

MI2294
PEMROGRAMAN VISUAL
MODUL PRAKTIKUM

Hanya dipergunakan di lingkungan Telkom Applied Science School



Departemen Teknologi Informasi
Telkom Applied Science School
2013

2 BAB II MATISSE BUILDER 2 (ACTION)

2.1 IDENTITAS

Kajian

Pengenalan Swing Java (Menggunakan Tools--Visual Editor/Matisse Builder)

Topik

1. Penggunaan Matisse Builder
2. Action Listener Komponen Swing

Referensi

1. <http://netbeans.org>

Kompetensi Utama

1. Mahasiswa memahami konsep pemrograman swing
2. Mahasiswa mampu membuat halaman sederhana menggunakan komponen swing dibantu tool gui builder
3. Mahasiswa mampu menambahkan action pada komponen swing

Lama Kegiatan Praktikum

1. Pertemuan Terbimbing : 1 x 50 menit
2. Kegiatan Mandiri : 1 x 50 menit

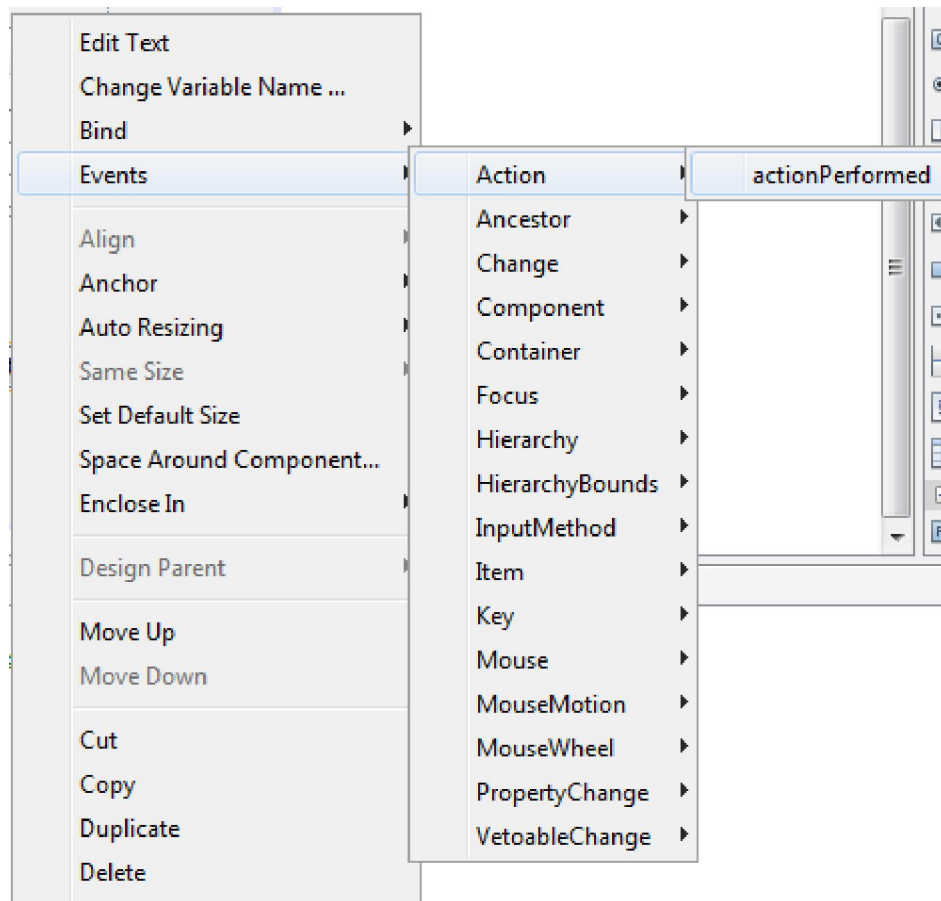
Parameter Penilaian

1. Tugas Pendahuluan
2. Jurnal Mandiri

2.2 EVENT LISTENER

Untuk mendapatkan kesan “responsif” terhadap input user, aplikasi gui yang dibangun menggunakan bahasa java dilengkapi oleh event handler. Event handler didapat dari **interface listener**. Interface listener mempunyai method yang harus dituliskan kembali sebagai method yang harus dijalankan saat mendapat input user.

Dengan menggunakan matisse builder, penambahan event handler menjadi lebih mudah. Tambahkan event dengan meng-klik kanan komponen yang akan ditambahkan event, lalu pilih event apa yang diinginkan (biasanya tipe action dengan method actionPerformed()).



