Nama : Adira Rahmana Akbar
NIM : 24060121140114
LAB : PBO C1

PRAKTIKUM 9 : Persistent Object

A. Menggunakan Persistent Object sebagai model basis data relasional

1. PersonDAO.java

```
/*
 * File : PersonDAO.java 31/05/23
 * Penulis : Adira Rahmana Akbar – 24060121140114
 * Deskripsi : Interface untuk person access object
 */

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

2. Person.java

```
/**
 * File : Person.java 31/05/23
 * Penulis : Adira Rahmana Akbar - 24060121140114
 * 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. MySQLPersonDAO.java

```
/**
 * File : MySQLPersonDAO.java 31/05/23
 * Penulis : Adira Rahmana Akbar - 24060121140114
 * Deskripsi : implementasi PersonDAO untuk MySQL
 */

import java.sql.*;

public class MySQLPersonDAO implements PersonDAO {
    public void savePerson(Person person) throws Exception {
        String name = person.getName();
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/pbo", "root",
"woles123");
        String query = "INSERT INTO person(name) VALUES ('" + name +
"')";
        System.out.println(query);
        Statement s = con.createStatement();
        s.executeUpdate(query);
        con.close();
    }
}
```

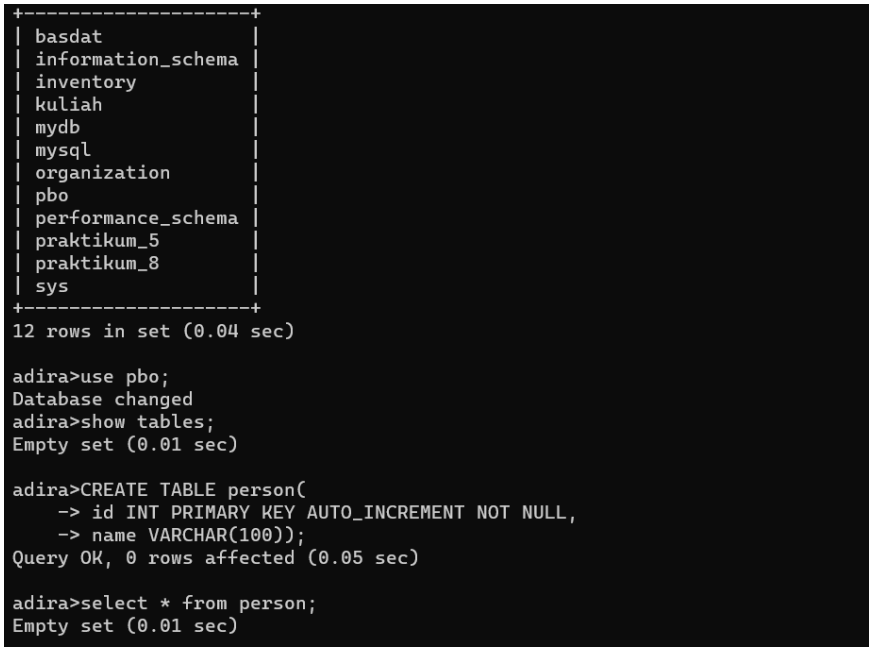
4. DAOManager.java

```
/**
 * File : DAOManager.java 31/05/23
 * Penulis : Adira Rahmana Akbar - 24060121140114
 * Deskripsi : pengelola DAO dalam program
 */

public class DAOManager {
    private PersonDAO personDAO;

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

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

5. MainDAO.java

```
/**
 * File : MainDAO.java 31/05/23
 * Penulis : Adira Rahmana Akbar - 24060121140114
 * Deskripsi : Main program untuk akses DAO
 */

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

6. Membuat database dengan nama “pbo” dan membuat tabel pada database.

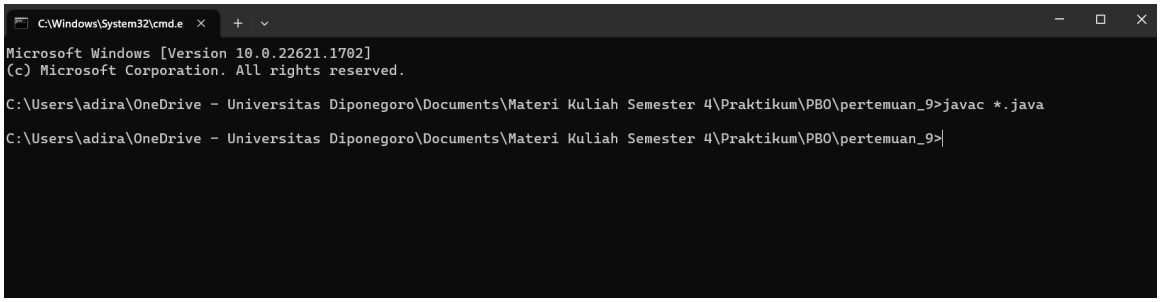


```
+--------------------+
| basdat             |
| information_schema |
| inventory          |
| kuliah             |
| mydb               |
| mysql              |
| organization       |
| pbo                |
| performance_schema |
| praktikum_5        |
| praktikum_8        |
| sys                |
+--------------------+
12 rows in set (0.04 sec)

adira>use pbo;
Database changed
adira>show tables;
Empty set (0.01 sec)

adira>CREATE TABLE person(
    -> id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
    -> name VARCHAR(100));
Query OK, 0 rows affected (0.05 sec)

adira>select * from person;
Empty set (0.01 sec)
```

