Nama : La Ode Muhammad Gazali
NIM : 222212696
Kelas : 2KS2

# TUGAS PRA-PERTEMUAN 11 PEMROGRAMAN BERORIENTASI OBJEK

# (Database Access/JDBC)

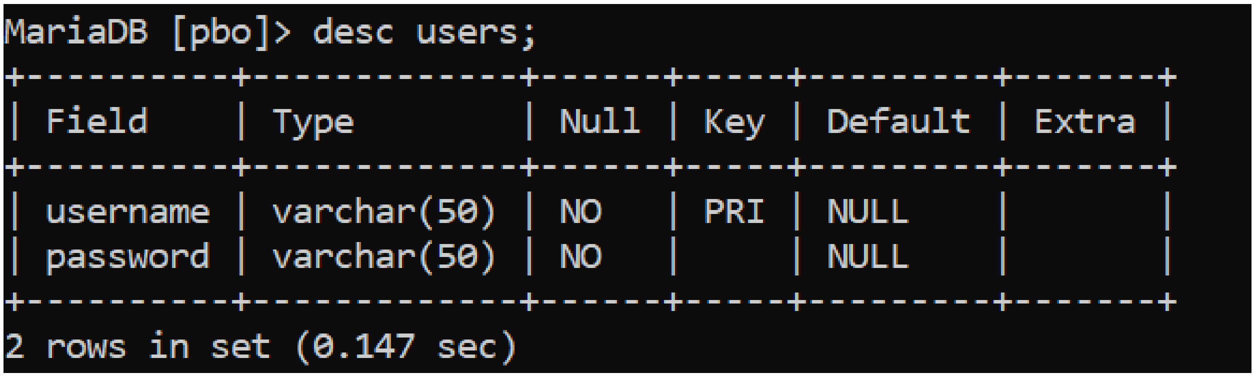
## Penugasan

Merujuk kembali pada modul praktikum (Modul 8 dan Modul 9) :

- Buat agar yang dapat melakukan login adalah username dan password yang telah tersimpan di database
- Simpan data input ke database
- Ketika aplikasi dijalankan, jika data sudah pernah diinput sebelumnya, maka data tersebut akan langsung ditampilkan pada jTable
- Tambahkan fitur untuk menghapus data dan memodifikasi data yang telah diinput

## Penyelesaian

Melanjutkan praktikum 10, setelah membuat database pbo dan tabel mehasiswa, selanjutnya untuk penanganan login perlu dibuat tabel users seperti berikut:



```
MariaDB [pbo]> desc users;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(50) | NO   | PRI | NULL    |       |
| password | varchar(50) | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.147 sec)
```

Lalu perlu dilakukan penyesuaian dibeberapa file dan panel

## Database.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
```

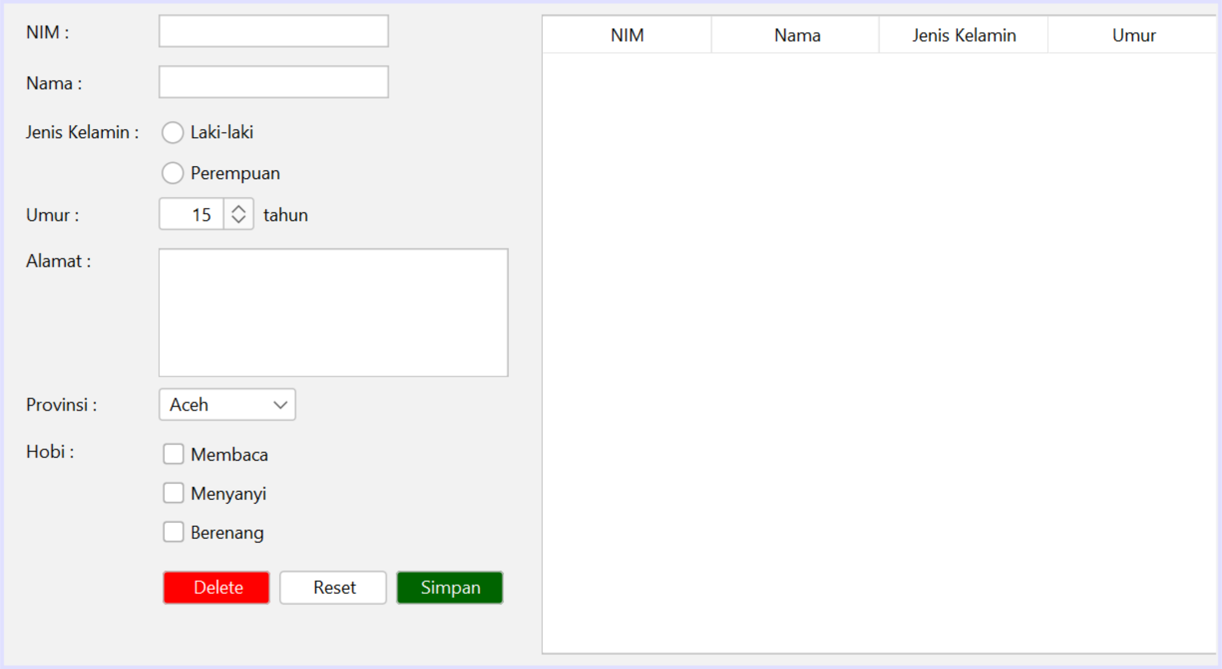
```java
package prapertemuan11;

import javax.swing.*;
import java.io.Serializable;
import java.sql.*;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

