LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

PRESISTENT OBJEK

Disusun Oleh :

Abida Akbar Rusyadin

24060119120041

LAB B

DEPARTEMEN ILMU KOMPUTER / INFORMATIKA

FAKULTAS SAINS DAN MATEMATIKA

UNIVERSITAS DIPONEGORO

SEMARANG

2023

A. Source Code

1. PersonDAO.java

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

2. Person.java

```
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;
    }
}
```

3. MySQLPersonDAO.java

```
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/pbo","root","");
        //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();
    }
}
```

4. DAOManager.java

```
public class DAOManager {
    private PersonDAO personDAO;

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

5. MainDAO.java

```
public class MainDAO {
    public static void main(String[]
args){
        Person person = new
Person("Indra");
        DAOManager m = new
DAOManager();
        m.setPersonDAP(new
MySQLPersonDAO());
        try{

m.getPersonDAO().savePerson(person);
        }catch(Exception e){
```

```
            e.printStackTrace();
        }
    }
}
```

6. SerializePerson.java

```
import java.io.*;

//class Person
class Person implements Serializable {
    private String name;

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

    public String getName(){
        return name;
    }
}

//class SerializePerson
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");
            s.close();
        }catch(IOException e){
            e.printStackTrace();
        }
    }
}
```

7. ReadSerializedPerson.java

```
import java.io.*;

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();
        }
    }
}
```