**TUGAS PERTEMUAN KE - 9**

|  |  |
| --- | --- |
| **NAMA** | Rayya Muhammad Idham |
| **NIM** | 2218083 |
| **KELAS** | C |
| **PEMBERI TUGAS** | DIMAS SURYO PRASETYO (2118042) |

**(PRAKTIKUM OOP 2022-2023)**

## Tugas Rumah 10 :

Judul : Penyewaan Peralatan Olahraga

Tema : Persewaan

Menerapkan CRUD Pada PeralatanOlahraga

Membuat Database Persewaan

A screenshot of a computer

Description automatically generated

### Gambar 10.1 Membuat database PeralatanOlahraga

*Structure* PeralatanOlahraga pada database Persewaan

A screenshot of a computer

Description automatically generated

### Gambar 10.2 Membuat table PeralatanOlahraga

*Source code* pada GUI:

|  |
| --- |
| import java.sql.Connection;  import java.sql.*Driver*Manager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel; |

*Source code method* koneksi

|  |
| --- |
| public void koneksi() throws SQLException {  try {  conn = null;  Class.forName("com.mysql.cj.jdbc.Driver");  conn = DriverManager.getConnection("jdbc:mysql://localhost/2218083Tugas\_Rumah?user=root&password=");  } catch (ClassNotFoundException ex) {  Logger.getLogger(PeralatanOlahraga.class.getName()).log(Level.SEVERE, null, ex);  } catch (SQLException e) {  Logger.getLogger(PeralatanOlahraga.class.getName()).log(Level.SEVERE, null, e);  } catch (Exception es) {  Logger.getLogger(PeralatanOlahraga.class.getName()).log(Level.SEVERE, null, es);  }  } |

*Source code* di *method* tampil*:*

|  |
| --- |
| public void tampil() {  DefaultTableModel tabelhead = new DefaultTableModel();  tabelhead.addColumn("NAMA");  tabelhead.addColumn("JUMLAH");  tabelhead.addColumn("JANGKA WAKTU");  tabelhead.addColumn("JENIS ALAT");  tabelhead.addColumn("BENTUK");  tabelhead.addColumn("UKURAN");  tabelhead.addColumn("BAHAN");  tabelhead.addColumn("HARGA");  try {  koneksi();  String sql = "SELECT \* FROM peralatanolahraga";  Statement stat = conn.createStatement();  ResultSet res = stat.executeQuery(sql);  while (res.next()) {  tabelhead.addRow(new Object[]{res.getString(2), res.getString(3), res.getString(4), res.getString(5), res.getString(6), res.getString(7), res.getString(8), res.getString(9),});  }  table.setModel(tabelhead);  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "BELUM TERKONEKSI");  }  } |

*Source code method refresh :*

|  |
| --- |
| public void refresh() {  new PeralatanOlahraga().setVisible(true);  this.setVisible(false);  } |

*Source code method tambah :*

|  |
| --- |
| public void insert() {  String Nama = txtNama.getText();  String Jumlah = txtJumlah.getText();  String jw;  if (radiobtn1Hari.isSelected()) {  jw = "1 Hari";  } else if (radiobtn3Hari.isSelected()){  jw = "3 Hari";  } else {  jw = "7 Hari";  }  String ja = txtAlat.getText();  String Bentuk = txtBentuk.getText();  String Ukuran = txtUkuran.getText();  String Bahan = txtBahan.getText();  String Harga = txtHarga.getText();  try {  koneksi();  Statement statement = conn.createStatement();  statement.executeUpdate("INSERT INTO peralatanolahraga (Nama, Jumlah, Jangka\_Waktu, Jenis\_Alat, Bentuk, Ukuran, Bahan, Harga)"  + "VALUES('" + Nama + "','" + Jumlah + "','" + jw + "','" + ja + "','" + Bentuk + "','" + Ukuran + "','" + Bahan + "','" + Harga + "')");  statement.close();  JOptionPane.showMessageDialog(null, "Berhasil Memasukan Data Pemesanan!" + "\n");  } catch (Exception e) {  e.printStackTrace();  JOptionPane.showMessageDialog(null, "Terjadi Kesalahan Input!");  }  refresh();  } |

