

Modul 9: antarmuka grafis (GUI) dalam window

Setelah mengikuti mata kuliah ini mahasiswa dapat membuat program *antarmuka grafis* (GUI) sederhana dalam lingkungan *window*

Pengantar :

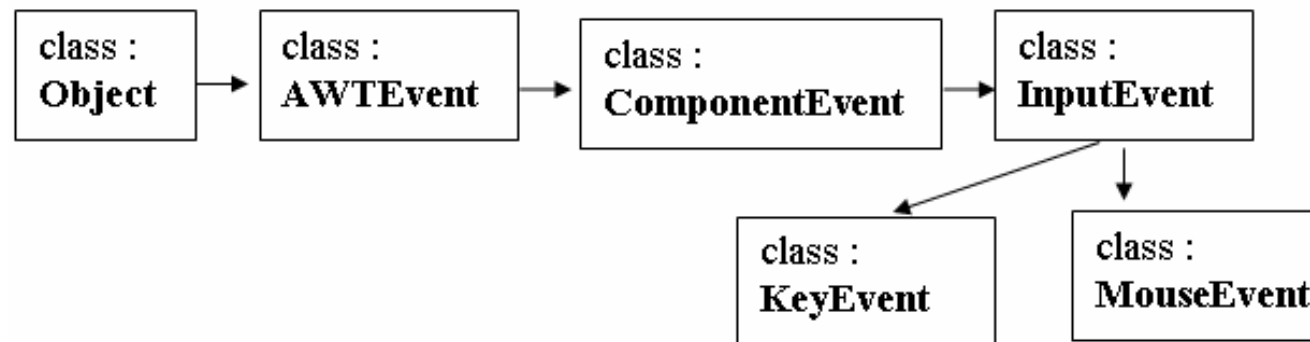
Dalam modul ini akan diuraikan beberapa topik bahasan yaitu :

1. Kendali mouse untuk windows
2. Pengertian antar muka grafis (Graphical User Interface =GUI)
3. Mengendalikan komponen GUI : *Button, Label, TextEdit, TextArea*

1. Kendali mouse dengan interface `MouseListener`

- Untuk mengendalikan mouse dapat digunakan interface **`MouseListener`**, **`MouseMoveListener`** dan **`MouseWheelListener`**
- Ketiga interface tersebut memiliki metod-metod yang berparameterkan **`MouseEvent`** yang memungkinkan kita mengetahui event apa yang kita lakukan pada mouse
- Struktur class dari `MouseEvent` adalah sbb

STRUKTUR CLASS UNTUK MENANGANI EVENT:



class MouseEvent dan interface MouseListener

Variabel dan method untuk Class MouseEvent

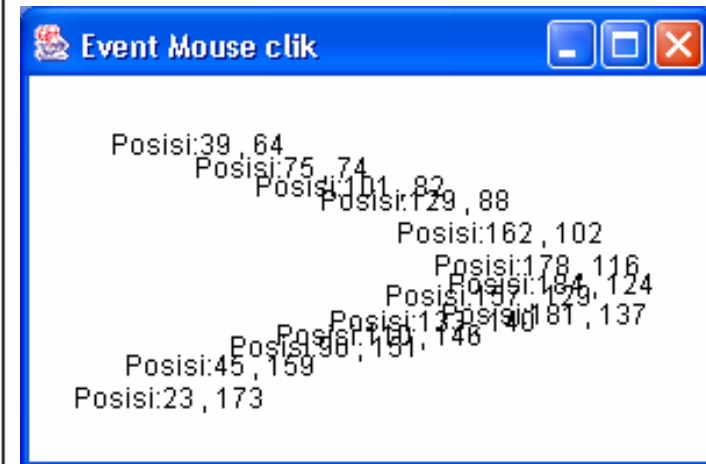
- Variabel : int → x,y, MOUSE_PRESSED, MOUSE_RELEASED, MOUSE_CLICKED, MOUSE_ENTERED, MOUSE_EXITED,
- MOUSE_MOVED, MOUSE_DRAGGED, MOUSE_WHEEL
- Method : getX(), getY(), getClickCount()

