

**LAPORAN PRAKTIKUM**  
**PEMROGRAMAN BERORIENTASI OBJEK**  
**MODUL 3**  
**MATERI**



Oleh :  
Muhammad Raflan Kemal  
2311103132  
S1SI-07-C

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

## I. GUIDED

1. Buat project baru dengan LatihanP7Nama. Nama diganti dengan nama kalian, contoh : *LatihanP7SenaWijayanto*
2. Buatlah program struktur program dasar dengan kelas-kelas berikut:
  - a. Person: Kelas abstrak yang menjadi *superclass* bagi Mahasiswa dan Dosen.
  - b. Mahasiswa: Kelas turunan dari Person yang memiliki informasi KRS.
  - c. Dosen: Kelas turunan dari Person yang mengelola daftar matakuliah yang diajarkan.
  - d. Matakuliah: Kelas yang berisi informasi matakuliah, dengan metode *aturJadwal* yang di-overload.
  - e. KRS: Interface yang mengatur pengambilan dan penghapusan matakuliah oleh mahasiswa.

Kode Program

a. Person.java

```
package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */
abstract class Person {

    protected String nama;
    protected String id;

    public Person(String nama, String id) {
        this.nama = nama;
        this.id = id;
    }

    public abstract void showInfo();
}
```

b. Mahasiswa.java

```
package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
```

```

* 07C
*/

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;
    }

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

    @Override
    public void tambahMatakuliah(Matakuliah mk) {
        daftarKRS.add(mk);
    }

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

```

c. Dosen.java

```

package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */

```

```

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;
    }

    @Override
    public void showInfo() {
        System.out.println("Nama Dosen: " + nama + ", NIP: " + nip);
    }

    public void tambahMatakuliah(Matakuliah mk) {
        daftarMatakuliah.add(mk);
    }
}

```

d. Matakuliah.java

```

package latihanp7raflankemal;

/**
 *
 * @author Muhammad RAflan Kemal
 * 2311103132
 * 07C
 */
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;
    }
    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);
    }
}

```

e. KRS.java

```

package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */
interface KRS {

    void tambahMatakuliah(MataKuliah mk);

    void hapusMatakuliah(MataKuliah mk);
}

```

f. Main.java

```

package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */
public class Main {

```

```
/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    Mahasiswa mhs = new Mahasiswa("Muhammad Raflan Kemal", "123",
"2211109001", "Sistem Informasi");
    Dosen dosen = new Dosen("Adzana Ashel", "456", "D001");

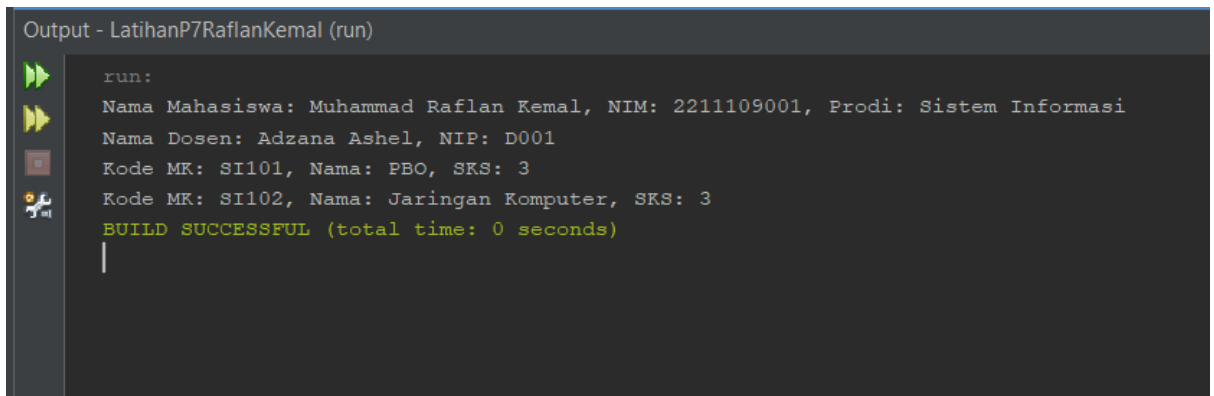
    MataKuliah mk1 = new MataKuliah("SI101", "PBO", 3);
    MataKuliah mk2 = new MataKuliah("SI102", "Jaringan Komputer", 3);

    mk1.aturJadwal("Senin", "08:00");
    mk2.aturJadwal("Selasa");

    mhs.tambahMatakuliah(mk1);
    mhs.tambahMatakuliah(mk2);

    mhs.showInfo();
    dosen.showInfo();
    mk1.showInfo();
    mk2.showInfo();
}
}
```

