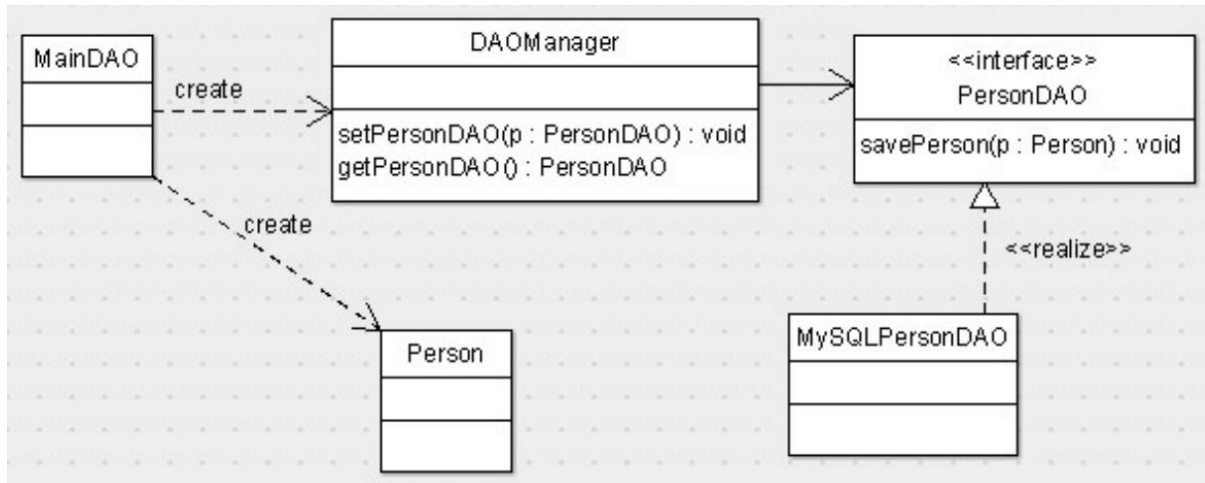


Nama: Rizki Utama Fauzi  
NIM: 24060121130050  
Prak PBO C2

## Pertemuan 9

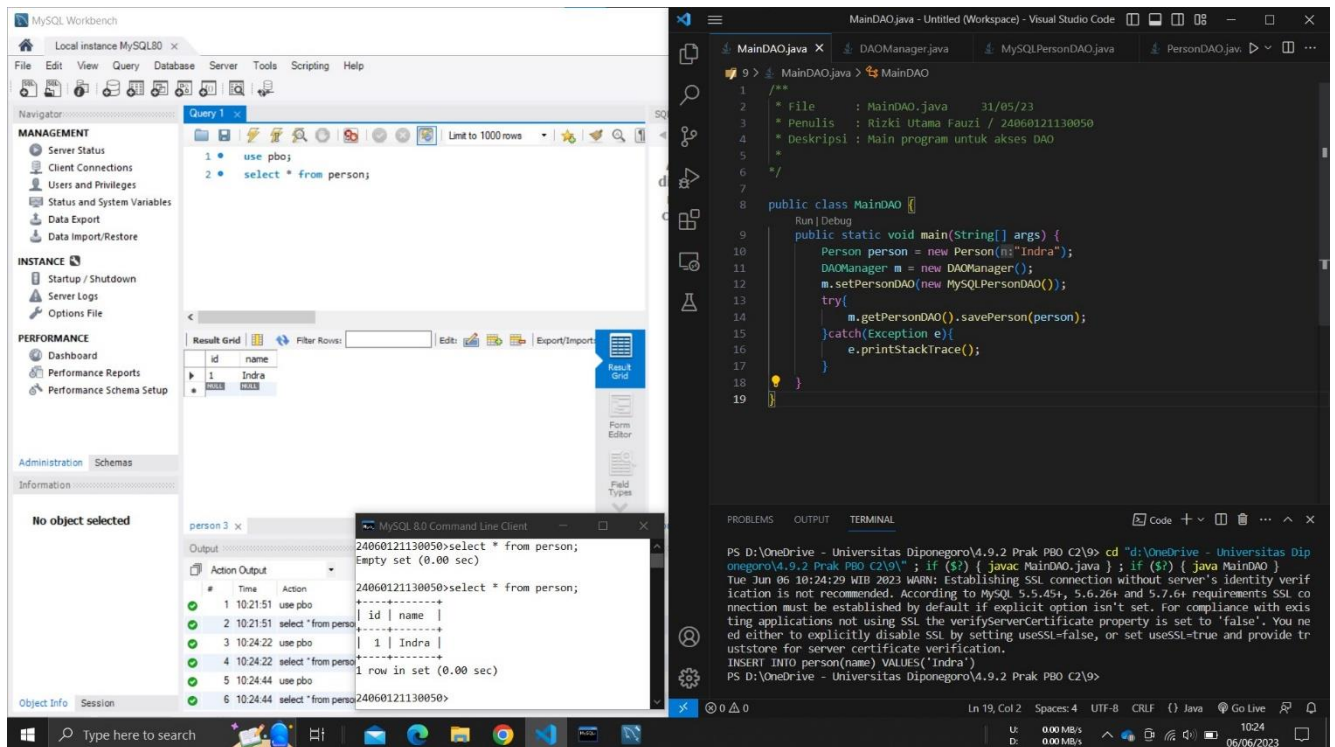
### 1. Menggunakan Persistent Object sebagai model basis data relasional