/**
 *
 * @author U53R
 */
public class Database implements Serializable {
    public static Database instance;

    private ArrayList<Mahasiswa> data = new ArrayList<>();
    private Database() {}

    public static synchronized Database getInstance() {
        if(instance == null) {
            instance = new Database();
        }
        return instance;
    }

    public void insertMahasiswa(Mahasiswa mahasiswa) throws SQLException {
        Connection conn = DBConn.getConnection();
        try {
            String sql = "INSERT INTO mahasiswa VALUES (?, ?, ?, ?, ?, ?, ?)";
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, mahasiswa.getNim());
            pstmt.setString(2, mahasiswa.getNama());
            pstmt.setString(3, mahasiswa.getJenisKelamin());
            pstmt.setInt(4, mahasiswa.getUmur());
            pstmt.setString(5, mahasiswa.getAlamat());
            pstmt.setString(6, mahasiswa.getProvinsi());
            pstmt.setString(7, String.join(",", mahasiswa.getHobi()));

            pstmt.executeUpdate();
        } catch (SQLException e) {
            throw new RuntimeException(e);
        } finally {
            if(conn != null) {
                conn.close();
            }
        }
    }

    public void updateMahasiswa(Mahasiswa mahasiswa) throws SQLException {
        Connection conn = DBConn.getConnection();
        try {
            String sql = "UPDATE mahasiswa SET nama = ?, jenis_kelamin = ?, umur
= ?, alamat = ?, provinsi = ?, hobi = ? WHERE nim = ?";
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, mahasiswa.getNama());
            pstmt.setString(2, mahasiswa.getJenisKelamin());
            pstmt.setInt(3, mahasiswa.getUmur());
            pstmt.setString(4, mahasiswa.getAlamat());
            pstmt.setString(5, mahasiswa.getProvinsi());
            pstmt.setString(6, String.join(",", mahasiswa.getHobi()));
            pstmt.setString(7, mahasiswa.getNim());

            pstmt.executeUpdate();
        } catch (SQLException e) {
            throw new RuntimeException(e);
        } finally {
            if(conn != null) {
                conn.close();
            }
        }
    }

    public Mahasiswa getMahasiswa(String nim) throws SQLException {
        Connection conn = DBConn.getConnection();
        Mahasiswa mhs = null;
        try {
            String sql = "SELECT * FROM mahasiswa WHERE nim=? LIMIT 1";
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, nim);

            ResultSet res = pstmt.executeQuery();
            while (res.next()) {
                mhs = new Mahasiswa();
                mhs.setNim(res.getString("nim"));
                mhs.setNama(res.getString("nama"));
                mhs.setJenisKelamin(res.getString("jenis_kelamin"));
                mhs.setUmur(res.getInt("umur"));
                mhs.setAlamat(res.getString("alamat"));
                mhs.setProvinsi(res.getString("provinsi"));
                mhs.setHobi(new
ArrayList<>(Arrays.asList(res.getString("hobi").split(","))));
            }
        } catch (SQLException e) {
            throw new RuntimeException(e);
        } finally {
            if (conn != null) {
                conn.close();
            }
        }
        return mhs;
    }

    public void deleteMahasiswa(String nim) throws SQLException {
        Connection conn = DBConn.getConnection();
        Mahasiswa mhs = new Mahasiswa();
        try {
            String sql = "DELETE FROM mahasiswa WHERE nim=?";
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, nim);

            pstmt.executeUpdate();
        } catch (SQLException e) {
            throw new RuntimeException(e);
        } finally {
            if (conn != null) {
                conn.close();
            }
        }
    }

    public List<Mahasiswa> getListMahasiswa() throws SQLException {
        List<Mahasiswa> mhsList = new ArrayList<>();
        Connection conn = DBConn.getConnection();
        try {
            String sql = "SELECT * FROM mahasiswa";
            Statement stmt = conn.createStatement();
            ResultSet res = stmt.executeQuery(sql);
            while (res.next()) {
                Mahasiswa mhs = new Mahasiswa();
                mhs.setNim(res.getString("nim"));
                mhs.setNama(res.getString("nama"));
                mhs.setJenisKelamin(res.getString("jenis_kelamin"));
                mhs.setUmur(res.getInt("umur"));
                mhs.setAlamat(res.getString("alamat"));
                mhs.setProvinsi(res.getString("provinsi"));
                mhs.setHobi(new
ArrayList<>(Arrays.asList(res.getString("hobi").split(","))));

