LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK

"Persistence Object"

DISUSUN OLEH:

RESMA ADI NUGROHO

24060121120021

DEPARTEMEN INFORMATIKA

FAKULTAS SAINS DAN MATEMATIKA

UNIVERSITAS DIPONEGORO

2023

A. Persistence Object pada Basis Data Relasional

1. File Person.java

```
/*
* Nama File    : Person.java
* Nama Pembuat : Resma Adi Nugroho
* NIM          : 24060121120021
* Deskripsi    : File Kelas Person (Model)
* Tanggal      : 03 Juni 2023
*/

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;
    }
}
```

File kelas ini berguna sebagai model objek yang akan dibuat atau digunakan pada penyimpanan data melalui MySQL.

2. File DAOManager.java

```
/*
* Nama File    : DAOManager.java
* Nama Pembuat : Resma Adi Nugroho
* NIM          : 24060121120021
* Deskripsi    : File kelas DAOManager
* Tanggal      : 03 Juni 2023
*/

public class DAOManager {
    private PersonDAO personDAO;

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

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

3. File PersonDAO.java

```
/*
* Nama File    : PersonDAO.java
* Nama Pembuat : Resma Adi Nugroho
* NIM          : 24060121120021
* Deskripsi    : Interface untuk person access object
* Tanggal      : 03 Juni 2023
*/

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

4. File MySQLPersonDAO.java

```
/*
* Nama File    : MySQLPersonDAO.java
* Nama Pembuat : Resma Adi Nugroho
* NIM          : 24060121120021
* Deskripsi    : File untuk config untuk mengkoneksi ke database
MySQL
* Tanggal      : 03 Juni 2023
*/

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
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/pbo", "root",
"");
        // kerjakan mysql query
        String query = "INSERT INTO person(name) VALUES ('" +
name + "')";
        System.out.println(query);
        Statement st = con.createStatement();
        st.executeUpdate(query);
        // tutup koneksi
        con.close();
    }
}
```