- **public interface MouseListener extends EventListener {**
- **public void mouseClicked(MouseEvent e);**
- **public void mousePressed(MouseEvent e);**
- **public void mouseReleased(MouseEvent e);**
- **public void mouseEntered(MouseEvent e);**
- **public void mouseExited(MouseEvent e);**
- **}**

Membuat efek mouseclick

```
import java.awt.*;
import java.awt.event.*;
class Mouse1 extends Frame implements MouseListener
{ public Mouse1()
  {setTitle("Event Mouse klik");
  addMouseListener(this);
  }
  public static void main(String[] args)
  { Frame f=new Mouse1();
  f.setSize(300,200);
  f.show();
  }
  public void mouseClicked(MouseEvent event) {
  Graphics g=getGraphics();
  int x=event.getX(); int y=event.getY();
  g.drawString("Posisi:" + x + " , " + y, x, y);}
  public void mouseReleased(MouseEvent event) {}
  public void mousePressed(MouseEvent event) {}
  public void mouseEntered(MouseEvent event) {}
  public void mouseExited(MouseEvent event) {}
}
```

Setelah di klik
beberapa kali tampilan
jadi



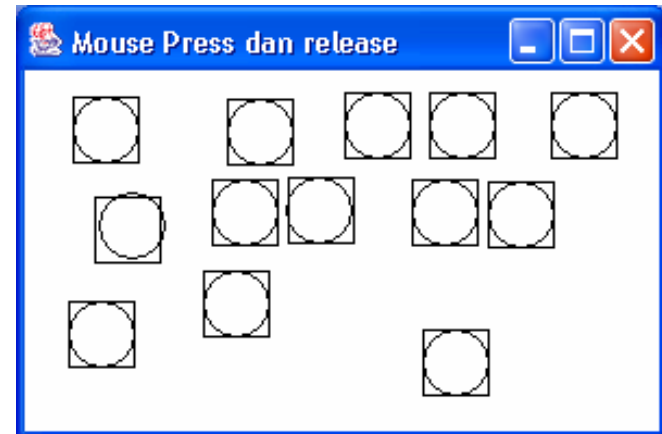
Efek yang lain dapat diberikan pada metod mousePressed,
mouseReleased dan lain-lain

mousePressed() dan mouseReleased()

```
import java.awt.*;
import java.awt.event.*;
class Mouse2 extends Frame implements MouseListener
{ public Mouse2()
  {setTitle("Mouse Press dan release");
  addMouseListener(this);
  }
  public static void main(String[] args)
  { Frame f=new Mouse2();
  f.setSize(300,200);
  f.show();
  }
  public void mouseClicked(MouseEvent event) {}
  public void mouseReleased(MouseEvent event) {
    Graphics g=getGraphics();
    int x=event.getX(); int y=event.getY();
    g.drawRect(x-15,y-15,30,30);}

  public void mousePressed(MouseEvent event) {
    Graphics g=getGraphics();
    int x=event.getX(); int y=event.getY();
    g.drawOval(x-15,y-15,30,30);}
  public void mouseEntered(MouseEvent event) {}
  public void mouseExited(MouseEvent event) {}
}
```

Hasil setelah tekan
lepas beberapa kali



Ketika mouse ditekan kita
menggambar lingkaran dan
ketika mouse dilepas kita
menggambar kotak

Mouse Gerak dan gulung

- Disamping aksi penekanan mouse, gerakan mouse dan roda mouse juga disediakan metod yang dapat diimplementasi jika event tersebut terjadi
- Metod tersebut ada dalam interface **MouseEventListener** dan **MouseWheelListener**, sbb :

