PEMROGRAMAN BERORIENTASI OBJEK

Pertemuan 10

Royana Afwani

Teknik Informatika Universitas Mataram

Graphical User Interface (GUI) di Java

API

- API = Application Programming Interface
- Seperangkat fungsi standar yang disediakan oleh OS atau Bahasa Pemrograman
- Dalam Java, API dimasukkan ke dalam package-package yang sesuai dengan fungsinya (Kumpulan kelas-kelas)
- Paket standar java : bahasa, utilitas, I/O, jaringan, windowing, teks, keamanan, RMI, SQL.
- https://docs.oracle.com/javase/8/docs/api/
- Cara memakai API : Dilakukan dengan mengimpor package/kelas

```
import java.util.Stack;
import javax.swing.*;
```

Overview API

- Java mengandung ratusan kelas standar (library/API)
  - J2SE: Edisi standar
  - J2EE: Edisi enterprise (lebih banyak kelas)
  - J2ME: Edisi Mobile app, Subset kelas standar
- Kelas-kelas ini memungkinkan pembuatan program dengan mudah
- API Java cukup lengkap
  - Mulai dari yang sederhana (misalnya struktur data Stack)
  - Sampai yang kompleks (seperti enkripsi dan akses file ZIP)

API untuk Aplikasi GUI di Java

1. AWT (Abstract Window Toolkit):

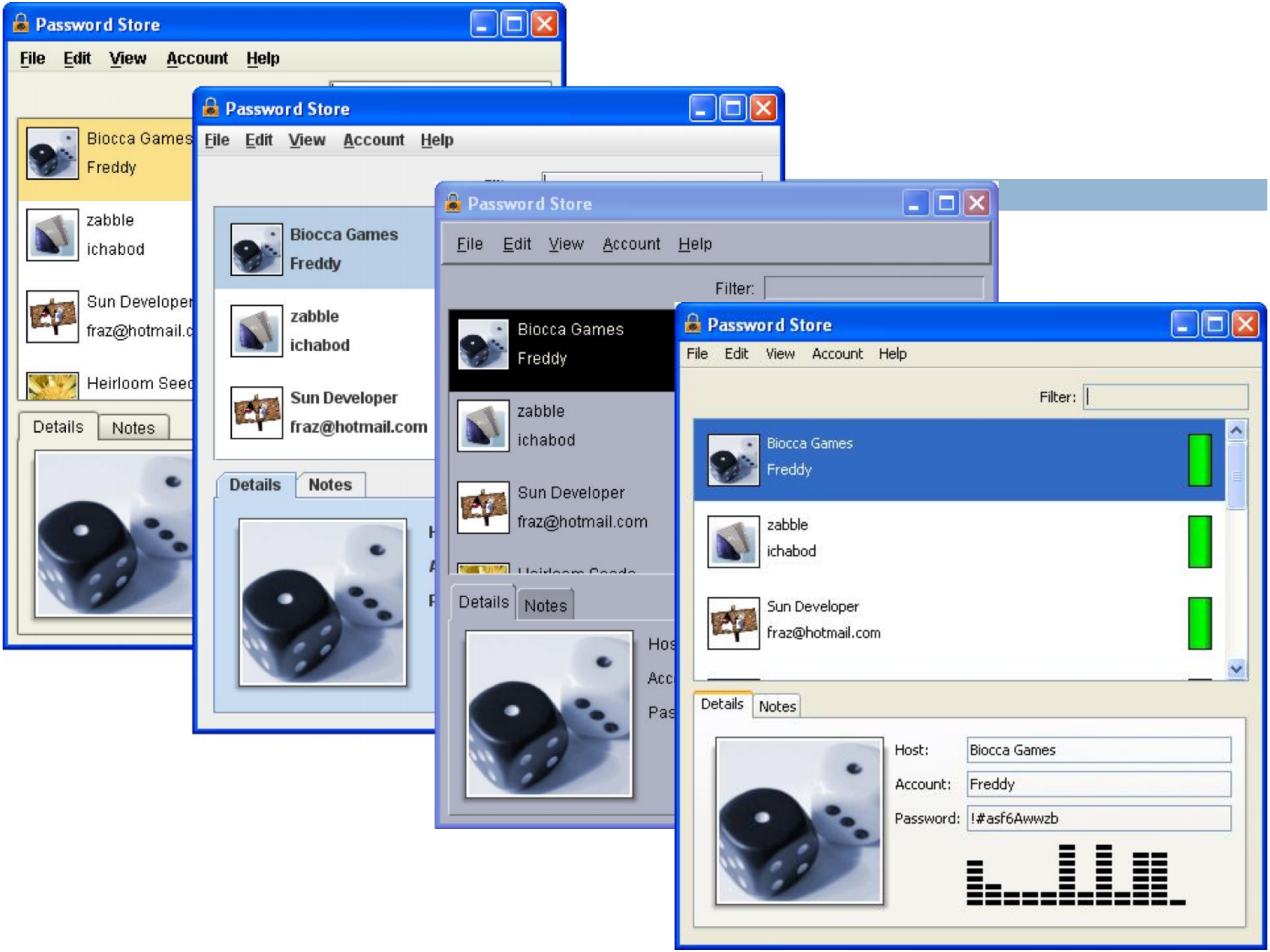
   Library dan komponen GUI (java.awt) yang pertama kali diperkenalkan oleh Java.

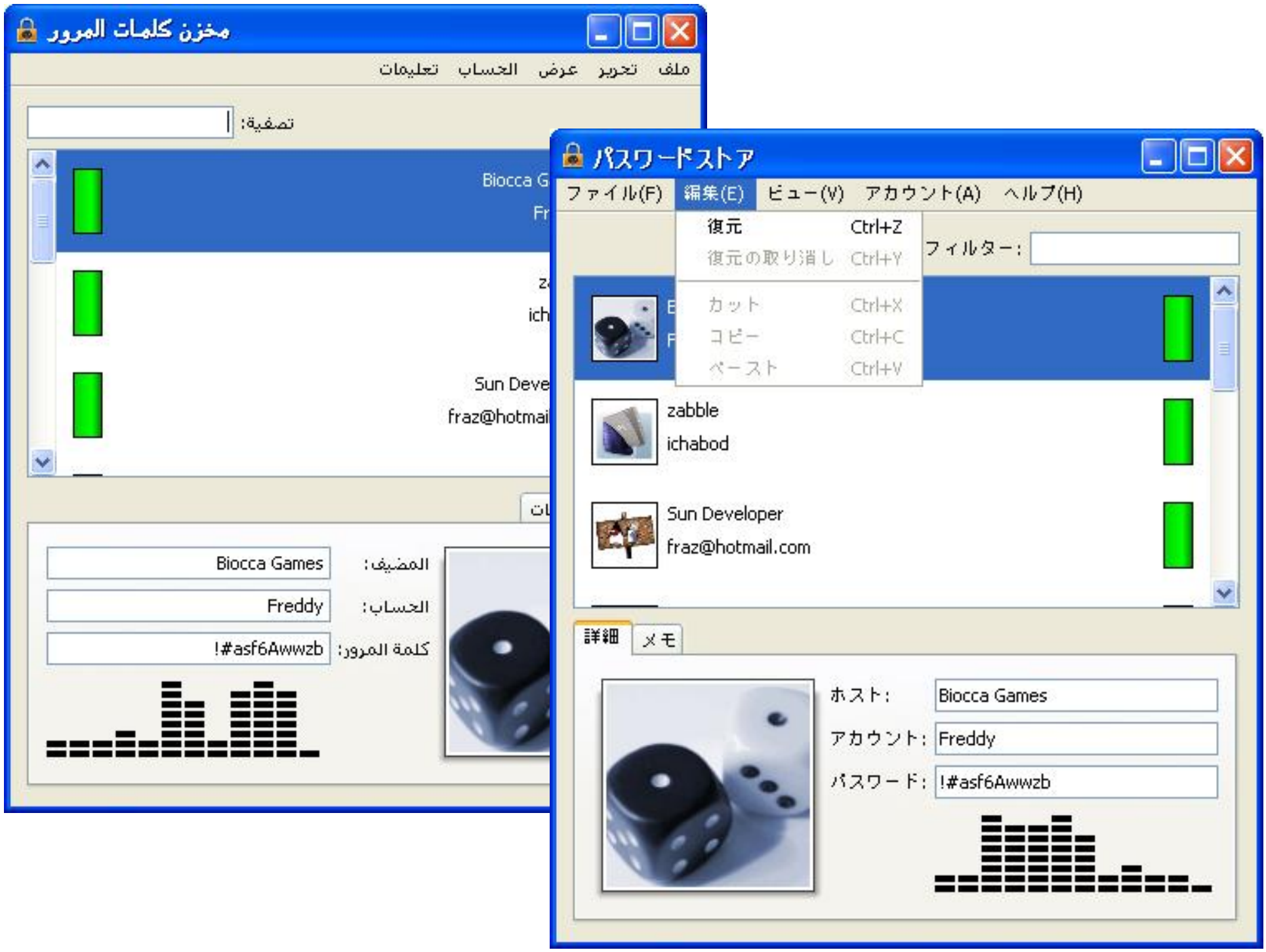
2. Swing or JFC (Java Foundation Class):

