

# **LAPORAN PRAKTIKUM PERTEMUAN-5**

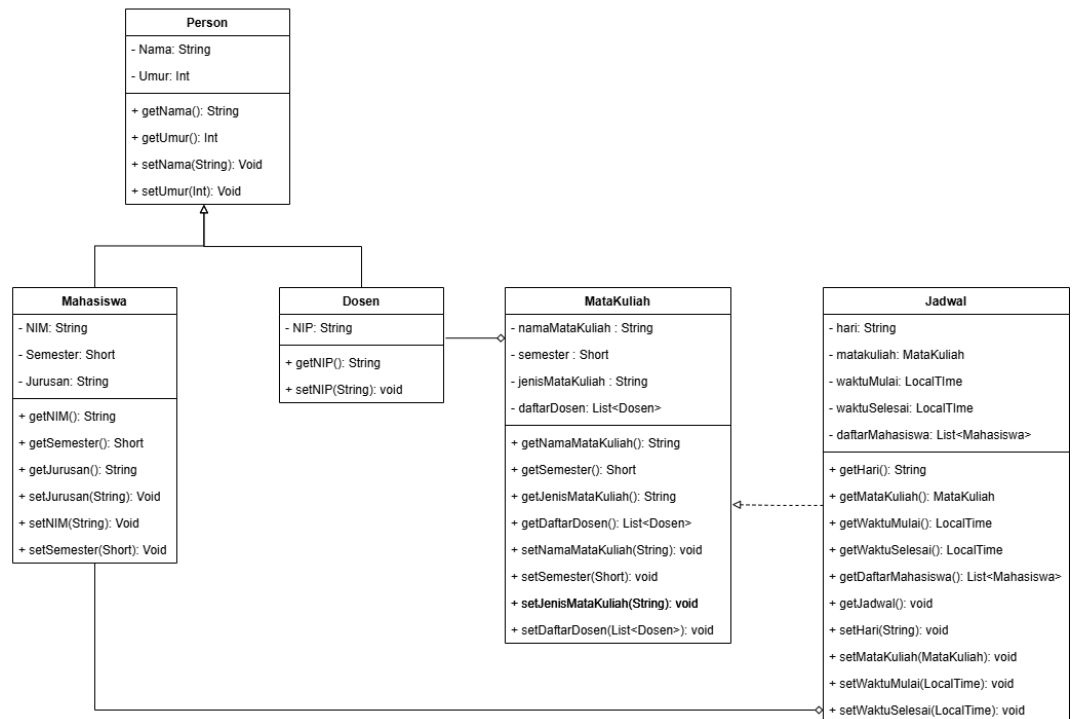
Diajukan untuk memenuhi salah satu tugas praktikum Mata kuliah Pemrograman Berorientasi Objek



**Disusun Oleh:  
Daiva Raditya Pradipa (231511039)  
Jurusan Teknik Komputer dan Informatika**

**Program Studi D-3 Teknik Informatika  
Politeknik Negeri Bandung  
2024**

## 1. Class Diagram



Penjelasan:

- Class Person merupakan Generalization dari class Mahasiswa dan Dosen yang bersifat abstract class (SuperClass)
- Class Mahasiswa dan Dosen Spesialization dari class Person (SubClass, concrete class). Kedua class ini dapat mewarisi atribut dan method yang dimiliki class Person
- Class Dosen beragregasi dengan class MataKuliah karena dalam 1 mata kuliah dapat diajar oleh lebih dari 1 dosen
- Class Jadwal berdependency dengan class MataKuliah karena dalam class ini memerlukan class MataKuliah dan Atribut ListDosen (yang ada pada class MataKuliah) agar dapat beroperasi.
- Class Mahasiswa beragregasi dengan class Jadwal, karena dalam 1 jadwal mata kuliah dapat diikuti oleh lebih dari 1 mahasiswa

## 2. Hasil akhir program

```
Masukan Pilihan :
1
List Mahasiswa :

Andi
Budi
Cici
Dina
Eko
Fina
Gilang
Hana
Ivan
Joko
```

