

Modul 8: awt

Setelah mengikuti mata kuliah ini mahasiswa dapat membuat program sederhana dalam lingkungan *window*

Pengantar :

- Dalam modul ini akan diuraikan beberapa topik bahasan yaitu :
 1. Mengenal package awt
 2. class *Frame* untuk membuat window
 3. Mengendalikan window dengan interface *WindowListener*
 4. class *Graphics* : method *drawstring*, *drawLine*, *drawRectangle*, *drawCircle*
 5. class *Image* untuk manipulasi foto

1. Mengenal package awt

- Package java.awt merupakan package yang berisi class-class yang dirancang untuk mendukung pemrograman dengan Windows
- Isi class tersebut antar lain Frame dengan silsilah class sbb:



class Component : graphical object that can be displayed and interact to user

variable : int → x,y, width,height

Boolean → visible, enable

String → name

Color : foreground, background , Font → font

Metod-metod penting class Component dan Container

Component :

**setX(x), setY(y), getX(), getY() , getHeigh(), getWidth()
paint(Graphics), update(Graphics), repaint()
getName() , setName(String s)
setEnabled(boolean) :JDK 1 enable() dan disable()
setVisible(boolean) : JDK 1 show() dan hide()
setFont(Font)
setLocation(int x, int y) : JDK 1 move (x,y) ; getLocation()
setSize(width,height) : JDK 1 resize(x,y) ; getSize()
dan lain-lain ada 300 method (170 halaman A4 !!)**

class : Container (object that can contain another component)

**method : add(Component); remove(Component) ; removeAll()
setLayout(LayoutManager)
paint(Graphics); repaint() ; update(Graphics)**

2. class Frame untuk membuat Window

- Pemrograman window dimulai dengan pembuatan objek Frame
- Dalam objek tersebut komponen-komponen window lain dilekatkan dan diberi kemampuan aksi

```
import java.awt.*;

class gr1 extends Frame
{ public gr1(){};

  public static void main(String[] arg)
  { Frame f=new gr1();
    f.setSize(300,200);
    f.setVisible(true);
  }
}
```

Hasil program



Hasil program berupa sebuah window tetapi belum dapat ditutup dan belum ada apapun. Untuk dapat menutup window tersebut harus didefinisikan mengimplementasi WindowListener

3. Kendali window dengan interface WindowListener

- Window yang dibuat membuat subclass dari Frame belum diberi kemampuan menutup
- Kemampuan window untuk menutup, dekecilkan dst ada pada implementasi interface WindowListener
- Dengan mengimplementasi salah satu dari 7 method tersebut efek event pada window dapat didefinisikan
- Misalnya user meng-klik ikon close pada window (klik ini
- Maka metod WindowClosing(.) akan aktif

```
public interface WindowListener extends EventListener {  
    public void windowOpened(WindowEvent e);  
    public void windowClosing(WindowEvent e);  
    public void windowClosed(WindowEvent e);  
    public void windowIconified(WindowEvent e);  
    public void windowDeiconified(WindowEvent e);  
    public void windowActivated(WindowEvent e);  
    public void windowDeactivated(WindowEvent e);  
}
```

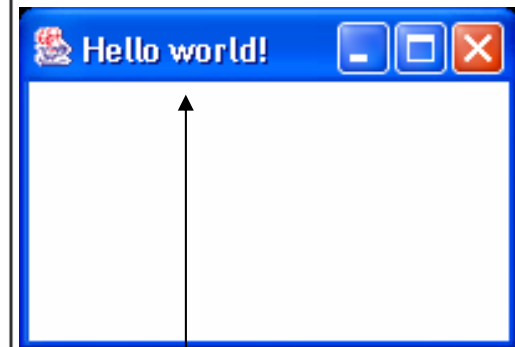


Agar window dapat ditutup maka tulis : `System.exit(0)` pada metod tsb

Kode programnya

```
import java.awt.*;
import java.awt.event.*;
class gr1 extends Frame implements WindowListener
{ public gr1(){
  setTitle("Hello world!");
  addWindowListener(this);
};

  public static void main(String[] arg)
  { Frame f=new gr1();
    f.setSize(300,200);
    f.setVisible(true);
  }
  public void windowOpened(WindowEvent e){};
  public void windowClosing(WindowEvent e){System.exit(0);};
  public void windowClosed(WindowEvent e){ };
  public void windowIconified(WindowEvent e){};
  public void windowDeiconified(WindowEvent e){};
  public void windowActivated(WindowEvent e){};
  public void windowDeactivated(WindowEvent e){};
}
```