                mhsList.add(mhs);
            }
        } catch (SQLException e) {
            throw new RuntimeException(e);
        } finally {
            if (conn != null) {
                conn.close();
            }
        }
        return mhsList;
    }
}
```

### LoginPanel.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to edit
this template
 */
package prapertemuan11;

import javax.swing.*;
import java.net.PasswordAuthentication;
import java.sql.*;
import BCrypt.BCrypt;
import com.sun.tools.javac.Main;
import java.util.ArrayList;
import java.util.Arrays;
import static java.util.regex.Pattern.matches;

/**
 *
 * @author U53R
 */
public class LoginPanel extends javax.swing.JPanel {

    private JScrollPane contentScrollPane;
    /**
     * Creates new form LoginPanel
     * @param contentScrollPane
     */
    public LoginPanel(JScrollPane contentScrollPane) {
        this.contentScrollPane = contentScrollPane;
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        userTextField = new javax.swing.JTextField();
        loginButton = new javax.swing.JButton();
        passwordField = new javax.swing.JPasswordField();

        jLabel1.setText("Username :");

        jLabel2.setText("Password :");

        loginButton.setText("Login");
        loginButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                loginButtonActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);
        this.setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(212, 212, 212)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alig
nment.TRAILING)
                    .addComponent(loginButton)
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.LEADING)
                            .addComponent(jLabel2)
                            .addComponent(jLabel1))
                        .addGap(18, 18, 18)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.LEADING, false)
                            .addComponent(userTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 253, Short.MAX_VALUE)
                            .addComponent(passwordField))))
                .addContainerGap(216, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(186, 186, 186)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alig
nment.LEADING)
                    .addComponent(userTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel1)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlaceme
nt.UNRELATED)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.BASELINE)
                            .addComponent(jLabel2)
                            .addComponent(passwordField,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))))
                .addGap(18, 18, 18)
                .addComponent(loginButton)
                .addContainerGap(186, Short.MAX_VALUE))
        );
    }// </editor-fold>//GEN-END:initComponents

    private void loginButtonActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_loginButtonActionPerformed
        // TODO add your handling code here:
        boolean check = false;
        try {
            check = auth(userTextField.getText(),
String.valueOf(passwordField.getPassword()));
```

```
        } catch (SQLException e) {
            System.err.println("Gagal auth");
        }
        if (check) {
            MainFrame.afterLogin();
            contentScrollPane.setViewportView(new HomePanel());
        } else {
            userTextField.setText("");
            passwordField.setText("");
            JOptionPane.showMessageDialog(this, "Akun tidak ditemukan.", "Error",
JOptionPane.ERROR_MESSAGE);
        }
    }//GEN-LAST:event_loginButtonActionPerformed

    private boolean auth(String username, String password) throws SQLException {
        Connection conn = DBConn.getConnection();
        boolean check = false;
        try {
            String sql = "SELECT * FROM users WHERE username=? LIMIT 1";
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, username);

            ResultSet res = pstmt.executeQuery();
            while(res.next()) {
                if (matches(password, res.getString("password"))) {
                    check = true;
                }
            }
        } catch (SQLException e) {
            throw new RuntimeException(e);
        } finally {
            if (conn != null) {
                conn.close();
            }
        }
        return check;
    }


    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JButton loginButton;
    private javax.swing.JPasswordField passwordField;
    private javax.swing.JTextField userTextField;
    // End of variables declaration//GEN-END:variables
}
```

Kemudian pada EntryPanel dilakukan penyesuaian dengan menambahkan tombol reset dan delete ketika data dipilih pada panel (Source code terlampir).



NIM :

Nama :

Jenis Kelamin : ○ Laki-laki ○ Perempuan

Umur : 15 tahun

Alamat :

Provinsi : Aceh

Hobi : □ Membaca □ Menyanyi □ Berenang

| NIM | Nama | Jenis Kelamin | Umur |
|---|---|---|---|

Agar tombol tersebut dapat berfungsi menyimpan data pada database, maka pada mainframe perlu ditambahkan beberapa Action Listener dan Action Performed, sebagai berikut:

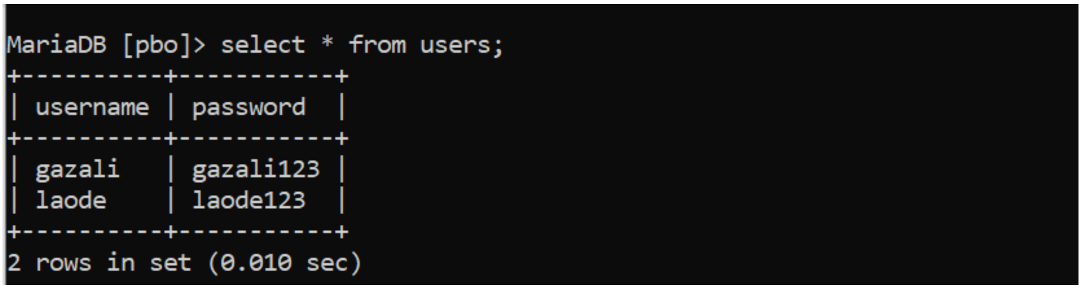
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit
this template
 */
package prapertemuan11;

