

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 3
LATIHAN KONSEP PBO



Oleh :

Rizqi Dwi Saputra

2311103133

S1SI – 07 – C

PROGRAM STUDI S1 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

```
package latihanp7rizqidwisaputra;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

abstract class Person {

    protected String nama;
    protected String id;

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

    // Abstarct method (Implmentasi khusus di SubClass)
    public abstract void ShowInfo();
}
```

```
}
```

MataKuliah.java

```
package latihanp7rizqidwisaputra;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

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);
    }

}
```

KRS.java

```
package latihanp7rizqidwisaputra;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

interface KRS {

    void tambahMataKuliah(MataKuliah mk);
    void hapusMataKuliah (MataKuliah mk);

}
```

Dosen.java

```
package latihanp7rizqidwisaputra;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

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);
    }

}

```

Mahasiswa.java

```

package latihanp7rizqidwisaputra;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

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);
    }

}

```

```
}
```

Main.java

```
package latihanp7rizqidwisaputra;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

public class Main {

    public static void main(String[] args) {

        //      Membuat objek Mahasiswa dan Dosen
        Mahasiswa mhs = new Mahasiswa("Rizqi Dwi Saputra", "001",
        "2311103133", "Sistem Informasi");
        Dosen dosen = new Dosen("Sena Wijayanto", "456", "D001");

        //      Membuat objek Mata Kuliah
        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

    }

}
```

Hasil Run Project

```
Nama Mahasiswa: Rizqi Dwi Saputra, NIM: 2311103133, Prodi: Sistem Informasi
Nama Dosen: Sena Wijayanto, NIP: D001
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3
```

Penjelasan:

Terdapat 5 kelas dan 1 file utama pada package latihanp7rizqidwisaputra yaitu,

1. Dosen.java
2. KRS.java

3. Mahasiswa.java

4. MataKuliah.java

5. Person.java

File utama

1. Main.java

1. Dosen.java

- Merupakan kelas turunan dari kelas Person yang didalamnya terdapat variabel 'nip' untuk menyimpan data NIP dosen, dan ArrayList untuk menyimpan data Matakuliah yang diampu.
- Terdapat konstruktor dengan 3 parameter yaitu nama, id, nip. Kemudian memanggil konstruktor dari superclass nya yaitu kelas Person untuk menginisialisasi nilai nama dan id. Kemudian, menginisialisasi nilai nip.
- Overriding method dari superclass yaitu kelas Person untuk menampilkan informasi nama dan nip Dosen.
- Method tambahMataKuliah(MataKuliah mk) berfungsi untuk menambahkan MataKuliah baru kedalam daftar matakuliah dosen menggunakan perintah add() dari ArrayList

2. KRS.java

Merupakan kelas Interface yang didalamnya terdapat method

- tambahMataKuliah (MataKuliah mk) yang berfungsi untuk menambahkan matakuliah kedalam KRS
- hapusMataKuliah (MataKuliah mk) yang berfungsi untuk menghapus matakuliah dari KRS
- Kedua method diimplementasikan pada kelas lain

3. Mahasisa.java

Merupakan kelas turunan dari kelas Person dan mengimplementasikan kelas KRS, didalamnya terdapat variabel private nim, prodi dan ada ArrayList. Perlu digaris bawahi bahwa variable private bertujuan agar variabel tersebut hanya dapat diakses dari dalam objeknya sendiri.

- Terdapat konstruktor dengan 4 parameter yaitu nama, id, nim, prodi. Kemudian memanggil konstruktor dari superclass nya yaitu kelas Person untuk menginisialisasi nama dan id, kemudian menginisialisasi nim dan prodi.
- Overriding method showInfo dari superclass (Person) untuk menampilkan informasi Nama mahasiswa, NIM, dan Prodi

- Overriding method tambahMatakuliah dari kelas interface (KRS) untuk menambahkan MataKuliah
- Overriding method hapusMatakuliah dari kelas interface (KRS) untuk menghapus MataKuliah

4. MataKuliah.java

Merupakan kelas biasa yang didalamnya terdapat 5 variabel yakni kode, namaMataKuliah, sks, jadwalHari dan jadwalJam.

