ke dalam daftar daftarMatakuliah.

### Mahasiswa.java

#### Source Code:

```

package LatihanP7FachrullahZhafranListiyono;

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);
    }
}

```

### Penjelasan:

Kelas Mahasiswa adalah subclass dari Person yang merepresentasikan mahasiswa dengan tambahan atribut nim, prodi, dan daftarKRS. Constructor-nya memanfaatkan super untuk mewarisi properti nama dan id dari Person. Kelas ini juga mengimplementasikan interface KRS sehingga memiliki metode tambahMatakuliah() dan hapusMatakuliah() yang digunakan untuk menambah atau menghapus mata kuliah dari daftar KRS. Metode showInfo() di sini di-override untuk menampilkan detail mahasiswa seperti nama, NIM, dan prodi.

### Main.java

#### Source Code:

```

package LatihanP7FachrullahZhafranListiyono;

public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Membuat Objek Mahasiswa dan Dosen
        Mahasiswa mhs = new Mahasiswa("Ci rusdi", "123",
"2211109001", "Sistem Informasi");
        Dosen dosen = new Dosen("Ambacik", "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:

Main adalah kelas utama yang menjalankan program. Di sini dibuat objek Mahasiswa dan Dosen, serta beberapa mata kuliah. Jadwal mata kuliah diatur menggunakan metode `aturJadwal` dengan overload untuk menunjukkan fleksibilitas penjadwalan. Kelas ini juga menambahkan mata kuliah ke daftar KRS mahasiswa dan menampilkan informasi mahasiswa, dosen, dan mata kuliah, menggunakan konsep polimorfisme dari metode `showInfo()`.

### Screenshot Output:

```
run:
Nama Mahasiswa: Ci rusdi, NIM: 2211109001, Prodi: Sistem Informasi
Nama Dosen: Ambacik, NIP: D001
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3
BUILD SUCCESSFUL (total time: 0 seconds)
```

## II. 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.
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.

**Jawaban :**

1. Fitur menambahkan daftar mata kuliah yang diambil oleh mahasiswa ditambahkan melalui metode **showKRS()** dalam kelas **Mahasiswa**. Metode ini akan menampilkan daftar mata kuliah yang terdapat pada KRS mahasiswa. Setiap mata kuliah yang ada dalam daftar **daftarKRS** akan ditampilkan melalui pemanggilan metode **showInfo()** dari kelas **MataKuliah**, yang sudah diimplementasikan untuk menampilkan informasi kode, nama mata kuliah, jumlah SKS, serta jadwalnya.
2. Fitur kedua yaitu menghitung total sks yang diambil oleh mahasiswa ditambahkan dengan membuat metode **hitungTotalSKS()** dalam kelas **Mahasiswa**. Metode ini bertujuan untuk menghitung total SKS dari seluruh mata kuliah yang diambil mahasiswa. Dengan menggunakan perulangan **for**, metode ini menjumlahkan SKS dari setiap mata kuliah yang ada dalam daftar **daftarKRS**.

### Person.java

#### Source Code:

```
package LatihanP7FachrullahZhafranListiyono;

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:

Kelas **Person** adalah kelas abstrak yang berfungsi sebagai cetakan dasar untuk entitas dengan properti **nama** dan **id**. Kelas ini memiliki constructor untuk menginisialisasi **nama** dan **ID** dari objek yang dibuat. Kelas ini juga memiliki metode abstrak **showInfo()** yang akan diimplementasikan di subclass (kelas turunan), sehingga setiap kelas turunan dapat menunjukkan informasi dengan cara yang khusus sesuai kebutuhan, seperti mahasiswa dan dosen.

### MataKuliah.java

#### Source Code:

```
package LatihanP7FachrullahZhafranListiyono;
```

```

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
            + ", Jadwal: " + jadwalHari + (jadwalJam != null ? "
" + jadwalJam : ""));
    }
}

```

### Penjelasan:

Kelas MataKuliah adalah kelas yang menyimpan informasi tentang mata kuliah, seperti kode mata kuliah, nama mata kuliah, jumlah SKS, dan jadwal. Di kelas ini, terdapat constructor untuk mengisi data awal seperti kode, namaMatakuliah, dan sks. Selain itu, ada metode aturJadwal() yang di-overload dengan dua versi: satu untuk mengatur jadwal hari dan jam, dan satu lagi hanya untuk mengatur hari saja. Metode showInfo() digunakan untuk menampilkan informasi lengkap mata kuliah, termasuk jadwal yang telah diatur.

