LAPORAN PRAKTIKUM 9
PEMROGRAMAN BERBASIS OBJEK
PERSISTENT OBJECT

Disusun oleh :

Nama : Raihan Gilang Firdausy

NIM : 24060121130065

Lab : B2

PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS DAN MATEMATIKA

UNIVERSITAS DIPONEGORO

SEMARANG

2023

A. Menggunakan Persistent Object Sebagai Model Basis Data Relasional

1. PersonDAO.java

```
/**
* File : PersonDAO.java 31/05/2023
* Nama : Raihan Gilang Firdausy / 24060121130065
* Deskripsi : interface untuk person access object
*/

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

2. Person.java

```
/**
* File : Person.java 31/05/2023
* Nama : Raihan Gilang Firdausy / 24060121130065
* 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;
 }
}
```

3. MySQLPersonDAO.java

```
/**
* File : MySQLPersonDAO.java 31/05/2023
* Nama : Raihan Gilang Firdausy / 24060121130065
* 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/p
bo","root","erge1221");
          //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

```
/**
* File : DAOManager.java 31/05/2023
* Nama : Raihan Gilang Firdausy / 24060121130065
* Deskripsi : pengelola DAO dalam program
*/

public class DAOManager{
private PersonDAO personDAO;

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

5. mainDAO.java

```
/**
* File : mainDAO.java 31/05/2023
* Nama : Raihan Gilang Firdausy / 24060121130065
* Deskripsi : Main program untuk akses DAO
*/
```

```
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();
      }
 }
 }
```

6. Buat database dengan nama ‘pbo’ dan tabel pada database tersebut dengan : CREATE TABLE person(id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,name VARCHAR(100))

```
mysql> prompt Gilang_24060121130065>
PROMPT set to 'Gilang_24060121130065> '
Gilang_24060121130065> create database pbo
    -> ;
Query OK, 1 row affected (0.65 sec)

Gilang_24060121130065> use pbo;
Database changed
Gilang_24060121130065> show tables;
Empty set (0.05 sec)

Gilang_24060121130065> CREATE TABLE person(
    -> id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
    -> name VARCHAR(100));
Query OK, 0 rows affected (7.04 sec)

Gilang_24060121130065> select * from person;
Empty set (0.04 sec)
```

7. Kompilasi semua source code dengan perintah: javac *.java

```
C:\Users\R O G\Pictures\PBO PRAK\PERT9> javac *.java
```

8. Jalankan MainDAO dengan perintah: java -classpath .\mysql-connector-j-8.0.33.jar;. MainDAO

```
C:\Users\R O G\Pictures\PBO PRAK\PERT9>java -classpath .\mysql-connector-j-8.0.33.jar;. MainDAO
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The drive
r is automatically registered via the SPI and manual loading of the driver class is generally unnecessary.
INSERT INTO person(name) VALUES('Indra')
```

9. Lihat apakah terjadi penambahan record pada tabel !

```
Gilang_24060121130065> select * from person;
+----+-------+
| id | name  |
+----+-------+
|  1 | Indra |
+----+-------+
1 row in set (0.00 sec)
```

B. Menggunakan Persistent Object sebagai Objek Terserialisasi

1. SerializePerson.java

```
/**
* File : SerializePerson.java 31/05/2023
* Nama : Raihan Gilang Firdausy / 24060121130065
* Deskripsi : Program untuk serialisasi objek Person
*/

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 person = new Person("Gilang");
        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();
        }
    }
}
```

2. Compile, dan jalankan program di atas dengan

- javac SerializePerson.java
- java SerializePerson

```
C:\Users\R O G\Pictures\PBO PRAK\PERT9\Serializable>javac SerializePerson.java

C:\Users\R O G\Pictures\PBO PRAK\PERT9\Serializable>java SerializePerson
Selesai menulis objek person
```

3. ReadSerializedPerson.java

```
/**
* File : ReadSerializedPerson.java 31/05/2023
* Nama : Raihan Gilang Firdausy / 24060121130065
* Deskripsi : Program untuk serialisasi objek Person
*/

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();
        }
    }
}
```

4. Compile, dan jalankan program di atas dengan

   - javac ReadSerializedPerson.java
   - java ReadSerializedPerson

```
C:\Users\R O G\Pictures\PBO PRAK\PERT9\Serializable>javac ReadSerializedPerson.java

C:\Users\R O G\Pictures\PBO PRAK\PERT9\Serializable>java ReadSerializedPerson
Serialized person name = Gilang
```