

Penggunaan Applet di Netbeans.

Sekilas cara membuat Java Applet (Manual)

Pembuatan aplikasi Java Applet dimulai dengan mengembangkan class Applet yang berada pada package `java.applet.*`. Terdapat dua langkah untuk membuat Java Applet:

1. Membuat sebuah class Applet.
2. Membuat dokumen html yang berasosiasi dengan class Applet yang telah dibuat pada langkah pertama.

Class Applet mendefinisikan tingkah laku dari applet yang menggambarkan program yang kita buat. Dokumen html yang berasosiasi dengan class Applet digunakan untuk menjalankan applet dengan ukuran tertentu pada web page.

Template untuk Applet

```
import java.applet.Applet;
import java.awt.*;

public class AppletTemplate extends Applet {
    // deklarasi variabel.

    public void init() {
        // inisialisasi variabel, image loading, dll.
    }

    public void paint(Graphics g) {
        // drawing operations.
    }
}
```

Template untuk HTML

Applet ditampilkan bersama dengan kode HTML sebagai berikut:

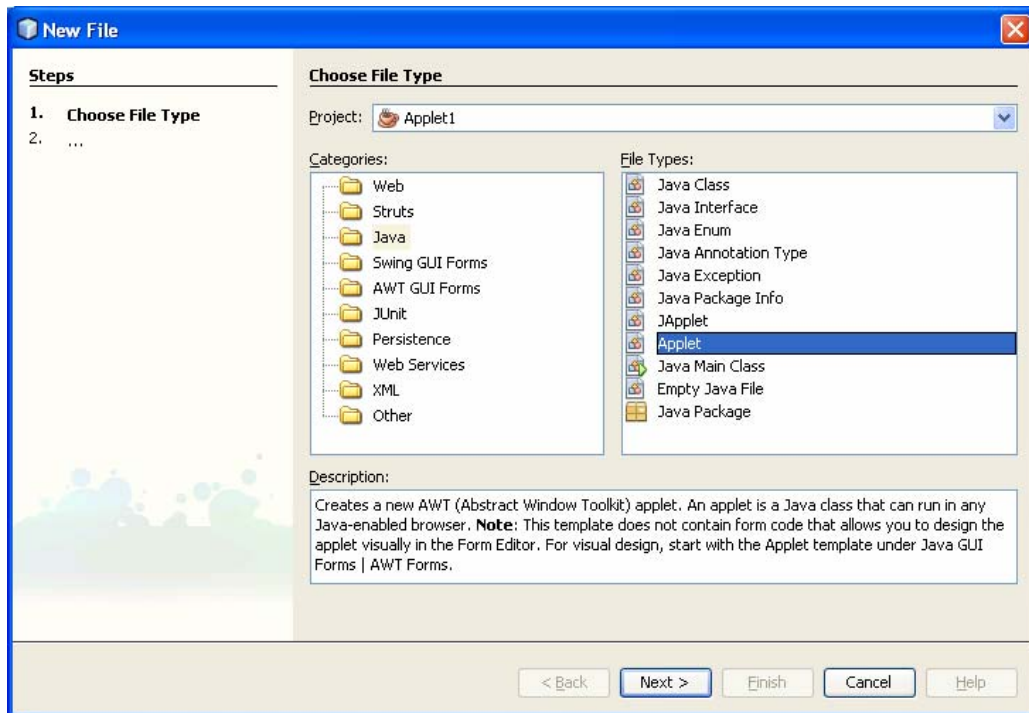
```
<HTML>
<HEAD>
    <TITLE> Template untuk loading applet</TITLE>
</HEAD>

<BODY>
<H1> Sebuah template untuk loading applet </H1>
<P>
<APPLET CODE="AppletTemplate.class" WIDTH=120 HEIGHT=60>
    <B> Error! Anda harus mengaktifkan Java-enabled browser.</B>
</APPLET>

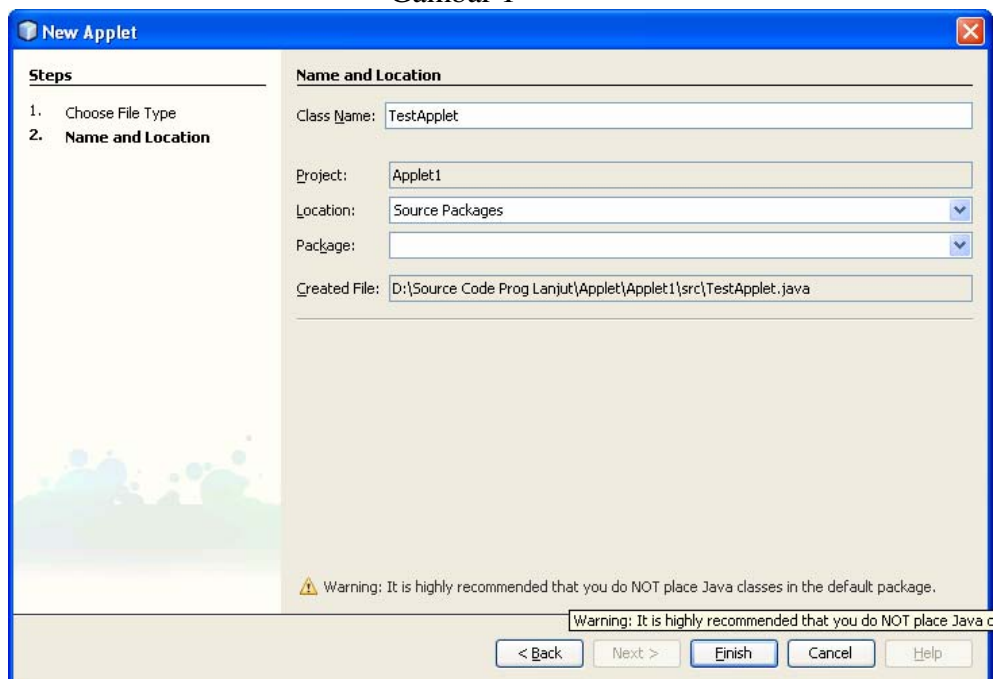
</BODY>
</HTML>
```

Cara Membuat Applet di Netbeans

Untuk membuat applet di netbeans, caranya mudah sekali. Buka File → New File → Pilih Categories : Java dengan file types : Applet / JApplet (gambar 1). Misalkan nama project adalah Applet1, beri nama file dengan TestApplet.java, file ini akan tersimpan dalam folder src (gambar 2). Isi dari file TestApplet.java seperti ditunjukkan gambar 3.



