

LAPORAN PRAKTIKUM

PRAKTIKUM 9:

“Persistent Object”



Disusun Oleh:

Victorio Cristiansa Putra

24060121130084

PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

LAB B2

DEPARTEMEN ILMU KOMPUTER / INFORMATIKA

FAKULTAS SAINS DAN MATEMATIKA

UNIVERSITAS DIPONEGORO

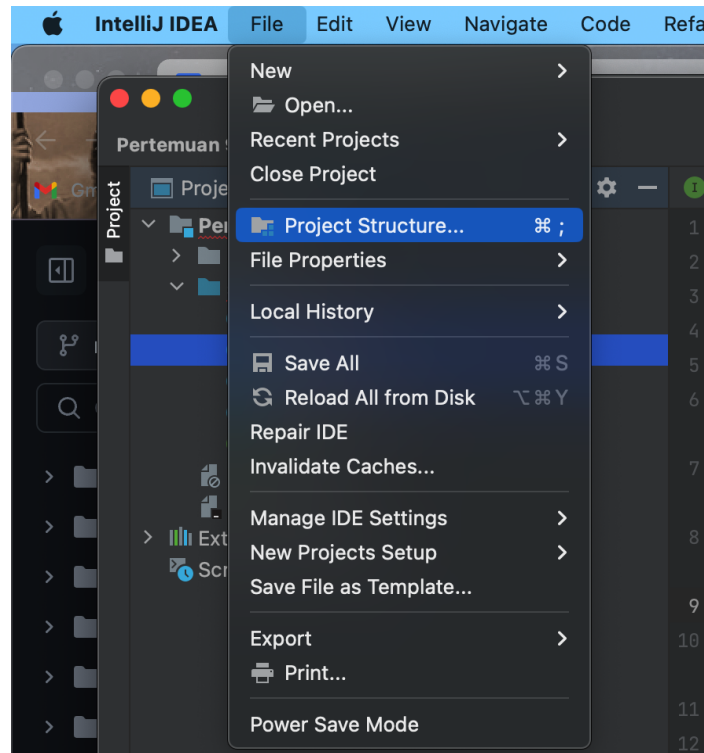
SEMARANG

2023

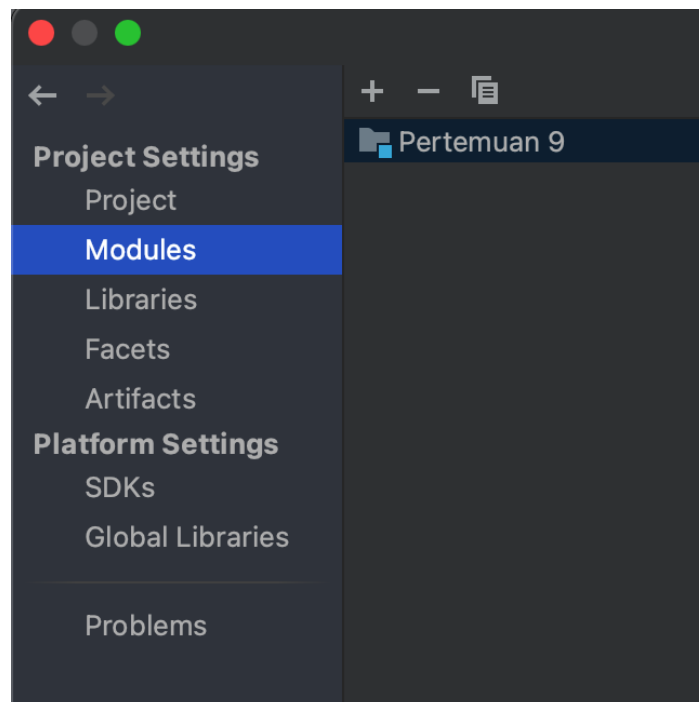
1. Persistent Objek Sebagai Model Basis Data Relasional

Praktikum ini akan membahas mengenai implementasi kelas-kelas pada java yang dapat terhubung dengan basis data. Dengan menghubungkan IDE yang kita gunakan dalam hal ini IntelliJ IDE dengan modul SQL connector. Berikut ini langkah-langkah untuk menghubungkannya :