   Library dan komponen GUI (javax.swing) terbaru dari Java dan yang direkomendasikan Sun untuk pemrograman GUI. Komponen Swing sebagian besar adalah turunan AWT dan lebih lengkap daripada AWT

Fitur Swing

- Komponen GUI Lengkap: button, listbox, combobox, textarea, dsb
- Pluggable Look-and-Feel: tampilan GUI dapat diubah sesuai dengan kehendak (tidak perlu mengikuti native sistem operasi)
- Data Transfer Antar Komponen: drag and drop, copy and paste
- Internationalization: proses desain aplikasi yang memungkinkan aplikasi dijalankan sesuai dengan preferensi tanpa rekompilasi
- Localization: proses translasi teks ke bahasa lokal dan menambahkan komponen lokal


Password Store
File Edit View Account Help
Biocca Games
Freddy
zabble
ichabod
Sun Developer
fraz@hotmail.com
Heirloom Seed
Details
Notes
Filter:
Host:
Biocca Games
Account:
Freddy
Password:
!#asf6Awwzb


مخزن كلمات المرور
ملف تحرير عرض الحساب تعليمات
تصفية:
Biocca G
Fr
ich
Sun Deve
fraz@hotmai
المضيف: Biocca Games
الحساب: Freddy
كلمة المرور: !#asf6Awwzb
パスワードストア
ファイル(F) 編集(E) ビュー(V) アカウント(A) ヘルプ(H)
復元 Ctrl+Z
復元の取り消し Ctrl+Y
カット Ctrl+X
コピー Ctrl+C
ペースト Ctrl+V
フィルター:
zabble
ichabod
Sun Developer
fraz@hotmail.com
詳細
メモ
ホスト: Biocca Games
アカウント: Freddy
パスワード: !#asf6Awwzb

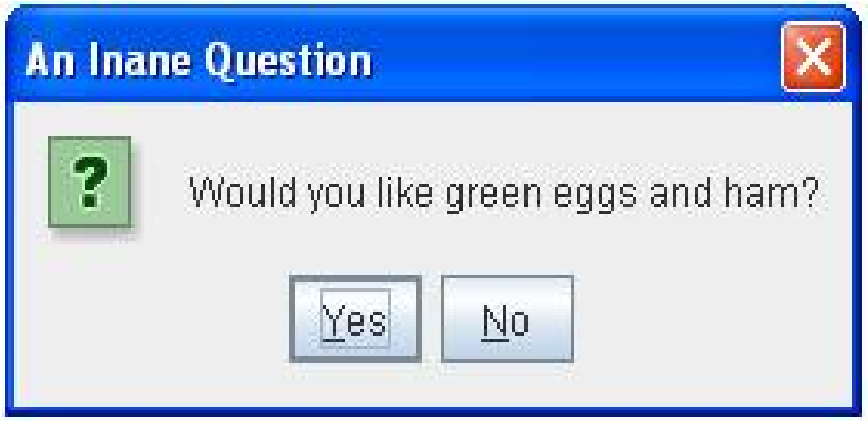
Komponen Dasar Swing

- Secara umum terdapat 5 bagian Swing yang akan sering digunakan :
  - ***Top-Level Container***
    - Container dasar dimana komponen lain diletakkan.
    - Ex : Frame (**JFrame**), Dialog (**JDialog**) & Applet (**JApplet**)
  - ***Intermediate Container***
    - Container perantara dimana komponen lain diletakkan
    - Ex : JPanel, dimana umumnya hanya digunakan sebagai tempat untuk meletakkan/mengelompokkan komponen-komponen yang digunakan, baik container atau berupa *atomic component*. Dan digunakan juga sebagai *scroll pane* (**JScrollPane & JTabbed Pane**).
  - ***Atomic Component***
    - Komponen yang memiliki fungsi spesifik, dimana umumnya user langsung berinteraksi dengan komponen ini
    - Ex : **JButton, JLabel, JTextField, JTextArea.**

Komponen Dasar Swing (*cont.*)

- ***Layout Manager***
  - Berfungsi untuk mengatur bagaimana tata letak/posisi dari komponen yang akan diletakkan, satu sama lain di dalam suatu container.
  - Secara default ada 6 buah layout : **BorderLayout, BoxLayout, FlowLayout, CardLayout, GridBagLayout & GridLayout**
- ***Event Handling***
  - Menangani event yang dilakukan oleh user seperti menekan tombol, memperbesar atau memperkecil ukuran frame, mengklik mouse, mengetik sesuatu dengan keyboard, dll.

