



GUI (Graphical User Interface)



Pengantar GUI

- GUI (Graphical User Interface), memungkinkan user untuk berinteraksi dengan komputer secara lebih baik.
- Pengguna berinteraksi dengan berbagai elemen grafis, diantaranya: button, dialog box, menu, dsb.
- Sejak 1994 Sun Microsystem berupaya untuk membuat JAVA memiliki perangkat bergaya desktop dengan ToolkitJAVA yang dikenal sebagai AWT (*Abstract Windowing Toolkit*)
- Tahun 1997, muncul JFC (*Java Foundation Class*) untuk kepentingan pembuatan aplikasi Window yang merupakan pengembangan dari AWT.
- JFC mempunyai kode swing yang terdapat pada package yang disediakan pada JAVA SDK versi 1.2 ke atas (JAVA 2)

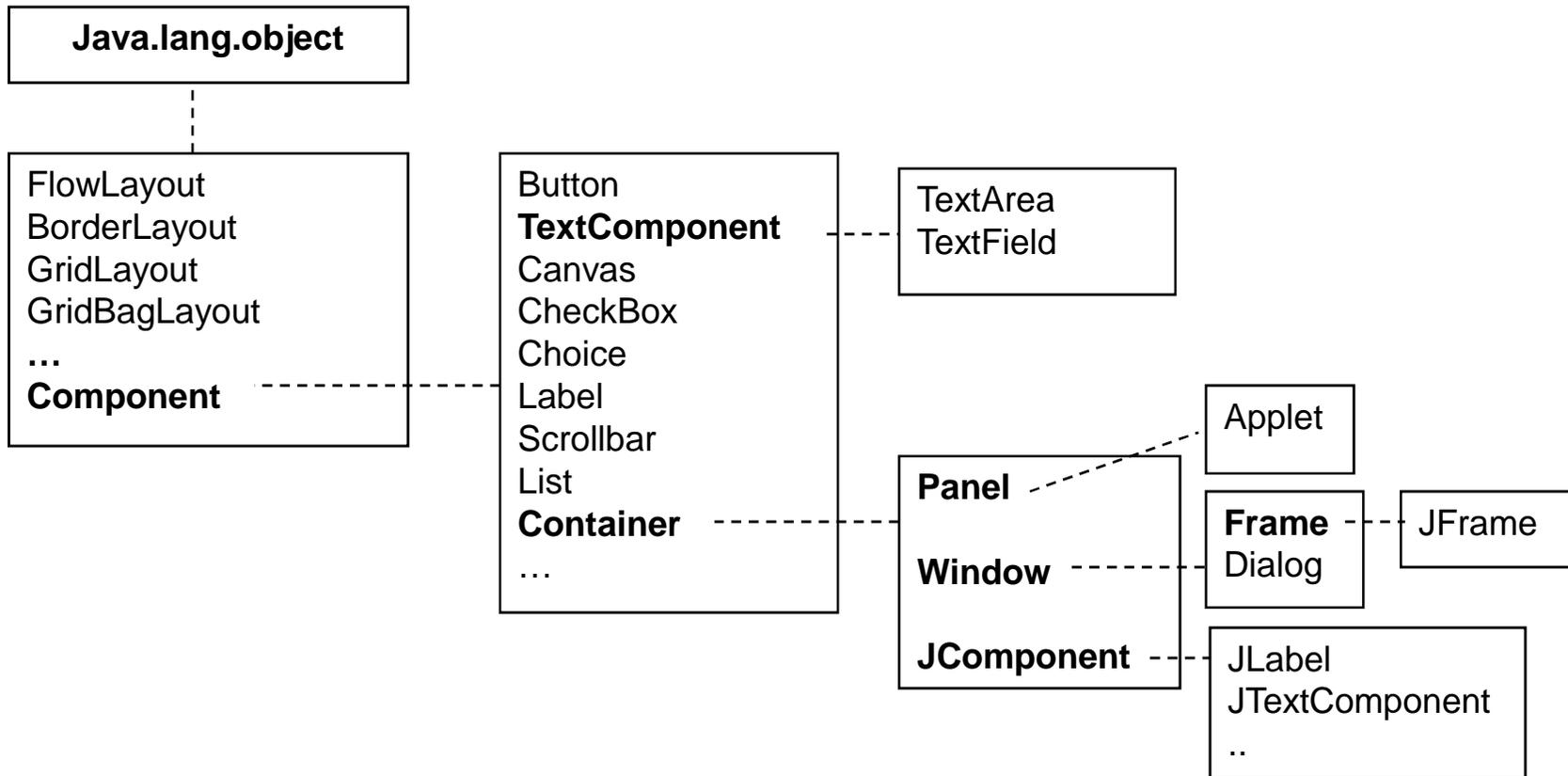
JAVA menyediakan dua *graphical library*.

