LAPORAN PRAKTIKUM
PRAKTIKUM 9:
“PERSISTENT OBJECT”

Disusun Oleh :

Fitra Syamli Yudhisaputra
24060121140124

PRAKTIKUM PEMROGRAMAN
BERORIENTASI OBJEK
LAB C1

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

A. Menggunakan Persistensi Object sebagai Model Basis Data Relasional

1. PersonDAO.java

```
/**
 * Penulis : Fitra Syamli Yudhisaputra - 24060121140124
 * Tanggal : 31/05/2023
 * File : PersonDAO.java
 * Deskripsi : interface untuk person access object
**/

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

2. Person.java

```
/**
 * Penulis : Fitra Syamli Yudhisaputra - 24060121140124
 * Tanggal : 31/05/2023
 * File : Person.java
 * 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

```
/**
 * Penulis : Fitra Syamli Yudhisaputra - 24060121140124
 * Tanggal : 31/05/2023
 * File : MySQLPersonDAO.java
 * 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/pbo","root","dawang123");
    //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

```java
/**
 * Penulis : Fitra Syamli Yudhisaputra - 24060121140124
 * Tanggal : 31/05/2023
 * File : DAOManager.java
 * 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

```java
/**
 * Penulis : Fitra Syamli Yudhisaputra - 24060121140124
 * Tanggal : 31/05/2023
 * File : MainDAO.java
 * 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. Membuat database dengan nama ‘pbo’ dan tabel dengan :

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

```
mysql> prompt 24060121140124>
PROMPT set to '24060121140124>'
24060121140124> create database pbo;
Query OK, 1 row affected (0.01 sec)

24060121140124>use pbo;
Database changed
24060121140124>show tables;
Empty set (0.09 sec)

24060121140124>CREATE TABLE person(
    -> id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
    -> name VARCHAR(100));
Query OK, 0 rows affected (0.05 sec)

24060121140124> select * from person;
```

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

```
C:\Users\ASUS\OneDrive\Documents\Semester 4\PBO\Praktikum PBO\Pertemuan 9>javac *.java

C:\Users\ASUS\OneDrive\Documents\Semester 4\PBO\Praktikum PBO\Pertemuan 9>
```

8. Jalankan MainDAO dengan perintah : **`java -classpath .\mysql-connector-java-[versi].jar;. MainDAO`**

```
C:\Users\ASUS\OneDrive\Documents\Semester 4\PBO\Praktikum PBO\Pertemuan 9> j
ava -classpath .\mysql-connector-j-8.0.33.jar;. MainDAO
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver cl
ass is `com.mysql.cj.jdbc.Driver'. The driver is automatically registered vi
a the SPI and manual loading of the driver class is generally unnecessary.
INSERT INTO person(name) VALUES ('Indra')
```

```
24060121140124> select * from person;
+----+-------+
| id | name  |
+----+-------+
|  1 | Indra |
+----+-------+
1 row in set (0.00 sec)
```

B. Menggunakan Persistent Object sebagai objek terserialisasi

1. SerializePerson.java

```java
/**
 * * Penulis : Fitra Syamli Yudhisaputra - 24060121140124
 * Tanggal : 31/05/2023
 * File : SerializePerson.java
 * Deskripsi : Program untuk serialisasi objek persoN
**/

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 onjek person");
            s.close();
        }catch(IOException e){
            e.printStackTrace();
        }
    }
}
```

```
C:\Users\ASUS\OneDrive\Documents\Semester 4\PBO\Praktikum PBO\Pertemuan 9\person.ser>javac SerializePerson.java

C:\Users\ASUS\OneDrive\Documents\Semester 4\PBO\Praktikum PBO\Pertemuan 9\person.ser>java SerializePerson
selesai menulis onjek person
```

2. ReadSerializedPerson.java

```java
/**
 * Penulis : Fitra Syamli Yudhisaputra - 24060121140124
 * Tanggal : 31/05/2023
 * File : ReadSerializedPerson.java
 * 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();
        }
    }
}
```

```
C:\Users\ASUS\OneDrive\Documents\Semester 4\PBO\Praktikum PBO\Pertemuan 9\person.ser> javac ReadSerializedPerson.java

C:\Users\ASUS\OneDrive\Documents\Semester 4\PBO\Praktikum PBO\Pertemuan 9\person.ser> java ReadSerializedPerson
serialized person name = Panji
```