

PRAKTIKUM 29

GUI: KALKULATOR MINI

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.
4. Mampu membuat program berbasis GUI yang bisa menerima respon user, baik melalui mouse maupun keyboard.

B. DASAR TEORI

Project ini kami beri nama Kalkulator Mini, sebuah project sederhana yang mengimplementasikan teori GUI di Java. Dari namanya sudah bisa ditebak kegiatan utama dari project ini adalah membuat sebuah program aplikasi yang mensimulasikan sebuah kalkulator dengan kemampuan yang sangat minim. Kalkulator ini didesain untuk bisa menangani beberapa operasi matematika biasa, meliputi penjumlahan, pengurangan, perkalian, pembagian dan perpangkatan. Karena buku ini tidak didesain sebagai buku acuan tentang materi Struktur Data, maka teknik yang digunakan untuk menangani proses matematika tersebut tidak terlalu rumit.

Program akan menerima tiga inputan :

- Data pertama berupa data numerik yang bertindak sebagai data pertama yang akan diproses.
- Data kedua berupa data karakter yang menyatakan jenis proses yang akan dilakukan.
- Data ketiga berupa data numerik menyatakan data kedua yang akan diproses.

Seluruh data inputan tersebut diberikan melalui pengaktifan sejumlah objek button yang tersedia di layar. Kita tidak melakukan pengetikan, misalnya angka, untuk memasukkan data numerik,

sehingga objek textfield yang digunakan kita pasifkan. Method `setEnabled(boolean)` digunakan secara luas untuk tujuan mempasifkan suatu objek agar tidak bisa diakses.

Desain Tampilan

Kalkulator ini hanya memiliki 5 (lima) button untuk lima proses matematika, yaitu "+", "-", "*", "/" dan "^". Sebagai tambahan, sebuah objek berjenis `JCheckBox` disediakan untuk mengatur penggunaan karakter pemisah ribuan

Area tampilan			
<input checked="" type="checkbox"/> Pemisah ribuan			
Hapus	Batal	Ulang	^
7	8	9	*
4	5	6	/
1	2	3	-
0	.	=	+
Area informasi			Exit

Berikut ini keterangan setiap *item* pada desain tampilan tersebut:

- "Area tampilan" merupakan objek berjenis `JTextField`, tempat dimana angka-angka yang di-*click* akan ditampilkan. Area ini juga digunakan sebagai tempat untuk menampilkan hasil proses.
- "Pemisah ribuan" merupakan objek berjenis `JCheckBox` yang digunakan sebagai *flag* atau tanda apakah kita akan menggunakan pemisah ribuan atau tidak. Untuk deretan angka yang cukup banyak, penggunaan pemisah ribuan akan sangat membantu kita membaca data yang tersaji.
- "Hapus" merupakan sebuah objek berjenis `JButton` yang digunakan untuk menghapus data yang sedang terlihat di layar satu karakter demi satu karakter dari belakang.
- "Batal" merupakan sebuah objek berjenis `JButton` yang digunakan untuk membatalkan data yang kali dimasukkan terakhir. Jika button ini di-*click* pada saat kita memasukkan data kedua, maka data kedua akan dibatalkan sedangkan data pertama tetap dianggap valid.
- "Ulang" merupakan sebuah objek berjenis `JButton` yang digunakan untuk membatalkan seluruh data yang telah dimasukkan. Dalam bagan automata di atas respon dari button ini sama dengan tindakan "cancel".
- "Exit" merupakan sebuah objek berjenis `JButton` yang digunakan untuk mengakhiri jalannya program. Window akan ditutup dan kendali dikembalikan ke sistem operasi.

- ".", "0" sampai "9" merupakan objek berjenis JButton yang digunakan untuk memasukkan data numerik dan tanda pemisah desimal pada bilangan pecahan.
- "^", "*", "/", "+ dan "-" merupakan objek berjenis JButton yang digunakan untuk menentukan jenis operator yang akan digunakan pada proses matematika.
- "=" merupakan objek berjenis JButton yang digunakan untuk memerintahkan komputer melakukan proses sesuai dengan data-data yang diberikan, lalu menampilkan hasilnya di "Area tampilan".
- "Area informasi" merupakan objek berjenis JLabel yang kita gunakan untuk menampilkan informasi biasa, misalnya nama kita.

