

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL III**

“ LATIHAN KONSEP PBO”



Oleh :

Wildan Restu Saputra

2311103136

S1SI-07-C

**PROGRAM STUDI S1 SISTEM INFORMASI
FAKULTAS REKAYASA INDUSTRI DAN DESAIN
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:
 - a. **Person**: Kelas abstrak yang menjadi *superclass* bagi **Mahasiswa** dan **Dosen**.
 - b. **Mahasiswa**: Kelas turunan dari **Person** yang memiliki informasi KRS.
 - c. **Dosen**: Kelas turunan dari **Person** yang mengelola daftar matakuliah yang diajarkan.
 - d. **Matakuliah**: Kelas yang berisi informasi matakuliah, dengan metode **aturJadwal** yang di-overload.
 - e. **KRS**: Interface yang mengatur pengambilan dan penghapusan matakuliah oleh mahasiswa.

Person.java (Source Code)

```
package latihanp7wildan;

/**
 * @author Wildan Restu Saputra
 * 2311103136
 * S1SI-07-C
 */
abstract class Person {

    protected String nama;
    protected String id;

    public Person(String nama, String id) {
```

```
        this.nama = nama;
        this.id = id;
    }

    public abstract void showInfo();
}
```

MataKuliah.java (Source Code)

```
package latihanp7wildan;

/**
 *
 * @author Wildan Restu Saputra
 * 2311103136
 * S1SI-07-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);
    }
}
```

KRS.java (Source Code)

```
package latihanp7wildan;

/**
 *
 * @author Wildan Restu Saputra
 * 2311103136
 * S1SI-07-C
 */
interface KRS {
    void tambahMatakuliah(MataKuliah mk);
    void hapusMatakuliah(MataKuliah mk);
}
```

Dosen.java (Source Code)

```
package latihanp7wildan;

/**
 *
 * @author Wildan Restu Saputra
 * 2311103136
 * S1SI-07-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;
    }

    // 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 (Source Code)

```
package latihanp7wildan;

/**
 *
 * @author Wildan Restu Saputra
 * 2311103136
 * S1SI-07-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);
    }
}
```

Main.java (Source Code)

```
package latihanp7wildan;
```

```
/**
 *
```

```

* @author Wildan Restu Saputra
* 2311103136
* S1SI-07-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("Wildan", "123", "2311103136", "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);

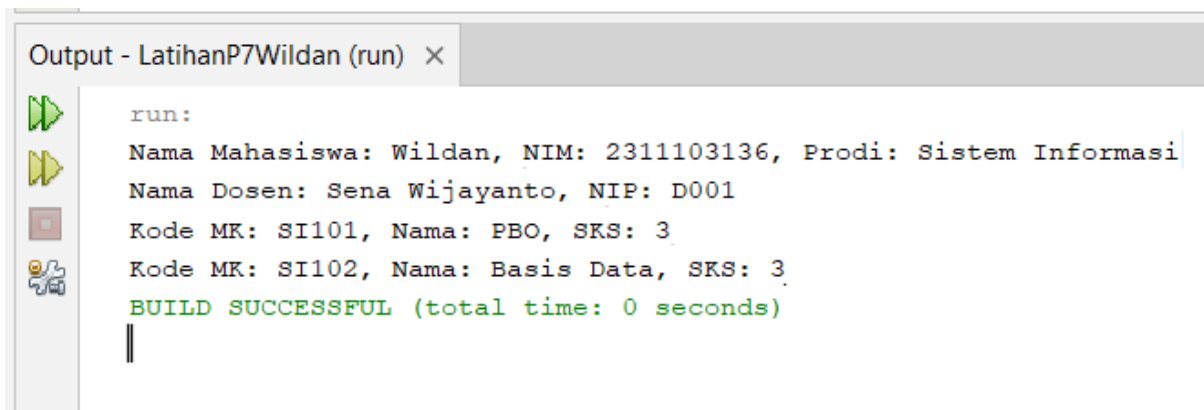
        // Mengatur Jadwal Matakuliah (Overloading)
        mk1.aturJadwal("Senin", "08:00");
        mk2.aturJadwal("Selasa");

        // 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 Output :



```

Output - LatihanP7Wildan (run) x
run:
Nama Mahasiswa: Wildan, NIM: 2311103136, 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)

```

Penjelasan : Dari program diatas merupakan sebuah sistem yang memberikan gambaran yang jelas tentang pengorganisasian informasi akademik. Dalam kelas Main ini objek-objek ini diinstansiasi untuk menunjukkan interaksi antara kelas-kelas tersebut dan menampilkan informasi mengenai dosen, mata kuliah, dan mahasiswa melalui metode `showInfo()`.

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. Tambahkan Menghitung Total SKS yang Diambil oleh Mahasiswa.
Tambahkan metode **`hitungTotalSKS()`** dalam kelas **`Mahasiswa`** untuk menghitung total SKS dari seluruh mata kuliah yang telah diambil.

Mahasiswa.java (Source Code)

```
package latihanp7wildan;

/**
 *
 * @author Wildan Restu Saputra
 * 2311103136
 * S1SI-07-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);
}

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

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

```

MataKuliah.java (Source Code)

```

package latihanp7wildan;

/**
 *
 * @author Wildan Restu Saputra
 * 2311103136
 * S1SI-07-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 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);
    }
}

```

Main.java (Source Code)

```

package latihanp7wildan;

/**
 *
 * @author Wildan Restu Saputra
 * 2311103136
 * S1SI-07-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("Wildan", "123", "2311103136", "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

mhs.showKRS();
int totalSKS = mhs.hitungTotalSKS();
System.out.println("Total SKS yang Diambil: " + totalSKS);
}
}

```

Hasil Output :

Output - LatihanP7Wildan (run)

```

run:
Nama Mahasiswa: Wildan, NIM: 2311103136, 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 oleh Wildan:
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)

```

Penjelasan : Pada kode program diatas merupakan pengembangan dari kode program guided pada class main, mahasiswa, dan matakuliah. Yaitu kita menambahkan dua fitur pada program unguided ini. Pertama yaitu showKRS() dimana mahasiswa bisa melihat daftar mata kuliah yang akan diambil. Kedua hitungTotalSKS() dimana akan menampilkan total SKS dari semua mata kuliah yang kita ambil. Selanjutnya metode getter getSks() ini dapat dengan mudah menjumlah semua SKS. Dari hasil ouput setelah program dijalankan mahasiswa dapat melihat informasi dari diri sendiri, dosen, mata kuliah, dan total SKS yang telah diambil.