- Terdapat konstruktor dengan 3 parameter yakni kode, namaMataKuliah dan sks, kemudian menginisialisasi kode, namaMataKuliah dan sks.
- Terdapat method aturJadwal yang di-overload, artinya akan mungkin terdapat lebih dari satu method dengan nama yang sama.
- Method aturJadwal pertama menginisialisasi hari dan jam
- Method aturJadwal kedua menginisialisasi hari saja, tidak ada jam
- Method showInfo untuk menampilkan informasi Kode MataKuliah, Nama MataKuliah dan SKS nya

5. Person.java

Kelas Person merupakan kelas abstrak, didalamnya terdapat variabel protected nama dan id. Penggunaan protected sendiri bertujuan agar variabel tersebut dapat diakses oleh kelas turunannya.

- Konstruktor dengan 2 parameter yaitu nama dan id kemudian menginisialisasi nama dan id.
- Method ShowInfo() merupakan method abstrak tanpa implementasi, tetapi akan diimplementasikan (harus . wajib) pada kelas turunannya untuk menampilkan data nama dan id Person.

6. Main.java

Merupakan file utama yang digunakan untuk menjalankan (Run) project. Didalamnya terdapat:

- Membuat objek Mahasiswa dengan nama “Rizqi Dwi Saputra”, ID “001”, NIM “2311103133”, Prodi “Sistem Informasi”.
- Membuat objek Dosen dengan nama “Sena Wijayanto”, ID “456”, NIP “D001”
- Membuat objek MataKuliah pertama dengan nilai: Kode MK “SI101”, Nama MK “PBO”, SKS 3
- Membuat objek MataKuliah kedua dengan nilai: Kode MK “SI102”, Nama MK “Basis Data”, SKS 3
- Mengatur jadwal MataKuliah pertama dengan nilai: Hari “Senin” Jam “08:00”
- Mengatur jadwal MataKuliah kedua dengan nilai: Hari “Selasa”
- Menambah MataKuliah pertama kedalam KRS Mahasiswa
- Menambah MataKuliah kedua kedalam KRS Mahasiswa

- Menampilkan informasi data Mahasiswa, Dosen, MataKuliah pertama, MataKuliah kedua

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.

Person.java

```
package unguided1;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

abstract class Person {

    protected String nama;
    protected String id;

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

    //    Abstarct method (Implmentasi khusus di SubClass)
    public abstract void ShowInfo();

}
```

MataKuliah.java

```
package unguided1;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

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 String getKode() {
        return kode;
    }

    public String getNamaMataKuliah() {
        return namaMataKuliah;
    }

    public int getSks() {
        return sks;
    }

    public String getJadwalHari() {
```

```

        return jadwalHari;
    }

    public String getJadwalJam() {
        return jadwalJam;
    }

    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);
    }
}

```

KRS.java

```

package unguided1;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

interface KRS {

    void tambahMataKuliah(MataKuliah mk);
    void hapusMataKuliah (MataKuliah mk);

}

```

Dosen.java

```

package unguided1;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

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);
}

}
```

Mahasiswa.java

```
package unguided1;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

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);
    }

    public void showKRS () {
        System.out.println("");
        System.out.println("Daftar MatKul " + nama + ":");
        for(int i = 0; i < daftarKRS.size(); i++){
            MataKuliah mk = daftarKRS.get(i);
            System.out.println((i+1) + ". Kode MatKul: " + mk.getKode()
                + ", Nama MatKul: " + mk.getNamaMataKuliah()
                + ", SKS: " + mk.getSks()
                + ", Jadwal: " + (mk.getJadwalHari() != null ?
mk.getJadwalHari(): "Tidak ada hari")
                + (mk.getJadwalJam() != null ? " " + mk.getJadwalJam() : ""));
        }
    }

}
```

