**Pertemuan 8**

# **JAVA GUI (GRAPHICAL USER INTERFACE)**

**Tujuan Instruksional :** Pengantar

Mengidentifikasi bagian dasar dari program java

* Membedakan mana yang termasuk ke dalam java literals, tipe data dasar, tipe variabel, pengidentifikasian dan operator.
* Mengembangkan program java sederhana menggunakan konsep pembelajaran pada bab ini.
* Menganalisa program java pertama

**Kompetensi yang Diharapkan :**

1. Praktikan dapat memahami dan menjelaskan konsep polimorfisme dalam perograman serta dapat mengimplementasikan ke dalam pemrograman Java

**Waktu Pertemuan**  : 100 menit

**GRAPHICAL USER INTERFACE (GUI)**

****

Graphical User Interface (GUI) merupakan desain aplikasi dengan tampilan visual sehingga pengguna dapat dengan mudah menggunakan aplikasi. The Java Foundation Class (JFC), merupakan bagian penting dari Java SDK, yang termasuk dalam koleksi dari API dimana dapat mempermudah pengembangan aplikasi JAVA GUI.

JFC termasuk diantara 5 bagian utama dari API yaitu AWT, Swing, Java2D, Accessibility, dan Drag & Drop. Kelima komponen tersebut membantu pengembang dalam mendesain dan mengimplementasikan aplikasi visual yang lebih baik.

AWT dan Swing menyediakan komponen GUI yang dapat digunakan dalam membuat aplikasi Java. Beberapa komponen AWT dibuat menggunakan native code, sedangkan pada Swing keseluruahan komponen ditulis menggunakan bahasa pemrograman Java.

Swing menyediakan implementasi platform-independent dimana aplikasi yang dikembangkan dengan platform yang berbeda dapat memiliki tampilan yang sama. Begitu juga dengan AWT menjamin tampilan look and feel pada aplikasi yang dijalankan pada dua mesin yang berbeda menjadi terlihat sama.

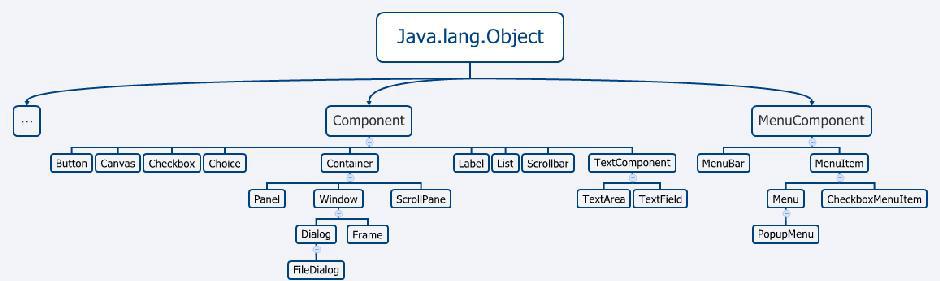
Swing API dibangun dari beberapa API yang mengimplementasikan beberapa jenis bagian dari AWT. Komponen AWT dapat digunakan bersama-sama dengan komponen Swing. Nama dari komponen GUI milik Swing hampir sama persis dengan komponen GUI milik AWT. Perbedaan jelas terdapat pada penamaan komponen.

Pada dasarnya, nama komponen Swing sama dengan nama komponen AWT tetapi dengan tambahan huruf “J” pada prefixnya. Sebagai contoh, satu komponen dalam AWT adalah Button class, pada Swning nama komponen tersebut menjadi Jbutton class.

**KOMPONEN GUI PADA AWT**

****

Pembagian komponen GUI pada AWT yang terdapat di dalam paket **java.awt** dapat dilihat pada gambar berikut.



**Gambar 8. 1**

Beberapa kelas penting pada kontainer yang telah disediakan oleh AWT antara lain:

* Component: Abstract class untuk objek yang dapat ditampilkan pada console dan berinteraksi dengan user. Merupakan bagian utama dari semua kelas AWT.
* Container: Abstract subclass dari Component class. Suatu komponen yang dapat menampung komponen lainnya.
* Panel: Turunan dari Container class. Suatu frame atau window tanpa titlebar, menubar tidak termasuk border.
* Window: Turunan dari Container class. Top level window, dimana artinya tidak bisa dimasukan dalam objek yang lainnya. Tidak memiliki border dan menu bar.
* Frame: Turunan dari Window class. Window dengan judul, menu bar, border, dan pengaturan ukuran di pojok.

