**TUGAS PERTEMUAN KE - 8**

|  |  |
| --- | --- |
| **NAMA** | Rayya Muhammad Idham |
| **NIM** | 2218083 |
| **KELAS** | C |
| **PEMBERI TUGAS** | MUHAMMAD RIDHO PUTRA SYALABI (2118014) |

**(PRAKTIKUM OOP 2022-2023)**

## Tugas Rumah 8 :

Judul : Penyewaan Peralatan Olahraga

Tema : Persewaan

Class diagram (Pemesanan.java):

*A screenshot of a diagram

Description automatically generated*

*Source code Object Class* (Pemesanan.java):

|  |
| --- |
| public class Pemesanan extends abstractClass implements Informasi\_DataPelanggan{  //Sebagai Atribut yang dimiliki oleh PersewaanPeralatanOlahraga  double jumlah,harga,total,hari;  String nama,jenisalat,jangkapenggunaan,bentuk,ukuran,bahan,atribut,A;    //constructor  public Pemesanan(){  this.bentuk = "Normal(Klasik)";  this.ukuran = "Normal";  this.harga = harga;  }    //Merupakan Method yang menggunakan void  void dataNama(String Nama){  this.nama = Nama;  }  void dataJenisAlat(String JenisAlat){  this.jenisalat = JenisAlat;  }  void dataJangkaPenggunaan(String JangkaPenggunaan){  this.jangkapenggunaan = JangkaPenggunaan;  }  void dataBentuk(String Bentuk){  this.bentuk = Bentuk;  }  void dataUkuran(String Ukuran){  this.ukuran = Ukuran;  }  void dataBahan(String Bahan){  this.bahan = Bahan;  }    public String dataAtribut(String Atribut){  return atribut = Atribut;  }  public String dataAtribut(String Bentuk, String Ukuran, String Bahan){  return atribut = Bentuk + "/" + Ukuran + "/" + Bahan;  }  //Merupakan Method yang Non-void  @Override  public String cetakNama(){  return nama;  }  public double cetakJumlah(){  return jumlah;  }  String cetakJenisAlat(){  return jenisalat;  }  String cetakJangkaPenggunaan(){  return jangkapenggunaan;  }  String cetakBentuk(){  return bentuk;  }  String cetakUkuran(){  return ukuran;  }  String cetakBahan(){  return bahan;  }  //metode  public double hitungHarga(){  return 0;  }  @Override  public String Status() {  return "Data Tidak Tersedia";  }  public String Status(String A) {  return "Data Tersedia";  }    boolean cek(){  return true;  }  boolean cek(int rowCount) {  throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody  }  } |

Desain *form* (GUI\_Peralatan.java):

A screenshot of a computer

Description automatically generated

Gambar 7.1 Desain PeralatanOlahraga.java

*Source code Object Class Absract* (StatusMember.java):

|  |
| --- |
| public abstract class StatusMember extends Pemesanan {  public abstract double hitungHarga();  abstract double cekMember(String input);  abstract String tampilkanMember();    String member(V1 member){  return "VIP1";  }    String member(V2 member){  return "VIP2";  }  } |

*Source code Interface Class* (.java):

|  |
| --- |
| public interface Informasi\_DataPelanggan {  abstract String cetakNama();  } |

*Source code Object Class* (V1 dan V2.java):

|  |
| --- |
| public class V1 extends StatusMember{  public String kode;  double diskon;    public V1(){  super();  this.kode = "V1";  }    public double hasil(){  return diskon;  }  @Override  double cekMember(String input){  if(input.compareTo(kode) == 0){  diskon = 0.2;  }else{  diskon = 0;  }  return diskon;  }  @Override  public double hitungHarga(){  total = (harga \* jumlah) \* hari;  diskon = total \* diskon;  total = total - diskon;  return total;  }    @Override  public String tampilkanMember(){  return "Diskon 20%";  }  }  public class V2 extends StatusMember{  public String kode;  double diskon;    public V2(){  super();  this.kode = "V2";  }    public double hasil(){  return diskon;  }  @Override  double cekMember(String input){  if(input.compareTo(kode) == 0){  diskon = 0.3;  }else{  diskon = 0;  }  return diskon;  }  @Override  public double hitungHarga(){  diskon = harga \* diskon;  total = (harga \* jumlah) \* hari;  total = total - diskon;  return total;  }    @Override  public String tampilkanMember(){  return "Diskon 30%";  }  } |