Gambar 1



Gambar 2

```
[-] /*
    * To change this template, choose Tools | Templates
    * and open the template in the editor.
    */

[-] import java.applet.Applet;

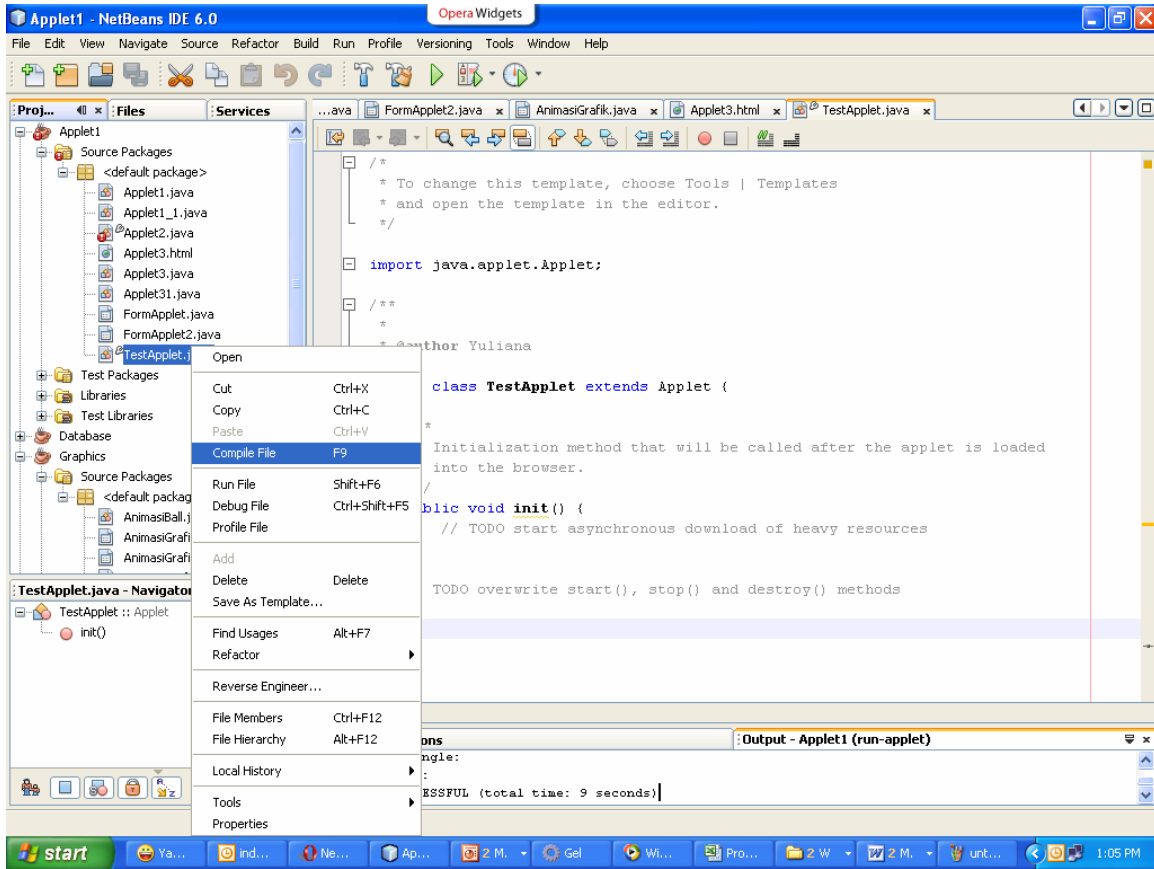
[-] /**
    *
    * @author Yuliana
    */
    public class TestApplet extends Applet {

        [-] /**
            * Initialization method that will be called after the applet is loaded
            * into the browser.
            */
            [-] public void init() {
                // TODO start asynchronous download of heavy resources
            }

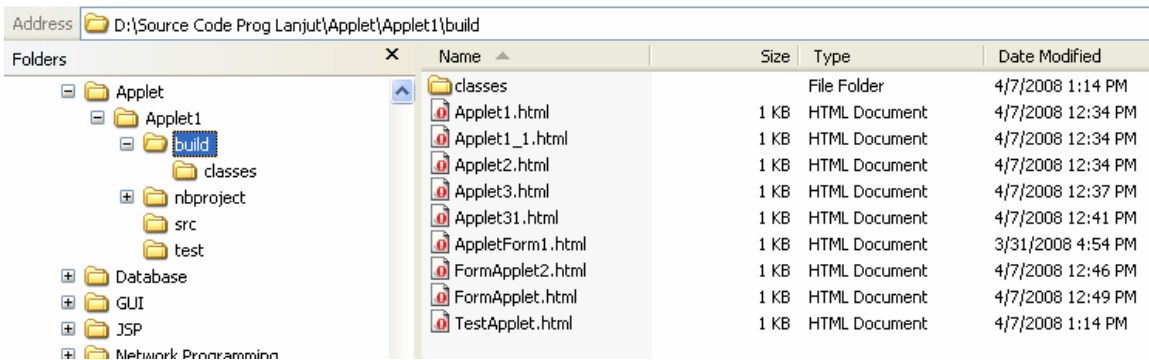
            // TODO overwrite start(), stop() and destroy() methods
        }
    }
```

Gambar 3

Untuk meng-compile klik kanan file TestApplet, pilih Compile File(gambar 4), maka terbentuk TestApplet.class, yang tersimpan pada folder Applet1/build/classes. Selanjutnya lakukan run file, maka akan terbentuk TestApplet.html yang tersimpan dalam folder folder Applet1/build (gambar 5).



Gambar 4



Gambar 5

Aplikasi 1 : Applet1.java

Kode program seperti ditunjukkan pada kode program 1. Class Applet1 dibuat dari class Applet, disamping itu juga bisa dibuat dari class Japplet. Program ini akan menggambar lingkaran berwarna merah pada dengan ujung kiri atas (50,50) dan diameter 50. Output ditunjukkan pada gambar 6.

```

public class Applet1 extends JApplet {

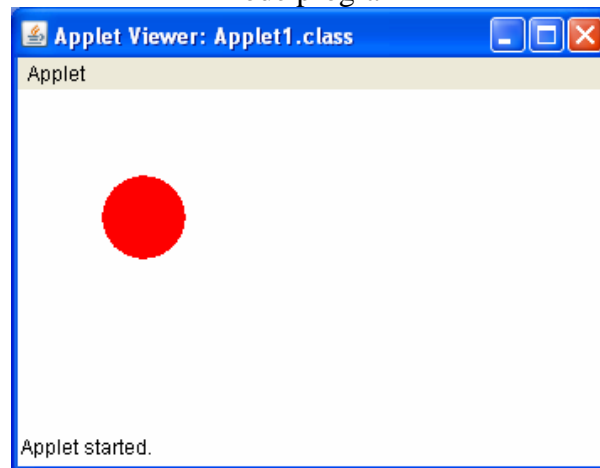
    public void init() {
        // TODO start asynchronous download of heavy resources
    }

    public void paint(Graphics g){
        g.setColor(Color.red);
        g.fillOval(50, 50, 50, 50);
    }

}

```

Kode program 1



Gambar 6

Aplikasi 2 : Applet2.java

Kembangkan dari aplikasi 1, dimana bola dapat bergerak. Supaya bola dapat bergerak, maka gunakan thread. Cara menggunakan thread dengan mengimplementasikan interface Runnable dan mendeklarasikan sebuah objek dengan tipe class Thread. Selanjutnya isi fungsi run(). Output ditunjukkan pada gambar 7.

```

public class Applet2 extends JApplet implements Runnable{
    private Thread runner ;
    private int x=0, y=0 ;

    public void init() {
        // TODO start asynchronous download of heavy resources
    }

    public void start(){
        if (runner==null){
            runner = new Thread(this);
            runner.start();
        }
    }

    public void stop(){
        if (runner != null){
            runner.stop();
            runner = null ;
        }
    }
}

```

```

public void run(){
    while(true){
        try{
            runner.sleep(250);
        }catch(Exception e){}
        if (x<=400)
            x = x+10 ;
        else
            x = 0 ;

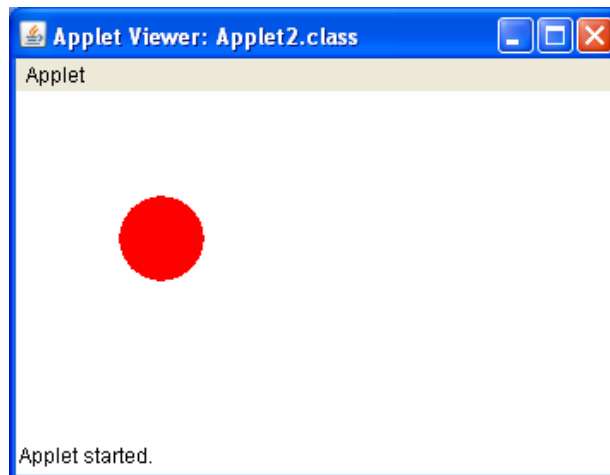
        if (y<=400) {
            y = y + 10 ;
        }
        else
            y = 0 ;

        repaint();
    }
}

public void paint(Graphics g){
    g.setColor(Color.white);
    g.fillRect(0, 0, 400, 400) ;
    g.setColor(Color.red);
    g.fillOval(x,y,50,50);
}
}

```

Kode program 2



Gambar 7

Aplikasi 3 : Applet3.java

Pada aplikasi 3 ini jika kita klik pada titik (x,y) maka akan tergambar lingkaran dengan warna yang dibangkitkan random dan diameter lingkaran yang dibangkitkan random pula. Buatlah class Applet3 dengan extends Applet, bukan JApplet. Program ini tidak berjalan jika extends Japplet (kalau menggunakan Japplet pada saat kita klik pada posisi (x,y) tidak muncul gambar lingkaran). Fungsi mouseDown() akan menangkap posisi x,y,

kemudian panggil fungsi repaint(). Fungsi repaint() ini untuk melakukan penggambaran ulang, fungsi ini akan memanggil fungsi update() dan selanjutnya memanggil fungsi paint(). Output ditunjukkan pada gambar 8.

```
public class Applet3 extends Applet {
    private int x = -100, y = -100 ;
    private int diameter ;

    public void init() {
    }

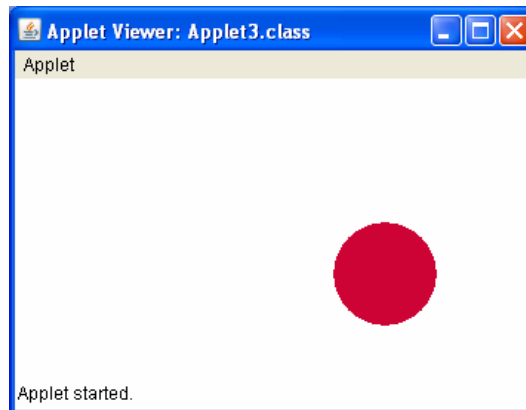
    public boolean mouseDown(Event e, int x, int y){
        this.x = x ;
        this.y = y ;
        repaint() ;
        return true ;
    }

    public void paint(Graphics g){

        int r1 = (int) (Math.random()*256) ;
        int g1 = (int) (Math.random()*256) ;
        int b1 = (int) (Math.random()*256) ;
        Color c = new Color(r1,g1,b1);

        diameter = (int) (Math.random()*100);
        g.setColor(c) ;
        g.fillOval(x-diameter/2, y-diameter/2, diameter, diameter);
    }
}
```

