

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 3
LATIAN KONSEP PBO



Oleh :

Geordy Sipho Samuel Damanik

2311103112

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

```
package LatihanP7GeordySiphoSamuelDamanik;

/**
 *
 * @author Geordy Sipho Samuel Damanik
 * SISI-07-C
 * 2311103112
 */
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();
}
```

Kode Mata Kuliah

```
package LatihanP7GeordySiphoSamuelDamanik;

/**
 *
 * @author Geordy Sipho Samuel Damanik
 * S1SI-07-C
 * 2311103112
 */
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);
    }

}
```

Kode KRS

```
package LatihanP7GeordySiphoSamuelDamanik;

/**
 *
 * @author Geordy Sipho Samuel Damanik
 * SISI-07-C
 * 2311103112
 */

interface KRS{

    void tambahMatakuliah(MataKuliah mk);
    void hapusMatakuliah(MataKuliah mk);
}
```

Kode Dosen

```
package LatihanP7GeordySiphoSamuelDamanik;

/**
 *
 * @author Geordy Sipho Samuel Damanik
 * SISI-07-C
 * 2311103112
 */

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);
    }
}
```

Kode Mahasiswa

```
package LatihanP7GeordySiphoSamuelDamanik;
/**
 *
 * @author Geordy Sipho Samuel Damanik
 * SISI-07-C
 * 2311103112
 */

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);
    }
}
```

Kode Main dengan nama Modul4

```
package LatihanP7GeordySiphoSamuelDamanik;

/**
 *
 * @author Geordy Sipho Samuel Damanik
 * SISI-07-C
 * 2311103112
 */
public class Modul4 {

    public static void main(String[] args) {
        // Membuat Objek Mahasiswa dan Dosen
        Mahasiswa mhs = new Mahasiswa("Geordy Sipho Samuel Damanik", "123",
"2311103112", "Sistem Informasi");
        Dosen dosen = new Dosen("Sena Wijayanto", "456", "D001");

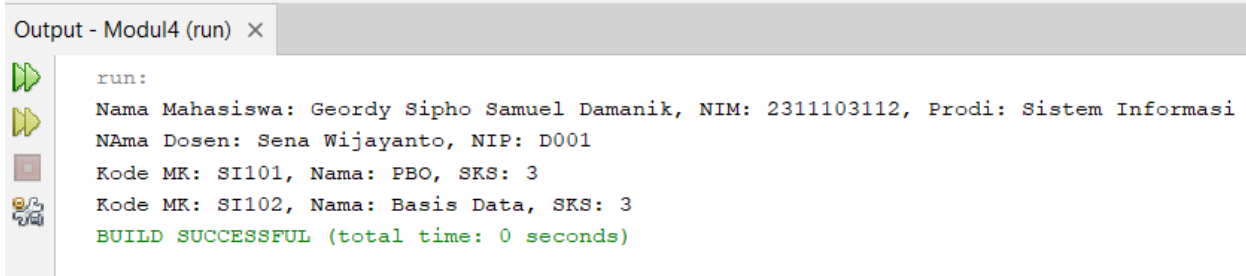
        // Membuat Objek Matakuliah
        MataKuliah mk1 = new MataKuliah("SI101", "PBO", 3);
        MataKuliah mk2 = new MataKuliah("SI102", "Basis Data", 3);

        // 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
    }
}
```

Hasil Running :

A screenshot of a terminal window titled "Output - Modul4 (run) X". The window contains the following text: "run:", "Nama Mahasiswa: Geordy Sipho Samuel Damanik, NIM: 2311103112, Prodi: Sistem Informasi", "Nama Dosen: Sena Wijayanto, NIP: D001", "Kode MK: SI101, Nama: PBO, SKS: 3", "Kode MK: SI102, Nama: Basis Data, SKS: 3", and "BUILD SUCCESSFUL (total time: 0 seconds)". On the left side of the terminal window, there is a vertical toolbar with icons for running (a green play button), stopping (a red square), and other development actions.

```
run:
Nama Mahasiswa: Geordy Sipho Samuel Damanik, NIM: 2311103112, Prodi: Sistem Informasi
Nama Dosen: Sena Wijayanto, NIP: D001
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3
BUILD SUCCESSFUL (total time: 0 seconds)
```

