

LAPORAN PRAKTIKUM

PRAKTIKUM 9:

“Persistent Object”



Disusun Oleh:

Darrel Arsa Putranto

(24060121140155)

PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK LAB B1

DEPARTEMEN ILMU KOMPUTER / INFORMATIKA

FAKULTAS SAINS DAN MATEMATIKA

UNIVERSITAS DIPONEGORO

SEMARANG

2023

1. Persistent Objek Sebagai Model Basis Data Relasional

Praktikum ini akan membahas mengenai implementasi kelas-kelas pada java yang dapat terhubung dengan basis data. Dengan menghubungkan IDE yang kita gunakan dalam hal ini IntelliJ IDE dengan modul SQL connector.

Kode program kelas-kelas java :

a. PersonDAO.java

Interface Untuk person access object.

```
/**
 * Penulis : Darrel Arsa Putranto
 * File : PersonDAO.java
 * Deskripsi : Interface untuk person access object
 */

public interface PersonDAO{
    public void savePerson(Person P) throws exception;
}
```

b. Person.java

Kelas ini berisi konstruktor-konstruktor person dan berisi *method getter* yaitu getId() dan getName().

```
/**
 * Penulis : Darrel Arsa Putranto
 * 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;
    }
}
```

c. MySQLPersonDAO.java

Kelas ini kita akan mengatur koneksi antara java dengan basis data pada line

```
Connection con = DriverManager.getConnection  
("jdbc:mysql://localhost/pbo","root","");
```

Kita dapat memilih schema dan memasukkan username dan password untuk basis data. Pada kelas ini, kita dapat membuat query dan mengeksekusinya untuk modifikasi pada tabel basis data.

```
/**  
 * Penulis : Darrel Arsa Putranto  
 * 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","");  
        //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

Pengelola DAO dalam program.

```
/**
 * Penulis : Darrel Arsa Putranto
 * 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

Main program untuk akses DAO

```
/**
 * Penulis : Darrel Arsa Putranto
 * File : MainDAO.java
 * Deskripsi : Main program untuk akses DAO
 */

public class MainDAO {
    public static void main(String args[]) {
        Person person = new Person("Budi");
        DAOManager m = new DAOManager();
        m.setPersonDAO(new MySQLPersonDAO());
        try {
            m.getPersonDAO().savePerson(person);
        }catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Kemudian masuk ke MySQL dan membuat schema dengan nama “PBO” dan membuat tabel dengan query :

```
CREATE TABLE person(id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,name VARCHAR(100))
```

2. Menggunakan Persistent Object sebagai objek terealisasi

a. SerializePerson.java

```
/**
 * Penulis : Darrel Arsa Putranto
 * File : SerializePerson.java
 * 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();
        }
    }
}
```

Hasil :

Selesai Menulis Objek Person

b. ReadSerializePerson.java

```
/**
 * Penulis : Darrel Arsa Putranto
 * 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();
        }
    }
}
```

Hasil :

Serialized person name = Joko