/**
 *
 * @author U53R
 */
public class MainFrame extends javax.swing.JFrame {

    /**
     * Creates new form MainFrame
     */
    public MainFrame() {
        initComponents();
        beforeLogin();
        contentScrollPane.setViewportView(new LoginPanel(contentScrollPane));
    }

    public void beforeLogin() {
        jMenu1.setEnabled(false);
        jMenu2.setEnabled(false);
    }

    public static void afterLogin() {
        jMenu1.setEnabled(true);
        jMenu2.setEnabled(true);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        contentScrollPane = new javax.swing.JScrollPane();
        jMenuBar1 = new javax.swing.JMenuBar();
        jMenu1 = new javax.swing.JMenu();
        homeMenuItem = new javax.swing.JMenuItem();
        logoutMenuItem = new javax.swing.JMenuItem();
        jMenu2 = new javax.swing.JMenu();
        mahasiswaMenuItem = new javax.swing.JMenuItem();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Data Diri Mahasiswa");

        jMenu1.setText("File");

        homeMenuItem.setText("Home");
        homeMenuItem.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                homeMenuItemActionPerformed(evt);
            }
        });
        jMenu1.add(homeMenuItem);

        logoutMenuItem.setText("Logout");
        logoutMenuItem.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                logoutMenuItemActionPerformed(evt);
            }
        });
        jMenu1.add(logoutMenuItem);

        jMenuBar1.add(jMenu1);

        jMenu2.setText("Entri");

        mahasiswaMenuItem.setText("Mahasiswa");
        mahasiswaMenuItem.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                mahasiswaMenuItemActionPerformed(evt);
            }
        });
        jMenu2.add(mahasiswaMenuItem);

        jMenuBar1.add(jMenu2);

        setJMenuBar(jMenuBar1);

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(contentScrollPane,
javax.swing.GroupLayout.PREFERRED_SIZE, 825,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(contentScrollPane,
javax.swing.GroupLayout.DEFAULT_SIZE, 464, Short.MAX_VALUE)
        );

        pack();
```

```
    }// </editor-fold>//GEN-END:initComponents

    private void homeMenuItemActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_homeMenuItemActionPerformed
        // TODO add your handling code here:
        contentScrollPane.setViewportView(new HomePanel());
    }//GEN-LAST:event_homeMenuItemActionPerformed

    private void mahasiswaMenuItemActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_mahasiswaMenuItemActionPerformed
        // TODO add your handling code here:
        contentScrollPane.setViewportView(new EntryPanel());
    }//GEN-LAST:event_mahasiswaMenuItemActionPerformed