= Program ini adalah program sederhana untuk mengelola data akademik, termasuk informasi mahasiswa, dosen, dan mata kuliah, mahasiswa dan dosen disini adalah entitas yang dapat menyimpan dan menampilkan informasi spesifik seperti nama, ID, NIM, dan NIP. Program ini juga menyediakan fitur untuk mengatur jadwal mata kuliah dan menambah atau menghapus mata kuliah dalam Kartu Rencana Studi (KRS) mahasiswa. Outputnya adalah tampilan informasi yang jelas dan terperinci tentang setiap entitas, yang ditampilkan di konsol, sehingga memudahkan pemantauan dan pengelolaan data akademik.

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.
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.

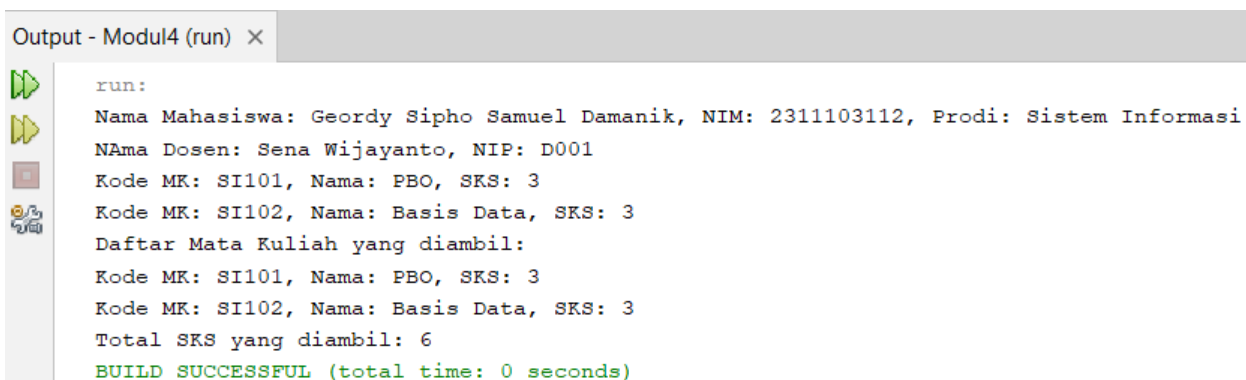
Jawab:

1. Langkah awal kita perlu lakukan modifikasi MataKuliah dengan menambahkan getter untuk atribut sks di kelas MataKuliah.

```
//Metode getter untuk atribut sks
public int getSks() {
    return sks;
}
```

2. Lalu kita modifikasi kelas Mahasiswa dengan menambahkan dua metode baru yaitu ShowKRS() untuk menampilkan daftar mata kuliah dan hitungTotalSKS() untuk menghitung total sks yang diambil.

```
public int hitungTotalSKS() {
    int totalSKS = 0;
    for (MataKuliah mk : daftarKRS) {
        totalSKS += mk.getSks(); //getter untuk mendapatkan nilai SKS
    }
    return totalSKS;
}
```



```
Output - Modul4 (run) x
run:
Nama Mahasiswa: Geordy Sipho Samuel Damanik, NIM: 2311103112, 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 Mata Kuliah yang diambil:
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3
Total SKS yang diambil: 6
BUILD SUCCESSFUL (total time: 0 seconds)
```

= Berikut adalah hasil running yang mana ketika kita modifikasi di kelas MataKuliah dengan menambahkan getter untuk atribut sks dan memodifikasi kelas Mahasiswa dengan menambahkan dua metode yaitu ShowKRS() dan hitungTotalSKS() maka akan tampil Daftar mata kuliah yang kita ambil dan total sks yang kita ambil seperti ss diatas.