Top Level Container

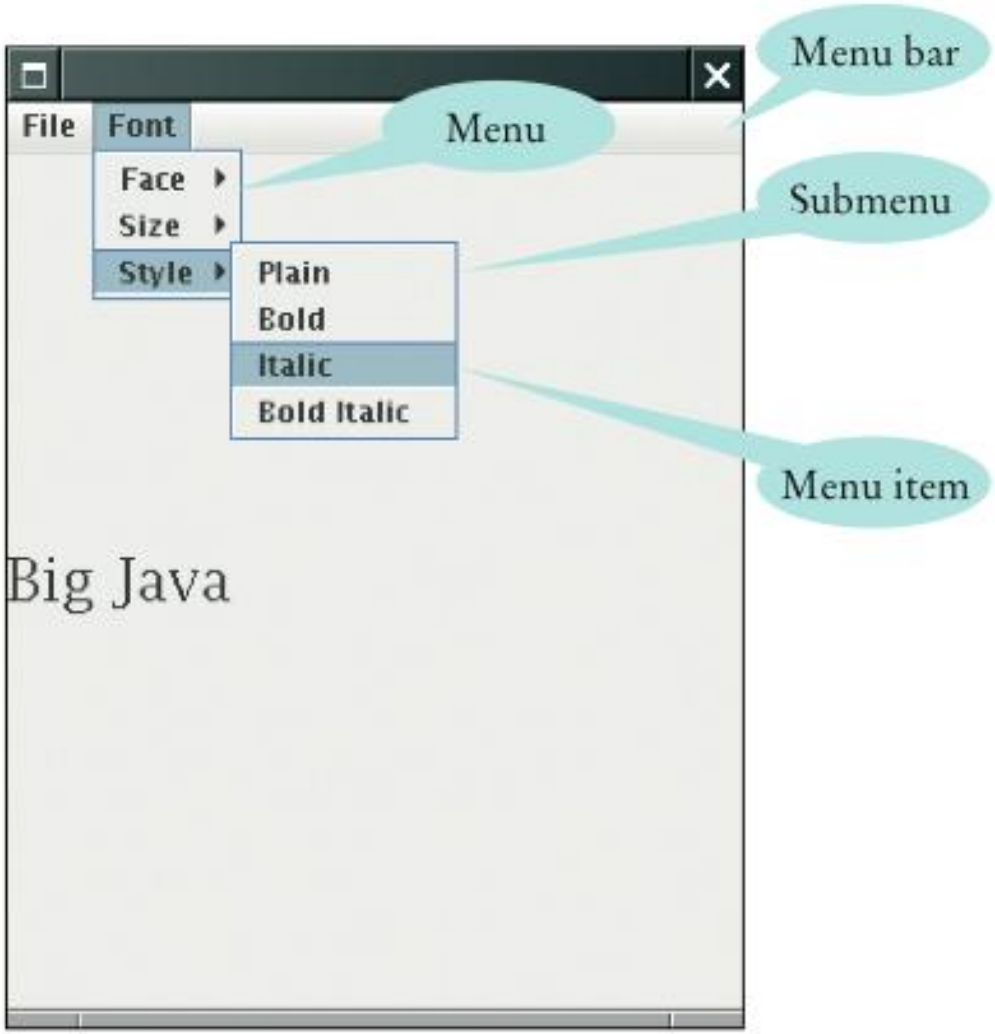


JApplet

JDialog

JFrame

Intermediate Container : Menu

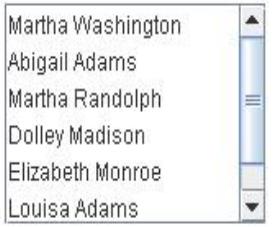


Atomic Component

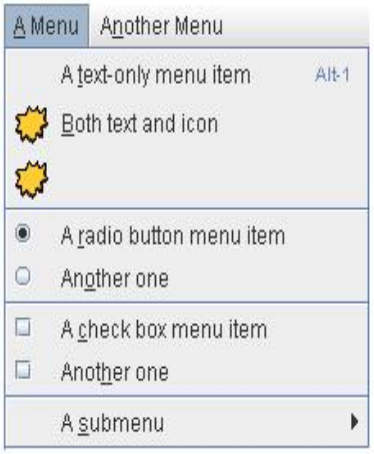
JButton

JCheckBox

JComboBox

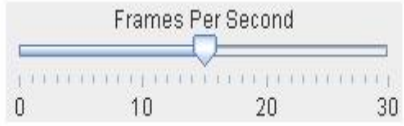


JList



JMenu

JRadioButton



JSlider

JSpinner

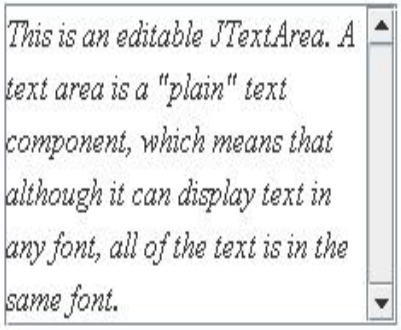
JTextField

JPasswordField

Atomic Component

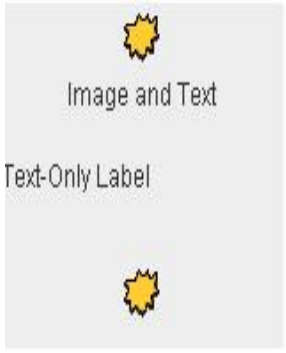
Host	User	Password	Last Modified
Biocca Games	Freddy	!#asf6Awwzb	Mar 16, 2006
zabble	ichabod	Tazb!34$fZ	Mar 6, 2006
Sun Developer	fraz@hotmail.co...	AasW541!fbZ	Feb 22, 2006
Heirloom Seeds	shams@gmail....	bkz[ADF78!	Jul 29, 2005
Pacific Zoo Shop	seal@hotmail.c...	vbAf124%z	Feb 22, 2006

JTable

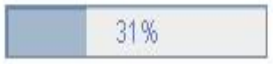


JTextArea

JTree

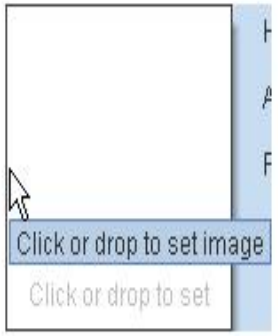


JLabel



JProgressBar

JSeparator



JToolTip

Atomic Component : JLabel

Untuk membuat tulisan pada frame dibutuhkan sebuah objek yang akan mewakili sebuah teks.

Inisialisasi

```
private JLabel label = new JLabel("Name");
```

Method

getText()	Untuk memperoleh teks pada label
setText()	Mengubah/memberikan text pada label
setFont()	Untuk mengubah jenis huruf pada tulisan yang ditampilkan
Dan lain lain	

Atomic Component : JButton

Untuk membuat objek tombol

Inisialisasi

```
private JButton tombol = new JButton("Save");
```

Method

setEnable(boolean)	tombol.setEnable(false)
setVisible(boolean)	
setText()	
setFont	tombol.setFont(new Font(“Arial”, Font.BOLD,29))
setForeground()	tombol.setForeground(Color.blue)
Dan lain lain	

Atomic Component : JTextField

Untuk menerima input dari user

Inisialisasi

```
private JTextField text = new JTextField();
```

Method

setEnable(boolean)	text.setEnable(false)
setVisible(boolean)	
setText()	
setFont	text.setFont(new Font("Arial", Font.BOLD,29))
setForeground()	text.setForeground(Color.blue)
Dan lain lain….	

Top-Level Container

- Ada 3 buah top-level container :
  - JFrame,
  - Jdialog,
  - Japplet → untuk GUI tampil di browser (web)

Top-Level Container
Frame

- Ada 2 kontrukstor yang sering digunakan untuk membuat frame :
  - **`JFrame()`**
  - **`JFrame(String title)`**

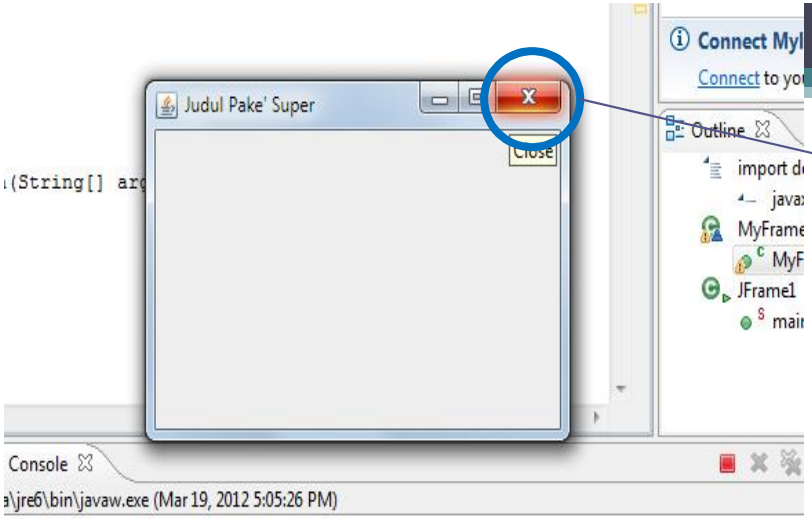
- ***Contoh 1 :***

```
1. import javax.swing.*;
2. public class Contoh1 {
3.       public static void main(String[] args) {
4.           JFrame frame = new JFrame("Contoh Frame");
5.           frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
6.           frame.setSize(400,150);
7.           frame.show();
8.       }
9. }
```

