

Modul 7 Praktikum Pemrograman Visual

1 IDENTITAS

Kajian

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

Topik

- **Komponen Swing**

Referensi

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

Kompetensi Utama

1. Mahasiswa memahami penggunaan komponen swing untuk membuat aplikasi desktop
2. Mahasiswa mampu membuat aplikasi desktop menggunakan komponen swing

Lama Kegiatan Praktikum

1. Pertemuan Terbimbing : 1 x 100 menit

Parameter Penilaian

-

Penyusun

1. Versi 1.0 : 2013 10 18 : Reza Budiawan
2. -

2 FRAME

Frame merupakan top-level-container dari komponen swing. Terdapat 2 cara untuk mengimplementasikan JFrame pada pemrograman swing java. Cara pertama dengan membuat objek dari class JFrame pada kelas yang mengimplementasikan GUI. Setelah objek JFrame dibentuk, komponen lain dapat ditambahkan. Cara kedua adalah dengan membuat sebuah class yang meng-extends JFrame tersebut.

Cara pertama:

```
package packagecomponent;

import javax.swing.*;

public class FrameDemo {
    private static void createAndShowGUI() {
        JFrame frame = new JFrame("FrameDemo");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //Display the window.
        frame.pack();
        frame.setVisible(true);
    }

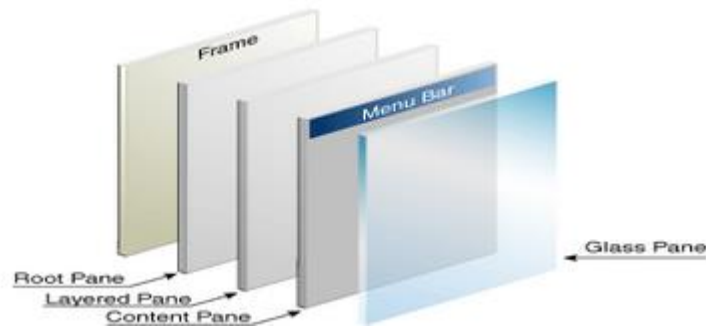
    public static void main(String[] args) {
        createAndShowGUI();
    }
}
```

Cara kedua:

```
public class KelasFrame extends javax.swing.JFrame {

    public KelasFrame() {
        initComponents();
    }
    private void initComponents() {
        //hasil generate tools
    }
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new KelasFrame().setVisible(true);
            }
        });
    }
}
```

Pada pemrograman swing GUI, terdapat hirarki sebagai berikut:



3 PANEL

JPanel merupakan container yang termasuk ke dalam content pane. Content pane merupakan tempat peletakan komponen swing seperti button, textfield dan komponen “swing control” lainnya. Komponen tersebut dapat diletakkan langsung ke dalam JFrame, tapi JFrame bertindak sebagai top-level-container, dan bukan content pane, sehingga lebih baik meletakkan komponen “swing control” pada JPanel dan bukan pada JFrame.

JPanel dapat dideklarasikan sebagai berikut:

```
JPanel p = new JPanel(new BorderLayout());
```

Atau

```
JPanel p = new JPanel();
```

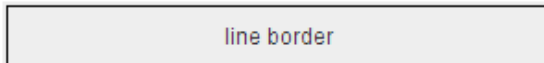
Perbedaan pembuatan objek cara pertama dengan kedua terletak pada inisialisasi layout yang digunakan (Lihat modul praktikum 9). Layout pada pendeklarasian pertama menggunakan Border layout, sedangkan pada cara kedua layout tidak ditentukan. Jika tidak ditentukan, maka panel akan memiliki layout bertipe Flow (default JFrame adalah Flow Layout).

4 BORDER

Merupakan garis pembatas dari sebuah panel atau komponen. Contoh:

```
JPanel pane = new JPanel();  
pane.setBorder(BorderFactory.createLineBorder(Color.black));
```

Hasil dari penulisan kode di atas adalah sebagai berikut:



Untuk menambahkan komponen swing ke content pane/container, digunakan method “add()”.