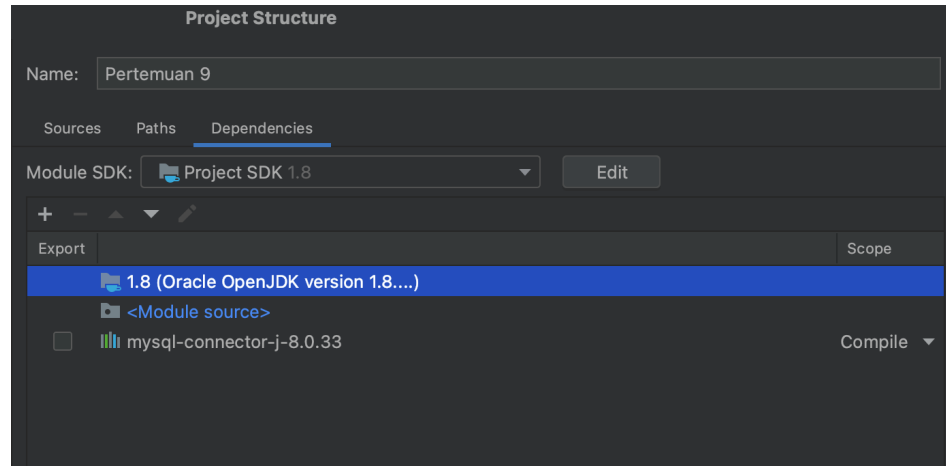
- Membuka File kemudian menuju ke “Project Structure”



- Memilih opsi modul pada Project Settings



- Memilih tab Dependencies lalu klik tambah untuk menambahkan file SQL conector



Kode program kelas-kelas java :

a. PersonDAO.java

Interface untuk person access object.

```
/**
 * Penulis : Victorio Cristiansa Putra 31/05/2023
 * File : PersonDAO.java
 * Deskripsi : interface untuk person access object
 */

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

b. Person.java

Kelas ini berisi konstruktor-konstruktor person dan berisi *method* getter yaitu getId() dan getName().

```
/**
 * Penulis : Victorio Cristiansa Putra 31/05/2023
 * File : PersonDAO.java
 * 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;
    }
}

```

c. MySQLPersonDAO.java

Kelas ini kita akan mengatur koneksi antara java dengan basis data pada line
`Connection con = DriverManager.getConnection`
`("jdbc:mysql://localhost/pbo", "root", "");`
 Kita dapat memilih schema dan memasukkan username dan password untuk
 basis data. Pada kelas ini, kita dapat membuat query dan mengeksekusinya
 untuk modifikasi pada tabel basis data.

```

/**
 * Penulis : Victorio Cristiansa Putra 31/05/2023
 * File : MySQLPersonDAO.java
 * Deskripsi : implementasi PersonDAO untuk MySQL
 */

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;

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","Victoriocp29")
        ;
        //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();
    }
}

```

d. DAOManager.java

Pengelola DAO dalam program.

```

/**
 * Penulis : Victorio Cristiansa Putra 31/05/2023
 * File : DAOManager.java
 * Deskripsi : pengelola DAO dalam program
 */

public class DAOManager {
    private PersonDAO personDAO;

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

    public PersonDAO getPersonDAO(){
        return personDAO;
    }
}

```

e. MainDAO.java

Main program untuk akses DAO.

```

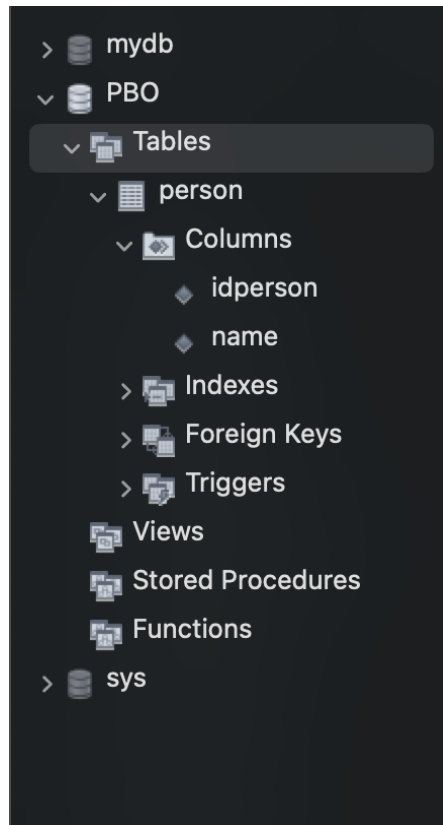
/**
 * Penulis : Victorio Cristiansa Putra 31/05/2023
 * File : MainDAO.java
 * Deskripsi : Main program untuk akses DAO
 */