Atau bisa juga dengan meng-klik 2x komponen yang ingin ditambahkan event. Untuk tombol (JButton), event yang terbentuk adalah action → actionPerformed(). Akan muncul bagian source code pada gui seperti berikut. Logika pemrograman dapat dituliskan di method tersebut.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //tuliskan logika pemrograman di blok ini
}
```

2.3 METHOD KOMPONEN SWING

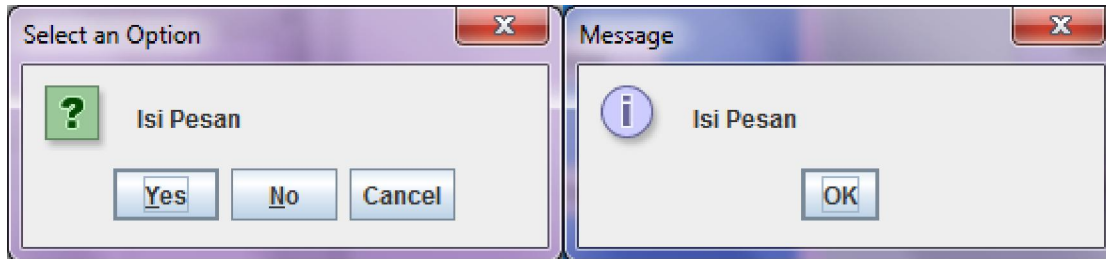
Beberapa komponen swing memiliki method yang bisa digunakan untuk dipakai pada logika pemrograman yang terdapat pada event handler. Beberapa yang paling sering digunakan adalah sebagai berikut:

Method	Komponen	Fungsi
setText("X")	JButton, JTextfield, JLabel	Menge-set tulisan yang muncul pada masing-masing komponen
getText()	JButton, JTextfield, JLabel	Mengambil nilai String yang terdapat pada masing-masing komponen
setToolTipText("X")	JButton, JTextfield, JLabel	Memberikan tooltip pada komponen
setEnabled(false)	JButton, JTextfield, JLabel, JComboBox, JRadioButton, JCheckBox	Meng-enable suatu komponen atau tidak (dapat di-klik atau tidak)
setSelected(true)	JRadioButton, JCheckBox	Membuat masing-masing komponen terpilih atau tidak
isSelected()	JRadioButton, JCheckBox	Menge-cek apakah suatu komponen sedang terpilih atau tidak
setSelectedIndex(4)	JComboBox	Membuat indeks pada angka tertentu sebagai komponen terpilih
getSelectedIndex()	JComboBox	Mengambil indeks terpilih dari komponen. Indeks dimulai dari 0
getSelectedItem()	JComboBox	Mengambil objek terpilih dari komponen. Dapat langsung ditampilkan jika yang terpilih adalah objek String.
insertItemAt(objek, indek)	JComboBox	Memasukkan item pilihan berupa sebuah objek bertipe Object pada index ke indek
setValueAt(Objek, baris, kolom)	JTable	Menge-set table dengan nilai Objek yang bertipe data Object pada baris dan kolom tertentu
getSelectedRow()	JTable	Mengambil indeks baris tabel terpilih. Indeks dimulai dari 0
getSelectedColumn()	JTable	Mengambil indeks kolom tabel terpilih. Indeks dimulai dari 0

2.4 PENGGUNAAN DIALOG (JOPTIONPANE)

Dialog di modul ini merujuk pada sebuah komponen yang muncul sebagai informasi atau peringatan bahkan meminta input user setelah melakukan sesuatu.

Bentuk:



Pada komponen swing, komponen ini bernama JOptionPane. Terdapat 4 buah option pane:

- showConfirmDialog : Untuk memberi konfirmasi yes/no/cancel
- showInputDialog: Untuk menampilkan pop-up sebagai input data
- showMessageDialog: Untuk menampilkan status/pesan dengan 1 tombol "OK".
- showOptionDialog: Gabungan dari ketiga komponen di atas. Ada kasi konfirmasi, meminta input dan menampilkan status/pesan

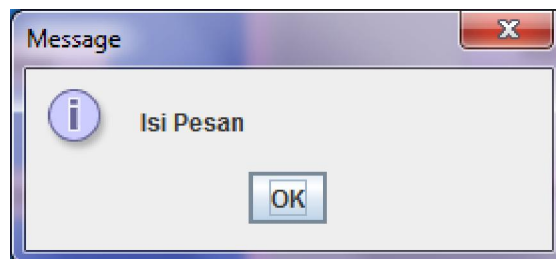
Pada modul praktikum ini hanya dibahas 2 tipe option pane, showConfirmDialog dan showMessageDialog. Setiap option pane dapat dibuat objeknya dengan menggunakan jumlah parameter yang berbeda-beda. Hal ini akan berpengaruh pada tampilan option pane yang dihasilkan.

2.4.1 showMessageDialog

Contoh Kode showMessageDialog dengan 2 parameter:

```
JOptionPane.showMessageDialog(null, "Isi Pesan");
```

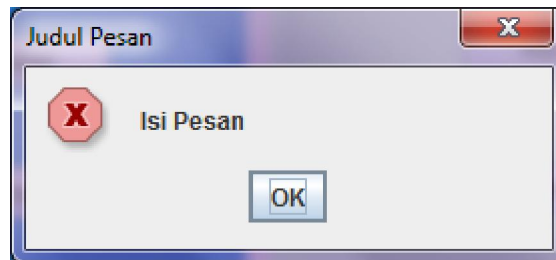
Hasil:







Contoh Kode showMessageDialog dengan 4 parameter:

```
JOptionPane.showMessageDialog(null, "Isi Pesan", "Judul Pesan", JOptionPane.ERROR_MESSAGE);
```

Hasil:



Lambang/symbol/icon yang dihasilkan (silang merah atau huruf “i” dalam lingkaran), tergantung pada parameter terakhir dari contoh kode di atas (untuk kode 4 parameter). Parameter ini disebut “option type”. Jenis-jenis dari option type adalah sebagai berikut:

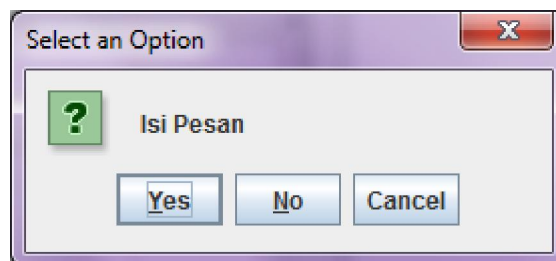
Icon	Code	IDE Value
No icon	JOptionPane.PLAIN_MESSAGE	-1
	JOptionPane.ERROR_MESSAGE	0
	JOptionPane.INFORMATION_MESSAGE	1
	JOptionPane.WARNING_MESSAGE	2
	JOptionPane.QUESTION_MESSAGE	3

2.4.2 showConfirmDialog

Contoh Kode showMessageDialog dengan 2 parameter:

```
JOptionPane.showConfirmDialog(null, "Isi Pesan");
```

Hasil:



Untuk mengambil nilai dari setiap tombol yang ditekan, gunakan kode sebagai berikut:

```
//meminta input yang ditekan sekaligus deklarasi option pane
int hasil = JOptionPane.showConfirmDialog(null, "Isi Pesan");