5 LABEL

Label merupakan komponen untuk menghasilkan “unselectable” gambar dan teks. Pada label dapat diletakkan gambar (objek dari kelas ImageIcon) dan Teks (objek String). Parameter pada konstruktor dapat berisi text dan image saja atau dapat berisi keduanya ditambah posisi tampilan label tersebut. Contoh pembuatan objek label:

```
ImageIcon icon = createImageIcon("images/middle.gif", "a pretty but meaningless splat");  
JLabel label1 = ("Image and Text", icon, JLabel.CENTER);
```

Atau:

```
JLabel label1 = new JLabel("Text-Only Label");  
JLabel label2 = new JLabel(icon);
```

Beberapa method yang digunakan beserta fungsinya yaitu:

Method	Fungsi
setText("X")	Untuk menge-set tulisan teks pada label
getText()	Mengambil tulisan teks pada label
setToolTipText()	Memberikan tooltip pada label

6 BUTTON-MNEMONIC

Merupakan komponen untuk membuat tombol. Kelas yang digunakan adalah JButton. Pada komponen ini, selain keterangan teks, dapat juga ditambahkan image/icon. Contoh pembuatan objek:

```
JButton b2 = new JButton("Tombol B2");
```

Atau:

```
ImageIcon leftButtonIcon = createImageIcon("images/right.gif");  
JButton b1 = new JButton("Tombol B1", leftButtonIcon);
```

Pada button dan menu dapat ditambahkan mnemonic. Mnemonic merupakan penggunaan tombol dengan menggunakan keyboard. Biasanya terdapat 1 huruf yang digunakan sebagai penanda yang berfungsi sama dengan menekan tombol. Mnemonic bekerja dengan menekan "Alt+huruf penanda". Penanda yang biasa digunakan adalah huruf pertama dari keterangan tombol. Contoh cara menge-set mnemonic (contohnya menunjukkan bahwa tombol akan aktif jika ALT+D ditekan):

```
b1.setMnemonic(KeyEvent.VK_D);
```

Beberapa method yang digunakan untuk button dan fungsinya adalah sebagai berikut:

Method	Fungsi
setText("X")	Untuk menge-set tulisan teks pada button
setMnemonic(KeyEvent.VK_M);	Menge-set mnemonic pada tombol
setToolTipText()	Memberikan tooltip pada button
setEnabled(false);	Menge-set button dapat di-klik atau tidak. Parameter "false" menyatakan button disable, dan sebaliknya
setActionCommand	Menge-set nama action performed dari button tersebut

Contoh:

```
/**
 *
 * @author Eja
 */
import javax.swing.AbstractButton;
import javax.swing.JButton;
import javax.swing.JPanel;
import javax.swing.JFrame;
import java.awt.event.KeyEvent;

public class SimpleButtonDemo extends JPanel{
    protected JButton b1;

    public SimpleButtonDemo() {

        b1 = new JButton("Tombol 1");
        b1.setVerticalTextPosition(AbstractButton.CENTER);
        b1.setHorizontalTextPosition(AbstractButton.LEADING); //aka LEFT, for left-to-right locales
        b1.setMnemonic(KeyEvent.VK_D);
        b1.setActionCommand("lakukan");

        //Tambahkan action listener untuk button
        b1.setToolTipText("Bentuk tombol 1");

        //menambahkan button ke kontainer
        //hal ini bisa dilakukan karena class meng-extends JPanel
        add(b1);
    }

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

        //membuat content pane
        SimpleButtonDemo newContentPane = new SimpleButtonDemo();
        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();
        }
    });
}
}

```

7 RADIO BUTTON—BUTTON GROUP

Contoh Konstruktor dan method:

```

//konstruktor radio button
JRadioButton birdButton = new JRadioButton(birdString);
birdButton.setMnemonic(KeyEvent.VK_B);
birdButton.setActionCommand(birdString);
birdButton.setSelected(true);

JRadioButton catButton = new JRadioButton(catString);
catButton.setMnemonic(KeyEvent.VK_C);
catButton.setActionCommand(catString);

//konstruktor button group
ButtonGroup group = new ButtonGroup();
group.add(birdButton);
group.add(catButton);

```

Beberapa method yang digunakan untuk komponen button group:

Method	Fungsi
add(radio1)	Menambahkan radio button untuk menjadi anggota pada grup tersebut
getButtonCount()	Mengembalikan nilai berupa jumlah radio button pada grup tersebut
clearSelection()	Menghapus state terpilih dari semua radio button pada grup tersebut
remove()	Menghapus radio button untuk menjadi anggota pada grup tersebut

Beberapa method yang digunakan untuk komponen radio button dan fungsinya adalah sebagai berikut:

Method	Fungsi
setText("X")	Untuk menge-set tulisan teks pada radio button
getText()	Untuk meng-get tulisan teks pada radio button
setMnemonic(KeyEvent.VK_M);	Menge-set mnemonic pada radio button
setToolTipText()	Memberikan tooltip pada radio button
setEnabled(false);	Menge-set button dapat di-klik atau tidak. Parameter "false" menyatakan radio button disable, dan sebaliknya
setSelected(true)	Menge-set radio button apakah mempunyai state dipilih saat pertama kali dijalankan atau tidak.
setActionCommand	Menge-set nama action performed dari radio button tersebut
isSelected()	Mengecek apakah radio button sedang dipilih atau tidak