Penjelasan:

Gunakan CREATE DATABASE pbo; untuk membuat database baru yaitu database pbo. Lalu gunakan database pbo untuk membuat tabel person dengan cara use pbo; lalu CREATE TABLE person. Setelah dimasukkan atribut dari tabel tersebut, bisa dicek terlebih dahulu apakah tabel tersebut berhasil dibuat, bisa dengan menggunakan SHOW TABLES; atau langsung mengecek isi dari tabel tersebut dengan cara SELECT * FROM person;. Karena tabel belum diisi dengan data apapun, maka output yang diberikan adalah empty set.

7. Compile kode dengan perintah: **`javac *.java`**

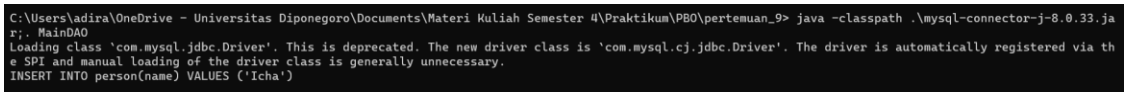


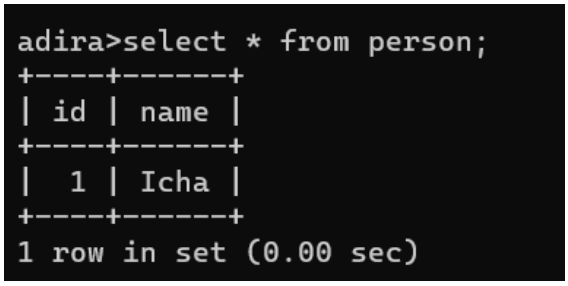
Penjelasan:

Ketika semua file dalam folder berhasil dijalankan, output yang dikeluarkan ketika dicompile tidak mengeluarkan apa-apa karena memang belum terhubung dengan SQL dan belum terdapat datanya. Hal tersebut mengisyaratkan bahwa kode tidak terdapat error.

8. Jalankan MainDAO.java dengan perintah:

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





Penjelasan:

Untuk menjalankan perintah tersebut, file MainDAO dan mysql.jar harus berada pada satu folder yang sama agar dapat berjalan sesuai terhadap database. Setelah menjalankan perintah tersebut, akan muncul pesan yang menunjukkan bahwa program MainDAO berhasil dijalankan dan perintah untuk memasukkan data ke dalam tabel person telah berhasil dilakukan. Hal ini dapat dilihat dari pesan INSERT INTO person(name) VALUES('Icha').

Setelahnya, dapat dibuka kembali kepada database pbo yang sudah dibuat, lalu masukkan command SELECT * FROM person;. Hasil atau output dari perintah tersebut adalah data yang sudah dimasukkan sebelumnya. Hal ini memperlihatkan bahwa program dan SQL CLI sudah terhubung.

B. Menggunakan Persistent Object sebagai Objek Terserialisasi

1. SerializePerson.java

```
/**
 * File : SerializePerson.java 31/05/23
 * Penulis : Adira Rahmana Akbar - 24060121140114
 * Deskripsi : pengelola DAO dalam program
 */

import java.io.*;

class Person implements Serializable {
    private String name;

    public Person(String n) {
        name = n;
    }

    public String getName() {
        return name;
    }
}

public class SerializePerson {
    public static void main(String[] args) {
        Person person = new Person("Kumala");
        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 keluaran:

```
PS C:\Users\adira\OneDrive - Universitas Diponegoro\Documents\Materi Kuliah Semester 4\Praktikum\PBO\pertemuan_9> cd "c:\Users\adira\OneDrive - Univer
sitas Diponegoro\Documents\Materi Kuliah Semester 4\Praktikum\PBO\pertemuan_9\" ; if ($?) { javac SerializePerson.java } ; if ($?) { java SerializePer
son }
Selesai menulis objek person
```

2. ReadSerializePerson.java

```
/**
 * File : Person.java 31/05/23
 * Penulis : Adira Rahmana Akbar - 24060121140114
 * 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 keluaran:

```
PS C:\Users\adira\OneDrive - Universitas Diponegoro\Documents\Materi Kuliah Semester 4\Praktikum\PBO\pertemuan_9> cd "c:\Users\adira\OneDrive - Universitas Di
negoro\Documents\Materi Kuliah Semester 4\Praktikum\PBO\pertemuan_9\" ; if ($?) { javac ReadSerializedPerson.java } ; if ($?) { java ReadSerializedPerson }
serialized person name = Kumala
```