LAPORAN PRAKTIKUM
PRAKTIKUM 9 :
“PERSISTENTS OBJECT”

Disusun Oleh :

Nama: Annisa Kumala Dewi
NIM : 24060121120025

PEMROGRAMAN BERBASIS OBJEK
LAB B

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

A. Menggunakan Persistent Object Sebagai Model Basis Data Relasional

1. Interface PersonDAO.java

```
/* File      : PersonDAO.java
   Penulis   : Annisa Kumala Dewi / 24060121120025
   Deskripsi : interface untuk person access object*/

public interface PersonDAO {
    public void savePerson(Person p) throws Exception;
}
```

2. Class Person.java

```
/* File      : Person.java
   Penulis   : Annisa Kumala Dewi / 24060121120025
   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. Class MySQL.PersonDAO.java

```
import java.sql.Statement;
/* File      : MySQLPersonDAO.java
   Penulis   : Annisa Kumala Dewi/ 24060121120025
  Deskripsi  :  program  penggunaan  objek  ArrayList  sebagai
Collection class */

import java.sql.*;

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();
    }
}
```

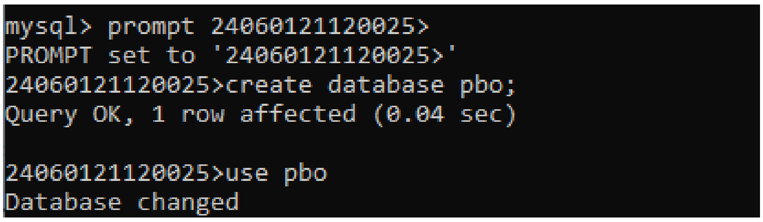
4. Class DAOManager.java

```
/* File      : DAOManager.java
   Penulis   : Annisa Kumala Dewi / 24060121120025
   Deskripsi : pengelola DAO dalam program */

public class DAOManager {
    private PersonDAO personDAO;

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

    public PersonDAO getPersonDAO() {
        return personDAO;
    }
}
```

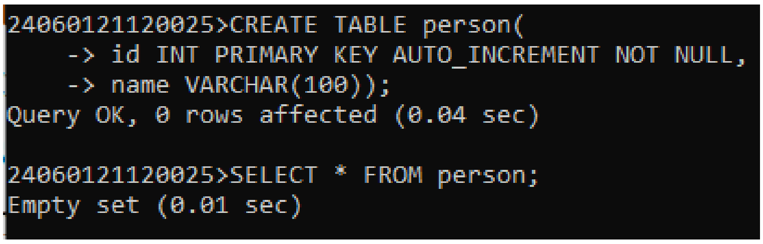
5. Class mainDAO.java

```
/* File      : MainDAO.java
   Penulis   : Annisa Kumala Dewi / 24060121120025
   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. Database pbo dengan Tabel person (id (PK) dan name (VARCHAR(100)))

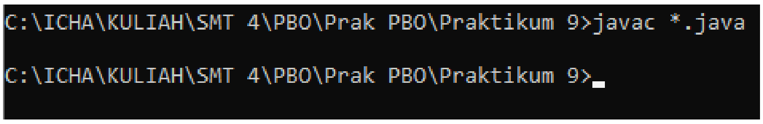


```
mysql> prompt 24060121120025>
PROMPT set to '24060121120025>'
24060121120025>create database pbo;
Query OK, 1 row affected (0.04 sec)

24060121120025>use pbo
Database changed
```



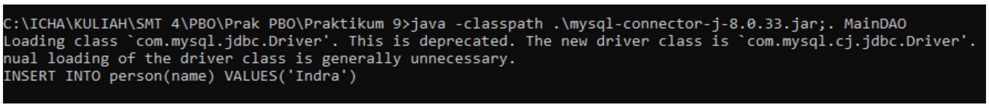
```
24060121120025>CREATE TABLE person(
    -> id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
    -> name VARCHAR(100));
Query OK, 0 rows affected (0.04 sec)

24060121120025>SELECT * FROM person;
Empty set (0.01 sec)
```

