

**TUGAS PRAKTIKUM**  
**PRAKTIKUM 9:**  
**“Persistent Object”**



**Disusun Oleh :**

Mahardika Putra Wardhana  
24060121130076

**PRAKTIKUM PBO**  
**LAB B2**

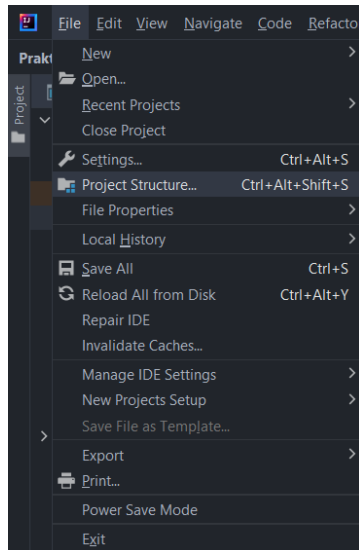
**DEPARTEMEN ILMU KOMPUTER / INFORMATIKA**  
**FAKULTAS SAINS DAN MATEMATIKA**  
**UNIVERSITAS DIPONEGORO**  
**SEMARANG**  
**2023**

## 1. Menggunakan Persistent Object sebagai model basis data relasional

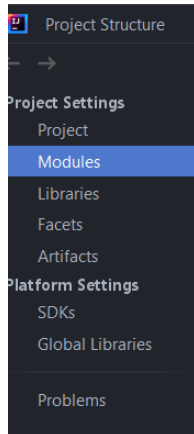
Pada praktikum ini, anda akan membuat implementasi kelas sebagaimana ditunjukkan diagram kelas berikut :

Sebelum memulai, IDE yang dipakai harus ditambahkan modul SQL Connector terlebih dahulu dengan sehingga *class-class* di java dapat terhubung dengan basis data. Pada kasus ini, IDE yang saya gunakan adalah IntelliJ IDE. Untuk menghubungkan IntelliJ dengan basis data, langkah-langkahnya adalah sebagai berikut:

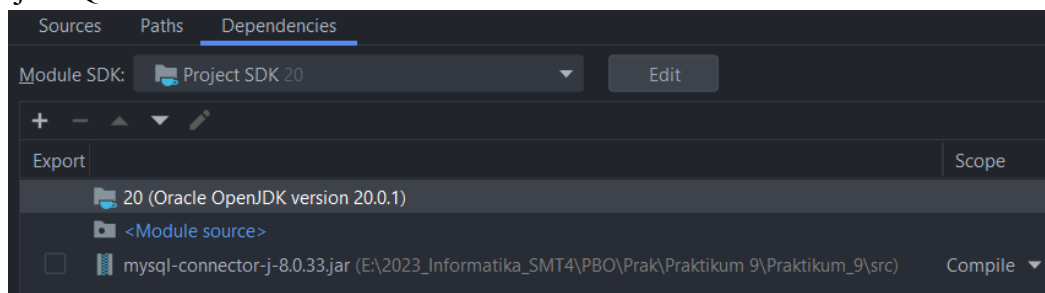
### a. Buka File -> Project Structure



### b. Lalu pilih opsi Modules pada Project Settings



### c. Pilih tab Dependencies dan kemudian klik tanda plus untuk menambahkan file .jar SQL Connector.



Untuk kode-kode *class* adalah sebagai berikut:

a. PersonDAO.java

*Interface untuk person access object*

```
/*
    PersonDAO.java  06/06/2023
    Nama          : Mahardika Putra Wardhana
    NIM           : 24060121130076
*/

public interface PersonDAO {
    public void savePerson(Person p) throws Exception;
}
```

b. Person.java

*Class ini berisi konstruktor-konstruktor person dan berisi method getter yaitu getId() dan getName().*

```
/*
    Person.java  06/06/2023
    Nama          : Mahardika Putra Wardhana
    NIM           : 24060121130076
*/

public class Person {
    private int id;
    private String name;

    public Person(String n){
        name = n;
    }

    public Person(int i, String n){
        id = i;
        name = n;
    }

    public int getId(){
        return id;
    }

    public String getName(){
        return name;
    }
}
```

c. MySQLPersonDAO.java

Di *class* ini, kita dapat mengatur koneksi antara java dan basis data. Kita dapat mengatur koneksi ke basis data pada line `Connection con = DriverManager.getConnection("jdbc:mysql://localhost/pbo", "root", "123456");` dimana kita dapat memilih schema dan memasukkan username dan password untuk basis data. Di *class* ini juga, kita