MENUNGGU AKSI MOUSE GERAK:

- `public interface MouseEventListener extends EventListener {`
- `public void mouseDragged(MouseEvent e);`
- `public void mouseMoved(MouseEvent e);`
- `}`

MENUNGGU AKSI MOUSE GULUNG:

- `public interface MouseWheelListener extends EventListener {`
- `public void mouseWheelMoved(MouseWheelEvent e);`
- `}`

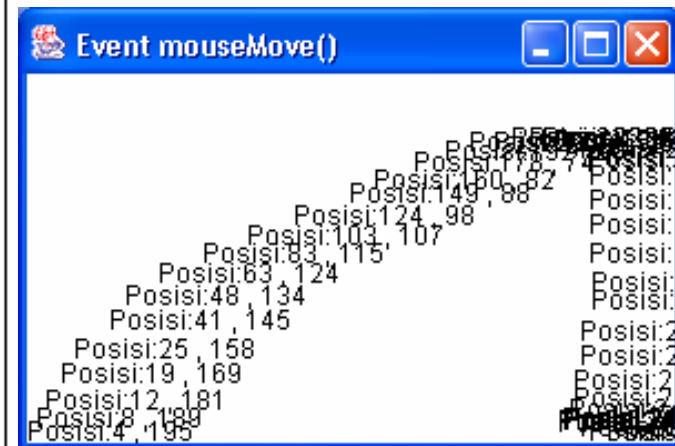
Memprogram mouse gerak dengan MouseMotionListener

```
import java.awt.*;
import java.awt.event.*;
class Mouse3 extends Frame implements MouseMotionListener
{ public Mouse3()
  {setTitle("Event mouseMove()");
   addMouseMotionListener(this);
  }

  public static void main(String[] args)
  { Frame f=new Mouse3();
    f.resize(300,200);
    f.show();
  }

  public void mouseDragged(MouseEvent e){}
  public void mouseMoved(MouseEvent e){
    Graphics g=getGraphics();
    int x=e.getX(); int y=e.getY();
    g.drawString("Posisi:"+x+" , "+y, x, y);
  }
}
```

Hasil ketika mouse
dilewatkan di atas frame



- Efek gerakan mouse dapat dibuat jauh lebih kompleks dari sekedar mencetak posisinya
- Efek dapat didefinisikan dalam gerak mouse atau mouse drag

2. Pengertian GUI

- GUI = Grafical User Interface adalah teknik pemrograman window dengan menggunakan komponen window sehingga user dapat berinteraksi dengan nyaman dengan window
- Seluruh komponen window seperti **Menu**, **Label**, **Button** dll adalah subclass dari class **Component**
- Teknis pembuatan GUI adalah sbb :
- **TAHAP MENDEFINISIKAN OBJEK :**
- Definisikan objek dari komponen yang akan ditaruh dalam frame pada saat frame dibuat (pada constructor frame yang kita definisikan)
- Letakkan objek dalam frame
- **TAHAP MENDEFINISIKAN AKSI :**
- Tahap ini didefinisikan tergantung dari jenis komponennya
- Pada komponen tombol aksi didefinisikan dengan cara mengimplementasi interface **ActionListener**
- Pada komponen lain seperti **Choice** dll ada cara tersendiri

3. Komponen utama GUI : Button, Label, TextField dan TextArea

Komponen **Button** :