//membandingkan hasil yang diterima dengan konstanta yang dimiliki option pane
if(hasil == JOptionPane.YES_OPTION){
    System.out.println("Yes!");
}else if(hasil == JOptionPane.NO_OPTION){
    System.out.println("No!");
}else if(hasil == JOptionPane.CANCEL_OPTION){
    System.out.println("Cancel!");
}
```

Pembandingan pesan terdiri dari:

- YES_OPTION: Jika memilih tombol "YES"
- OK_OPTION: Jika memilih tombol "OK"
- NO_OPTION: Jika memilih tombol "NO"
- CLOSED_OPTION: Jika memilih tombol "CLOSED"
- CANCEL_OPTION: Jika memilih tombol "CANCEL"

2.5 PRAKTIK

2.5.1 Penginapan Mine Nakahara

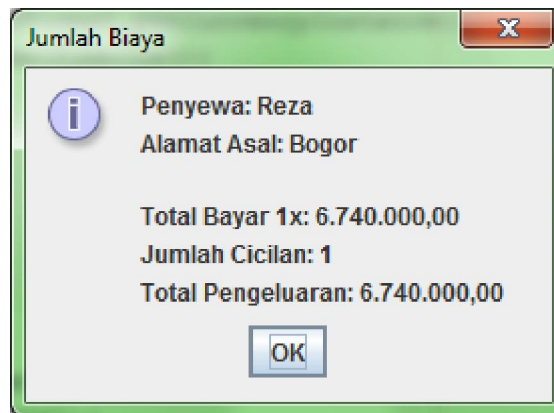
Mine Nakahara merupakan pemilik penginapan terkenal di desa konoha. Nakahara ingin memperluas cakupan bisnisnya ke bidang IT. Salah satu langkah nyata yang ia ambil adalah membuat aplikasi untuk menghitung biaya penginapan bagi para penyewa penginapannya.

Nakahara ingin membuat aplikasi tersebut dengan GUI sebagai berikut:



The screenshot shows a Windows-style application window titled "Mine Nakahara Corps". The window contains a menu bar with "Pilihan" and "Kosongkan". The main area is titled "Aplikasi Perhitungan Biaya Kamar". It features several input fields: "Nama Penyewa:" (text box), "Alamat Asal:" (text box), "Periode" (radio buttons for "Tahunan" and "Bulanan"), "Cicilan" (dropdown menu with "--Pilihan--"), and "Fasilitas" (checkboxes for "Dispenser", "Kipas Angin", and "Televisi"). A "Hitung Biaya" button is located at the bottom center.

Buatlah aplikasi berdasarkan GUI di atas dan ketika tombol "Hitung Biaya" ditekan, maka muncul pesan informasi sebagai berikut:



Dengan informasi:

A screenshot of the main application window titled "Aplikasi Perhitungan Biaya Kamar" from "Mine Nakahara Corps". The window has a "Pilihan" header. The form contains the following fields and controls: "Nama Penyewa:" with a text box containing "Reza"; "Alamat Asal:" with a text box containing "Bogor"; "Periode" with radio buttons for "Tahunan" (selected) and "Bulanan"; "Cicilan" with a dropdown menu showing "1 Tahun"; and "Fasilitas" with checkboxes for "Dispenser" (checked), "Kipas Angin", and "Televisi". A "Hitung Biaya" button is located at the bottom center.

Aturan aksi:

- Jika memilih periode "Tahunan" pilihan cicilan berubah menjadi {"-pilihan-", "3 bulan", "6 bulan", "1 tahun"}
- Jika memilih periode "Bulanan" pilihan cicilan berubah menjadi {"-pilihan-", "mingguan", "bulanan"}

Aturan Perhitungan:

Aturan	Biaya	Aturan	Biaya
Periode Tahunan	Total: Rp. 6.500.000	Cicilan 3 Bulan	Total: total/4 Jumlah cicilan: 4
		Cicilan 6 Bulan	Total: total/2 Jumlah cicilan: 2
		Cicilan 1 tahun	Jumlah cicilan: 1
Periode Bulanan	Total: Rp. 800.000	Cicilan Mingguan	Total: total/3 Jumlah cicilan: 3
		Cicilan Bulanan	Jumlah cicilan: 1
Dispenser		Fasilitas Dispenser Bulanan	Total: Total + 30.000
		Fasilitas Dispenser Tahunan	Total: Total + 240.000
Televisi		Fasilitas Televisi Bulanan/Tahunan	Total: Total + 272.000
Kipas Angin		Fasilitas Kipas Angin Bulanan	Total: Total + 5000
		Fasilitas Kipas Angin Tahunan	Total: Total + 15000

2.5.2 Penyelesaian Contoh Kasus

Langkah pertama, munculkan pilihan pada combo box melalui pilihan radio button. Caranya, klik 2x di radio button pertama (pilihan tahunan), dan window akan memunculkan source dari class tersebut dan membuat method sebagai berikut. Tapi sebelumnya, ubah nama radio button menjadi "radioTahunan" (klik kanan, pilih "change variable name"):

```
private void radioTahunanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
```

Method tersebut berarti menambahkan suatu "aksi" jika radio button dipilih. Dalam blok yang sama dengan baris "// TODO add your handling code here:", tambahkan logika untuk menambahkan komponen combo box (sebelumnya, ubah nama combo box menjadi "comboCicilan").