Main.java

```
package unguided1;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

public class Main {

    public static void main(String[] args) {

        //      Membuat objek Mahasiswa dan Dosen
        Mahasiswa mhs = new Mahasiswa("Rizqi Dwi Saputra", "001",
        "2311103133", "Sistem Informasi");
        Dosen dosen = new Dosen("Sena Wijayanto", "456", "D001");

        //      Membuat objek Mata Kuliah
        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
        mhs.showKRS(); // Informasi daftar MatKul Putra ganteng

    }

}
```

Hasil Run Project

```
run:
Nama Mahasiswa: Rizqi Dwi Saputra, NIM: 2311103133, Prodi: Sistem Informasi
Nama Dosen: Sena Wijayanto, NIP: D001
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3

Daftar MatKul Rizqi Dwi Saputra:
1. Kode MatKul: SI101, Nama MatKul: PBO, SKS: 3, Jadwal: Senin 08:00
2. Kode MatKul: SI102, Nama MatKul: Basis Data, SKS: 3, Jadwal: Selasa
```

Penjelasan:

Mengubah kode pada kelas MataKuliah dengan menambahkan Getter. Getter berfungsi untuk memudahkan mendapatkan dan membaca nilai dari variabel.

- Getter untuk membaca nilai variabel kode (public string getKode(){ return kode; })
- Getter untuk membaca nilai variabel sks (public string getSks(){ return sks; })

- Getter untuk membaca nilai variabel namaMataKuliah (public string getNameMataKuliah(){ return namaMataKuliah; }
- Getter untuk membaca nilai variabel jadwalHari (public string getJadwalHari(){ return jadwalHari; }
- Getter untuk membaca nilai variabel jadwalJam (public string getJadwalJam(){ return jadwalJam; }

Kemudian, menambahkan method showKRS pada kelas mahasiswa dengan tujuan menampilkan informasi berupa Detail Matakuliah apa saja yang diambil oleh Mahasiswa (Kode MatKul, Nama MatKul, SKS, Jadwal). Untuk menampilkan informasi detail tersebut, gunakan:

- Perulangan for dengan variabel i bernilai 0. Logika nya adalah perulangan akan berhenti ketika nilai i tidak lebih kecil / sama dengan ukuran data yang ada daftarSKS.
- Mencetak nilai i+1
- Tampilkan informasi kode matkul dengan memanggil method getter (mk.getKode)
- Tampilkan informasi nama matkul dengan memanggil method getter (mk.getNameMataKuliah)
- Tampilkan informasi sks dengan memanggil method getter (mk.getSks)
- Jika hari tidak sama dengan null tampilkan informasi jadwal dengan memanggil method getter (mk.getJadwalHari) tetapi, jika jadwalHari bernilai null maka tampilkan pesan “Tidak ada hari”
- Jika jam tidak sama dengan null tampilkan informasi jam dengan memanggil method getter (mk.getJadwalJam) tetapi, jika jam sama dengan null maka tidak akan tampil (yang terpanggil adalah method aturJadwal kedua yg tanpa jam)

Terakhir, panggil method showKRS() pada file utama (Main.java)

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.

Person.java

```
package unguided2;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

abstract class Person {

    protected String nama;
    protected String id;

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

    //    Abstarct method (Implmentasi ...)
    public abstract void ShowInfo();
}
```

```
}
```

MataKuliah.java

```
package unguided2;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

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 String getKode() {
        return kode;
    }

    public String getNamaMataKuliah() {
        return namaMataKuliah;
    }

    public int getSks() {
        return sks;
    }

    public String getJadwalHari() {
        return jadwalHari;
    }

    public String getJadwalJam() {
        return jadwalJam;
    }

    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);
    }
}
```

KRS.java

```

package unguided2;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

interface KRS {

    void tambahMataKuliah(MataKuliah mk);
    void hapusMataKuliah (MataKuliah mk);

}

```

Dosen.java

```

package unguided2;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

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);
    }

}

