

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 3
LATIHAN KONSEP PBO



Oleh :

Muhammad Raflan Kemal

2311103132

S1SI-07-C

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

I. GUIDED

1. Buat project baru dengan LatihanP7Nama. 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.

Kode Program

- a. Person.java

```
package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */
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();
}
```

Kelas abstrak bernama Person, yang dirancang sebagai dasar untuk merepresentasikan orang dengan atribut nama dan id. Kelas ini memiliki konstruktor untuk mengatur nilai kedua atribut tersebut. Selain itu, ada metode abstrak *showInfo()*, yang harus diimplementasikan oleh kelas-kelas turunan agar dapat menampilkan informasi spesifik tentang setiap orang. Dengan demikian, kelas ini memungkinkan pengorganisasian dan pengelolaan data tentang individu secara lebih terstruktur dalam pemrograman berorientasi objek.

b. Mahasiswa.java

```
package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 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);
    }
}
```

Kelas Mahasiswa, yang merupakan turunan dari kelas abstrak Person dan mengimplementasikan interface KRS. Kelas ini menyimpan informasi penting tentang mahasiswa, seperti nim (nomor induk mahasiswa), prodi (program

studi), dan daftarKRS, yang berfungsi untuk mencatat mata kuliah yang mereka ambil. Konstruktor Mahasiswa digunakan untuk menginisialisasi nama, ID, NIM, dan program studi dengan memanggil konstruktor dari kelas Person. Metode showInfo() digunakan untuk menampilkan informasi dasar tentang mahasiswa, termasuk nama, NIM, dan program studi mereka.

c. Dosen.java

```
package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 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
    public void showInfo() {
        System.out.println("Nama Dosen: " + nama + ", NIP: " + nip);
    }

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

}
```

Kelas Dosen, yang merupakan subclass dari kelas abstrak Person. Kelas ini memiliki atribut khusus seperti nip (nomor induk pegawai) dan daftarMatakuliah, yang merupakan list untuk menyimpan mata kuliah yang diajarkan oleh dosen. Konstruktor Dosen digunakan untuk menginisialisasi nama, ID, dan NIP dengan memanggil konstruktor dari kelas Person. Metode

showInfo() menampilkan informasi dasar tentang dosen, termasuk nama dan NIP mereka.

d. MataKuliah.java

```
package latihanp7raflankemal;

/**
 *
 * @author Muhammad RAflan Kemal
 * 2311103132
 * 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;
    }

    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);
    }
}
```

Kelas MataKuliah, yang dirancang untuk merepresentasikan informasi tentang mata kuliah di dalam sistem akademik. Kelas ini memiliki beberapa atribut penting, seperti kode (kode mata kuliah), namaMatakuliah (nama mata kuliah), sks (satuan kredit semester), serta jadwalHari dan jadwalJam, yang menyimpan informasi mengenai jadwal mata kuliah.

e. KRS.java

```
package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */
interface KRS {

    void tambahMatakuliah(MataKuliah mk);

    void hapusMatakuliah(MataKuliah mk);
}
```

Interface KRS, yang mendefinisikan kontrak untuk pengelolaan mata kuliah dalam sistem akademik. Interface ini memiliki dua metode penting `tambahMatakuliah(MataKuliah mk)` dan `hapusMatakuliah(MataKuliah mk)`. Metode `tambahMatakuliah(MataKuliah mk)` digunakan untuk menambahkan mata kuliah ke dalam daftar yang dimiliki oleh kelas yang mengimplementasikan interface ini, sedangkan `hapusMatakuliah(MataKuliah mk)` berfungsi untuk menghapus mata kuliah dari daftar tersebut.

f. Main.java

```
package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Mahasiswa mhs = new Mahasiswa("Muhammad Raflan Kemal", "123",
"2211109001", "Sistem Informasi");
        Dosen dosen = new Dosen("Adzana Ashel", "456", "D001");

        MataKuliah mk1 = new MataKuliah("SI101", "PBO", 3);
        MataKuliah mk2 = new MataKuliah("SI102", "Jaringan Komputer", 3);
    }
}
```