## Hasil Output :

A screenshot of an IDE's output window titled "Output - LatihanP7RaflanKemal (run)". The window has a dark background with a light-colored border. On the left side of the window, there are four icons: a green play button, a yellow play button, a red square, and a white icon with a red circle. The output text is as follows:

```
run:
Nama Mahasiswa: Muhammad Raflan Kemal, NIM: 2211109001, Prodi: Sistem Informasi
Nama Dosen: Adzana Ashel, NIP: D001
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Jaringan Komputer, SKS: 3
BUILD SUCCESSFUL (total time: 0 seconds)
```

## Penjelasan :

Pada kode program di atas terdiri dari berbagai kelas yang saling terkait. Pertama, ada kelas abstrak `Person` yang menjadi dasar bagi dua kelas turunan, yaitu `Mahasiswa` dan `Dosen`. Kelas `Person` menyimpan informasi dasar seperti nama dan ID, serta memiliki metode `showInfo()` untuk menampilkan data tersebut. Di sisi lain, kelas `Mahasiswa` menambahkan detail seperti NIM, program studi, dan daftar mata kuliah yang diambil, serta mengimplementasikan interface `KRS` untuk mengelola penambahan dan penghapusan mata kuliah. Kelas `Dosen` bertugas mengatur mata kuliah yang diajarkan. Selain itu, kelas `MataKuliah` menyimpan informasi tentang setiap mata kuliah, termasuk kode dan jadwal, serta memiliki metode overload untuk pengaturan jadwal. Dalam kelas `Main`, objek-objek ini diinstansiasi untuk menunjukkan interaksi antara kelas-kelas tersebut dan menampilkan informasi mengenai mahasiswa, dosen, dan mata kuliah melalui metode `showInfo()`. Secara keseluruhan, sistem ini memberikan gambaran yang jelas tentang pengorganisasian informasi akademik.

## 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.

Kode Program

update Mahasiswa.java

```
package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */

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;
    }

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

    @Override
    public void tambahMatakuliah(MataKuliah mk) {
        daftarKRS.add(mk);
    }

    @Override
```



```

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

public void showKRS() {
    System.out.println("Daftar Mata Kuliah yang Diambil oleh " + nama + ":");
    for (MataKuliah mk : daftarKRS) {
        mk.showInfo();
    }
}

public int hitungTotalSKS() {
    int totalSKS = 0;
    for (MataKuliah mk : daftarKRS) {
        totalSKS += mk.getSks();
    }
    return totalSKS;
}
}

```

update MataKuliah.java

```

package latihanp7raflankemal;

/**
 *
 * @author Muhammad RAflan Kemal
 * 2311103132
 * 07C
 */
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;
    }

    public int getSks() {
        return sks;
    }
}