Kode program 3



Gambar 8

Bagaimana cara mengubah besar dari applet? Coba buka file Applet3.html yang berada di folder build ubah seperti kode program 4. Ubahlah

```
<APPLET codebase="classes" code="Applet3.class" width=350 height=200></APPLET>
```

Sehingga menjadi

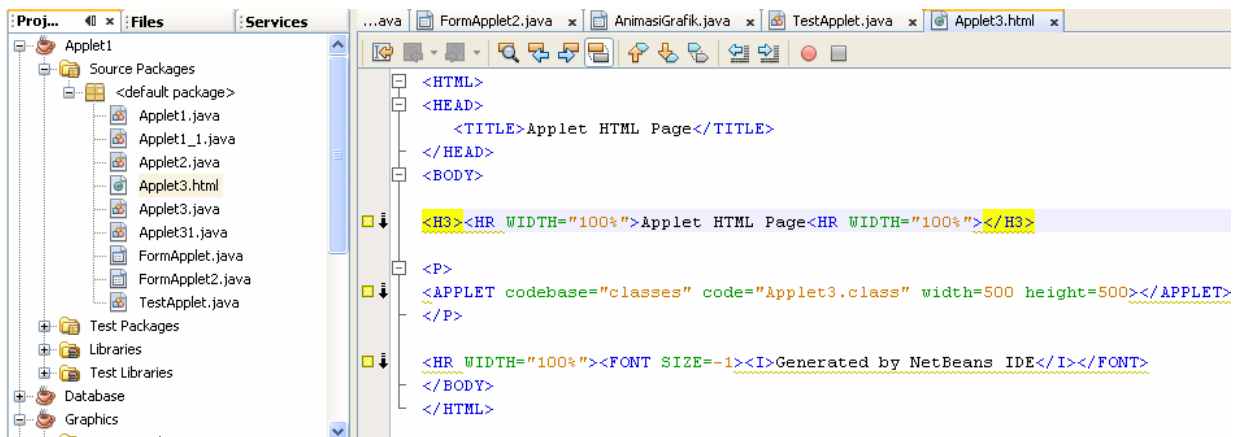
```
<APPLET codebase="classes" code="Applet3.class" width=400 height=400></APPLET>
```

```
<HTML>
<HEAD>
  <TITLE>Applet HTML Page</TITLE>
</HEAD>
<BODY>
<H3><HR WIDTH="100%">Applet HTML Page<HR WIDTH="100%"></H3>
<P>
<APPLET codebase="classes" code="Applet3.class" width=350
height=200></APPLET>
</P>
<HR WIDTH="100%"><FONT SIZE=-1><I>Generated by NetBeans
IDE</I></FONT>
</BODY>
</HTML>
```

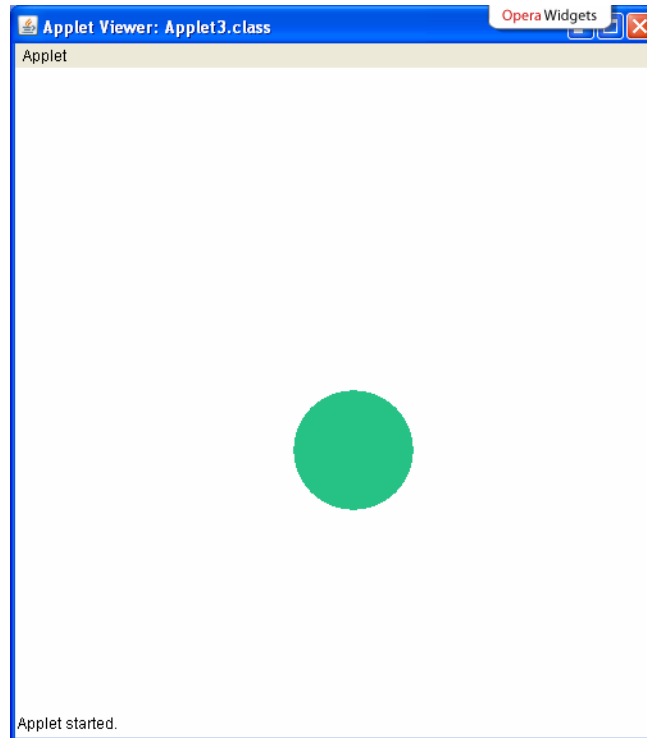
Kode program 4

Lalu compile dan run ulang, apa yang terjadi? ternyata besar applet akan kembali seperti semula yaitu width=350 height=200. Mengapa? Tiap kali kita run file maka Applet3.html akan di overwrite, sehingga jangan melakukan modifikasi file *.html di folder build. Ikuti langkah-langkah di bawah ini:

1. Copy Applet3.html dari folder build ke folder src. Nama file harus sama.
2. Lakukan pengeditan pada file Applet3.html yang berada pada folder src (Gambar 9). Selanjutnya build file Applet3.java, maka Applet3.html akan dikopikan dari folder src ke folder build. Hasil output setelah modifikasi ditunjukkan pada gambar 10.



Gambar 9



Gambar 10

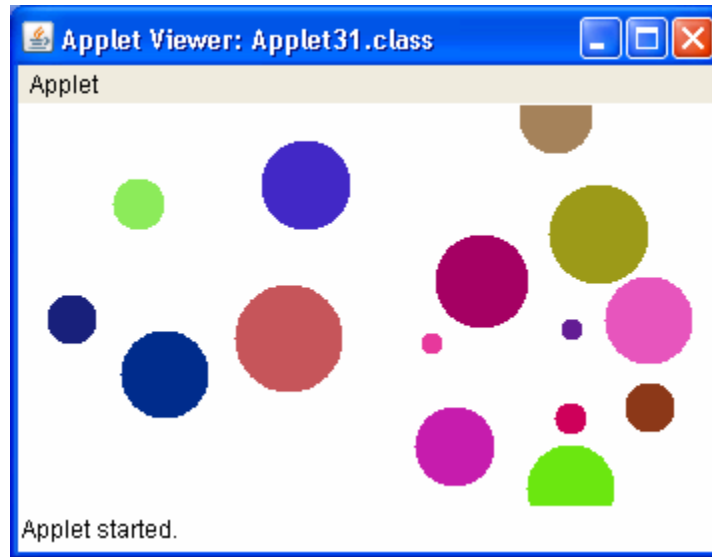
Aplikasi 3.1 :

Pada aplikasi 3.1 ini sama dengan aplikasi 3, pada saat user mengklik mouse pada posisi (x,y) maka akan tergambar lingkaran dengan warna yang dibangkitkan random dan diameter lingkaran yang dibangkitkan random pula. Kemudian kalau user mengklik lagi maka akan tergambar lingkaran tapi lingkaran yang sebelumnya tidak akan hilang.

Proses penyelesaian aplikasi 3.1 ini agak berbeda dengan sebelumnya yaitu menggunakan double-buffering.

Double-buffering merupakan teknik animasi yang membuat semua penggambaran dan operasi citra pada suatu citra dilakukan di memori. Saat penggambaran lengkap, citra yang lengkap dipindahkan ke layar dengan cepat sehingga pemakai melihat citra yang bergerak. Untuk melakukan penggambaran, mula-mula perlu dialokasikan memori untuk citra. Setelah itu, alokasikan obyek image dan graphics baru sebagai tempat melakukan penggambaran. Metode drawImage digunakan untuk menggambar Image pada konteks obyek Graphics. Metode drawImage akan bernilai true jika citra telah di load.

Output program ditunjukkan gambar 11. Kode program ditunjukkan pada kode program 5. Terdapat fungsi update(Graphics g) yang memanggil fungsi paint(g), untuk mencegah terlihatnya penghapusan latarbelakang.