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

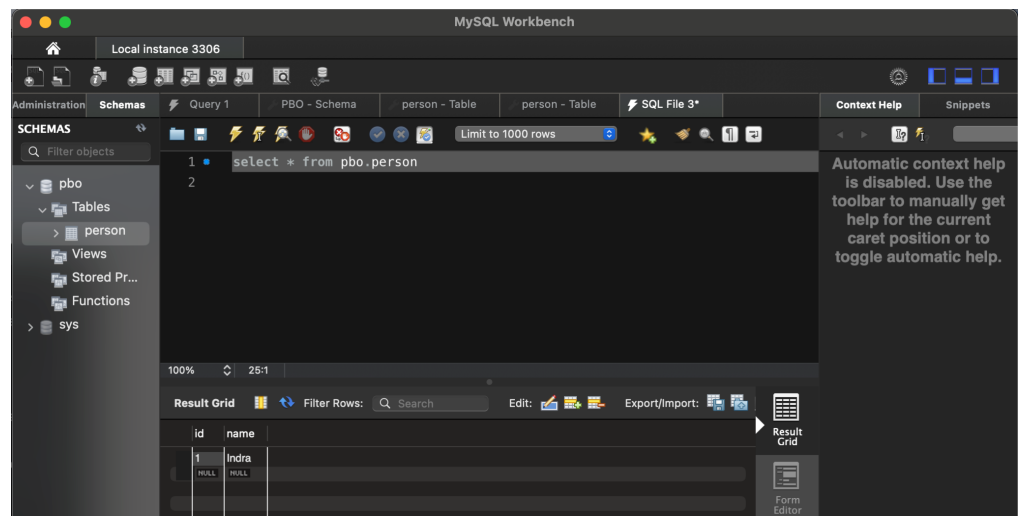
```

Kemudian masuk ke MySQL dan membuat schema dengan nama “PBO” dan membuat tabel dengan query :

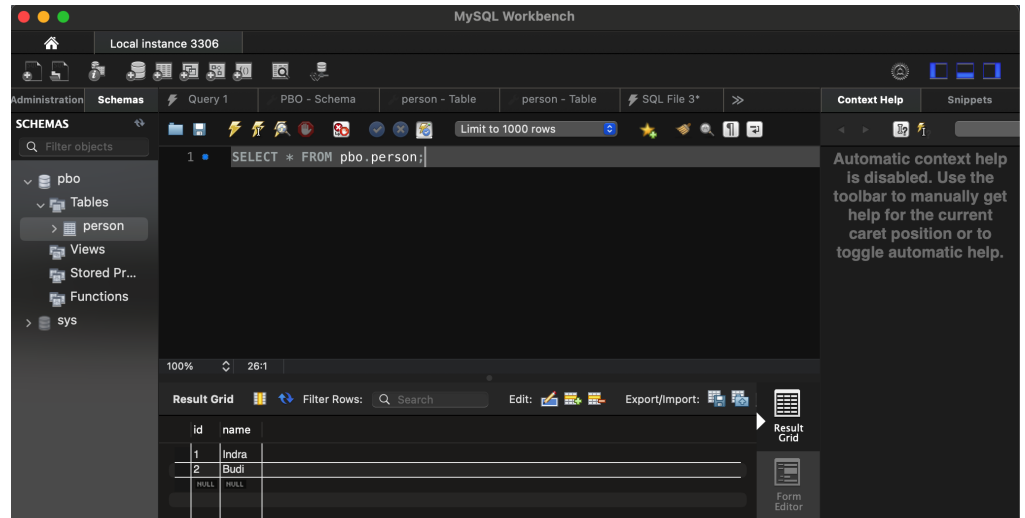
```
CREATE TABLE person(id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,name VARCHAR(100))
```



- Perbedaan tabel ketika sebelum dan sesudah program java dieksekusi :
 - a. Kondisi awal tabel



- b. Kondisi setelah *run* MainDAO.java, isi tabel telah dilakukan perubahan



2. Menggunakan Persistent Object sebagai objek terealisasi
a. SerializePerson.java

```
/**
 * Penulis : Victorio Cristiansa Putra 31/05/2023
 * File : SerializePerson.java
 * 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();
    }
}
}

```

Hasil :

Selesai menulis objek person

b. ReadSerializedPerson.java

```

/**
 * Penulis : Victorio Cristiansa Putra 31/05/2023
 * File : ReadSerializedPerson.java
 * 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 :

serialized person name = Panji