Contoh:

```
import java.awt.event.*;
import javax.swing.*;

/**
 *
 * @author Eja
 */
public class SimpleRadioButtonDemo extends JPanel{

    static String birdString = "Bird";
    static String catString = "Cat";
    static String rabbitString = "Rabbit";
    JRadioButton birdButton, catButton, rabbitButton;

    public SimpleRadioButtonDemo() {
        super();

        //Membuat Radio Button
        birdButton = new JRadioButton(birdString);
        birdButton.setMnemonic(KeyEvent.VK_B);
        birdButton.setActionCommand(birdString);
        birdButton.setSelected(true);
```



```

catButton = new JRadioButton(catString);
catButton.setMnemonic(KeyEvent.VK_C);
catButton.setActionCommand(catString);

rabbitButton = new JRadioButton(rabbitString);
rabbitButton.setMnemonic(KeyEvent.VK_R);
rabbitButton.setActionCommand(rabbitString);

//Membuat Grup Button
ButtonGroup group = new ButtonGroup();
//Memasukkan radio button ke grup
group.add(birdButton);
group.add(catButton);
group.add(rabbitButton);

//menambahkan radio button pada panel
this.add(birdButton);
this.add(catButton);
this.add(rabbitButton);
}

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

    //membuat content pane
    SimpleRadioButtonDemo newContentPane = new SimpleRadioButtonDemo();
    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();
        }
    });
}
}

```

8 CHECK BOX

Contoh konstruktor dan method:

```
chinButton = new JCheckBox("Chin");
chinButton.setMnemonic(KeyEvent.VK_C);
chinButton.setSelected(true);
```

Method yang sering digunakan:

Method	Fungsi
setText("X")	Untuk menge-set tulisan teks
getText()	Untuk meng-get tulisan teks
setMnemonic(KeyEvent.VK_M);	Menge-set mnemonic
setToolTipText()	Memberikan tooltip
setEnabled(false);	Menge-set apakah komponen dapat diklik atau tidak. Parameter "false" menyatakan radio button disable, dan sebaliknya
setSelected(true)	Menge-set check box apakah mempunyai state dipilih saat pertama kali dijalankan atau tidak.
setActionCommand	Menge-set nama action performed dari komponen tersebut
isSelected()	Mengecek apakah check box sedang dipilih atau tidak

Contoh:

```
/**
 *
 * @author Eja
 */
import java.awt.event.*;
import javax.swing.*;

public class SimpleCheckBoxDemo extends JPanel {
    JCheckBox miButton;
    JCheckBox kaButton;
    JCheckBox tkButton;

    StringBuffer choices;

    public SimpleCheckBoxDemo() {
        super();
```

```

//Create the check boxes.
miButton = new JCheckBox("MI");
miButton.setMnemonic(KeyEvent.VK_M);
miButton.setSelected(true);

kaButton = new JCheckBox("KA");
kaButton.setMnemonic(KeyEvent.VK_K);
kaButton.setSelected(true);

tkButton = new JCheckBox("TK");
tkButton.setMnemonic(KeyEvent.VK_T);
tkButton.setSelected(true);

//Pilihan pertama secara default
//semua radio button terpilih
choices = new StringBuffer("mkt");

//Tambahkan cek box di panel
this.add(miButton);
this.add(kaButton);
this.add(tkButton);
}

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

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

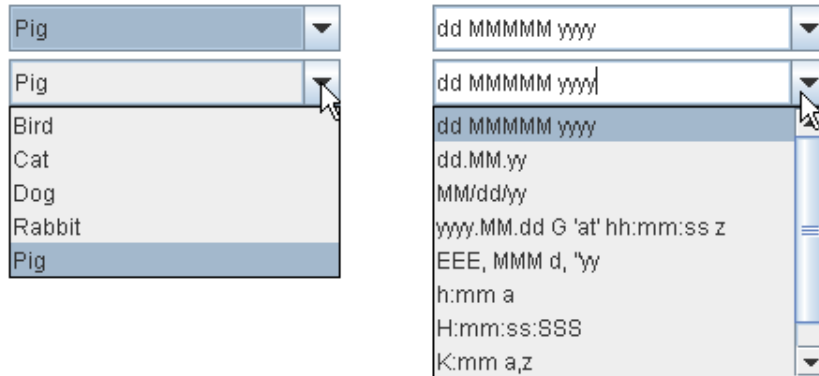
    //Memunculkan window
    frame.pack();
    frame.setVisible(true);
}

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

```

9 COMBO BOX

Combo box merupakan pemilihan menu melalui menu drop down. Sifatnya harus memilih salah satu. Terdapat 2 tipe combo box, editable combo box (kanan) dan uneditable combo box (kiri). Perbedaannya, untuk editable combo box, combo box dapat dituliskan layaknya text field.