Berikut contoh kode program untuk membuat GUI dengan AWT:



**Program 8-1**

****

* *contohAwt1.java* import java.awt.\*;

**

*/\* Agar kelas contohAwt1 berbentuk frame*

* *maka kelas tersebut harus meng-extend kelas Frame*

*\*/*

public class contohAwt1 extends Frame { contohAwt1() {

Button b = new Button("Tombol");

b.setBounds(50, 50, 50, 50); *// mengatur posisi tombol pada layar* add(b); *// menambahkan tombol pada frame*

setSize(500, 200); *// mengatur ukuran frame*

setTitle("Ini adalah contoh AWT pertama"); *// memberi judul frame* setLayout(null); *// pengaturan layout*

setVisible(true); *// secara default frame tidak visible*

}

public static void main(String [] args) {

* *membuat instance (obyek) frame* contohAwt1 Fr = new contohAwt1();

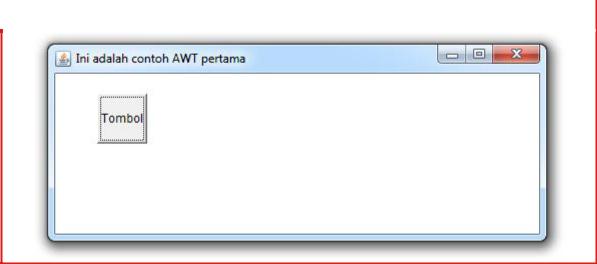
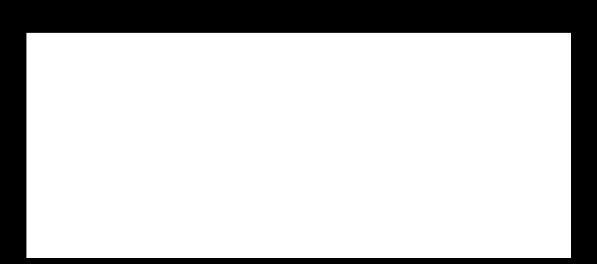
}

}





**Hasil Output**

****

**KOMPONEN GUI PADA SWING**

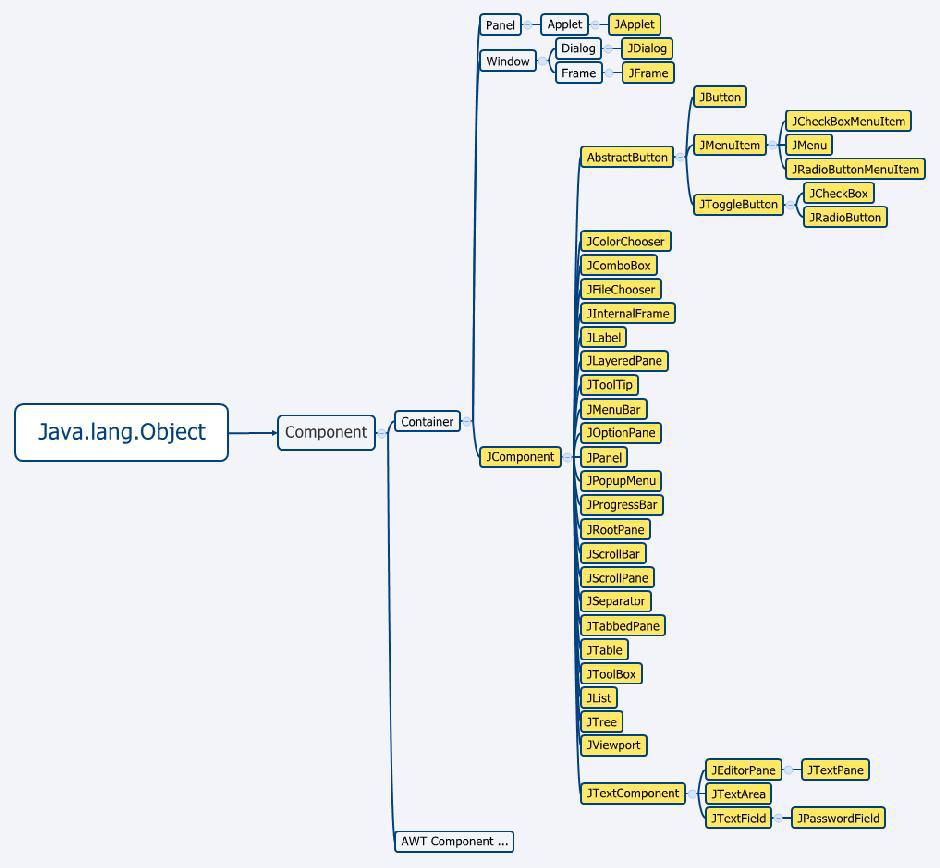
****

Komponen GUI pada Swing terdapat dalam paket **javax.swing**. Berikut adalah daftar dari beberapa kompoen Swing dan penjelasannya.