- **Constructor**

Button() : mendefinisikan tombol tanpa label

Button(String lbl) : mendefinisikan tombol dengan label lbl

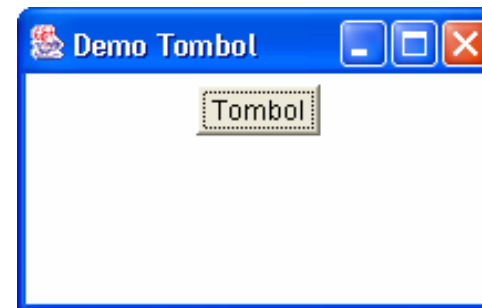
- **Method :**

setLabel(String lbl) : untuk mengganti label dari tombol

getLabel() : untuk mendapatkan label dari suatu tombol

```
import java.awt.*;
import java.awt.event.*;
class Komponen1 extends Frame
{ public Komponen1()
  {setTitle("Demo Tombol");
   setLayout(new FlowLayout());
   Button bt=new Button("Tombol");
   add(bt);
  }
  public static void main(String[] args)
  { Frame f=new Komponen1();
    f.setSize(300,200);
    f.setVisible(true) ;
  }
}
```

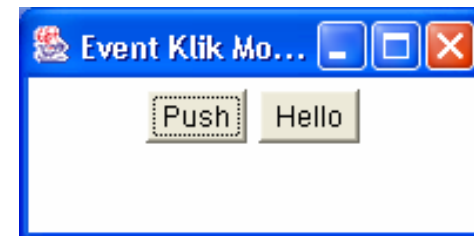
Tombol ini belum dapat beraksi



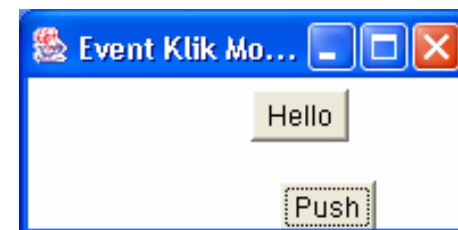
Contoh : Aksi Tombol 1 (jika tombol diklik pindah)

```
import java.awt.*;
import java.awt.event.*;
class push1 extends Frame implements ActionListener
{ private Button bt1, bt2;
  public push1()
  { setTitle("Event Klik Mouse");
    setLayout(new FlowLayout());
    bt1=new Button("Push");
    bt2=new Button("Hello");
    bt1.addActionListener(this);
    bt2.addActionListener(this);
    add(bt1);add(bt2);
  }
  public static void main(String[] args)
  { Frame f=new push1();
    f.setSize(200,100);
    f.show();
  }
  public void actionPerformed(ActionEvent e){
    String arg=e.getActionCommand();
    if(arg.equals("Push"))
    { Point loc= bt1.getLocation();
      bt1.setLocation(new Point(loc.x+15,loc.y+10));
    }
    else if(arg.equals("Hello"))
    { Point loc= bt2.getLocation();
      bt2.setLocation(loc.x-15,loc.y+10);
    }
  }
}
```

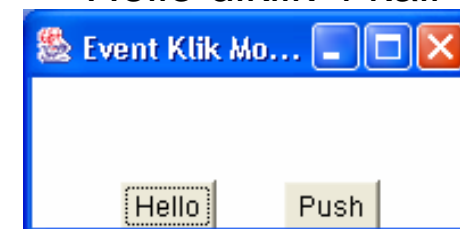
Posisi mula-mula



Push diklik 4 kali

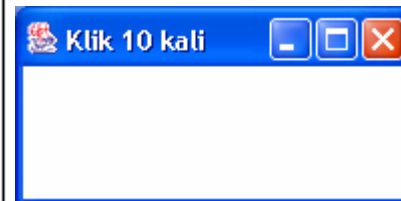


Hello diklik 4 kali



Contoh : Aksi Tombol 2 (jika tombol diklik label-nya berubah)

```
import java.awt.*;
import java.awt.event.*;
public class push3 extends Frame implements
    ActionListener
{
    public push3()
    {
        setTitle("Klik 10 kali");
        t1=new Button(" Hello ");
        t1.addActionListener(this);
        setLayout(new FlowLayout());
        add(t1);
    }
    public void actionPerformed(ActionEvent e)
    {
        pencacah++;
        if(pencacah>=10)System.exit(0);
        else{
            String plh=e.getActionCommand();
            kata=dafkata[(int)(Math.random()*5)];
            t1.setLabel(kata);
            if(kata.equals("STOP"))remove(t1);
        }
    }
    public static void main(String[] arg)
    {
        Frame f=new push3();
        f.setSize(200,100);
        f.setVisible(true);
    }
    Button t1; int pencacah=0; String kata="";
    String[] dafkata=
        {"ANI", "SATE", "STOP", "AKPRIND", "OK..."};
}
```



Jika tombol diklik maka bilangan random 0 s/d 1 dibangkitkan untuk mengambil kata acak yang dijadikan sbg label.