Konstruktor dan method:

```
String[] petStrings = { "Bird", "Cat", "Dog", "Rabbit", "Pig" };  
JComboBox petList = new JComboBox(petStrings);
```

Konstruktor memiliki parameter masukan berupa model dari comboBox (bisa dideklarasikan terpisah). Selain itu, Model dapat berupa Array of Object (Object[]) atau Vector. Method yang paling sering digunakan dan fungsinya adalah sebagai berikut:

Method	Fungsi
setEditable(true)	Menge-set tipe combo box, apakah editable combo box atau uneditable combo box
setSelectedIndex(4)	Menge-set indeks item pada combo box. Item dihitung dari nilai 0. (jika seperti contoh di atas, maka "Pig" yang dipilih)
getSelectedItem()	Mengembalikan objek terpilih dari combo box. Jika ingin dimasukkan ke dalam String, harus dilakukan casting terlebih dahulu
getSelectedIndex()	Mengembalikan index item terpilih
getItemAt(3)	Mengembalikan item pada index yang ditentukan
insertItemAt(objek, indek)	Memasukkan item pilihan berupa sebuah objek bertipe Object pada index ke indek
getItemCount()	Mengembalikan jumlah item pada combo box

Contoh:

```
/**
 *
 * @author Eja
 */
import javax.swing.*;

public class SimpleComboBoxDemo extends JPanel{

    public SimpleComboBoxDemo() {
        super();

        String[] daftarPilihan = {"-Pilih-", "Komputer", "Ekonomi", "Bahasa", "Eksakta"};

        JComboBox pilMK = new JComboBox(daftarPilihan);
        pilMK.setSelectedIndex(0);
        add(pilMK);
    }

    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 MENU

Konstruktor dan method:

```
JMenuBar menuBar = new JMenuBar();
JMenu menu = new JMenu("A Menu");
JMenuItem menuItem = new JMenuItem("A text-only menu item",
    KeyEvent.VK_T);

menuBar.add(menu);
menu.add(menuItem);
menu.addSeparator();
```

Method dari menu, menu bar dan menu item beserta fungsinya:

Method	Komponen	Fungsi
add()	MenuBar, MenuItem	Menambahkan komponen kepada komponen lain
setMnemonic()	MenuBar, MenuItem	
addSeparator()	MenuBar, MenuItem	Menambahkan separator

Contoh:

```
/**
 *
 * @author Eja
 */
import java.awt.*;
import java.awt.event.*;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import javax.swing.JMenuBar;
import javax.swing.KeyStroke;

import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JScrollPane;
import javax.swing.JFrame;

public class SimpleMenuDemo{
    JTextArea output;
    JScrollPane scrollPane;
    String newline = "\n";

    public JMenuBar createMenuBar() {
```

```

JMenuBar menuBar;
JMenu menu, submenu;
JMenuItem menuItem;

//Create the menu bar.
menuBar = new JMenuBar();

//Menu Pertama
menu = new JMenu("Menu Pertama");
menu.setMnemonic(KeyEvent.VK_A);
menu.getAccessibleContext().setAccessibleDescription(
    "The only menu in this program that has menu items");
menuBar.add(menu);

//grup menu item
menuItem = new JMenuItem("Menu Item Pertama",
    KeyEvent.VK_T);
//membuat menu item
//dengan konstruktor yang mengandung mnemonic
menuItem.setAccelerator(KeyStroke.getKeyStroke(
    KeyEvent.VK_1, ActionEvent.ALT_MASK));
menuItem.getAccessibleContext().setAccessibleDescription(
    "Tidak Melakukan Apapun");
menu.add(menuItem);

menuItem = new JMenuItem("Menu Item Kedua");
menuItem.setMnemonic(KeyEvent.VK_B);
menu.add(menuItem);

menu.addSeparator();
menuItem = new JMenuItem("Menu Item Ketiga");
menuItem.setMnemonic(KeyEvent.VK_D);
menu.add(menuItem);

//Menu kedua di Menu Bar
menu = new JMenu("Menu Kedua");
menu.setMnemonic(KeyEvent.VK_N);
menu.getAccessibleContext().setAccessibleDescription(
    "Menunya tidak melakukan apapun");
menuBar.add(menu);

return menuBar;
}

public Container createContentPane() {
    //membuat panel
    JPanel contentPane = new JPanel(new BorderLayout());

```

