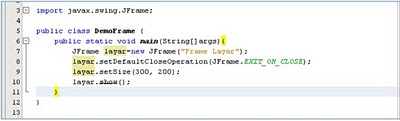
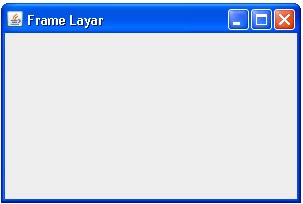
1. **Container**
2. **JFrame** adalah salah satu class dalam java yang dapat digunakan untuk membuat frame sebagai top-level container. Top-level container adalah container dasar di mana komponen swing lainnya diletakkan. Komponen lain yang dimaksudkan antara lain: button, textField, label, dll. Jadi frame di java identik dengan istilah form dalam visual basic.  
   Contoh penggunaan dalam program:  
   [](http://1.bp.blogspot.com/_mSJ_gC3mgWU/TBEMY4MuOnI/AAAAAAAAACs/yms_LizgEUs/s1600/untitled.bmp)  
   Keterangan:  
   Baris 3. Pada baris ini kita mengimport fungsi class JFrame dari package javax.swing. instruksi import akan selalu digunakan apabila kita memanfaatkan fungsi suatu class yang terdapat di luar package di mana program kita dibuat.  
   Baris 7. Pembuatan objek layar dari class JFrame. Pembuata objek disini secara otomatis memanggil konstruktor JFrame(String title). String title berisi teks yang akan dimunculkan sebagai judul frame.  
   Baris 8. Terdapat instruksi pemanggilan method setDefaultCloseOperation() yang digunakan untuk menentukan aktivitas apa yang dilakukan jika frame ditutup. Aktivitas tersebut terdapat dalam parameter method setDefaultCloseOperation() yang dipanggil melalui class JFrame. Pada contoh di atas kita menggunakan aktifitas EXIT\_ON\_CLOSE yang akan menghentikan eksekusi program ketika frame ditutup.  
   Baris 9. Objek layar memanggil method setSize() yang berfungsi mengatur panjang dan lebar frame yang kita buat. Panjang dan lebar frame ditentukan dalam parameter method setSize(). Pada contoh di atas kita membuat frame dengan panjang 300 pixel dan lebar 200pixel.  
   Baris 10. Method show() berfungsi menampilkan frame. Objek layar perlu memanggil method ini karena secara default frame yang kita buat tidak langsung ditampilkan.  
   Ketika program dijalankan maka pada pojok kiri atas dari layar akan muncul frame yang telah kita buat. Output Program:  
   [](http://3.bp.blogspot.com/_mSJ_gC3mgWU/TBEQzTS-ntI/AAAAAAAAAC0/WTgE2xgDNw4/s1600/untitled2.bmp)
3. **Jpanel** merupakan java class untuk membuat panel, dimana panel berfungsi sebagai container bagi komponen java lainnya, seperti : label, button, textfield, table, dan komponen lainnya.

Berikut langkah-langkah untuk membuat panel menggunakan JPanel.

1. Buat file java baru dengan nama TampilPanel. Kemudian masukkan skrip berikut ini kedalamnya.

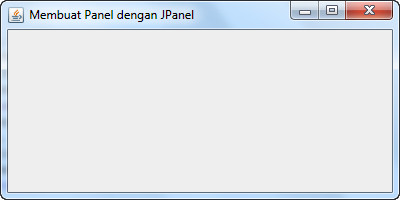
import javax.swing.\*;public class TampilPanel extends JFrame {private JPanel TPanel = new JPanel();public TampilPanel() {super(“Membuat Panel dengan JPanel”);TPanel.setLayout(null);setSize(400, 200);setDefaultCloseOperation(EXIT\_ON\_CLOSE);getContentPane().add(TPanel);setVisible(true);}

public static void main(String[] args){TampilPanel TFrame = new TampilPanel();}}

2. Lakukan kompilasi dengan mengetikkan perintah javac TampilPanel.java

3. Jalankan hasil kompilasi tadi dengan perintah java TampilPanel.

Gambar dibawah ini menunjukkan hasil dari skrip diatas.