Top-Level Container

Frame

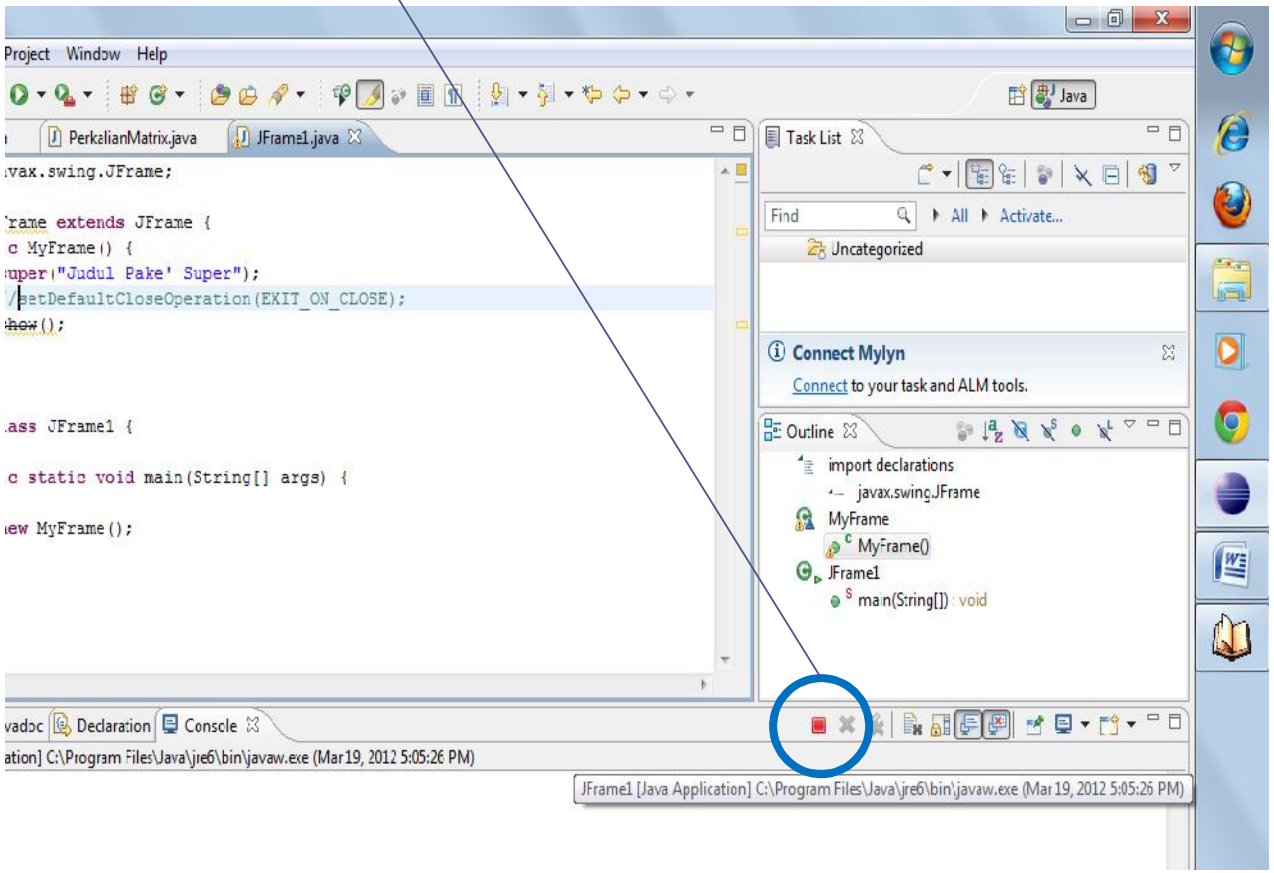
- Class JFrame mendeklarasikan 4 jenis aktivitas :
  - ***DO_NOTHING_ON_CLOSE***
    - Secara internal tidak ada aktivitas apapun yang dilakukan secara otomatis jika kita menutup frame tsb. Biasanya digunakan jika kita ingin menangani sendiri aktivitas tsb.
  - ***HIDE_ON_CLOSE***
    - Merupakan aktivitas default, dimana frame hanya disembunyikan atau tidak ditampilkan ke layar, namun secara fisik frame ini masih ada di memori sehingga jika diinginkan dapat ditampilkan kembali
  - ***DISPOSE_ON_CLOSE***
    - Menghapus tampilan frame dari layar, menghapusnya dari memori & membebaskan resource yang dipakai.
  - ***EXIT_ON_CLOSE***
    - Menghentikan eksekusi program. Cocok digunakan untuk frame utama, dimana jika frame tsb ditutup mengakibatkan eksekusi program berhenti



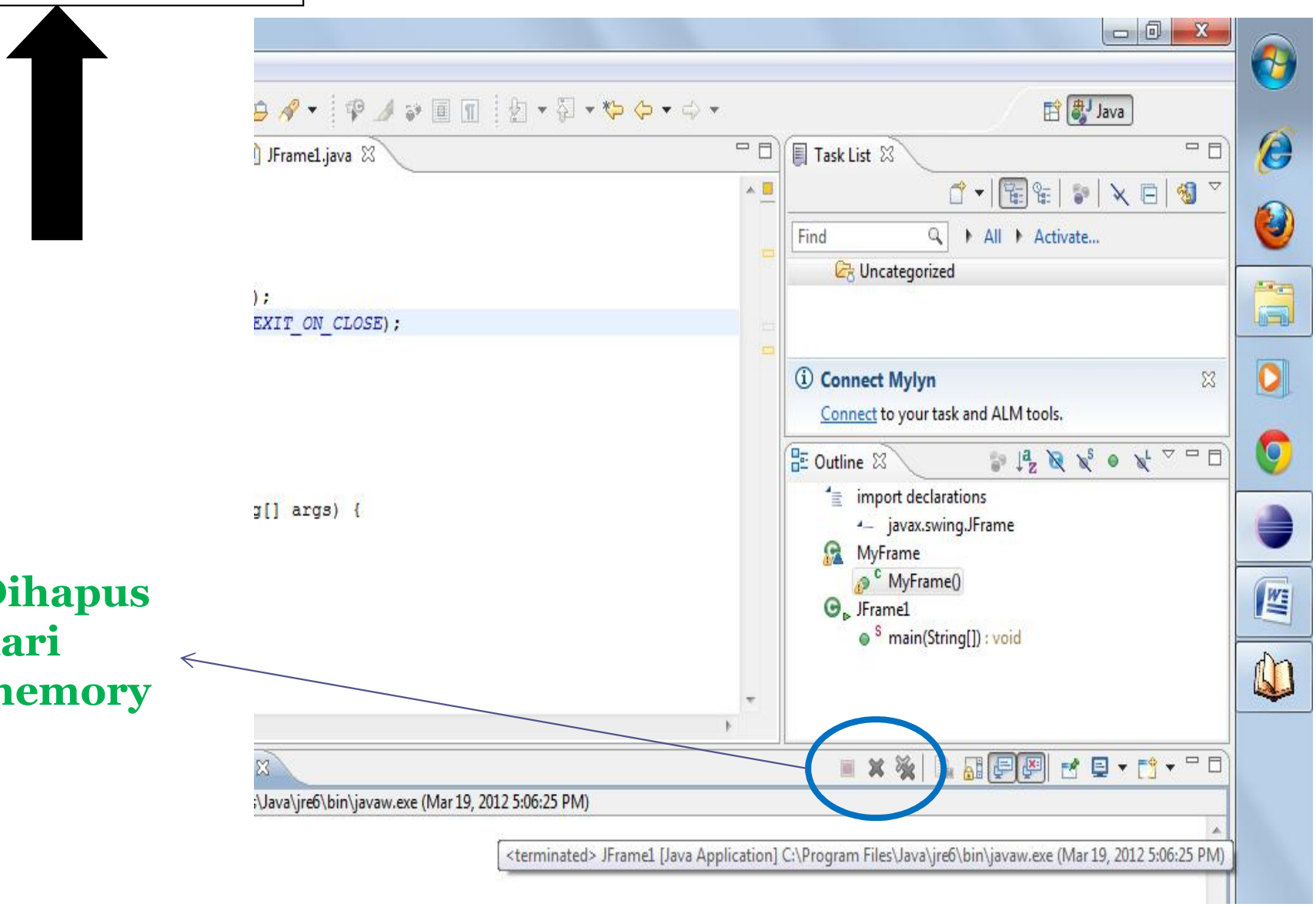
Click "close"

