LAPORAN PRAKTIKUM
PRAKTIKUM 9:
PERSISTENT OBJECT

Disusun Oleh:

Muhammad Naufal Arkan
24060121130073

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
LAB B

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

========== 3.1 ==========

1. Interface PersonDao.java

```
/**
 * File : PersonDAO.java        31/05/2023
 * Penulis  : Muhammad Naufal Arkan/24060121130073
 * Deskripsi : Interface untuk person access object
*/

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

2. Class Person.java

```
/**
 * File : Person.java        31/05/2023
 * Penulis  : Muhammad Naufal Arkan/24060121130073
 * 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 MySQLPersonDao.java

```
/**
 * File : MySQLPersonDAO.java        31/05/2023
 * Penulis  : Muhammad Naufal Arkan/24060121130073
 * Deskripsi : implemnetasi PersonDAO untuk MySQL
 */
```

```
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

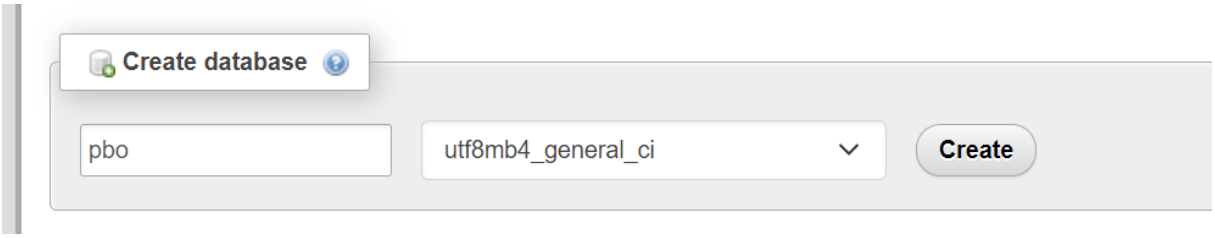
```
/**
 * File : DAOManager.java        31/05/2023
 * Penulis  : Muhammad Naufal Arkan/24060121130073
 * Deskripsi : pengelola DAO dalam program
*/

public class DAOManager {
    private PersonDAO personDAO;

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

5. Class MainDAO.java

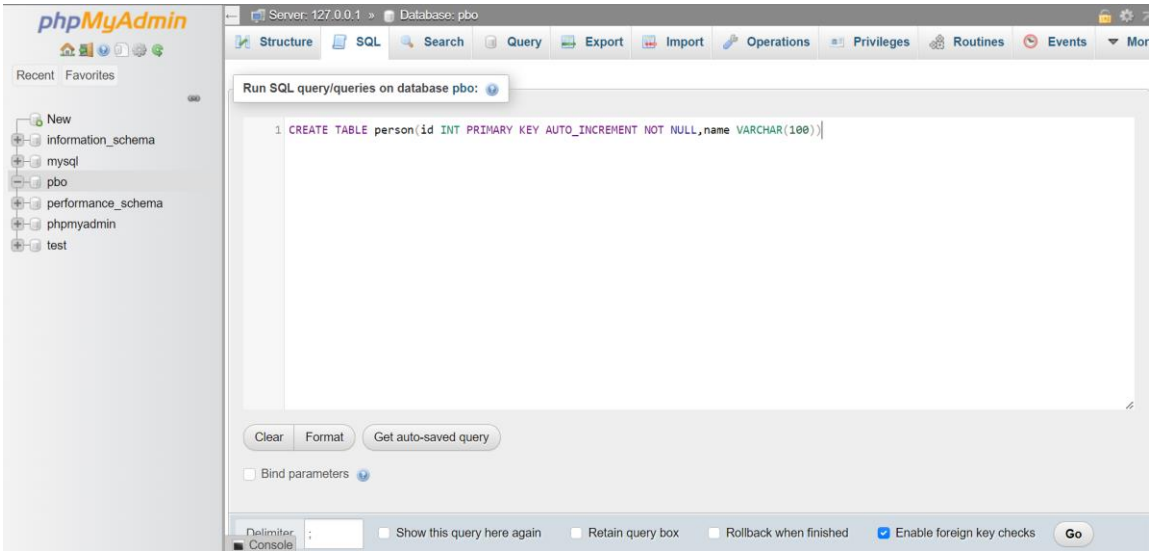
```
/**
 * File : MainDAO.java        31/05/2023
 * Penulis  : Muhammad Naufal Arkan/24060121130073
 * Deskripsi : Main program untuk akses DAO
*/
public class MainDAO {
    public static void main(String args[]){
        Person person = new Person("Indra");
```

```
            DAOManager m = new DAOManager();
            m.setPersonDAO(new MySQLPersonDAO());
            try{
                m.getPersonDAO().savePerson(person);
            }catch(Exception e){
                e.printStackTrace();
            }
        }
}
```