dapat melakukan pembuatan *query* dan mengeksekusinya untuk memodifikasi tabel pada basis data.

```
/*
    MySQLPersonDAO.java 06/06/2023
    Nama      : Mahardika Putra Wardhana
    NIM       : 24060121130076
*/

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;

public class MySQLPersonDAO implements PersonDAO{
    public void savePerson(Person person) throws Exception{
        String name = person.getName();

        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/pbo","root","123456");

        String query = "INSERT INTO person(name)
VALUES ('"+name+"')";
        System.out.println(query);
        Statement s = con.createStatement();
        s.executeUpdate(query);

        con.close();
    }
}
```

d. DAOManager.java  
Pengelola DAO dalam program

```
/*
    DAOManager.java 06/06/2023
    Nama      : Mahardika Putra Wardhana
    NIM       : 24060121130076
*/

public class DAOManager{
    private PersonDAO personDAO;

    public void setPersonDAO(PersonDAO person){
        personDAO = person;
    }

    public PersonDAO getPersonDAO() {
        return personDAO;
    }
}
```

e. MainDAO.java

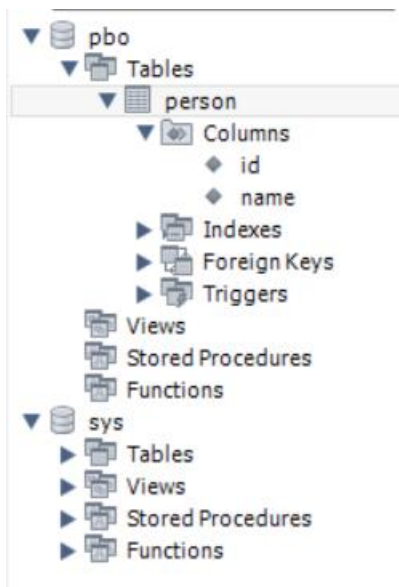
Main program untuk akses DAO dimana semua *class* diinisialisasi

```
/*
    MainDAO.java  06/06/2023
    Nama          : Mahardika Putra Wardhana
    NIM           : 24060121130076
*/

public class MainDAO {
    public static void main(String[] args){
        Person person = new Person("Rose");
        DAOManager m = new DAOManager();
        m.setPersonDAO(new MySQLPersonDAO());
        try{
            m.getPersonDAO().savePerson(person);
        }catch(Exception e){
            e.printStackTrace();
        }
    }
}
```

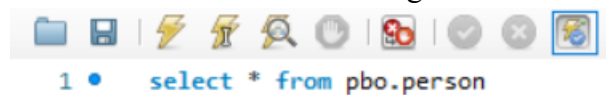
Kemudian, masuk ke MySQL dan buat *schema* bernama 'pbo' dan buat tabel dengan *query*

```
CREATE TABLE person(id INT PRIMARY KEY AUTO_INCREMENT NOT
NULL,name VARCHAR(100))
```



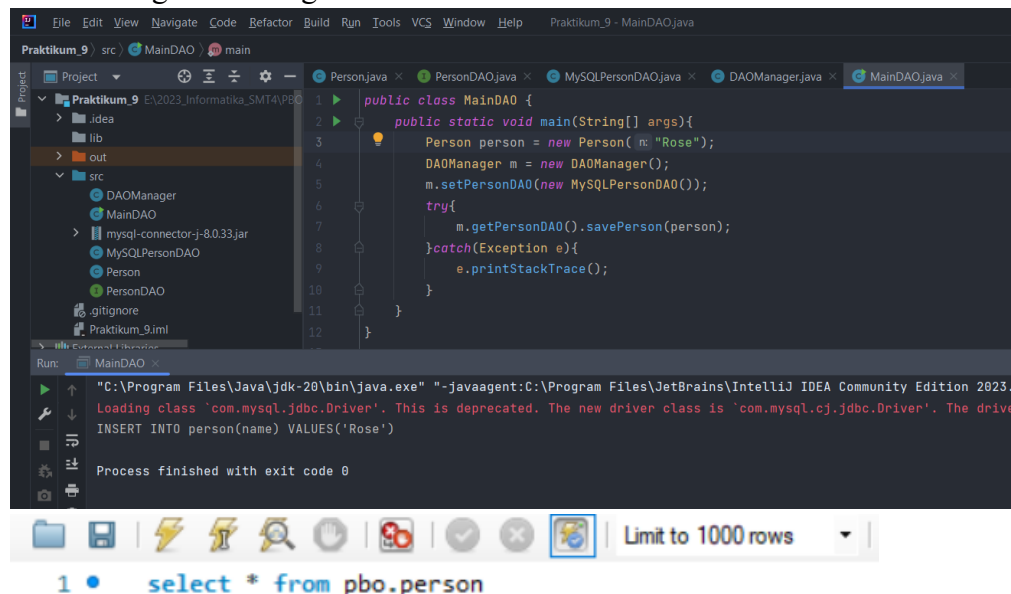
Berikut adalah perbedaan tabel sebelum dan setelah program java dijalankan:

- a. Kondisi awal tabel adalah sebagai berikut:



| Result Grid |       | Filter Rows: |
|-------------|-------|--------------|
| id          | name  |              |
| 1           | Indra |              |
| NULL        | NULL  |              |

- b. Setelah melakukan *running* MainDAO.java, maka isi tabel person telah diubah dengan isi sebagai berikut:



| Result Grid |       | Filter Rows: | Edit: | Export |
|-------------|-------|--------------|-------|--------|
| id          | name  |              |       |        |
| 1           | Indra |              |       |        |
| 2           | Rose  |              |       |        |
| NULL        | NULL  |              |       |        |

## 2. Menggunakan Persistent Object sebagai objek terserialisasi

- a. SerializePerson.java

```
import java.io.*;
```

```

/*
    SerializePerson.java  06/06/2023
    Nama      : Mahardika Putra Wardhana
    NIM       : 24060121130076
    Deskripsi  : Program untuk serialisasi objek person
*/

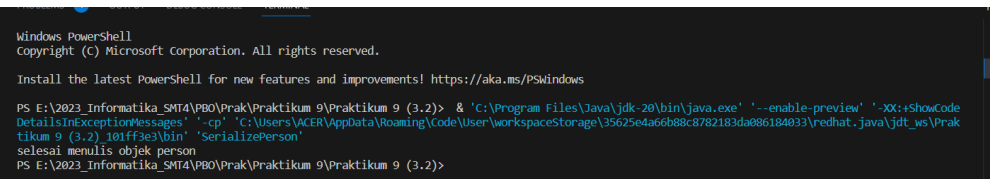
class Person implements Serializable{
    private String name;
    public Person(String n){
        name = n;
    }

    public String getName(){
        return name;
    }
}

public class SerializePerson {
    public static void main(String[] args){
        Person person = new Person("Panji");
        try{
            FileOutputStream f = new
FileOutputStream("person.ser");
            ObjectOutputStream s = new ObjectOutputStream(f);
            s.writeObject(person);
            System.out.println("selesai menulis objek
person");
        }
        catch(IOException e){
            e.printStackTrace();
        }
    }
}

```

Hasil *run*:



```

Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS E:\2023 Informatika_SMT4\PRO\Prak\Praktikum 9\Praktikum 9 (3.2)> & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\VACER\AppData\Roaming\Code\User\workspaceStorage\35625e4a66b88c8782183da886184033\redhat_java\jdt_ws\Praktikum 9 (3.2)\01ff3e3\bin' 'SerializePerson'
selesai menulis objek person
PS E:\2023 Informatika_SMT4\PRO\Prak\Praktikum 9\Praktikum 9 (3.2)>

```

selesai menulis objek person

## b. ReadSerializedPerson.java

```

import java.io.*;

/*
    ReadSerializedPerson.java  06/06/2023
    Nama      : Mahardika Putra Wardhana
    NIM       : 24060121130076

```

```
Deskripsi : Program untuk serialisasi objek person
*/

public class ReadSerializedPerson {
    public static void main(String[] args){
        Person person = null;
        try{
            FileInputStream f = new
FileInputStream("person.ser");
            ObjectInputStream s = new ObjectInputStream(f);
            person = (Person)s.readObject();
            s.close();
            System.out.println("serialized person name =
"+person.getName());
        }catch(Exception ioe){
            ioe.printStackTrace();
        }
    }
}
```

Hasil *run*:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS E:\2023 Informatika SMT4\Prak\Praktikum 9\Praktikum 9 (3.2)> & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ACER\AppData\Roaming\Code\User\workspaceStorage\35625e4a66b88c8782183da086184033\redhat.java\jdt_ws\Praktikum 9 (3.2)_101ff3e3\bin' 'ReadSerializedPerson'
serialized person name = Panji
PS E:\2023 Informatika SMT4\Prak\Praktikum 9\Praktikum 9 (3.2)>
```

serialized person name = Panji