

# PRAKTIKUM 27

---

## ANTAR MUKA GRAFIS

---

### A. TUJUAN PEMBELAJARAN

1. Memahami konsep Graphical User Interface (GUI).
2. Mampu membuat program yang menerapkan konsep GUI.
3. Mampu membuat program berbasis GUI yang melibatkan penggunaan komponen GUI.

### B. DASAR TEORI

Ciri utama dari tampilan model ini adalah seluruh data ditampilkan dalam bentuk teks (*text based*). Model tampilan ini memiliki kelemahan yaitu lebih sulit membuat tampilan program menjadi indah dan menarik, dikarenakan kurangnya dukungan warna. Monitor model teks hanya mendukung warna sampai 16, berbeda dengan model grafik yang mendukung warna sampai  $2^{24}$  (2 pangkat 24) bahkan bisa lebih, tergantung *graphic-card* yang digunakan.

Bab ini akan membahas tentang Graphical User Interface (GUI), suatu metode untuk antar-muka komputer berbasis grafis. GUI digunakan dalam pembuatan program aplikasi dengan mempertimbangkan dua aspek yaitu keindahan tampilan dan kemudahan penggunaan program. Keindahan tampilan akan mempengaruhi orang untuk melihat dan merasa senang terhadap program yang kita buat, dan kemudahan penggunaan program akan berdampak pada ketertarikan orang untuk terus menggunakan program yang kita buat. Di dunia bisnis, hal ini merupakan target jangka panjang.

Materi dalam bab ini akan disajikan secara sistematis dan terstruktur, mulai dari pengenalan objek utama, penambahan objek-objek pendukung (lebih dikenal sebagai komponen) sampai interaksi komputer dengan user. Untuk melengkapi uraian tentang GUI ini, pada akhir bab kita membuat sebuah project yang diberi nama "Kalkulator Mini". Project ini menyajikan pembuatan sebuah kalkulator sederhana yang bisa digunakan untuk melakukan proses perhitungan

matematika biasa. Beberapa fungsi yang rumit dari kalkulator sesungguhnya tidak dimasukkan dalam project ini. Anda bisa mengembangkan project ini secara mandiri.

Pembahasan kita mulai class JFrame. Dalam lingkup Java, sebuah interface *window* dikenal dengan sebutan *frame*. Bagi anda yang pernah menggunakan software Delphi (bahasa visual untuk Pascal) tentu pernah menjumpai komponen TFrame. Perlu diingat bahwa frame dalam lingkup Delphi berbeda dengan frame dalam lingkup Java. Frame dalam lingkup Java identik dengan TForm di Delphi atau Form di Visual Basic. Perlu juga diketahui bahwa seluruh pembahasan pada bab ini mengacu kepada sistem operasi Windows, tepatnya Windows 2000 atau Windows XP. Anda yang menggunakan sistem operasi lain seperti UNIX atau LINUX mungkin akan melihat beberapa perbedaan tampilan yang tidak mengganggu.

Jika anda membuka program Windows Explorer maka secara tidak sadar anda telah mengaktifkan sebuah window (atau frame). Frame adalah sebuah objek yang merepresentasikan suatu area di layar yang berisi sejumlah objek lain untuk menyampaikan informasi kepada user. Kita bisa menambahkan sejumlah komponen lain ke dalam sebuah framw dengan tujuan yang sesuai dengan fungsi komponen tersebut.

Sebuah objek yang merupakan *instansiasi* dari class JFrame bisa dipanggil melalui dua cara, yaitu:

1. Dipanggil melalui class utama yang mengandung fungsi main().
2. Dipanggil melalui class lain yang merupakan class tersendiri turunan dari JFrame.

### **Manajemen Pemasangan Komponen**

Manajemen pemasangan komponen (*layout management*) diperlukan untuk mengatur penempatan komponen di dalam frame agar bisa menghasilkan bentuk *interface* yang menarik. Penggunaan *layout-manager* dalam mengatur letak komponen juga akan memudahkan kita menempelkan komponen pada frame.

Java menyediakan sejumlah metode dalam mengatur penempelan komponen ke dalam frame. Pada bagian ini kita hanya akan membahas tiga metode: FlowLayout, GridLayout dan BorderLayout. Masing-masing metode memiliki keunggulan dan kekurangan dibandingkan dengan metode yang lain. Hal yang menarik adalah kita tidak harus terpaku pada satu metode

dalam mengatur pemasangan komponen; kita boleh mengkombinasikan beberapa metode agar pemasangan komponen menjadi lebih baik dan mudah.

Method `setLayout(LayoutManager)` digunakan untuk mengatur jenis metode yang digunakan pada saat memasang komponen. Method ini bisa kita gunakan pada class `JFrame`, `JPanel` dan `Container`. Pembahasan selanjutnya akan menunjukkan penggunaan berbagai metode pengaturan penempelan komponen pada frame.

### C. TUGAS PENDAHULUAN

1. Pelajari layout – layout yang bisa digunakan untuk mengatur tata letak komponen, kemudian buat daftar mengenai layout tersebut, beserta penjelasan dan contoh tampilannya.

