

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL III
INHERITANCE, ABSTRACT, INTERFACE



Oleh :

Nama : Najwa Darmawan

NIM : 2311103108

Kelas : S1SI07C

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

I. GUIDED

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.

Jawab:

Kode Person

```
package LatihanP7najwa;
```

```
/**
```

```
*
```

```
* @author Najwa Darmawan
```

```
* 2311103108
```

```
* S1SI07C
```

```
*/
```

```
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 MataKuliah

```

package LatihanP7najwa;

/**
 *
 * @author Najwa Darmawan
 * 2311103108
 * S1SI07C
 */
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);
    }
}

```

Kode Mahasiswa

```

package LatihanP7najwa;
import java.util.ArrayList;
import java.util.List;
/**
 *
 * @author Najwa Darmawan
 * 2311103108
 * S1SI07C
 */
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 Dosen

```

package LatihanP7najwa;
import java.util.ArrayList;
import java.util.List;
/**
 *
 * @author Najwa Darmawan

```

```

* 2311103108
* S1SI07C
*/
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 KRS

```

package LatihanP7najwa;

/**
 *
 * @author Najwa Darmawan
 * 2311103108
 * S1SI07C

```

```

*/

interface KRS {

    void tambahMatakuliah(MataKuliah mk);

    void hapusMatakuliah(MataKuliah mk);

}

```

Kode Main

```

package LatihanP7najwa;

/**
 *
 * @author Najwa Darmawan
 * 2311103108
 * S1SI07C]'
 *
 */
public class Main {

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

    public static void main(String[] args) {

        // Membuat Objek Mahasiswa dan Dosen

        Mahasiswa mhs = new Mahasiswa("Najwa", "123", "2311103108",
"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)

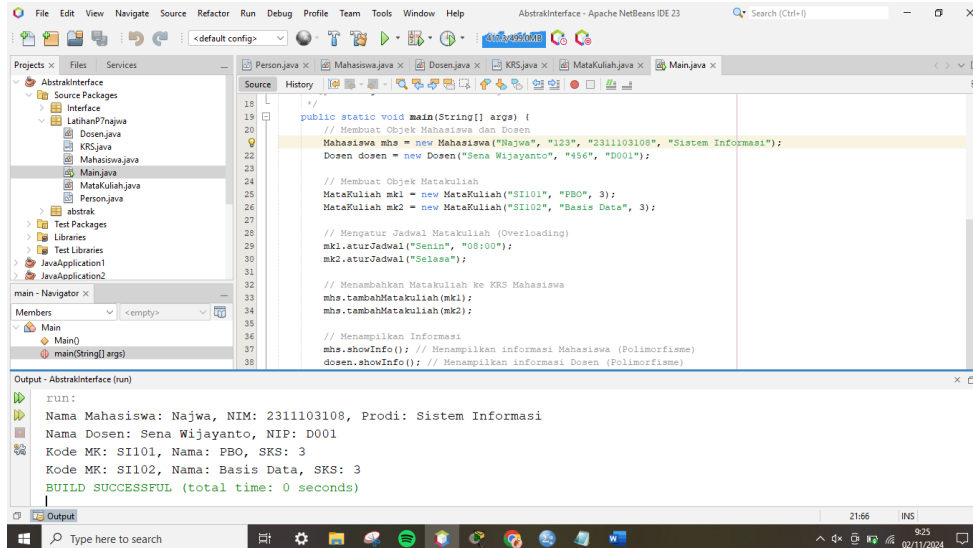
mk1.showInfo(); // Informasi Matakuliah 1
mk2.showInfo(); // Informasi Matakuliah 2

}

}

```

Hasil Run



Penjelasan: Kode program diatas, merupakan kode program yang berisi implementasi konsep abstract dan juga interface didalamnya. Terdapat beberapa kode program, yang terbagi menjadi superclass, dan juga kelas turunan. Hasil run akan menampilkan informasi berupa nama mahasiswa, nim, prodi, nama dosen, nip dosen, dan juga kode mata kuliah.

II. 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 modifikasi Mahasiswa

```
package LatihanP7najwa;

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

/**
 *
 * @author Najwa Darmawan
 * 2311103108
 * S1SI07C
 */

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);
    }

    // Feature 1: Menampilkan Daftar Mata Kuliah yang Diambil
    public void showKRS() {
        System.out.println("\nDaftar KRS Mahasiswa:");
        System.out.println("Nama: " + nama);
        System.out.println("NIM: " + nim);
        System.out.println("Program Studi: " + prodi);
        System.out.println("\nMata Kuliah yang Diambil:");

        if (daftarKRS.isEmpty()) {
            System.out.println("Belum ada mata kuliah yang
diambil.");
        } else {
            for (MataKuliah mk : daftarKRS) {
                mk.showInfo();
            }
            System.out.println("Total SKS: " + hitungTotalSKS());
        }
    }
}

```

