



MI2294
PEMROGRAMAN VISUAL
MODUL PRAKTIKUM

Hanya dipergunakan di lingkungan Telkom Applied Science School



Departemen Teknologi Informasi
Telkom Applied Science School
2013

10 BAB X GUI-DATABASE

10.1 IDENTITAS

Kajian

Memahami pembuatan komponen swing (native-way) dan pengaksesan database (MySQL) dengan bahasa pemrograman java

Topik

1. Menghubungkan antara GUI dan database
2. Menampilkan data database pada JTable
3. Menampilkan data database pada JComboBox

Referensi

1. <http://docs.oracle.com/javase/tutorial/uiswing/index.html>
2. <http://docs.oracle.com/javase/tutorial/jdbc/index.html>

Kompetensi Utama

1. Mampu membuat aplikasi berbasis GUI untuk memanipulasi basis data.
2. Mampu mem-populate isi untuk komponen swing GUI dengan isi dari database.
3. Mampu melakukan CRUD database via aplikasi swing java

Lama Kegiatan Praktikum

1. Kegiatan Mandiri : 2 x 100 menit

Parameter Penilaian

1. Tugas Pendahuluan
2. Jurnal Mandiri

10.2 PRAKTIK

10.2.1 Insert Database—Form GUI

Dengan memanfaatkan modul pengaksesan database yang telah ada sebelumnya, Input ke GUI dengan menerima input user melalui text field, proses menjadi lebih gampang untuk dilakukan.

Kasus: Input ke table DesaNinja yang terdiri dari atribut id_desa, nama, pemimpin.

Urutannya:

- a) Buat 3 textfield di aplikasi gui dan 1 tombol “simpan”
- b) Ambil input user dengan method getText. Simpan input tersebut ke beberapa variabel.
- c) Buat 1 objek dari clas DesaNinja dengan parameter konstruktor adalah variabel yang telah dimiliki sebelumnya.
- d) Panggil method yang merepresentasikan method save() atau method untuk menyimpan data tersebut.
- e) Tampilkan pesan “Berhasil” jika input user berhasil dimasukkan ke database; tampilkan gagal jika gagal

Secara detil: Langkah pertama, buat kelas “KoneksiDB” seperti yang tercantum di modul 5.

```
//Langkah pertama, import package terkait
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

/**
 *
 * @author Eja
 */
public class KoneksiDB {
    // driver JDBC driver dan database URL

    private final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    private final String DB_URL = "jdbc:mysql://localhost/pis12noltiga";
    // Database credentials
    private final String USER = "root";
    private final String PASS = "";
    private Connection conn = null;
```

```

public void bukaKoneksi() {

    boolean flag = false;
    try {
        //Langkah ke-2: Registrasi JDBC
        Class.forName(JDBC_DRIVER);
    } catch (Exception e) {
        System.out.println(e.getMessage());
        flag = true;
    }
    if (!flag) {
        try {
            //Langkah ke-3: buka koneksi
            conn = DriverManager.getConnection(DB_URL, USER, PASS);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
}

public Connection getConn() {
    return conn;
}
}

```

Buatlah Kelas DesaNinja (seperti yang tercantum di modul 5/6—Fokus hanya ke method untuk menyimpan data saja).

```

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

/**
 *
 * @author Eja
 */
public class DesaNinja {

    private String id_desa;
    private String nama;
    private String pemimpin;
    //ambil objek KoneksiDB karena membutuhkan connection dll
    private KoneksiDB kdb = new KoneksiDB();
}

```

```
public DesaNinja(String id_desa, String desa, String pemimpin) {
    this.id_desa = id_desa;
    this.nama = desa;
    this.pemimpin = pemimpin;
}
public boolean masukkanData() throws SQLException {
    //deklarasi connection dan preparedStatement
    Connection dbConnection = null;
    PreparedStatement ps = null;
    int rowAffect = 0;

    String insertTableSQL = "INSERT INTO desa_ninja"
        + "(id_desa, nama, pemimpin) VALUES"
        + "(?,?,?)";
    try {
        //buka koneksi saat objek dari desa ninja dibentuk
        kdb.bukaKoneksi();
        //inisialisasi dbConnection dari objek Connection
        dbConnection = kdb.getConn();

        //Langkah ke 4 bagian 1
        ps = dbConnection.prepareStatement(insertTableSQL);
        ps.setString(1, this.id_desa);
        ps.setString(2, this.nama);
        ps.setString(3, this.pemimpin);
        //langkah 4: eksekusi query
        rowAffect = ps.executeUpdate();

    } catch (Exception e) {
        System.out.println(e.getMessage());
    } finally {
        //langkah ke 6
        ps.close();
    }
    //langkah ke 5
    if (rowAffect > 0) {
        return true;
    } else {
        return false;
    }
}
}
```