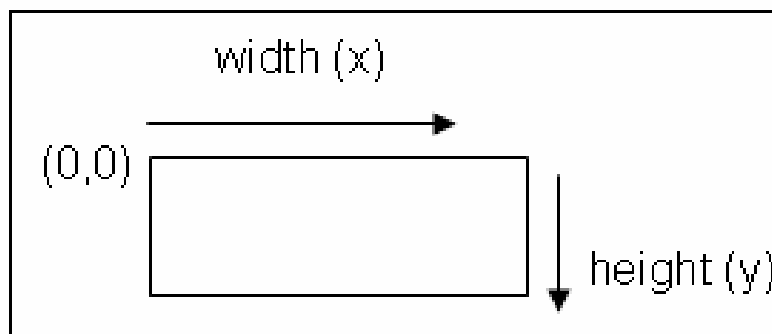
Dalam constructor gr1 terdapat dua statemen : setTitle() untuk **judul window** dan **addWindowListener(this)** untuk memberi window kemampuan ditutup

method paint(Graphics g)

- Metod paint() adalah metod yang otomatis akan dijalankan jika suatu objek Frame dibuat
- Jika dengan sengaja metod paint() akan dipanggil kembali pada tahapan lain maka cara yang dilakukan adalah memanggil metod repaint()
- Parameter metod paint() adalah objek dari class Graphics
- Class Graphics memiliki banyak metod yang berkaitan dengan operasi grafis seperti :
 - Membuat garis, kotak, lingkaran dll
 - Melukis String
 - Menempelkan foto
 - dll

Skala layar grafis

- Dalam pemrograman awt untuk menaruh setiap komponen dan menaruh grafis sangat diperlukan pengenalan koordinat layar grafis
- Koordinat layar grafis adalah :
 - x → nilai pixel kearah horizontal
 - y → nilai pixel grafis kearah vertikal
- Satuan posisi dan ukuran lebar dan tinggi komponen atau gambar adalah menggunakan ukuran pixel
- Perhatikan skema gambar berikut



4. Class Graphics

- Beberapa metod yang poenting dalam class Graohics adalah sebagai berikut :
 - `drawString(String s, x,y)` : cetak string dikoordinat x,y
 - `drawLine(x1,y1,x2,y2)`: cetak garis
 - `drawRect(x1,y1,width,height)`: cetak kotak
 - `fillRect(x1,y1,width,height)`: mengisikotak berisi
 - `drawOval(x1,y1,width,height)` : cetak lingkaran
 - `drawRoundRect(x1,y1,x2,y2)` : cetak kotak pojoknya bulat
 - `fillRoundRect(x1,y1,x2,y2)`: mengisi kotak
 - `drawPolygon(x1,y1,x2,y2)` : mencetak poligon
 - `drawArc(x,y,width,height,startangle,stopangle)`

Contoh drawString() dan drawRect()

```
import java.awt.*;
public class gr2 extends Frame
{
    public gr2()
    {setTitle("Kotak bertulisan");} //untuk judul windows
    public void paint(Graphics g)
    {g.drawRect(50,50,100,60);
    g.drawRect(48,48,104,64);
    g.setColor(Color.BLUE);
    g.drawString("HELLO MAS ",65,65);
    g.setColor(Color.RED);
    g.setFont(new Font("Times", Font.BOLD, 18));
    g.drawString("JAVA ",70,90);
    }
    public boolean handleEvent(Event evt)
    { if (evt.id == Event.WINDOW_DESTROY) System.exit(0);
    return false;
    }
    public static void main(String args[])
    { Frame f = new gr2();
    f.setSize(300,200);
    f.setVisible(true);
    }
}
```

Metod ini adalah gaya JDK1, pada JDK2 diganti dengan WindowListener



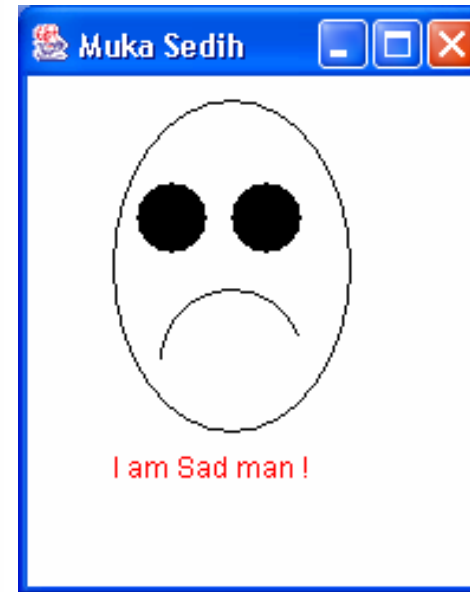
Metod setColor() digunakan untuk menset warna tulisan pada drawString(). Efek akan efektif sampai ditemukan setColor yang baru

