

LAPORAN PRAKTIKUM
PRAKTIKUM 9:
“PERSISTENT OBJECT”



Disusun Oleh :

Aditya Suryandaru
24060121140105

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
LAB B

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

A. Menggunakan *Persistent Object* sebagai Model Basis Data Relasional

1. PersonDAO.java

```
/**
 * File      : PersonDAO.java
 * Penulis   : Aditya Suryandaru - 24060121140105
 * Tanggal   : 30 Mei 2023
 * Deskripsi  : interface untuk person access object
 */
public interface PersonDAO{
    public void savePerson (Person p) throws Exception;
}
```

2. Person.java

```
/**
 * File      : Person.java
 * Penulis   : Aditya Suryandaru - 24060121140105
 * Tanggal    : 30 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

```
/**
 * File      : MySQLPersonDAO.java
 * Penulis   : Aditya Suryandaru - 24060121140105
 * Tanggal    : 30 Mei 2023
 * Deskripsi  : Implementasi PersonDAO untuk MySQL
 */

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",
        "");

        // 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

```
/**
 * File      : DAOManager.java
 * Penulis   : Aditya Suryandaru - 24060121140105
 * Tanggal    : 30 Mei 2023
 * Deskripsi  : Pengelola DAO dalam program
 */

public class DAOManager {
    private PersonDAO personDAO;

    public void setPersonDAO(PersonDAO person) {
        personDAO = person;
    }

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

5. MainDAO.java

```
/**
 * File      : MainDAO.java
 * Penulis   : Aditya Suryandaru - 24060121140105
 * Tanggal    : 30 Mei 2023
 * 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();
        }
    }
}
```

6. Buat database dengan nama 'pbo' dan tabel pada database tersebut

Create database

pbo

utf8mb4_general_ci

Create

1 CREATE TABLE person(id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,name VARCHAR(100))

Update query Submit query

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> person	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
1 table	Sum	0	InnoDB	utf8mb4_general_ci	16.0 KiB	0 B

7. Kompilasi semua source code dan jalankan MainDAO.java

Praktikum 9

sources root, D:\UNDIP\Praktikum\Pemrog...

out

DAOManager

DAOManager.class

MainDAO

MainDAO.class

mysql-connector-j-8.0.33.jar

MySQLPersonDAO

MySQLPersonDAO.class

Person

Person.class

PersonDAO

PersonDAO.class

Praktikum 9.iml

External Libraries

Scratches and Consoles

1 /**

2 * File : Person.java

3 * Penulis : Aditya Suryandaru - 24060121140105

4 * Tanggal : 30 Mei 2023

5 * Deskripsi : Person database model

6 */

7

8 4 usages

9 public class Person {

10 2 usages

11 private int id;

12 3 usages

13 private String name;

14

15 1 usage

16 public Person(String n) { name = n; }

17

18 no usages

Terminal: Local

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS D:\UNDIP\Praktikum\Pemrograman Berorientasi Objek\Praktikum 9> javac *.java

PS D:\UNDIP\Praktikum\Pemrograman Berorientasi Objek\Praktikum 9> java -cp ".\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')

PS D:\UNDIP\Praktikum\Pemrograman Berorientasi Objek\Praktikum 9>

Version Control

TODO

Problems

Terminal

Services

Microsoft Defender configuration: The IDE has detected Microsoft Defender with Real-Time Protection enabled. It might severely de... (2 minutes a






6:4

CRLF

UTF-8

4 spaces

8. Hasil Akhir

				id	name
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Indra

B. Menggunakan *Persistent Object* sebagai objek terserialisasi

1. SerializePerson.java

```
/**
 * File      : SerializePerson.java
 * Penulis   : Aditya Suryandaru - 24060121140105
 * Tanggal    : 30 Mei 2023
 * 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 o = new ObjectOutputStream(f);
            o.writeObject(person);
            System.out.println("Selesai menulis objek person");
            o.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

2. Compile dan jalankan program diatas

The screenshot shows the Visual Studio Code interface with the file `SerializePerson.java` open. The code defines a `Person` class implementing `Serializable` and a `SerializePerson` class with a `main` method that creates a `Person` object and serializes it to a file named `person.ser`.

The terminal output shows the command to compile and run the program:

```
PS D:\UNDIP\Praktikum\Pemrograman Berorientasi Objek\Praktikum 9.1> 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ASUS\AppData\Roaming\Code\User\workspaceStorage\aa1788cc834641de87a8536b3863a56b\redhat.java\jdt_ws\Praktikum 9.1_f0832519\bin' 'SerializePerson'
```