    private void logoutMenuItemActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_logoutMenuItemActionPerformed
        // TODO add your handling code here:
        beforeLogin();
        contentScrollPane.setViewportView(new LoginPanel(contentScrollPane));
    }//GEN-LAST:event_logoutMenuItemActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the
default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(jav
a.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(jav
a.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(jav
a.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(jav
a.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new MainFrame().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JScrollPane contentScrollPane;
    private javax.swing.JMenuItem homeMenuItem;
    private static javax.swing.JMenu jMenu1;
    private static javax.swing.JMenu jMenu2;
    private javax.swing.JMenuBar jMenuBar1;
    private javax.swing.JMenuItem logoutMenuItem;
    private javax.swing.JMenuItem mahasiswaMenuItem;
    // End of variables declaration//GEN-END:variables
}
```

## Simulasi Kerja GUI

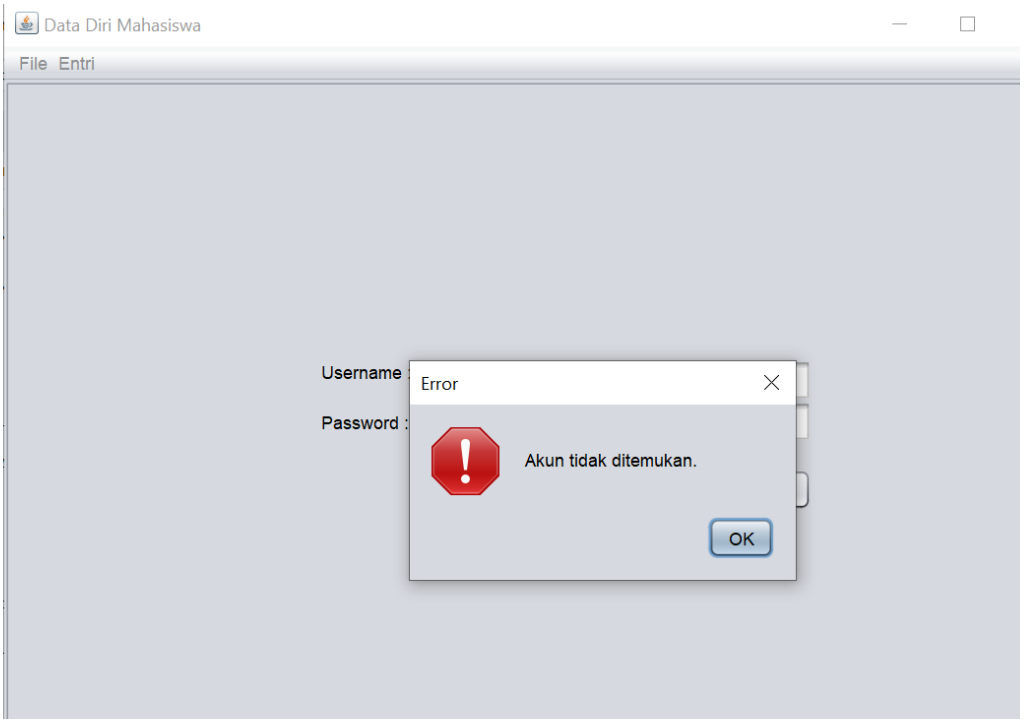
### Proses Login

Misal kita telah mempunyai table users sebagai berikut:



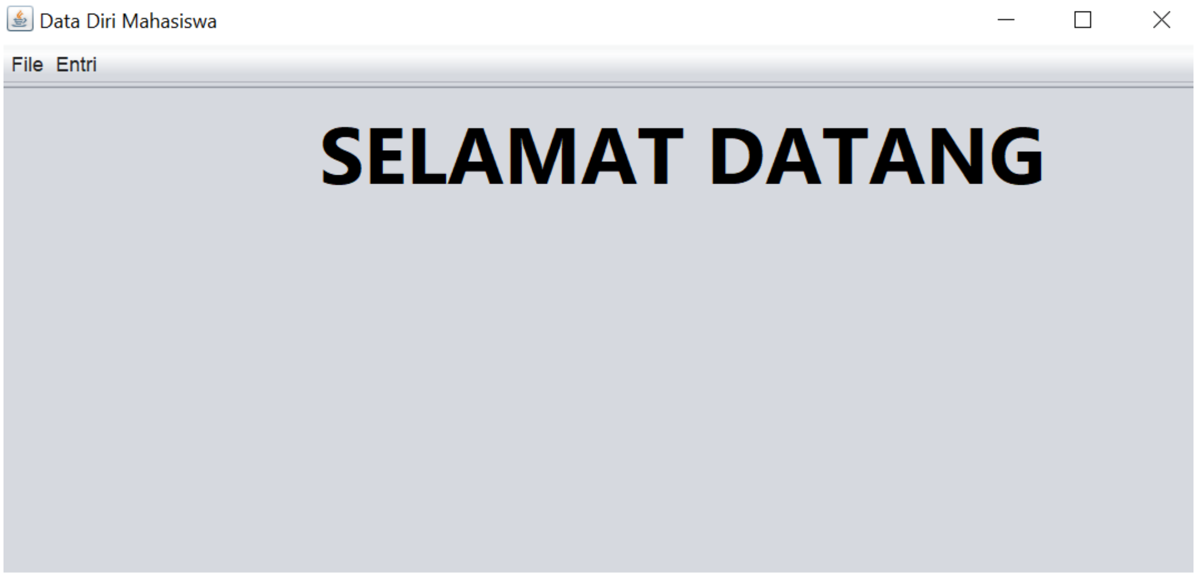
```
MariaDB [pbo]> select * from users;