* JComponent: Kelas induk untuk semua komponen Swing, tidak termasuk top-level Container.
* JFrame: Turunan dan korepondensi pada frame class dalam package AWT tetapi keduanya sedikit tidak cocok dalam kaitannya dengan menambahkan komponen pada kontainer. Perlu mendapatkan content pane yang terbaru sebelum menambah sebuah komponen.
* JPanel: Turunan Jcomponent. Kontainer class sederhana tetapi bukan top-level.
* JApplet: Turunan dan korepondensi ke Applet class dalam package AWT. Juga sedikit tidak cocok dengan applet class dalam kaitannya dengan menambahkan komponen pada container.
* JButton: Tombol “push”. JButton adalah komponen berbentuk tombol. Komponen ini banyak digunakan sebagai eksekusi terhadap tindakan yang diinginkan. Pada aplikasi komputer, biasanya dibutuhkan tombol untuk mengeksekusi sebuah perintah.
* JLabel: komponen yang digunakan untuk membuat tulisan atau gambar pada frame sebagai suatuinformasi untuk pengguna program.
* JTextField: komponen yang digunakan untuk memasukkan sebaris string yang selanjutnya dapat digunakan sebagai input bagi proses selanjutnya.
* JTextArea: komponen yang mirip dengan JTextField tetapi dapat menampung lebih dari 1 baris.
* JCheckBox: komponen yang digunakan ketika pengguna memerlukan komponen untuk melakukan satu atau banyak pilihan sekaligus.
* JRadioButton: komponen yang digunakan ketika pengguna perlu memilih satu diantara beberapa pilihan.
* JComboBox: komponen yang digunakan untuk memilih satu diantara sekian banyak pilihan yang berbentuk semacam TextField dan ada panah ke bawah.
* JFileChooser: Mengijinkan pengguna untuk memilih sebuah file.
* JColorChooser: Turunan Jcomponent. Mengijinkan pengguna untuk memilih warna.
* JTable: digunakan untuk menampilkan data dalam bentuk tabel (biasanya dalam pemrograman database).

* JScrollPane: komponen yang digunakan untuk menggerakkan obyek ke atas, ke bawah, atau ke samping agar semua obyek terlihat di layar.
* JMenu: komponen yang digunakan untuk membuat menu.
* JInternalFrame: frame yang hanya dapat berada dalam frame lain.
* JOptionPane: Turunan Jcomponent. Disediakan untuk mempermudah menampilkan popup kotak dialog.
* JDialog: Turunan dan korespondensi pada dialog class dalam package AWT. Biasanya digunakan untuk menginformasikan sesuatu kepada pengguna atau prompt pengguna untuk input.

Pembagian komponen GUI pada Swing (berwarna kuning) dapat dilihat pada gambar di bawah.



**Gambar 8. 2**

Berikut contoh kode program untuk membuat GUI dengan Swing:



**Program 8-2**

****

* *contohSwing1.java* import javax.swing.\*;

**

public class contohSwing1 {

contohSwing1() {

JFrame f = new JFrame("Ini adalah contoh Swing"); *// membuat frame* f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); JPanel p = new JPanel(); *// membuat obyek panel*

p.add(new JButton("Tombol 1")); *// menambahkan tombol 1* p.add(new JButton("Tombol 2")); *// menambahkan tombol 2* p.add(new JButton("Tombol 3"));

f.add(p); *// mengaitkan panel ke frame* f.setSize(500, 200); *// membuat ukuran* f.setVisible(true); *// menampilkan frame*

}

public static void main(String [] args) { contohSwing1 fr = new contohSwing1();

}

}



**Hasil Output**

****

**LAYOUT**

****

Layout digunakan untuk mengatur posisi dari komponen visual penyusun program sesuai dengan desain user interface. Java telah menyediakan beberapa pilihan layout, dimana keputusan untuk menggunakan jenis layout bergantung pada jenis aplikasi yang ingin dibuat serta tingkat kerapian yang diinginkan. Berikut beberapa layout manajer yang terdapat di dalam Java:

* NoneLayout: jenis layout yang dapat menghasilkan tampilan yang rapi karena kita dapat mengatur posisi komponen secara detail berdasarkan koordinatnya. Konsekuensinya dengan layout ini waktu yang diperlukan relatif lebih banyak dibandingkan layout yang lain karena perlu menentukan posisi koordinat setiap komponen.
* FlowLayout: jenis pengaturan layout yang paling sederhana, dimana semua komponen akan tersusun dari kiri ke kanan sepanjang frame, dan akan pindah ke bawah bila telah sampai batas kanan frame.

Layout manager diatur dengan menggunakan metode setLayout dari Container class.



