Nama : Lulus Dwiyan Mita

Nim: 24060121120029

Lab B1 Pemrograman Berorientasi Objek

```
/*
*PersonDAO.java 06/06/2023
*Penulis: Lulus Dwiyan Mita 24060121120029
*Deskripsi: interface untuk person access object
*Lab: B1
*/
public interface PersonDAO{
    public void savePerson(Person p) throws Exception;
}
```

```
/*
*Person.java 06/06/2023
*Penulis: Lulus Dwiyan Mita 24060121120029
*Deskripsi: Person database model
*Lab: B1
*/
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;
   }
}
```

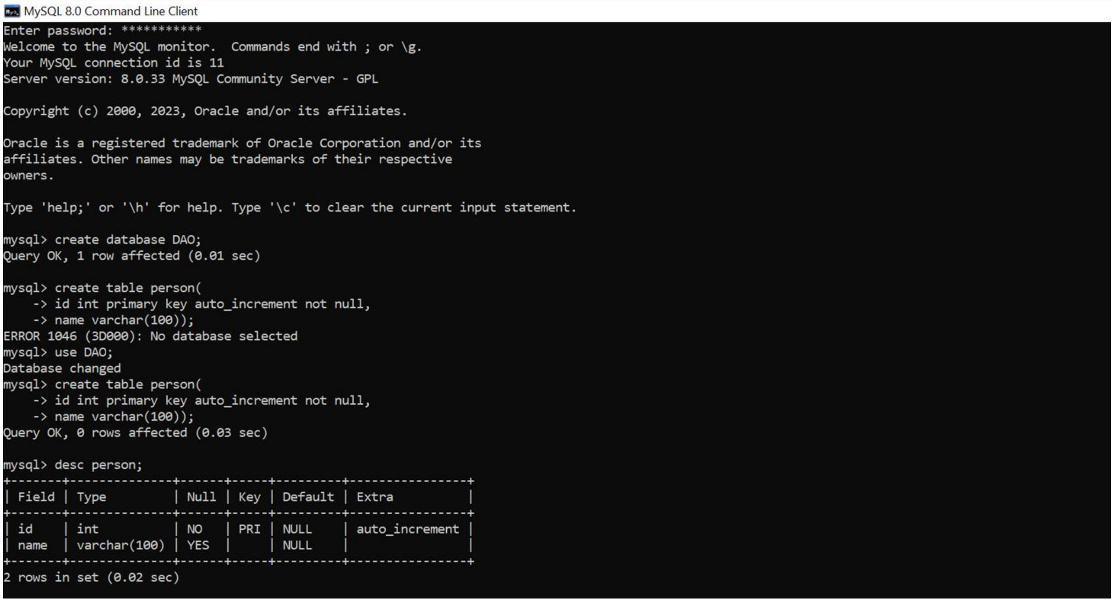
```
import java.sql.*;
/*
*MySQLPersonDAO.java 06/06/2023
*Penulis: Lulus Dwiyan Mita 24060121120029
*Deskripsi: implementasi PersonDAO untuk MySQL
*Lab: B1
*/

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/DAO","root","lulu
smita13");
         //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();
     }
}
```

```
/*
*DAOManager.java 06/06/2023
*Penulis: Lulus Dwiyan Mita 24060121120029
*Deskripsi: pengelola DAO dalam program
*Lab: B1
*/

public class DAOManager{
   private PersonDAO personDAO;

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

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

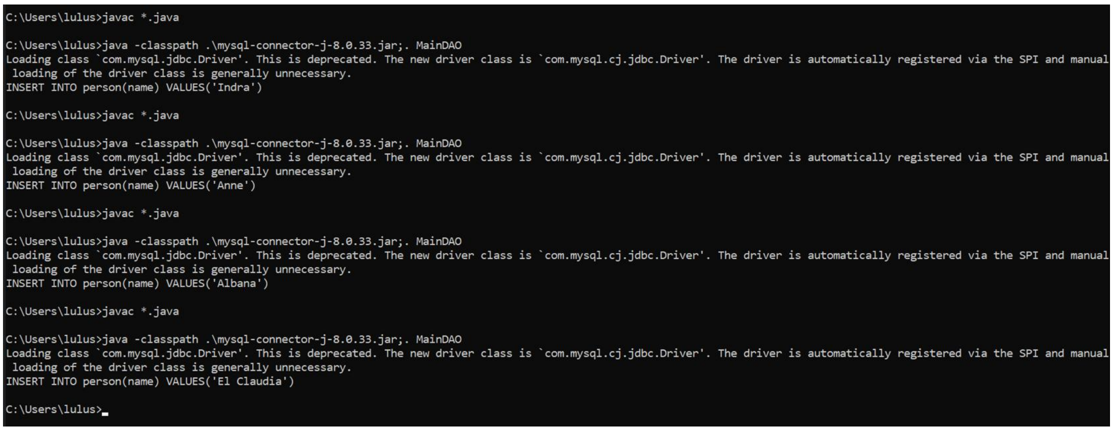
```
/*
*MainDAO.java 06/06/2023
*Penulis: Lulus Dwiyan Mita 24060121120029
*Deskripsi: Main program untuk akses DAO
*Lab: B1
*/