*Source code method ubah :*

|  |
| --- |
| public void update() {  String nama = txtNama.getText();  String jumlah = txtJumlah.getText();  String jw;  if (radiobtn1Hari.isSelected()) {  jw = "1 Hari";  } else if (radiobtn3Hari.isSelected()){  jw = "3 Hari";  } else {  jw = "7 Hari";  }  String ja = txtAlat.getText();  String bentuk = txtBentuk.getText();  String ukuran = txtUkuran.getText();  String bahan = txtBahan.getText();  String harga = txtHarga.getText();  String namaLama = nama1;  try {  Statement statement = conn.createStatement();  statement.executeUpdate("UPDATE peralatanolahraga SET Nama='" + nama + "'," + "Jumlah='" + jumlah + "',"  + "Jangka\_Waktu='" + jw + "'," + "Jenis\_Alat='" + ja + "',bentuk='" + bentuk + "',ukuran='"  + ukuran + "',Bahan='" + bahan + "',Harga='" + harga + "' WHERE Nama = '" + namaLama + "'");  statement.close();  conn.close();  JOptionPane.showMessageDialog(null, "Update Data Mahasiswa Berhasil!");  } catch (Exception e) {  System.out.println("Error : " + e);  }  refresh();  } |

*Source code method hapus :*

|  |
| --- |
| public void delete() {  int ok = JOptionPane.showConfirmDialog(null, "Apakah Anda yakin akan menghapus data ?", "Konfirmasi", JOptionPane.YES\_NO\_OPTION);  if (ok == 0) {  try {  String sql = "DELETE FROM peralatanolahraga WHERE Nama ='" + txtNama.getText() + "'";  java.sql.PreparedStatement stmt = conn.prepareStatement(sql);  stmt.executeUpdate();  JOptionPane.showMessageDialog(null, "Data Berhasil di hapus");  batal();  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "Data gagal di hapus");  }  }  refresh();  } |

Tambahkan library mysql.connector.jar pada library dengan cara klik kanan pada library, kemudian pilih **Add JAR/Folder...**

A black text on a white background

Description automatically generated

### Gambar 10.3 Menambahkan *Library Connection*

Desain *form* PeralatanOlahraga

A screenshot of a computer

Description automatically generated

### Gambar 10.4 Desain Form PeralatanOlahraga

#### Properti Desain PeralatanOlahraga (Database)

|  |  |  |  |
| --- | --- | --- | --- |
| **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  7 | jLabel6  jLabel7 | Text | Bentuk |
| Text | Ukuran |
| 8  9 | jLabel8  jLabel9 | Text | Bahan |
| Text | Harga/unit |
| 10 | jTextField1 | Name | txtNama |
| Text |  |
| 11 | jTextField2 | Name | txtJumlah |
| 10  12 | jTextField3  jTextField4 | Text |  |
| Name | txtBentuk |
| 11  13 | jTextField4  jTextField5 | Text |  |
| Name | TxtUkuran |
| 13  14 | jButton1  jTextField6 | Text |  |
| Name | txtBahan |
| 13 | JButton2 | Text |  |
| 15 | jTextField7 | Name | txtHarga |
| 15 | Combo box1 | Text |  |
| 16 | jButton1 | Name | btnCetak |
| 17 | JradioButton2 | Text | Cetak |
| 17 | jButton2 | Name | btnClose |
| 19 | JCekbox2 | Text | Close |
| 18 | jButton3 | Name | btnHapus |
| 21 | JCekbox4 | Text | Hapus |
| 19 | jButton4 | Name | btnBatal |
| 23 | JCekbox6 | Text | Batal |
| 20 | jButton5 | Name | btnCek |
|  |  | Text | Cek |
| 21 | jTable1 | Name | table |
| 25 | jTable | Text |  |
| 22 | jRadioButton | Name | Radiobtn1Hari |
|  |  | Text | 1 Hari |

*Source code* pada tombol tambah*:*

|  |
| --- |
| private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {          // TODO add your handling code here:          insert();      } |

*Source code* pada tombol Ubah*:*

|  |
| --- |
| private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {          // TODO add your handling code here:          update();      } |

*Source code* pada tombol hapus*:*

|  |
| --- |
| private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {          // TODO add your handling code here:          delete();      } |

*Source code* pada tombol cari*:*

|  |
| --- |
| private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {          // TODO add your handling code here:          cari();      } |

*Source code* pada tombol batal*:*

|  |
| --- |
| private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {          // TODO add your handling code here:          batal();      } |

Tampilan Hasil:

A screenshot of a computer

Description automatically generated

### Gambar 10.5 Tampilan Hasil PeralatanOlahraga

Analisa:

*Source code* PeralatanOlahraga memanggil PeralatanOlahraga menggunakan objek gpb lalu ditampilkan menggunakan perintah show dan PeralatanOlahraga diatas menerapkan Polimorfisme lalu menggunakan casting bertipe downcasting, lalu ada combobox yang berfungsi untuk memilih inputan dan akan berubah sesuai dengan inputan yang dipilih dan akan menampilkan *message* apabila telah menekan *button* rekap dan mengisi Pesanan kamar,makanan dan minuman

Menerapkan CRUD Pada GUI\_Jaminan

Membuat Database Persewaan

A screenshot of a computer

Description automatically generated

### Gambar 10.6 Membuat database Persewaan

*Structure*  pada database

A screenshot of a computer

Description automatically generated

### Gambar 10.7 Membuat table

*Source code* pada GUI:

|  |
| --- |
| import java.sql.Connection;  import java.sql.*Driver*Manager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel; |

*Source code method* koneksi

|  |
| --- |
| public void koneksi() throws SQLException {  try {  conn = null;  Class.forName("com.mysql.cj.jdbc.Driver");  conn = DriverManager.getConnection("jdbc:mysql://localhost/2218083Tugas\_Rumah?user=root&password=");  } catch (ClassNotFoundException ex) {  Logger.getLogger(GUI\_Jaminan.class.getName()).log(Level.SEVERE, null, ex);  } catch (SQLException e) {  Logger.getLogger(GUI\_Jaminan.class.getName()).log(Level.SEVERE, null, e);  } catch (Exception es) {  Logger.getLogger(GUI\_Jaminan.class.getName()).log(Level.SEVERE, null, es);  }  } |

*Source code* di *method* tampil*:*

|  |
| --- |
| public void tampil() {  DefaultTableModel tabelhead = new DefaultTableModel();  tabelhead.addColumn("NAMA");  tabelhead.addColumn("JENIS ALAT");  tabelhead.addColumn("KONTRAK");  tabelhead.addColumn("JAMINAN");  tabelhead.addColumn("PENGEMBALIAN");  try {  koneksi();  String sql = "SELECT \* FROM gui\_jaminan";  Statement stat = conn.createStatement();  ResultSet res = stat.executeQuery(sql);  while (res.next()) {  tabelhead.addRow(new Object[]{res.getString(2), res.getString(3), res.getString(4), res.getString(5), res.getString(6),});  }  table.setModel(tabelhead);  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "BELUM TERKONEKSI");  }  } |

*Source code method refresh :*

|  |
| --- |
| public void refresh() {  new GUI\_Jaminan().setVisible(true);  this.setVisible(false);  } |

*Source code method tambah :*

|  |
| --- |
| public void insert() {  String nama = txtNama.getText();  String ja = txtAlat.getText();  String kontrak = txtKontrak.getText();  String jaminan = txtHarga.getText();  String pengembalian = txtPemulihan.getText();  try {  koneksi();  Statement statement = conn.createStatement();  statement.executeUpdate("INSERT INTO gui\_jaminan(Nama, Jenis\_Alat, Kontrak, Jaminan, Pengembalian)"  + "VALUES('" + nama + "','" + ja + "','" + kontrak + "','" + jaminan + "','" + pengembalian + "')");  statement.close();  JOptionPane.showMessageDialog(null, "Berhasil Memasukan Data Matakuliah!");  } catch (Exception e) {  e.printStackTrace();  JOptionPane.showMessageDialog(null, "Terjadi Kesalahan Input!");  }  refresh();  } |

*Source code method ubah :*

|  |
| --- |
| public void update() {  String nama = txtNama.getText();  String ja = txtAlat.getText();  String kontrak = txtKontrak.getText();  String jaminan = txtHarga.getText();  String pengembalian = txtPemulihan.getText();  String namaLama = nama1;  try {  Statement statement = conn.createStatement();  statement.executeUpdate("UPDATE gui\_jaminan SET Nama='" + nama + "'," + "Jenis\_Alat='" + ja + "',"  + "Kontrak='" + kontrak + "'" + ",Jaminan='" + jaminan + ",Pengembalian='" + pengembalian + "'WHERE Nama = '" + namaLama + "'");  statement.close();  conn.close();  JOptionPane.showMessageDialog(null, "Update Data MataKuliah!");  } catch (Exception e) {  System.out.println("Error : " + e);  }  refresh();  } |

*Source code method hapus :*

|  |
| --- |
| public void delete() {  int ok = JOptionPane.showConfirmDialog(null, "Apakah Anda yakin akan menghapus data ?", "Konfirmasi", JOptionPane.YES\_NO\_OPTION);  if (ok == 0) {  try {  String sql = "DELETE FROM gui\_jaminan WHERE Nama='" + txtNama.getText() + "'";  java.sql.PreparedStatement stmt = conn.prepareStatement(sql);  stmt.executeUpdate();  JOptionPane.showMessageDialog(null, "Data Berhasil di hapus");  clear();  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "Data gagal di hapus");  }  }  refresh();  } |