+----------+-----------+
| username | password  |
+----------+-----------+
| gazali   | gazali123 |
| laode    | laode123  |
+----------+-----------+
2 rows in set (0.010 sec)
```

Jika username atau password yang diinputkan pada form login salah maka akan muncul window error



Akun tidak ditemukan.

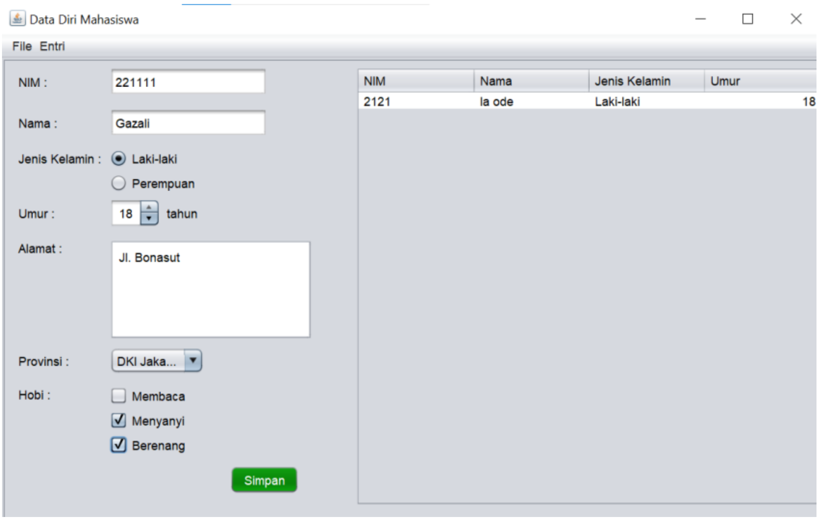
Namun, ketika data username dan password benar maka akan langsung diarahkan ke laman home



SELAMAT DATANG

### Proses Input Data

Kemudian jika ke menu entri maka, informasi data mahsiswa yang telah diinputkan sebelumnya akan tetap muncul, karena telah diambil dari database.



NIM : 221111

Nama : Gazali

Jenis Kelamin : ● Laki-laki ○ Perempuan

Umur : 18 tahun

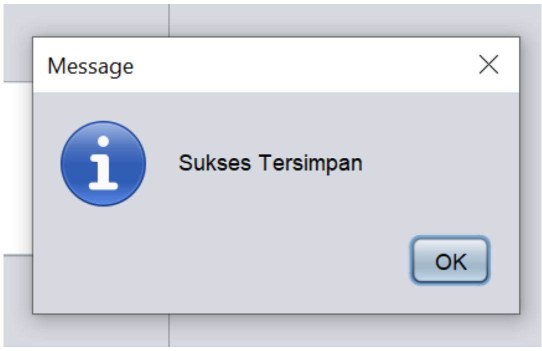
Alamat : Jl. Bonasut

Provinsi : DKI Jaka...

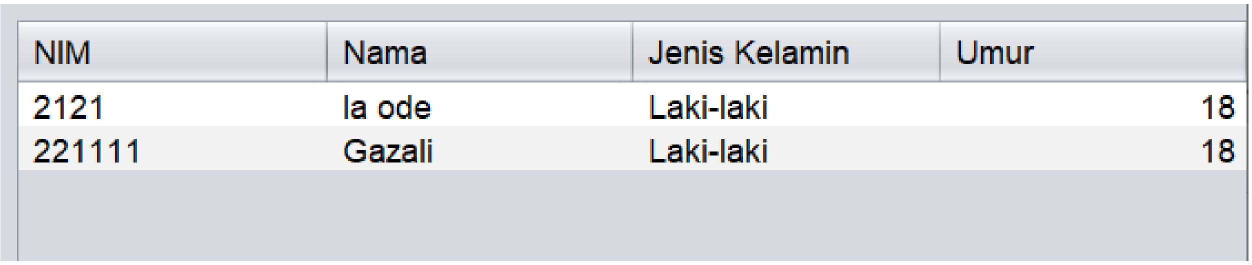
Hobi : □ Membaca ☑ Menyanyi ☑ Berenang

| NIM | Nama | Jenis Kelamin | Umur |
|---|---|---|---|
| 2121 | la ode | Laki-laki | 18 |

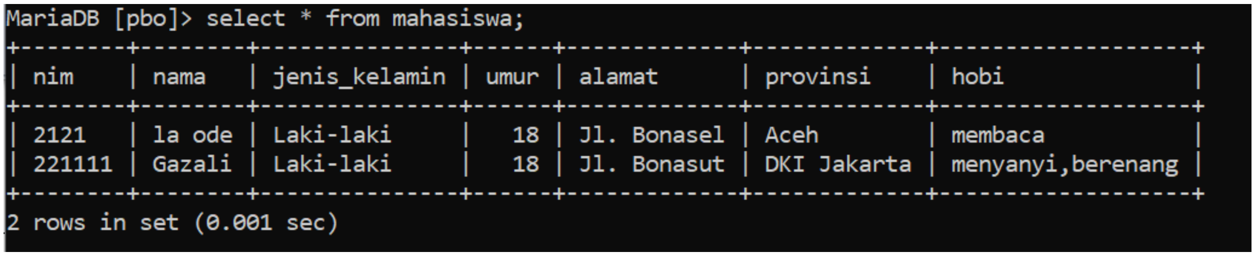
Dan ketika entry data baru berhasil maka akan muncul Message dan data otomatis muncul pada panel tabel dan database, sebagai berikut:



Sukses Tersimpan



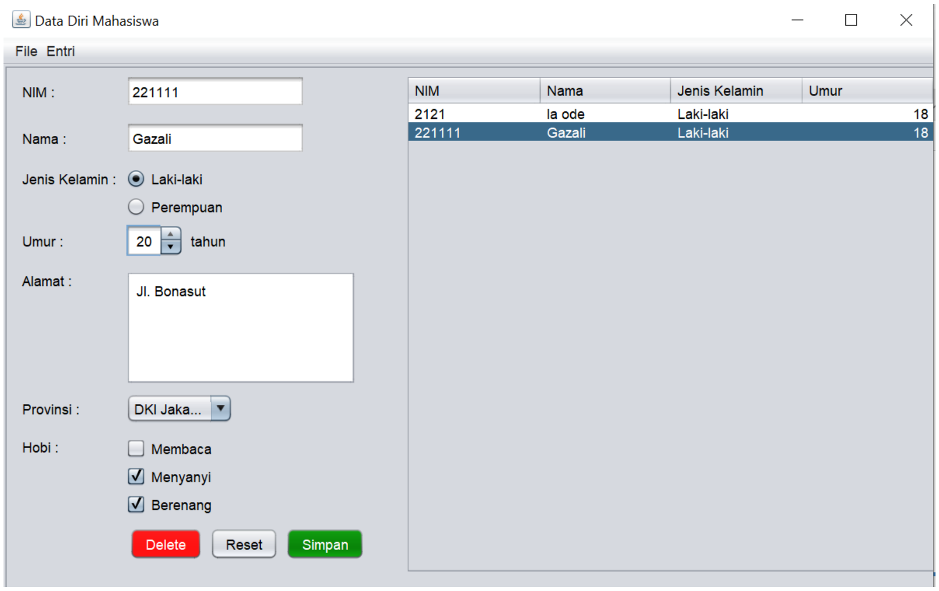
| NIM | Nama | Jenis Kelamin | Umur |
|---|---|---|---|
| 2121 | la ode | Laki-laki | 18 |
| 221111 | Gazali | Laki-laki | 18 |



