

# **LAPORAN PRAKTIKUM**

## **PRAKTIKUM 9:**

### **“PERSISTENT OBJECT”**



**Disusun Oleh:**

Nama : Ajeng Nurhaliza Ar Rachman

NIM : 24060121140109

**PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK**

**LAB B1**

**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
 * Pembuat   : Ajeng Nurhaliza Ar Rachman
 * NIM/Lab   : 24060121140109 / B1
 * 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
 * Pembuat   : Ajeng Nurhaliza Ar Rachman
 * NIM/Lab   : 24060121140109 / B1
 * 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
 * Pembuat   : Ajeng Nurhaliza Ar Rachman
 * NIM/Lab   : 24060121140109 / B1
 * 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","");
        // 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
 * Pembuat   : Ajeng Nurhaliza Ar Rachman
 * NIM/Lab   : 24060121140109 / B1
 * 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
 * Pembuat   : Ajeng Nurhaliza Ar Rachman

```

```

* NIM/Lab   : 24060121140109 / B1
* 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* dan tabel

```

24060121140109> CREATE DATABASE PBO;
Query OK, 1 row affected (0.02 sec)

```

```

24060121140109> show databases;
+-----+
| Database |
+-----+
| information_schema |
| inventory          |
| mydb               |
| mysql              |
| organisasi         |
| pbo                 |
| performance_schema |
| praktikum1         |
| sys                |
| tugasmdb02         |
+-----+
10 rows in set (0.00 sec)

```

```

24060121140109> USE PBO;
Database changed
24060121140109> SHOW TABLES;
Empty set (0.03 sec)

24060121140109> CREATE TABLE Person(
    -> id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
    -> name VARCHAR(100));
Query OK, 0 rows affected (0.07 sec)

```

```

24060121140109> show tables;
+-----+
| Tables_in_pbo |
+-----+
| person        |
+-----+
1 row in set (0.02 sec)

```

```
24060121140109> SELECT * FROM Person;
Empty set (0.02 sec)
```

## 7. Kompilasi semua *source code* dan jalankan MainDAO

```
D:\UNDIP\Semester 4\PBO\Praktikum\Praktikum 9>javac *.java

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

## 8. Hasil Akhir

```
24060121140109> select * from person;
+----+-----+
| id | name |
+----+-----+
|  1 | Indra |
+----+-----+
1 row in set (0.00 sec)
```

## B. Menggunakan Persistent Object sebagai Objek Terserialisasi

### 1. SerializePerson.java

```
/*
 * Nama file : SerializePerson.java
 * Tanggal   : 31 Mei 2023
 * Pembuat   : Ajeng Nurhaliza Ar Rachman
 * NIM/Lab   : 24060121140109 / B1
 * 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 object person");
        s.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

```
D:\UNDIP\Semester 4\PBO\Praktikum\Praktikum 9>javac SerializePerson.java
```

```
D:\UNDIP\Semester 4\PBO\Praktikum\Praktikum 9>java SerializePerson
selesai menulis object person
```

## 2. ReadSerializedPerson.java

```

/*
 * Nama file : ReadSerializedPerson.java
 * Tanggal   : 31 Mei 2023
 * Pembuat   : Ajeng Nurhaliza Ar Rachman
 * NIM/Lab   : 24060121140109 / B1
 * 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();
        }
    }
}

```

```
D:\UNDIP\Semester 4\PBO\Praktikum\Praktikum 9>javac ReadSerializedPerson.java
```

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