

**LAPORAN PRAKTIKUM**  
**PEMROGRAMAN BERORIENTASI OBJEK**  
**MODUL 3**  
**Latihan Konsep PBO**



Oleh :  
Raditya Putra Anugra Pratama  
2311103143  
S1SI07-C

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

## I. GUIDED

### Source Code

Person.java

```
package com.mycompany.latihanp7radityaputra;

/**
 *
 * @author Raditya Putra Anugra Pratama
 * 2311103143
 * S1SI07-C
 */
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 adalah kelas abstrak yang mendefinisikan entitas dengan atribut nama dan id, serta sebuah konstruktor untuk menginisialisasi kedua atribut tersebut. Kelas ini juga memiliki metode abstrak showInfo(), yang harus diimplementasikan oleh kelas-kelas turunan yang mewarisi kelas Person.

## Matakuliah.java

```
package com.mycompany.latihanp7radityaputra;

/**
 *
 * @author Raditya Putra Anugra Pratama
 * 2311103143
 * S1SI07-C
 */
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 :

Kelas MataKuliah memiliki atribut privat, yaitu ‘kode’, ‘namaMatakuliah’, ‘sks’, ‘jadwalHari’, dan ‘jadwalJam’, yang menyimpan informasi penting tentang matakuliah tersebut. Konstruktor ‘MataKuliah’ menerima tiga parameter ‘kode’, ‘namaMatakuliah’, dan ‘sks’ untuk menginisialisasi atribut-atribut yang relevan saat objek baru dibuat. kKelas ini juga menerapkan konsep overloading dengan menyediakan dua versi metode ‘aturJadwal’. Versi pertama memungkinkan pengguna untuk mengatur jadwal mata kuliah dengan menentukan hari dan jam, sedangkan versi kedua hanya memerlukan parameter hari, memberikan fleksibilitas dalam mengatur jadwal. Selain itu, metode ‘showInfo’ digunakan untuk menampilkan iformasi dasar mengenai matakuliah, seperti kode, nama, dan jumlah SKS, dalam format yang mudah dibaca.

KRS.java

```

package com.mycompany.latihanp7radityaputra;

/**
 *
 * @author Raditya Putra Anugra Pratama
 * 2311103143
 * S1SI07-C
 */
interface KRS {

    void tambahMatakuliah(MataKuliah mk);

    void hapusMatakuliah(MataKuliah mk);

}

```

Penjelasan:

Interface KRS mendefinisikan kontrak untuk pengelolaan mata kuliah, dengan dua metode: `tambahMataKuliah(MataKuliah mk)` untuk menambahkan mata kuliah dan `hapus MataKuliah(MataKuliah mk)` untuk menghapus mata kuliah. Kelas yang mengimplementasikan interface ini diharuskan untuk menyediakan logika spesifik untuk kedua operasi tersebut.

Dosen.java

```
package com.mycompany.latihanp7radityaputra;

/**
 *
 * @author Raditya Putra Anugra Pratama
 * 2311103143
 * S1SI07-C
 */

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);
    }
}

```

Penjelasan:

Merupakan subclass dari kelas abstrak 'Person', yang dirancang untuk merepresentasikan dosen dalam sistem akademik. Kelas ini memiliki atribut privat 'nip' untuk menyimpan Nomor Induk Pegawai dosen, serta 'daftarMatakuliah', yang merupakan sebuah list untuk menyimpan objek-objek 'MataKuliah' yang diampu oleh dosen tersebut. Konstruktor 'Dosen' memanggil konstruktor superclass 'Person' untuk menginisialisasi atribut nama dan id, serta mengatur 'nip'. Metode 'showInfo' diimplementasikan kembali (overriding) untuk menampilkan informasi spesifik mengenai dosen, termasuk nama dan NIP, dengan format yang jelas. Selain itu, kelas ini juga memiliki metode 'tambahMatakuliah', yang memungkinkan dosen untuk menambahkan mata kuliah ke dalam daftar mereka.

Mahasiswa.java

```

package com.mycompany.latihanp7radityaputra;

/**
 *
 * @author Raditya Putra Anugra Pratama
 * 2311103143
 * S1SI07-C
 */

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);
}
}

```

Penjelasan :

Kelas Mahasiswa adalah subclass dari Person.java yang juga mengimplementasikan interface KRS, sehingga berfungsi untuk merepresentasikan seorang mahasiswa. Kelas ini memiliki atribut tambahan nim, prodi, dan daftar matakuliah yang diambil (daftarKRS). Konstruktor kelas ini menginisialisasi nama, ID, NIM, dan program studi.