**Program 8-3**

*// FlowLayout.java*

import javax**.**swing**.**JButton**;**

import javax**.**swing**.**JFrame**;**

import javax**.**swing**.**JPanel**;**

public class FlowLayout **{**

public FlowLayout**() {**

JFrame fr **=** new JFrame**(**"FlowLayout Demo"**);**

fr**.**setDefaultCloseOperation**(**JFrame**.**EXIT\_ON\_CLOSE**);**

JPanel pl **=** new JPanel**();**

pl**.**setLayout**(**new java**.**awt**.**FlowLayout**());**

pl**.**add**(**new JButton**(**"Button 1"**));**

pl**.**add**(**new JButton**(**"Button 2"**));**

pl**.**add**(**new JButton**(**"Button 3"**));**

pl**.**add**(**new JButton**(**"Long-Named Button 4"**));**

pl**.**add**(**new JButton**(**"5"**));**

fr**.**add**(**pl**);**

fr**.**pack**();**

fr**.**setVisible**(**true**);**

**}**

public static void main**(**String **[]** args**) {** new FlowLayout**();**

**}**

**}**

****

**Hasil Output**

****

* BorderLayout: jenis layout yang bekerja dengan membagi frame menjadi lima bagian yaitu top, bottom, left, right, dan center. Komponen visual dapat diletakan pada bagian-bagian tersebut.



**Program 8-4**

* *BorderLayoutDemo.java* import java**.**awt**.**BorderLayout**;** import javax**.**swing**.**JButton**;** import javax**.**swing**.**JFrame**;** import javax**.**swing**.**JPanel**;**

public class BorderLayoutDemo **{**

public BorderLayoutDemo**() {**

JFrame fr **=** new JFrame**(**"BorderLayout Demo"**);**

fr**.**setDefaultCloseOperation**(**JFrame**.**EXIT\_ON\_CLOSE**);**

JPanel pl **=** new JPanel**();**

pl**.**setLayout**(**new BorderLayout**());**

pl**.**add**(**new JButton**(**"Button 1 (PAGE\_START)"**),** BorderLayout**.**PAGE\_START**);**

pl**.**add**(**new JButton**(**"Button 2 (CENTER)"**),** BorderLayout**.**CENTER**)** pl**.**add**(**new JButton**(**"Button 3 (LINE\_START)"**),**

BorderLayout**.**LINE\_START**);**

pl**.**add**(**new JButton**(**"Long-Named Button 4 (PAGE END)"**),** BorderLayout**.**PAGE\_END**);**

pl**.**add**(**new JButton**(**"5 (LINE END)"**),** BorderLayout**.**LINE\_END**);**

fr**.**add**(**pl**);**

fr**.**pack**();**

fr**.**setVisible**(**true**);**

**}**

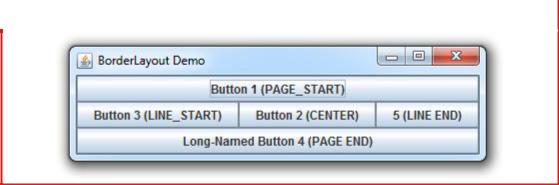
public static void main**(**String **[]** args**) {** new BorderLayoutDemo**();**

**}**

**}**

****

**Hasil Output**

****

* GridLayout: jenis layout yang bekerja berdasarkan baris dan kolom. Dengan layout ini kita dapat memberikan argumen banyaknya baris dan kolom sesuai dengan kebutuhan.



**Program 8-4**

* *GridLayoutDemo.java* import java**.**awt**.**GridLayout**;** import javax**.**swing**.**JButton**;** import javax**.**swing**.**JFrame**;** import javax**.**swing**.**JPanel**;**

public class GridLayoutDemo **{**

public GridLayoutDemo**() {**

JFrame fr **=** new JFrame**(**"GridLayout Demo"**);**

fr**.**setDefaultCloseOperation**(**JFrame**.**EXIT\_ON\_CLOSE**);**

JPanel pl **=** new JPanel**();**

pl**.**setLayout**(**new GridLayout**(**3**,** 2**));**

pl**.**add**(**new JButton**(**"Button 1"**));**

pl**.**add**(**new JButton**(**"Button 2"**));**

pl**.**add**(**new JButton**(**"Button 3"**));**

pl**.**add**(**new JButton**(**"Long-Named Button 4"**));** pl**.**add**(**new JButton**(**"5"**));**

fr**.**add**(**pl**);**

fr**.**pack**();**

fr**.**setVisible**(**true**);**

**}**

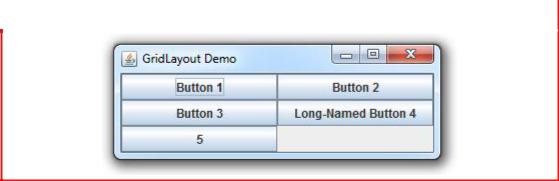
public static void main**(**String **[]** args**) {** new GridLayoutDemo**();**

**}**

**}**

****

**Hasil Output**

****

* CardLayout: komponen ditumpuk seperti kartu dan hanya satu komponen yang tampil pada satu waktu. Contoh berikut adalah penggunaan dari cardlayout, dimana apabila tombol pertama diklik maka akan menampilkan tombol kedua dan seterusnya.