```

```

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);
}
}

```

update Main.java

```

package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */
public class Main {
    public static void main(String[] args) {
        Mahasiswa mhs = new Mahasiswa("Muhammad Raflan Kemal", "123",
"2311103132", "Sistem Informasi");
        Dosen dosen = new Dosen("Adzana Ashel", "456", "D001");

        Matakuliah mk1 = new Matakuliah("SI101", "PBO", 3);
        Matakuliah mk2 = new Matakuliah("SI102", "Jaringan Komputer", 3);

        mk1.aturJadwal("Senin", "08:00");
        mk2.aturJadwal("Selasa");

        mhs.tambahMatakuliah(mk1);
        mhs.tambahMatakuliah(mk2);

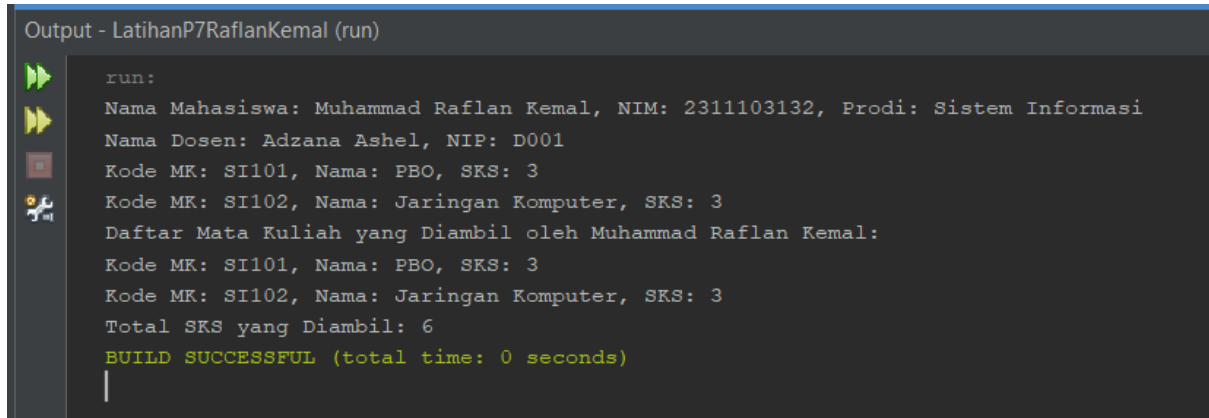
        mhs.showInfo();
        dosen.showInfo();
        mk1.showInfo();
        mk2.showInfo();

        mhs.showKRS();
    }
}

```

```
        int totalSKS = mhs.hitungTotalSKS();
        System.out.println("Total SKS yang Diambil: " + totalSKS);
    }
}
```

## Hasil Output

The image shows a screenshot of a Java IDE's output window. The title bar reads "Output - LatihanP7RaflanKemal (run)". On the left side, there are icons for running (a green play button), stepping through (a yellow play button), and debugging (a red bug icon). The output text is as follows:

```
run:
Nama Mahasiswa: Muhammad Raflan Kemal, NIM: 2311103132, Prodi: Sistem Informasi
Nama Dosen: Adzana Ashel, NIP: D001
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Jaringan Komputer, SKS: 3
Daftar Mata Kuliah yang Diambil oleh Muhammad Raflan Kemal:
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Jaringan Komputer, SKS: 3
Total SKS yang Diambil: 6
BUILD SUCCESSFUL (total time: 0 seconds)
```

## Penjelasan :

Pada kode program unguide ini saya menambahkan dua fitur. Yang pertama adalah metode `showKRS()`, yang memungkinkan mahasiswa untuk melihat daftar mata kuliah yang mereka ambil. Metode ini berfungsi dengan mengiterasi daftar mata kuliah dan menampilkan informasi tentang masing-masing. Yang kedua, saya juga memperkenalkan metode `hitungTotalSKS()`, yang bertugas menghitung total SKS dari semua mata kuliah yang diambil. Dengan memanfaatkan getter `getSks()` di kelas `MataKuliah`, metode ini dapat dengan mudah menjumlahkan semua SKS. Di dalam kelas `Main`, saya mengintegrasikan kedua metode ini sehingga ketika program dijalankan, mahasiswa dapat melihat informasi diri, dosen, mata kuliah, serta total SKS yang telah mereka ambil. Penambahan ini membuat sistem lebih fungsional dan membantu mahasiswa dalam merencanakan studi mereka dengan lebih efektif.