Kelas untuk swing java:

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;

/**
 *
 * @author Eja
 */
public class SimpleInsertGui extends JPanel implements ActionListener {

    //3 textfield di deklarasikan
    private JTextField id, nama, pemimpin;
    private JButton eksekusi;

    public SimpleInsertGui() {
        //langkah a, buat 3 textfield (beserta label) dan 1 tombol
        id = new JTextField(20);
        nama = new JTextField(20);
        pemimpin = new JTextField(20);

        //buat label untuk tiap textfield:
        JLabel l1 = new JLabel("No Registrasi Desa: ");
        JLabel l2 = new JLabel("Nama: ");
        JLabel l3 = new JLabel("Pemimpin: ");

        eksekusi = new JButton("Eksekusi");
        eksekusi.setActionCommand("oke");
        eksekusi.addActionListener(this);

        //tambahkan label, text dan button ke panel
        add(l1);add(id);
        add(l2);add(nama);
        add(l3);add(pemimpin);
        add(eksekusi);
    }
}
```

```

public void actionPerformed(ActionEvent e) {
    if(e.getActionCommand().equals("oke")){
        //langkah b, ambil semua isian textfield
        String n = nama.getText();
        String i = id.getText();
        String p = pemimpin.getText();

        //cek kalau isian kosong
        if (n.isEmpty() || i.isEmpty() || p.isEmpty()) {
            JOptionPane.showMessageDialog(this, "Ada Field yang Kosong",
                "Peringatan",JOptionPane.WARNING_MESSAGE);
        }else{
            //langkah c, buat objek desa Ninja
            DesaNinja dn = new DesaNinja(i, n, p);
            boolean status = false;
            try {
                //langkah d, panggil method yang merepresentasikan 'save()'
                status = dn.masukkanData();
            } catch (SQLException ex) {
                Logger.getLogger(SimpleInsertGui.class.getName()).log(Level.SEVERE, null, ex);
            }
            if(status){
                //langkah e, tampilkan pesan berhasil jika eksekusi berhasil
                JOptionPane.showMessageDialog(this, "Eksekusi Berhasil",
                    "Status",JOptionPane.INFORMATION_MESSAGE);
            }else{
                //langkah e, tampilkan pesan gagal jika eksekusi gagal
                JOptionPane.showMessageDialog(this, "Eksekusi Gagal",
                    "Status",JOptionPane.ERROR_MESSAGE);
            }
        }
    }
}

private static void createAndShowGUI() {

    //membuat frame
    JFrame frame = new JFrame("ButtonDemo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //membuat content pane
    SimpleInsertGui newContentPane = new SimpleInsertGui();
    newContentPane.setOpaque(true);
    frame.setContentPane(newContentPane);
}

```

```

//Memunculkan window
frame.pack();
frame.setVisible(true);
frame.setLocationRelativeTo(null);
}
public static void main(String[] args) {

    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
}

```

10.2.2 Select Database—JTable

Mengisi tabel dengan data dari database mempunyai banyak cara. Salah satunya yang dituliskan pada modul ini, dengan tetap memanfaatkan kelas dari modul praktikum sebelumnya (modul 5/6).

Langkah-langkah:

- a) Gunakan Kelas “KoneksiDB” untuk membuka koneksi dan lain-lain
- b) Deklarasikan JTable dengan modelnya pada kelas yang digunakan untuk penempatan swing.
- c) Ikuti langkah pengaksesan data, kecuali langkah ke-5 harus dimodifikasi. Tuliskan method pengaksesan data ini di kelas yang sama dengan kelas untuk menuliskan swing (boleh diletakkan berbeda, tapi yang dituliskan di modul ini adalah cara yang cukup sederhana).
- d) Langkah ke-5 dimodifikasi dengan tidak menampilkan hasil ResultSet tapi memasukkannya ke dalam model JTable.

Lengkapnya:

```

/**
 *
 * @author Eja
 */

import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import java.awt.Dimension;

```



```
import java.awt.GridLayout;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.table.DefaultTableModel;

public class SimpleTableDemoDB extends JPanel {
    //langkah b
    private DefaultTableModel model;
    private JTable table;

    //langkah a
    private KoneksiDB kdb = new KoneksiDB();

    public SimpleTableDemoDB() {
        super(new GridLayout(1,0));

        model = new DefaultTableModel();
        table = new JTable(model);
        table.setPreferredScrollableViewportSize(new Dimension(500, 70));
        table.setFillViewportHeight(true);

        //Membuat scroll pane pada table
        JScrollPane scrollPane = new JScrollPane(table);
        //menambah scroll pane dan table di panel
        add(scrollPane);
        try {
            isiDataTabel();
        } catch (SQLException ex) {
            Logger.getLogger(SimpleTableDemoDB.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    //langkah c
    private void isiDataTabel() throws SQLException {
        Connection dbConnection = null;
        PreparedStatement ps = null;
        ResultSet rs = null;

        String viewTableSQL = "SELECT * FROM desa_ninja";

        try {
            //buka koneksi saat objek dari desa ninja dibentuk
            kdb.bukaKoneksi();
            //inisialisasi dbConnection dari objek Connection
```

```
dbConnection = kdb.getConn();

//Langkah ke 4 bagian 1
ps = dbConnection.prepareStatement(viewTableSQL);

//langkah 4: eksekusi query
rs = ps.executeQuery();

//tentukan header tabel
model.addColumn("Nomor Registrasi Desa");
model.addColumn("Nama Desa");
model.addColumn("Pemimpin");

//langkah ke 5
//ekstrak data dari ResultSet dan masukkan ke table
while (rs.next()) {
    Object[] o = new Object[3];
    //langkah d
    o[0] = rs.getString(1);
    o[1] = rs.getString(2);
    o[2] = rs.getString(3);
    model.addRow(o);
}

} catch (Exception e) {
    System.out.println(e.getMessage());
} finally {
    //langkah ke-6
    ps.close();
}
}

private static void createAndShowGUI() {
    //Membuat frame
    JFrame frame = new JFrame("TableDemo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //Membuat content pane
    SimpleTableDemoDB newContentPane = new SimpleTableDemoDB();
    newContentPane.setOpaque(true);
    frame.setContentPane(newContentPane);

    //menampilkan window
    frame.pack();
    frame.setVisible(true);
}
```

```

public static void main(String[] args) {
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
}

```

10.2.3 Select Database—JComboBox

Mengisi combo box dengan data yang berasal dari database, sama seperti mengisi data dari database untuk diisikan ke table. Ada banyak cara yang bisa dilakukan. Urutan langkah pengerjaan yang dilakukan oleh modul praktikum ini adalah sebagai berikut:

- a) Gunakan Kelas “KoneksiDB” untuk membuka koneksi dan lain-lain
- b) Deklarasikan JComboBox pada kelas yang digunakan untuk penempatan swing.
- c) Ikuti langkah pengaksesan data, kecuali langkah ke-5 harus dimodifikasi. Tuliskan method pengaksesan data ini di kelas yang sama dengan kelas untuk menuliskan swing (alasan yang sama dengan sebelumnya).
- d) Panggil method pengaksesan data tersebut pada konstruktor, sehingga saat ditampilkan pertama kali, combo box sudah menampilkan isi dari database.
- e) Langkah ke-5 dimodifikasi dengan tidak menampilkan hasil ResultSet tapi memasukkannya ke dalam komponen JComboBox.

Lengkapnya sebagai berikut:

```

/**
 *
 * @author Eja
 */
import java.util.logging.Level;
import java.util.logging.Logger;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.*.*;

public class SimpleComboBoxDemo extends JPanel implements ActionListener {

```

```
//langkah a
private KoneksiDB kdb = new KoneksiDB();
//langkah b
private JComboBox kombo;

public SimpleComboBoxDemo() {
    super();
    kombo = new JComboBox();

    try {
        //langkah d
        this.isiDataKombo();
    } catch (SQLException ex) {
        Logger.getLogger(SimpleComboBoxDemo.class.getName()).log(Level.SEVERE, null, ex);
    }
    kombo.addActionListener(this);

    add(kombo);
}

/** Listener combo box */
public void actionPerformed(ActionEvent e) {
    JComboBox cb = (JComboBox) e.getSource();
    int pil = cb.getSelectedIndex();
    if (pil == 0) {
        System.out.println("Tidak ada yang terpilih");
    } else {
        System.out.println(cb.getSelectedItem());
    }
}

