

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 3



Oleh :

Erick Evan Nicholas

2311103115

S1SI-07C

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

LATIHAN KONSEP PBO

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.

Jawaban :

- Person.java :

```
package latihanp7erickevannicholas;
```

```
/**
```

```
 * 2311103115
```

```
 * @author Erick Evan Nicholas
```

```
 * S1SI-07C
```

```
 */
```

```
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 merupakan kelas abstrak yang menjadi superclass atau kelas induk dari Mahasiswa dan Dosen. Kelas ini menyimpan informasi dasar seperti nama dan ID, dan memiliki metode abstrak showInfo(), yang akan diimplementasikan berbeda di setiap kelas turunan.

- Mahasiswa.java :

```

package latihanp7erickevannicholas;

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

/**
 * 2311103115
 * @author Erick Evan Nicholas
 * S1SI-07C
 */
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

```

```

@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 turunan dari Person dan mengimplementasikan interface KRS. Kelas ini menyimpan informasi khusus untuk mahasiswa, seperti NIM, program studi, dan daftar KRS. Selain itu, Mahasiswa juga mengimplementasikan metode tambahMatakuliah dan hapusMatakuliah dari interface KRS, yang memungkinkan mahasiswa untuk menambah atau menghapus matakuliah dalam KRS mereka. Metode showInfo() di sini menampilkan informasi mahasiswa seperti nama, NIM, dan prodi.

- Dosen.java :

```

package latihanp7erickevannicholas;

/**
 * 2311103115
 * @author Erick Evan Nicholas
 * S1SI-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;
}

// 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 turunan dari Person yang menyimpan data dosen, seperti NIP dan daftar matakuliah yang diajarkan. Kelas ini memiliki metode tambahMatakuliah, yang memungkinkan dosen menambahkan matakuliah yang diajarkan ke dalam daftar. showInfo() pada kelas ini diimplementasikan untuk menampilkan informasi dosen, seperti nama dan NIP.

- MataKuliah.java :

```

package latihanp7erickevannicholas;

/**
 * 2311103115
 * @author Erick Evan Nicholas
 * S1SI-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) {

```

```

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 berisi data mengenai matakuliah, seperti kode matakuliah, nama, dan jumlah SKS. Kelas ini juga memiliki metode aturJadwal yang dioverload, artinya ada dua versi metode ini, satu untuk mengatur hari dan jam sekaligus, dan satu lagi hanya untuk mengatur hari saja. Lalu showInfo() digunakan untuk menampilkan detail dari matakuliah.

- KRS.java :

```

package latihanp7erickevannicholas;

/**
 * 2311103115
 * @author Erick Evan Nicholas
 * S1SI-07C
 */
interface KRS {

    void tambahMatakuliah(MataKuliah mk);

    void hapusMatakuliah(MataKuliah mk);

}

```

Penjelasan : Interface KRS untuk penambahan dan penghapusan matakuliah dari KRS mahasiswa. Interface ini bertujuan untuk memastikan bahwa setiap kelas yang mengimplementasikan KRS memiliki metode tambahMatakuliah dan hapusMatakuliah.

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 :

Kode Program yang ditambahkan,

- Mahasiswa.java :

```
package latihanp7erickevannicholas;

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

/**
 * 2311103115
 * @author Erick Evan Nicholas
 * S1SI-07C
 */
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);  
}
```

// Metode untuk menampilkan daftar matakuliah yang diambil oleh mahasiswa

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

// Metode untuk menghitung total SKS yang diambil oleh mahasiswa

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


Penjelasan : Saat ingin mengetahui total SKS yang diambil, dapat memanggil metode `hitungTotalSKS()`. Program akan menghitung total SKS dengan menggunakan getter `getSKS()` untuk mengakses nilai SKS dari setiap matakuliah, lalu menjumlahkannya dengan looping. Total SKS ini akan membantu mahasiswa memastikan bahwa mereka tidak melebihi atau kurang dari batas SKS yang disarankan.

- MataKuliah.java :

```
package latihanp7erickevannicholas;

/**
 * 2311103115
 * @author Erick Evan Nicholas
 * S1SI-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;
    }
}
```

```

        this.jadwalHari = hari;
    }

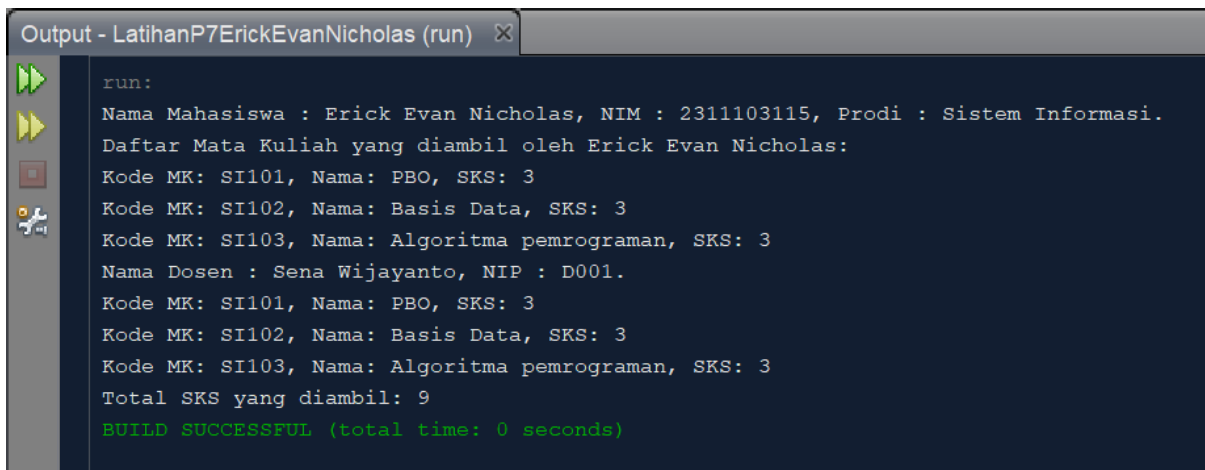
    public void showInfo() {
        System.out.println("Kode MK: " + kode + ", Nama: " + namaMatakuliah + ", SKS: " +
sks);
    }

    public int getSKS() {
        return sks;
    }
}

```

Penjelasan : Saat mahasiswa ingin melihat daftar matakuliah yang telah mereka ambil, mereka bisa memanggil metode showKRS(). Program akan menggunakan looping untuk menampilkan setiap matakuliah yang telah ditambahkan ke daftar KRS mahasiswa. Informasi yang ditampilkan mencakup kode, nama, dan SKS setiap matakuliah.

Runing Program :



```

Output - LatihanP7ErickEvanNicholas (run) ✕
run:
Nama Mahasiswa : Erick Evan Nicholas, NIM : 2311103115, Prodi : Sistem Informasi.
Daftar Mata Kuliah yang diambil oleh Erick Evan Nicholas:
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3
Kode MK: SI103, Nama: Algoritma pemrograman, SKS: 3
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: Algoritma pemrograman, SKS: 3
Total SKS yang diambil: 9
BUILD SUCCESSFUL (total time: 0 seconds)

```

Penjelasan : Saat program dirun, outputnya akan menampilkan informasi mahasiswa, daftar matakuliah yang sudah diambil, dan total SKS yang diambil mahasiswa. Ini menunjukkan bahwa metode baru bekerja sesuai yang diharapkan, memberikan kemudahan untuk melihat matakuliah yang diambil dan menghitung total SKS.