```

contentPane.setOpaque(true);

//membuat text area dan
//menambahkan scroll pane pada komponen tsb
output = new JTextArea(5, 30);
output.setEditable(false);
scrollPane = new JScrollPane(output);

//menambahkan text area ke content pane
//perhatikan yang ditambahkan adalah objek scroll pane
contentPane.add(scrollPane, BorderLayout.CENTER);

return contentPane;
}

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

    //membuat objek class dan membuat menu bar
    //menambahkan menu bar ke frame
    SimpleMenuDemo demo = new SimpleMenuDemo();
    frame.setJMenuBar(demo.createMenuBar());
    frame.setContentPane(demo.createContentPane());

    //menampilkan window
    frame.setSize(450, 260);
    frame.setVisible(true);
}

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

```


11 TABLE—SCROLL BAR

Konstruktor dan method:

```
JTable table = new JTable(new MyTableModel());  
table.setFillViewportHeight(true);  
JScrollPane scrollPane = new JScrollPane(table);  
add(scrollPane);
```

Konstruktor dari Jtable mempunyai parameter masukan dengan tipe data AbstractTableModel. Selain itu juga dapat mempunyai parameter masukan berupa array of Object sebagai row yang akan ditampilkan dan kolom.

```
JTable table = new JTable(rowData, columnNames)
```

Selain konstruktor dengan contoh di atas, terdapat juga bentuk konstruktor dengan parameter masukan lainnya, seperti vector atau jumlah baris dan kolom yang ditampilkan.

Method yang paling sering digunakan dan fungsinya:

Method	Fungsi
setModel(x)	Menge-set model dari JTable
setValueAt(Objek, baris, kolom)	Menge-set table dengan nilai Objek yang bertipe data Object pada baris dan kolom tertentu
getRowCount()	Mengembalikan jumlah baris yang ditampilkan di tabel
getSelectedColumn()	Mengambil indeks kolom tabel terpilih. Indeks dimulai dari 0
getSelectedRow()	Mengambil indeks baris tabel terpilih. Indeks dimulai dari 0
setPreferredSize()	Menge-set ukuran viewport dari objek jtable
setFillViewportHeight(true)	Menge-set apakah table “mengisi penuh” container tempat objek jtable berada.

Selain menggunakan objek dari table, manipulasi isi dan bentuk table juga bisa dilakukan via model jtable (lihat modul praktikum 10).

Contoh penggunaan:

```
/**
 *
 * @edit by Eja
 */

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.AbstractTableModel;
import java.awt.Dimension;
import java.awt.GridLayout;

public class TableDemo extends JPanel {
    private boolean DEBUG = false;

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

        JTable table = new JTable(new MyTableModel());
        table.setPreferredScrollableViewportSize(new Dimension(500, 70));
        table.setFillsViewportHeight(true);

        //Membuat scroll pane pada table
        JScrollPane scrollPane = new JScrollPane(table);

        //menambah scroll pane dan table di panel
        add(scrollPane);
    }

    class MyTableModel extends AbstractTableModel {
        private String[] columnNames = {"First Name",
            "Last Name",
            "Sport",
            "# of Years",
            "Vegetarian"};

        private Object[][] data = {
            {"Kathy", "Smith",
            "Snowboarding", new Integer(5), false},
            {"John", "Doe",
            "Rowing", new Integer(3), true},
            {"Sue", "Black",
            "Knitting", new Integer(2), false},
            {"Jane", "White",
            "Speed reading", new Integer(20), true},
            {"Joe", "Brown",
```

```

        "Pool", new Integer(10), false}
    };

    public int getColumnCount() {
        return columnNames.length;
    }

    public int getRowCount() {
        return data.length;
    }

    public String getColumnName(int col) {
        return columnNames[col];
    }

    public Object getValueAt(int row, int col) {
        return data[row][col];
    }

    /*
     * Dipergunakan untuk me-render bentuk table.
     * Jika tidak meng-implement method berikut,
     * kolom terakhir akan bernilai true/false saja
     * bukan berbentuk check box
     */
    @Override
    public Class getColumnClass(int c) {
        return getValueAt(0, c).getClass();
    }

    /*
     * Digunakan untuk mengubah table menjadi dapat diubah nilainya
     */
    @Override
    public boolean isCellEditable(int row, int col) {
        //Hanya dapat mengubah isi table kolom 1 dan kolom 0
        if (col < 2) {
            return false;
        } else {
            return true;
        }
    }

    @Override
    public void setValueAt(Object value, int row, int col) {
        data[row][col] = value;
        fireTableCellUpdated(row, col);
    }

```