Jika kata kebetulan "STOP" tombol diremove.

Jika 10 x klik program ditutup

Komponen Label

Komponen Label :

- **Constructor**
Label (String lbl) : mendefinisikan Label dengan label lbl
- **Method :**
setText(String lbl) : untuk mengganti label dari Label
getText() : untuk mendapatkan label dari suatu Label

Contoh : akan dibuat program mengendalikan efek tombol untuk membesarkan dan mengecilkan tulisan pada Label
merubah ukuran label (besar, kecil) diatur dengan efek klik tombol

Mula-mula



klik besarkan



klik kecilkan



Programnya...

```
import java.awt.*; import java.awt.event.*;
public class TombolLabel extends Frame implements ActionListener
{ public TombolLabel()
  { setTitle("Efek Label");
    setLayout(new FlowLayout());
    bt1=new Button("Besarkan"); bt1.addActionListener(this);
    bt2=new Button("Kecilkan");bt2.addActionListener(this);
    bt3=new Button("Tutup");bt3.addActionListener(this);
    add(bt1); add(bt2); add(bt3);
    label=new Label(teks);
    label.setFont(new Font("Arial",Font.BOLD,ukuran));
    add(label);
  }
  public void actionPerformed(ActionEvent e)
  { String plh=e.getActionCommand();
    if(plh.equals("Besarkan") )
      {ukuran+=1; ubah(); }
      else if(plh.equals("Kecilkan") )
        {ukuran-=1;ubah();}
      else if(plh.equals("Tutup") )System.exit(0);
    }
  public static void main(String[] args)
  { Frame f=new TombolLabel();
    f.setSize(250,120); f.setVisible(true);
  }
  private void ubah()
  { label.setFont(new Font("Arial",Font.BOLD,ukuran)); }
  private Button bt1, bt2, bt3;
  private Label label; private int ukuran=14;
  private String teks="Hello Informatika";
}
```

TextField

Komponen **TextField** :

- **Constructor**

TextField (String txt) : mendefinisikan TextField berisi teks txt

TextField (int n) : mendefinisikan TextField selebar n kolom

- **Method :**

setText(String txt) : untuk mengganti teks pada TextField dg txt

getText() : untuk mendapatkan text dari suatu TextField

- setEchoChar(char) : menset char yang muncul jika diketik

- getEchoChar() : mendapatkan char yang muncul

Contoh :

- akan dibuat program membaca angka bulat dari keyboard dan mencari nilai faktorialnya
- Angka dibaca dengan TextField tf1, hasil ditampilkan dengan TextField tf2

Program faktorial dengan TextField

```
import java.awt.*; import java.awt.event.*;
public class TField1 extends Frame implements ActionListener
{ public TField1()
  { setTitle("Cari faktorial");
    setLayout(new FlowLayout());
    bt1=new Button("Hitung"); bt1.addActionListener(this);
    bt2=new Button("Hapus");bt2.addActionListener(this);
    bt3=new Button("Tutup");bt3.addActionListener(this);
    add(bt1);add(bt2);add(bt3);
    add(new Label("Masukkan n"));
    tf1=new TextField(6); add(tf1);
    add(new Label("Faktorial"));
    tf2=new TextField(15); add(tf2);  }
  public void actionPerformed(ActionEvent e)
  { String plh=e.getActionCommand();
    if(plh.equals("Hitung") )
      {long n =new Long(tf1.getText()).longValue();
        String x=new Long(faktorial(n)).toString();
        tf2.setText(x);}
    else if(plh.equals("Hapus") )
      {tf2.setText(""); tf1.setText(""); }
    else if(plh.equals("Tutup") )System.exit(0); }
  public static void main(String[] args)
  { Frame f=new TField1();
    f.setSize(220,170);f.show(); }
  private long faktorial(long n) {
    long f=1; for(long i=1;i<=n;i++)f*=i;
    return f;
  }
  private Button bt1, bt2, bt3;
  private TextField tf1, tf2;
}
```