**Program 8-4**

* *CardLayoutDemo.java* import java**.**awt**.**CardLayout**;** import java**.**awt**.**event**.**ActionEvent**;** import java**.**awt**.**event**.**ActionListener**;** import javax**.**swing**.**JButton**;**

import javax**.**swing**.**JFrame**;** import javax**.**swing**.**JPanel**;**

public class CardLayoutDemo extends JPanel **{** CardLayout cards **=** new CardLayout**( );** public CardLayoutDemo**( ) {**

setLayout**(**cards**);**

ActionListener listener **=** new ActionListener**() {** public void actionPerformed**(**ActionEvent e**) {**

cards**.**next**(**CardLayoutDemo**.**this**);**

**}**

**};**

JButton button**;**

button **=** new JButton**(**"Button 1"**);**

button**.**addActionListener**(**listener**);**

add**(**button**,** "Button 1"**);**

button **=** new JButton**(**"Button 2"**);**

button**.**addActionListener**(**listener**);**

add**(**button**,** "Button 2"**);**

button **=** new JButton**(**"Button 3"**);** button**.**addActionListener**(**listener**);** add**(**button**,** "Button 3"**);**

**}**

public static void main**(**String**[]** args**) {** JFrame frame **=** new JFrame**(**"Card"**);**

frame**.**setDefaultCloseOperation**(** JFrame**.**EXIT\_ON\_CLOSE **);** frame**.**setSize**(**200**,** 200**);**

frame**.**setLocation**(**200**,** 200**);**

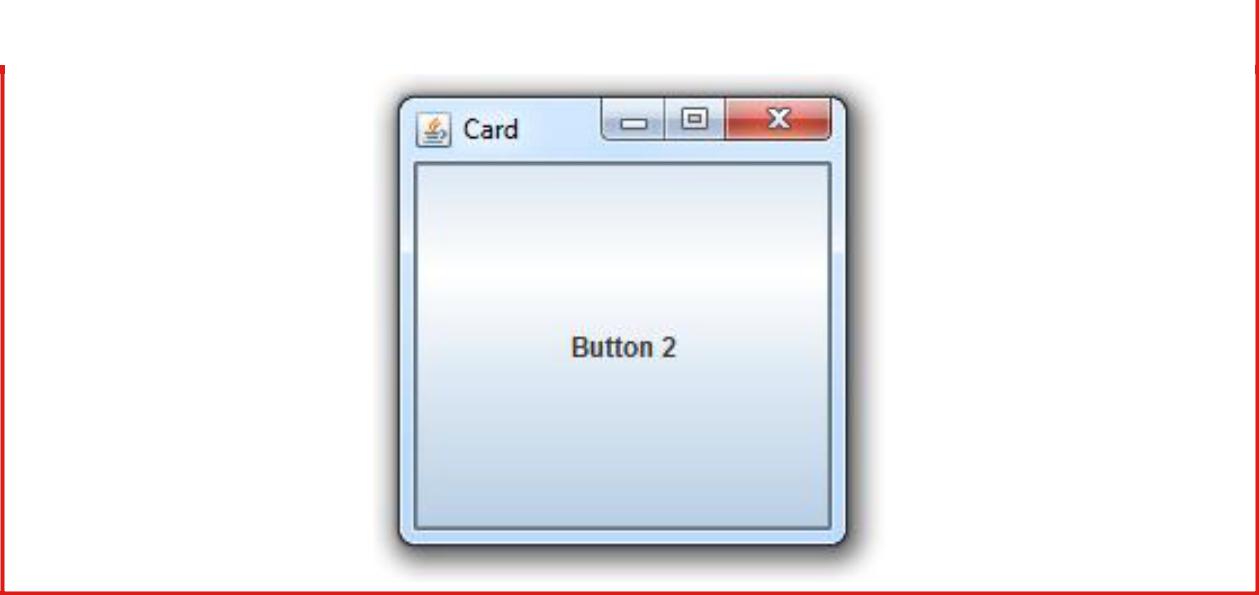
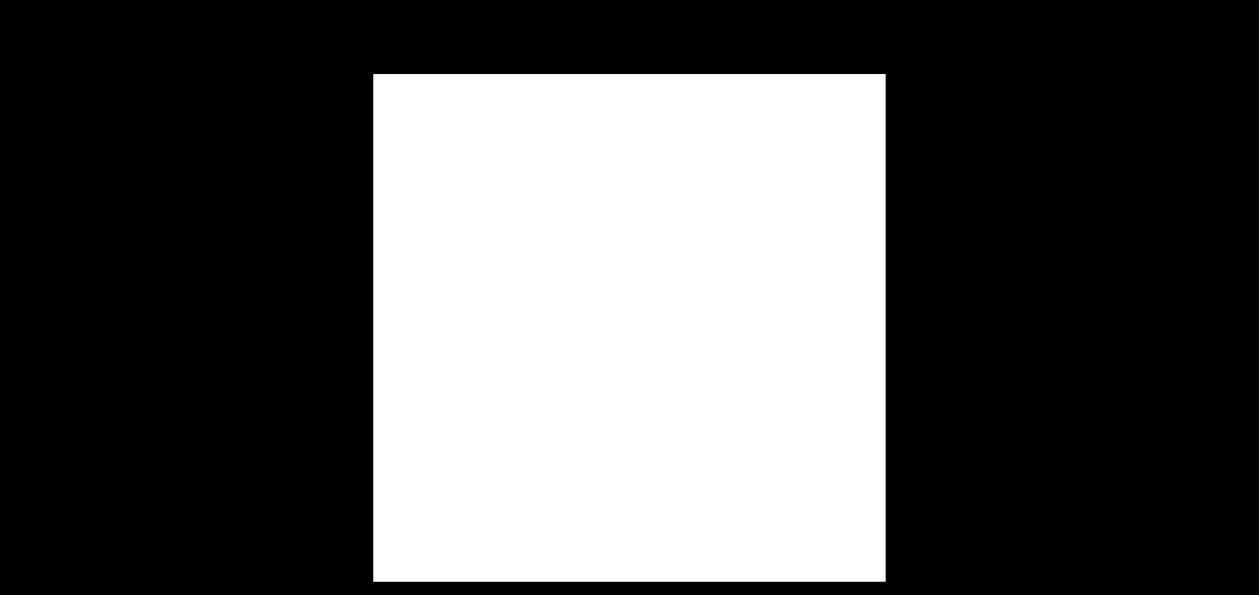
frame**.**setContentPane**(**new CardLayoutDemo**());** frame**.**setVisible**(**true**);**

