

**LAPORAN PRAKTIKUM**  
**PRAKTIKUM 9:**  
**“PERSISTENT OBJECT”**



**Disusun Oleh :**

Thifa Ziada Taqiyya  
24060121130080

**PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK**  
**LAB B1**

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

## A. Menggunakan Persistensi Object sebagai Model Basis Data Relasional

### 1. PersonDAO.java

```
/**
 * Penulis : Thifa Ziada Taqiyya 31/05/2023
 * File : PersonDAO.java
 * Deskripsi : interface untuk person access object
 */

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

### 2. Person.java

```
/**
 * Penulis : Thifa Ziada Taqiyya 31/05/2023
 * File : Person.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;
    }
}
```

### 3. MySQLPersonDAO.java

```
/**
 * Penulis : Thifa Ziada Taqiyya 31/05/2023
 * File : MySQLPersonDAO.java
 * Deskripsi : implementasi 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","taqiyya13");
        //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

```

/**
 * Penulis : Thifa Ziada Taqiyya 31/05/2023
 * File : DAOManager.java
 * Deskripsi : pengelola DAO dalam program
 */

public class DAOManager{
    private PersonDAO personDAO;

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

```

#### 5. MainDAO.java

```

/**
 * Penulis : Thifa Ziada Taqiyya 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 ("Thifa");
        DAOManager m = new DAOManager();
        m.setPersonDAO (new MySQLPersonDAO());
        try{
            m.getPersonDAO().SavePerson(person);
        }catch(Exception e){
            e.printStackTrace();
        }
    }
}

```

```
}  
}  
}
```

6. Membuat database pbo dengan tabel id dan name :

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

```
mysql> CREATE DATABASE pbo;  
Query OK, 1 row affected (0.10 sec)  
  
mysql> use pbo;  
Database changed  
mysql> CREATE TABLE person(  
-> id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,  
-> name VARCHAR(100));  
Query OK, 0 rows affected (0.15 sec)
```

7. Kompilasi semua source code dengan perintah : javac \*.java

```
D:\KULIAH\SEMESTER 4\Pemrograman Berorientasi Objek\Praktikum 9>javac *.java  
D:\KULIAH\SEMESTER 4\Pemrograman Berorientasi Objek\Praktikum 9>|
```

8. Jalankan MainDAO dengan perintah : java -classpath .\mysql-connector-java-[versi].jar;. MainDAO

```
D:\KULIAH\SEMESTER 4\Pemrograman Berorientasi Objek\Praktikum 9>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 ('Thifa')
```

Hasil :

```
mysql> select * from person;  
+----+-----+  
| id | name |  
+----+-----+  
|  1 | Indra |  
|  2 | Thifa |  
+----+-----+  
2 rows in set (0.00 sec)
```

## B. Menggunakan Persistent Object sebagai objek terserialisas

1. SerializePerson.java

```

/**
 * Penulis : Thifa Ziada Taqiyya 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("Ziada");
        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();
        }
    }
}

```

```
D:\KULIAH\SEMESTER 4\Pemrograman Berorientasi Objek\Praktikum 9>javac SerializePerson.java
```

```
D:\KULIAH\SEMESTER 4\Pemrograman Berorientasi Objek\Praktikum 9>java SerializePerson
selesai menulis objek person
```

## 2. ReadSerializedPerson.java

```

/**
 * Penulis : Thifa Ziada Taqiyya 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();
        }
    }
}
```

D:\KULIAH\SEMESTER 4\Pemrograman Berorientasi Objek\Praktikum 9>javac ReadSerializedPerson.java

D:\KULIAH\SEMESTER 4\Pemrograman Berorientasi Objek\Praktikum 9>java ReadSerializedPerson  
serialized person name = Ziada