Tambahkan library mysql.connector.jar pada library dengan cara klik kanan pada library, kemudian pilih **Add JAR/Folder...**

A black text on a white background

Description automatically generated

### Gambar 10.8 Menambahkan *Library Connection*

Desain *form* GUI\_Jaminan

A screenshot of a computer

Description automatically generated

### Gambar 10.9 Desain Form GUI\_Jaminan

#### Properti Desain GUI\_Jaminan (Database)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Komponen** | **Properti** | **Value** |
| 1 | jLabel1 | Text | Jaminan Barang |
| 2 | jLabel2 | Text | Nama |
| 3 | jLabel3 | Text | Jenis Alat |
| 4 | jLabel4 | Text | Ketentuan Kontrak |
| 5 | jLabel5 | Text | Harga Jaminan |
| 6  7 | jLabel6  jTextField1 | Text | Pemulihan Jaminan |
| Name | txtNama |
| 7  8 | jLabel7  jTextField2 | Text |  |
| Name | txtJenisAlat |
| 8  9 | jTextField1  jTextField3 | Text |  |
| Name | txtKontrak |
| 9 | jTextField2 | Text |  |
| 10 | jTextField4 | Name | txtHarga |
| Text |  |
| 11 | jTextField5 | Name | txtPemulihan |
| Text |  |
| 12 | jButton1 | Name | btnCetak |
| Text | Cetak |
| 13 | jButton2 | Name | btnClose |
| 14 | Jbutton3 | Text | Close |
| 14 | jTextArea | Name | DaftarJaminan |
| 16 | JRadiobutton1 | Text |  |
| 15 | jComboBox | Model | [0] = Aman  [1] = Rusak |
| 18 | JCekbox1 | Name | cmb\_pilih |
| **No** | **Nama Komponen** | **Properti** | **Value** |
| 1 | jLabel1 | Text | Jaminan Barang |
| 2 | jLabel2 | Text | Nama |
| 3 | jLabel3 | Text | Jenis Alat |
| 4 | jLabel4 | Text | Ketentuan Kontrak |
| 5 | jLabel5 | Text | Harga Jaminan |
| 6 | jLabel6 | Text | Pemulihan Jaminan |
| 7 | jTextField1 | Name | txtNama |
| 25 | jTable | Text |  |
| 8 | jTextField2 | Name | txtJenisAlat |
|  |  | Text |  |

*Source code* pada tombol tambah*:*

|  |
| --- |
| private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {          // TODO add your handling code here:          insert();      } |

*Source code* pada tombol Ubah*:*

|  |
| --- |
| private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {          // TODO add your handling code here:          update();      } |

*Source code* pada tombol hapus*:*

|  |
| --- |
| private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {          // TODO add your handling code here:          delete();      } |

*Source code* pada tombol cari*:*

|  |
| --- |
| private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {          // TODO add your handling code here:          cari();      } |

*Source code* pada tombol batal*:*

|  |
| --- |
| private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {          // TODO add your handling code here:          batal();      } |

Tampilan Hasil:

A screenshot of a computer

Description automatically generated

### Gambar 10.10 Tampilan Hasil GUI\_Jaminan

Analisa:

*Source code* GUI\_Jaminan memanggil GUI\_Jaminan menggunakan objek gpb lalu ditampilkan menggunakan perintah show dan GUI\_Jaminan diatas menerapkan Polimorfisme lalu menggunakan casting bertipe downcasting, lalu ada combobox yang berfungsi untuk memilih inputan dan akan berubah sesuai dengan inputan yang dipilih dan akan menampilkan *message* apabila telah menekan *button* rekap.

## Kesimpulan

1. Exception merupakan kondisi yang tidak diinginkan yang dapat terjadi selama eksekusi program.
2. Exception Handling merupakan mekanisme yang diperlukan dalam menangani error yang terjadi pada saat runtime (program berjalan) atau yang lebih dikenal dengan sebutan runtime error. Secara umum, adanya kesalahan / error yang terjadi pada program pada saat runtime.
3. Basis data (database) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.
4. MySQL adalah salah satu DBMS (Database Management System) yang menggunakan bahasa SQL (Structured Query Language) untuk mengelola informasi dalam basis data.