**}**

**}**

****

**Hasil Output**

****

* GridBagLayout: ukuran grid bisa berubah dan lebih dari satu komponen dapat masuk pada satu grid.



**Program 8-5**

*// GridBagLayoutDemo.java*

import java**.**awt**.**GridBagConstraints**;**

import java**.**awt**.**GridBagLayout**;**

import java**.**awt**.**Insets**;**

import javax**.**swing**.**JButton**;**

import javax**.**swing**.**JFrame**;**

import javax**.**swing**.**JPanel**;**

public class GridBagLayoutDemo **{**

public GridBagLayoutDemo**() {**

JFrame frame **=** new JFrame**(**"GridBagLayout Demo"**);**

frame**.**setDefaultCloseOperation**(** JFrame**.**EXIT\_ON\_CLOSE **);**

JPanel pl **=** new JPanel**();**

JButton btn**;**

pl**.**setLayout**(**new GridBagLayout**());** GridBagConstraints c **=** new GridBagConstraints**();** c**.**fill **=** GridBagConstraints**.**HORIZONTAL**;**

btn **=** new JButton**(**"Button 1"**);**

c**.**weightx **=** 0.5**;**

c**.**gridx **=** 0**;**

c**.**gridy **=** 0**;**

pl**.**add**(**btn**,** c**);**

btn **=** new JButton**(**"Button 2"**);**

c**.**fill **=** GridBagConstraints**.**HORIZONTAL**;** c**.**weightx **=** 0.5**;**

c**.**gridx **=** 1**;**

c**.**gridy **=** 0**;**

pl**.**add**(**btn**,** c**);**

btn **=** new JButton**(**"Button 3"**);**

c**.**fill **=** GridBagConstraints**.**HORIZONTAL**;** c**.**weightx **=** 0.5**;**

c**.**gridx **=** 2**;**

c**.**gridy **=** 0**;**

pl**.**add**(**btn**,** c**);**

btn **=** new JButton**(**"Long-Named Button 4"**);** c**.**fill **=** GridBagConstraints**.**HORIZONTAL**;**

