

**LAPORAN PRAKTIKUM PRAKTIKUM  
9: PERSISTENT OBJECT**



**Disusun Oleh :**

**Muhammad Raihan Ridho  
24060121140131**

**PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK  
LAB B2**

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

**PRAKTIKUM 9: Persistent Object**

## A. Source Code yang digunakan Persistent Object sebagai Model Basis Data

### 1. PersonDAO.java

```
/**
 * File      : PersonDAO.java
 * Deskripsi  : interface untuk person access object
 * Nama      : Muhammad Raihan Ridho
 * NIM       : 24060121140131
 * Tanggal   : 2 juli 2023
 */

public interface PersonDAO {
```

### 2. Person.java

```
/**
 * File      : PersonDAO.java
 * Deskripsi  : Person database model
 * Nama      : Muhammad Raihan Ridho
 * NIM       : 24060121140131
 * Tanggal   : 2 juli 2023
 */

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;
    }
}
```

### 3. MySQLPersonDAO.java

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;

/**
 * Nama    : Muhammad Raihan Ridho
 * NIM     : 24060121140131
 * Tanggal : 2 juli 2023
 * File    : MySQLPersonDAO.java
 * 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","mraihanrdh21.");
        //kerjakan mysql query
        String query = "INSERT INTO person(name) VALUES('"+name+"')";
        System.out.println(query);
    }
}
```

### 4. DAOManager.java

```

/**
 * File : DAOManager.java
 * Nama   : Muhammad Raihan Ridho
 * NIM    : 24060121140131
 * Tanggal : 2 juli 2023
 * File : DAOManager.java
 * Deskripsi : pengelola DAO dalam program
 */

public class DAOManager {
    private PersonDAO personDAO;

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

```

#### 5. MainDAO.java

```

/**
 * File   : MainDAO.java
 * Nama    : Muhammad Raihan Ridho
 * NIM     : 24060121140131
 * Tanggal : 2 juli 2023
 * 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) {
        }
    }
}

```

#### 6. Buat database dengan nama pbo dan tabel pada database tersebut

```
mysql> create database pbo;
Query OK, 1 row affected (0.02 sec)
```

```
mysql> show databases;
```

Database
cafe
company
departments
information_schema
inventory
Kos_Budhe
mahasiswa
mysql
pbo
performance_schema
perusahaan
sys

Buat database menggunakan query CREATE DATABASE pbo lalu cek database sudah terbuat .

```
mysql> CREATE TABLE person( id int primary key auto_increment not null, name varchar(100));
```

Buat database sesuai dengan query diatas.

## 7. Compile file java pada terminal

```
raaihanrdh : javac *.java
```

Buka terminal baru di folder yang dimana berisi sourcode java dan pa ada hasil compile di atas, terlihat bahwa setelah perintah dijalankan, tidak ada pesan error yang muncul. Dengan tidak adanya pesan error saat menjalankan perintah, maka dapat dipastikan source code yang dibuat berhasil. Hal ini berarti tidak terdapat kesalahan sintaks atau masalah lainnya dalam source code yang dapat menghambat jalannya program.

## 8. Jalankan mainDAO dengan command

```
raaihanrdh : 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')
```

Untuk menjalankan program ini pastikan aplikasi code kalian sudah terhubung ke mysql connector, jika sudah dipastikan sudah terhubung ke mysqlconnector dan berada maka bisa jalankan perintah dengan **java -classpath ./mysql-connector-j-8.0.33.jar;. MainDAO** Setelah menjalankan perintah tersebut, akan muncul pesan yang menunjukkan bahwa program MainDAO berhasil dijalankan dan perintah untuk memasukkan data ke dalam tabel person telah berhasil dilakukan. Hal ini dapat dilihat dari pesan INSERT INTO person(name) VALUES('Indra'). Selanjutnya, dilakukan pemanggilan untuk memverifikasi data telah ditambahkan ke dalam tabel sesuai dengan perintah yang dijalankan sebelumnya. Pemanggilan perintah select \* from person ketika telah menambahkan tabel sebelumnya, data pada tabel person masih kosong (empty set).

Namun, setelah menjalankan perintah java seperti yang dijelaskan sebelumnya, kemudian menjalankan perintah `select * from person`, maka data dengan id 1 bernama Indra berhasil tercantum dalam tabel person. Hal ini menunjukkan bahwa program dan SQL CLI telah terhubung, sebagaimana ditandai dengan penambahan data ke dalam tabel person melalui perintah java yang telah dijalankan.

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