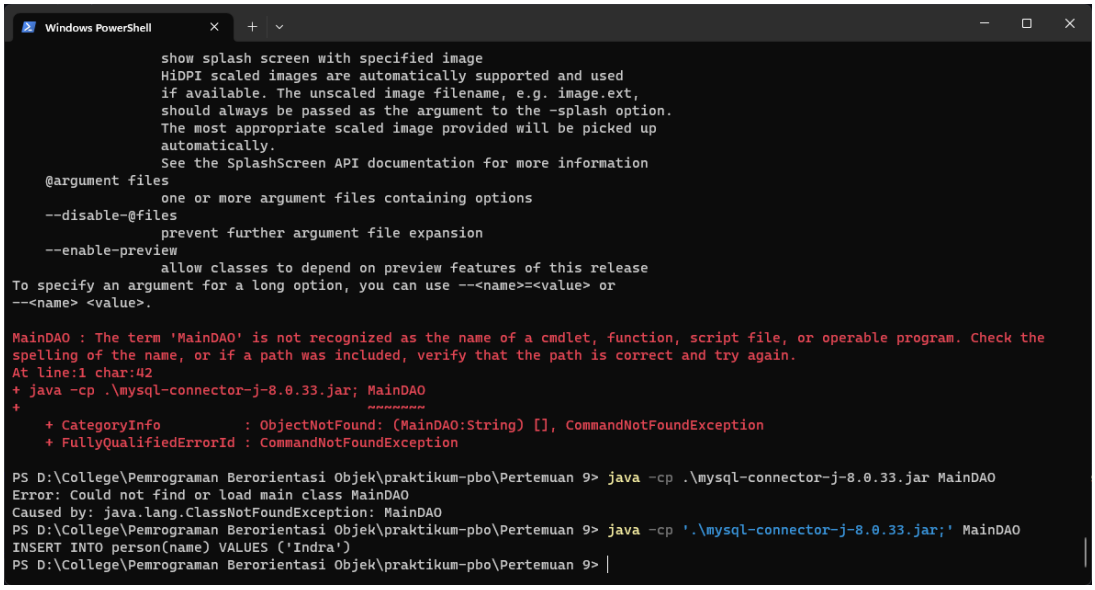
5. File MainDAO.java

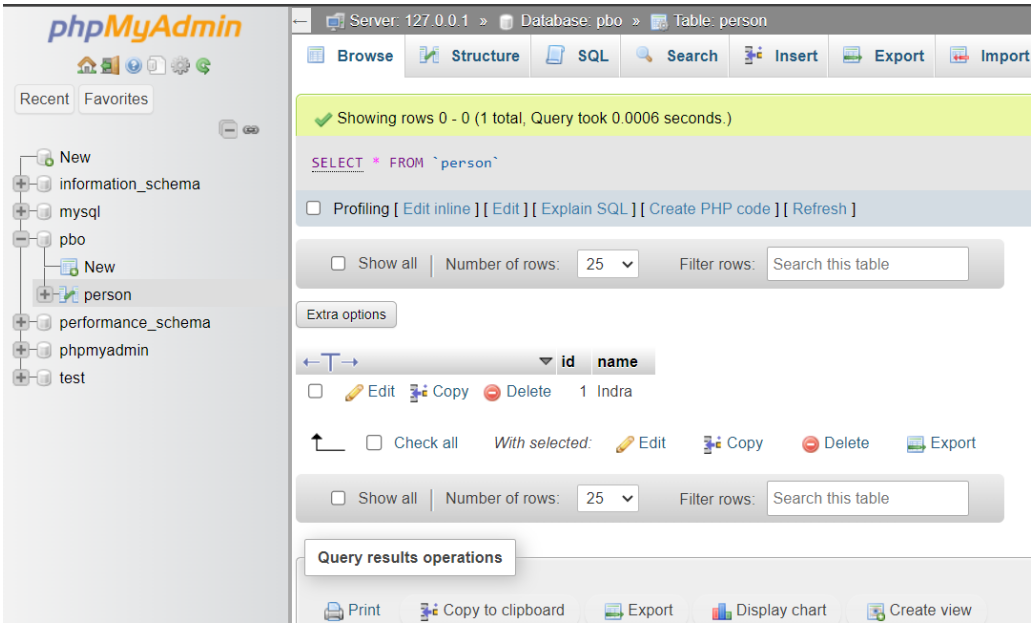
```
/*
* Nama File    : MainDAO.java
* Nama Pembuat : Resma Adi Nugroho
* NIM          : 24060121120021
* Deskripsi    : File Main untuk menjalankan program
* Tanggal      : 03 Juni 2023
*/

public class MainDAO {
    public static void main(String[] args) {
```

```
            Person person = new Person("Indra");
            DAOManager m = new DAOManager();
            m.setPersonDAO(new MySQLPersonDAO());
            try {
                m.getPersonDAO().savePerson(person);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
```

Hasil run program dan bukti bahwa data ‘Indra’ telah berhasil dimasukan ke dalam database MySQL





B. Persistence Object pada Serialisasi

1. File SerializePerson.java

```
/*
* Nama File    : SerializePerson.java
* Nama Pembuat : Resma Adi Nugroho
* NIM          : 24060121120021
* Deskripsi    : Main file untuk menyimpan data melalui
serialisasi
* Tanggal      : 03 Juni 2023
*/
import java.io.*;

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 p = new Person("Panji");
        try {
            FileOutputStream fos = new
FileOutputStream("person.ser");
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(p);
            System.out.println("Selesai menulis objek person");
            oos.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

2. File ReadSerializedPerson.java

```
/*
* Nama File    : ReadSerializedPerson.java
* Nama Pembuat : Resma Adi Nugroho
* NIM          : 24060121120021
* Deskripsi    : Main file untuk membaca data dari serialisasi
* Tanggal      : 03 Juni 2023
*/

import java.io.*;
public class ReadSerializedPerson {
    public static void main(String[] args) {
        Person p = null;
        try {
            FileInputStream fis = new
FileInputStream("person.ser");
            ObjectInputStream ois = new ObjectInputStream(fis);
            p = (Person) ois.readObject();
            ois.close();
```

```
            System.out.println("Serialized person name =
"+p.getName());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

File SerializePerson digunakan untuk menuliskan atau menyimpan data ke dalam file serialisasi yakni dalam file person.ser kemudian untuk membaca file serialisasi tersebut diperlukan file ReadSerializedPerson agar data yang ada dalam file serialisasi dapat dibaca dan ditampilkan.

Hasil run program dapat dilihat di bawah