Gambar 11

```

public class Applet31 extends Applet {

    private Image Buffer ;
    private Graphics gBuffer ;

    public void init() {
        Buffer = createImage(size().width,size().height) ;
        gBuffer = Buffer.getGraphics();
    }

    public boolean mouseDown(Event e, int x, int y){
        gambar(x,y);
        repaint() ;
        return true ;
    }

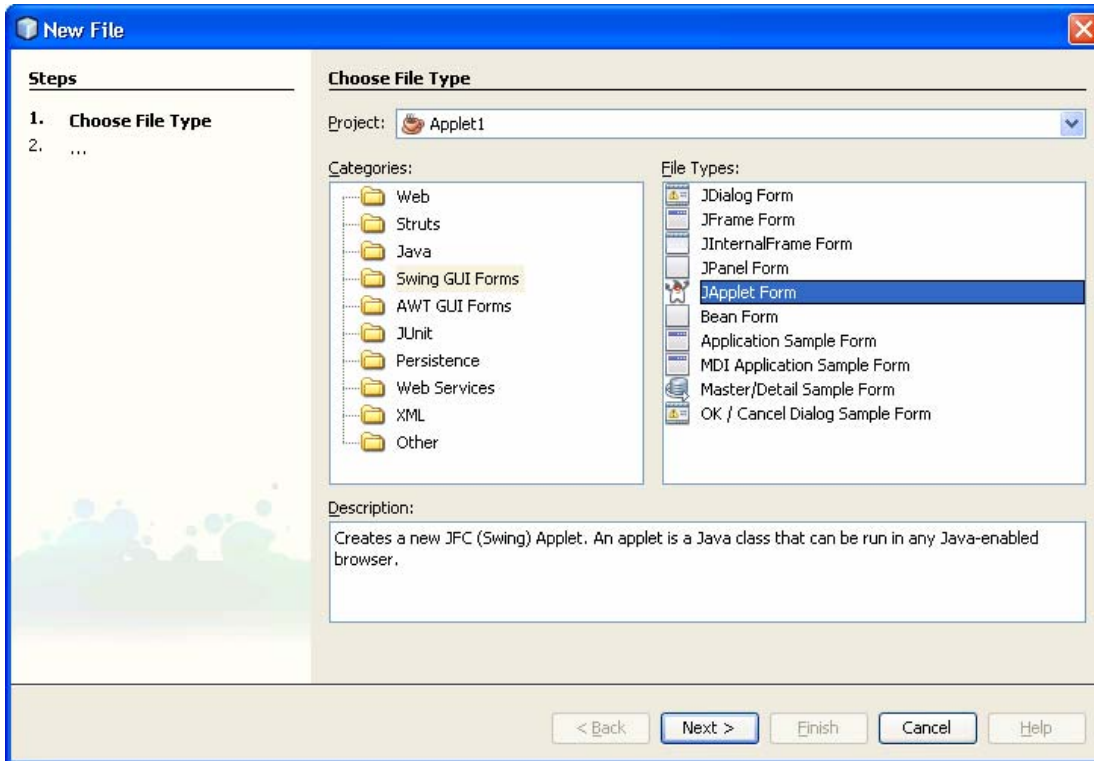
    public void gambar(int x, int y){
        int r1 = (int) (Math.random()*256) ;
        int g1 = (int) (Math.random()*256) ;
        int b1 = (int) (Math.random()*256) ;
        Color c = new Color(r1,g1,b1);
        int d = (int) (Math.random()*50 + 10);
        gBuffer.setColor(c);
        gBuffer.fillOval(x-25, y-25, d, d);
    }
    //mencegah terlihatnya penghapusan latarbelakang
    public void update(Graphics g){
        paint(g);
    }
    public void paint(Graphics g){
        g.drawImage(Buffer, 0, 0, this);
    }
}

```

Kode program 5

Membuat Form pada Applet

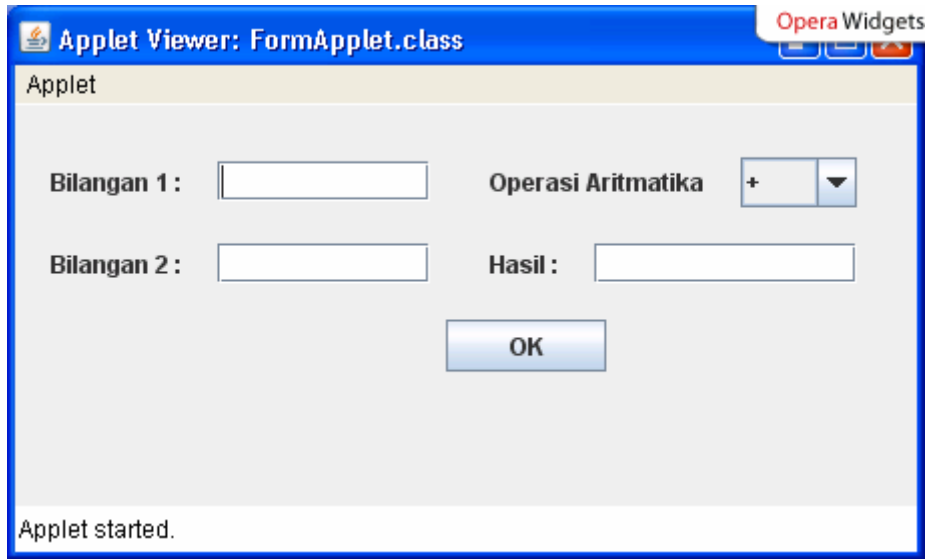
Untuk membuat form pada applet Buka File → New File → Pilih Categories : Swing GUI Forms atau AWT GUI Forms dengan file types : JApplet Form (gambar 12). Desain aplikasi seperti aplikasi 4.



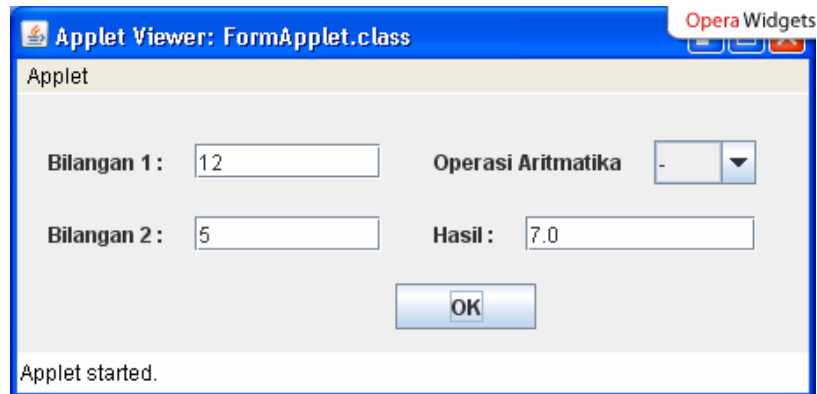
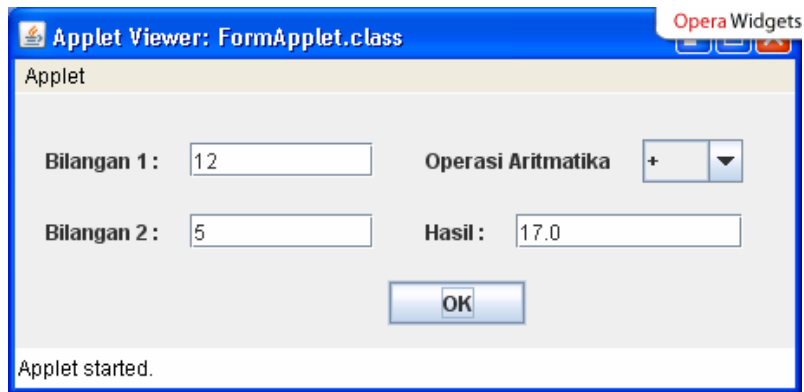
Gambar 12

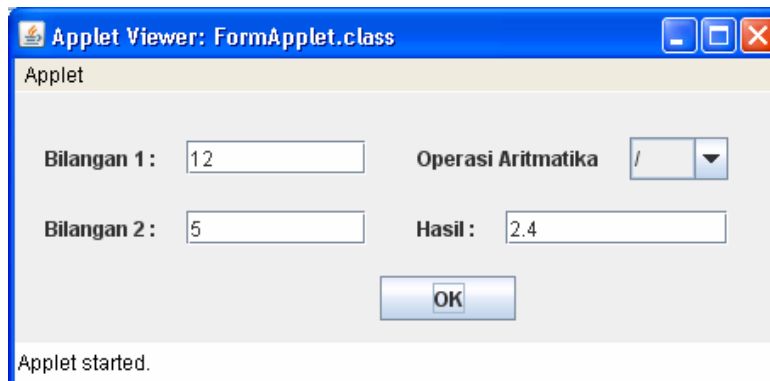
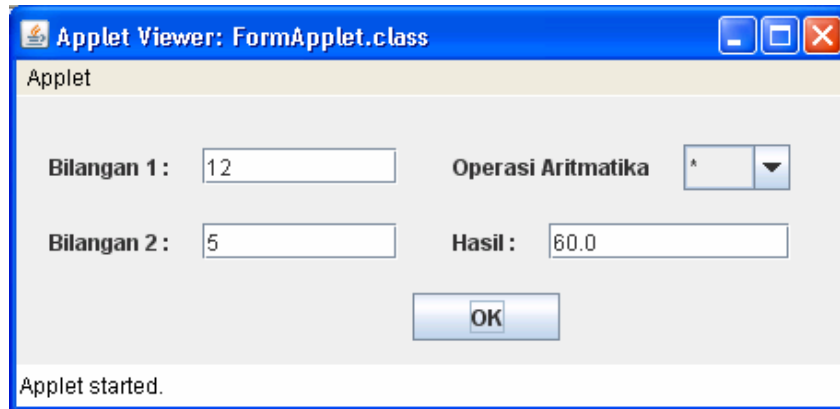
Aplikasi 4 : FormApplet.java

Pada aplikasi 4 kita masukkan 2 bilangan double, selanjutnya kita pilih dari comboBox, operasi aritmatika apa yang ingin kita lakukan. Apakah operasi penjumlahan, pengurangan, perkalian dan pembagian. Selanjutnya hasil operasi aritmatika akan ditampilkan pada textfield. Desain dari aplikasi ini ditunjukkan pada gambar 13. Output program ditunjukkan pada gambar 14.