C. TUGAS PENDAHULUAN

1. Pelajari kelas – kelas di Java yang berhubungan dengan User Interfaces. Kemudian buatlah daftar kelasnya dan fungsi kelas tersebut.

D. PERCOBAAN

Percobaan 1: Implementasi Kalkulator Mini

```

1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4
5  class MiniCalculator extends JFrame
6                          implements ActionListener
7  {
8      private JPanel panelAtas = new JPanel();
9      private JPanel panelTengah = new JPanel();
10     private JPanel panelBawah = new JPanel();
11
12     private JTextField tampilan = new JTextField ("");
13     private JCheckBox pemisahRibuan =
14         new JCheckBox ("Pemisah ribuan", true);
15
16     private Button[] tombolFungsi =
17         { new Button ("Hapus"), new Button ("Batal"),
18           new Button ("Ulang"), new Button ("^"),
19           new Button ("7"),     new Button ("8"),
20           new Button ("9"),     new Button ("*"),

```

```

19         new Button ("4"),      new Button ("5"),
20         new Button ("6"),      new Button ("/"),
21         new Button ("1"),      new Button ("2"),
22         new Button ("3"),      new Button ("-"),
23         new Button ("0"),      new Button ("."),
24         new Button ("="),      new Button ("+") };
25
26     private JLabel informasi =
27         new JLabel (" R.Purnama, 2003");
28
29     private char operator = ' ';
30     private double dataPertama = 0.0;
31     private double dataKedua = 0.0;
32     private double dataHasil = 0.0;
33     private boolean awalKetikan = true;
34     private boolean entryPertama = true;
35     private boolean entryDesimal = false;
36
37     //-----
38     // Constructor class
39     //-----
40     public MiniCalculator()
41     {
42         super ("Kalkulator sederhana");
43
44         setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
45         setLocation (100,100);
46         setSize (300,200);
47         setResizable (false);
48
49         setPanelAtas();
50         setPanelTengah();
51         setPanelBawah();
52         resetNilai();
53
54         getContentPane().setLayout (new BorderLayout());
55         getContentPane().add (panelAtas,
56                               BorderLayout.NORTH);
57         getContentPane().add (panelTengah,
58                               BorderLayout.CENTER);
59         getContentPane().add (panelBawah,
60                               BorderLayout.SOUTH);
61
62         show();
63     }

```

```

62 //-----
63 // Mereset seluruh nilai
64 //-----
65 private void resetNilai()
66 {
67     operator = ' ';
68
69     dataPertama = 0.0;
70     dataKedua = 0.0;
71     dataHasil = 0.0;
72
73     awalKetikan = true;
74     entryPertama = true;
75     entryDesimal = false;
76 }
77
78 //-----
79 // Mengatur panel atas
80 //-----
81 private void setPanelAtas()
82 {
83     pemisahRibuan.setForeground (new Color (0,0,0));
84     pemisahRibuan.setFont (
85         new Font ("arial", Font.PLAIN, 11));
86
87     tampilan.setEnabled (false);
88     tampilan.setHorizontalAlignment (JTextField.RIGHT);
89     tampilan.setFont (new Font ("arial",Font.BOLD,15));
90
91     panelAtas.setLayout (new BorderLayout());
92     panelAtas.add (tampilan, BorderLayout.CENTER);
93     panelAtas.add (pemisahRibuan, BorderLayout.SOUTH);
94 }
95 //-----
96 // Mengatur panel tengah
97 //-----
98 private void setPanelTengah()
99 {
100     panelTengah.setLayout (new GridLayout (5,4));
101
102     for (int i=0; i<5*4; i++)
103     {
104         tombolFungsi[i].addActionListener (this);
105         tombolFungsi[i].setFont (
106             new Font ("arial", Font.BOLD, 11));