```

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

    //Cmembuat content pane
    TableDemo newContentPane = new TableDemo();
    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();
        }
    });
}
}
}

```

12 TEXT FIELD

Konstruktor:

```

JTextField entry = new JTextField();
JTextField entry2 = new JTextField(25);

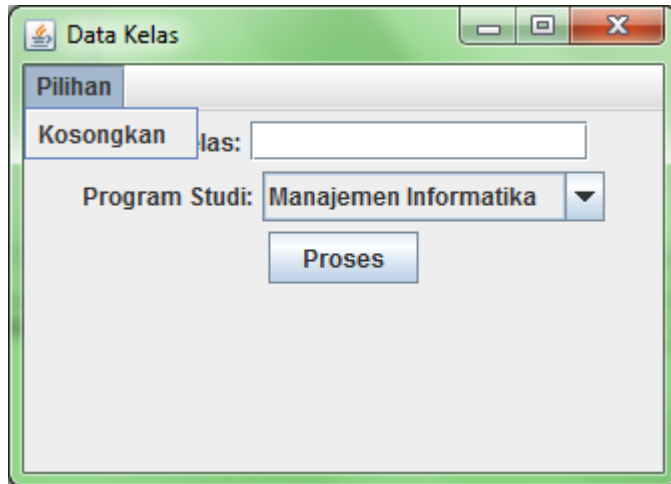
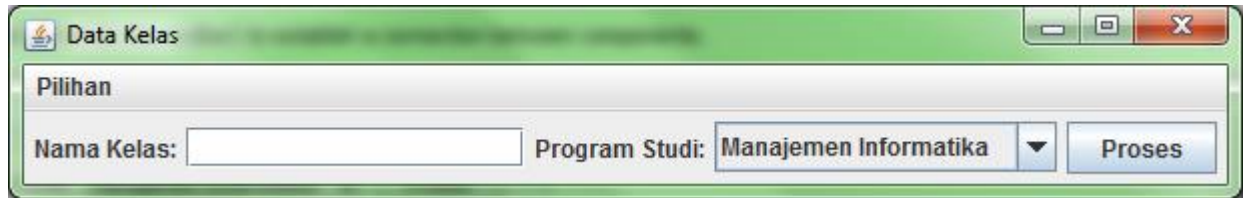
```

Method yang paling sering digunakan dan fungsinya:

Method	Fungsi
setText("x")	Memasukkan text pada komponen
getText()	Mengambil text dari komponen

13 CONTOH KASUS

Buatlah GUI sebagai berikut tanpa menggunakan gui builder:



Pada menu "Pilihan" terdapat menu item "Kosongkan". Gunakan layout bawaan masing-masing container.

14 PENYELESAIAN CONTOH KASUS

Pertama, tentukan konsep GUI yang mau dibuat. Apakah kelas akan meng-extends JFrame atau meng-extends JPanel. Hal ini berpengaruh untuk melihat kelas tersebut bertugas sebagai container pane atau top-level-container.

Di penyelesaian ini, kelas-nya bertugas sebagai top-level-container, sedangkan JPanel sebagai container pane akan dideklarasikan dengan membuat objek barunya.

```
import javax.swing.JFrame;
/**
 *
 * @author EJA
 */
public class DataKelasGUI extends JFrame{

}
```

Deklarasikan semua komponen GUI yang ada. Perhatikan bahwa aplikasi ini memiliki komponen:

- 1 menu bar dengan 1 menu dan 1 menu item
- 2 label (labelKelas & labelProdi)
- 1 text field (teksKelas)
- 1 combo box (comboProdi)
- 1 button (tombolProses)

Sehingga, pada kelas dapat ditambahkan sebagai berikut:

```
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JTextField;

/**
 *
 * @author EJA
 */
public class DataKelasGUI extends JFrame{

    private JLabel labelKelas, labelProdi;
    private JTextField teksKelas;
    private JComboBox comboProdi;
    private JButton tombolProses;
    private JMenuBar menuBar;
    private JMenu menu;
    private JMenuItem menuItem;

}
```

Karena turunan dari JFrame, jangan lupa untuk mengambil konstruktor JFrame untuk menambahkan title saat membuat objeknya:

```
public class DataKelasGUI extends JFrame{

    private JLabel labelKelas, labelProdi;
    private JTextField teksKelas;
    private JComboBox comboProdi;
    private JButton tombolProses;
    private JMenuBar menuBar;
    private JMenu menu;
    private JMenuItem menuItem;

    public DataKelasGUI(String judul){
        super(judul);
    }

}
```

Lalu, tambahkan 1 method “createAndShowGUI” yang akan membuat 1 objek dari kelas yang dimaksud yang sekaligus membuat objek dari JFrame sebagai top level container. Method ini bersifat private (agar tidak bisa diakses dari luar), dan static (agar method bisa diakses tanpa dibentuk object-nya).