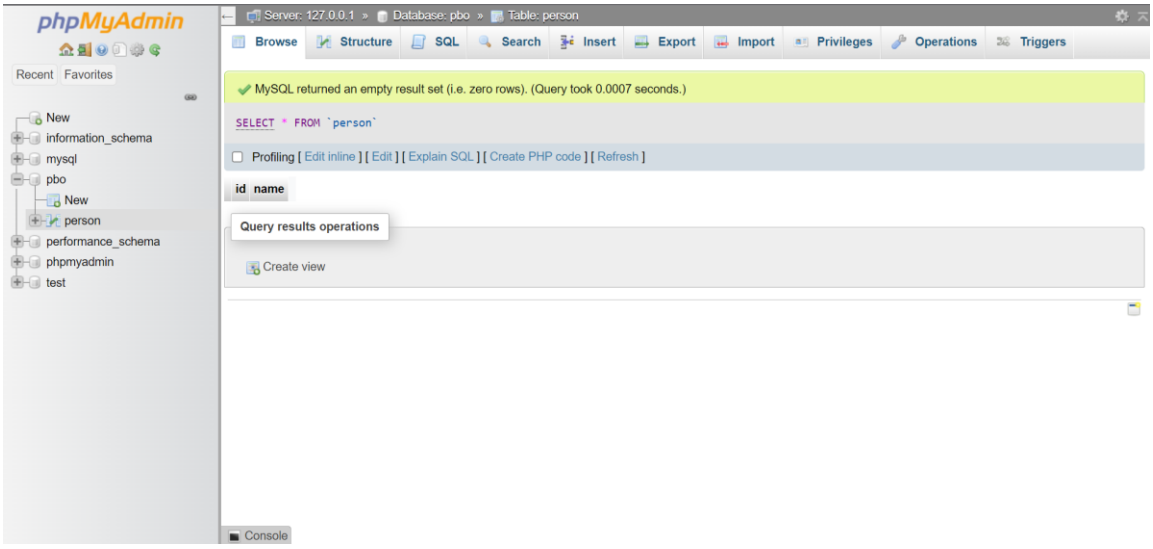
6. Buat database dengan nama ‘pbo’



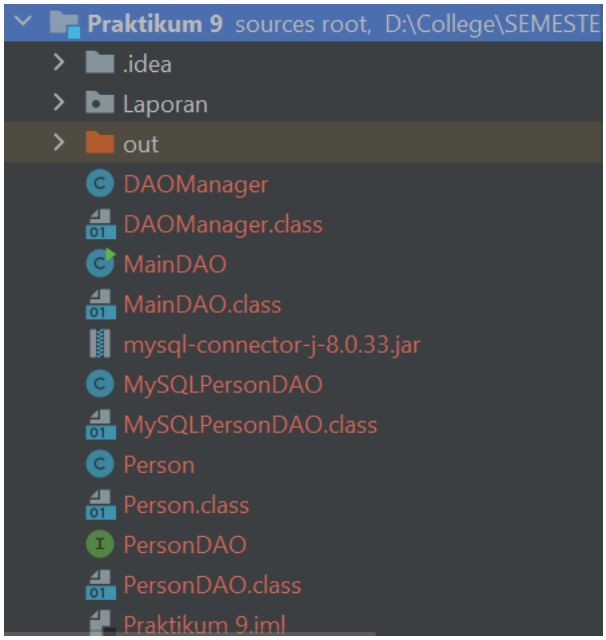
Buat tabel Person dengan query yang ada di modul:



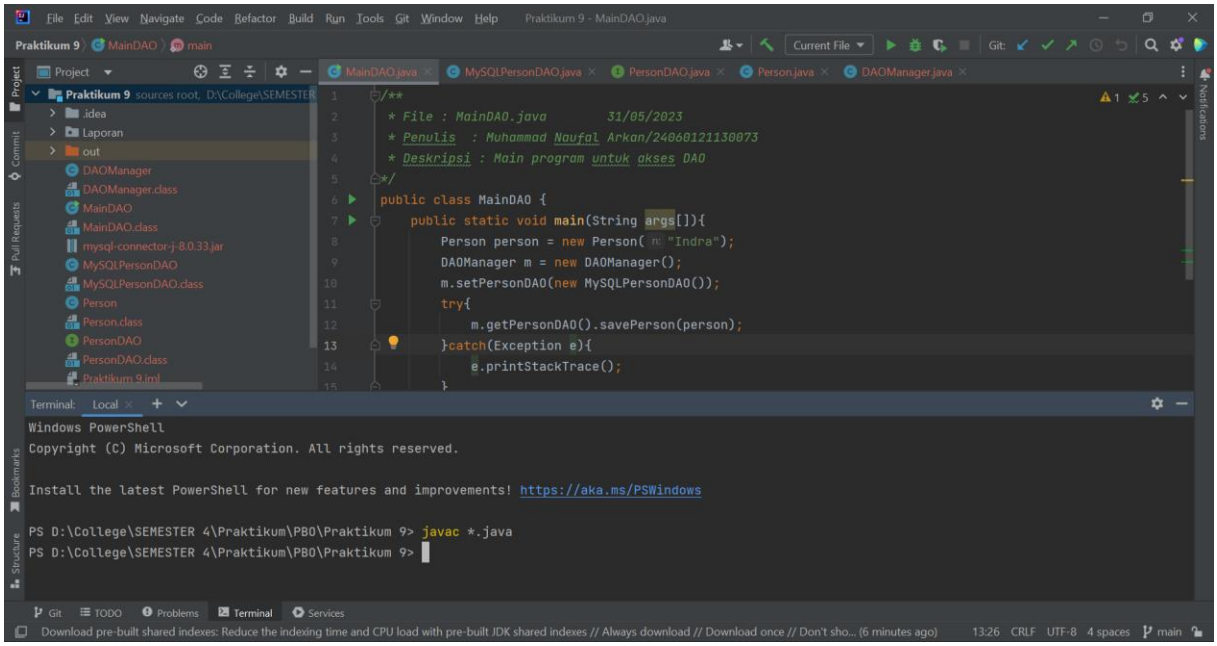
Tabel Person masih kosong.



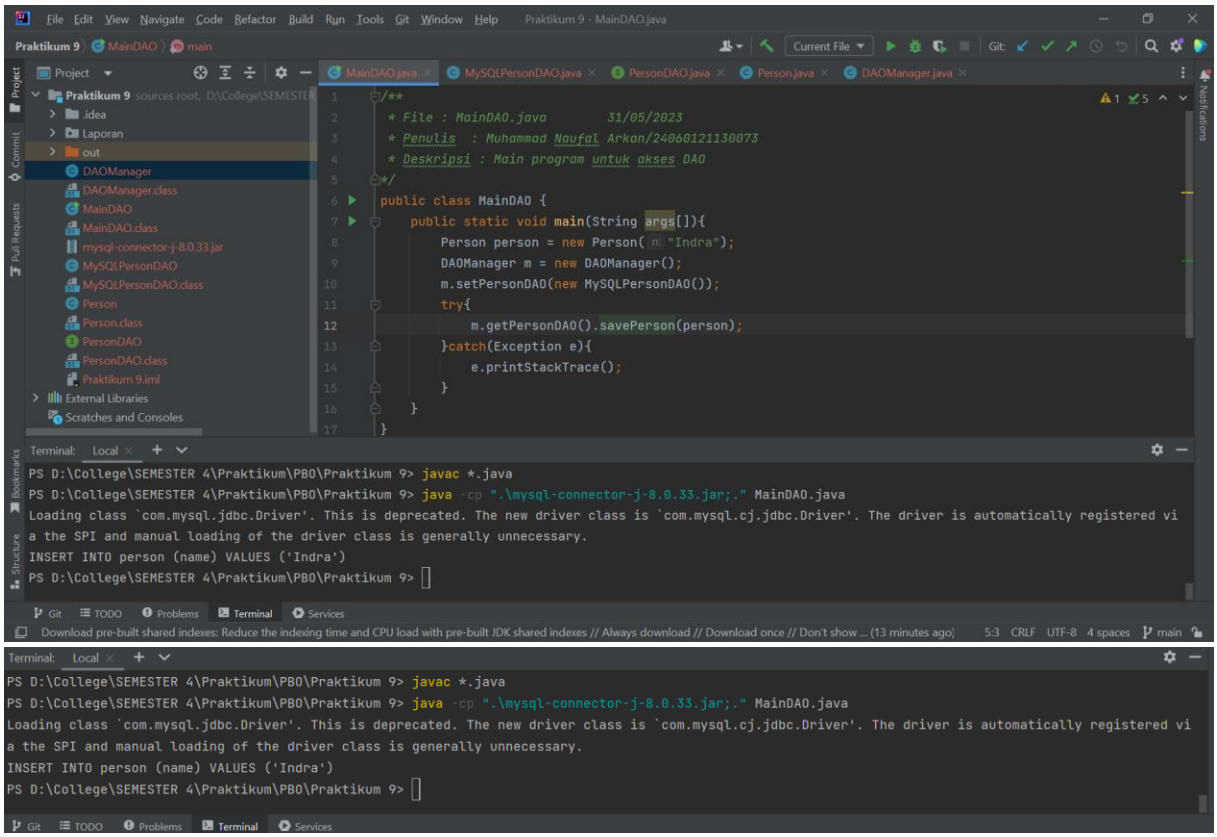
7. MySQL driver diletakkan satu direktori dengan source code program



