

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 3
LATIHAN KONSEP PBO



Oleh :
Anggit Refiyan
2311103142
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.

Kode

Person.java

```
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:

Kode program diatas merupakan kelas person yang dibuat untuk mendefinisikan kelas abstrak yang akan menjadi kelas dasar untuk kelas lain. Pada kelas ini terdapat 2 atribut yaitu, nama dan juga id. Pada kelas ini juga memiliki konstruktor untuk menginialisasikan atribut nama dan id. Pada kelas ini juga terdapat abstrak void showinfo untuk mengimplementasikan khusus pada subclass.

MataKuliah.java

```
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:

Pada kelas matakuliah berfungsi untuk mempresentasikan informasi mengenai mata kuliah, yang terdapat beberapa atribut seperti kode, namaMatakuliah, sks, jadwalhari, dan jadwaljam, semua atribut itu bersifat private yang artinya hanya dapat diakses di dalam kelas matakuliah. Selain atribut, kelas ini terdapat konstruktor yang digunakan untuk menginisialisasi atribut kode, namaMatakuliah, dan sks. Kelas ini menerapkan konsep overloading dimana Kelas ini memiliki dua metode aturJadwal dengan nama yang sama tetapi parameter yang berbeda. Yang terakhir showinfo digunakan untuk menampilkan informasi mata kuliah.

KRS.java

```
interface KRS {
```

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

Penjelasan:

Pada interface KRS hanya mendefinisikan 2 metode untuk mengelola daftar matakuliah, yang pertama tambahmatakuliah untuk menambahkan data nama matakuliah dan juga yang kedua hapusmatakuliah untuk menghapus data matakuliah.

Dosen.java

```
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 merupakan subclass dari kelas abstrak person dengan atribut khusus seperti nip dan daftar mata kuliah yang diajarkan (daftarMatakuliah). Kelas ini mengimplementasikan metode abstrak showInfo untuk menampilkan informasi dosen, dan menambahkan metode tambahMatakuliah yang memungkinkan penambahan mata kuliah ke dalam daftar mata kuliah dosen.

Mahasiswa.java

```
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 merupakan subclass dari kelas person dan mengimplementasikan interface krs. Kelas ini memiliki atribut tambahan untuk menyimpan informasi NIM dan program studi mahasiswa, dan daftar KRS dalam bentuk list daftarKRS. Kelas Mahasiswa menyediakan metode tambahMatakuliah dan hapusMatakuliah untuk mengelola daftar mata kuliah dalam KRS, dan memiliki implementasi khusus untuk metode showInfo yang menampilkan informasi lengkap mahasiswa.

Main.java

```

public class Main {

    /**
     * @param args the command line arguments
     */
    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);

        // 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:

Kelas main merupakan kelas utama yang digunakan untuk melakukan perintah membuat objek baru dari objek mahasiswa dan juga dosen, kemudian berisi kode program yang digunakan untuk memanggil perintah lain seperti tambahmatakuliah, membuat objek matakuliah, mengaturjadwal, dan juga menampilkan informasi dari seluruh data yang sudah ditambahkan.

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.

Kode Program

1. Person

```
package latihanp7anggitrefiyan;

/**
 *
 * @author Anggit Refiyan
 * 2311103142
 * 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();
}
```

2. Mahasiswa

```
package latihanp7anggitrefiyan;

/**
 *
 * @author Anggit Refiyan
 * 2311103142
 * 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();
    }
}

//menghitung total sks
public int hitungSKS(){
    int totalSKS = 0;
    for (MataKuliah mk : daftarKRS){
        totalSKS += mk.getSKS();
    }
    return totalSKS;
}
}

```

3. Dosen

```
package latihanp7anggitrefiyan;
```

```
/**
```



```

*
* @author Anggit Refiyan
* 2311103142
* 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 showinfo (implemen khusus dosen)
    @Override
    public void showinfo(){
        System.out.println("Nama Dosen: " + nama + ", NIP: " +nip);
    }

    public void tambahMataKuliah(MataKuliah mk){
        daftarMatakuliah.add(mk);
    }
}

```

4. KRS

```
package latihanp7anggitrefiyan;
```

```

/**
 *
 * @author Anggit Refiyan
 * 2311103142
 * 07C
 */
interface KRS {

    void tambahMataKuliah(MataKuliah mk);

    void hapusMatakuliah(MataKuliah mk);
}

```

5. MataKuliah

```
package latihanp7anggitrefiyan;
```

```
/**
```

```
*
```

```
* @author Anggit Refiyan
```

```
* 2311103142
```

```
* 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;
```

```
    }
```

```
    //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);
```

```
    }
```

```
}
```

6. Main (LatihanP7AnggitRefiyan)

```
package latihanp7anggitrefiyan;
```

```
/**
```

```
*
```

```

* @author Anggit Refiyan
* 2311103142
* 07C
*/
public class LatihanP7AnggitRefiyan {

    public static void main(String[] args) {
        Mahasiswa mhs = new Mahasiswa("Anggit", "123", "2311103142", "Sistem Informasi");
        Dosen dosen = new Dosen("Refiyan", "456", "A01");

        MataKuliah mk1 = new MataKuliah("SI07", "PBO", 3);
        MataKuliah mk2 = new MataKuliah("SI07", "Perancangan Interaksi", 3);

        mk1.aturJadwal("senin", "07.00");
        mk2.aturJadwal("selasa", "07.30");

        mhs.tambahMataKuliah(mk1);
        mhs.tambahMataKuliah(mk2);