```
MariaDB [pbo]> select * from mahasiswa;
+--------+--------+---------------+------+-------------+-------------+-------------------+
| nim    | nama   | jenis_kelamin | umur | alamat      | provinsi    | hobi              |
+--------+--------+---------------+------+-------------+-------------+-------------------+
| 2121   | la ode | Laki-laki     |   18 | Jl. Bonasel | Aceh        | membaca           |
| 221111 | Gazali | Laki-laki     |   18 | Jl. Bonasut | DKI Jakarta | menyanyi,berenang |
+--------+--------+---------------+------+-------------+-------------+-------------------+
2 rows in set (0.001 sec)
```

**Reset (Edit)**

Misal ingin mngedit data kedua dengan mengganti umur dari 18 menjadi 20, dapat dilakukan dengan memilih data yang diedit, kemudian mengganti kolom isian yang ingin diedit, kemudian klik reset



NIM : 221111

Nama : Gazali

Jenis Kelamin : ● Laki-laki ○ Perempuan

Umur : 20 tahun

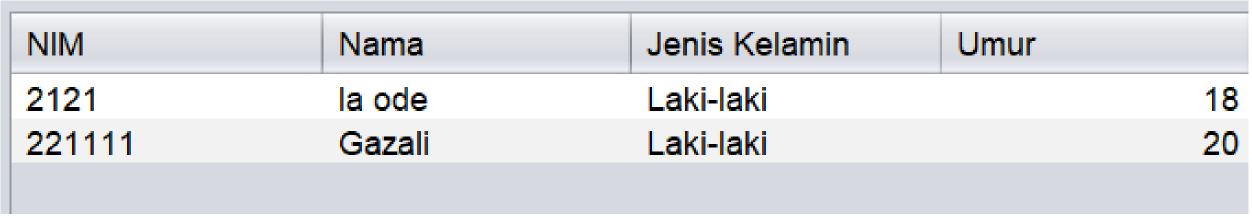
Alamat : Jl. Bonasut

Provinsi : DKI Jaka...

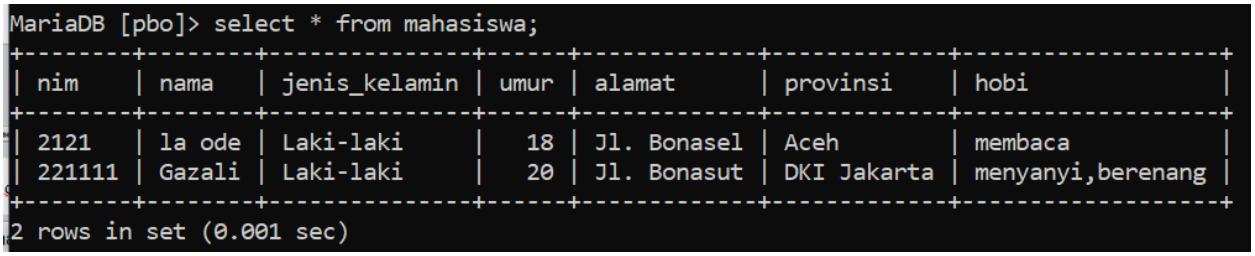
Hobi : □ Membaca ☑ Menyanyi ☑ Berenang

| NIM | Nama | Jenis Kelamin | Umur |
|---|---|---|---|
| 2121 | la ode | Laki-laki | 18 |
| 221111 | Gazali | Laki-laki | 18 |

Maka data akan berhasil diedit dan muncul pada panel tabel dan database



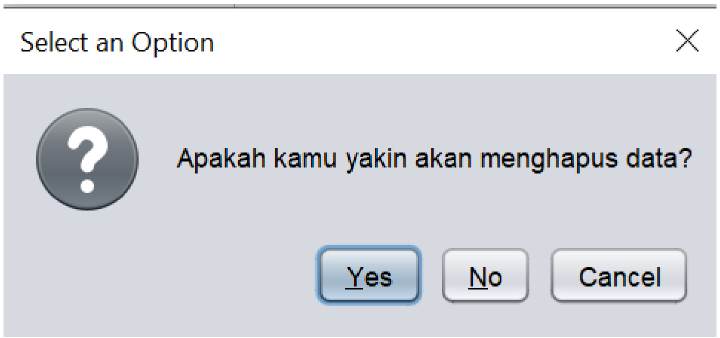
| NIM | Nama | Jenis Kelamin | Umur |
|---|---|---|---|
| 2121 | la ode | Laki-laki | 18 |
| 221111 | Gazali | Laki-laki | 20 |



