

PRAKTIKUM 9
PEMROGRAMAN BERORIENTASI OBJEK
LAPORAN



DISUSUN OLEH :
BONIFASIUS BRYAN SURYANINGTYAS
(24060121130066)

ASISTEN PRAKTIKUM :
1. M. DRAJAT DANY DARMAWAN
2. AGUNG RAMADHANI

PROGRAM STUDI INFORMATIKA LAB C1
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO
SEMARANG
2023

A. MENGGUNAKAN *PERSISTENT OBJECT* SEBAGAI MODEL BASIS DATA

1. Kelas *Interface* PersonDAO.java

```
// Nama : Bonifasius Bryan Suryaningtyas
// NIM : 24060121130066
// Pertemuan 9 Pemrograman Berorientasi Objek Lab C1
// Hari, tanggal : Rabu, 30 Mei 2023
// Nama File : PersonDAO.java
// Deskripsi : Interface untuk Person Access Object

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

2. Kelas Person.java

```
// Nama : Bonifasius Bryan Suryaningtyas
// NIM : 24060121130066
// Pertemuan 9 Pemrograman Berorientasi Objek Lab C1
// Hari, tanggal : Rabu, 30 Mei 2023
// Nama 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;
    }
}
```

3. Kelas MySQLPersonDAO.java

```
import java.sql.*;

// Nama : Bonifasius Bryan Suryaningtyas
// NIM : 24060121130066
// Pertemuan 9 Pemrograman Berorientasi Objek Lab C1
// Hari, tanggal : Rabu, 30 Mei 2023
// Nama File : MySQLPersonDAO.java
// Deskripsi : Implementasi PersonDAO untuk MySQL
```

```

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",
"waitforit");

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

        con.close();
    }
}

```

4. Kelas DAOManager.java

```

// Nama : Bonifasius Bryan Suryaningtyas
// NIM : 24060121130066
// Pertemuan 9 Pemrograman Berorientasi Objek Lab C1
// Hari, tanggal : Rabu, 30 Mei 2023
// Nama 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. Kelas MainDAO.java

```

// Nama : Bonifasius Bryan Suryaningtyas
// NIM : 24060121130066
// Pertemuan 9 Pemrograman Berorientasi Objek Lab C1
// Har tanggal : Rabu, 30 Mei 2023
// Nama File : MainDAO.java
// Deskripsi : Main Program untuk Akses DAO

public class MainDAO {
    public static void main(String[] args) {
        Person person = new Person("Bryan");
        DAOManager m = new DAOManager();
        m.setPersonDAO(new MySQLPersonDAO());
        try {

```

```

        m.getPersonDAO().savePerson(person);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

6. Buat Database “PBO”

```

mysql> prompt bryan>>
PROMPT set to 'bryan>>'
bryan>>prompt bryan >
PROMPT set to 'bryan >'
bryan >show databases;
+-----+
| Database |
+-----+
| corporate |
| information_schema |
| inventory |
| mbdc_24060121130066 |
| mysql |
| organisasi |
| performance_schema |
| perpustakaan |
| prak8 |
| sys |
+-----+
10 rows in set (0.08 sec)

bryan >CREATE DATABASE pbo;
Query OK, 1 row affected (0.01 sec)

bryan >use pbo;
Database changed
bryan >CREATE TABLE person (
    -> id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
    -> name VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.03 sec)

bryan >select * from person;
Empty set (0.02 sec)

```

7. Kompilasi Semua Source Code dengan Perintah :

```
javac *.java
```

8. Jalankan MainDAO dengan Perintah :

```
java -classpath .\mysql-connector-java-8.0.33.jar;. MainDAO
```

9. Cek Tabel Person pada Database “PBO”

```

bryan >SELECT * FROM PERSON;
+-----+-----+
| id | name |
+-----+-----+
| 1 | Bryan |
+-----+-----+
1 row in set (0.00 sec)

```

B. MENGGUNAKAN *PERSISTENT OBJECT* SEBAGAI OBJEK TEREALISASI

1. Kelas SerializePerson.java

```
// Nama : Bonifasius Bryan Suryaningtyas
// NIM      : 24060121130066
// Pertemuan 9 Pemrograman Berorientasi Objek Lab C1
// Hari, tanggal : Rabu, 30 Mei 2023
// Nama File : SerializePerson.java
// Deskripsi : Program untuk Serialisasi Objek Person

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("Bryan");
        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\PersistentObjectTerealisasi>javac SerializePerson.java
D:\Kuliah\Semester 4\Pemrograman Berorientasi Objek\Praktikum 9\PersistentObjectTerealisasi>java SerializePerson
Selesai Menulis Objek Person
```

2. Kelas ReadSerializedPerson.java

```
// Nama : Bonifasius Bryan Suryaningtyas
// NIM      : 24060121130066
// Pertemuan 9 Pemrograman Berorientasi Objek Lab C1
// Hari, tanggal : Rabu, 30 Mei 2023
// Nama 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("Serialisasi Person Name = " +
person.getName());
    } catch (Exception ioe) {
        ioe.printStackTrace();
    }
}
```

```
D:\Kuliah\Semester 4\Permograman Berorientasi Objek\Praktikum 9\PersistentObjectTerealisasi>javac ReadSerializedPerson.j
ava
D:\Kuliah\Semester 4\Permograman Berorientasi Objek\Praktikum 9\PersistentObjectTerealisasi>java ReadSerializedPerson
Serialisasi Person Name = Bryan
```