```

Mahasiswa.java

```

package unguided2;
//@author Rizqi Dwi Saputra 2311103133 / S1SI07C

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);
    }

    public void showKRS() {
        System.out.println("");
        System.out.println("Daftar MatKul " + nama + ":");
        for(int i = 0; i < daftarKRS.size(); i++){
            MataKuliah mk = daftarKRS.get(i);
            System.out.println((i+1) + ". Kode MatKul: " + mk.getKode()
                + ", Nama MatKul: " + mk.getNamaMataKuliah()
                + ", SKS: " + mk.getSks()
                + ", Jadwal: " + (mk.getJadwalHari() != null ?
mk.getJadwalHari(): "Tidak ada hari")
                + (mk.getJadwalJam() != null ? " " + mk.getJadwalJam() : ""));
        }
    }

    public void hitungTotalSKS() {
        int totalSKS = 0;
        for(MataKuliah mk : daftarKRS){
            totalSKS += mk.getSks();
        }
        System.out.println("Total SKS yang diambil oleh " + nama + ": " +
totalSKS + " SKS");
    }
}

```

Main.java

```

package unguided2;
//@author Rizqi Dwi Saputra ganteng bgt 2311103133 / S1SI07C

public class Main {

    public static void main(String[] args) {

        //      Membuat objek Mahasiswa dan Dosen
        Mahasiswa mhs = new Mahasiswa("Rizqi Dwi Saputra", "001",
"2311103133", "Sistem Informasi");
        Dosen dosen = new Dosen("Sena Wijayanto", "456", "D001");
    }
}

```



```
//      Membuat objek Mata Kuliah
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

mhs.showKRS(); // Informasi daftar MatKul
mhs.hitungTotalSKS();

}

}
```

Hasil Run Project

```

Nama Mahasiswa: Rizqi Dwi Saputra, NIM: 2311103133, Prodi: Sistem Informasi
Nama Dosen: Sena Wijayanto, NIP: D001
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3

Daftar MatKul Rizqi Dwi Saputra:
1. Kode MatKul: SI101, Nama MatKul: PBO, SKS: 3, Jadwal: Senin 08:00
2. Kode MatKul: SI102, Nama MatKul: Basis Data, SKS: 3, Jadwal: Selasa
Total SKS yang diambil oleh Rizqi Dwi Saputra: 6 SKS

```

Penjelasan:

Membuat method `hitungTotalSKS()` pada kelas Mahasiswa, kemudian:

- Menginisialisasi variabel `totalSKS` dengan nilai 0
- Menggunakan perulangan `for` dimana perulangan akan berulang sebanyak data `daftarKRS`
- Didalam perulangan, tambahkan `totalSKS` dengan jumlah SKS yang ada dengan memanggil method `getter (mk.getKRS())`:
- Tampilkan informasi jumlah SKS yang diambil oleh nama Mahasiswa

Terakhir, pada file utama (Main.java) panggil method `hitungTotalSKS` untuk menampilkan informasi jumlah SKS yang diambil oleh mahasiswa.

Saya telah merekayasa program bertujuan untuk menguji apakah logika method `hitungTotalSKS()` sudah benar atau belum dengan cara menambahkan MataKuliah lagi untuk mahasiswa, berikut hasilnya:

run:

Nama Mahasiswa: Rizqi Dwi Saputra, NIM: 2311103133, Prodi: Sistem Informasi

Nama Dosen: Sena Wijayanto, NIP: D001

Kode MK: SI101, Nama: PBO, SKS: 3

Kode MK: SI102, Nama: Basis Data, SKS: 3

Kode MK: SI103, Nama: Struktur Data, SKS: 2

Daftar MatKul Rizqi Dwi Saputra:

1. Kode MatKul: SI101, Nama MatKul: PBO, SKS: 3, Jadwal: Senin 08:00

2. Kode MatKul: SI102, Nama MatKul: Basis Data, SKS: 3, Jadwal: Selasa

3. Kode MatKul: SI103, Nama MatKul: Struktur Data, SKS: 2, Jadwal: Tidak ada hari

Total SKS yang diambil oleh Rizqi Dwi Saputra: 8 SKS