### KRS.java

#### Source Code:

```

package LatihanP7FachrullahZhafranListiyono;

interface KRS {

    void tambahMatakuliah(MataKuliah mk);

    void hapusMatakuliah(MataKuliah mk);
}

```



```
}
```

**Penjelasan:**

KRS adalah interface yang mengatur kontrak untuk mengelola KRS mahasiswa, dengan dua metode, yaitu `tambahMatakuliah()` dan `hapusMatakuliah()`. Interface ini memastikan bahwa kelas yang mengimplementasinya harus menyediakan cara untuk menambahkan dan menghapus mata kuliah dari daftar KRS mahasiswa, sehingga setiap kelas yang mengikuti kontrak ini bisa mendukung fitur pengelolaan KRS.

**Dosen.java****Source Code:**

```
package LatihanP7FachrullahZhafranListiyono;

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

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:**

elas Dosen adalah turunan dari kelas Person dan dirancang untuk menyimpan informasi khusus dosen. Selain nama dan id yang diwarisi dari Person, kelas ini memiliki atribut tambahan nip dan daftar `daftarMatakuliah` untuk menyimpan mata kuliah yang diajarkan oleh dosen. Constructor kelas ini menggunakan super untuk memanggil constructor Person, lalu mengisi nip. Kelas ini juga memiliki metode `showInfo()` yang telah di-*override* untuk menampilkan informasi dosen, serta metode `tambahMatakuliah()` yang digunakan untuk menambahkan mata kuliah ke dalam daftar `daftarMatakuliah`.

**Mahasiswa.java****Source Code:**

```
package LatihanP7FachrullahZhafranListiyono;

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

```

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);
    }

    // Menampilkan daftar mata kuliah yang diambil oleh mahasiswa
    public void showKRS() {
        System.out.println("Daftar Mata Kuliah yang diambil oleh " +
nama + ":");
        for (MataKuliah mk : daftarKRS) {
            mk.showInfo();
        }
    }

    // Menghitung total SKS yang diambil oleh mahasiswa
    public int hitungTotalSKS() {
        int totalSKS = 0;
        for (MataKuliah mk : daftarKRS) {
            totalSKS += mk.getSks();
        }
        return totalSKS;
    }
}

```

### Penjelasan:

Kelas Mahasiswa adalah turunan dari kelas Person yang juga mengimplementasikan interface KRS. Kelas ini menyimpan informasi tambahan berupa nim, prodi, dan daftar daftarKRS yang berisi mata kuliah yang diambil oleh mahasiswa. Metode tambahMatakuliah() dan hapusMatakuliah() diimplementasikan sesuai kontrak dari

interface KRS, sehingga mahasiswa dapat menambah atau menghapus mata kuliah dari KRS-nya. Selain itu, terdapat metode showKRS() yang menampilkan daftar mata kuliah yang telah diambil, serta metode hitungTotalSKS() yang menghitung jumlah total SKS dari semua mata kuliah dalam daftar daftarKRS.

### Main.java

#### Source Code:

```
package LatihanP7FachrullahZhafranListiyono;

public class Main {

    public static void main(String[] args) {
        // Membuat Objek Mahasiswa dan Dosen
        Mahasiswa mhs = new Mahasiswa("Budi", "123", "2211109001",
        "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");
        mk3.aturJadwal("Rabu", "10.00");

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

        // Menampilkan informasi Mahasiswa
        mhs.showInfo();

        // Menampilkan daftar MataKuliah (KRS) yang diambil Mahasiswa
        mhs.showKRS();

        // Menghitung dan menampilkan total SKS yang diambil
        Mahasiswa
        System.out.println("Total SKS yang diambil: " +
        mhs.hitungTotalSKS());
    }
}
```

### Penjelasan:

Kelas Main adalah tempat utama untuk menjalankan program. Di sini, objek-objek mahasiswa dan dosen dibuat, serta beberapa mata kuliah ditambahkan sebagai contoh. Setelah mahasiswa mengambil beberapa mata kuliah dengan menggunakan metode tambahMatakuliah(), program ini akan menampilkan informasi mahasiswa, daftar mata kuliah yang diambil, dan menghitung total SKS yang telah diambil mahasiswa. Kelas ini berfungsi sebagai tempat percobaan untuk memastikan semua fitur yang diinginkan berjalan sesuai dengan keinginan.

### Screenshot Output:

---

```
run:
Nama Mahasiswa: Budi, NIM: 2211109001, Prodi: Sistem Informasi
Daftar Mata Kuliah yang diambil oleh Budi:
Kode MK: SI101, Nama: PBO, SKS: 3, Jadwal: Senin 08:00
Kode MK: SI102, Nama: Basis Data, SKS: 3, Jadwal: Selasa
Kode MK: SI103, Nama: Jaringan Komputer, SKS: 3, Jadwal: Rabu 10.00
Total SKS yang diambil: 9
BUILD SUCCESSFUL (total time: 0 seconds)
```