Program hanya di-hide, masih ada di memory

Default Setting
HIDE_ON_CLOSE



EXIT_ON_CLOSE



Dihapus dari memory

Top-Level Container
Dialog

- Perbedaan utama antara frame & dialog
  - Dialog pada umumnya tidak dibuat untuk berdiri sendiri. Dialog biasanya digunakan bersamaan dengan frame atau dialog lainnya yang bertindak sebagai parent.
  - Jika parent dialog tsb dihapus dari memori maka dialog tersebut juga akan dihapus dari memori.
- Cara termudah untuk menampilkan dialog → dengan class **JOptionPane**

Top-Level Container
Dialog

- Contoh 2 :

```
1. import javax.swing.*;
2. public class Contoh2 {
3.      public static void main(String[] args) {
4.         JFrame frame = new JFrame("Contoh Frame");
5.         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
6.         frame.show();

7.         JOptionPane.showConfirmDialog(frame,
8.            "Contoh dialog konfirmasi ...",
9.            "Judul Dialog",
10.           JOptionPane.OK_CANCEL_OPTION,    //Jenis Tombol
11.           JOptionPane.QUESTION_MESSAGE);   //Icon
12.     }
13. }
```

Top-Level Container
Dialog

- ***Jenis Tombol :***
  - **`JOptionPane.OK_CANCEL_OPTION`**
  - **`JOptionPane.YES_NO_OPTION`**
  - **`JOptionPane.YES_NO_CANCEL_OPTION`**
- ***Jenis Icon :***
  - **`JOptionPane.QUESTION_MESSAGE`**
  - **`JOptionPane.INFORMATION_MESSAGE`**
  - **`JOptionPane.WARNING_MESSAGE`**
  - **`JOptionPane.ERROR_MESSAGE`**

Top-Level Container - *Dialog*

Contoh 3 :

```
1.  import javax.swing.*;
2.  public class Contoh3 {
3.      public static void main(String[] args) {
4.          JFrame frame = new JFrame("Contoh Frame");
5.          frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
6.          frame.show();

7.          int result =
8.              JOptionPane.showConfirmDialog(frame,
9.                  "Contoh dialog konfirmasi ...",
10.                 "Judul Dialog",
11.                 JOptionPane.OK_CANCEL_OPTION,
12.                 JOptionPane.QUESTION_MESSAGE);

13.         String message;
14.         if (result==JOptionPane.OK_OPTION)
15.             message = "Anda memilih ok";
16.         else if (result==JOptionPane.CANCEL_OPTION)
17.             message = "Anda memilih cancel";
18.         else
19.             message = "Anda tidak memilih apapun";
```

Top-Level Container - *Dialog*

Contoh 3 (lanjutan) :

```
20.             JOptionPane.showMessageDialog(frame,
21.                 message,
22.                 "Pilihan Anda",
23.                 JOptionPane.INFORMATION_MESSAGE);
24.         }
25.     }
```

Top-Level Container
Dialog

- Selain itu, kita juga bisa membuat suatu dialog sesuai dengan selera kita dengan menggunakan **JDialog**.

- Beberapa konstruktor dari JDialog :
  - `JDialog()`
  - `JDialog(Dialog owner)`
  - `JDialog(Dialog owner, String title)`
  - `JDialog(Frame owner)`
  - `JDialog(Frame owner, String title)`
  - `JDialog(Frame owner, String title, boolean modal)`
  - `dll`

Top-Level Container - *Dialog*

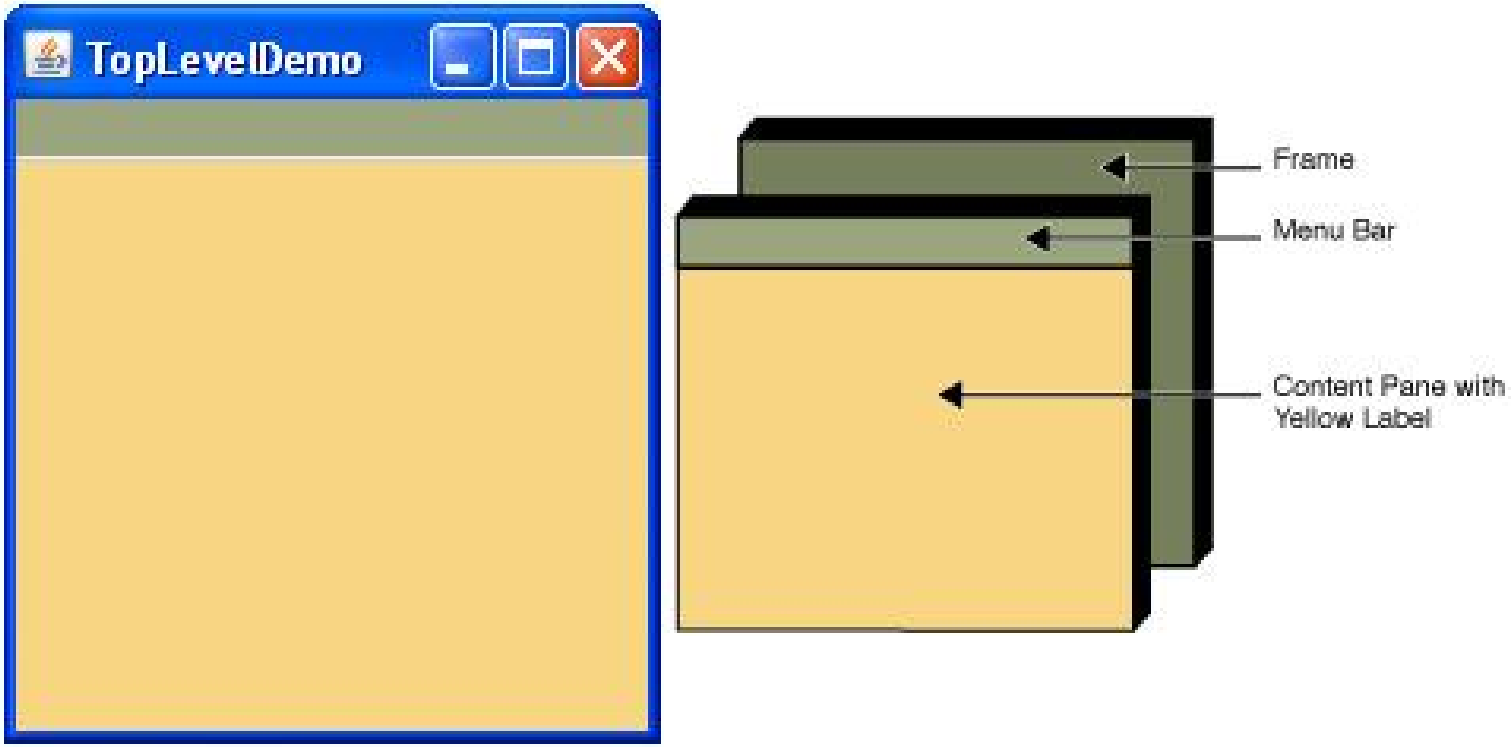
Contoh 4 :