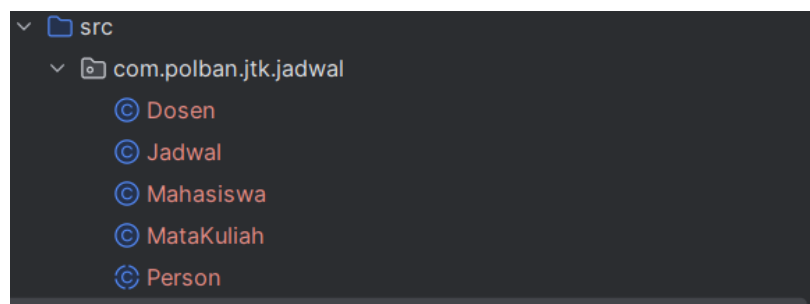
Process finished with exit code 0

```
Run Main x Main x
Masukan Pilihan :
2
List Dosen :
Dr. Budi
Dr. Sari
Dr. Andi
Dr. Rina
Dr. Ali
Dr. Maya
Dr. Johan
Dr. Lisa
Dr. Agus
Dr. Tina
Process finished with exit code 0
```

```
Run Main x Main x
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.0.2\lib\idea_rt.jar=49887:C:\Program Files\JetBrains\IntelliJ IDEA 202
1. Tampilkan list mahasiswa
2. Tampilkan list Dosen
3. Tampilkan jadwal (senin-jumaat)
Masukan Pilihan :
3
TABEL JADWAL KULIAH
+-----+-----+-----+-----+-----+-----+-----+
| Hari | Waktu Mulai | Waktu Selesai | Ruangan | Mata Kuliah | Dosen | Mahasiswa |
+-----+-----+-----+-----+-----+-----+-----+
| Senin | 08:00 | 10:00 | Ruang 101 | Pemrograman Java | Dr. Budi | Mahasiswa: | |
| | | | | | | | Andi |
| | | | | | | | Budi |
| | | | | | | | Cici |
| | | | | | | | Dina |
| | | | | | | | Eko |
| | | | | | | | Fina |
| | | | | | | | Gilang |
| | | | | | | | Hana |
| | | | | | | | Ivan |
| | | | | | | | Joko |
+-----+-----+-----+-----+-----+-----+-----+
| Senin | 10:00 | 12:00 | Ruang 102 | Struktur Data | Dr. Sari | Mahasiswa: |
| | | | | | | | Andi |
+-----+-----+-----+-----+-----+-----+-----+
```

Penjelasan:

Untuk mengimplementasikan class diagram tersebut dalam program Java, pertama disini saya membuat package terlebih dahulu dengan nama “com.polban.jtk.jadwal” yang akan menampung seluruh class class yang akan kita buat sesuai class diagram yang telah dibuat