//langkah c
private void isiDataKombo() throws SQLException {
    Connection dbConnection = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    String viewTableSQL = "SELECT * FROM desa_ninja";
    try {
        //buka koneksi saat objek dari desa ninja dibentuk
        kdb.bukaKoneksi();
        //inisialisasi dbConnection dari objek Connection
        dbConnection = kdb.getConn();

        //Langkah ke 4 bagian 1
        ps = dbConnection.prepareStatement(viewTableSQL);
```

```
//langkah 4: eksekusi query
rs = ps.executeQuery();

//langkah ke 5
//ekstrak data dari ResultSet

while (rs.next()) {
    //langkah e
    kombo.addItem(rs.getString(2));
}
kombo.setSelectedIndex(0);

} catch (Exception e) {
    System.out.println(e.getMessage());
} finally {
    //langkah ke-6
    ps.close();
}
}
private static void createAndShowGUI() {
    //membuat frame
    JFrame frame = new JFrame("ComboBoxDemo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //membuat content pane
    JComponent newContentPane = new SimpleComboBoxDemo();
    newContentPane.setOpaque(true); //content panes must be opaque
    frame.setContentPane(newContentPane);

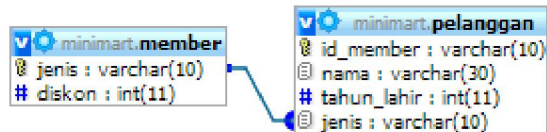
    //Memunculkan window
    frame.pack();
    frame.setVisible(true);
}

public static void main(String[] args) {
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
}
```

10.3 LATIHAN

Diketahui sebuah tabel dengan struktur dan isi sebagai berikut:

Struktur:



Isi:

EDIT	JENIS	DISKON
	gold	2
	platinum	10
row(s) 1 - 2 of 2		

EDIT	ID_MEMBER	NAMA	TAHUN_LAHIR	JENIS
	MEM-001	Melody Nurramdhani Laksani	1992	gold
	MEM-002	Sonya Pandarmawan	1996	gold
	MEM-003	Nabilah Ratna Ayu Azalia	1999	platinum
row(s) 1 - 3 of 3				

Query DDL dan DML untuk tabel di atas adalah sebagai berikut:

Oracle:

```
CREATE table "MEMBER" (  
  "JENIS"  VARCHAR2(10) NOT NULL,  
  "DISKON"  NUMBER,  
  constraint "MEMBER_PK" primary key ("JENIS")  
)  
/  
  
CREATE table "PELANGGAN" (  
  "ID_MEMBER"  VARCHAR2(10),  
  "NAMA"  VARCHAR2(30),  
  "TAHUN_LAHIR"  NUMBER,  
  "JENIS"  VARCHAR2(10),  
  constraint "PELANGGAN_PK" primary key ("ID_MEMBER")  
)  
/  
  
ALTER TABLE "PELANGGAN" ADD CONSTRAINT "PELANGGAN_FK"  
FOREIGN KEY ("JENIS")  
REFERENCES "MEMBER" ("JENIS")  
ON DELETE CASCADE  
  