The output of the program is:

```
Selesai menulis objek person
```

3. ReadSerializePerson.java

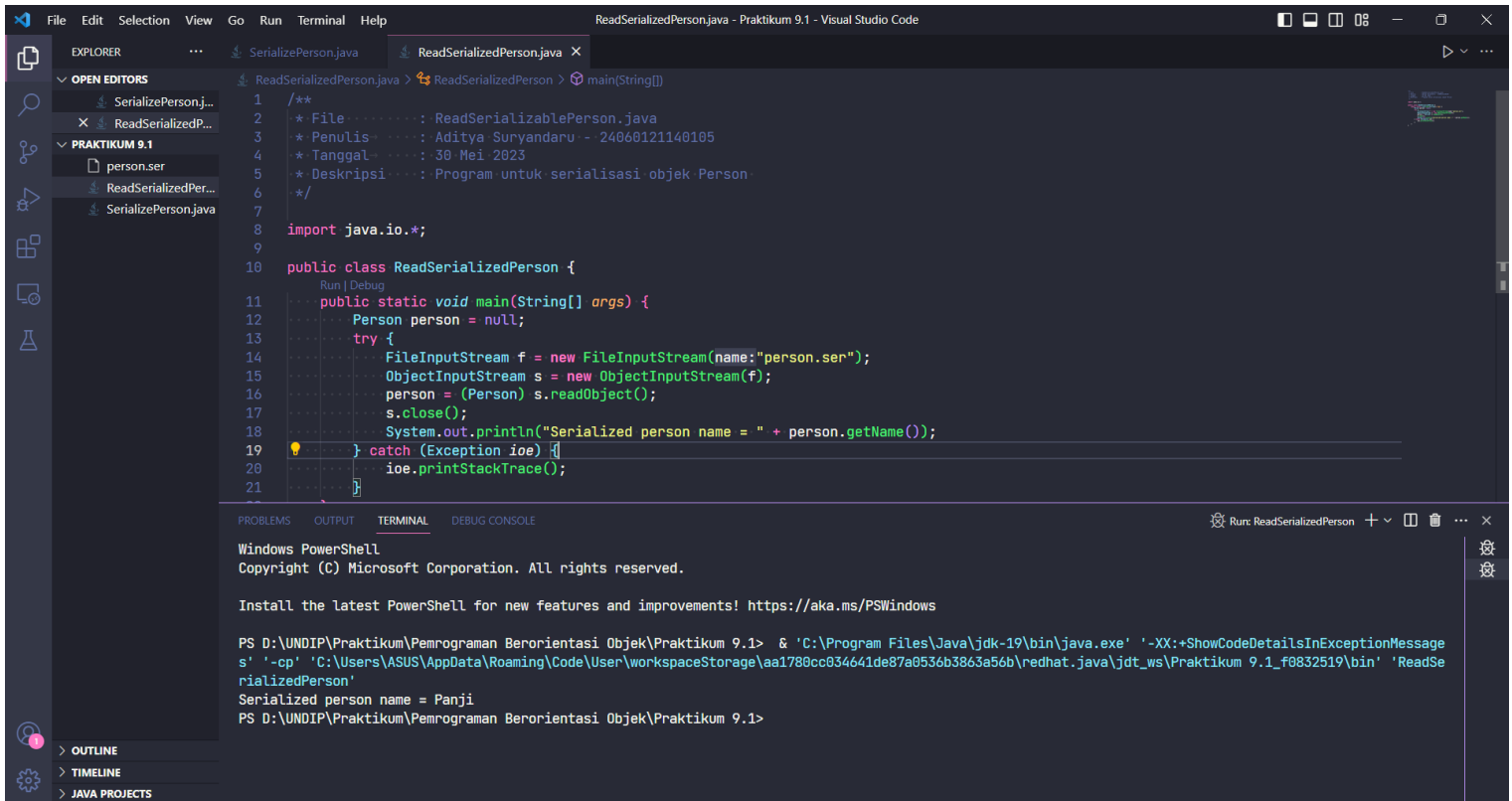
```
/**
 * File      : ReadSerializablePerson.java
 * Penulis   : Aditya Suryandaru - 24060121140105
 * Tanggal    : 30 Mei 2023
 * Deskripsi  : Program untuk serialisasi objek Person
 */

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();
        }
    }
}
```

4. Compile dan jalankan program diatas



```
File Edit Selection View Go Run Terminal Help
ReadSerializedPerson.java - Praktikum 9.1 - Visual Studio Code

EXPLORER
  OPEN EDITORS
    SerializePerson.j...
    ReadSerializedP...
  PRAKTIKUM 9.1
    person.ser
    ReadSerializedPer...
    SerializePerson.java

ReadSerializedPerson.java
1  /**
2  * File .....: ReadSerializablePerson.java
3  * Penulis .....: Aditya Suryandaru - 24060121140105
4  * Tanggal .....: 30 Mei 2023
5  * Deskripsi .....: Program untuk serialisasi objek Person
6  */
7
8  import java.io.*;
9
10 public class ReadSerializedPerson {
11     public static void main(String[] args) {
12         Person person = null;
13         try {
14             FileInputStream f = new FileInputStream(name:"person.ser");
15             ObjectInputStream s = new ObjectInputStream(f);
16             person = (Person) s.readObject();
17             s.close();
18             System.out.println("Serialized person name = " + person.getName());
19         } catch (Exception ioe) {}
20         ioe.printStackTrace();
21     }
22 }

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\UNDIP\Praktikum\Pemrograman Berorientasi Objek\Praktikum 9.1> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessage' '-cp' 'C:\Users\ASUS\AppData\Roaming\Code\User\workspaceStorage\aa1780cc834641de87a0536b3863a56b\redhat.java\jdt_ws\Praktikum 9.1_f0832519\bin' 'ReadSerializedPerson'
Serialized person name = Panji
PS D:\UNDIP\Praktikum\Pemrograman Berorientasi Objek\Praktikum 9.1>
```