Kemudian kita perlu untuk mengisi atribut dan method pada setiap class sesuai dengan yang ada pada class diagram,

```

package com.polban.jtk.jadwal;

abstract class Person {
    private String name;
    private int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setAge(int age) {
        this.age = age;
    }
}

package com.polban.jtk.jadwal;

public class Dosen extends Person {
    private String nip;

    public Dosen(String nama, int age, String nip) {
        super(nama, age);
        this.nip = nip;
    }

    public String getNip() {
        return nip;
    }

    public void setNip(String nip) {
        this.nip = nip;
    }
}

package com.polban.jtk.jadwal;

import java.util.List;

public class Mahasiswa extends Person {
    private String rnk;
    private int semester;
    private String jurusan;
    private String Prodi;

    public Mahasiswa(String nama, int age, String RNM, short semester, String Jurusan, String Prodi) {
        super(nama, age);
        this.rnk = RNM;
        this.semester = semester;
        this.jurusan = Jurusan;
        this.Prodi = Prodi;
    }

    public String getRNM() {
        return rnk;
    }

    public int getSemester() {
        return semester;
    }
}

package com.polban.jtk.jadwal;

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

public class MataKuliah {
    private String namaMataKuliah;
    private short semester;
    private String jenisMataKuliah;
    private List<Dosen> dosen;

    // Kontruktor
    public MataKuliah(String namaMataKuliah, short semester, String jenisMataKuliah, List<Dosen> dosen) {
        this.namaMataKuliah = namaMataKuliah;
        this.semester = semester;
        this.jenisMataKuliah = jenisMataKuliah;
        this.dosen = dosen;
    }

    // Getter dan Setter untuk namaMataKuliah
    public String getNamaMataKuliah() {
        return namaMataKuliah;
    }

    public void setNamaMataKuliah(String namaMataKuliah) {
        this.namaMataKuliah = namaMataKuliah;
    }
}

package com.polban.jtk.jadwal;

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.List;

public class Jadwal {
    private String hari;
    private MataKuliah matakuliah;
    private LocalDate waktuMulai;
    private LocalDate waktuSelesai;
    private String ruangan;
    private List<Mahasiswa> daftarMahasiswa;
    private String namaDosen;

    // Constructor
    public Jadwal(String hari, MataKuliah matakuliah, LocalDate waktuMulai, LocalDate waktuSelesai, String ruangan, List<Mahasiswa> daftarMahasiswa, String namaDosen) {
        this.hari = hari;
        this.matakuliah = matakuliah;
        this.waktuMulai = waktuMulai;
        this.waktuSelesai = waktuSelesai;
        this.ruangan = ruangan;
        this.daftarMahasiswa = daftarMahasiswa;
        this.namaDosen = namaDosen;
    }

    public String getHari() {
        return hari;
    }
}

```

Pada class Jadwal, terdapat fungsi displayJadwal yang berfungsi untuk menampilkan satu baris tabel jadwal. Fungsi ini menampilkan informasi penting tentang jadwal mata kuliah seperti hari, waktu mulai dan selesai, ruangan, mata kuliah, dosen pengajar, serta daftar mahasiswa yang terdaftar pada jadwal tersebut.

```

public void displayJadwal() {
    DateTimeFormatter timeFormatter = DateTimeFormatter.ofPattern("HH:mm");

    System.out.printf("| %-17s | %-17s | %-17s | %-17s | %-17s | %-17s | Mahasiswa:\n",
        hari,
        waktuMulai.format(timeFormatter),
        waktuSelesai.format(timeFormatter),
        ruangan,
        matakuliah.getNamaMataKuliah(),
        matakuliah.getDosen(namaDosen).getName());

    // Menampilkan daftar mahasiswa
    for (Mahasiswa mhs : daftarMahasiswa) {
        System.out.printf("| %-17s | %-17s | %-17s | %-17s | %-17s | %-17s | \n",
            mhs.getRNM(),
            mhs.getSemester(),
            mhs.getJurusan(),
            mhs.getProdi(),
            mhs.getNamaMataKuliah(),
            mhs.getNamaDosen());
    }
    System.out.println("-----");
}

```

Kemudian, kita hanya perlu membuat object pada masing masing class (kecuali class person karena dia merupakan sebuah abстракт class sehingga tidak bisa di instansikan atau dibuat object) pada main file. Dikarenakan pada soal kita diminta untuk