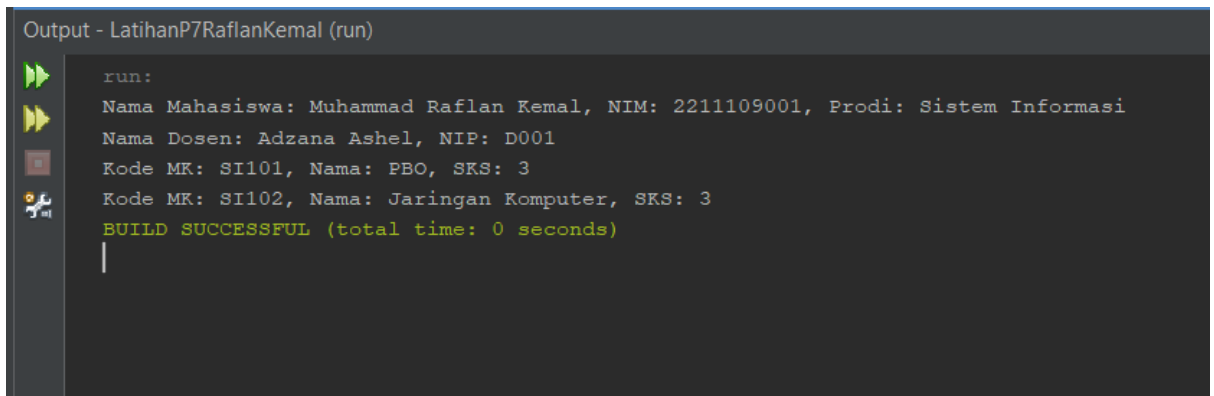
```
mk1.aturJadwal("Senin", "08:00");
mk2.aturJadwal("Selasa");

mhs.tambahMatakuliah(mk1);
mhs.tambahMatakuliah(mk2);

mhs.showInfo();
dosen.showInfo();
mk1.showInfo();
mk2.showInfo();
}
}
```

Kelas Main, yang berfungsi sebagai titik masuk (entry point) untuk menjalankan program. Di dalam metode main, program ini membuat objek-objek yang diperlukan untuk mendemonstrasikan fungsionalitas sistem akademik. Pertama, objek Mahasiswa dan Dosen dibuat dengan memberikan informasi dasar seperti nama, ID, NIM, dan NIP. Kemudian, dua objek MataKuliah dibuat, masing-masing untuk mata kuliah "PBO" dan "Jaringan Komputer", lengkap dengan kode dan jumlah SKS yang sesuai. Jadwal untuk kedua mata kuliah diatur menggunakan metode aturJadwal(), di mana satu mata kuliah memiliki jadwal hari dan jam, sedangkan yang lainnya hanya memiliki jadwal hari. Setelah itu, kedua mata kuliah ditambahkan ke dalam daftar mata kuliah yang diambil oleh mahasiswa dengan menggunakan metode tambahMatakuliah(). Terakhir, program menampilkan informasi tentang mahasiswa, dosen, dan kedua mata kuliah dengan memanggil metode showInfo() pada masing-masing objek. Dengan demikian, kelas Main mengilustrasikan bagaimana semua komponen dalam sistem akademik saling berinteraksi dan berfungsi secara keseluruhan.

Hasil Output :

A screenshot of an IDE's output window titled "Output - LatihanP7RaflanKemal (run)". The window has a dark background with a light-colored text area. On the left side of the text area, there are four icons: a green play button, a yellow play button, a red square, and a blue icon with a white 'e'. The text in the output area is as follows:

```
run:
Nama Mahasiswa: Muhammad Raflan Kemal, NIM: 2211109001, Prodi: Sistem Informasi
Nama Dosen: Adzana Ashel, NIP: D001
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Jaringan Komputer, SKS: 3
BUILD SUCCESSFUL (total time: 0 seconds)
```

Penjelasan :