```
public class DataKelasGUI extends JFrame{

    private JLabel labelKelas, labelProdi;
    private JTextField teksKelas;
    private JComboBox comboProdi;
    private JButton tombolProses;
    private JMenuBar menuBar;
    private JMenu menu;
    private JMenuItem menuItem;

    public DataKelasGUI(String judul){
        super(judul);
    }
    private static void createShowGUI(){
        DataKelasGUI dkg = new DataKelasGUI("Data Kelas");
        dkg.setDefaultCloseOperation(EXIT_ON_CLOSE);
        dkg.pack();
        dkg.setVisible(true);
    }
}
```

Perhatikan bahwa method tersebut membuat objek dari JFrame (dengan membuat objek dari kelas yang meng-extends JFrame), lalu menge-set apa yang harus dilakukan jika tombol close (x) ditekan (EXIT_ON_CLOSE maksudnya membuat aplikasi keluar dan di-terminate. Pilihan lain ada DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE dan HIDE_ON_CLOSE).

Setelah itu, frame di-pack. Maksud dari method pack() adalah frame tersebut akan dibuat ukurannya sesuai dengan method setPreferredSize() jika method ini dipanggil atau akan dibuat sesuai ukuran dari layout dan sub-komponen pada frame tersebut. Method setVisible(), berfungsi untuk menampilkan frame tersebut ke layar.

Setelah membentuk objek JFrame, pastikan aplikasi bisa dijalankan dengan membuat method main-nya.

```
public class DataKelasGUI extends JFrame{

    private JLabel labelKelas, labelProdi;
    private JTextField teksKelas;
    private JComboBox comboProdi;
    private JButton tombolProses;
    private JMenuBar menuBar;
    private JMenu menu;
    private JMenuItem menuItem;

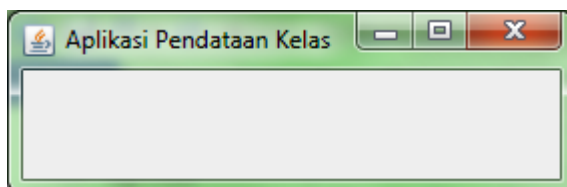
    public DataKelasGUI(String judul){
        super(judul);
    }
    private static void createShowGUI(){
        DataKelasGUI dkg = new DataKelasGUI("Aplikasi Pendataan Kelas");
        dkg.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        dkg.pack();
        dkg.setVisible(true);
    }

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

Perhatikan bahwa gui tersebut dijalankan dengan menggunakan thread. Hal ini sifatnya optional. Method main tersebut dapat dibuat sebagai berikut

```
public static void main(String[] args) {
    createShowGUI();
}
```

Jika di-run, maka akan keluar window sebagai berikut:



Frame masih kosong, sehingga kita perlu menambahkan panel dan pada panel dapat ditambahkan komponen swing yang sudah dideklarasikan (label dll). Penambahan panel ini dapat dilakukan pada sebuah method. Method ini dinamai “addComponentsToPane”.

```
public void addComponentsToPane(final Container pane){  
  
}
```

Method di atas dipanggil pada main dengan cara:

```
private static void createShowGUI(){  
    DataKelasGUI dkg = new DataKelasGUI("Data Kelas");  
    dkg.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    dkg.addComponentsToPane(dkg.getContentPane());  
    dkg.pack();  
    dkg.setVisible(true);  
}
```

Hal ini dimaksudkan seolah-olah frame ada container pane karena akan diletakkan panel di dalamnya. Karena untuk menambahkan panel ke dalam sebuah frame, biasanya method yang digunakan adalah setContentPane:

```
JFrame frame = new JFrame("Frame Baru");  
frame.setContentPane(newContentPane);
```

Dengan newContentPane adalah objek dari JPanel. Tapi hal ini tidak kita lakukan, karena sekarang seolah-olah JFrame bertindak sebagai container pane.