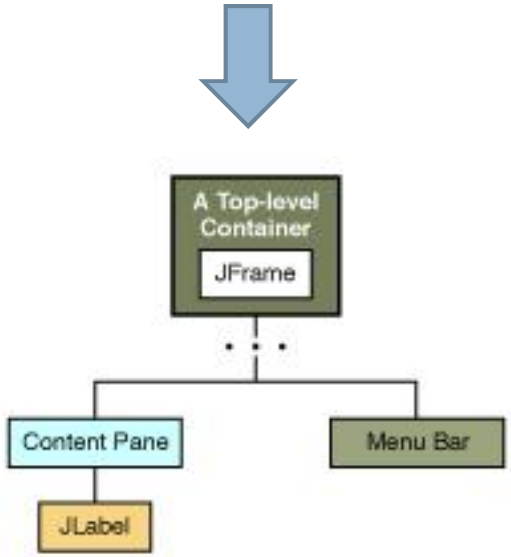
```
1.    import javax.swing.*;
2.    public class Contoh4 {
3.          public static void main(String[] args) {
4.              JFrame frame = new JFrame("Contoh Frame");
5.              frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
6.              frame.setSize(400,150);
7.              frame.show();
8.
9.              JDialog dlg = new JDialog (frame,"Dialog 1",true);
10.             dlg.setSize(200,100);
11.             dlg.show();
12.         }
13.   }
```

Layout Management

- Untuk mengatur layout dari setiap komponen yang diletakkan pada container, digunakan *layout manager*.
- Setiap pane secara default pasti memiliki layout manager. Jika ingin mengubah layout-nya, gunakan :
  - `void setLayout(LayoutManager mgr)`

- Penting !
  - Kita tidak pernah langsung menset layout manager dari top-level container melainkan kita menset layout manager dari content pane-nya.
  - Untuk mendapatkan content pane dari top-level container, gunakan method :
    - `Container getContentPane()`





Each top-level container has a content pane that, generally speaking, contains (directly or indirectly) the visible components in that top-level container's GUI.
You can optionally add a menu bar to a top-level container. The menu bar is by convention positioned within the top-level container, but outside the content pane.

Layout Management

- Contoh kode :

```
//buat object top-level container
JFrame frame = new JFrame("FlowLayout");

//buat object layout manager
FlowLayout layout = new FlowLayout(FlowLayout.LEFT);

//ambil content pane dan set Layout
frame.getContentPane().setLayout(layout);
```

Layout Management

- Java menyediakan 6 buah class standar yang dapat digunakan sebagai layout manager yang terdapat dalam package **java.awt**, yaitu :
  - **`FlowLayout`**
  - **`GridLayout`**
  - **`BorderLayout`**
  - **`CardLayout`**
  - **`GridBagLayout`**
  - **`BoxLayout`**

Layout Management

BorderLayout

- BorderLayout memiliki 5 buah area, yaitu : north, south, east, west & center.

- **Contoh 5:**

```
1.  import javax.swing.*;
2.  import java.awt.*;

3.  public class Contoh5 {
4.          public static void main(String[] args) {
5.              JFrame frame = new JFrame ("Contoh Border Layout");
6.              BorderLayout layout = new BorderLayout(1,1);
7.              frame.getContentPane().setLayout(layout);
8.
9.              //atomic component
10.             JButton btnNorth  = new JButton("Posisi NORTH");
11.             JButton btnSouth  = new JButton("Posisi SOUTH");
12.             JButton btnEast   = new JButton("Posisi EAST");
13.             JButton btnWest   = new JButton("Posisi WEST");
14.             JButton btnCenter = new JButton("Posisi CENTER");
```

Layout Management

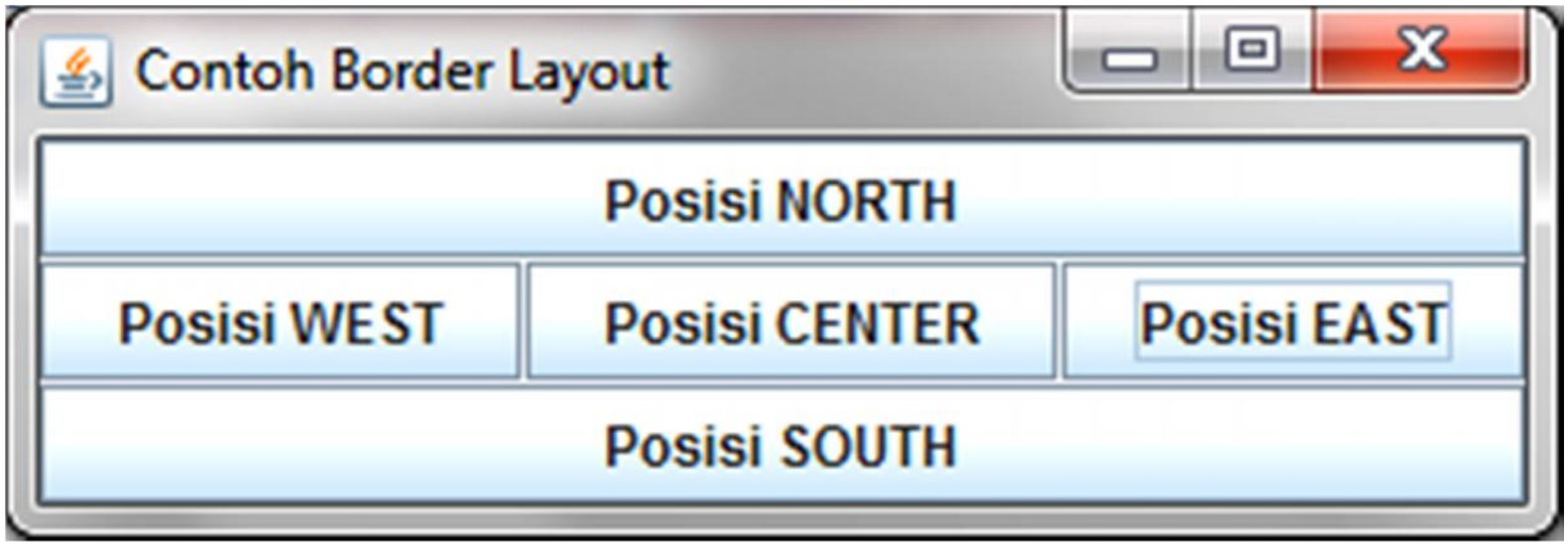
BorderLayout

- **Contoh 5 (lanjutan) :**

```
15.         frame.getContentPane().add(btnNorth,BorderLayout.NORTH);
16.         frame.getContentPane().add(btnSouth,BorderLayout.SOUTH);
17.         frame.getContentPane().add(btnEast,BorderLayout.EAST);
18.         frame.getContentPane().add(btnWest,BorderLayout.WEST);
19.         frame.getContentPane().add(btnCenter,BorderLayout.CENTER);

20.         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21.         frame.pack();
22.         frame.show();
23.         }
24. }
```

Layout Management

BorderLayout



Layout Management

BoxLayout

- **BoxLayout** akan meletakkan komponen berurutan ke kanan (sumbu X) atau berurutan ke bawah (sumbu Y).
- Penggunaan **BoxLayout** secara langsung agak rumit, sehingga digunakan class **Box** yang merupakan container yang secara internal telah menggunakan **BoxLayout** sebagai layout manager-nya.

