

LAPORAN PRAKTIKUM
PRAKTIKUM 9:
“PERSISTENT OBJECT”



Disusun Oleh :

Rafif Abbrar Maheswara
24060121140163

PRAKTIKUM JARINGAN KOMPUTER
LAB B1

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

PERSISTENT OBJECT

A. Menggunakan *Persistent Object* sebagai model basis data relasional

a. PersonDAO.java

```
/**
 * Nama : Rafif Abbrar Maheswara
 * NIM : 24060121140163
 * File : PesrsonDAO.java
 * Deskripsi : interface untuk person access object
 */
public interface PersonDAO{
    public void savePerson(Person p) throws Exception;
}
```

b. Person.java

```
/**
 * Nama : Rafif Abbrar Maheswara
 * NIM : 24060121140163
 * 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;
    }
}
```

c. MySQLPersonDAO.java

```
import java.sql.*;

/**
 * Nama      : Rafif Abbrar Maheswara
 * NIM       : 24060121140163
 * File      : MySQLPersonDAO.java
 * Deskripsi  : implentasi 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();
    }
}
```

d. DAOManager.java

```
/**
 * Nama      : Rafif Abbrar Mahesawara
 * NIM       : 24060121140163
 * 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;
    }
}
```

e. MainDAO.java

```
/**
 * Nama      : Rafif Abbrar Mahesawara
 * NIM       : 240601211140163
 * File      : MainDAO.java
 * Deskripsi  : file main
 */
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();
        }
    }
}
```

Buat Database dengan nama “pbo” dan tabel pada database tersebut

```
mysql> prompt 24060121140163>  
PROMPT set to '24060121140163> '  
24060121140163> create database pbo;  
Query OK, 1 row affected (0.02 sec)
```

```
24060121140163> use pbo;  
Database changed  
24060121140163> show tables;  
Empty set (0.02 sec)
```

```
24060121140163> CREATE TABLE person(  
    -> id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,  
    -> name VARCHAR(100));  
Query OK, 0 rows affected (0.03 sec)  
  
24060121140163> select * from person;  
Empty set (0.01 sec)
```

Untuk pembuatan database menggunakan SQL CLI dengan perintah seperti gambar di atas, pertama membuat database pbo seperti screenshot pertama, kemudian gunakan “use pbo” untuk mengaktifkan database pbo, saat show tables masih empty set, karena belum ada tabel yang dibuat

selanjutnya buat tabel sesuai perintah modul yaitu membuat tabel person dengan atribut atau kolom id dan name serta tipe data seperti SS kedua. Kemudian pada SS ketiga karena belum diisi data apapun maka saat select * from person masih empty set karena belum ada data.

Kompilasi semua source code menggunakan perintah; `javac *.java`

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

Setelah membuat database dan tabel, selanjutnya menjalankan semua source code yang telah dibuat menggunakan perintah pada gambar diatas. Setelah tidak ada masalah, semua source code berhasil di compile.

Jalankan MainDAO dengan perintah

```
java -classpath .\mysql-connector-j-8.0.33.jar;. MainDAO
```

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

Setelah menjalankan perintah di atas untuk membuat program dan SQL CLI terhubung. Semua file source code dan file mysql.jar dijadikan dalam 1 folder. Setelah menjalankan perintah di atas, maka muncul pesan seperti gambar di atas 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.

```
24060121140163> select * from person;
+----+-----+
| id | name  |
+----+-----+
|  1 | Indra  |
|  2 | Indra  |
+----+-----+
2 rows in set (0.00 sec)
```

Data tabel sudah otomatis terisi saat menjalankan perintah `select * from person;` pada mysql

Menggunakan persistent object sebagai objek terserialisasi

- SerializePerson.java

```
/*
 * Nama      : Rafif Abbrar Mahesawara
 * NIM       : 24060121140163
 * File      : SerializePerson.java
 * Deskripsi  : program untuk serialisasi program
 */
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();
        }
    }
}
```

Compile dan jalankan program menggunakan

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

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

Program berhasil di compile dan dijalankan dan menghasilkan “selesai menulis objek person” seperti pada modul.

ReadSerializedPerson.java

```
/**
 * Nama      : Rafif Abbrar Maheswara
 * NIM       : 24060121140163
 * 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();
        }
    }
}
```

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

Jika compile berhasil, akan muncul tulisan “serialized person name = Panji”.