TextArea

- **Constructor :**
TextArea(cacah baris , cacah kolom)
- **Method Penting :**
- **setText(String s) → untuk merubah isi TextArea**
- **getText() → untuk mengambil isi TextArea**
- **append(String s) → menambahkan s ke dalam TextArea**
- **insert(String s, int p) → insert string s dalam area posisi p**
- **replace(String s, int i, int j) → replace text s dalam area mulai i s/d j**
- **select(int awal , int akhir) → memilih (blok) seleksi text**
- **getSelectedText() → mengambil teks terseleksi oleh select()**

Contoh Efek Append, Insert dan Replace

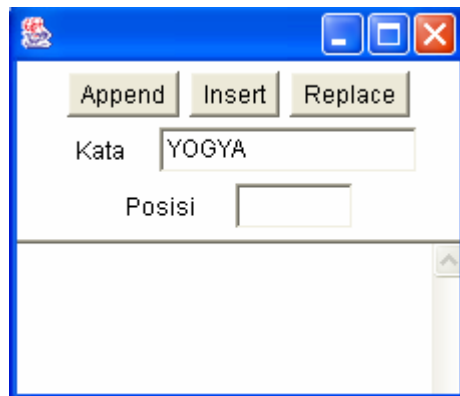
```
import java.awt.*;
import java.awt.event.*;
public class ApliTArea1 extends Frame implements ActionListener
{ public ApliTArea1()
  { //setTitle("TextArea ");
    setLayout(new BorderLayout());
    bt1=new Button("Append"); bt1.addActionListener(this);
    add(bt1);
    bt2=new Button("Insert"); bt2.addActionListener(this);
    add(bt2);
    bt3=new Button("Replace"); bt3.addActionListener(this);
    add(bt3);

    add(new Label(" Kata"));
    tf1=new TextField(15); add(tf1);
    add(new Label("Posisi "));
    tf2=new TextField(5); add(tf2);
    ta=new TextArea(6,30);
    add(ta);
  }
  public static void main(String[] args)
  { Frame f=new ApliTArea1();
    f.setSize(300,200);
    f.show();
  }
}
```

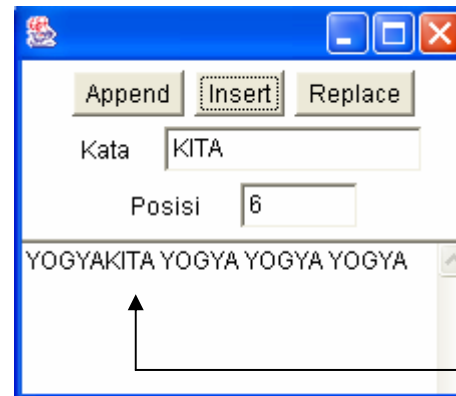
Lanjutnya...

```
public void actionPerformed(ActionEvent e)
{ String plh=e.getActionCommand();
  if(plh.equals("Append") )
    {String s=tf1.getText();
     ta.append(s+" ");
    }
  else if(plh.equals("Insert") )
    {String s=tf1.getText();
     String sp=tf2.getText();
     int posisi=new Integer(sp).intValue();
     ta.insert(s,posisi);
    }
  else if(plh.equals("Replace") )
    {String s=tf1.getText();
     int posisi=new Integer(tf2.getText()).intValue();
     ta.replaceRange(s,posisi,posisi+5);
    }
}
private Button bt1, bt2, bt3;
private TextField tf1, tf2;
private TextArea ta;
}
```

