

PBO PersistentObject

1. DAOManager.java

```
DAOManager.java > DAOManager
1  /**
2   * File : DAOManager.java 31/05/2023
3   * Nama : Abyan Izzah / 24060121130082
4   * Deskripsi : Pengelola DAO dalam program
5   */
6
7  public class DAOManager {
8      private PersonDAO PersonDAO;
9
10     public void setPersonDAO(PersonDAO person){
11         PersonDAO = person;
12     }
13
14     public PersonDAO getPersonDAO(){
15         return PersonDAO;
16     }
17 }
18
19
20
```

2. MainDAO.java

```
MainDAO.java > MainDAO > main(String[])
1  /**
2   * File : MainDAO.java 31/05/2023
3   * Nama : Abyan Izzah / 24060121130082
4   * Deskripsi : main program untuk akses DAO
5   */
6
7  public class MainDAO {
8      Run | Debug
9      public static void main (String args[]){
10         Person person = new Person(n:"Abyan");
11         DAOManager m = new DAOManager();
12         m.setPersonDAO(new MySQLPersonDAO());
13         try{
14             m.getPersonDAO().savePerson(person);
15         }catch(Exception e){
16             e.printStackTrace();
17         }
18     }
19 }
20
```

3. MySQLPersonDAO.java

```
1  /**
2   * File : MySQLPersonDAO.java 31/05/2023
3   * Nama : Aryan Izzah / 24060121130082
4   * Deskripsi : Implementasi PersonDAO untuk MySQL
5   */
6
7  import java.sql.*;
8
9  public class MySQLPersonDAO implements PersonDAO{
10     public void savePerson(Person person) throws Exception{
11         String name = person.getName();
12
13         // Membuat Koneksi, Nama DB, User, Password Menyesuaikan
14         Class.forName(className:"com.mysql.jdbc.Driver");
15         Connection con = DriverManager.getConnection(url:"jdbc:mysql://localhost/PBO", user:"root", password:"");
16
17         // Kerjakan MySQL Query
18         String query = "INSERT INTO person(name) VALUES ('"+name+"')";
19         System.out.println(query);
20         Statement s = con.createStatement();
21         s.executeUpdate(query);
22
23         // Tutup Koneksi Database
24         con.close();
25     }
26 }
27
28 }
```

4. Person.java

```
1  /**
2   * File : Person.java 31/05/2023
3   * Nama : Aryan Izzah / 24060121130082
4   * Deskripsi : Person Database Model
5   */
6
7  public class Person {
8      private int id;
9      private String name;
10
11      public Person(String n){
12          name = n;
13      }
14
15      public Person(int i, String n){
16          id = i;
17          name = n;
18      }
19
20      public int getId(){
21          return id;
22      }
23
24      public String getName(){
25          return name;
26      }
27 }
28 }
```

5. PersonDAO.java

```
PersonDAO.java > PersonDAO
1  /**
2   *   File : PersonDAO.java 31/05/2023
3   *   Nama : Abyan Izzah / 24060121130082
4   *   Deskripsi : Interface untuk person access Object
5   */
6
7  public interface PersonDAO{
8      public void savePerson(Person p) throws Exception;
9  }
```

6. Hasil Run MainDAO dalam Command Prompt

```
D:\Kuliah Momen\Semester 4\PBO\Praktikum\Pertemuan 9\Coba2\PersistentObject>javac MySQLPersonDAO.java
D:\Kuliah Momen\Semester 4\PBO\Praktikum\Pertemuan 9\Coba2\PersistentObject>java MainDAO
INSERT INTO person(name) VALUES ('Indra')
D:\Kuliah Momen\Semester 4\PBO\Praktikum\Pertemuan 9\Coba2\PersistentObject>javac MySQLPersonDAO.java
D:\Kuliah Momen\Semester 4\PBO\Praktikum\Pertemuan 9\Coba2\PersistentObject>javac MainDAO.java
D:\Kuliah Momen\Semester 4\PBO\Praktikum\Pertemuan 9\Coba2\PersistentObject>java MainDAO
INSERT INTO person(name) VALUES ('Abyan')
D:\Kuliah Momen\Semester 4\PBO\Praktikum\Pertemuan 9\Coba2\PersistentObject>
```

7. Tabel Person dalam Database

localhost/phpmyadmin/index.php?route=/database/sql&db=pbo

Server: 127.0.0.1 » Database: pbo » Table: person

Browse Structure SQL Search Insert Export

Show query box

Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

`select * from person;`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]



☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

				id	name
<input type="checkbox"/>	Edit	Copy	Delete	1	Indra
<input type="checkbox"/>	Edit	Copy	Delete	2	Abyan