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();
        }
    }
}
```

Membuat database dengan nama DAO dan tabel yang terdapat pada database tersebut



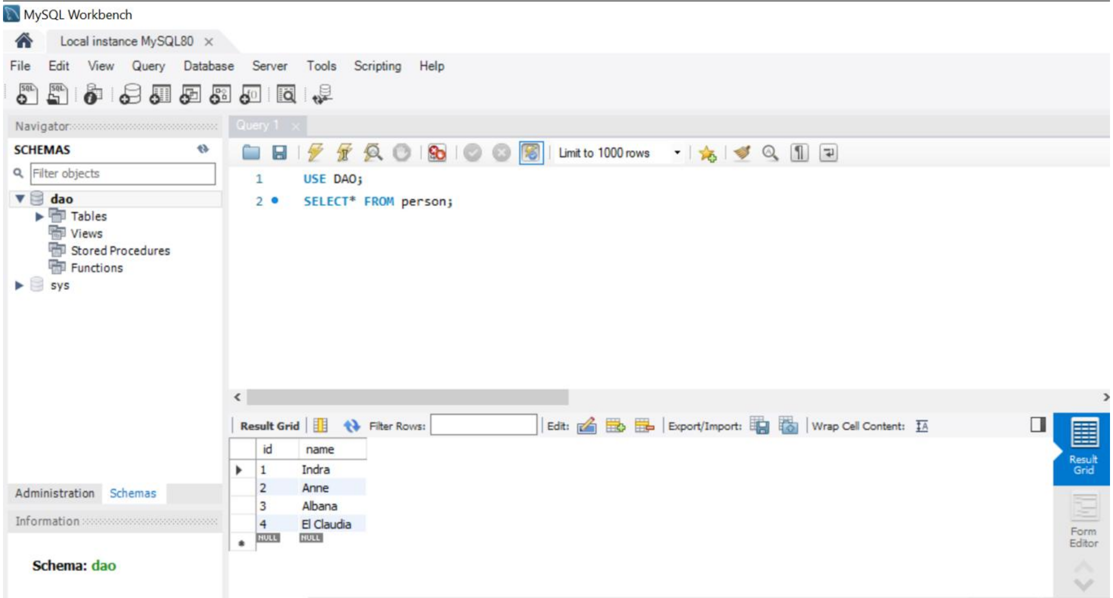
```
Enter password: ***********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database DAO;
Query OK, 1 row affected (0.01 sec)

mysql> create table person(
    -> id int primary key auto_increment not null,
    -> name varchar(100));
ERROR 1046 (3D000): No database selected
mysql> use DAO;
Database changed
mysql> create table person(
    -> id int primary key auto_increment not null,
    -> name varchar(100));
Query OK, 0 rows affected (0.03 sec)