- AWT (Abstract Windowing Toolkit)
- Swing

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



Hierarki Kelas





AWT (Abstract Windowing Toolkit)

AWT terdapat dalam package **java.awt** package ini berisi komponen-komponen GUI yang bersifat platform oriented atau tergantung pada suatu platform sistem operasi.

Kelas-Kelas yang perlu dipahami :

- 1. Kelas Komponen (Component) = Kelas Abstrak dari semua komponen yang digunakan untuk merancang GUI. Kelas ini memiliki beberapa kelas konkrit dari komponen-komponen kontrol GUI. Seperti Button, Label, CheckBox dan sebagainya.**
- 2. Kelas Kontainer (Container) = sub kelas dari Component yang berfungsi untuk “memuat” komponen-komponen lainnya. Kelas ini memiliki beberapa subkelas seperti Panel, Window, Frame, dan sebagainya.**



AWT (Abstract Windowing Toolkit)

- 3. Kelas Window = Turunan dari Container class. Kelas window tidak bisa dimasukkan dalam komponen atau kontainer lainnya karena merupakan top level class. Tidak memiliki Title, border dan menubar. Kelas ini jarang digunakan, namun ada dua subkelas yang sering dipakai, yaitu kelas Dialog dan Kelas Frame.**
- 4. Kelas Frame = Turunan dari window class. memiliki judul, menubar, border, icon dan menu. Dapat diubah ukurannya. Kontainer yang sering digunakan untuk menempatkan komponen secara langsung.**
- 5. Kelas Panel = Turunan dari Container Class. Sebuah frame atau window tanpa titlebar, menubar. Superclass dari applet class. Panel ini sering digunakan dalam pengorganisasian tata letak komponen(layout). Ada baiknya kita mengemas komponen-komponen dalam panel lebih dahulu, selanjutnya panel-panel kita tambahkan ke frame.**



Layout Management

Ketika merancang GUI, Selain menentukan komponen apa yang akan digunakan, anda juga harus mempertimbangkan bagaimana tata letak komponen- komponen tersebut. Pengaturan tata letak komponen dalam kontainer ini dikenal sebagai Layout Management.

Pada Java, ketika merancang dan membuat aplikasi menggunakan development tool (kompiler dan interpreter) yang berbasis pemrograman text, kita harus menentukan komponen dan layoutnya secara manual. Namun untuk development tool yang mendukung GUI anda cukup mengatur penempatan komponen saja.



Layout Management

Java Menyediakan lima kelas untuk mendukung proses perancangan tata letak GUI :

1. **FlowLayout** = Didasarkan pada baris yaitu mulai dari kiri ke kanan dan baris atas ke baris bawah. Default untuk kelas panel dan Applet.
2. **GridLayout** = Didasarkan pada persegi panjang grid. Komponen ditambahkan dari kiri ke kanan, atas ke bawah. Semua area berukuran sama. Konstruktor menetapkan baris dan kolom.
3. **BorderLayout** = Didasarkan pada lima posisi, yaitu east, west, north, south dan center. Default untuk window, frame, dan Dialog.
4. **CardLayout** = Komponen diperlakukan mirip tumpukan kartu. yang visible komponen teratas.
5. **GridBagLayout** = sama seperti GridLayout namun suatu komponen dapat menempati multiple cell.



Layout Management

Layout manager dapat diatur menggunakan metode *setLayout* dari *Container* class. Metode ini dapat ditulis sebagai berikut.

```
NamaObjekContainer.setLayout(new JenisLayout);
```

Contoh : `frame.setLayout(new FlowLayout());`

Jika Anda memilih untuk tidak menggunakan layout manager, Anda dapat mengisi null sebagai argumen untuk metode ini.

