LAPORAN PRAKTIKUM
PEMROGRAM BERORIENTASI OBJEK
MODUL 3

Oleh :

Teuku Bintang Hadi Negara

2311103139

S1SI07C

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS REKAYASA INDUSTRI
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:
   - **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 : Kelas di atas adalah kelas abstrak bernama `Person`, yang berfungsi sebagai template untuk objek-objek yang mewakili individu dengan atribut tertentu. Kelas ini memiliki dua atribut yang bersifat dilindungi (`protected`), yaitu `nama` dan `id`, yang dapat diakses oleh kelas-kelas turunannya. Kelas ini juga memiliki konstruktor yang menerima dua parameter, `nama` dan `id`, yang digunakan untuk menginisialisasi atribut-atribut tersebut saat objek dibuat. Selain itu, kelas `Person` mendeklarasikan metode abstrak `showInfo()`, yang berarti bahwa setiap kelas yang mewarisi `Person` harus menyediakan implementasi spesifik untuk metode ini.

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 : Kelas di atas bernama `MataKuliah`, yang dirancang untuk merepresentasikan mata kuliah di sebuah institusi pendidikan. Kelas ini memiliki atribut privat, yaitu `kode`, `namaMatakuliah`, `sks`, `jadwalHari`, dan `jadwalJam`, yang menyimpan informasi penting tentang mata kuliah tersebut. Konstruktor `MataKuliah` menerima tiga parameter—`kode`, `namaMatakuliah`, dan `sks`—untuk menginisialisasi atribut-atribut yang relevan saat objek baru

dibuat. Kelas 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 informasi dasar mengenai mata kuliah, seperti kode, nama, dan jumlah SKS, dalam format yang mudah dibaca.

KRS.java

```
interface KRS {
    void tambahMatakuliah(MataKuliah mk);
    void hapusMatakuliah(MataKuliah mk);
}
```

Penjelasan : Interface KRS didefinisikan sebagai kontrak yang menetapkan dua metode dasar yang harus diimplementasikan oleh kelas yang mengadopsinya. Metode pertama, tambahMatakuliah(MataKuliah mk), bertujuan untuk menambahkan objek mata kuliah ke dalam sistem Kartu Rencana Studi (KRS), sementara metode kedua, hapusMatakuliah(MataKuliah mk), digunakan untuk menghapus mata kuliah yang telah ditambahkan sebelumnya.

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`, 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 atribut `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

```
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 kelas abstrak `Person` dan mengimplementasikan interface `KRS`, yang memungkinkan pengelolaan Kartu Rencana Studi (KRS) bagi mahasiswa. Kelas ini memiliki atribut privat `nim` untuk menyimpan Nomor Induk

Mahasiswa, serta `prodi` yang menunjukkan program studi yang diambil. Sebuah list bernama `daftarKRS` digunakan untuk menyimpan mata kuliah yang dipilih oleh mahasiswa. Konstruktor `Mahasiswa` memanggil konstruktor dari superclass `Person` untuk menginisialisasi atribut nama dan id, serta mengatur nilai untuk `nim` dan `prodi`. Metode `showInfo` diimplementasikan untuk menampilkan informasi spesifik tentang mahasiswa, termasuk nama, NIM, dan program studi. Selain itu, kelas ini juga menyediakan implementasi untuk metode dari interface `KRS`, yaitu `tambahMatakuliah` dan `hapusMatakuliah`, yang memungkinkan mahasiswa untuk menambahkan atau menghapus mata kuliah dari daftar KRS mereka.

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("Bintang", "123", "2311103139", "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 : Di dalam metode `main`, objek `Mahasiswa` dan `Dosen` dibuat dengan memberikan informasi seperti nama, id, NIM, dan program studi untuk mahasiswa, serta NIP untuk dosen. Selanjutnya, dua objek `MataKuliah` dibuat, yaitu `mk1` dan `mk2`, dengan kode, nama, dan jumlah SKS yang sesuai. Jadwal untuk kedua mata kuliah tersebut diatur menggunakan metode

overloading `aturJadwal`, di mana `mk1` diberikan jadwal lengkap dengan hari dan jam, sementara `mk2` hanya ditentukan hari saja. Setelah itu, mahasiswa menambahkan kedua mata kuliah tersebut ke dalam daftar Kartu Rencana Studi (KRS) mereka. Terakhir, informasi mengenai mahasiswa, dosen, dan mata kuliah ditampilkan menggunakan metode `showInfo`, yang menonjolkan konsep polimorfisme, di mana metode yang sama dapat digunakan untuk objek dari kelas yang berbeda, memberikan tampilan yang konsisten dan terstruktur terhadap data yang relevan.

Output

```
Nama Mahasiswa: Bintang, NIM: 2311103139, Prodi: Sistem Informasi
Nama Dosen: Sena Wijayanto, NIP: D001
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3
```

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.

   Mahasiswa.java

```
package latihanp7teukubintanghadinegara;