Kembali ke method “addComponentsToPane”, di method ini, flow layout dan panel dideklarasikan. Flow layout boleh tidak dibuat objeknya, karena panel memiliki default layout “flow layout”. Tapi hal ini tetap dilakukan karena kita akan menge-set alignment dari layout menjadi center (peletakan komponen menjadi rata tengah).

```
public void addComponentsToPane(final Container pane){  
    FlowLayout tataLetak = new FlowLayout();  
    final JPanel panelKomponen = new JPanel();  
    panelKomponen.setLayout(tataLetak);  
    tataLetak.setAlignment(FlowLayout.CENTER);  
}
```

Tambahkan kode untuk mengatur peletakan komponen pada objek panel. Dimulai dari menu bar dan teman-temannya. Hanya saja, menu bar ini harus dimasukkan langsung di frame bukan melalui panel, jadi dibutuhkan 1 method yang akan mengembalikan menu bar tersebut. Method tersebut bernama "createMenuBar".

```
public JMenuBar createMenuBar() {  
    menuBar = new JMenuBar();  
    menu = new JMenu("Pilihan");  
    menuBar.add(menu);  
  
    menuItem = new JMenuItem("Kosongkan");  
    menu.add(menuItem);  
    return menuBar;  
}
```

Hasil dari method ini akan digunakan di "createShowGUI" dengan cara memanggil method "setJMenuBar" dan parameter masukannya adalah return value dari method "createMenuBar":

```
private static void createShowGUI() {  
    DataKelasGUI dkg = new DataKelasGUI("Aplikasi Pendataan Kelas");  
    dkg.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    dkg.setJMenuBar(dkg.createMenuBar());  
    dkg.addComponentToPane(dkg.getContentPane());  
    dkg.pack();  
    dkg.setVisible(true);  
}
```

Jika dijalankan maka akan tampil sebagai berikut:



Tugas berikutnya adalah menambahkan komponen lain ke panel. Meneruskan method "addComponentsToPane"

```
public void addComponentsToPane(final Container pane) {
    FlowLayout tataLetak = new FlowLayout();
    final JPanel panelKomponen = new JPanel();
    panelKomponen.setLayout(tataLetak);
    tataLetak.setAlignment(FlowLayout.CENTER);

    labelKelas = new JLabel("Nama Kelas: ");
    panelKomponen.add(labelKelas);

    teksKelas = new JTextField(15);
    panelKomponen.add(teksKelas);

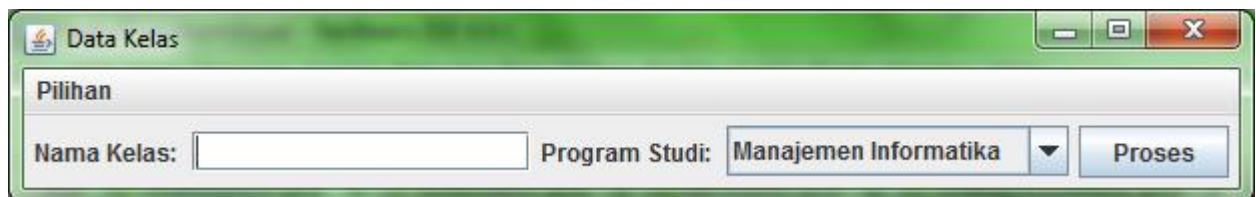
    labelProdi = new JLabel("Program Studi: ");
    panelKomponen.add(labelProdi);

    String daftarPilihan[] = {"Manajemen Informatika",
        "Komputerisasi Akuntansi",
        "Teknik Komputer"};
    comboProdi = new JComboBox(daftarPilihan);
    panelKomponen.add(comboProdi);

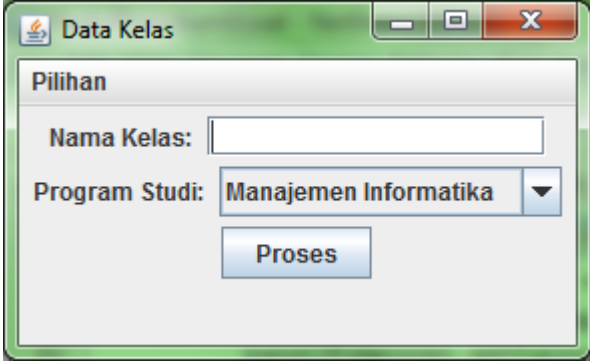
    tombolProses = new JButton("Proses");
    panelKomponen.add(tombolProses);

    pane.add(panelKomponen);
}
```

Jika dijalankan, hasilnya:



Dan karena sifatnya yang “flow”, maka ketika di-resize ke ukuran yang tepat, maka tampilannya adalah sebagai berikut:



The image shows a screenshot of a Java Swing window titled "Data Kelas". The window has a standard title bar with minimize, maximize, and close buttons. Below the title bar, there is a section titled "Pilihan". Inside this section, there are two input fields: "Nama Kelas:" followed by a text input box, and "Program Studi:" followed by a dropdown menu currently displaying "Manajemen Informatika". Below these fields is a button labeled "Proses".