

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

Persistent object, dalam pemrograman berorientasi objek, adalah objek yang memiliki kemampuan untuk menyimpan data dalam aplikasi. Dalam konteks pemrograman, "persisten" berarti bahwa objek tersebut dapat bertahan melewati siklus hidup program.

Keberadaan persistent object sangat penting dalam sistem yang membutuhkan penyimpanan data yang persisten, seperti aplikasi basis data, sistem manajemen informasi, atau aplikasi yang memerlukan pemulihan data setelah restart.

Dalam paradigma pemrograman berorientasi objek, persistent object sering kali mewakili entitas dalam dunia nyata yang memiliki atribut dan perilaku. Objek-objek ini disimpan dan diakses melalui mekanisme yang disediakan oleh sistem manajemen basis data atau sistem penyimpanan persisten lainnya.

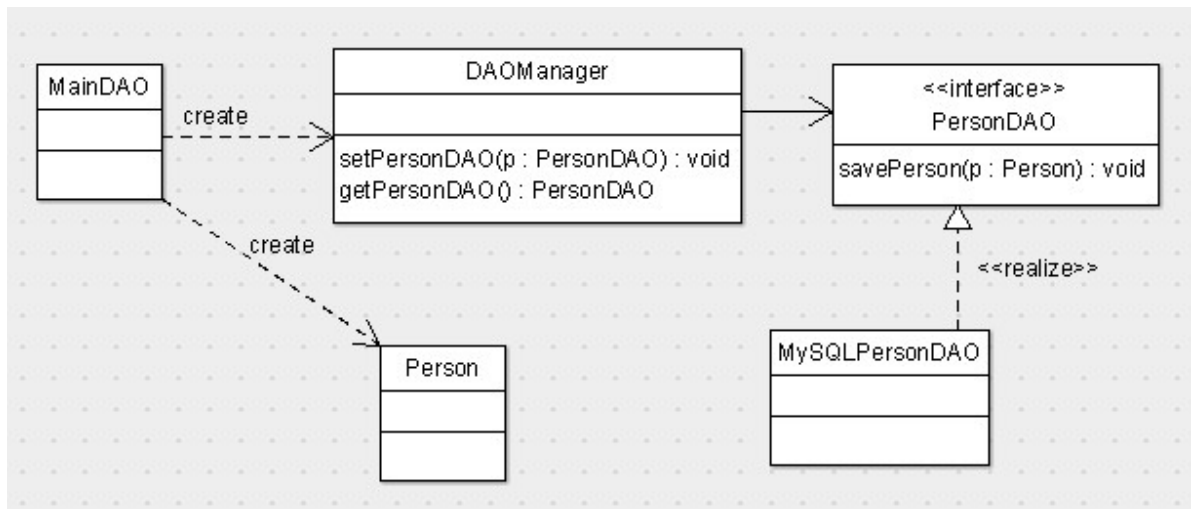
Pembuatan persistent objek berbasis Java dilakukan menggunakan POJO (*Plain Old Java Object*). POJO hanya berisi atribut, yang mewakili kolom-kolom basis data relasional, serta *getter* dan *setter*. Pemanfaatan POJO bisa dilakukan secara langsung ataupun menggunakan pola akses data semisal *Data Access Object* (DAO).

Pada praktikum ini akan dijelaskan disampaikan pemanfaatan persisten objek menggunakan pola Data Access Object (DAO). Selain itu, disampaikan juga materi serialisasi objek.

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

```

1  /**
2   *   File : Person.java
3   *   Deskripsi : Person database model
4   */
5  public class Person{
6      private int id;
7      private String name;
8
9      public Person(String n){
10         name = n;
11     }
12
13     public Person(int i,String n){
14         id = i;
15         name = n;
16     }
17
18     public int getId(){
19         return id;
20     }
21
22     public String getName(){
23         return name;
24     }
25 }
  
```

3. Buatlah kelas MySQLPersonDAO.java

```
1  import java.sql.*;
2  /**
3   * File : MySQLPersonDAO.java
4   * Deskripsi : implementasi PersonDAO untuk MySQL
5   */
6  public class MySQLPersonDAO implements PersonDAO{
7      public void savePerson(Person person) throws Exception{
8          String name = person.getName();
9          //membuat koneksi, nama db,user,password menyesuaikan
10         Class.forName("com.mysql.jdbc.Driver");
11         Connection con = DriverManager.getConnection(
12             "jdbc:mysql://localhost/pbo","root","");
13         //kerjakan mysql query
14         String query = "INSERT INTO person(name) VALUES ('"+name+"')";
15         System.out.println(query);
16         Statement s = con.createStatement();
17         s.executeUpdate(query);
18         //tutup koneksi database
19         con.close();
20     }
21 }
```

4. Buatlah kelas DAOManager.java

```
1  /**
2   * File : DAOManager.java
3   * Deskripsi : pengelola DAO dalam program
4   */
5  public class DAOManager{
6      private PersonDAO personDAO;
7
8      public void setPersonDAO(PersonDAO person){
9          personDAO = person;
10     }
11     public PersonDAO getPersonDAO(){
12         return personDAO;
13     }
14 }
```

5. Buatlah kelas MainDAO.java

```

1  /**
2   * File : MainDAO.java
3   * Deskripsi : Main program untuk akses DAO
4   */
5  public class MainDAO{
6      public static void main(String args[]){
7          Person person = new Person("Indra");
8          DAOManager m = new DAOManager();
9          m.setPersonDAO(new MySQLPersonDAO());
10         try{
11             m.getPersonDAO().savePerson(person);
12         }catch(Exception e){
13             e.printStackTrace();
14         }
15     }
16 }

```

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" :

```

1  /**
2   * File : SerializePerson.java
3   * Deskripsi : Program untuk serialisasi objek Person
4   */
5   import java.io.*;
6   //class Person
7   class Person implements Serializable{
8       private String name;
9       public Person(String n){
10           name = n;
11       }
12       public String getName(){
13           return name;
14       }
15   }
16   //class SerializePerson
17   public class SerializePerson{
18       public static void main(String[] args){
19           Person person = new Person("Panji");
20           try{
21               FileOutputStream f= new FileOutputStream("person.ser");
22               ObjectOutputStream s = new ObjectOutputStream(f);
23               s.writeObject(person);
24               System.out.println("selesai menulis objek person");
25               s.close();
26           }catch(IOException e){
27               e.printStackTrace();
28           }
29       }
30   }

```

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 :

```

1  /**
2   * File : ReadSerializedPerson.java
3   * Deskripsi : Program untuk serialisasi objek Person
4   */
5   import java.io.*;
6
7   public class ReadSerializedPerson{
8       public static void main(String[] args){
9           Person person = null;
10          try{
11              FileInputStream f = new FileInputStream("person.ser");
12              ObjectInputStream s = new ObjectInputStream(f);
13              person = (Person)s.readObject();
14              s.close();
15              System.out.println("serialized person name = "+person.getName());
16          }catch(Exception ioe){
17              ioe.printStackTrace();
18          }
19      }
20  }

```

4. Compile dan jalankan kelas di atas dengan :

javac ReadSerializedPerson.java

java ReadSerializedPerson

-- Selamat Mengerjakan --