```

```

107         panelTengah.add (tombolFungsi[i]);
108     }
109 }
110
111 //-----
112 // Mengatur panel bawah
113 //-----
114 private void setPanelBawah()
115 {
116     tombolExit.addActionListener (this);
117
118     informasi.setFont (
119         new Font ("arial", Font.BOLD, 11));
120     tombolExit.setFont (
121         new Font ("arial", Font.BOLD, 11));
122
123     panelBawah.setLayout (new BorderLayout());
124     panelBawah.add (informasi, BorderLayout.WEST);
125     panelBawah.add (tombolExit, BorderLayout.EAST);
126 }
127
128 //-----
129 // Proses pemisahan data ribuan
130 //-----
131 private String pisahkan (StringBuffer data)
132 {
133     String temp = data.toString();
134
135     if (data.length() > 3)
136     {
137         temp = data.substring (data.length()-3);
138         data.delete (data.length()-3, data.length());
139
140         temp = pisahkan (data) + ',' + temp.toString();
141     }
142
143     return (temp);
144 }
145
146 //-----
147 // Pemisah ribuan
148 //-----
149 private String pisahkanRibuan (double data)
150 {
151     String string = Double.toString (data);
152     int titik = string.indexOf ('.');
153     String pecahan = string.substring (titik);

```

```

152     long bulat = new Double (dataHasil).longValue();
153
154     string = Long.toString (bulat);
155     string = pisahkan (new StringBuffer (string));
156
157     return (string + pecahan);
158 }
159
160 //-----
161 // Menghapus karakter terakhir
162 //-----
163 private void hapusKarakter()
164 {
165     if (tampilan.getText().length() > 0)
166     {
167         StringBuffer data =
168             new StringBuffer (tampilan.getText());
169         char terakhir = data.charAt (data.length()-1);
170
171         if (terakhir == '.')
172             entryDesimal = false;
173
174         data.deleteCharAt (data.length()-1);
175         tampilan.setText (data.toString());
176     }
177
178 //-----
179 // Membatalkan data terakhir
180 //-----
181 private void batalkanData()
182 {
183     if (entryPertama)
184         dataPertama = 0.0;
185     else
186         dataKedua = 0.0;
187
188     tampilan.setText ("");
189 }
190
191 //-----
192 // Mengupdate data angka
193 //-----
194 private void updateData (int index)
195 {
196     if (awalKetikan)
197         tampilan.setText ("");

```

```

198
199     String label = tombolFungsi[index].getLabel();
200     char karakter = label.charAt(0);
201     StringBuffer data =
                new StringBuffer (tampilan.getText());
202     tampilan.setText (data.toString() + karakter);
203
204     awalKetikan = false;
205 }
206
207 //-----
208 // Mengupdate data operator
209 //-----
210 private void updateOperator (int index)
211 {
212     if (entryPertama)
213     {
214         StringBuffer data =
                new StringBuffer (tampilan.getText());
215         dataPertama =
                Double.parseDouble (data.toString());
216     }
217
218     String label = tombolFungsi[index].getLabel();
219     operator = label.charAt(0);
220     entryPertama = false;
221     awalKetikan = true;
222 }
223
224 //-----
225 // Melakukan proses perhitungan
226 //-----
227 private void prosesPerhitungan()
228 {
229     StringBuffer data =
                new StringBuffer (tampilan.getText());
230     dataKedua = Double.parseDouble (data.toString());
231
232     switch (operator)
233     {
234         case '+' : dataHasil = dataPertama + dataKedua;
235                 break;
236         case '-' : dataHasil = dataPertama - dataKedua;
237                 break;
238         case '*' : dataHasil = dataPertama * dataKedua;
239                 break;
240         case '/' : dataHasil = dataPertama / dataKedua;