Hasil program

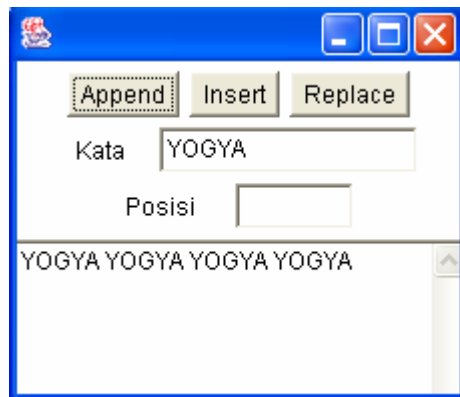


*Mula-
mula*

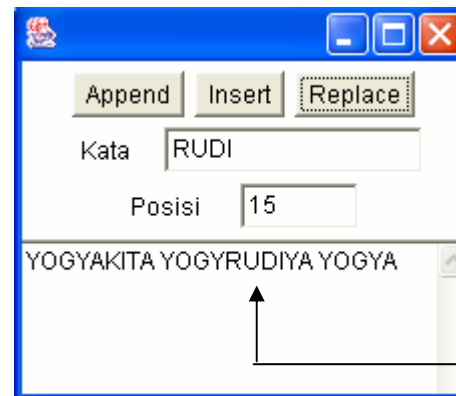


*Isi KITA
pada Kata
Dan 6 pada
posisi*

Klik Insert



*Tulis
YOGYA
dan klik
Append
4 kali*



*Isi Kata dg
RUDI*

*Dan posisi dg
15*

Klik Replace

Rangkuman

- Grafical User Interface (GUI) memfasilitasi programmer untuk merancang antar muka grafis
- Kendali GUI utamanya dilakukan dengan cara implementasi interface java yang sudah tersedia
- Pengendalian mouse, baik gerakan aksi tekan mouse atau gerak mouse dilakukan dengan mendefinisikan metod dibawah interface MouseListener dan MouseMotionListener
- Pengendalian tombol didefinisikan dengan mengimplementasi interface ActionListener yaitu mendefinisikan metod ActionPerformed
- Untuk menginput teks satu baris digunakan class TextField
- Untuk membaca teks banyak baris digunakan class TetArea

Latihan

1. Buatlah program dengan komponen antar muka grafis untuk membaca Panjang dan Lebar suatu empat persegi panjang dengan EditText dan menampilkan Luas dan Keliling. Layout sebagai berikut :
 - Kotak untuk memasukkan Panjang dan Lebar adalah EditText. Tombol HitungLuas dan HitungKell apabila diklik maka pada kotak luas dan kotak keliling akan muncul hasilnya apabila dalam kotak Panjang dan Lebar ada datanya. Bila tombol Clear di klik semua EditText dibersihkan.

PROGRAM HITUNG LUAS DAN KELILING		
PANJANG:	<input type="text"/>	
LEBAR:	<input type="text"/>	
LUAS :	<input type="text"/>	
KELILING:	<input type="text"/>	
<input type="button" value="HitungLuas"/>	<input type="button" value="HitungKell"/>	<input type="button" value="Clear"/>

Latihan

2. Buatlah program kamus Inggris-Indonesia dengan antar muka GUI sebagai berikut

KAMUS INGGRIS-INDONESIA

KATA INGGRIS

INDONESIANYA

Skenario:

Jika tombol CARI ditekan maka akan menampilkan kata indonesia dari suatu kata inggris yang diinput dikotak KATA INGGRIS, atau tulisan TIDAK DITEMUKAN jika kata tersebut tidak ada