### D. PERCOBAAN

#### Percobaan 1: Membuat jendela sederhana dengan `JFrame`

```
1  import javax.swing.JFrame;
2
3  //-----
4  //  Definisi class MYFRAME
5  //-----
6  class MyFrame extends JFrame
7  {
8      MyFrame()
9      {
10         super ("Contoh frame");
11
12         show();
13     }
14 }
15
16 //-----
17 //  Class utama
18 //-----
19 public class JFrame_4
20 {
21     public static void main (String[] args)
22     {
23         new MyFrame();
24     }
25 }
```

## Percobaan 2: Menghapus Frame dari Memori

```
1  import javax.swing.JFrame;
2
3  //-----
4  //  Definisi class MYFRAME
5  //-----
6  class MyFrame extends JFrame
7  {
8      MyFrame()
9      {
10         super ("Contoh frame");
11         setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
12
13         show();
14     }
15 }
16
17 //-----
18 //  Class utama
19 //-----
20 public class JFrame_5
21 {
22     public static void main (String[] args)
23     {
24         new MyFrame();
25     }
26 }
```

## Percobaan 3: Frame dengan Ukuran Tetap

```
1  import javax.swing.JFrame;
2
3  //-----
4  //  Definisi class MYFRAME
5  //-----
6  class MyFrame extends JFrame
7  {
8      MyFrame()
9      {
10         super ("Contoh frame");
11         setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
12         setLocation (200,100);
13         setSize (300,100);
14         setResizable (false);
15
16         show();
17     }
18 }
19
20 //-----
```

```

21 // Class utama
22 //-----
23 public class JFrame_8
24 {
25     public static void main (String[] args)
26     {
27         new MyFrame();
28     }
29 }

```

#### Percobaan 4: Strategi “Tempel Langsung”

```

1  import javax.swing.*;
2  import java.awt.*;
3
4  class MyFrame extends JFrame
5  {
6      private Container container = new Container();
7      private JButton tombol = new JButton ("Tombol");
8
9      MyFrame()
10     {
11         super ("Strategi 'Tempel-Langsung'");
12
13         setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
14         setSize (250,80);
15         setLocation (100,100);
16
17         container = getContentPane();
18
19         container.add (tombol);
20
21         show();
22     }
23 }
24
25 public class TempelLangsung_1
26 {
27     public static void main (String[] args)
28     {
29         new MyFrame();
30     }
31 }

```

#### Percobaan 5: Manajemen Pemasangan Komponen – Flowlayout

```

1  import javax.swing.*;
2  import java.awt.*;
3

```

```

4  class MyFrame extends JFrame
5  {
6      private Container container = new Container();
7      private JButton tombol = new JButton ("Tombol");
8      private JLabel tulisan = new JLabel ("Tulisan");
9
10     MyFrame()
11     {
12         super ("Demo FlowLayout");
13
14         setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
15         setSize (250,80);
16         setLocation (100,100);
17
18         container = getContentPane();
19         container.setLayout (new FlowLayout());
20
21         container.add (tombol);
22         container.add (tulisan);
23
24         show();
25     }
26 }
27
28 public class FlowLayout_1
29 {
30     public static void main (String[] args)
31     {
32         new MyFrame();
33     }
34 }

```

### Percobaan 6: Manajemen Pemasangan Komponen – GridLayout

```

1  import javax.swing.*;
2  import java.awt.*;
3
4  class MyFrame extends JFrame
5  {
6      private Container container = new Container();
7      private JButton tombol = new JButton ("Tombol");
8      private JLabel tulisan = new JLabel ("Tulisan");
9
10     MyFrame()
11     {
12         super ("GridLayout 1-baris 2-kolom");
13
14         setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
15         setSize (250,80);
16         setLocation (100,100);
17

```

```

18     container = getContentPane();
19     container.setLayout (new GridLayout(1,2));
20
21     container.add (tulisan);
22     container.add (tombol);
23
24     show();
25 }
26 }
27
28 public class GridLayout_1
29 {
30     public static void main (String[] args)
31     {
32         new MyFrame();
33     }
34 }

```

### Percobaan 7: Manajemen Pemasangan Komponen – BorderLayout

```

1  import javax.swing.*;
2  import java.awt.*;
3
4  class MyFrame extends JFrame
5  {
6      private Container container = new Container();
7
8      private JButton tombolUtara = new JButton ("NORTH");
9      private JButton tombolSelatan = new JButton ("SOUTH");
10     private JButton tombolTimur = new JButton ("EAST");
11     private JButton tombolBarat = new JButton ("WEST");
12     private JButton tombolTengah = new JButton ("CENTER");
13
14     MyFrame()
15     {
16         super ("Demo BorderLayout");
17
18         setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
19         setSize (250,130);
20         setLocation (100,100);
21
22         container = getContentPane();
23         container.setLayout (new BorderLayout());
24
25         container.add (tombolUtara, BorderLayout.NORTH);
26         container.add (tombolSelatan, BorderLayout.SOUTH);
27         container.add (tombolTimur, BorderLayout.EAST);
28         container.add (tombolBarat, BorderLayout.WEST);
29         container.add (tombolTengah, BorderLayout.CENTER);
30
31         show();