Metode showInfo() di override untuk menampilkan informasi mahasiswa , sementara metode dari interface KRS, yaitu tambahMataKuliah(MataKuliah mk) dan hapusMataKuliah (MataKuliah mk) digunakan untuk mengubah dan menghapus mata kuliah dari daftar KRS.

Main.java

```
package com.mycompany.latihanp7radityaputra;

/**
 *
 * @author Raditya Putra Anugra Pratama
 * 2311103143
 * S1SI07-C
 */

public class Main {

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

    public static void main(String[] args) {

        // Membuat Objek Mahasiswa dan Dosen

        Mahasiswa mhs = new Mahasiswa("Raditya", "123", "2311103143", "Sistem
Informasi");

        Dosen dosen = new Dosen("Raditya Putra Anugra Pratama", "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 berfungsi sebagai titik masuk untuk menjalankan program. Di dalam metode main, objek Mahasiswa dan Dosen diciptakan dengan memberikan informasi dalam seperti nama, ID, NIM, dan NIP. Dua objek MataKuliah juga dibuat, dan jadwal untuk masing-masing mata kuliah diatur menggunakan metode aturJadwal. Mahasiswa kemudian menambahkan mata kuliah tersebut ke dalam daftar KRS mereka dengan memanggil metode tambahMataKuliah. Terakhir, program menampilkan informasi tentang mahasiswa, dosen, dan mata kuliah yang telah dibuat dengan memanggil metode showInfo() untuk masing-masing objek.

#### Output :

```

--- compiler:3.13.0:compile (default-compile) @ LatihanP7RadityaPutra ---
Nothing to compile - all classes are up to date.

--- exec:3.1.0:exec (default-cli) @ LatihanP7RadityaPutra ---
Nama Mahasiswa: Raditya, NIM: 2311103143, Prodi: Sistem Informasi
Nama Dosen : Raditya Putra Anugra Pratama, NIP : D001
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3

-----
BUILD SUCCESS
-----

Total time:  0.845 s
Finished at: 2024-11-03T18:31:55+07:00
-----

```

## II. UNGUIDED

Kembangkan Sistem Informasi Akademik dengan menambahkan fitur berikut:

1. Menampilkan Daftar Mata Kuliah yang Diambil Mahasiswa.  
Tambahkan metode **showKRS()** dalam kelas **Mahasiswa** yang menampilkan 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:

1. Menampilkan Daftar Mata Kuliah yang Diambil Mahasiswa.  
Tambahkan metode **showKRS()** dalam kelas **Mahasiswa** yang menampilkan mata kuliah yang ada dalam daftar KRS mahasiswa tersebut

Mahasiswa.java

```
package Unguided;

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

/**
 *
 * @author Raditya Putra Anugra Pratama
 * 2311103143
 * SISI07-C
 */
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("\n===== Daftar KRS Mahasiswa " + nama + " =====");
    if (daftarKRS.isEmpty()) {
        System.out.println("Tidak ada mata kuliah yang diambil.");
    } else {
        for (MataKuliah mk : daftarKRS) {
            mk.showInfo();
        }
        System.out.println("Total SKS yang diambil: " + hitungTotalSKS());
    }
}

// Metode untuk menghitung total SKS

```

```

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

```

Penjelasan:

Kelas mahasiswa adalah turunan dari kelas abstrak Person dan juga mengimplementasikan interface KRS. Kelas ini menyimpan informasi penting seperti NIM, prodi, dan daftarKRS, yang berfungsi untuk menyimpan mata kuliah yang diambil. Dalam konstruktor Mahasiswa, nama, ID, Nim, dan program studi diatur dengan memanggil konstruktor dari kelas Person. Metode showInfo() digunakan untuk menampilkan informasi dasar tentang mahasiswa, termasuk nama, NIM, dan program studi.

MataKuliah.java

```

package Unguided;

/**
 *
 * @author Raditya Putra Anugra Pratama
 * 2311103143
 * S1SI07-C
 */
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);
    }

    // Metode untuk mendapatkan jumlah SKS

    public int getSks() {

        return sks;
    }
}

```

Penjelasan:

Kelas MataKuliah mempresentasikan informasi tentang mata kuliah yang ditawarkan dalam program akademik. Kelas ini atribut kode, namaMataKuliah, dan sks untuk menyimpan kode mata kuliah, nama mata kuliah, dan jumlah SKS yang diberikan. Kelas ini juga memiliki dua metode aturJadwal yang di-overload, yang memungkinkan pengguna untuk mengatur jadwal mata kuliah dengan menyertakan hari dan jam, atau hanya hari saja. Metode showInfo() digunakan untuk menampilkan informasi lengkap tentang mata kuliah, termasuk kode, nama, dan jumlah sks. Yang terakhir terdapat metode getSks() yang mengembalikan sks dan mata kuliah, memungkinkan kelas lain untuk mengakses informasi tersebut.

## Main.java

