

LAPORAN PRAKTIKUM
PRAKTIKUM 9:
“PERSISTENT OBJECT”



Disusun Oleh:

Mochammad Arya Jadmika
24060121130085

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
LAB B2

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

A. Menggunakan Persistent Object sebagai Model Basis Data Relasional

1. PersonDAO.java

```
/**
 * Nama file      :PersonDAO.java
 * Tanggal       :31 Mei 2023
 * Penulis       :Mochammad Arya Jadmika / 24060121130085
 * Deskripsi     :Interface untuk person access object
 */

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

2. Person.java

```
/**
 * Nama file      :Person.java
 * Tanggal       :31 Mei 2023
 * Penulis       :Mochammad Arya Jadmika / 24060121130085
 * 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

```
import java.sql.*;
/**
 * Nama File      :MySQLPersonDAO.java
 * Tanggal       :31 Mei 2023
 * Penulis       :Mochammad Arya Jadmika / 24060121130085
 * Deskripsi     : implementasi PersonDAO untuk MySQL
 */
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",
"Pastisukses45*");
        //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

```

/**
Nama file    :DAOManager.java
Tanggal      :31 Mei 2023
Penulis      :Mochammad Arya Jadmika / 24060121130085
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

```

/**
Nama file    :MainDAO.java
Tanggal      :31 Mei 2023
Penulis      :Mochammad Arya Jadmika / 24060121130085
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 pada database

```
mysql> prompt 24060121130085>
PROMPT set to '24060121130085>'
24060121130085> create database pbo;
Query OK, 1 row affected (0.06 sec)

24060121130085> use pbo;
Database changed
24060121130085> CREATE TABLE person(
  -> id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
  -> name VARCHAR(100));
Query OK, 0 rows affected (0.12 sec)

24060121130085> SELECT * FROM PERSON;
Empty set (0.01 sec)
```

Setelah membuat kelas-kelas sebelumnya seperti PersonDAO.java, Person.java, MySQLPersonDAO.java, DAOManager.java, MainDAO.java, maka dilanjutkan membuat database pada MySQL Command Line Client. Disini untuk pembuatan database diberi nama 'pbo' dan dilanjutkan membuat tabel yang bernama 'person'. Ketika dilakukan Select * From Person, maka data tabel masih kosong.

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

```
D:\Informatika\Mata Kuliah\Semester 4\PBO\Praktikum\9>javac *.java
```

8. Jalankan MainDAO dengan perintah:

```
java -classpath .\mysql-connector-java-[versi].jar;. MainDAO
```

```
D:\Informatika\Mata Kuliah\Semester 4\PBO\Praktikum\9>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 driver is
s automatically registered via the SPI and manual loading of the driver class is generally unnecessary.
INSERT INTO person(name) VALUES('Indra')
```

Setelah melakukan kompilasi semua source code dengan perintah javac *.java dilanjutkan menjalankan MainDAO dengan perintah diatas. Setelah berhasil, maka otomatis akan menampilkan INSERT INTO person(name) VALUES ('Indra') yang mana data tersebut berhasil dimasukkan kedalam tabel person. Setelah itu untuk melihat isi datanya bisa menggunakan perintah pada MySQL Command Line Client dengan perintah Select * FROM Person maka akan menampilkan seperti gambar berikut.

```
mysql> select * from person;
+----+-----+
| id | name |
+----+-----+
|  1 | Indra |
+----+-----+
1 row in set (0.01 sec)
```

B. Menggunakan Persistent Object sebagai objek terserialisasi

1. SerializePerson.java

```
/**
 * Nama file   :SerializePerson.java
 * Tanggal    :31 Mei 2023
 * Penulis     :Mochammad Arya Jadmika / 24060121130085
 * 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 objek person");
            s.close();
        }catch(IOException e){
            e.printStackTrace();
        }
    }
}
```

2. Compile dan jalankan program diatas dengan

```
javac SerializePerson.java
java SerializePerson
```

```
D:\Informatika\Mata Kuliah\Semester 4\PBO\Praktikum\9>javac SerializePerson.java
D:\Informatika\Mata Kuliah\Semester 4\PBO\Praktikum\9>java SerializePerson
selesai menulis objek person
```

3. ReadSerializedPerson.java

```
/**
 * Nama file   :ReadSerializedPerson.java
 * Tanggal    :31 Mei 2023
 * Penulis     :Mochammad Arya Jadmika / 24060121130085
 * 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 diatas dengan

```
javac ReadSerializedPerson.java
```

```
java ReadSerializedPerson
```

```
D:\Informatika\Mata Kuliah\Semester 4\PBO\Praktikum\9>javac ReadSerializedPerson.java
```

```
D:\Informatika\Mata Kuliah\Semester 4\PBO\Praktikum\9>java ReadSerializedPerson
serialized person name = Panji
```