Pada kode program di atas terdiri dari berbagai kelas yang saling terkait. Pertama, ada kelas abstrak `Person` yang menjadi dasar bagi dua kelas turunan, yaitu `Mahasiswa` dan `Dosen`. Kelas `Person` menyimpan informasi dasar seperti nama dan ID, serta memiliki metode `showInfo()` untuk menampilkan data tersebut. Di sisi lain, kelas `Mahasiswa` menambahkan detail seperti NIM, program studi, dan daftar mata kuliah yang diambil, serta mengimplementasikan interface `KRS` untuk mengelola penambahan dan penghapusan mata kuliah. Kelas `Dosen` bertugas mengatur mata kuliah yang diajarkan. Selain itu, kelas `MataKuliah` menyimpan informasi tentang setiap mata kuliah, termasuk kode dan jadwal, serta memiliki metode overload untuk pengaturan jadwal. Dalam kelas `Main`, objek-objek ini diinstansiasi untuk menunjukkan interaksi antara kelas-kelas tersebut dan menampilkan informasi mengenai mahasiswa, dosen, dan mata kuliah melalui metode `showInfo()`. Secara keseluruhan, sistem ini memberikan gambaran yang jelas tentang pengorganisasian informasi akademik.

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.

Kode Program

update Mahasiswa.java

```
package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 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();
    }
}

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

```

Kelas Mahasiswa adalah turunan dari kelas abstrak Person dan juga mengimplementasikan interface KRS. Kelas ini menyimpan informasi penting seperti nim (nomor induk mahasiswa), prodi (program studi), 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

update MataKuliah.java

```

package latihanp7raflankemal;

/**
 *
 * @author Muhammad RAflan Kemal
 * 2311103132
 * 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;
    }
}

```

```

    }

    public int getSks() {
        return 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);
    }
}

```

Kelas MataKuliah, yang dirancang untuk merepresentasikan mata kuliah dalam sistem akademik. Kelas ini memiliki beberapa atribut penting, termasuk kode (kode mata kuliah), namaMatakuliah (nama mata kuliah), sks (satuan kredit semester), serta atribut untuk menyimpan jadwal, yaitu jadwalHari dan jadwalJam. Konstruktor MataKuliah digunakan untuk menginisialisasi nilai kode, nama mata kuliah, dan SKS saat objek baru dibuat. Kelas ini juga menyediakan metode getSks() untuk mengambil jumlah SKS dari mata kuliah tersebut. Metode aturJadwal(String hari, String jam) memungkinkan pengguna untuk menetapkan jadwal lengkap dengan hari dan jam, sedangkan overload metode aturJadwal(String hari) memungkinkan pengguna untuk hanya menetapkan hari saja. Metode showInfo() berfungsi untuk menampilkan informasi dasar mengenai mata kuliah, termasuk kode, nama, dan jumlah SKS. Dengan demikian, kelas MataKuliah berperan penting dalam pengelolaan data akademik, memungkinkan mahasiswa dan dosen untuk mengakses dan mengatur informasi terkait perkuliahan

update Main.java

```

package latihanp7raflankemal;

/**
 *
 * @author Muhammad Raflan Kemal
 * 2311103132
 * 07C
 */
public class Main {
    public static void main(String[] args) {

```

```

        Mahasiswa mhs = new Mahasiswa("Muhammad Raflan Kemal", "123",
"2311103132", "Sistem Informasi");
        Dosen dosen = new Dosen("Adzana Ashel", "456", "D001");

        MataKuliah mk1 = new MataKuliah("SI101", "PBO", 3);
        MataKuliah mk2 = new MataKuliah("SI102", "Jaringan Komputer", 3);

        mk1.aturJadwal("Senin", "08:00");
        mk2.aturJadwal("Selasa");

        mhs.tambahMatakuliah(mk1);
        mhs.tambahMatakuliah(mk2);

        mhs.showInfo();
        dosen.showInfo();
        mk1.showInfo();
        mk2.showInfo();

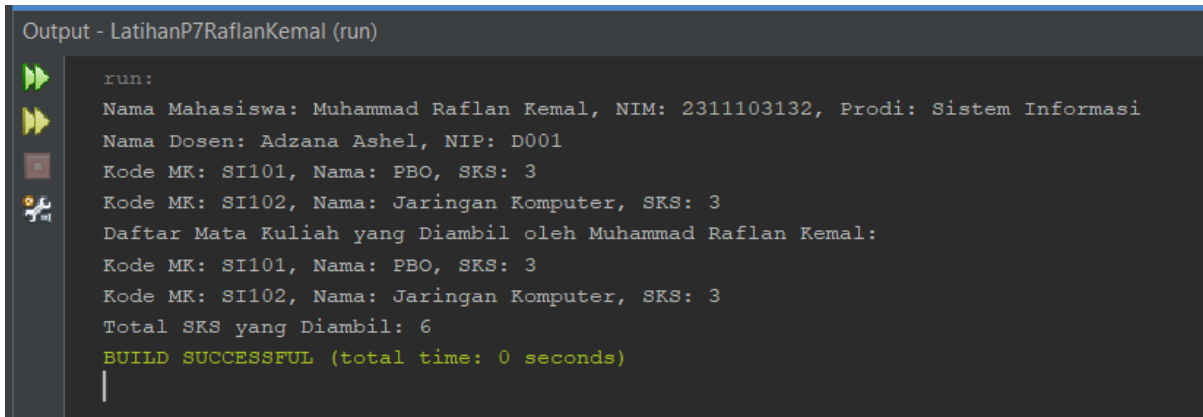
        mhs.showKRS();

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