/
```

MySQL:

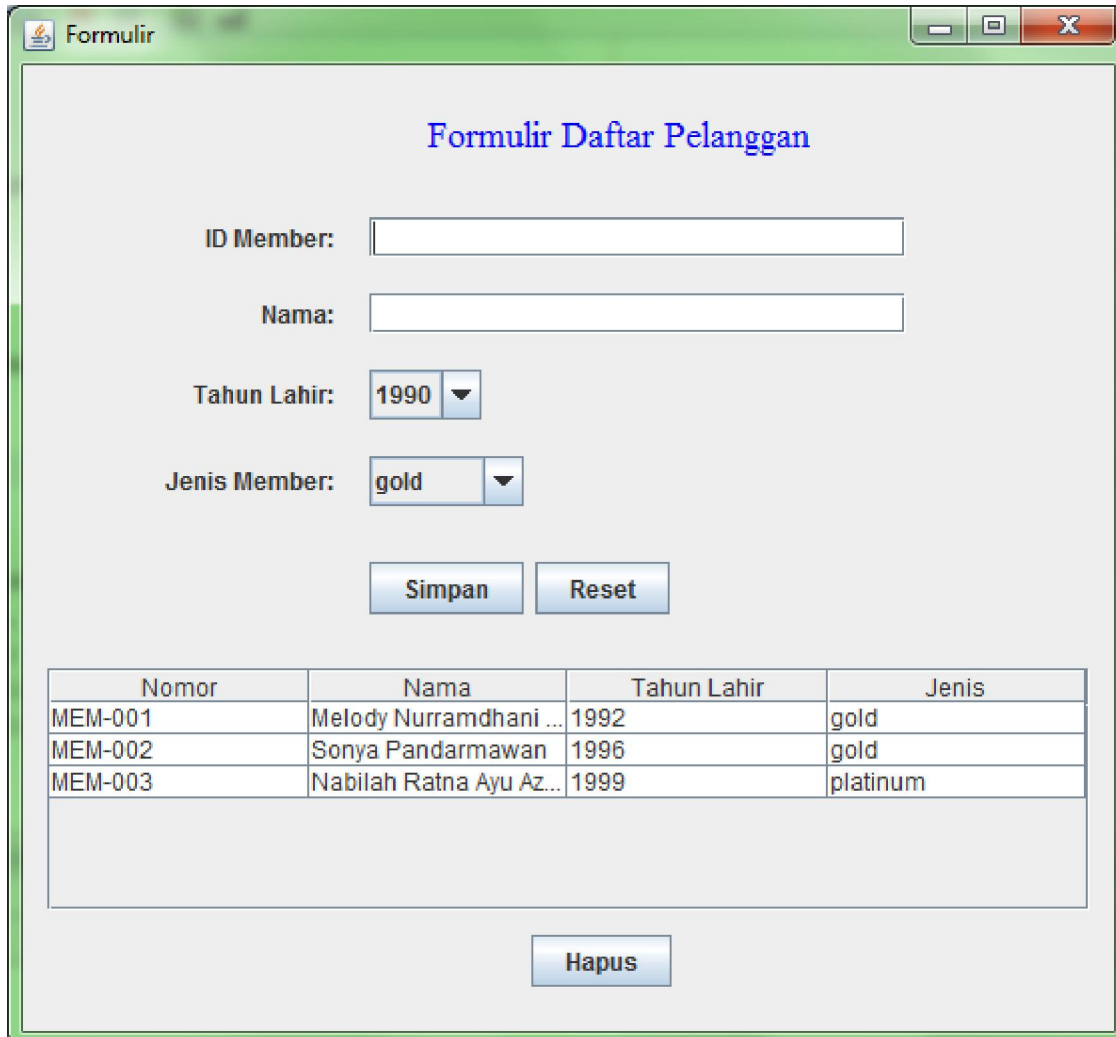
```
CREATE TABLE IF NOT EXISTS `member` (  
  `jenis` varchar(10) NOT NULL,  
  `diskon` int(11) NOT NULL,  
  PRIMARY KEY (`jenis`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
CREATE TABLE IF NOT EXISTS `pelanggan` (  
  `id_member` varchar(10) NOT NULL,  
  `nama` varchar(30) NOT NULL,  
  `tahun_lahir` int(11) NOT NULL,  
  `jenis` varchar(10) NOT NULL,  
  PRIMARY KEY (`id_member`),  
  KEY `jenis` (`jenis`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
ALTER TABLE `pelanggan`  
  ADD CONSTRAINT `pelanggan_ibfk_1` FOREIGN KEY (`jenis`) REFERENCES `member` (`jenis`) ON  
  DELETE CASCADE ON UPDATE CASCADE;
```

Isi:

```
INSERT INTO `minimart`.`member` (`jenis`, `diskon`) VALUES ('gold', '2');  
INSERT INTO `minimart`.`member` (`jenis`, `diskon`) VALUES ('platinum', '10');  
  