Layout Management - *BoxLayout*

Contoh 6:

```
1.  import javax.swing.*;
2.  import java.awt.*;

3.  public class Contoh6 {
4.     public static void main(String[] args) {
5.        JFrame frame = new JFrame ("Contoh Box Layout");
6.        Box comp = new Box(BoxLayout.Y_AXIS);//X_AXIS

7.        JButton btn1  = new JButton("Posisi 1");
8.        JButton btn2  = new JButton("Posisi 2");
9.        JButton btn3  = new JButton("Posisi 3");
10.       JButton btn4  = new JButton("Posisi 4");
11.       JButton btn5  = new JButton("Posisi 5");
```

Layout Management - *BoxLayout*

Contoh 6 (lanjutan):

```
12.         comp.add(btn1);
13.         comp.add(btn2);
14.         comp.add(btn3);
15.         comp.add(btn4);
16.         comp.add(btn5);

17.         frame.getContentPane().add(comp);

18.         frame.setDefaultCloseOperation
                                  (JFrame.EXIT_ON_CLOSE);
19.         frame.pack();
20.         frame.show();
21.     }
22. }
```

Layout Management

FlowLayout

- Merupakan layout manager default yang digunakan JPanel.
- Pada dasarnya, layout manager ini hanya meletakkan kompenen yg ada secara berurutan dari kiri ke kanan & jika diperlukan akan berpindah baris.
- Kita juga bisa menentukan sendiri seberapa besar jarak antar komponen baik secara vertikal maupun horisontal.
- Selain itu, kita bisa menentukan alignment dari komponen yang diletakkan, yaitu rata kanan, rata kiri atau di tengah

Layout Management - *FlowLayout*

Contoh 7:

```
1.  import javax.swing.*;
2.  import java.awt.*;

3.  public class Contoh7 {
4.          public static void main(String[] args) {
5.                  JFrame frame = new JFrame ("Contoh Flow Layout");
6.                  FlowLayout layout = new FlowLayout(FlowLayout.LEFT);
7.                  layout.setVgap(10);//jarak vertikal antar komponen
8.                  layout.setHgap(10);//jarak horisontal antar komponen
9.                  frame.getContentPane().setLayout(layout);

10.                 JButton btn1  = new JButton("Posisi 1");
11.                 JButton btn2  = new JButton("Posisi 2");
12.                 JButton btn3  = new JButton("Posisi 3");
13.                 JButton btn4  = new JButton("Posisi 4");
14.                 JButton btn5  = new JButton("Posisi 5");

15.                 frame.getContentPane().add(btn1);
16.                 frame.getContentPane().add(btn2);
17.                 frame.getContentPane().add(btn3);
18.                 frame.getContentPane().add(btn4);
19.                 frame.getContentPane().add(btn5);

20.                 frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21.                 frame.setSize(300,150);
22.                 frame.show();
23.         }
24. }
```

Layout Management

CardLayout

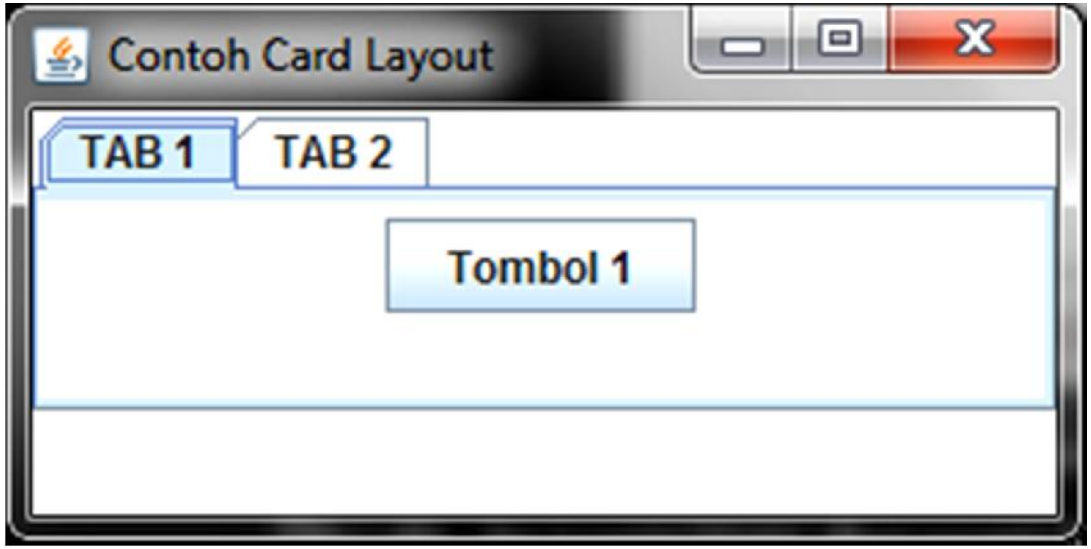
- Layout ini menyusun komponen seperti tumpukan kartu, sehingga hanya satu buah komponen yang terlihat pada satu waktu.
- Biasanya komponen yang kita letakkan dengan layout ini berupa object yang bertipe *intermediate container* yang di dalamnya terdapat atomic component.
- Salah satu contoh *intermediate container* adalah *tabbed pane* (**JTabbedPane**)

Layout Management - *CardLayout*

Contoh 8:

```
1.  import javax.swing.*;
2.  import java.awt.*;
3.  public class Contoh8 {
4.     public static void main(String[] args) {
5.          JFrame frame  = new JFrame ("Contoh Card Layout");
6.          JPanel panel1 = new JPanel();
7.          JPanel panel2 = new JPanel();
8.          JButton btn1  = new JButton("Tombol 1");
9.          JButton btn2  = new JButton("Tombol 2");

10.         panel1.add(btn1);
11.         panel2.add(btn2);

12.         JTabbedPane tab = new JTabbedPane();
13.         tab.add(panel1,"TAB 1");
14.         tab.add(panel2,"TAB 2");

15.         frame.getContentPane().add(tab,BorderLayout.NORTH);
16.         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
17.         frame.setSize(300,150);
18.         frame.show();
19.    }
20. }
```

Layout Management - *CardLayout*



Layout Management

GridLayout

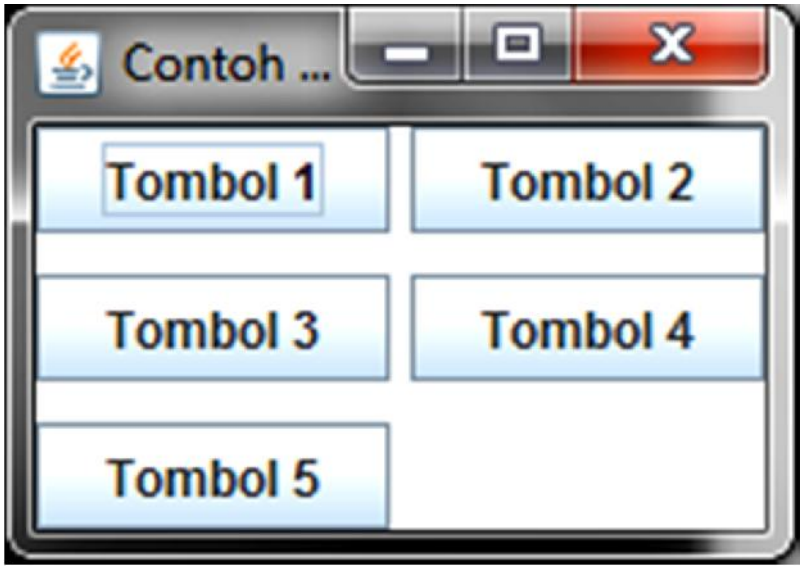
- Layout ini pada dasarnya meletakkan setiap komponen yang ada ke dalam baris & kolom yang telah ditentukan.
- Setiap komponen yang diletakkan akan memiliki ukuran yang sama.
- Kita juga bisa menentukan jarak vertikal & horisontal antar komponen.