c**.**ipady **=** 40**;** *//make this component tall*

c**.**weightx **=** 0.0**;**

c**.**gridwidth **=** 3**;**

c**.**gridx **=** 0**;**

c**.**gridy **=** 1**;**

pl**.**add**(**btn**,** c**);**

btn **=** new JButton**(**"5"**);**

c**.**fill **=** GridBagConstraints**.**HORIZONTAL**;**

|  |  |  |
| --- | --- | --- |
| c**.**ipady **=** | 0**;** | *//reset to default* |
| c**.**weighty | **=** 1.0**;** | *//request any extra vertical space* |

c**.**anchor **=** GridBagConstraints**.**PAGE\_END**;** *//bottom of space* c**.**insets **=** new Insets**(**10**,**0**,**0**,**0**);** *//top padding*

|  |  |  |  |
| --- | --- | --- | --- |
| c**.**gridx | **=** | 1**;** | *//aligned with button 2* |
| c**.**gridwidth **=** 2**;** | | | *//2 columns wide* |
| c**.**gridy | **=** | 2**;** | *//third row* |
| pl**.**add**(**btn**,** c**);** | | |  |
|  |  |  |  |

frame**.**add**(**pl**);**

frame**.**pack**();**

frame**.**setVisible**(**true**);**

**}**

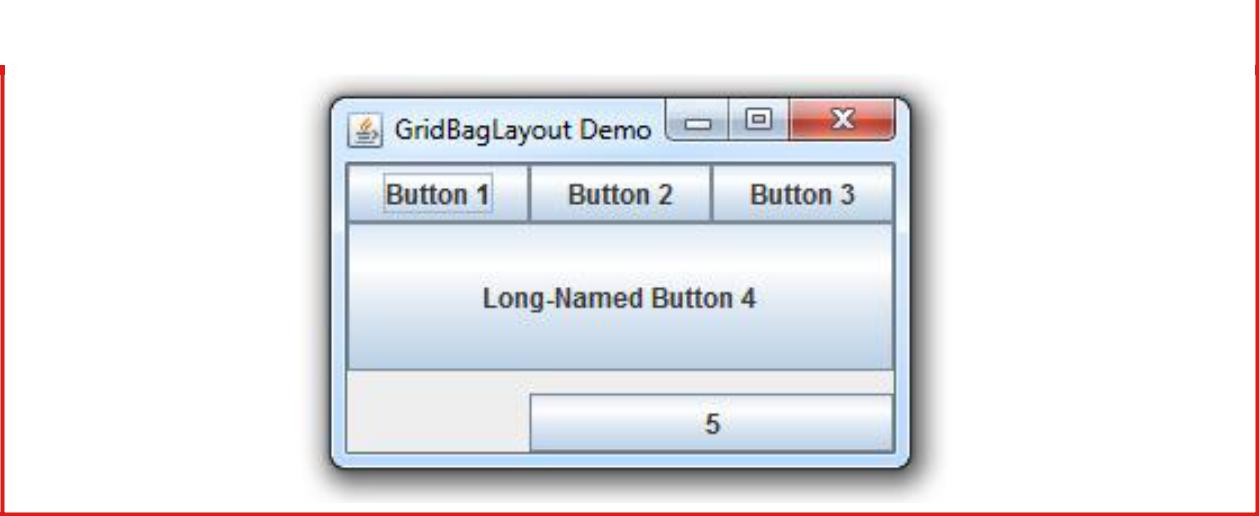
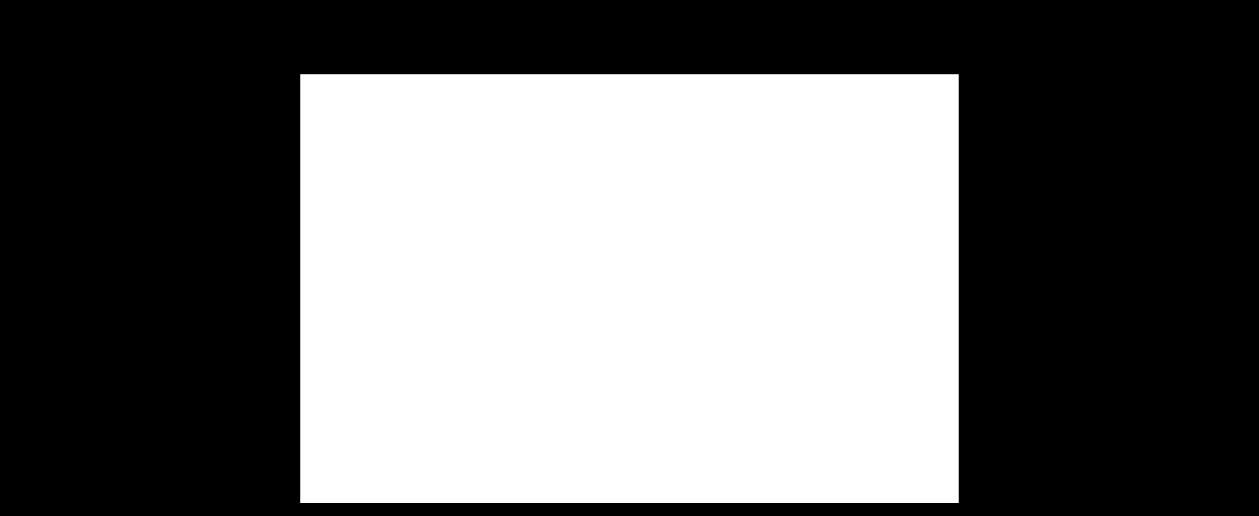
public static void main**(**String**[]** args**) {** new GridBagLayoutDemo**();**