Contoh : `frame.setLayout(null);`

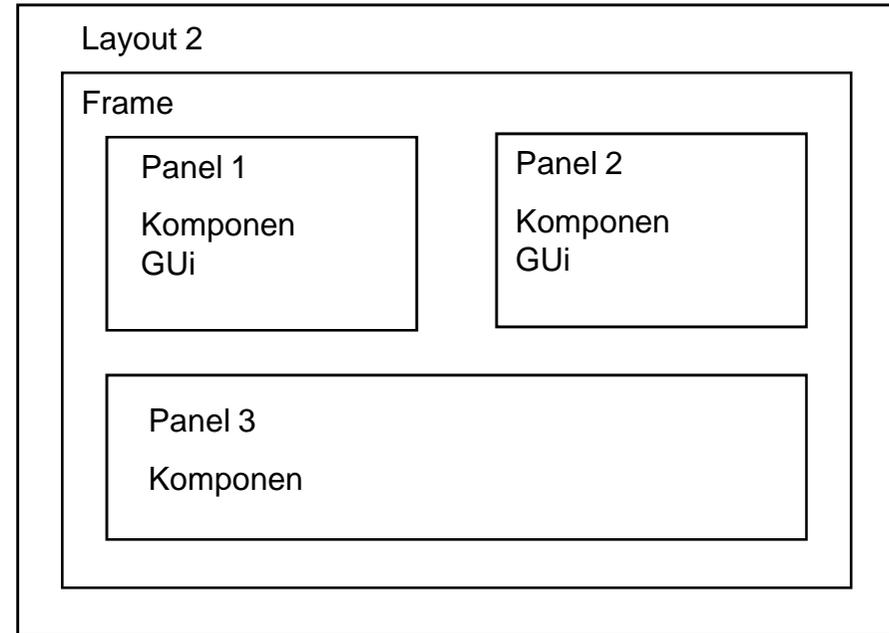
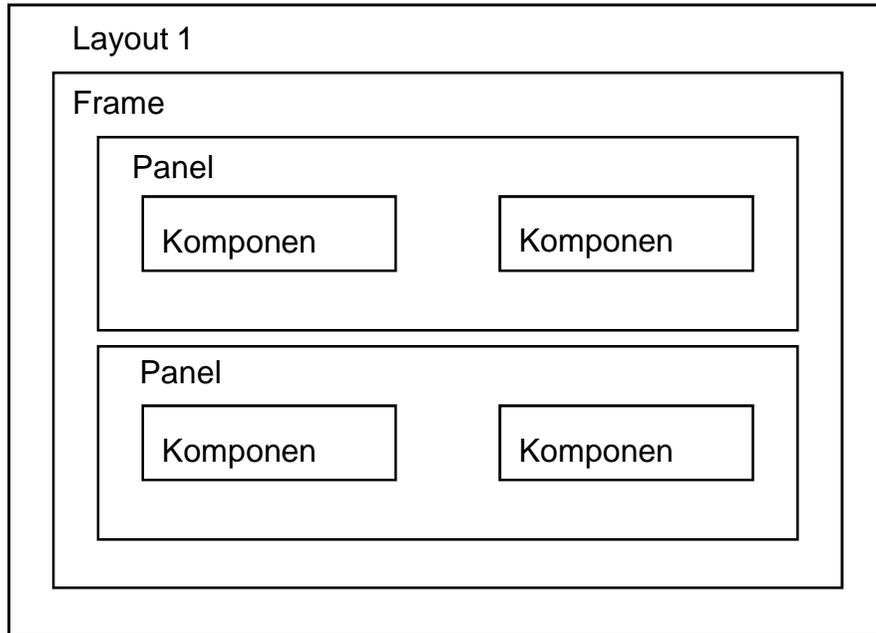
Tetapi selanjutnya, Anda akan mengatur posisi elemen secara manual dengan menggunakan metode *setBounds* dari *Components* class.

```
public void setBounds(int x, int y, int width, int height)
```

Contoh : `Button1.setBounds(10,10,200,100);`



Tata Letak





Contoh Membuat Frame dengan Frame

```
import java.awt.*;
public class ContohFrame1 extends Frame{ // Dengan Frame
    public static void main(String args[]) {
        ContohFrame1 frmUtama = new ContohFrame1();
        frmUtama.setTitle("Contoh Frame");
        frmUtama.setSize(500, 500); //Coba hilangkan baris ini
        frmUtama.setResizable(true); // Coba False
        frmUtama.setVisible(true); //Coba hilangkan baris ini
    }
}
```

Catatan :

setSize(int,int); untuk panjang dan lebar frame
setResizable(boolean); memungkinkan frame dirubah ukurannya
setVisible(boolean); frame terlihat jika true / tidak terlihat jika
false



Contoh Membuat Frame dengan JFrame

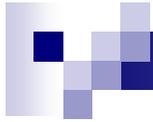
```
import javax.swing.*;
public class ContohFrame2 extends JFrame{
    public static void main(String args[]) {
        ContohFrame2 frmUtama = new ContohFrame2();
        frmUtama.setTitle("Contoh Frame");
        // Setting Ukuran Frame setSize atau SetBounds
        frmUtama.setSize(500, 500); //Coba dengan setSize
        // frmUtama.setBounds(300,100,500,500); // coba dengan setBounds
        frmUtama.setResizable(true); // Coba False
        frmUtama.setVisible(true); //Coba hilangkan baris ini
    }
}
```

Catatan :

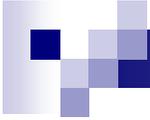
```
setBounds(int x,int y,int width,int Height) ;
```



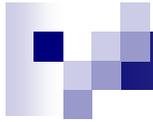
Contoh FlowLayout Manager



Contoh BorderLayout Manager



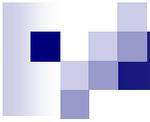
Contoh GridLayout Manager



Contoh GridBagLayout Manager



Contoh CardLayout Manager



Latihan