

**LAPORAN PRAKTIKUM
PEMOGRAMAN BERORIENTASI OBJEK
PRAKTIKUM 9**



**Nama : Nanda Farrel Chevalerie
NIM : 24060121140165**

**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
// Nama : Nanda Farrel Chevalerie
// NIM : 24060121140165
// Deskripsi : Interface untuk person access object
// Tgl Pembuatan : 6 Juni 2023

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

2. Person.java

```
// Nama file : Personjava
// Nama : Nanda Farrel Chevalerie
// NIM : 24060121140165
// Deskripsi : Person database model
// 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
// Nama : Nanda Farrel Chevalerie
// NIM : 24060121140165
// Deskripsi : Implementasi PersonDAO untuk MySQL
// Tgl Pembuatan : 6 Juni 2023

import java.sql.*;
```

4. DAOManajer.java

```
// Nama file : DAOManager.java
// Nama : Nanda Farrel Chevalerie
// NIM : 24060121140165
// Deskripsi : Pengelola DAO dalam program
// 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
// Nama : Nanda Farrel Chevalerie
// NIM : 24060121140165
// Deskripsi : Main program untuk akses DAO
// 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 D:\Farrel> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '@C:\Users\User\AppData\Local\Temp\cp_5b9b82muitq3hhtqxSquvygey.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 ('Farrel')
PS D:\Farrel>
```

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

8. Check table Person

```
mysql> select * from person
-> ;
Empty set (0.00 sec)

mysql> select * from person;
+----+-----+
| id | name  |
+----+-----+
| 3  | Farrel |
+----+-----+
1 row in set (0.00 sec)
```

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\
// Nama : Nanda Farrel Chevalerie
// NIM : 24060121140165
// Deskripsi : Program untuk serialisasi objek Person
// 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 D:\Farrel> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '@C:\Users\User\AppData\Local\Temp\cp_5b9b82mu1q3hh1tqx5quvygey.argfile' 'SerializePerson'
selesai menulis objek person
PS D:\Farrel>

```

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

3. ReadSerializedPerson.java

```

// Nama file : SerializePerson.java
// Nama : Nanda Farrel Chevalerie
// NIM : 24060121140165
// Deskripsi : Program untuk serialisasi objek Person
// 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 D:\Farrel> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '@C:\Users\User\AppData\Local\Temp\cp_5b9b82mu1q3hh1tqx5quvygey.argfile' 'ReadSerializedPerson'
serialized person name = Panji
PS D:\Farrel>

```

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