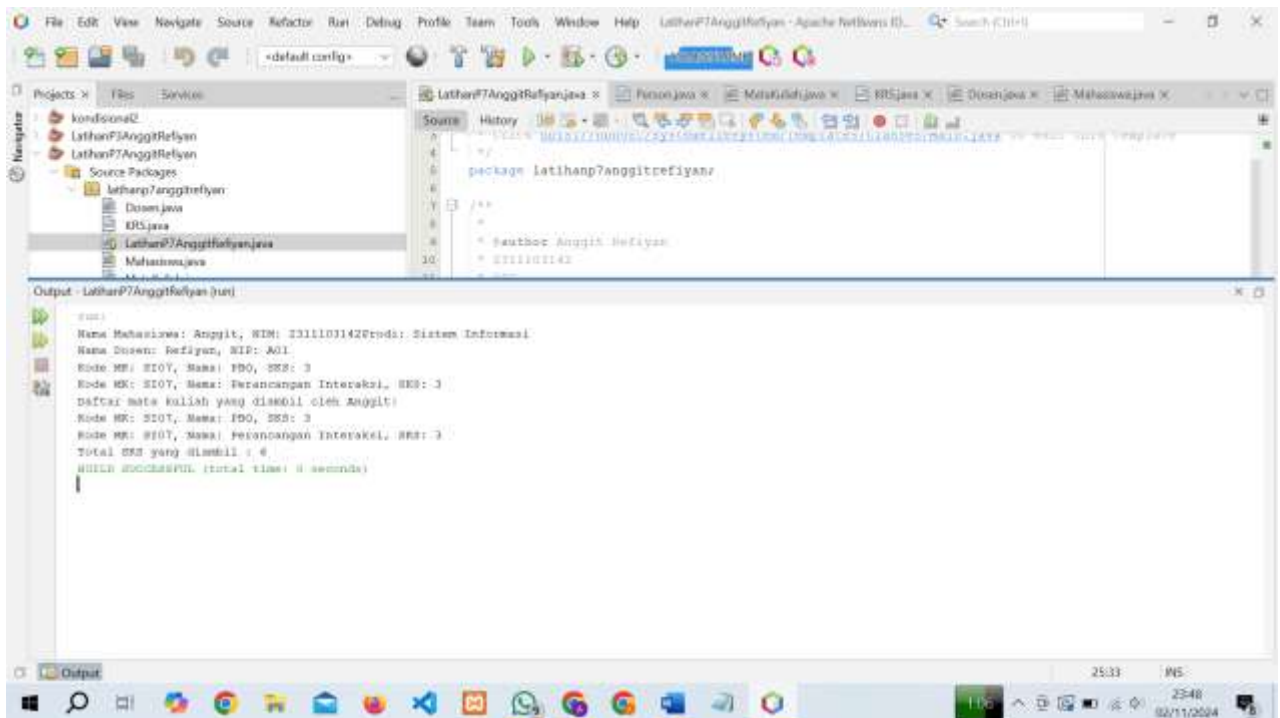
        // Menampilkan Informasi
        mhs.showinfo(); // Menampilkan informasi Mahasiswa (Polimorfisme)
        dosen.showinfo(); // Menampilkan informasi Dosen (Polimorfisme)
        mk1.showInfo(); // Informasi Matakuliah 1
        mk2.showInfo(); // Informasi Matakuliah 2

        //menampilkan KRS
        mhs.showKRS();

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

```

Hasil Running dan Penjelasan



Penjelasan:

Gambar diatas merupakan hasil modifikasi dari kode program guided sebelumnya, pada soal unguided disuruh untuk menampilkan Daftar Mata Kuliah yang Diambil oleh Mahasiswa, kemudian Tambahkan metode **showKRS()** dalam kelas **Mahasiswa** yang menampilkan seluruh mata kuliah yang ada dalam daftar KRS mahasiswa tersebut. Perintah kedua menghitung Total SKS yang Diambil oleh Mahasiswa dengan menambahkan metode **hitungTotalSKS()** dalam kelas **Mahasiswa** untuk menghitung total SKS dari seluruh mata kuliah yang telah diambil.

Kode program yang saya tambahkan untuk menyelesaikan soal diatas saya menambahkan public void showKRS yang digunakan untuk menampilkan daftar mata kuliah yang diambil oleh mahasiswa tersebut, di dalam public voidKRS saya menambahkan perulangan for untuk mengambil setiap MataKuliah dari daftarKRS. Selanjutnya saya menambahkan getter di kelas MataKuliah agar dapat mengakses atribut kode,namaMatakuliah dan juga SKS. Selanjutnya saya menambahkan kode program hitungTotalSKS dalam kelas mahasiswa untuk menghitung total sks, dengan mendeklarasikan variabel totalSKS dengan nilai 0, kemudian menambahkan for untuk mengakses setiap MataKuliah di dalam daftarKRS. Setelah kode tersebut ditambahkan, Langkah terakhir tambahkan kode program untuk menampilkan krs dan juga menghitungKRS pada kelas main(latihan7anggitreflyan) seperti berikut ini:

mhs.showKRS(); -> Menampilkan KRS

System.out.println("Total SKS yang diambil : " + mhs.hitungSKS()); -> menampilkan output total sks yang diambil oleh si mahasiswa kemudian menambahkan kode untuk menghitung sks, sehingga output akhir menampilkan total sks yang diambil oleh mahasiswa itu .