1. **JLabel** juga berasal dari package Swing, yang berguna untuk membuat label atau keterangan serta menjelaskan komponen GUI, atau dengan kata lain menambahkan komentar pada suatu frame di GUI.

import javax.swing.\*;  
public class Main extends JFrame {  
private JMenuBar menubar = new JMenuBar();  
private JMenu menu = new JMenu(“File”),  
menu2 = new JMenu(“Help”);  
private JMenuItem file = new JMenuItem(“New”),  
save = new JMenuItem(“Save”),  
open = new JMenuItem(“Open”),  
exit = new JMenuItem(“Exit”);  
private JMenuItem about = new JMenuItem(“About”),  
what = new JMenuItem(“What’s This?”),  
lainnya = new JMenuItem(“Lain-Lain”);  
private JLabel label = new JLabel(“Selamat Datang”);  
private JLabel label2 = new JLabel(“Selamat Menikmati”);  
private JLabel label3 = new JLabel(“~RyuQuiorra~”);  
private JPanel panelku = new JPanel();~  
Main(){  
setTitle(“Pemrograman Java“);  
panelku.setLayout(null);  
setSize(300, 150);  
setDefaultCloseOperation(EXIT\_ON\_CLOSE);  
menubar.add(menu);  
menubar.add(menu2);  
menu.add(file);  
menu.addSeparator();  
menu.add(save);  
menu.add(open);  
menu.addSeparator();  
menu.add(exit);  
menu2.add(what);  
menu2.add(lainnya);  
menu2.addSeparator();  
menu2.add(about);  
setJMenuBar(menubar);  
label.setBounds(76, 15, 160, 25);  
label2.setBounds(76, 35, 160, 25);  
label3.setBounds(76, 55, 160, 25);  
panelku.add(label);  
panelku.add(label2);  
panelku.add(label3);  
getContentPane().add(panelku);  
setVisible(true);  
}  
public static void main(String[] args){  
Main label = new Main();  
}  
}

1. **JOptionPane** merupakan class yang digunakan sebagai standar dialog box  untuk memberikan informasi kepada pengguna aplikasi. Sebenarnya JOptionPane memiliki banyak method, dan kebayakan method tersebut **static method.** Berikut ini 4 method dari JOptionPane yang sering digunakan.
2. showConfirmDialog
3. showInputDialog
4. showMessageDialog
5. showOptionDialog

Untuk parameter dari 4 method tersebut mempunyai pattern yang sama. Dan setiap method tersebut memiliki 4 – 7 varian tergantung dari jumlah parameter setiap method ( kecuali showOptionDialog yang cuma memiliki satu variant) .  Berikut ini beberapa parameter yang digunakan pada 4 method tersebut.

* parentComponent : mendefinisikan component yang menjadi parent dari dialog box. Jika diisi dengan **null**, maka secara default Frame akan menjadi parent component.
* message : merupakan message yang akan ditampilkan dalam dialog box. Message bisa berupa String. Tapi type sebenarnya dari parameter ini adalah **Object** class.
* title : merupakan title untuk dialog box
* initialValue  : merupakan input value
* messageType : mendefinisikan style dari message. Jenis dari messageType sebagai berikut ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, PLAIN\_MESSAGE
* optionType : mendefinisikan kumpulan button option yang muncul pada dialog box. Nilai dari parameter sebagai berikut. DEFAULT\_OPTION, YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION, OK\_CANCEL\_OPTION ( Nilai dari variable ini dapat di-customize)

Berikut ini contoh codenya. ( Kembali kedasar, tanpa menggunakan pallete pada NetBeans )

Nama file : **JOptionDemo.java**

/\*\*  
\*  
\* @author Taeyeon  
\*/  
import java.awt.Component;  
import java.awt.FlowLayout;  
import javax.swing.JComponent.\*;  
import javax.swing.JFrame;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import javax.swing.JButton;  
import javax.swing.JOptionPane;

public class JOptionDemo extends JFrame {

public JOptionDemo(String title){  
super(title);  
setSize(500,100);  
JButton button1 = new JButton(“Click Me!”);  
button1.addActionListener(new ActionListener() {  
public void actionPerformed(ActionEvent event) {  
JOptionPane.showMessageDialog((Component) event.getSource(),”Selamat Pagi Dunia”,”Message Dialog example”,3);  
}  
});

JButton button2 = new JButton(“What is your name?”);  
button2.addActionListener(new ActionListener() {  
public void actionPerformed(ActionEvent event) {  
String nameUser = JOptionPane.showInputDialog((Component)event.getSource(),”Input Your Name” );  
if(nameUser!=null && nameUser!=”"){  
JOptionPane.showMessageDialog((Component) event.getSource(),”Arigatou “+ nameUser);  
}}  
});

JButton button3 = new JButton(“Close Application”);  
button3.addActionListener(new ActionListener() {  
public void actionPerformed(ActionEvent event) {  
int option = JOptionPane.showConfirmDialog((Component)event.getSource(), “Close Application?”);  
if (option == JOptionPane.YES\_OPTION){  
System.exit(0);  
} else if(option==JOptionPane.NO\_OPTION){  
//Do Nothing  
}

}  
});

setLayout(new FlowLayout(FlowLayout.CENTER));  
add(button1);  
add(button2);  
add(button3);

}  
public static void main(String[] args){  
new JOptionDemo(“Dialog Demo”).setVisible(true);  
}  
}