```
package Unguided;

/**
 *
 * @author Raditya Putra Anugra Pratama
 * 2311103143
 * S1SI07-C
 */
public class Main {

    /**
     * Main method
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        // Membuat objek Mahasiswa dan Dosen

        System.out.println("----- Informasi Mahasiswa -----");

        Mahasiswa mhs = new Mahasiswa("Raditya", "123", "2311103143", "Sistem Informasi");

        Dosen dosen = new Dosen("Raditya Putra Anugra Pratama", "456", "D001");


        // Membuat objek MataKuliah

        MataKuliah mk1 = new MataKuliah("SI101", "PBO", 3);

        MataKuliah mk2 = new MataKuliah("SI102", "Basis Data", 3);

        MataKuliah mk3 = new MataKuliah("SI103", "Perancangan Interaksi", 3);


        // Mengatur jadwal MataKuliah (Overloading)

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

        mk2.aturJadwal("Selasa");
```

```

// Menambahkan MataKuliah ke KRS Mahasiswa
mhs.tambahMatakuliah(mk1);

mhs.tambahMatakuliah(mk2);

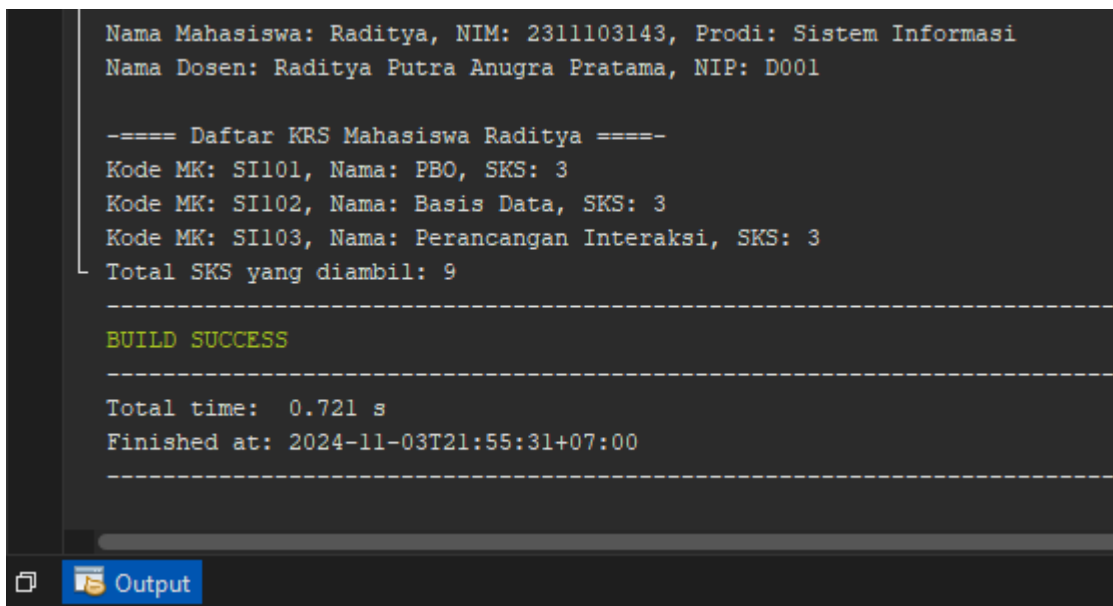
mhs.tambahMatakuliah(mk3);


// Menampilkan informasi
mhs.showInfo();    // Menampilkan informasi Mahasiswa (Polimorfisme)
dosen.showInfo();  // Menampilkan informasi Dosen (Polimorfisme)
mhs.showKRS();     // Menampilkan daftar MataKuliah
}
}

```

Penjelasan:

Menambahkan pemanggilan fungsi showKRS() pada file main sehingga ketika file dijalankan akan menampilkan daftar mata kuliah yang diambil mahasiswa tadi.



```

Nama Mahasiswa: Raditya, NIM: 2311103143, Prodi: Sistem Informasi
Nama Dosen: Raditya Putra Anugra Pratama, NIP: D001

----- Daftar KRS Mahasiswa Raditya -----
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3
Kode MK: SI103, Nama: Perancangan Interaksi, SKS: 3
Total SKS yang diambil: 9

-----
BUILD SUCCESS
-----

Total time:  0.721 s
Finished at: 2024-11-03T21:55:31+07:00
-----

```

**Output:**

Pada output memberikan informasi tentang mahasiswa yang mengambil mata kuliah, dosen yang mengajar, serta detail mata kuliah yang telah diambil dan total SKS yang diperoleh mahasiswa.