Class Diagram:



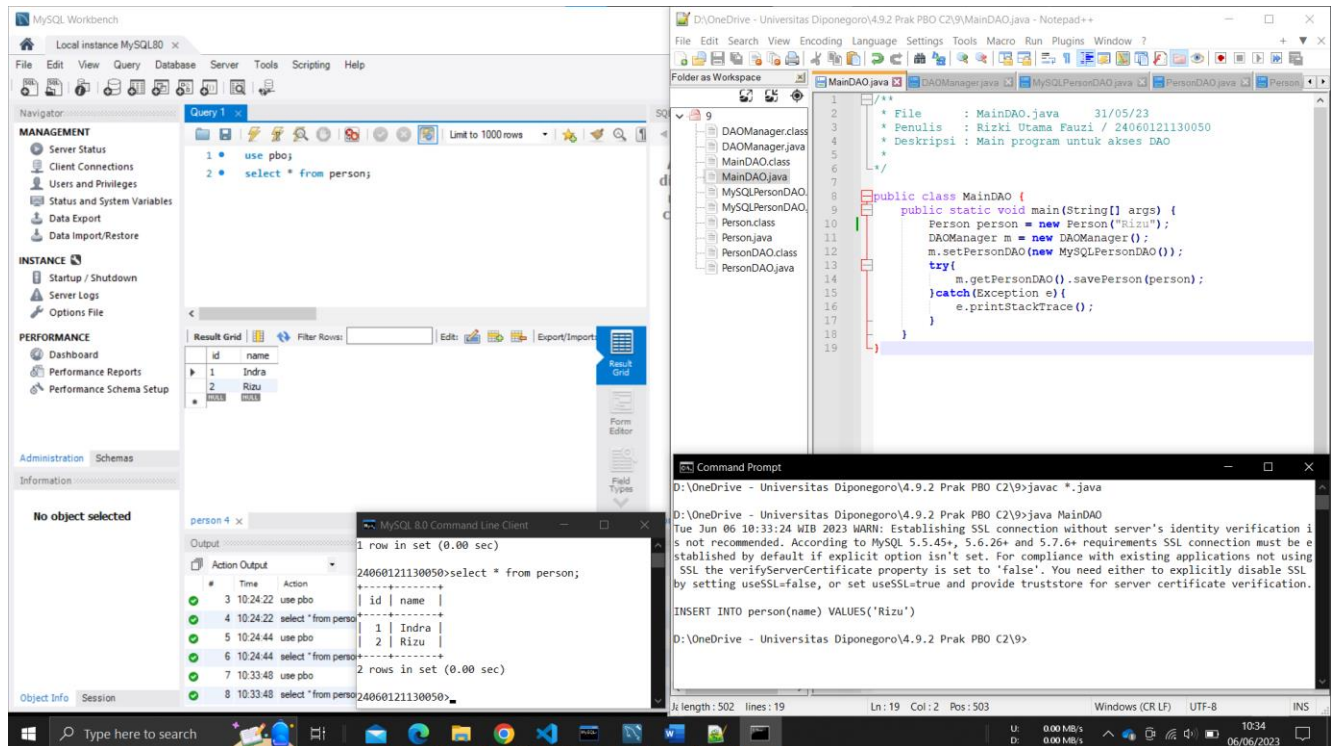
Screenshot program MainDAO dan hasil running (terminal dan tabel di MySQL)

- Menggunakan VS Code



- Menggunakan Notepad++

Nama: Rizki Utama Fauzi  
NIM: 24060121130050  
Prak PBO C2



Untuk source code kelas-kelas atau file java yang digunakan adalah sebagai berikut.