```



```

241         break;
242         case '^' : dataHasil = Math.pow (dataPertama,
                                           dataKedua);
243     }
244
245     if (pemisahRibuan.isSelected())
246         tampilan.setText (pisahkanRibuan (dataHasil));
247     else
248         tampilan.setText (Double.toString (dataHasil));
249
250     entryPertama = true;
251     awalKetikan = true;
252 }
253
254 //-----
255 // Menambah tanda desimal
256 //-----
257 private void tambahTandaDesimal()
258 {
259     if (!entryDesimal && !awalKetikan)
260     {
261         entryDesimal = true;
262
263         StringBuffer data =
264             new StringBuffer (tampilan.getText());
265         tampilan.setText (data.toString() + '.');
266     }
267
268 //-----
269 // Action terhadap respon user
270 //-----
271 public void actionPerformed (ActionEvent event)
272 {
273     Object objek = event.getSource();
274     int lokasi = 0;
275
276     if (objek == tombolExit)
277         System.exit (0);
278     else
279     {
280         for ( ; lokasi<20; lokasi++)
281             if (objek == tombolFungsi[lokasi])
282                 break;
283
284         switch (lokasi)
285         {

```

```

286         case 0 : hapusKarakter();
287             break;
288         case 1 : batalkanData();
289             break;
290         case 2 : resetNilai();
291             tampilan.setText ("");
292             break;
293         case 16 : ;
294         case 12 : ;
295         case 13 : ;
296         case 14 : ;
297         case 8 : ;
298         case 9 : ;
299         case 10 : ;
300         case 4 : ;
301         case 5 : ;
302         case 6 : updateData (lokasi);
303             break;
304         case 3 : ;
305         case 7 : ;
306         case 11 : ;
307         case 15 : ;
308         case 19 : updateOperator (lokasi);
309             break;
310         case 18 : prosesPerhitungan();
311             break;
312         case 17 : tambahTandaDesimal();
313             break;
314     }
315 }
316 }
317 }
318
319 //-----
320 // Class utama
321 //-----
322 public class CobaCalculator
323 {
324     public static void main (String[] args)
325     {
326         new MiniCalculator();
327     }
328 }

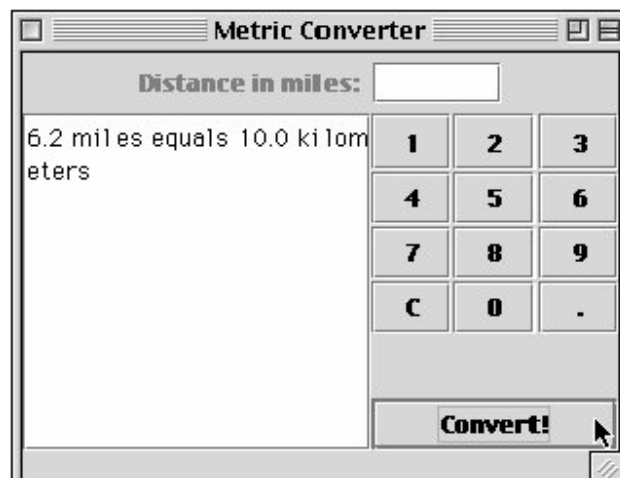
```

E. LATIHAN

Latihan 1: Modifikasi project kalkulator

Modifikasi kalkulator mini tersebut, sehingga dapat melakukan perhitungan saintifik lain, seperti trigonometri (sin, cos, dll), perpangkatan, logarirmik.

Latihan 2: Buatlah sebuah aplikasi converter untuk mengubah jarak dalam satuan mil ke km seperti pada gambar berikut



Latihan 3: Buatlah sebuah aplikasi yang menampilkan pilihan checkbox dan radiobutton seperti pada gambar



Latihan 4: Modifikasi latihan 2 sehingga aplikasi dapat mengubah jarak dari km ke mil.

Petunjuk : Berikan tambahan pilihan konversi berupa radio button.

Latihan 5: Aplikasi konversi suhu Fahrenheit - Celcius.

Buatlah aplikasi GUI untuk mengkonversi input suhu dari Fahrenheit ke Celcius dan sebaliknya.

F. TUGAS

Buatlah sebuah game Memory. Dalam aplikasi ini diberikan grid kotak 2 dimensi yang berisi pasangan gambar atau string yang sesuai. Ketika mengklik kotak, isinya akan terbuka. Kemudian klik kotak yang lain. Jika isinya sama, maka kotak akan tetap terlihat isinya. Jika tidak akan tertutup kembali. Aplikasi dapat dimainkan ulang dengan posisi isi kotak yang teracak, tidak sama dengan sebelumnya.

G. LAPORAN RESMI

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