Gambar 13





Gambar 14

Tambahkan kode program yang bercetak tebal pada program yang kalian buat.

```

public class FormApplet extends javax.swing.JApplet {

    /** Initializes the applet FormApplet */
    public void init() {
        try {
            java.awt.EventQueue.invokeLaterAndWait(new Runnable() {
                public void run() {
                    initComponents();
                }
            });
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    private void bOKActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        double bil1 = Double.parseDouble(tfBill1.getText());
        double bil2 = Double.parseDouble(tfBill2.getText());
        double hasil = 0 ;
        int op = cbOperator.getSelectedIndex();
        switch(op){
            case 0 : hasil = bil1 + bil2 ; break ;
            case 1 : hasil = bil1 - bil2 ; break ;
            case 2 : hasil = bil1 * bil2 ; break ;
            case 3 : hasil = bil1 / bil2 ; break ;
        }
        tfHasil.setText(hasil+"");
    }
}

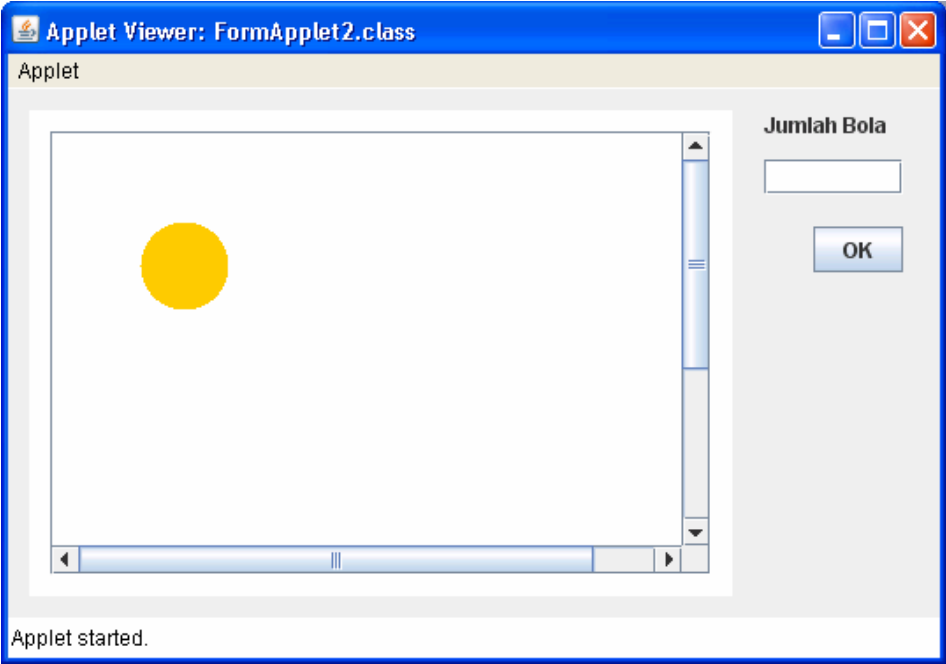
```

```
// Variables declaration - do not modify
private javax.swing.JButton bOK;
private javax.swing.JComboBox cbOperator;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JTextField tfBill1;
private javax.swing.JTextField tfBill2;
private javax.swing.JTextField tfH asil;
// End of variables declaration
}
```

Kode program 6

Aplikasi 5 : FormApplet.java

Pada aplikasi 5 ini kita akan menggabungkan grafik dengan GUI. Cara pengerjaannya hampir sama pada saat kita belajar membuat grafik (Lihat materi sebelumnya). Pada applet masukkan panel dan di atasnya beri scroll pane. Tambahkan label, textfield dan sebuah button (Hanya untuk tampilan saja, fungsi button kosong). Tampilan aplikasi seperti gambar 15. Program ditunjukkan pada kode program 7.



Gambar 15

Terdapat perbedaan antara aplikasi 3 dan aplikasi 5. Pada aplikasi 3, lingkaran digambar diatas applet, applet akan memanggil fungsi paint(). Tapi pada aplikasi 5, lingkaran digambar diatas panel. Caranya adalah buat class yang extends dari JPanel. Buat fungsi

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
}
```

```
public class FormApplet2 extends javax.swing.JApplet {
    private int nBola ;
    private Gambar gb ;
    private Dimension area;

    /** Initializes the applet FormApplet2 */
    public void init() {
        try {
            java.awt.EventQueue.invokeLaterAndWait(new Runnable() {
                public void run() {
                    initComponents();
                    area = new Dimension(400,400);
                    gb = new Gambar();
                    jScrollPane1.setViewportViewView(gb);
                    jPanel1.setPreferredSize(area);
                    gb.setBackground(Color.white);
                    gb.setPreferredSize(area);
                    gb.setAutoScrolls(true);
                }
            });
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    class Gambar extends JPanel{
        protected void paintComponent(Graphics g) {
            super.paintComponent(g);
            g.setColor(Color.ORANGE);
            g.fillOval(50, 50, 50, 50);
        }
    }

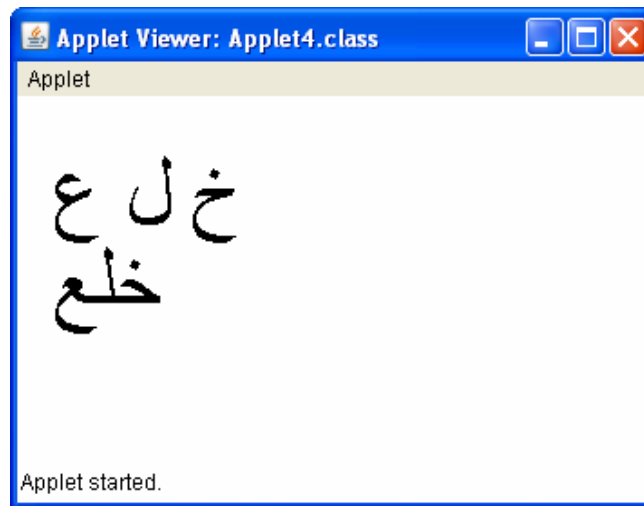
    // Variables declaration - do not modify
    private javax.swing.JButton bOK;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextField tfBola;
}
```

Kode program 7

Contoh Aplikasi Applet lainnya :

Contoh 1 :

Karakter di Java disebut dengan karakter Unicode, disimpan dalam 16 bit. Dibawah ini adalah karakter unicode \u062e \u0644 \u0639 dan \u062e\u0644\u0639 (gambar 16).



Gambar 16

```

public class Applet4 extends Applet {

    public void init() {
    }

    public void paint(Graphics g){
        Font f = new Font("Arial", Font.PLAIN,48);
        g.setFont(f);
        g.drawString("\u062e \u0644 \u0639",20,70);
        g.drawString("\u062e\u0644\u0639",20,120);
    }
}

```

Kode program 8

Contoh 2:

Kembangkan dari Applet3.java, ubahlah warna applet menjadi hitam (gambar 17). Untuk mengubah menjadi hitam tambahkan kode program seperti di bawah ini:

```

public void paint(Graphics g){
    g.setColor(Color.black);
    g.fillRect(0, 0, size().width, size().height);
    * * *
}

```

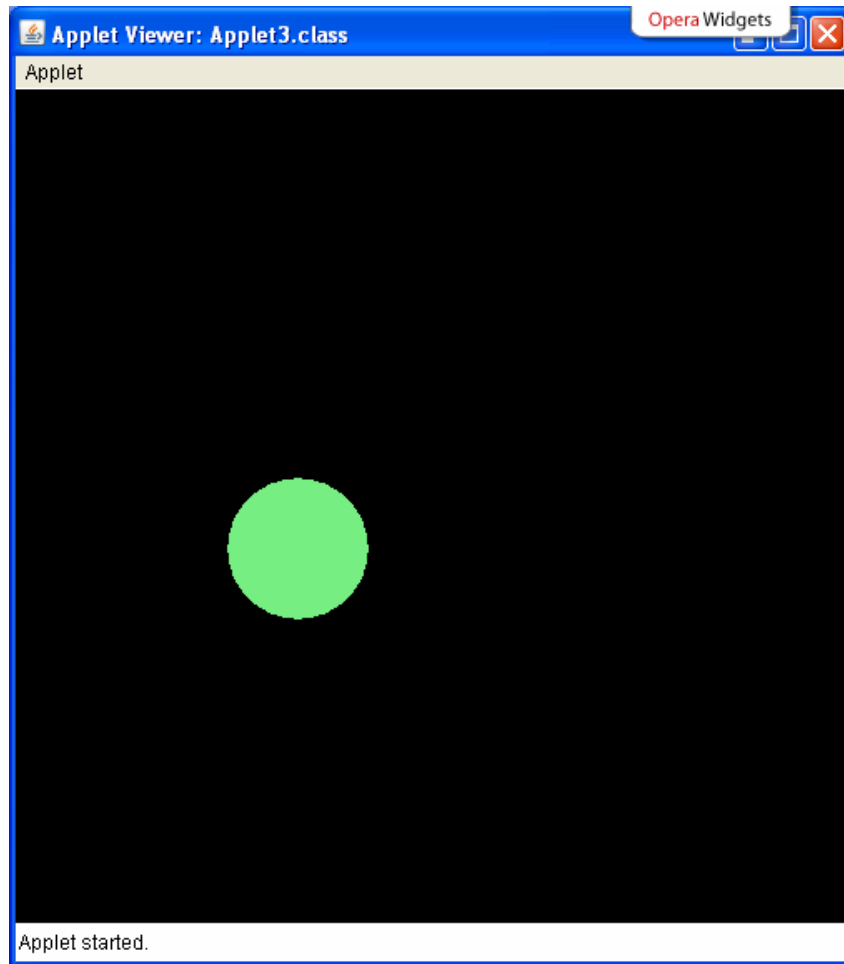
Apa yang terjadi jika program ini dijalankan? Pada saat user mengklik, maka tergambar lingkaran, klik lagi maka akan muncul lingkaran lagi. Tapi layar tampak kelihatan berkedip. Untuk mengatasi hal ini tambahkan

```

public void update(Graphics gr){
    paint(gr);
}

```

Tujuan penambahan fungsi ini untuk mencegah terlihatnya penghapusan latar belakang.



Gambar 17

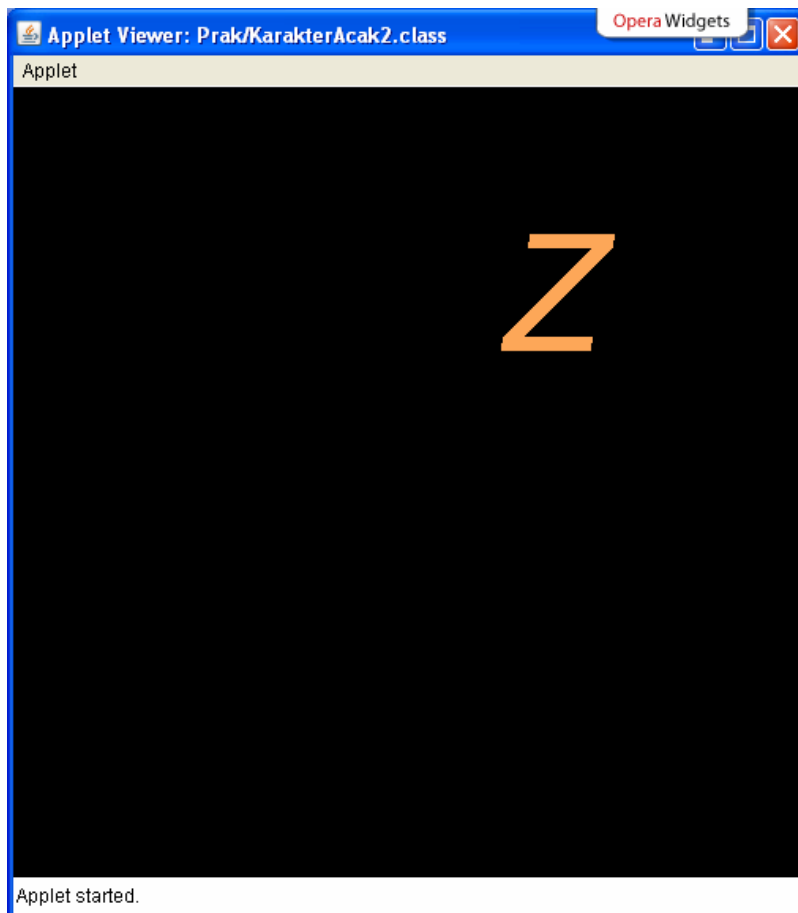
Tugas Praktikum :

Praktikum 1:

Buatlah aplikasi jika user mengklik applet, maka tergambar sebuah Huruf dengan warna yang dibangkitkan random. Dengan karakter, jenis huruf, besar huruf yang berbeda. Setiap huruf dapat berupa huruf biasa (Font.PLAIN), huruf miring (Font.ITALIC), huruf tebal (Font.BOLD). Jenis Huruf yang dibangkitkan "Arial", "Helvetica", "Times New Roman", "Courier". Cara membangkitkan sebuah huruf/Font adalah sebagai berikut:

```
g.setColor(Color.red);  
g.setFont(new Font("Arial",Font.BOLD,100));  
g.drawString("K", 50, 50);
```

Kode program diatas untuk membangkitkan huruf K, dengan warna merah. Huruf yang dibangkitkan dengan jenis huruf Arial, bercetak tebal dan besar huruf adalah 100.



Gambar 18

Praktikum 2:

Kembangkan dari tugas praktikum sebelumnya, jika user mengklik berulang kali pada applet, maka huruf yang sebelumnya tidak akan hilang (gambar 19). Buatlah class Huruf untuk menyimpan informasi dari huruf. Informasi yang disimpan adalah

- posisi (x,y)
- String
- Warna huruf
- Font yang dibangkitkan (jenis font, huruf biasa/miring/tebal, besar font).

```
public class Huruf {
    private String str ;
    private Color clr ;
    private Font font ;
    private int x;
    private int y ;

    public Huruf(){

    }

    public Huruf(int x, int y , String str, Color c, Font f1){
        this.x = x ;
        this.y = y ;
        this.str = str ;
    }
}
```

```

        this.clr = c ;
        this.font = f1 ;
    }

    public void setX(int x){
        this.x = x ;
    }

    public int getX(){
        return x ;
    }

    public void setY(int y){
        this.y = y ;
    }

    public int getY(){
        return y ;
    }

    public void setString(String str){
        this.str = str ;
    }

    public String getString(){
        return str ;
    }

    public void setColor(Color c){
        this.clr = c ;
    }

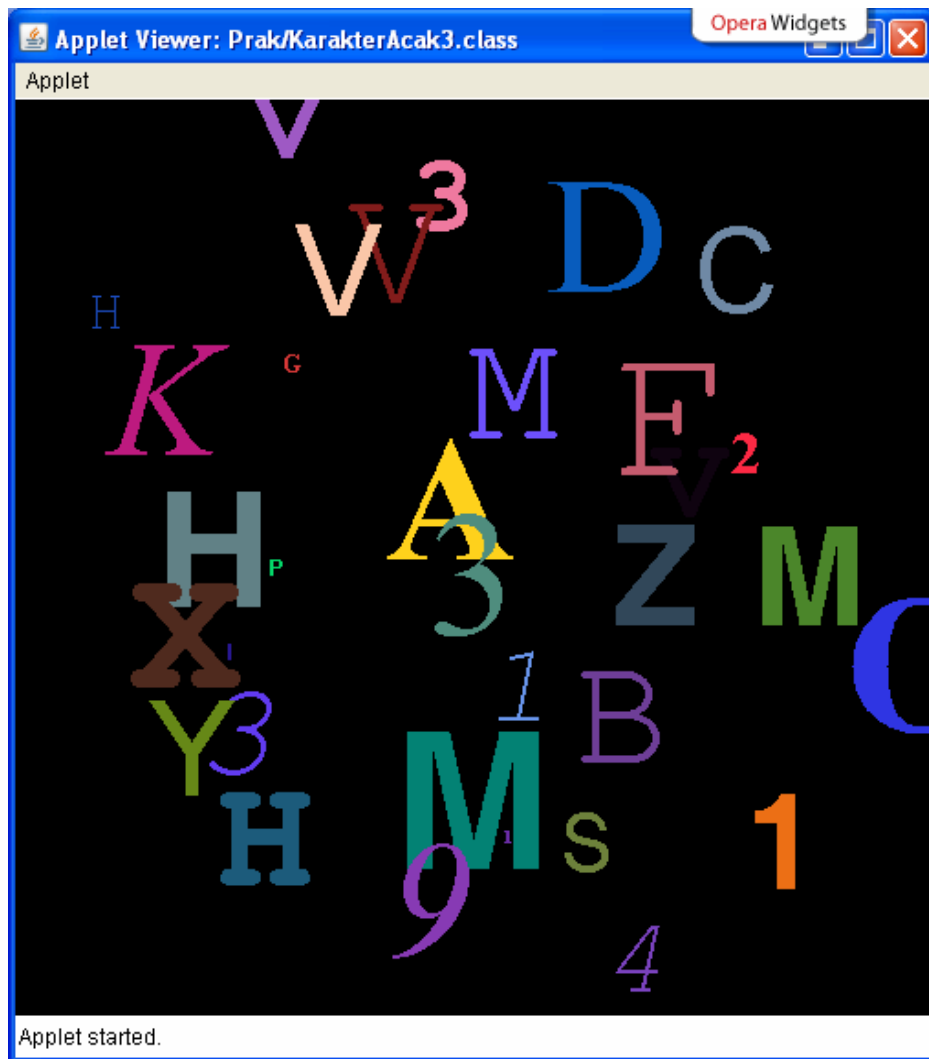
    public Color getColor(){return clr ;}