```
private void radioTahunanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    comboCicilan.removeAllItems();
    comboCicilan.addItem("-Pilihan-");
    comboCicilan.addItem("3 Bulan");
    comboCicilan.addItem("6 Bulan");
    comboCicilan.addItem("1 Tahun");
}
```

Cara lain yang bisa dilakukan untuk menambahkan method tersebut adalah dengan klik kanan di komponen radio button, pilih “Events → Action → actionPerformed”.

Lakukan hal yang sama dengan radio button yang lain (radioBulanan):

```
private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    comboCicilan.removeAllItems();
    comboCicilan.addItem("-Pilihan-");
    comboCicilan.addItem("Mingguan");
    comboCicilan.addItem("Bulanan");
}
```

Setelah itu, lakukan perhitungan saat tombol ditekan. Klik 2x pada tombol (ganti variabel name menjadi “tombolHitung”) untuk menambahkan method actionPerformed:

```
private void tombolHitungActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
```

Tambahkan logika perhitungan biaya dan cara menampilkannya sebagai berikut:

```
//deklarasi variabel untuk menyimpan total biaya
//dan fasilitas yang harus dibayar
double total = 0, fasilitas = 0;

//deklarasi variabel untuk menyimpan total cicilan
//dan status jika user tidak memilih pilihan
int jumCicilan = 0;
boolean salah = false;
```

```
//ambil harga dasar bulanan atau tahunan
if (radioTahunan.isSelected()) {
    total = 6500000;
    //pilihan cicilan
    if (comboCicilan.getSelectedIndex() == 1) {
        jumCicilan = 4;
    } else if (comboCicilan.getSelectedIndex() == 2) {
        jumCicilan = 2;
    } else if (comboCicilan.getSelectedIndex() == 3) {
        jumCicilan = 1;
    } else {
        salah = true;
    }

    //pilihan fasilitas
    if (cekDispenser.isSelected()) {
        fasilitas = fasilitas + 240000;
    }
    if (cekTv.isSelected()) {
        fasilitas = fasilitas + 272000;
    }
    if (cekKipas.isSelected()) {
        fasilitas = fasilitas + 15000;
    }

} else if (radioBulanan.isSelected()) {
    total = 800000;
    //pilihan cicilan
    if (comboCicilan.getSelectedIndex() == 1) {
        jumCicilan = 3;
    } else if (comboCicilan.getSelectedIndex() == 2) {
        jumCicilan = 1;
    } else {
        salah = true;
    }

    //pilihan fasilitas
    if (cekDispenser.isSelected()) {
        fasilitas = fasilitas + 30000;
    }
    if (cekTv.isSelected()) {
        fasilitas = fasilitas + 272000;
    }
    if (cekKipas.isSelected()) {
        fasilitas = fasilitas + 5000;
    }
}
```

```

    }

    } else {
        salah = true;
    }
    //hitung total biaya per-1x bayar ditambah fasilitas
    total = total + fasilitas;
    total = total / jumCicilan;

    if (teksNama.getText().isEmpty() || teksAlamat.getText().isEmpty()) {
        salah = true;
    }

```

Cek input sudah benar atau belum, jika sudah tampilkan total harga, jika belum tampilkan pesan peringatan untuk mengecek masukan yang ada

```

//cek apakah radio button sudah dipilih
//cek apakah combo box sudah dipilih
if (salah) {
    JOptionPane.showMessageDialog(this, "Periksa Nama & Alamat,\n"+
        "Pilihan Bulanan/Tahunan,\n"
        + "dan Jumlah Cicilan", "Status", JOptionPane.WARNING_MESSAGE);
} else {
    //format tampilan uang
    DecimalFormat df = (DecimalFormat) DecimalFormat.getCurrencyInstance();
    DecimalFormatSymbols dfs = new DecimalFormatSymbols();
    dfs.setCurrencySymbol("");
    dfs.setMonetaryDecimalSeparator(',');
    dfs.setGroupingSeparator('.');
    df.setDecimalFormatSymbols(dfs);

    //format tampilan dialog
    String pesan = "";
    pesan = "Penyewa: " + teksNama.getText();
    pesan += "\nAlamat Asal: " + teksAlamat.getText() + "\n\n";
    pesan += "Total Bayar 1x: " + df.format(total);
    pesan += "\nJumlah Cicilan: " + jumCicilan;
    pesan += "\nTotal Pengeluaran: " + df.format(total * jumCicilan);
    JOptionPane.showMessageDialog(this, pesan, "Jumlah Biaya",
        JOptionPane.INFORMATION_MESSAGE);
}

```

2.5.3 Solusi Lengkap

```
private void tombolHitungActionPerformed(java.awt.event.ActionEvent evt) {
    //deklarasi variabel untuk menyimpan total biaya
    //dan fasilitas yang harus dibayar
    double total = 0, fasilitas = 0;

    //deklarasi variabel untuk menyimpan total cicilan
    //dan status jika user tidak memilih pilihan
    int jumCicilan = 0;
    boolean salah = false;