drawOval() dan fillOval()

```
import java.awt.*;
public class gr3 extends Frame
{
    public gr3()
    {setTitle("Muka Sedih");}
    public void paint(Graphics g)
    {g.drawOval(40,40,100,140);
    g.fillOval(50,75,30,30);
    g.fillOval(90,75,30,30);
    g.drawArc(60,120,60,60,20,160);

    g.setColor(Color.RED);
    g.drawString("I am Sad man !",40,200);
    }
    public static void main(String args[])
    {
        Frame f = new gr3();
        f.setSize(200,250);
        f.setVisible(true);
    }
}
```

Hasil program



Untuk menghemat baris, Program diatas tidak dilengkapi fasilitas menutup window. Jika akan mengakhiri program tekan Ctrl-C dari command line

5. Class Image

- Salah satu class yang menarik dalam awt adalah class abstract Image yang memungkinkan kita untuk menampilkan dan memanipulasi data Image dalam berbagai format, seperti GIF, SPEG, BMP dan lain-lain
- class Image adalah class yang memfasilitasi kita untuk menaruh citra dalam frame yang kita buat
- Langkah menggambar citra pada frame adalah sbb :
- Mengambil gambar dari file dengan metod
`Toolkit.getDefaultToolkit().getImage(nmFILE)`
- Menempelkan citra kedalam frame dengan metod
`drawImage(objekgambar, x, y, skalaX, skalaY);`

Menaruh gambar tanpa skala

```
import java.awt.*;
import java.awt.Image.*;

public class Poto extends Frame
{
    String nmFILE="bb19.gif";
    Image gb=
    Toolkit.getDefaultToolkit().getImage(nmFILE);

    public void paint(Graphics g)
    { setTitle("My funny Bear ");
      g.setColor(Color.RED);
      g.drawString
        ("I am very Hungry..!! ", 40, 260);
      g.drawImage(gb, 40, 60, this);
    }

    public static void main(String args[])
    {
        Frame f = new Poto();
        f.setSize(200, 300);
        f.setVisible(true);
    }
}
```



Beberapa
hasil

Citra bb19.gif adalah citra animasi dalam format gif

Menaruh gambar dengan skala

```
import java.awt.*;
import java.awt.Image.*;

public class Poto2 extends Frame
{
    String nmFILE="wajah.jpeg";
    Image gb=
    Toolkit.getDefaultToolkit().getImage(nmFILE);

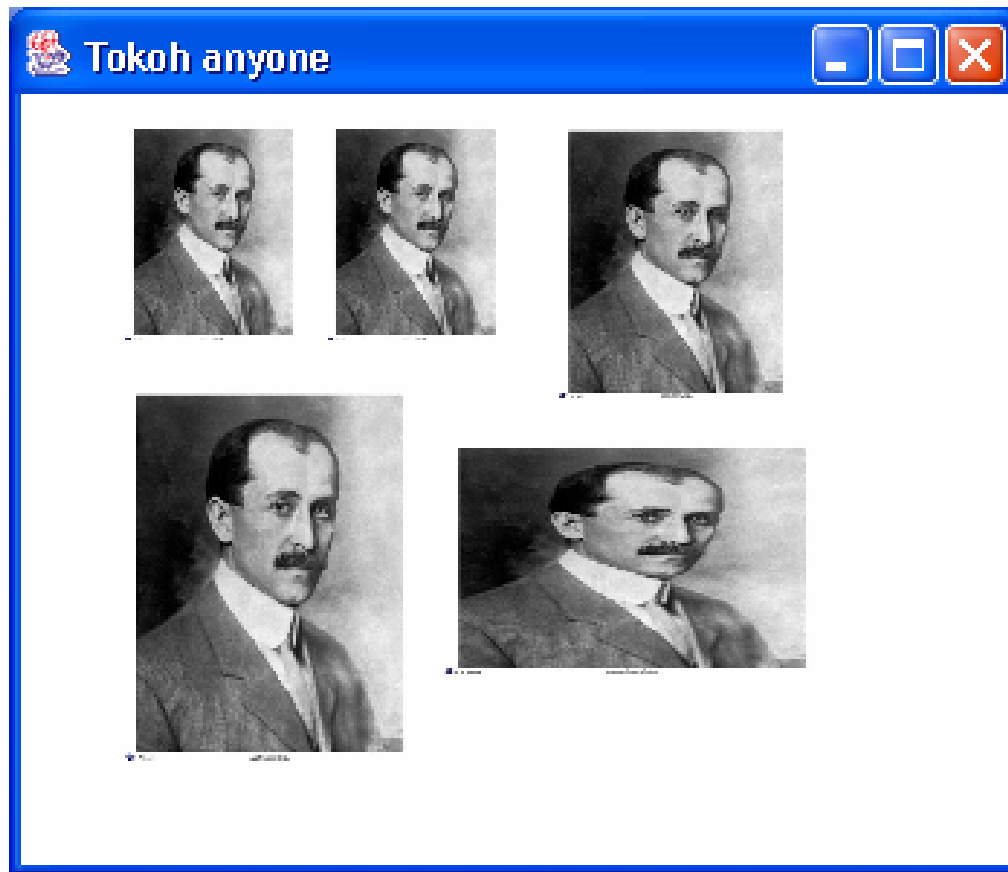
    public void paint(Graphics g)
    { setTitle("Tokoh anyone ");
      g.drawImage(gb,40,40,60,75,this);
      g.drawImage(gb,110,40,60,75,this);
      g.drawImage(gb,190,40,80,95,this);
      g.drawImage(gb,40,130,100,130,this);
      g.drawImage(gb,150,150,130,80,this);
    }

    public static void main(String args[])
    {
        Frame f = new Poto2();
        f.setSize(350,300);
        f.setVisible(true);
    }
}
```