menampilkan 10 mahasiswa, dan 10 dosen, maka kita akan membuat 10 object mahasiswa dan dosen yang akan dibuat dalam bentuk list.

```
List<Dosen> listDosen = new ArrayList<>();

// Membuat objek Dosen
Dosen dsn1 = new Dosen(nama: "Dr. Budi", age: 45, nip: "12321511321");
Dosen dsn2 = new Dosen(nama: "Dr. Sari", age: 40, nip: "12321511322");
Dosen dsn3 = new Dosen(nama: "Dr. Andi", age: 50, nip: "12321511323");
Dosen dsn4 = new Dosen(nama: "Dr. Rina", age: 35, nip: "12321511324");
Dosen dsn5 = new Dosen(nama: "Dr. Ali", age: 55, nip: "12321511325");
Dosen dsn6 = new Dosen(nama: "Dr. Maya", age: 42, nip: "12321511326");
Dosen dsn7 = new Dosen(nama: "Dr. Johan", age: 48, nip: "12321511327");
Dosen dsn8 = new Dosen(nama: "Dr. Lisa", age: 38, nip: "12321511328");
Dosen dsn9 = new Dosen(nama: "Dr. Agus", age: 52, nip: "12321511329");
Dosen dsn10 = new Dosen(nama: "Dr. Tina", age: 37, nip: "12321511330");

listDosen.add(dsn1);
listDosen.add(dsn2);
listDosen.add(dsn3);
listDosen.add(dsn4);
listDosen.add(dsn5);
listDosen.add(dsn6);
listDosen.add(dsn7);
listDosen.add(dsn8);
listDosen.add(dsn9);
listDosen.add(dsn10);

// Membuat objek Mahasiswa
Mahasiswa mhs1 = new Mahasiswa(nama: "Andi", age: 20, NIM: "1900001", listDosen);
Mahasiswa mhs2 = new Mahasiswa(nama: "Budi", age: 21, NIM: "1900002", listDosen);
Mahasiswa mhs3 = new Mahasiswa(nama: "Cici", age: 19, NIM: "1900003", listDosen);
Mahasiswa mhs4 = new Mahasiswa(nama: "Dina", age: 22, NIM: "1900004", listDosen);
Mahasiswa mhs5 = new Mahasiswa(nama: "Eko", age: 20, NIM: "1900005", listDosen);
Mahasiswa mhs6 = new Mahasiswa(nama: "Fina", age: 21, NIM: "1900006", listDosen);
Mahasiswa mhs7 = new Mahasiswa(nama: "Gilang", age: 19, NIM: "1900007", listDosen);
Mahasiswa mhs8 = new Mahasiswa(nama: "Hana", age: 20, NIM: "1900008", listDosen);
Mahasiswa mhs9 = new Mahasiswa(nama: "Ivan", age: 22, NIM: "1900009", listDosen);
Mahasiswa mhs10 = new Mahasiswa(nama: "Joko", age: 21, NIM: "1900010", listDosen);

listMahasiswa.add(mhs1);
listMahasiswa.add(mhs2);
listMahasiswa.add(mhs3);
listMahasiswa.add(mhs4);
listMahasiswa.add(mhs5);
listMahasiswa.add(mhs6);
listMahasiswa.add(mhs7);
listMahasiswa.add(mhs8);
listMahasiswa.add(mhs9);
listMahasiswa.add(mhs10);
```

Selanjutnya, saya membuat 5 objek Mata Kuliah. Sesuai dengan class diagram yang telah dijelaskan sebelumnya, satu mata kuliah dapat diampu oleh lebih dari satu dosen. Oleh karena itu, pada saat pembuatan objek mata kuliah, terdapat parameter konstruktor yang digunakan untuk menerima List dosen yang mengajar mata kuliah tersebut.

Untuk memenuhi kebutuhan ini, saya membagi daftar dosen yang sudah dibuat menjadi beberapa List yang lebih kecil, di mana masing-masing List berisi dosen yang akan mengajar mata kuliah tertentu. Dengan cara ini, setiap mata kuliah akan memiliki daftar dosen yang sesuai.

```
List<Dosen> dosenPemrograman = new ArrayList<>();
dosenPemrograman.add(dsn1);
dosenPemrograman.add(dsn2);

List<Dosen> dosenStrukturData = new ArrayList<>();
dosenStrukturData.add(dsn3);
dosenStrukturData.add(dsn4);

List<Dosen> dosenBasisData = new ArrayList<>();
dosenBasisData.add(dsn5);

List<Dosen> dosenSistemOperasi = new ArrayList<>();
dosenSistemOperasi.add(dsn6);
dosenSistemOperasi.add(dsn7);

List<Dosen> dosenJaringanKomputer = new ArrayList<>();
dosenJaringanKomputer.add(dsn8);
dosenJaringanKomputer.add(dsn9);
dosenJaringanKomputer.add(dsn10);
```

```
// Membuat objek MataKuliah
MataKuliah mk1 = new MataKuliah(namaMataKuliah: "Pemrograman Java", (short) 3, jenisMataKuliah: "Wajib", dosenPemrograman);
MataKuliah mk2 = new MataKuliah(namaMataKuliah: "Struktur Data", (short) 3, jenisMataKuliah: "Wajib", dosenStrukturData);
MataKuliah mk3 = new MataKuliah(namaMataKuliah: "Basis Data", (short) 3, jenisMataKuliah: "Wajib", dosenBasisData);
MataKuliah mk4 = new MataKuliah(namaMataKuliah: "Sistem Operasi", (short) 3, jenisMataKuliah: "Wajib", dosenSistemOperasi);
MataKuliah mk5 = new MataKuliah(namaMataKuliah: "Jaringan Komputer", (short) 3, jenisMataKuliah: "Wajib", dosenJaringanKomputer);
```