### MainDAO.java

```
/**
 * File : MainDAO.java 31/05/23
 * Penulis : Rizki Utama Fauzi / 24060121130050
 * Deskripsi : Main program untuk akses DAO
 */

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

### DAOManager.java

```
/**
 * File : DAOManager.java 31/05/23
 * Penulis : Rizki Utama Fauzi / 24060121130050
 * Deskripsi : Pengelola DAO dalam program
 */

public class DAOManager {
    private PersonDAO personDAO;

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

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

### MySQLPersonDAO.java

Nama: Rizki Utama Fauzi  
NIM: 24060121130050  
Prak PBO C2

```
/**
 * File      : MySQLPersonDAO.java      31/05/23
 * Penulis   : Rizki Utama Fauzi / 24060121130050
 * Deskripsi  : Implementasi PersonDAO untuk MySQL
 */

import java.sql.*;

public class MySQLPersonDAO implements PersonDAO {
    public void savePerson(Person person) throws Exception{
        String name = person.getName();
        //membuat koneksi, nama db,user,password menyesuaikan
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3307/pbo", "root", "Rizu2002");
        //kerjakan mysql query
        String query = "INSERT INTO person(name) VALUES('"+name+"')";
        System.out.println(query);
        Statement s = con.createStatement();
        s.executeUpdate(query);
        //tutup koneksi database
        con.close();
    }
}
```

### *PersonDAO.java*

```
/**
 * File      : PersonDAO.java      31/05/23
 * Penulis   : Rizki Utama Fauzi / 24060121130050
 * Deskripsi  : interface untuk person access object
 */

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

### *Person.java*

```
/**
 * File      : Person.java      31/05/23
 * Penulis   : Rizki Utama Fauzi / 24060121130050
 * Deskripsi  : Person database model
 */

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;
    }
}
```

## 2. Menggunakan Persistent Object sebagai objek terserialisasi

- Screenshot mengcompile dan menjalankan program SerializePerson.java

Nama: Rizki Utama Fauzi  
NIM: 24060121130050  
Prak PBO C2

The screenshot shows a Notepad++ window with the source code of `ReadSerializedPerson.java`. The code imports `java.io.FileInputStream` and `java.io.ObjectInputStream`, and defines a `main` method that reads a serialized object from a file named `person.ser`. Below the code editor, a Command Prompt window shows the execution of the program. The output indicates that the program successfully read the serialized object and printed the name "Rizki".

```
1  /**
2   * File      : ReadSerializedPerson.java      31/05/23
3   * Penulis   : Rizki Utama Fauzi / 24060121130050
4   * Deskripsi  : Program untuk serialisasi objek Person
5   */
6
7
8  import java.io.FileInputStream;
9  import java.io.ObjectInputStream;
10
11 public class ReadSerializedPerson {
12     public static void main(String[] args) {
13         Person person = null;
14         try {
15             FileInputStream f = new FileInputStream("person.ser");
16             ObjectInputStream s = new ObjectInputStream(f);
17             person = (Person)s.readObject();
18             s.close();
19             System.out.println("serialized person name = "+person.getName());
20         } catch (Exception ioe) {
21             ioe.printStackTrace();
22         }
23     }
24 }
25
```

```
D:\OneDrive - Universitas Diponegoro\4.9.2 Prak PBO C2\9b>cd ../9b
D:\OneDrive - Universitas Diponegoro\4.9.2 Prak PBO C2\9b>javac SerializePerson.java
D:\OneDrive - Universitas Diponegoro\4.9.2 Prak PBO C2\9b>java SerializePerson
selesai menulis objek person
D:\OneDrive - Universitas Diponegoro\4.9.2 Prak PBO C2\9b>
```

- Screenshot mengcompile dan menjalankan program `ReadSerializedPerson.java`

The screenshot shows a Notepad++ window with the source code of `ReadSerializedPerson.java`. The code imports `java.io.FileInputStream` and `java.io.ObjectInputStream`, and defines a `main` method that reads a serialized object from a file named `person.ser`. Below the code editor, a Command Prompt window shows the execution of the program. The output indicates that the program successfully read the serialized object and printed the name "Rizki".

```
1  /**
2   * File      : ReadSerializedPerson.java      31/05/23
3   * Penulis   : Rizki Utama Fauzi / 24060121130050
4   * Deskripsi  : Program untuk serialisasi objek Person
5   */
6
7
8  import java.io.FileInputStream;
9  import java.io.ObjectInputStream;
10
11 public class ReadSerializedPerson {
12     public static void main(String[] args) {
13         Person person = null;
14         try {
15             FileInputStream f = new FileInputStream("person.ser");
16             ObjectInputStream s = new ObjectInputStream(f);
17             person = (Person)s.readObject();
18             s.close();
19             System.out.println("serialized person name = "+person.getName());
20         } catch (Exception ioe) {
21             ioe.printStackTrace();
22         }
23     }
24 }
25
```

```
D:\OneDrive - Universitas Diponegoro\4.9.2 Prak PBO C2\9b>javac ReadSerializedPerson.java
D:\OneDrive - Universitas Diponegoro\4.9.2 Prak PBO C2\9b>java ReadSerializedPerson
serialized person name = Rizki
D:\OneDrive - Universitas Diponegoro\4.9.2 Prak PBO C2\9b>
```