```
MariaDB [pbo]> select * from mahasiswa;
+--------+--------+---------------+------+-------------+-------------+-------------------+
| nim    | nama   | jenis_kelamin | umur | alamat      | provinsi    | hobi              |
+--------+--------+---------------+------+-------------+-------------+-------------------+
| 2121   | la ode | Laki-laki     |   18 | Jl. Bonasel | Aceh        | membaca           |
| 221111 | Gazali | Laki-laki     |   20 | Jl. Bonasut | DKI Jakarta | menyanyi,berenang |
+--------+--------+---------------+------+-------------+-------------+-------------------+
2 rows in set (0.001 sec)
```

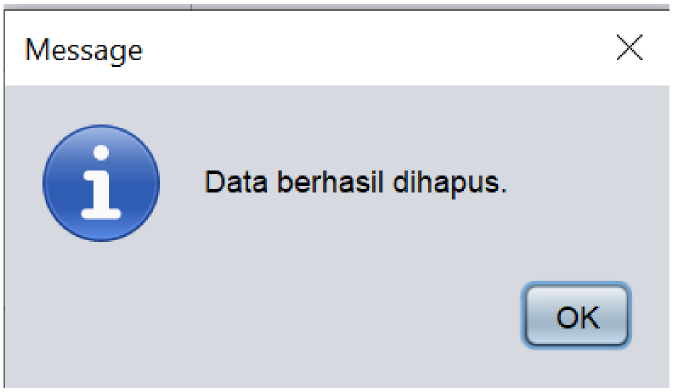
**Delete (Hapus)**

Untuk melakukan penghapusan data bisa langsung memilih data yang ingin dihapus dan klik tombol hapus. Maka akan muncul peringatan sebagai berikut:

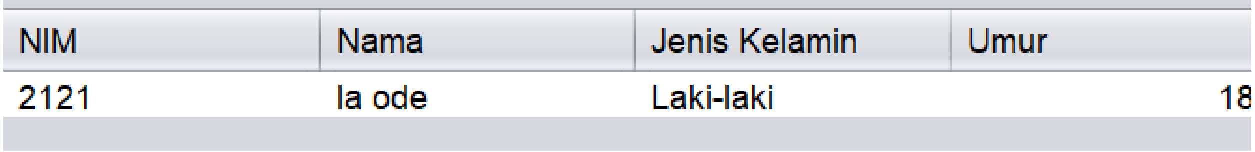


Apakah kamu yakin akan menghapus data?

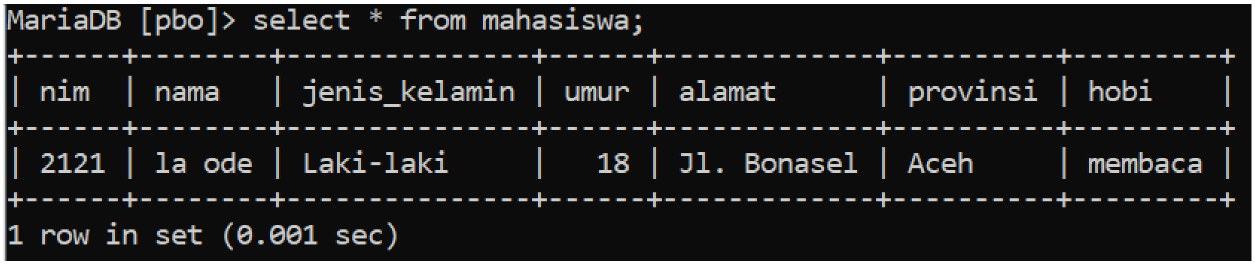
Jika memilih yes maka data akan otomatis terhapus dari database.



Data berhasil dihapus.



| NIM | Nama | Jenis Kelamin | Umur |
|---|---|---|---|
| 2121 | la ode | Laki-laki | 18 |



```
MariaDB [pbo]> select * from mahasiswa;
+------+--------+---------------+------+--------------+----------+---------+
| nim  | nama   | jenis_kelamin | umur | alamat       | provinsi | hobi    |
+------+--------+---------------+------+--------------+----------+---------+
| 2121 | la ode | Laki-laki     |   18 | Jl. Bonasel  | Aceh     | membaca |
+------+--------+---------------+------+--------------+----------+---------+
1 row in set (0.001 sec)
```