

**LAPORAN PRAKTIKUM**

**PRAKTIKUM 9:**

**“PERSISTENT OBJECT”**



**Andre Fitara Agun Ni'am Muta'ali**

**24060121140129**

**PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK**

**LAB B**

**DEPARTEMEN ILMU KOMPUTER / INFORMATIKA**

**FAKULTAS SAINS DAN MATEMATIKA**

**UNIVERSITAS DIPONEGORO**

**SEMARANG**

**2023**

## A. Menggunakan Persistent Object sebagai Model Basis Data Relasional

### 1. PersonDAO.java

```
/**
 * File : PersonDAO.java
 * Penulis : Andre Fitara Agun Ni'am Muta'ali
 * Deskripsi : interface untuk person access object
 */
public interface PersonDAO{
    public void savePerson (Person p) throws Exception;
}
```

### 2. Person.java

```
/**
 * File : Person.java
 * Penulis : Andre Fitara Agun Ni'am Muta'ali
 * 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
 * Penulis : Andre Fitara Agun Ni'am Muta'ali
 * 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",
"");

        // 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
 * Penulis : Andre Fitara Agun Ni'am Muta'ali
 * 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
 * Penulis : Andre Fitara
 * 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

```
CREATE DATABASE
```

## 7. Jalankan MainDAO

semua file source code dan file mysql.jar dijadikan dalam 1 folder. Setelah menjalankan perintah, maka muncul pesan seperti SS di bawah dan telah ada INSERT INTO person(name) VALUES('Indra') yang berarti MainDAO berhasil dijalankan dan perintah untuk insert data ke dalam tabel person juga telah berhasil dilakukan.

```
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. Check Tabel

Periksa apakah ada lebih banyak catatan di tabel sesuai dengan perintah dijalankan sebelumnya dengan membuka SQL CLI dan menjalankan query seperti dibawah ini, sebelumnya terlihat saat memilih \* from person masih empty set, setelah menjalankan perintah java seperti di atas kemudian select \* from person maka sudah tertera data nama Indra dengan id 1 pada tabel person yang berarti program dan SQL CLI sudah terhubung yang ditandai dengan adanya penambahan data pada tabel person melalui perintah java yang dijalankan.

```
24060121140129>select * from person;  
+----+-----+  
| id | name |  
+----+-----+  
| 3 | Indra |  
+----+-----+  
1 row in set (0.00 sec)  
24060121140129>
```