#### Tabel 7.1 Properti Desain GUI PeralatanOlahraga.java

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Komponen** | **Properti** | **Value** |
| 1 | jLabel1 | Text | Persewaan Peralatan Olahraga |
| 2 | jLabel2 | Text | Atas Nama |
| 3 | jLabel3 | Text | Jumlah |
| 4 | jLabel4 | Text | Jenis Alat |
| 5 | jLabel5 | Text | Jangka Penggunaan |
| 6 | jLabel6 | Text | Bentuk |
| 7 | jLabel7 | Text | Ukuran |
| 8 | jLabel8 | Text | Bahan |
| 9 | jLabel9 | Text | Harga/unit |
| 10 | jTextField1 | Name | txtNama |
| Text |  |
| 11 | jTextField2 | Name | txtJumlah |
| Text |  |
| 12 | jTextField4 | Name | txtBentuk |
| Text |  |
| 13 | jTextField5 | Name | TxtUkuran |
| Text |  |
| 14 | jTextField6 | Name | txtBahan |
| Text |  |
| 15 | jTextField7 | Name | txtHarga |
| Text |  |
| 16 | jButton1 | Name | btnCetak |
| Text | Cetak |
| 17 | jButton2 | Name | btnClose |
| Text | Close |
| 18 | jButton3 | Name | btnHapus |
| Text | Hapus |
| 19 | jButton4 | Name | btnBatal |
| Text | Batal |
| 20 | jButton5 | Name | btnCek |
| Text | Cek |
| 21 | jTable1 | Name | table |
| Text |  |
| 22 | jRadioButton | Name | Radiobtn1Hari |
| Text | 1 Hari |
| 23 | jRadioButton | Name | Radiobtn3Hari |
| Text | 3 Hari |
| 24 | jRadioButton | Name | Radiobtn7Hari |
| Text | 7 Hari |
| 25 | jComboBox | Model | [0] = Bola  [1] = Raket  [2] = Sepatu Sepasang  [3] =Sarung Tinju |
| Nama | cmb\_pilih |

*Source code Button/combobox Cetak and Close*:

|  |
| --- |
| package Tugas\_Laporan;  import java.util.ArrayList;  import java.util.List;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableColumnModel;  import javax.swing.table.DefaultTableModel;  private void btnCetakActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  JOptionPane.showMessageDialog(null, "Data Anda Ditambahkan ke Tabel");  DefaultTableModel dataModel = (DefaultTableModel) table.getModel();  List list = new ArrayList<>();  table.setAutoCreateColumnsFromModel(true);  Pemesanan PPO = new Pemesanan();  StatusMember Member;  Member = new V1();    if (Member instanceof V1){  V1 person = (V1) Member;  PPO.dataNama(txtNama.getText());  person.jumlah =Double.parseDouble(txtJumlah.getText());  if (radiobtn1Hari.isSelected()){  PPO.dataJangkaPenggunaan(radiobtn1Hari.getText());  person.hari = 1;  }  else if (radiobtn3Hari.isSelected()){  PPO.dataJangkaPenggunaan(radiobtn3Hari.getText());  person.hari = 1.5;  }  else{  PPO.dataJangkaPenggunaan(radiobtn7Hari.getText());  person.hari = 3;  }    PPO.dataBentuk(txtBentuk.getText());  PPO.dataUkuran(txtUkuran.getText());  PPO.dataBahan(txtBahan.getText());    if(cmb\_pilih.getSelectedIndex() == 0){  PPO.jenisalat = "Bola";  PPO.bahan = "Kulit";  person.harga = 20000;  list.add(PPO.cetakNama());  list.add(person.cetakJumlah());  list.add(PPO.cetakJangkaPenggunaan());  list.add(PPO.cetakJenisAlat());  list.add(PPO.cetakBentuk());  list.add(PPO.cetakUkuran());  list.add(PPO.cetakBahan());  list.add(person.tampilkanMember());  list.add(person.hitungHarga());  dataModel.addRow(list.toArray());  }else if(cmb\_pilih.getSelectedIndex() == 1){  PPO.jenisalat = "Raket";  PPO.bahan = "Aluminium/Nilon";  person.harga = 10000;  list.add(PPO.cetakNama());  list.add(PPO.cetakJumlah());  list.add(PPO.cetakJangkaPenggunaan());  list.add(PPO.cetakJenisAlat());  list.add(PPO.cetakBentuk());  list.add(PPO.cetakUkuran());  list.add(PPO.cetakBahan());  list.add(person.tampilkanMember());  list.add(person.hitungHarga());  dataModel.addRow(list.toArray());  }else if(cmb\_pilih.getSelectedIndex() == 2){  PPO.jenisalat = "Sepatu Sepasang";  PPO.bahan = "Kulit";  person.harga = 42000;  list.add(PPO.cetakNama());  list.add(PPO.cetakJumlah());  list.add(PPO.cetakJangkaPenggunaan());  list.add(PPO.cetakJenisAlat());  list.add(PPO.cetakBentuk());  list.add(PPO.cetakUkuran());  list.add(PPO.cetakBahan());  list.add(person.tampilkanMember());  list.add(person.hitungHarga());  dataModel.addRow(list.toArray());  }else if(cmb\_pilih.getSelectedIndex() == 3){  PPO.jenisalat = "Sarung Tinju";  PPO.bahan = "Nilon";  person.harga = 50000;  list.add(PPO.cetakNama());  list.add(PPO.cetakJumlah());  list.add(PPO.cetakJangkaPenggunaan());  list.add(PPO.cetakJenisAlat());  list.add(PPO.cetakBentuk());  list.add(PPO.cetakUkuran());  list.add(PPO.cetakBahan());  list.add(person.tampilkanMember());  list.add(person.hitungHarga());  dataModel.addRow(list.toArray());  }  }  private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  DefaultTableModel dataModel = (DefaultTableModel) table.getModel();  int rowCount = dataModel.getRowCount();  while (rowCount > 0){  dataModel.removeRow(rowCount - 1);  rowCount = dataModel.getRowCount();  }    }  private void btnBatalActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  clear();  }  private void btnCloseActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  dispose();  }  private void btnCekActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  DefaultTableModel dataModel = (DefaultTableModel) table.getModel();  int rowCount = dataModel.getRowCount();  Pemesanan PPO = new Pemesanan();  boolean isAuthenticated;  isAuthenticated = PPO.cek(rowCount = dataModel.getRowCount());  if(isAuthenticated){  JOptionPane.showMessageDialog(rootPane, ""+PPO.Status(PPO.A));  }else{  JOptionPane.showMessageDialog(rootPane, ""+PPO.Status());  }  } |

Hasil Tampilan:

A screenshot of a computer

Description automatically generated

Gambar 7.2 Tampilan Hasil Running

Analisa:

Program GUI Sistem Penyewaan Peralatan Olahraga ini menggunakan Java Swing dengan *JFrame*, memberikan pengguna kemampuan untuk menambah dan menghapus data merk pada tabel. Ketika tombol "Simpan" ditekan, data yang dimasukkan oleh pengguna akan ditambahkan sebagai baris baru dalam tabel. Sementara itu, tombol "Hapus" memungkinkan pengguna untuk menghapus baris yang dipilih dari tabel, meningkatkan fungsionalitas dan kenyamanan dalam pengelolaan data.

## Kesimpulan

* + - 1. *Abstract Class* adalah sebuah *class* yang tidak bisa di-instansiasi (tidak bisadibuat menjadi objek) , berperan sebagai ‘kerangka dasar’ bagi *classturunannya*. Di dalam sebuah *abstract class* setidaknya memiliki satu atau lebih *method abstrak*
      2. Fungsi dari class abstract ini adalah untuk mempertahan kanhirarky dari *parent class* ke kelas turunan dari induknya
      3. *Abstract class* digunakan di dalam inheritance (pewarisan class) untuk ‘memaksakan’implementasi method yang sama bagi seluruh class yang diturunkan dari *abstract class*.