INSERT INTO pelanggan (id_member, nama, tahun_lahir, jenis) VALUES ('MEM-001', Melody  
Nurramdhani Laksani', 1992, 'gold');  
  
INSERT INTO pelanggan (id_member, nama, tahun_lahir, jenis) VALUES ('MEM-002', 'Sonya  
Pandarmawan', 1996, 'gold');  
  
INSERT INTO pelanggan (id_member, nama, tahun_lahir, jenis) VALUES ('MEM-003', 'Nabilah Ratna  
Ayu Azalia', 1999, 'platinum');
```


Tugas:

Buatlah sebuah form untuk memasukkan data pelanggan ke database (pilih salah satu Antara oracle/mysql) sebagai berikut:



The screenshot shows a web browser window titled "Formulir". The main heading is "Formulir Daftar Pelanggan". The form contains the following fields and controls:

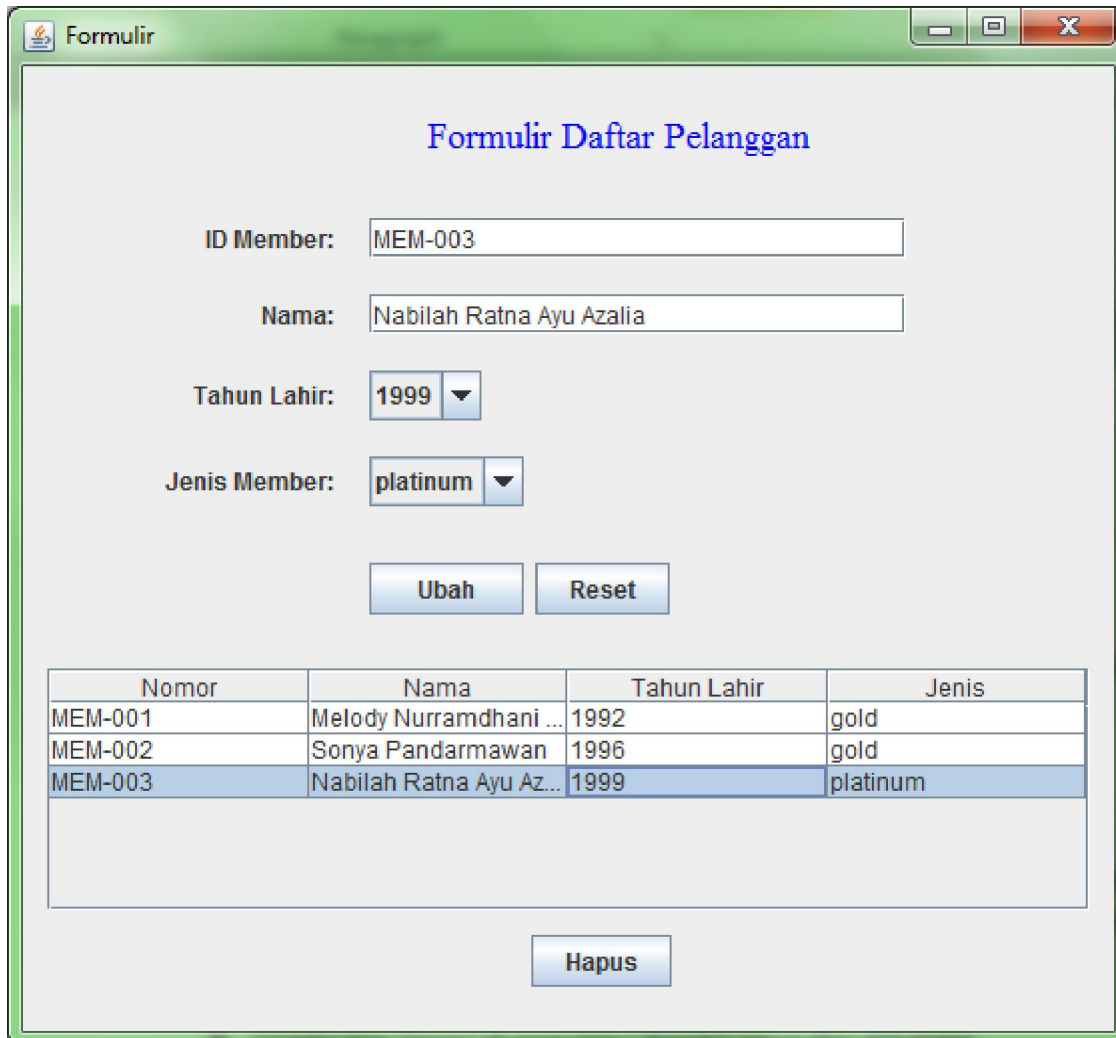
- ID Member:
- Nama:
- Tahun Lahir: (dropdown arrow)
- Jenis Member: (dropdown arrow)
- Buttons: and

Below the form is a table with the following data:

Nomor	Nama	Tahun Lahir	Jenis
MEM-001	Melody Nurramdhani ...	1992	gold
MEM-002	Sonya Pandarmawan	1996	gold
MEM-003	Nabilah Ratna Ayu Az...	1999	platinum

At the bottom of the table area is a button:

Ketika tabel dipilih:



The screenshot shows a web application window titled "Formulir" with a green border. The main heading is "Formulir Daftar Pelanggan". The form contains the following fields:

- ID Member:
- Nama:
- Tahun Lahir: (with a dropdown arrow)
- Jenis Member: (with a dropdown arrow)

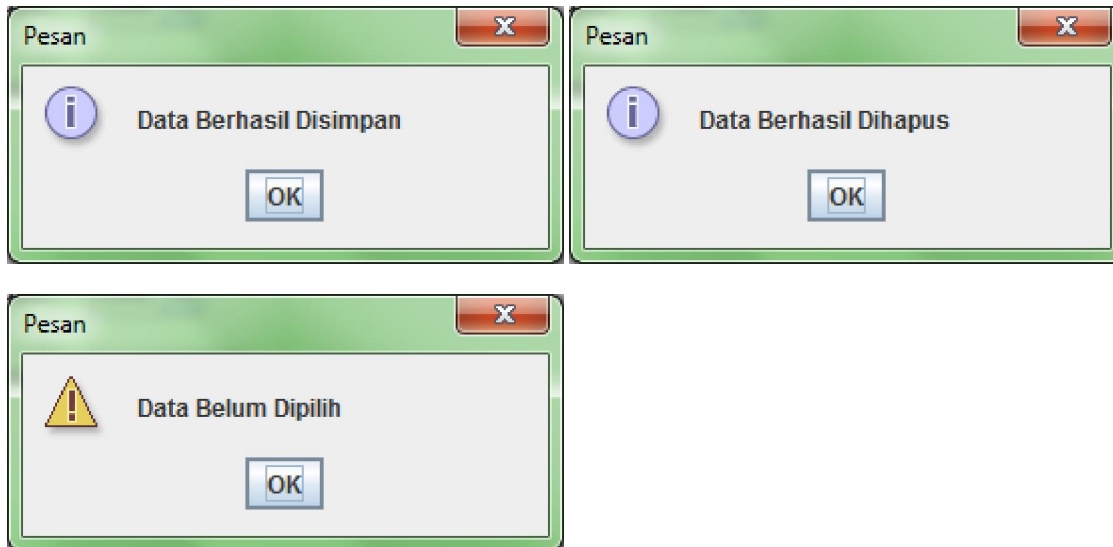
Below the form are two buttons: "Ubah" and "Reset".

At the bottom of the form is a table with the following data:

Nomor	Nama	Tahun Lahir	Jenis
MEM-001	Melody Nurramdhani ...	1992	gold
MEM-002	Sonya Pandarmawan	1996	gold
MEM-003	Nabilah Ratna Ayu Az...	1999	platinum

Below the table is a "Hapus" button.

Pesan yang dimunculkan:



Keterangan:

- Ketika tabel dipilih, setiap textfield berisikan data sesuai baris terpilih
- Tombol "Simpan" berubah menjadi "Ubah" ketika data di tabel dipilih.
- Tabel tidak bisa diubah nilainya dengan meng-klik baris tabel 2x, tapi masih bisa dipilih.
- Tombol "Reset" akan mengosongkan setiap textfield dan mengembalikan tombol ke state "Simpan"
- Setelah menekan tombol "Simpan" atau "Ubah", kembali ke state seperti habis di-reset
- Combo box "tahun lahir" berisi tahun 1990-2000
- Combo box "jenis" di-populate/ diambil isinya dari database
- Boleh menggunakan GUI-Builder (Matisse Builder)