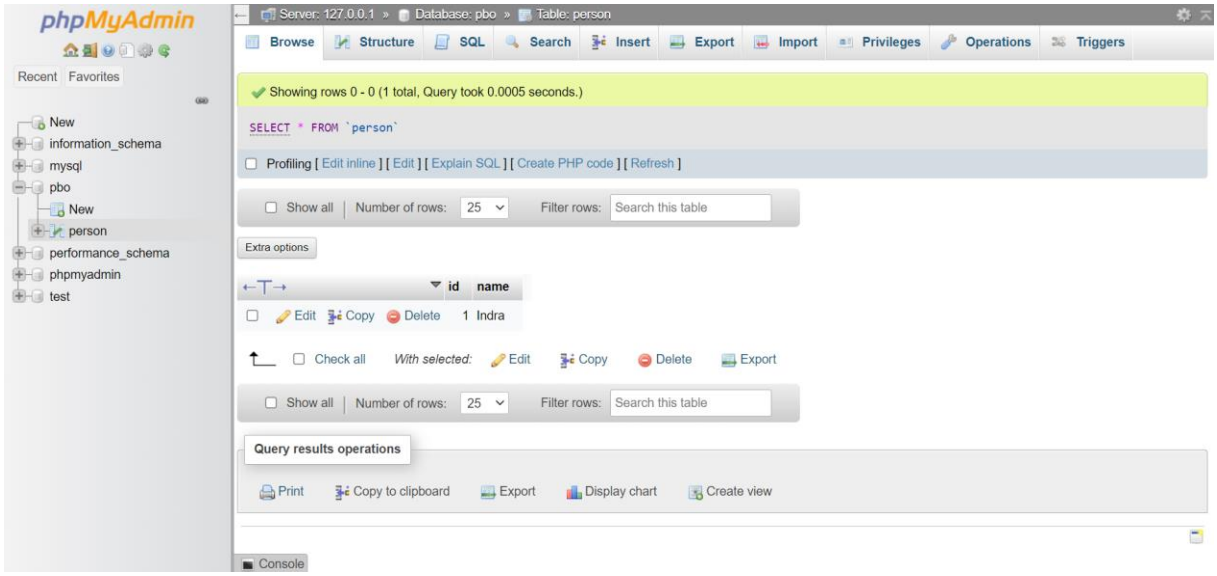
8. Kompilasi semua source code

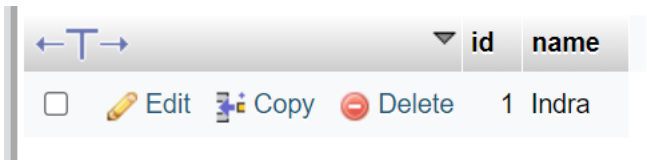


9. Jalankan MainDAO



10. Terjadi penambahan record pada tabel



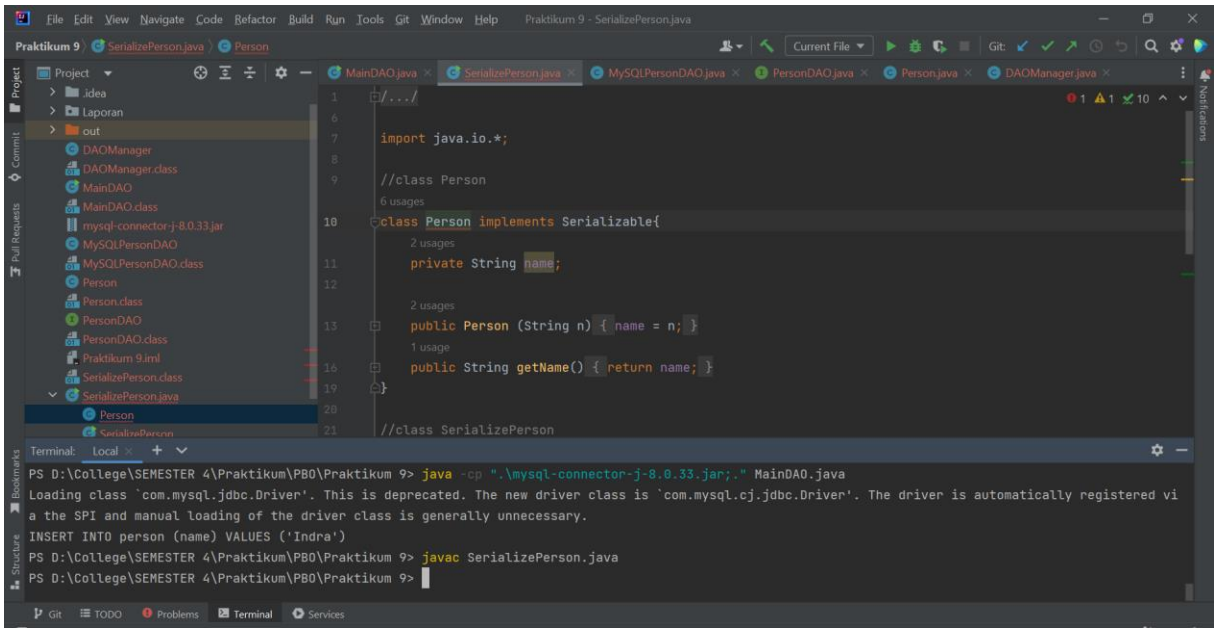


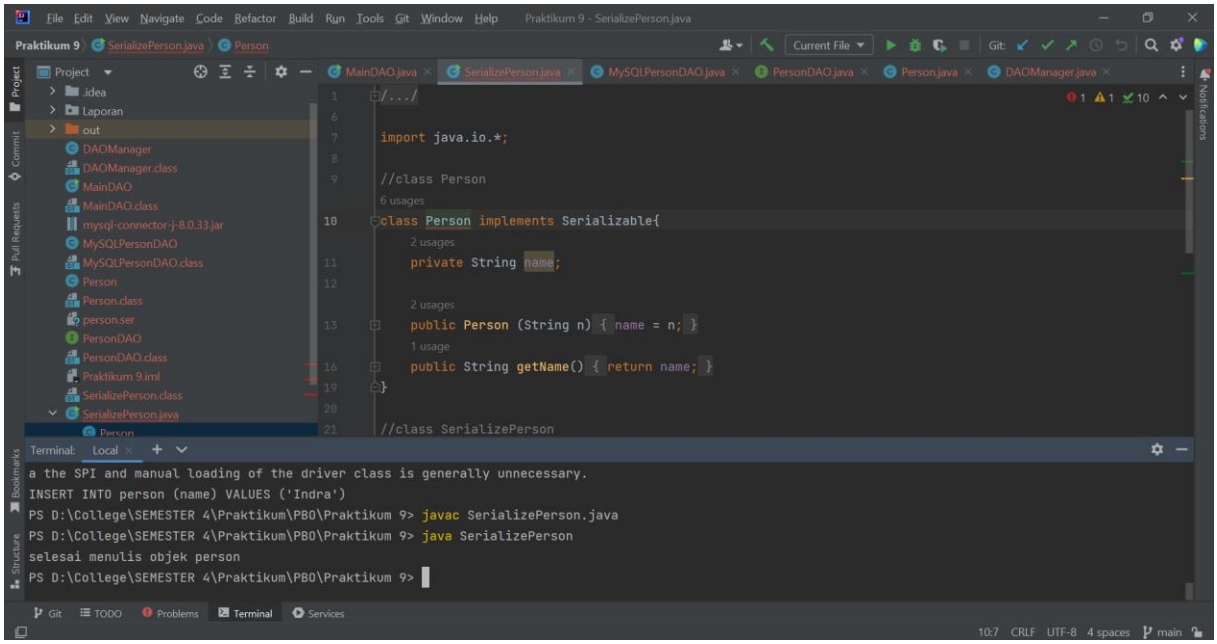
========== 3.2 ==========

1. Class SerializePerson.java

```
/**
 * File : SerializePerson.java        31/05/2023
 * Penulis  : Muhammad Naufal Arkan/24060121130073
 * 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





3. Class ReadSerializedPerson.java

```
/**
 * File : ReadSerializedPerson.java        31/05/2023
 * Penulis  : Muhammad Naufal Arkan/24060121130073
 * 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