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("");
        System.out.println("===== Daftar KRS Mahasiswa " + nama + "=====");
        if (daftarKRS.isEmpty()) {
            System.out.println("Tidak ada mata kuliah yang diambil.");
        } else {
            for (MataKuliah mk : daftarKRS) {
                mk.showInfo(); // Memanggil showInfo() dari kelas MataKuliah
            }
        }
    }
}
```

Penjelasan : Dalam kodingan diatas kita menambahkan metode showKRS(), metode ini mencetak judul yang menunjukkan nama mahasiswa yang bersangkutan. Selanjutnya, metode melakukan pengecekan apakah daftar mata kuliah (daftarKRS) kosong. Jika kosong, program akan mencetak pesan yang menyatakan bahwa tidak ada mata kuliah yang diambil, memberi tahu pengguna bahwa mahasiswa tersebut belum mendaftar untuk mata kuliah apa pun. Sebaliknya, jika daftar tidak kosong, metode akan menggunakan loop for-each untuk iterasi melalui setiap objek MataKuliah dalam daftar tersebut. Dalam setiap iterasi, metode showInfo() dari objek MataKuliah dipanggil untuk menampilkan informasi lengkap mengenai setiap mata kuliah, seperti kode, nama, dan jumlah SKS. Dengan cara ini, metode showKRS() memberikan gambaran yang jelas tentang mata kuliah yang diambil mahasiswa, serta memperlihatkan status akademik mereka.

Main.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package latihanp7teukubintanghadinegara;

/**
 *
 * @author TeukuBintangHadiNegara
 */
public class Latihanp7TeukuBintangHadiNegara {

    /**
     * @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("Bintang", "123", "2311103139", "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", "Jaringan Komputer", 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 mata kuliah
    }
```

```
}
```

Penjelasan : Untuk di file main kita hanya memanggil showKRS() untuk menampilkan daftar mata kuliah yang diambil mahasiswa.

Output

```
run:
===== Informasi Mahasiswa =====
Nama Mahasiswa: Bintang, NIM: 2311103139, Prodi: Sistem Informasi

===== Informasi Dosen =====
Nama Dosen: Sena Wijayanto, NIP: D001

===== Daftar KRS Mahasiswa Bintang=====
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3
Kode MK: SI103, Nama: Jaringan Komputer, SKS: 3
BUILD SUCCESSFUL (total time: 0 seconds)
```

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.

Mahasiswa.java

```
package latihanp7teukubintanghadinegara;

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);
    }

    // Metode untuk menghitung total SKS
    public int hitungTotalSKS() {
        int totalSks = 0; // Inisialisasi total SKS
        for (MataKuliah mk : daftarKRS) {
            totalSks += mk.getSks(); // Menggunakan getter untuk menambahkan SKS
        }
        return totalSks; // Mengembalikan total SKS
    }

    public void showKRS() {
        System.out.println("");
        System.out.println("===== Daftar KRS Mahasiswa " + nama + "=====");
        if (daftarKRS.isEmpty()) {
            System.out.println("Tidak ada mata kuliah yang diambil.");
        } else {
            for (MataKuliah mk : daftarKRS) {
                mk.showInfo(); // Memanggil showInfo() dari kelas MataKuliah
            }
            System.out.println("Total SKS: " + hitungTotalSKS()); // Menampilkan total SKS
        }
    }
}
```

Penjelasan : Dalam kodingan diatas kita menambahkan metode hitungTotalSKS() metode ini bertanggung jawab untuk menghitung total SKS dari seluruh mata kuliah yang diambil oleh mahasiswa. Proses dimulai dengan menginisialisasi variabel totalSks dengan nilai 0, yang akan digunakan untuk menyimpan hasil penjumlahan SKS. Selanjutnya, metode ini menggunakan loop for-each untuk iterasi melalui setiap objek MataKuliah dalam daftar KRS (daftarKRS). Dalam setiap iterasi, metode getSks() dari objek MataKuliah dipanggil untuk mendapatkan jumlah SKS mata kuliah tersebut, dan nilai ini ditambahkan ke totalSks. Setelah loop selesai, metode ini mengembalikan nilai totalSks, memberikan total SKS yang diambil mahasiswa

sebagai hasil. Dan di metode showKRS() kita memamnggil metode hitungTotalSKS() untuk menjumlahkan sks yang ada

MataKuliah.java

```
package latihanp7teukubintanghadinegara;

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);
    }
}
```

Penjelasan : Dalam kodingan diatas kita menambahkan metode getSks(), metode getSks() adalah sebuah *getter* yang dirancang untuk memberikan akses ke nilai atribut sks di dalam kelas MataKuliah. Atribut sks, yang menyimpan jumlah sks dari suatu mata kuliah, dideklarasikan sebagai private, yang berarti bahwa nilai ini tidak dapat diakses langsung dari luar kelas tersebut. Dengan adanya metode getter ini, pengguna kelas MataKuliah dapat mengambil nilai SKS tanpa perlu mengubah atau mengakses atribut secara langsung.

Output

```
run:
===== Informasi Mahasiswa =====
Nama Mahasiswa: Bintang, NIM: 2311103139, Prodi: Sistem Informasi

===== Informasi Dosen =====
Nama Dosen: Sena Wijayanto, NIP: D001

===== Daftar KRS Mahasiswa Bintang=====
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Basis Data, SKS: 3
Kode MK: SI103, Nama: Jaringan Komputer, SKS: 3
Total SKS: 9
```