```

// Feature 2: Menghitung Total SKS
public int hitungTotalSKS() {
    int totalSKS = 0;
    for (MataKuliah mk : daftarKRS) {
        totalSKS += mk.getSKS(); // Assuming you'll add getSKS()
method in MataKuliah class
    }
    return totalSKS;
}
}

```

Kode modifikasi MataKuliah

```

package LatihanP7najwa;

/**
 *
 * @author Najwa Darmawan
 * 2311103108
 * S1SI07C
 */
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);
    }

    // Added getter method for SKS
    public int getSKS() {
        return this.sks;
    }
}

```

Kode modifikasi Main

```

package LatihanP7najwa;

/**
 *
 * @author Najwa Darmawan
 * 2311103108
 * S1SI07C]'
 *
 */
public class Main {
    public static void main(String[] args) {
        // Membuat Objek Mahasiswa dan Dosen
    }
}

```

```

        Mahasiswa mhs = new Mahasiswa("Najwa", "123", "2311103108",
"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);
        MataKuliah mk3 = new MataKuliah("SI103", "Algoritma", 4);


        // Mengatur Jadwal Matakuliah (Overloading)
        mk1.aturJadwal("Senin", "08:00");
        mk2.aturJadwal("Selasa");
        mk3.aturJadwal("Rabu", "10:00");


        // Menambahkan Matakuliah ke KRS Mahasiswa
        mhs.tambahMatakuliah(mk1);
        mhs.tambahMatakuliah(mk2);
        mhs.tambahMatakuliah(mk3);


        // Menampilkan Informasi

        mhs.showInfo();    //    Menampilkan    informasi    Mahasiswa
(Polimorfisme)

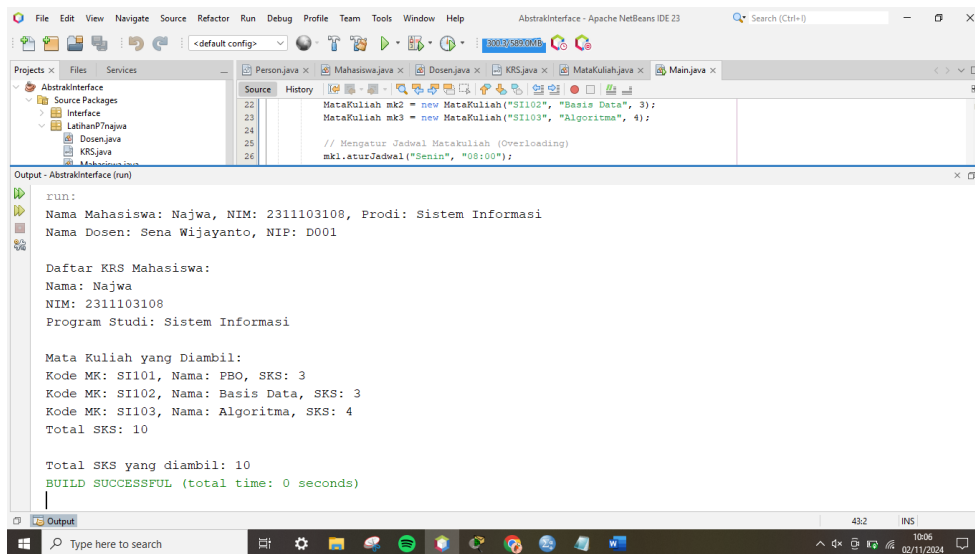
        dosen.showInfo();    //    Menampilkan    informasi    Dosen
(Polimorfisme)


        // Menampilkan KRS dan Total SKS
        mhs.showKRS();

        System.out.println("\nTotal    SKS    yang    diambil:    "    +
mhs.hitungTotalSKS());
    }
}

```

Hasil Run



```
run:
Nama Mahasiswa: Najwa, NIM: 2311103108, Prodi: Sistem Informasi
Nama Dosen: Sena Wijayanto, NIP: D001

Daftar KRS Mahasiswa:
Nama: Najwa
NIM: 2311103108
Program Studi: Sistem Informasi

Mata Kuliah yang Diambil:
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3
Kode MK: SI103, Nama: Algoritma, SKS: 4
Total SKS: 10

Total SKS yang diambil: 10
BUILD SUCCESSFUL (total time: 0 seconds)
```

Penjelasan: Kode program berikut merupakan program modifikasi dari program sebelumnya. Dimana dalam kode program ini, terutama 3 bagian atas tadi terdapat modifikasi yang berupa method `showKRS()` yang berfungsi untuk menampilkan informasi dasar mahasiswa. Selain itu, terdapat juga method `hitungTotalSKS()` yang berfungsi untuk menghitung total atau jumlah SKS keseluruhan yang diambil oleh mahasiswa. Lalu pada hasil run, terdapat bagian yang menampilkan daftar lengkap mata kuliah yang diambil oleh mahasiswa, setiap mata kuliah ditampilkan dengan kode, nama, dan jumlah SKS, total SKS yang diambil pada contoh diatas adalah 10 (hasil penjumlahan $3 + 3 + 4$).