    //ambil harga dasar bulanan atau tahunan
    if (radioTahunan.isSelected()) {
        total = 6500000;
        //pilihan cicilan
        if (comboCicilan.getSelectedIndex() == 1) {
            jumCicilan = 4;
        } else if (comboCicilan.getSelectedIndex() == 2) {
            jumCicilan = 2;
        } else if (comboCicilan.getSelectedIndex() == 3) {
            jumCicilan = 1;
        } else {
            salah = true;
        }
        //pilihan fasilitas
        if (cekDispenser.isSelected()) {
            fasilitas = fasilitas + 240000;
        }
        if (cekTv.isSelected()) {
            fasilitas = fasilitas + 272000;
        }
        if (cekKipas.isSelected()) {
            fasilitas = fasilitas + 15000;
        }
    } else if (radioBulanan.isSelected()) {
        total = 800000;
        //pilihan cicilan
        if (comboCicilan.getSelectedIndex() == 1) {
            jumCicilan = 3;
        } else if (comboCicilan.getSelectedIndex() == 2) {
            jumCicilan = 1;
        } else {
            salah = true;
        }
    }
}
```

```
//pilihan fasilitas
if (cekDispenser.isSelected()) {
    fasilitas = fasilitas + 30000;
}
if (cekTv.isSelected()) {
    fasilitas = fasilitas + 272000;
}
if (cekKipas.isSelected()) {
    fasilitas = fasilitas + 5000;
}

} else {
    salah = true;
}

//hitung total biaya per-1x bayar ditambah fasilitas
total = total + fasilitas;
total = total / jumCicilan;

if (teksNama.getText().isEmpty() || teksAlamat.getText().isEmpty()) {
    salah = true;
}

//cek apakah radio button sudah dipilih
//cek apakah combo box sudah dipilih
if (salah) {
    JOptionPane.showMessageDialog(this, "Periksa Nama & Alamat,\n"+
        "Pilihan Bulanan/Tahunan,\n"
        + "dan Jumlah Cicilan", "Status", JOptionPane.WARNING_MESSAGE);
} else {
    //format tampilan uang
    DecimalFormat df = (DecimalFormat) DecimalFormat.getCurrencyInstance();
    DecimalFormatSymbols dfs = new DecimalFormatSymbols();
    dfs.setCurrencySymbol("");
    dfs.setMonetaryDecimalSeparator(',');
    dfs.setGroupingSeparator('.');
    df.setDecimalFormatSymbols(dfs);
}
```

```
//format tampilan dialog
String pesan = "";
pesan = "Penyewa: " + teksNama.getText();
pesan += "\nAlamat Asal: " + teksAlamat.getText() + "\n\n";
pesan += "Total Bayar 1x: " + df.format(total);
pesan += "\nJumlah Cicilan: " + jumCicilan;
pesan += "\nTotal Pengeluaran: " + df.format(total * jumCicilan);
JOptionPane.showMessageDialog(this, pesan, "Jumlah Biaya",
    JOptionPane.INFORMATION_MESSAGE);
}
```

2.6 LATIHAN

Buatlah GUI sebagai berikut menggunakan komponen swing java:

The screenshot shows a Java Swing window titled "Formulir Data Diri". Inside the window, the title "Formulir Biodata" is displayed in green. Below the title, there are two input fields: "Nama:" followed by a text box, and "Umur:" followed by a smaller text box. Below these fields are two buttons: "Proses" and "Kosongkan". At the bottom of the window, there is a table with two columns labeled "nama" and "umur". Below the table is a "Hapus" button.

Keterangan:

- Masukkan komponen dalam panel; Jangan memasukkan langsung komponen dalam frame
- Terdapat 2 input
- Frame muncul di tengah screen
- Setelah menekan "OK" data input muncul di tabel
- Tabel dapat ditambah dan dihapus tapi tidak dapat diubah
- Jika ada yang salah satu textfield yang kosong, maka program menampilkan pesan kesalahan

Hasil Akhir:

nama	umur
Irfan	21
Eja	24
X	17
Y	20

Latihan Tambahan:

- ✓ Gantilah "Umur", menjadi combo box yang berisi tahun lahir. Tapi nantinya tetap tampilkan umur di table. Beri Option pane untuk masing-masing tombol. Contohnya, setelah menekan tombol "proses" muncul pesan yang menyatakan bahwa data sudah dimasukkan.