Layout Management - *GridLayout*

Contoh 9 :

```
1.      import javax.swing.*;
2.      import java.awt.*;
3.      public class Contoh9 {
4.          public static void main(String[] args) {
5.                JFrame frame  = new JFrame ("Contoh Grid Layout");
6.                GridLayout layout = new GridLayout(3,2);
7.                layout.setHgap(5);
8.                layout.setVgap(10);
9.                frame.getContentPane().setLayout(layout);

10.               JButton btn1  = new JButton("Tombol 1");
11.               JButton btn2  = new JButton("Tombol 2");
12.               JButton btn3  = new JButton("Tombol 3");
13.               JButton btn4  = new JButton("Tombol 4");
14.               JButton btn5  = new JButton("Tombol 5");

15.               frame.getContentPane().add(btn1);
16.               frame.getContentPane().add(btn2);
17.               frame.getContentPane().add(btn3);
18.               frame.getContentPane().add(btn4);
19.               frame.getContentPane().add(btn5);

20.               frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21.               frame.pack();
22.               frame.show();
23.         }
24.     }
```

Layout Management - *GridLayout*



Layout Management

GridBagLayout

- Hampir mirip dengan GridLayout, karena masih bekerja dengan grid.
- Layout ini paling flexibel, dimana bisa menentukan atribut dari setiap komponen secara individual sehingga tampilan untuk setiap komponen yang ada dapat berbeda-beda.
- Untuk menentukan atribut ini digunakan *constraint*, dimana untuk membuat harus membuat object dari class **GridBagConstraints**.

Layout Management

GridBagLayout

- Beberapa variabel dalam class **GridBagConstraints** :
  - ***gridx, gridy***
    - Menentukan kolom & baris berapa komponen akan diletakkan
  - ***gridwidth, gridheight***
    - Menentukan seberapa banyak kolom/baris yang akan digunakan untuk menampilkan komponen
  - ***fill***
    - Menentukan bagaimana menampilkan komponen jika ternyata ukuran komponen lebih besar dari daerah tampilannya.
  - ***ipadx, ipady***
    - Menentukan seberapa banyak pixel yang akan ditambahkan ke ukuran minimum dari komponen.
  - ***insets***
    - Menentukan seberapa besar jarak antar komponen dengan komponen lainnya
  - ***anchor***
    - Menentukan letak komponen jika ukuran komponen lebih kecil dari ukuran daerah tampilannya
  - ***weightx, weighty***
    - Menentukan bagaimana mendistribusikan jarak di antara baris & kolom. Sangat penting untuk menentukan sifat dari peletakan komponen jka terjadi operasi resizing.

Layout Management - *GridBagLayout*

Contoh 10 :

```
1.  import javax.swing.*;
2.  import java.awt.*;
3.  public class Contoh10 {
4.     public static void main(String[] args) {
5.        JFrame frame  = new JFrame ("Contoh Grid Bag Layout");
6.        GridBagLayout layout = new GridBagLayout();
7.        GridBagConstraints c = new GridBagConstraints();
8.        frame.getContentPane().setLayout(layout);
9.        c.fill = GridBagConstraints.HORIZONTAL;

10.       JButton btn1  = new JButton("Tombol 1");
11.          c.gridx   = 0;   //kolom 0
12.          c.gridy   = 0;    //baris 0
13.          layout.setConstraints(btn1,c);
14.          frame.getContentPane().add(btn1);

15.       JButton btn2  = new JButton("Tombol 2");
16.          c.gridx   = 1;    //kolom 1
17.          c.gridy   = 0;    //baris 0
18.          layout.setConstraints(btn2,c);
19.          frame.getContentPane().add(btn2);
```

Layout Management - *GridBagLayout*

Contoh 10 (lanjutan) :

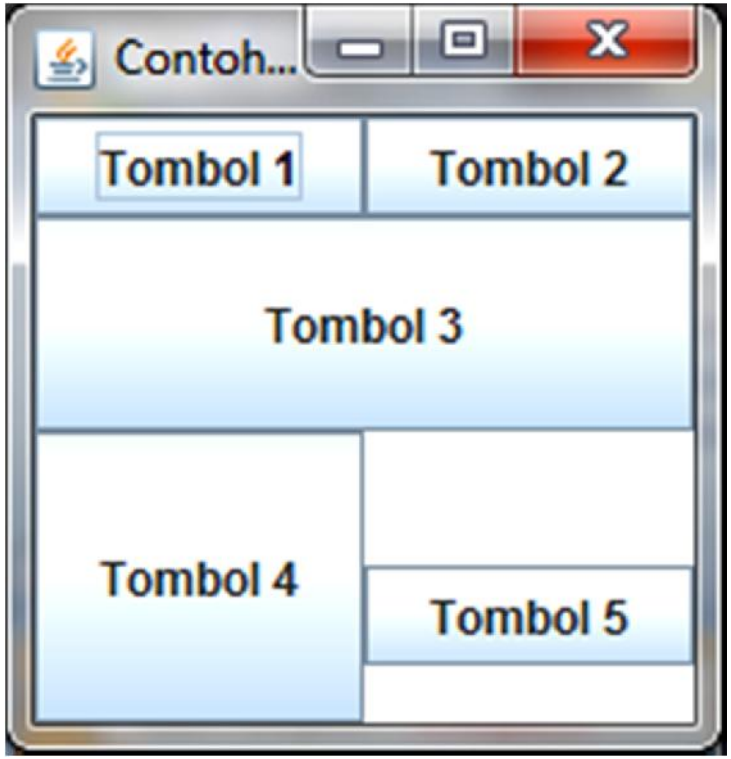
```
21.         JButton btn3  = new JButton("Tombol 3");
22.            c.ipady      = 30;//perbesar tinggi
23.            c.gridwidth = 2; //menempati 2 kolom
24.            c.gridx      = 0; //kolom 0
25.            c.gridy      = 1; //baris 1
26.            layout.setConstraints(btn3,c);
27.            frame.getContentPane().add(btn3);

28.         JButton btn4  = new JButton("Tombol 4");
29.            c.ipady      = 50;//perbesar tinggi
30.            c.gridwidth = 1; //menempati 2 kolom
31.            c.gridx      = 0; //kolom 0
32.            c.gridy      = 2; //baris 2
33.            layout.setConstraints(btn4,c);
34.            frame.getContentPane().add(btn4);
```

Layout Management - *GridBagLayout*

Contoh 10 (lanjutan) :

```
35.         JButton btn5  = new JButton("Tombol 5");
36.            c.ipady      = 0; //tinggi normal
37.            c.gridwidth = 1; //menempati 1 kolom
38.            c.gridx      = 1; //kolom 1
39.            c.gridy      = 2; //baris 2
40.            c.insets     = new Insets(10,0,0,0); //t,l,b,r
41.            c.anchor     = GridBagConstraints.SOUTH;
42.            layout.setConstraints(btn5,c);
43.            frame.getContentPane().add(btn5);
44.
45.         frame.setDefaultCloseOperation
                                        (JFrame.EXIT_ON_CLOSE);
46.         frame.pack();
47.         frame.show();
48.      }
49.   }
```

Layout Management - *GridBagLayout*



Next : Penanganan Kejadian (*Event Handling*)

TERIMA KASIH