```

```

32     }
33 }
34
35 public class BorderLayout_1
36 {
37     public static void main (String[] args)
38     {
39         new MyFrame();
40     }
41 }

```

### Percobaan 8: Layout Komponen dalam JPanel

```

1  import javax.swing.*;
2  import java.awt.*;
3
4  class MyFrame extends JFrame
5  {
6      private Container container = new Container();
7      private JPanel jPanel = new JPanel();
8
9      private JLabel labelUserName =
10         new JLabel ("User name :");
11     private JLabel labelPassword =
12         new JLabel ("Password :");
13     private JTextField textFieldUserName =
14         new JTextField ("", 8);
15     private JTextField textFieldPassword =
16         new JTextField ("", 8);
17     private JButton buttonOK = new JButton ("OK");
18     private JButton buttonCancel = new JButton ("Cancel");
19
20     MyFrame()
21     {
22         super ("Demo JPanel - Login");
23
24         setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
25         setSize (250,10);
26         setLocation (100,100);
27
28         jPanel.setLayout (new GridLayout (3,2));
29         jPanel.add (labelUserName);
30         jPanel.add (textFieldUserName);
31         jPanel.add (labelPassword);
32         jPanel.add (textFieldPassword);
33         jPanel.add (buttonOK);
34         jPanel.add (buttonCancel);
35
36         container = getContentPane();
37         container.add (jPanel);
38     }
39 }

```



```

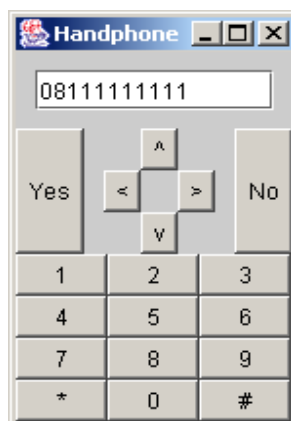
35     show();
36 }
37 }
38
39 public class JPanel_2
40 {
41     public static void main (String[] args)
42     {
43         new MyFrame();
44     }
45 }

```

## E. LATIHAN

### Latihan 1: Membuat tampilan HP mini

Buatlah tampilan HP seperti pada gambar berikut :

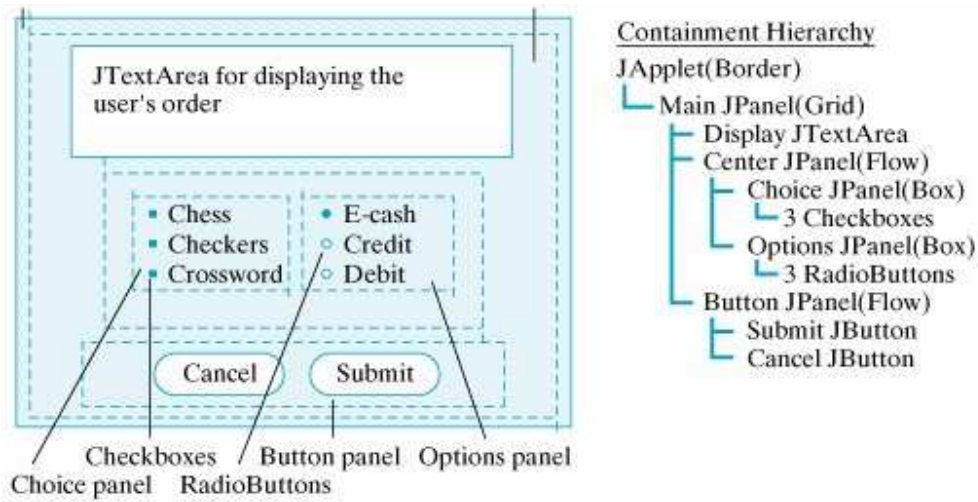


### Latihan 2: Membuat tampilan input data

Buatlah tampilan input data seperti gambar berikut :



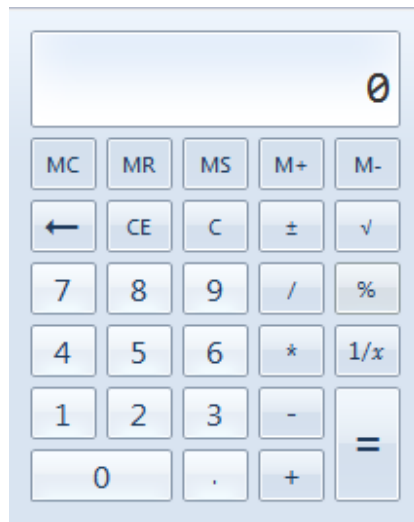
**Latihan 3: Desain user interface sesuai petunjuk pada gambar berikut**



Petunjuk : Gantilah JApplet dengan JPanel

**F. TUGAS**

Buatlah tampilan kalkulator mini seperti pada gambar berikut :



**G. LAPORAN RESMI**

Kumpulkan hasil percobaan di atas dan tambahkan analisa untuk tiap percobaan, latihan, dan tugas yang telah dibuat.