Terakhir, saya akan membuat 6 objek Jadwal. Sesuai dengan class diagram yang telah dijelaskan sebelumnya, 1 jadwal dapat diikuti oleh lebih dari satu mahasiswa. Setiap jadwal hanya terkait dengan 1 mata kuliah, dan jadwal ini bergantung pada class MataKuliah untuk menentukan informasi mengenai mata kuliah yang diajarkan, seperti nama mata kuliah dan dosen yang mengajar.

Selain itu, setiap jadwal juga akan memiliki daftar mahasiswa yang mengikuti mata kuliah pada jadwal tersebut, yang dapat terdiri dari beberapa mahasiswa.

```
Jadwal jadwal1 = new Jadwal( hari: "Senin", mk1, LocalTime.of( hour: 8, minute: 0), LocalTime.of( hour: 10, minute: 0), ruangan: "Ruang  
jadwal1.setDaftarMahasiswa(new ArrayList<>(listMahasiswa.subList(0, 5))); // Mahasiswa 1-5  
  
Jadwal jadwal2 = new Jadwal( hari: "Senin", mk2, LocalTime.of( hour: 10, minute: 0), LocalTime.of( hour: 12, minute: 0), ruangan: "Ruar  
jadwal2.setDaftarMahasiswa(new ArrayList<>(listMahasiswa.subList(5, 10))); // Mahasiswa 6-10  
  
Jadwal jadwal3 = new Jadwal( hari: "Selasa", mk3, LocalTime.of( hour: 8, minute: 0), LocalTime.of( hour: 10, minute: 0), ruangan: "Ruar  
jadwal3.setDaftarMahasiswa(new ArrayList<>(listMahasiswa.subList(0, 3))); // Mahasiswa 1-3  
  
Jadwal jadwal4 = new Jadwal( hari: "Selasa", mk4, LocalTime.of( hour: 10, minute: 0), LocalTime.of( hour: 12, minute: 0), ruangan: "Ruar  
jadwal4.setDaftarMahasiswa(new ArrayList<>(listMahasiswa.subList(3, 7))); // Mahasiswa 4-7  
  
Jadwal jadwal5 = new Jadwal( hari: "Rabu", mk5, LocalTime.of( hour: 8, minute: 0), LocalTime.of( hour: 10, minute: 0), ruangan: "Ruang  
jadwal5.setDaftarMahasiswa(new ArrayList<>(listMahasiswa.subList(7, 10))); // Mahasiswa 8-10  
  
Jadwal jadwal6 = new Jadwal( hari: "Kamis", mk1, LocalTime.of( hour: 10, minute: 0), LocalTime.of( hour: 12, minute: 0), ruangan: "Ruar  
jadwal6.setDaftarMahasiswa(new ArrayList<>(listMahasiswa)); // Semua Mahasiswa
```

Masalah:-

Solusi:-

Teman yang membantu: -

Link github source code week-1 [Tugas-PBO/Week-5 at main · RaditZX/Tugas-PBO \(github.com\)](https://github.com/RaditZX/Tugas-PBO/Week-5)