

**LAPORAN PRAKTIKUM**  
**PRAKTIKUM 9:**

**“PERSISTENT OBJECT”**



**Disusun Oleh :**  
**Handhika Pranata Kusuma Wardana**  
**24060121140112**

**PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK**  
**LAB B1**

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

## A. Menggunakan Persistent Object sebagai model basis data relasional

Program Java :

### 1. PersonDAO.java

6

### 2. Person.java

```
/**
 * File      : Person.java 06/06/2023
 * Penulis   : Fadhail Athaillah Bima - 24060121140172
 * 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;
    }
}
```

### 3. MySQLPersonDAO.java

```
/**
 * File      : MySQLPersonDAO.java 06/06/2023
 * Penulis   : Fadhail Athaillah Bima - 24060121140172
 * Deskripsi  : Implementasi PersonDAO untuk MySQL
 * Lab       : B1
 */

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. DAOManager.java

```

/**
 * File      : DAOManager.java 06/06/2023
 * Penulis   : Fadhail Athaillah Bima - 24060121140172
 * 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;
    }
}

```

#### 5. MainDAO.java

```

/**
 * File      : MainDAO.java 06/06/2023
 * Penulis   : Fadhail Athaillah Bima - 24060121140172
 * 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();
        }
    }
}

```

## Hasil run program Java

```
E:\PBO\Prak PBO\Praktikum 9>javac MainDAO.java

E:\PBO\Prak PBO\Praktikum 9>javac MySQLPersonDAO.java

E:\PBO\Prak PBO\Praktikum 9>java -classpath .\mysql-connector-j-8.0.31.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')
```

## Tampilan tabel person pada database pbo pada MySQL CLI

```
mysql> use pbo;
Database changed
mysql> select * from person;
+----+-----+
| id | name |
+----+-----+
|  1 | Indra |
+----+-----+
1 row in set (0.00 sec)
```

## B. Menggunakan Persistent Object sebagai model basis data relasional

### SerializePerson.java

```
/**
 * File      : SerializePerson.java 06/06/2023
 * Penulis   : Fadhail Athaillah Bima - 24060121140172
 * 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 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 run program

```

E:\PBO\Prak PBO\Praktikum 9>javac SerializePerson.java

E:\PBO\Prak PBO\Praktikum 9>java SerializePerson
selesai menulis objek person

E:\PBO\Prak PBO\Praktikum 9>|

```

ReadSerializedPerson.java

```

/**
 * File      : ReadSerializedPerson.java 06/06/2023
 * Penulis   : Fadhail Athaillah Bima - 24060121140172
 * 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();
        }
    }
}

```

Hasil run program

```

E:\PBO\Prak PBO\Praktikum 9>javac ReadSerializedPerson.java

E:\PBO\Prak PBO\Praktikum 9>java ReadSerializedPerson
serialized person name =Panji

E:\PBO\Prak PBO\Praktikum 9>|

```