

**LAPORAN PRAKTIKUM
PEMOGRAMAN BERORIENTASI OBJEK
PRAKTIKUM 9 : Persistent Object**



**Nama : Muhammad Alif Mukhlis
NIM : 24060121120001
LAB : B1**

**PROGRAM STUDI S-1 INFORMATIKA
DEPARTEMEN INFORMATIKAFAKULTAS SAINS
DAN MATEMATIKA UNIVERSITAS
DIPONEGORO
SEMARANG
2023**

A. Menggunakan Persistent Object Sebagai Model Basis Data Relasional

1. PersonDAO.java

```
/*
Nama file: PersonDAO.java
Deskripsi: interface PersonDAO
pembuat: Muhammad Alif Mukhlis - 24060121120001
Tgl Pembuatan: 6 Juni 2023
*/
public interface PersonDAO{
    public void savePerson(Person p) throws Exception;
}
```

2. Person.java

```
/*
Nama file: Personjava
Deskripsi: class Person definisi kolom person sql
pembuat: Muhammad Alif Mukhlis - 24060121120001
Tgl Pembuatan: 6 Juni 2023
*/

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

```
/*
Nama file: MySQLPersonDAO.java
Deskripsi: class MySQLPersonDAO untuk menyambungkan ke
sql dan insert table
pembuat: Muhammad Alif Mukhlis - 24060121120001
Tgl Pembuatan: 6 Juni 2023
*/

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","m4suk4j4");

        String query = "INSERT INTO person(name) VALUES
('"+name+"')";
        System.out.println(query);
        Statement s = con.createStatement();
        s.executeUpdate(query);

        con.close();
    }
}

```

4. DAOManajer.java

```

/*
Nama file: DAOManager.java
Deskripsi: class DAOManager
pembuat: Muhammad Alif Mukhlis - 24060121120001
Tgl Pembuatan: 6 Juni 2023
*/
public class DAOManager{
    private PersonDAO personDAO;

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

```

5. MainDAO.java

```

/*
Nama file: MainDAO.java
Deskripsi: class main MainDAO
pembuat: Muhammad Alif Mukhlis - 24060121120001
Tgl Pembuatan: 6 Juni 2023
*/

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. Membuat Database “pbo”

```

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

```

7. Compile dan Run Source Code

Untuk compile semua src bisa menggunakan command javac *.java pada cli.
Selanjutnya run MainDAO beserta injector yang telah download.

```

PS C:\Users\User\Documents\prak9> c:\; cd 'C:\Users\User\Documents\prak9'; & "C:\Program Files\Java\jdk-19\bin\java.exe" -gc:\Users\User\AppData\Local\Temp\cp_4
2f8b6a2xhobuvm218qoxs-argfile' '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 SP
I and manual loading of the driver class is generally unnecessary.
INSERT INTO person(name) VALUES ('Indra')
PS C:\Users\User\Documents\prak9>

```

Screenshot diatas merupakan hasil run src dan input person “Indra” ke database pbo.

8. Check table Person

```

mysql> prompt alifMukhlis-24060121120001>
PROMPT set to 'alifMukhlis-24060121120001>'
alifMukhlis-24060121120001>use pbo;
Database changed
alifMukhlis-24060121120001>select * from person;
+----+-----+
| id | name |
+----+-----+
|  1 | Indra |
+----+-----+
1 row in set (0.00 sec)

alifMukhlis-24060121120001>|

```

Tampak pada screenshot diatas person “Indra telah masuk ke table person di database pbo.

B. Menggunakan Persistent Object Sebagai Objek Terealisasi

1. SerializePerson.java

```
/*
Nama file: SerializePerson.java
Deskripsi: class SerializePerson untuk membuat
person.rer
pembuat: Muhammad Alif Mukhlis - 24060121120001
Tgl Pembuatan: 6 Juni 2023
*/

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("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 Run SerializePerson.java

```
PS C:\Users\User\Documents\prak9> javac .\SerializePerson.java
PS C:\Users\User\Documents\prak9> java SerializePerson
selesai menulis objek person
PS C:\Users\User\Documents\prak9> █
```

Screenshote diatas menampilkan proses compile dan run SerializePerson. Setelah compile akan terbuat file person.ser

3. ReadSerializedPerson.java

```

/*
Nama file: SerializePerson.java
Deskripsi: class SerializePerson untuk membuat
person.rer
pembuat: Muhammad Alif Mukhlis - 24060121120001
Tgl Pembuatan: 6 Juni 2023
*/

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();
            System.out.println("serialized person name =
"+person.getName());
        }catch(Exception ioe){
            ioe.printStackTrace();
        }
    }
}

```

4. Compile dan Run ReadSerializePerson.java

```

PS C:\Users\User\Documents\prak9> javac .\ReadSerializedPerson.java
PS C:\Users\User\Documents\prak9> java ReadSerializedPerson
serialized person name = Panji

```

Screenshot diatas menampilkan proses compile dan run ReadSerializePerson.java. setelah berhasil run panji akan masuk ke list person.