

LAPORAN PRAKTIKUM
PRAKTIKUM 9:
“PERSISTENT OBJECT”



Disusun Oleh :

Nafis Mufadhal
24060121120033

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
LAB B1

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

A. Menggunakan Persistent Object sebagai model basis data

1. PersonDAO.java

```
/*
 * Nama File      : personDAO.java
 * Penulis       : Nafis Mufadhal
 * NIM          : 24060121120033
 * Tanggal      : 31 Mei 2023
 * Deskripsi     : interface person
 */

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

2. Person.java

```
/*
 * Nama File      : Person.java
 * Penulis       : Nafis Mufadhal
 * NIM          : 24060121120033
 * Tanggal      : 31 Mei 2023
 * 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

```
/*  
* Nama File      : MySQLPersonDAO.java  
* Penulis       : Nafis Mufadhal  
* NIM           : 24060121120033  
* Tanggal      : 31 Mei 2023  
* Deskripsi     : Implementasi PersonDAO  
*/  
  
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"  
"  
,"1234567890");  
        //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

```
/*
 * Nama File      : DAOManager.java
 * Penulis       : Nafis Mufadhal
 * NIM           : 24060121120033
 * Tanggal       : 31 Mei 2023
 * Deskripsi      : DAO program
 */

public class DAOManager {
    private PersonDAO personDAO;

    public DAOManager(PersonDAO personDAO) {
        this.personDAO = personDAO;
    }

    public DAOManager() {
    }

    public void savePerson(Person p) throws Exception {
        personDAO.savePerson(p);
    }

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

    public PersonDAO getPersonDAO() {
        return personDAO;
    }
}
```

5. MainDAO.java

```
/*
 * Nama File      : MainDAO.java
 * Penulis       : Nafis Mufadhal
 * NIM           : 24060121120033
 * Tanggal       : 31 Mei 2023
 * Deskripsi      : Main Program akses DAO

```

```

*/

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. Create Database

```

Nafis> create database PBO
-> ;
Query OK, 1 row affected (0.14 sec)

```

Membuat database dengan nama PBO, yaitu membuat perintah Create database PBO; . Diberi nama PBO sesuai dengan nama praktikum, selanjutnya jika sudah muncul “Query OK, I row affected” maka database sudah terbuat.

Use Database

```

Nafis> use PBO;
Database changed

```

Selanjutnya setelah membuat database yaitu harus menggunakannya dengan perintah “USE PBO” yaitu supaya database bisa diakses untuk menambahkan table.

Membuat table

```

Nafis> CREATE TABLE person(id INT PRIMARY KEY AUTO_INCREMENT NOT
-> NULL,name VARCHAR(100))
-> ;
Query OK, 0 rows affected (0.12 sec)

```

Kemudian membuat table dengan perintah create table person(id int primary key auto_increment not null, name varchar(100)); digunakan untuk membuat tabel baru bernama "person" dalam database "pbo"

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

```
C:\Users\TUF Gaming\Desktop\PBO\Praktikum 9> javac *java
```

Setelah berhasil menjalankan perintah "`javac *.java`" untuk mengkompilasi semua file dengan ekstensi `.java` dalam direktori. Setiap file `.java` akan dikompilasi menjadi file `.class` yang sesuai.

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

```
C:\Users\TUF Gaming\Desktop\PBO\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('Indra')
```

9. Lihat apakah terjadi penambahan record pada tabel

```
Nafis> select* from person;
+----+-----+
| id | name  |
+----+-----+
|  1 | Indra  |
+----+-----+
1 row in set (0.01 sec)
```

B. Menggunakan Persistent Object sebagai objek terserialisasi

1. Kelas `SerializePerson.java` berikut, untuk menyimpan objek dalam file yang bernama "person.ser"

```
/*
 * Nama File      : SerializedPerson.java
 * Penulis       : Nafis Mufadhal
 * NIM           : 24060121120033
 * Tanggal       : 31 Mei 2023
 * Deskripsi      : Program 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("Nafis");
        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 jalankan program

```

C:\Users\TUF Gaming\Desktop\PB0\Praktikum 9> javac SerializePerson.java
C:\Users\TUF Gaming\Desktop\PB0\Praktikum 9> java SerializePerson
selesai menulis objek person

```

3. Kelas ReadSerializedPerson.java berikut untuk membaca objek yang telah terserialisasi

```

/*
* Nama File      : ReadSerializedPerson.java

```

```

* Penulis      : Nafis Mufadhal
* NIM          : 24060121120033
* Tanggal      : 31 Mei 2023
* Deskripsi     : Program 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();
        }
    }
}

```

4. Compile dan jalankan program

```

C:\Users\TUF Gaming\Desktop\PBO\Praktikum 9> javac ReadSerializedPerson.java

C:\Users\TUF Gaming\Desktop\PBO\Praktikum 9> java ReadSerializedPerson
serialized person name = Nafis

```