```

Kelas Main, yang berfungsi sebagai titik awal untuk menjalankan program. Di dalam metode main, program ini menciptakan objek-objek yang diperlukan untuk mendemonstrasikan bagaimana sistem akademik bekerja. Pertama, objek Mahasiswa dan Dosen dibuat dengan memberikan informasi dasar seperti nama, ID, NIM, dan NIP. Selanjutnya, dua objek MataKuliah dibuat untuk mata kuliah "PBO" dan "Jaringan Komputer", lengkap dengan kode dan jumlah SKS yang sesuai. Setelah itu, jadwal untuk masing-masing mata kuliah diatur menggunakan metode aturJadwal(), di mana satu mata kuliah mendapatkan jadwal hari dan jam, sementara yang lainnya hanya diatur untuk hari saja. Kemudian, kedua mata kuliah ditambahkan ke dalam daftar yang diambil oleh mahasiswa menggunakan metode tambahMatakuliah(). Program ini kemudian menampilkan informasi tentang mahasiswa, dosen, dan kedua mata kuliah dengan memanggil metode showInfo() pada masing-masing objek. Selain itu, metode showKRS() digunakan untuk menampilkan daftar mata kuliah yang diambil oleh mahasiswa, dan total SKS yang diambil dihitung dengan menggunakan metode hitungTotalSKS(), lalu hasilnya ditampilkan di konsol. Dengan demikian, kelas Main memberikan gambaran tentang bagaimana semua komponen dalam sistem akademik saling berinteraksi dan berfungsi dengan baik.

Hasil Output :



```
Output - LatihanP7RaflanKemal (run)

run:
Nama Mahasiswa: Muhammad Raflan Kemal, NIM: 2311103132, Prodi: Sistem Informasi
Nama Dosen: Adzana Ashel, NIP: D001
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Jaringan Komputer, SKS: 3
Daftar Mata Kuliah yang Diambil oleh Muhammad Raflan Kemal:
Kode MK: SI101, Nama: PBO, SKS: 3
Kode MK: SI102, Nama: Jaringan Komputer, SKS: 3
Total SKS yang Diambil: 6
BUILD SUCCESSFUL (total time: 0 seconds)
```

Penjelasan :

Pada kode program unguide ini saya menambahkan dua fitur. Yang pertama adalah metode `showKRS()`, yang memungkinkan mahasiswa untuk melihat daftar mata kuliah yang mereka ambil. Metode ini berfungsi dengan mengiterasi daftar mata kuliah dan menampilkan informasi tentang masing-masing. Yang kedua, saya juga memperkenalkan metode `hitungTotalSKS()`, yang bertugas menghitung total SKS dari semua mata kuliah yang diambil. Dengan memanfaatkan getter `getSks()` di kelas `MataKuliah`, metode ini dapat dengan mudah menjumlahkan semua SKS. Di dalam kelas `Main`, saya mengintegrasikan kedua metode ini sehingga ketika program dijalankan, mahasiswa dapat melihat informasi diri, dosen, mata kuliah, serta total SKS yang telah mereka ambil. Penambahan ini membuat sistem lebih fungsional dan membantu mahasiswa dalam merencanakan studi mereka dengan lebih efektif.