Praktikum 9

Persistent Object

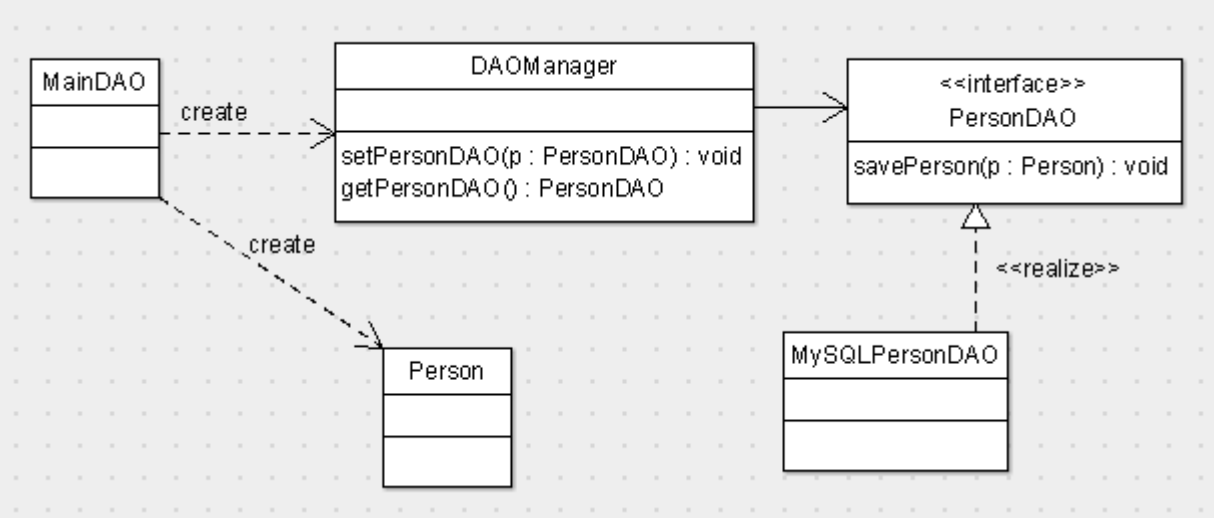
1. **Tujuan**
   a. Mahasiswa mampu membuat dan menggunakan persistent object
   b. Mahasiswa mampu menggunakan persistent object sebagai model basis data relasional
   c. Mahasiswa mampu menggunakan persistent object sebagai objek terserialisasi

2. **Landasan Teori**
   [lihat slide kuliah]

3. **Langkah Praktikum**

3.1. Menggunakan Persistent Object sebagai model basis data relasional

Pada praktikum ini, anda akan membuat implementasi kelas sebagaimana ditunjukkan diagram kelas berikut :



1. Buatlah interface PersonDAO.java :

```
1  /**
2   *  File : PersonDAO.java
3   *  Deskripsi : interface untuk person access object
4   */
5  public interface PersonDAO{
6      public void savePerson(Person p) throws Exception;
7  }
```

2. Buatlah kelas Person.java

```java
/**
 *  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;
    }
}
```

3. Buatlah kelas MySQLPersonDAO.java

```java
import java.sql.*;
/**
 *  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();
    }
}
```

4. Buatlah kelas DAOManager.java

```java
/**
 *  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;
    }
}
```

5. Buatlah kelas MainDAO.java

```java
/**
 *  File : MainDAO.java
 *  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. Buatlah database dengan nama 'pbo' dan tabel pada database tersebut dengan :
   `CREATE TABLE person(id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,name VARCHAR(100))`

7. Download MySQL Driver dari http://dev.mysql.com/downloads/connector/j/, ekstrak file berekstensi *.jar (mysql-connector-java-[versi].jar) dan letakkan satu direktori dengan source code program.

8. Kompilasi semua source code dengan perintah : **`javac *.java`**

9. Jalankan MainDAO dengan perintah :
   `java -classpath .\mysql-connector-java-[versi].jar;. MainDAO`

10. Lihat apakah terjadi penambahan record pada tabel !

3.2. Menggunakan Persistent Object sebagai objek terserialisasi

1. Buatlah kelas SerializePerson.java berikut, untuk menyimpan objek dalam file yang bernama "person.ser" :

```java
/**
 * 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();
        }
    }
}
```

2. Compile, dan jalankan program di atas dengan

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

3. Buatlah kelas ReadSerializedPerson.java berikut untuk membaca objek yang telah terserialisasi :

```java
/**
 * 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();
        }
    }
}
```

4. Compile dan jalankan kelas di atas dengan :

`javac ReadSerializedPerson.java`

`java ReadSerializedPerson`

-- Selamat Mengerjakan --