- Citra wajah.jpeg digambarkan pada frame dengan skala yang berbeda-beda

- Skala akan merubah tampilan sesuai skala dan tidak sesuai dengan ukuran gambar asli

Hasil programnya ...



Metod `getGraphics()`

- Diluar metod `paint()` kita dapat mengakses frame dengan mengambil handelnya menggunakan metod `getGraphics()`
- Sintak yang digunakan adalah :
`Graphics g=getGraphics()`
- Selanjutnya objek `g` dapat digunakan untuk mengakses metod-metod grafis dalam class `Graphics`.
- Contoh :
- Dengan memanfaatkan metod dalam interface `WindowListener` kita akan membuat kendali grafis
- Jika windows sedang dibuka kita membuat tulisan dan kotak, lingkaran.
- Perintah tersebut kita letakkan dengan `getGraphics` pada metod `windowOpening()`

Tulisan dan gambar dibuat saat window dibuka

```
import java.awt.*;
import java.awt.event.*;
class gr4 extends Frame implements WindowListener
{ public gr4(){
  setTitle("Hello Otomatis!");
  addWindowListener(this);
};

  public static void main(String[] arg)
  { Frame f=new gr4();
    f.setSize(250,180);
    f.setVisible(true);
  }
  public void windowOpened(WindowEvent e)
  {Graphics g = getGraphics();
   g.setColor(Color.BLUE);
   g.fillOval(50,50,40,40);
   g.drawRect(50,50,40,40);
   g.drawString("Hello saya otomatis nongolll",50,110);};
  public void windowClosing(WindowEvent e){System.exit(0);};
  public void windowClosed(WindowEvent e){ };
  public void windowIconified(WindowEvent e){ };
  public void windowDeiconified(WindowEvent e){ };
  public void windowActivated(WindowEvent e){};
  public void windowDeactivated(WindowEvent e){};
}
```

- Saat window dibuka tulisan dan gambar dibuat
- Saat icon close ditutup, window ditutup



Rangkuman

- Membuat objek grafis dimulai dengan mendefinisikan class yang mengextends Frame
- Untuk memberikan efek grafis , misalnya membuat garis , kotak lingkaran menempelkan citra pada frame digunakan metod paint() dan grafis
- Parameter objek grafis dalam pemanggilan paint()
- Metod paint akan otomatis dieksekusi saat objek frame dibuat
- Dalam membuat objek gambar pada frame ukuran disesuaikan ukuran gambar
- Jika diinginkan ukuran berubah dapat ditempuh dengan merubah skala penampilan gambar dalam atatement drawImage()
- drawImage() juga dapat digunakan menampilkan animasi yang diformat dalam format GIF

Latihan

1. Buatlah gambar seperti gambar berikut dengan MS-Paint
2. Simpanlah dengan nama Mobil.bmp
3. Buatlah program java untuk menampilkan gambar mobil.bmp tersebut dengan 3 macam ukuran
4. Cantumkan ukuran pixel gambar tersebut di bawah setiap gambar yang ditampilkan

