

LAPORAN PRAKTIKUM
PRAKTIKUM 9:
“PERSISTENT OBJECT”



Disusun Oleh :

Vanesya F. W.
24060121140133

PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK
LAB B2

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

A. Persistent Object sebagai model basis data relasional

1. Interface PersonDAO.java

```
/**
 * PersonDAO.java 31/05/2023
 * Penulis : Vanesya F. W.
 * Deskripsi : interface untuk person access object
 *
 */

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

2. Kelas Person.java

```
/**
 * Person.java 31/05/2023
 * Penulis : Vanesya F. W.
 * 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

```
/**
 * MySQLPersonDAO.java 31/05/2023
```

```

* Penulis : Vanesya F. W.
* Deskripsi : implementasi PersonDAO untuk MySQL
*
*/

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","");
        //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. Kelas DAOManager.java

```

/**
* Nama File : DAOManager.java 31/05/2023
* Penulis : Vanesya F. W.
* Deskripsi : Pegelola 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 File : MainDAO.java 31/05/2023
* Penulis : Vanesya F. W.
* Deskripsi : Main program untuk 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();
        }
    }
}

```

B. Persistent Object sebagai object terserialisasi

1. Kelas SerializePerson.java untuk menyimpan objek dalam file yang Bernama “person.ser”

```

/**
 * SerializePerson.java 31/05/2023
 * Penulis : Vanesya F. W.
 * 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("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. Kelas ReadSerializedPerson.java untuk membaca objek yang telah terserialisasi

```

/**
 * Nama File           : ReadSerializedPerson.java 31/05/2023
 * Penulis            : Vanesya F. W.
 * 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();
        }
    }
}

```