mysql> desc person;
+-------+--------------+------+-----+---------+----------------+
| Field | Type         | Null | Key | Default | Extra          |
+-------+--------------+------+-----+---------+----------------+
| id    | int          | NO   | PRI | NULL    | auto_increment |
| name  | varchar(100) | YES  |     | NULL    |                |
+-------+--------------+------+-----+---------+----------------+
2 rows in set (0.02 sec)
```

Meng-compile semua source code dan menjalankan MainDAO



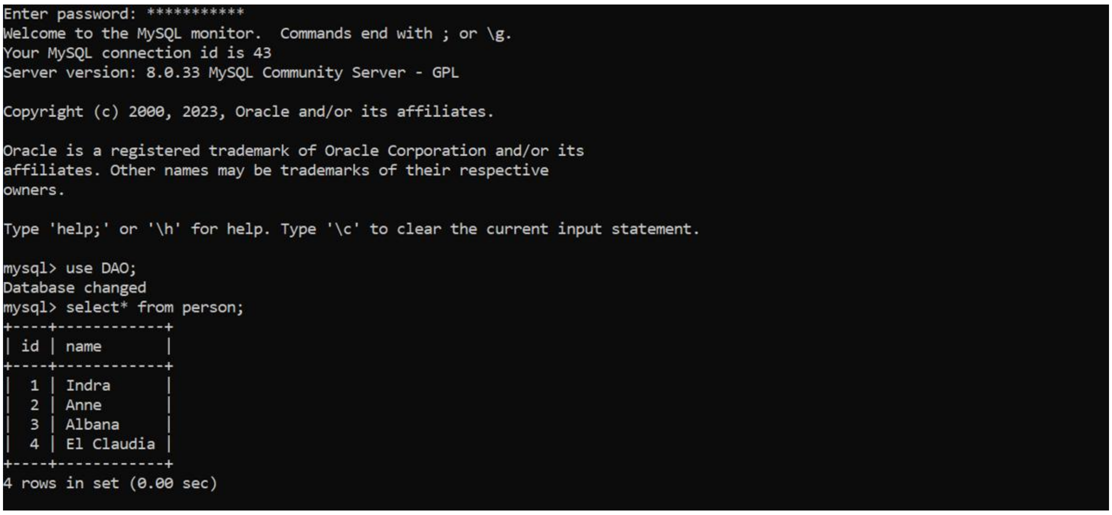
```
C:\Users\lulus>javac *.java

C:\Users\lulus>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'. The driver is automatically registered via the SPI and manual
 loading of the driver class is generally unnecessary.
INSERT INTO person(name) VALUES('Indra')

C:\Users\lulus>javac *.java

C:\Users\lulus>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'. The driver is automatically registered via the SPI and manual
 loading of the driver class is generally unnecessary.
INSERT INTO person(name) VALUES('Anne')

C:\Users\lulus>javac *.java

C:\Users\lulus>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'. The driver is automatically registered via the SPI and manual
 loading of the driver class is generally unnecessary.
INSERT INTO person(name) VALUES('Albana')

C:\Users\lulus>javac *.java

C:\Users\lulus>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'. The driver is automatically registered via the SPI and manual
 loading of the driver class is generally unnecessary.
INSERT INTO person(name) VALUES('El Claudia')

C:\Users\lulus>_
```

MySQL Workbench



```
USE DAO;
SELECT* FROM person;
```

id	name
1	Indra
2	Anne
3	Albana
4	El Claudia
NULL	NULL



```
Enter password: ***********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use DAO;
Database changed
mysql> select* from person;
+----+------------+
| id | name       |
+----+------------+
|  1 | Indra      |
|  2 | Anne       |
|  3 | Albana     |
|  4 | El Claudia |
+----+------------+
4 rows in set (0.00 sec)
```

```
/*
*SerializePerson.java 06/06/2023
*Penulis: Lulus Dwiyan Mita 24060121120029
*Deskripsi: Program untuk serialisasi objek Person
*Lab: B1
*/
import java.io.*;
//class Person
class Person implements Serializable{
      private String name;
      public Person(String n){
             name=n;
      }
      public String getName(){
             return name;
      }
}
//class SerializePersonFil
public class SerializePerson{
    public static void main(String[] args){
         Person person = new Person("Lulus");
         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();
          }
     }
}
```

```
/*
*ReadSerializedPerson.java 06/06/2023
*Penulis: Lulus Dwiyan Mita 24060121120029
*Deskripsi: Program untuk serialisasi objek Person
*Lab: B1
*/
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();
          }
    }
}
```