    public void setFont(Font f){
        this.font = f ;
    }

    public Font getFont(){return font ;}
}

```

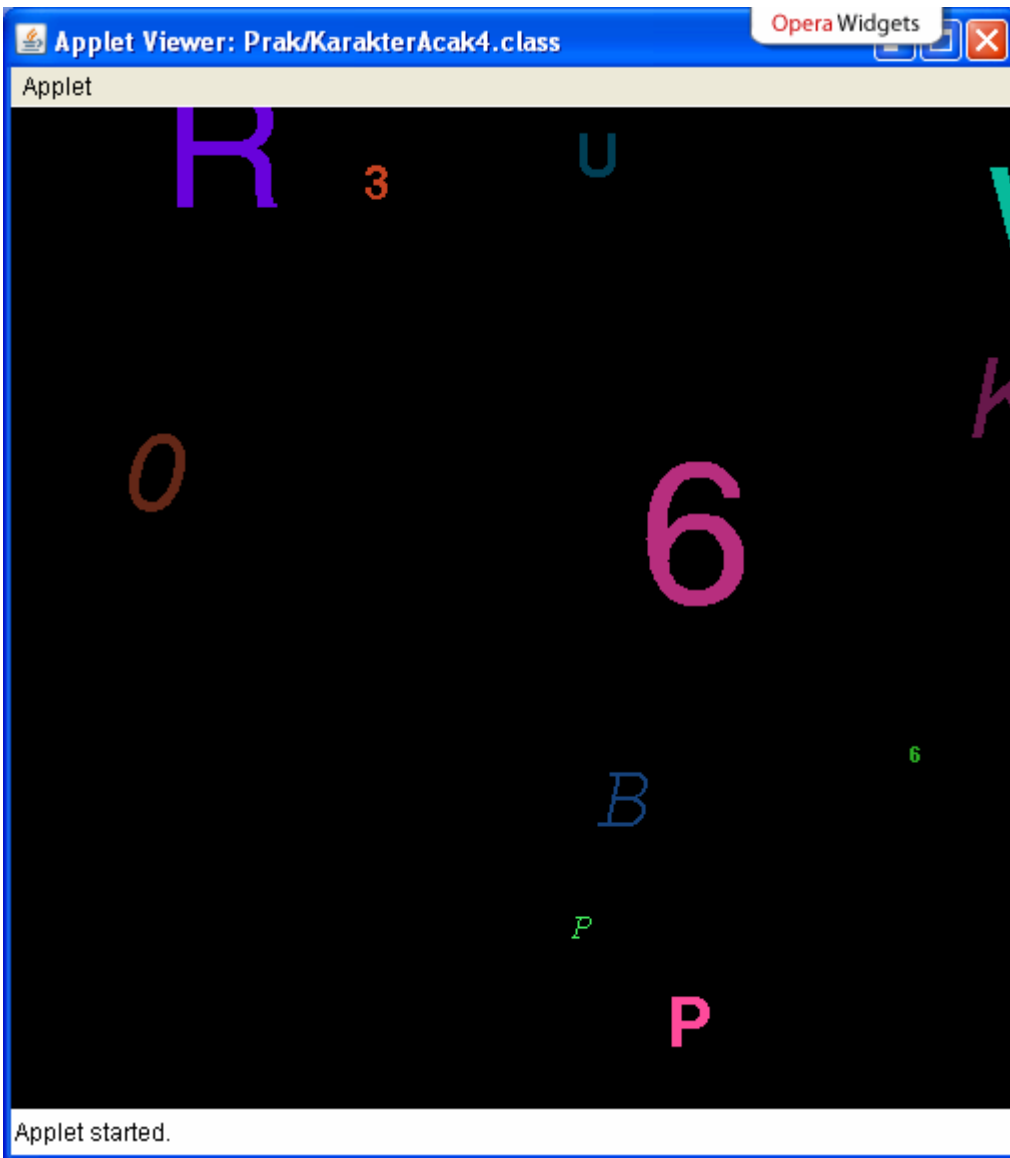
Kode program 9

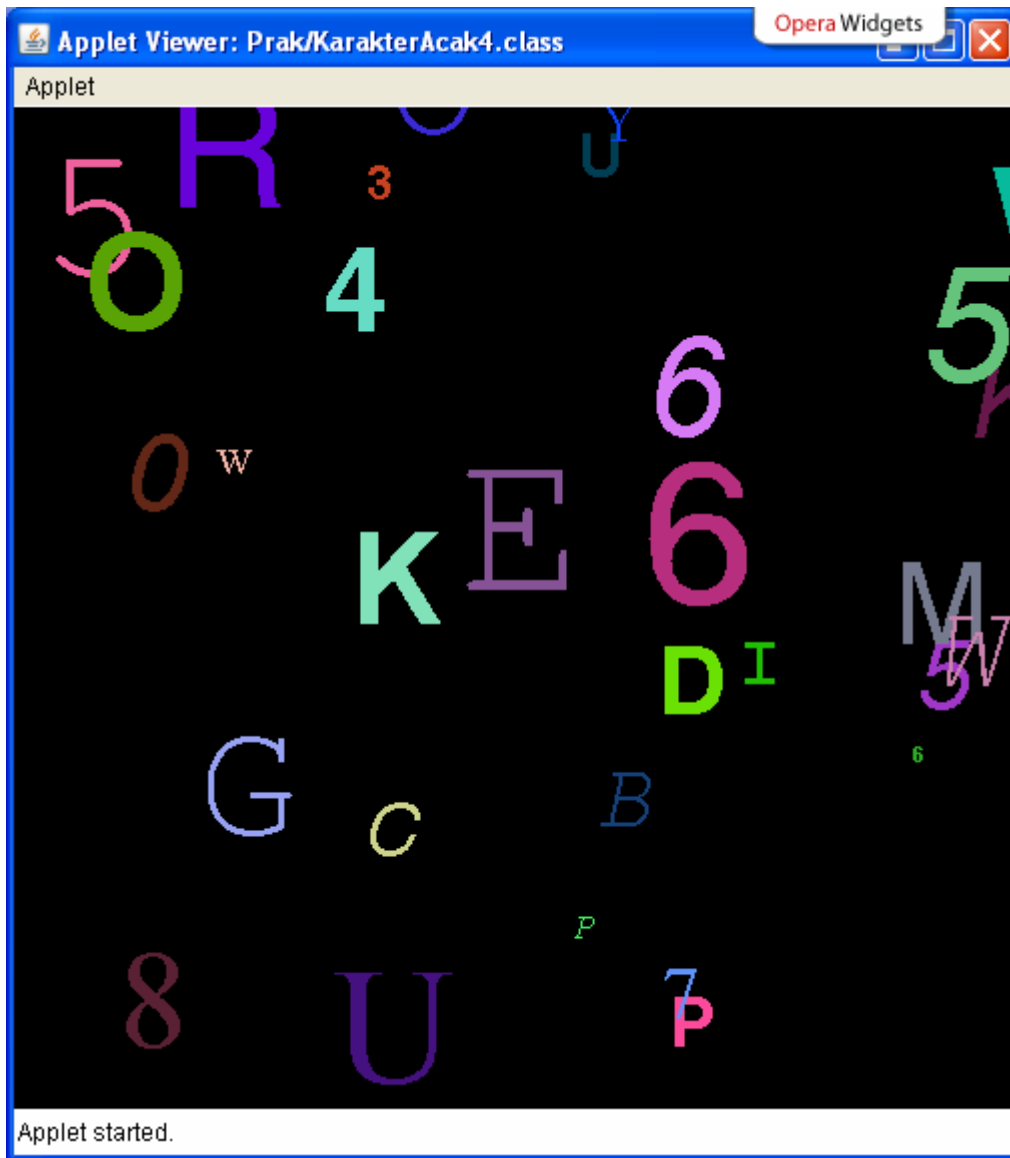


Gambar 19

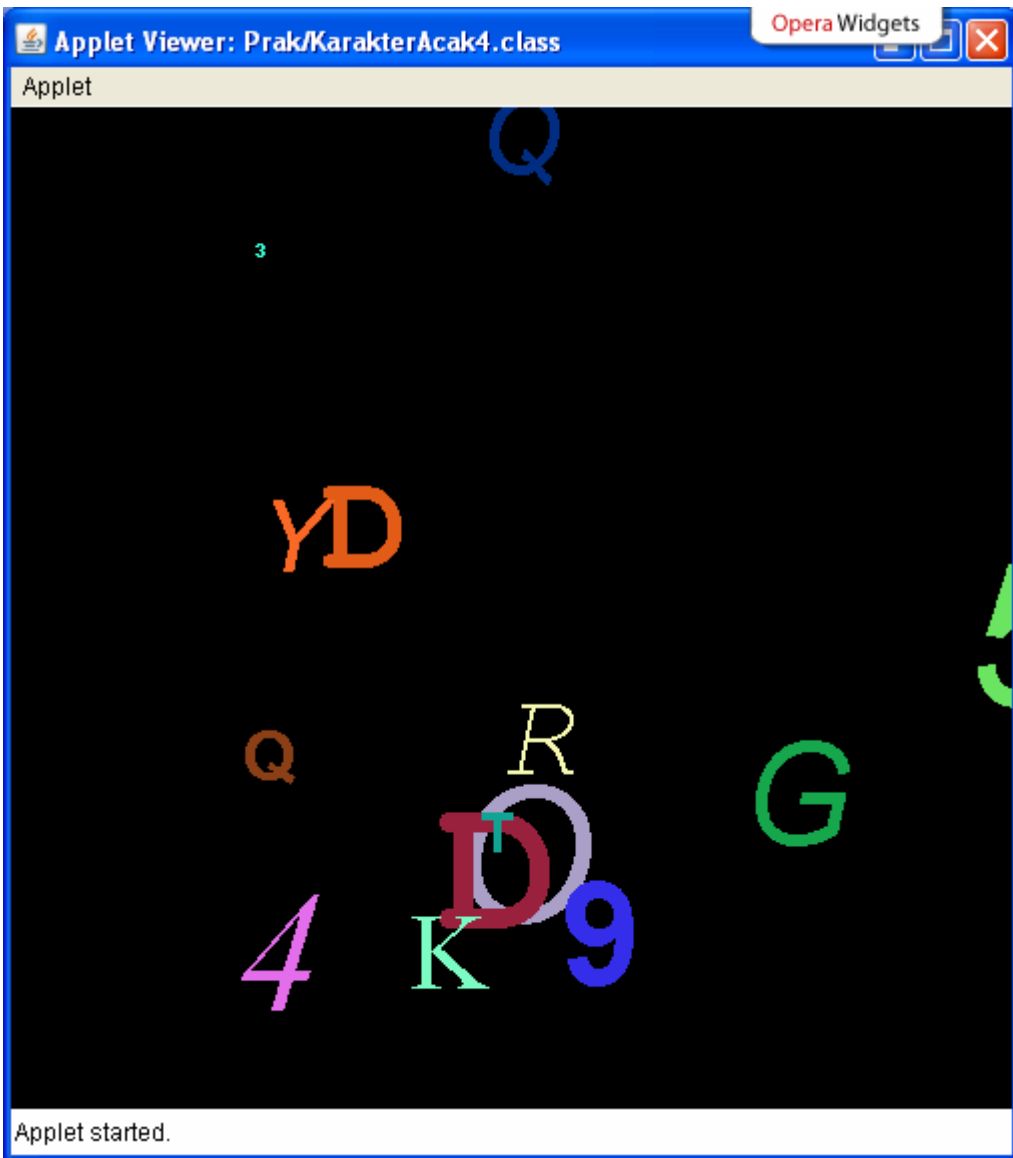
Praktikum 3 :

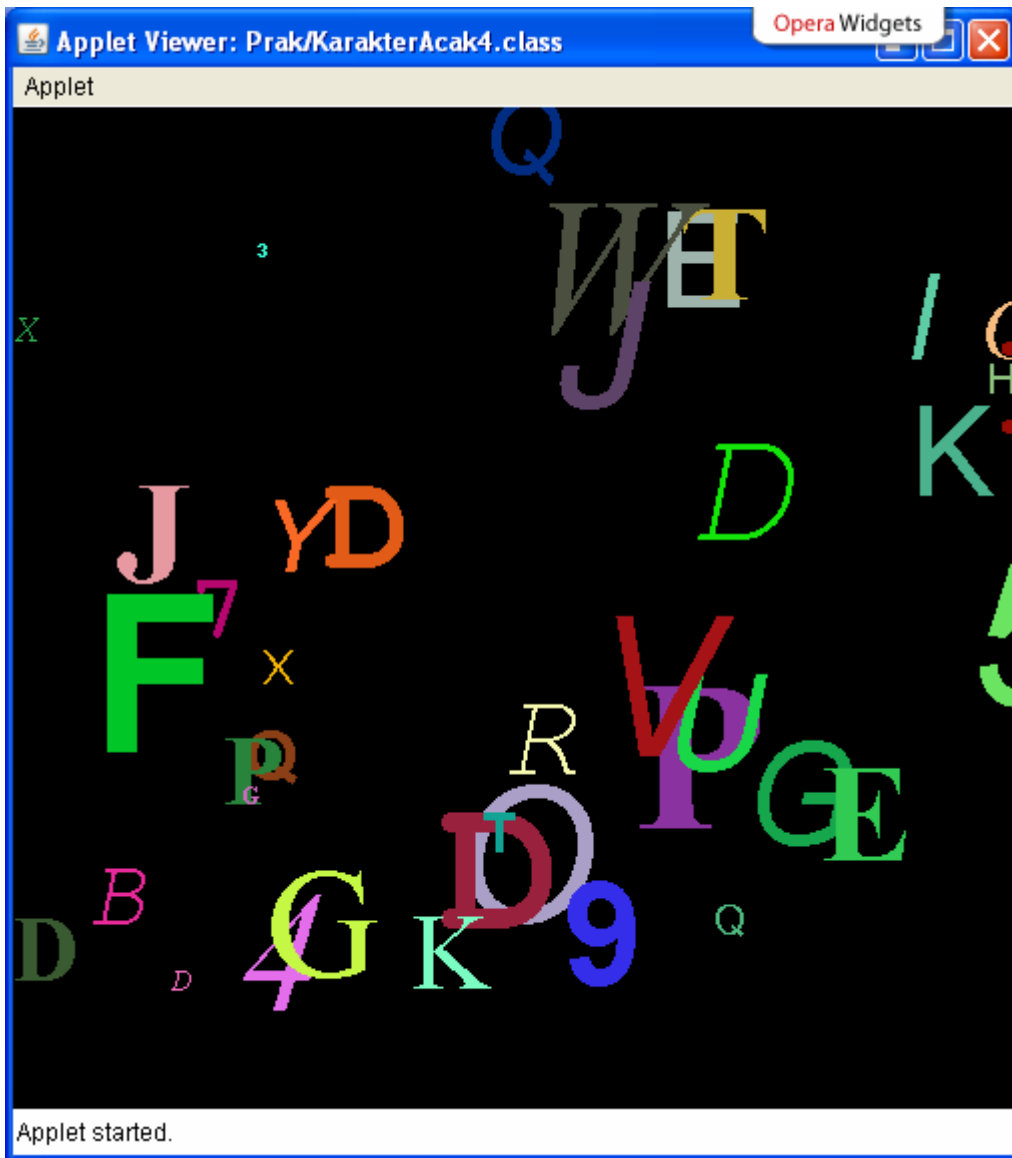
Praktikum 3 ini sama seperti praktikum 2 tapi menggunakan thread. Pertama kali bangkitkan huruf sebanyak 50. Huruf akan muncul pada applet satu persatu sampai 50 huruf. Jika huruf sudah mencapai 50 maka layar akan kembali bersih dan huruf akan muncul satu persatu lagi.





Gambar 20





Gambar 21

***** Selamat Mengerjakan*****