

# **LAPORAN PRAKTIKUM PBO 9**



**Disusun Oleh :**

**Faizal Husain Adiasha 24060121140035**

**PRAKTIKUM PBO  
LAB B2**

**DEPARTEMEN ILMU KOMPUTER / INFORMATIKA  
FAKULTAS SAINS DAN MATEMATIKA  
UNIVERSITAS DIPONEGORO  
SEMARANG  
2022**

## MySQL

### A. Kelas PersonDAO

```
// personDAO.java 6/6/23
// Penulis : Faizal Husain Adiasha
// NIM : 24060121140115
// Deskripsi: Interface untuk person access object
public interface PersonDAO {
    public void savePerson(Person p) throws Exception;
}
```

### B. Kelas Person

```
// Person.java 6/6/23
// Penulis : Faizal Husain Adiasha
// NIM : 24060121140115
// Deskripsi: Kelas Person Database Model
public class Person {
    private int id;
    private String name;

    public Person(String n) {
        this.name = n;
    }

    public Person(int i, String n) {
        this.id = i;
        this.name = n;
    }

    public int getId() {
        return this.id;
    }

    public String getName() {
        return this.name;
    }
}
```

### C. Kelas MySQLPersonDAO

```
// MySQLPersonDAO.java 6/6/23
// Penulis : Faizal Husain Adiasha
```

```
// NIM      : 24060121140115
// Deskripsi: Implementasi PersonDao untuk MySQL
import java.sql.*;
public class MySQLPersonDao implements PersonDAO {
    public void savePerson(Person person) throws Exception {
        String name = person.getName();
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con =
            DriverManager.getConnection("jdbc:mysql://localhost:3310/Person"
                , "root", "");
        String query = "INSERT INTO Person (name) VALUES ('" + name +
            "')";
        Statement s = con.createStatement();
        s.executeUpdate(query);
        s.close();
    }
}
```

#### D. Kelas DAOManager.java

```
// DAOManager.java 6/6/23
// Penulis  : Faizal Husain Adiasha
// NIM      : 24060121140115
// Deskripsi: Implementasi PersonDao untuk MySQL
public class DAOManager {
    private PersonDAO personDAO;

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

#### E. Kelas MainDAO

```
// MainDAO.java 6/6/23
// Penulis  : Faizal Husain Adiasha
// NIM      : 24060121140115
// Deskripsi: Kelas Person Database Model
public class MainDAO {
    public static void main(String[] args) {
        Person person = new Person("Faizal Husain Adiasha");
        DAOManager m = new DAOManager();
        m.setPersonDAO(new MySQLPersonDao());
        try {
            m.getPersonDAO().savePerson(person);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```



## Serialized

### A. SerializePerson

```
// SerializePerson.java 6/6/23
// Penulis : Faizal Husain Adiasha
// NIM : 24060121140115
// Deskripsi: Program Untuk Serialisasi Objek Person
import java.io.*;
public 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("Faizal Husain Adiasha");
        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 i) {
            i.printStackTrace();
        }
    }
}
```

### B. ReadSerializedPerson

```
// ReadSerializedPerson.java 6/6/23
// Penulis : Faizal Husain Adiasha
// NIM : 24060121140115
// 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();
        }
    }
}
```

```
    }  
}  
}
```

### C. Mencoba Serialize person

```
PS D:\PBO\praktikum-pbo\Praktikum 9\Persistent> javac ReadSerializedPerson.java  
PS D:\PBO\praktikum-pbo\Praktikum 9\Persistent> javac ReadSerializedPerson.java  
PS D:\PBO\praktikum-pbo\Praktikum 9\Persistent> java ReadSerializedPerson  
Serialized person name =Faizal Husain Adiasha  
PS D:\PBO\praktikum-pbo\Praktikum 9\Persistent>  
* History restored  
PS D:\PBO\praktikum-pbo\Praktikum 9\Persistent> java ReadSerializedPerson  
Serialized person name =Faizal Husain Adiasha  
PS D:\PBO\praktikum-pbo\Praktikum 9\Persistent> |
```