PersistentObjectSerial

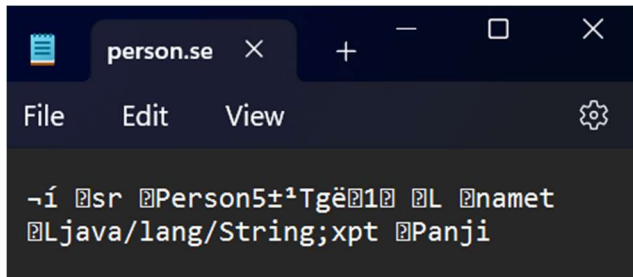
1. SerializePerson.java

```
SerializePerson.java >  SerializePerson >  main(String[])
1  import java.io.*;
2  /**
3   * File : SerializePerson.java 31/05/2023
4   * Nama : Abyan Izzah / 24060121130082
5   * Deskripsi : Program untuk serialisasi objek Person
6   */
7  // Class Person
8  class Person implements Serializable{
9      private String name;
10     public Person(String n){
11         name = n;
12     }
13     public String getName(){
14         return name;
15     }
16 }
17 // Class Serialize Person
18 public class SerializePerson{
19     Run | Debug
20     public static void main(String[] args){
21         Person person = new Person(n:"Panji");
22         try{
23             FileOutputStream f = new FileOutputStream(name:"person.ser");
24             ObjectOutputStream s = new ObjectOutputStream(f);
25             s.writeObject(person);
26             System.out.println(x:"Selesai Menulis Objek Person");
27             s.close();
28         }catch(IOException e){
29             e.printStackTrace();
30         }
31     }
```

2. Hasil Run SerializePerson.java

```
D:\Kuliah Momen\Semester 4\PBO\Praktikum\Pertemuan 9\Coba2\PersistentObjectSerial>javac SerializePerson.java
D:\Kuliah Momen\Semester 4\PBO\Praktikum\Pertemuan 9\Coba2\PersistentObjectSerial>java SerializePerson
Selesai Menulis Objek Person
```

3. Isi Person.ser



Lambda

1. DiskonLambda.java

```
DiskonLambda.java > ...  
1  /**  
2   * File : DiskonLambda.java 31/05/2023  
3   * Nama : Abyan Izzah / 24060121130082  
4   * Deskripsi :Ekspresi Lambda Dasar, Untuk Menghitung Diskon  
5   */  
6  
7  interface IDiskon{  
8      public double hitungDiskon(int harga);  
9  }  
10  
11  public class DiskonLambda{  
12      Run | Debug  
13      public static void main(String[] args){  
14          // tanpa Lambda  
15          IDiskon diskonMerdeka = new IDiskon(){  
16              public double hitungDiskon(int harga){  
17                  return harga - (harga * 0.3);  
18              }  
19          };  
20  
21          // dengan Lambda  
22          IDiskon diskonLebaran = (harga) -> harga - (harga * 0.4);  
23  
24          // dengan Lambda dengan blok statement  
25          IDiskon diskonBiasa = (harga) -> {  
26              return harga - (harga * 0.1);  
27          };  
28  
29          System.out.println("Diskon Merdeka : " + diskonMerdeka.hitungDiskon(harga:450000));  
30          System.out.println("Diskon Lebaran : " + diskonLebaran.hitungDiskon(harga:450000));  
31          System.out.println("Diskon Biasa : " + diskonBiasa.hitungDiskon(harga:450000));  
32      }  
33  }
```

2. Jawaban Pertanyaan

```
}  
}  
  
/*  
 * Perbedaan antara diskonLebaran dan diskonBiasa saat diimplementasikan adalah pada diskonLebaran diimplementasikan  
 * tanpa menggunakan blok statement "{}", sehingga lebih sederhana dan lebih ringkas. Berbeda dengan diskonBiasa yang  
 * menggunakan blok statement "{}", biasanya lebih rumit dan kompleks sehingga lebih cocok digunakan pada algoritma  
 * yang lebih kompleks, seperti adanya if else, while, dll. Dalam kasus ini, lebih cocok menggunakan bentuk lambda  
 * pada diskonLebaran karena kasus yang diberikan tidak terlalu kompleks.  
 */
```

3. Hasil Run DiskonLambda.java

```
PS D:\Kuliah Momen\Semester 4\PBO\Praktikum\Pertemuan 9\Coba3\Lambda>  
ambda'; & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-XX:+ShowCodeDe  
User\workspaceStorage\82d6d8f27a6175d3100285b79eeaebd9\redhat.java\jdt  
Diskon Merdeka : 315000.0  
Diskon Lebaran : 270000.0  
Diskon Biasa : 405000.0  
PS D:\Kuliah Momen\Semester 4\PBO\Praktikum\Pertemuan 9\Coba3\Lambda>
```

4. LambdaList.java

```
LambdaList.java > ...  
1  /**  
2   * File : LambdaList.java 31/05/2023  
3   * Nama : Abyan Izzah / 24060121130082  
4   * Deskripsi : Implementasi Lambda dalam list, Digunakan sebagai parameter dalam method  
5   */  
6  import java.util.ArrayList;  
7  
8  
9  public class LambdaList {  
10     Run | Debug  
11     public static void main(String[] args){  
12         ArrayList<String> mahasiswaList = new ArrayList<>();  
13         mahasiswaList.add(e:"Adi");  
14         mahasiswaList.add(e:"Abyan");  
15         mahasiswaList.add(e:"Dezach");  
16         mahasiswaList.add(e:"Budi");  
17  
18         // Lambda dijadikan Paramater  
19         mahasiswaList.forEach((nama) -> System.out.println(nama));  
20     }  
21 }
```


5. Hasil Run LambdaList.java

```
PS D:\Kuliah Momen\Semester 4\PBO\Praktikum\Pertemuan 9\Coba3\Lambda>  
ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\abyan\AppData\Roam  
b79eeae9d9\redhat.java\jdt_ws\Lambda_12ccdcaf\bin' 'LambdaList'  
Adi  
Abyan  
Dezach  
Budi
```