**}**

**}**

****

**Hasil Output**

****

* BoxLayout: komponen diletakan ke dalam satu baris atau kolom

**Program 8-6**

****

* *BoxLayoutDemo.java* import java**.**awt**.**Component**;** import java**.**awt**.**Container**;** import javax**.**swing**.**BoxLayout**;** import javax**.**swing**.**JButton**;** import javax**.**swing**.**JFrame**;** import javax**.**swing**.**JPanel**;**

public class BoxLayoutDemo **{**

public BoxLayoutDemo**() {**

JFrame frame **=** new JFrame**(**"BoxLayoutDemo"**);**

frame**.**setDefaultCloseOperation**(**JFrame**.**EXIT\_ON\_CLOSE**);**

JPanel pl **=** new JPanel**();**

pl**.**setLayout**(**new BoxLayout**(**pl**,** BoxLayout**.**Y\_AXIS**));**

addButton**(**"Button 1"**,** pl**);**

addButton**(**"Button 2"**,** pl**);**

addButton**(**"Button 3"**,** pl**);**

addButton**(**"Long-Named Button 4"**,** pl**);**

addButton**(**"5"**,** pl**);**

frame**.**add**(**pl**);**

frame**.**pack**();**

frame**.**setVisible**(**true**);**

**}**

private void addButton**(**String text**,** Container container**) {**

JButton btn **=** new JButton**(**text**);**

btn**.**setAlignmentX**(**Component**.**CENTER\_ALIGNMENT**);**

container**.**add**(**btn**);**

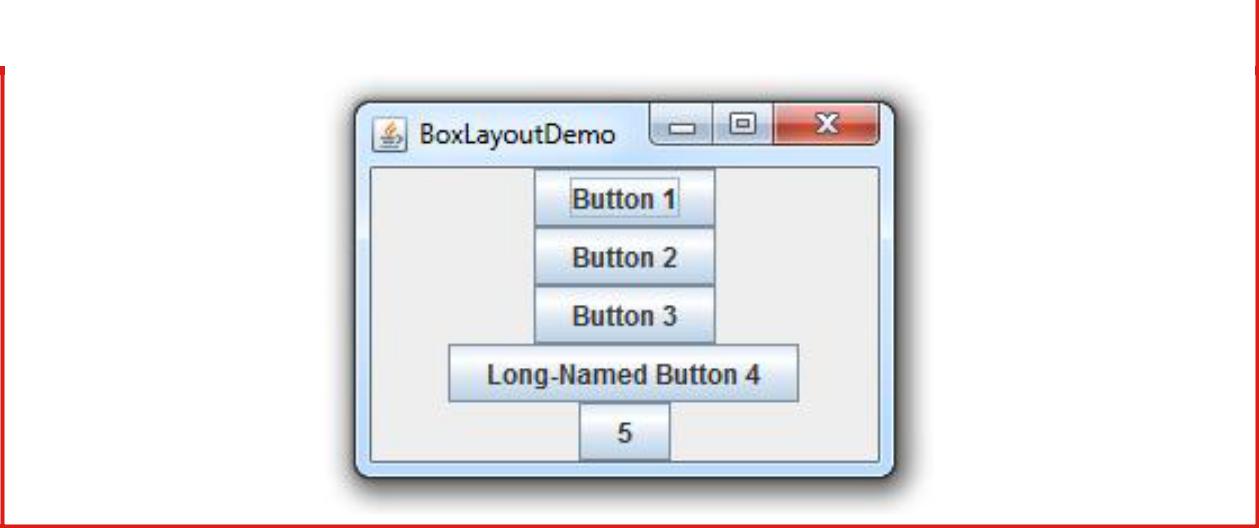
**}**

public static void main**(**String **[]** args**) {** new BoxLayoutDemo**();**

**}**

**}**

****

**Hasil Output**

****