7. Compile All Code With javac *.java



```
C:\ICHA\KULIAH\SMT 4\PBO\Prak PBO\Praktikum 9>javac *.java

C:\ICHA\KULIAH\SMT 4\PBO\Prak PBO\Praktikum 9>
```

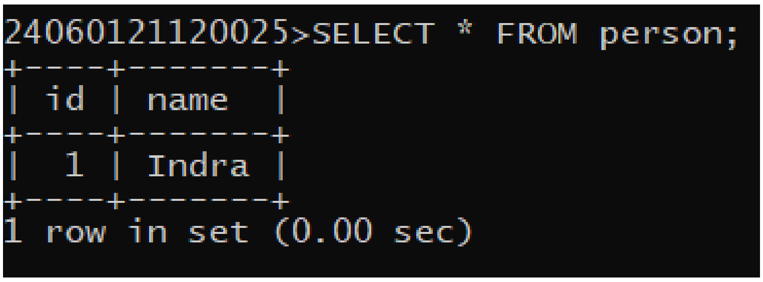
8. Run MainDAO dengan java -classpath .mysql-connector-j-8.0.33.jar;. MainDAO



```
C:\ICHA\KULIAH\SMT 4\PBO\Prak 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'.
nual loading of the driver class is generally unnecessary.
INSERT INTO person(name) VALUES('Indra')
```

Perintah **java -classpath .mysql-connector-j-8.0.33.jar;. MainDAO** digunakan untuk menghubungkan program dengan MySQL. Ketika berhasil dijalankan akan muncul pesan INSERT INTO person(name) VALUES(‘indra’) yang menunjukan bahwa program MainDAO berhasil dijalankan dengan memasukkan data kedalam tabel person.

9. Hasil Penambahan Record Tabel

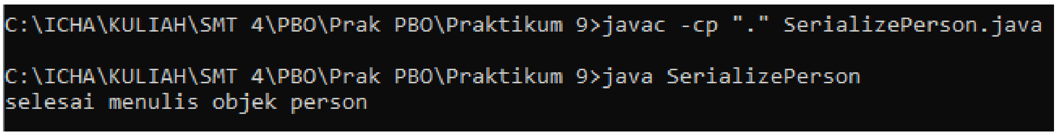


```
24060121120025>SELECT * FROM person;
+----+-------+
| id | name  |
+----+-------+
|  1 | Indra |
+----+-------+
1 row in set (0.00 sec)
```

B. Menggunakan Persistent Object Sebagai Objek Terealisasi

1. Class SerializePerson.java

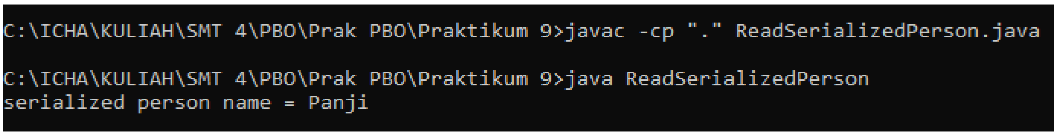
```
/* File      : SerializePerson.java
   Penulis   : Annisa Kumala Dewi / 24060121120025
   Deskripsi : Program untuk serialisasi objek Person*/

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();
        }
    }
}
```



```
C:\ICHA\KULIAH\SMT 4\PBO\Prak PBO\Praktikum 9>javac -cp "." SerializePerson.java

C:\ICHA\KULIAH\SMT 4\PBO\Prak PBO\Praktikum 9>java SerializePerson
selesai menulis objek person
```

2. Class ReadSerializePerson.java

```
/* File      : ReadSerializePerson.java
   Penulis   : Annisa Kumala Dewi / 24060121120025
   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();
        }
    }
}
```



```
C:\ICHA\KULIAH\SMT 4\PBO\Prak PBO\Praktikum 9>javac -cp "." ReadSerializedPerson.java

C:\ICHA\KULIAH\SMT 4\PBO\Prak PBO\Praktikum 9>java ReadSerializedPerson
serialized person name = Panji
```