

BAB 11

Applet

11.1 Tujuan

Applet adalah satu fitur yang paling menarik dalam java. Applet merupakan program yang Anda jalankan melalui web browser. Pada pembelajaran ini Anda akan belajar tentang membuat sebuah applet.

Pada akhir pembahasan, diharapkan pembaca dapat :

1. Mendefinisikan apa yang dimaksud dengan applet
2. Membuat applet sendiri
3. Mengetahui siklus yang terdapat pada applet
 - *init*
 - *start*
 - *stop*
 - *destroy*
4. Menggunakan method applet yang lain
 - *paint*
 - *showStatus*
 - Method untuk memainkan sebuah audio clip
5. Memahami tag html pada applet

11.2 Membentuk Applet

Sebuah applet adalah tipe yang spesial dari program java yang dieksekusi melalui internet. Secara khusus berjalan pada suatu web browser seperti Netscape Navigator, Mozilla, atau Microsoft Internet Explorer. Bagaimanapun juga, jika dibandingkan dengan aplikasi Java yang normal, tidak diperbolehkan mengakses applet pada komputer yang dijalankan untuk alasan keamanan. Applet ini cukup terbatas jika dibandingkan dengan aplikasi Java.

Pada module ini, Anda akan mempelajari tentang membuat applet menggunakan AWT.

11.2.1 Hello World Applet

Class *Applet* adalah sebuah subclass dari class *Panel* yang didefinisikan dalam AWT. Cara terbaik untuk memahami bagaimana membuat applet adalah dengan contoh. Jadi, berikut ini adalah contoh applet sederhana yang menampilkan "Hello world!".

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

```
/* masukkan bagian ini dalam kode html
  <applet code="AppletDemo" width=300 height=100>
  </applet>
*/

public class AppletDemo extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello world!", 80, 25);
    }
}
```

Setelah proses kompilasi, usahakan menjalankan contoh ini menggunakan perintah java. Apa yang terjadi? Ingat bahwa applet adalah aplikasi java yang spesial. Mereka tidak dieksekusi menggunakan perintah java. Bahkan applet berjalan pada web browser atau menggunakan applet viewer. Untuk membuka applet melalui sebuah web browser, secara sederhana buka dokumen HTML dimana applet terintegrasi ke dalamnya menggunakan applet HTML tag (Perintah mengeluarkan kode pada contoh Hello World).

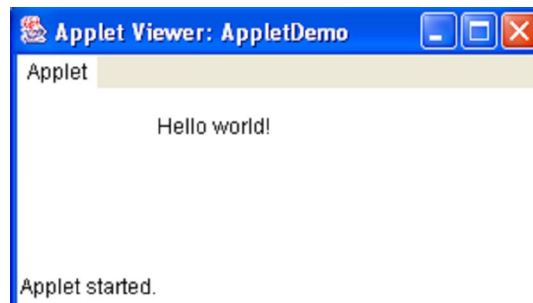
Cara lain untuk menjalankan sebuah applet adalah melalui perintah appletviewer. Untuk lebih mudahnya ikuti syntax berikut ini:

```
appletviewer <java filename>
```

Untuk menjalankan contoh applet yang dijalankan , menggunakan:

```
appletviewer AppletDemo.java
```

Tag HTML pada contoh yang diberikan mengindikasikan bahwa sebuah applet dibuat dengan lebar 300 pixel dan tinggi 100 pixel. Kemudian, method drawString menggambar string "Hello world!" pada posisi pixel (80,25) menghitung kebawah dari bagian kanan.



Gambar 1.1.1: Contoh Applet

Ketika membuat sebuah applet, diharuskan meng-extend class *Applet*. Sebagaimana yang disebutkan sebelumnya, classnya dapat ditemukan dalam package *java.applet*. Oleh karena itu, mengimport package *java.applet* merupakan suatu keharusan. Juga, telah disebutkan sebelumnya bahwa class *Applet* adalah subclass dari class *Panel*. Hal ini mengimplikasikan bahwa beberapa method dari class *Applet* ditemukan dalam class *Panel*. Untuk mengakses method atau field dalam class *Panel* atau class-class induk, diperlukan suatu aksi untuk import package *java.awt*.

11.3 Method-Method Applet

Bagian ini membahas manfaat-manfaat method applet.

11.3.1 Siklus Applet (*The Applet Life Cycle*)

Bahkan untuk memulai eksekusi pada method *main* seperti dalam aplikasi khas Java, browser atau applet viewer berhubungan dengan applet melalui method-method berikut:

1. *init()*

init adalah method yang dipanggil pertama kali. Yang sebenarnya berisi permintaan pertama ketika applet di load.

2. *start()*

Setelah meminta method *init*, mulai dengan method yang dipanggil selanjutnya. method ini meminta dokumen HTML yang ditampilkan applet setiap waktu. Eksekusi ringkasan dengan method ini dilakukan ketika applet ditampilkan kembali.

3. *stop()*

Ketika web browser meninggalkan dokumen HTML applet, method ini dipanggil untuk menginformasikan applet bahwa dia harus menghentikan proses eksekusinya.

4. *destroy()*

Method ini dipanggil ketika applet perlu dihapus dari memory. Method *stop* selalu dipanggil sebelum method ini diminta untuk dijalankan.

Ketika membuat applet, sedikitnya beberapa dari method ini telah meng-override. contoh applet berikut meng-override method berikut.

```
import java.applet.*;
import java.awt.*;
/*
   <applet code="LifeCycleDemo" width=300 height=100>
   </applet>
*/

class LifeCycleDemo extends Applet {
    String msg = "";
    public void init() {
        msg += "initializing... ";
        repaint();
    }
    public void start() {
        msg += "starting... ";
        repaint();
    }
    public void stop() {
        msg += "stopping... ";
        repaint();
    }
    public void destroy() {
        msg += "preparing for unloading...";
        repaint();
    }
    public void paint(Graphics g) {
        g.drawString(msg, 15, 15);
    }
}
```

Contoh dokumen html berikut tergabung dalam applet LifeCycleDemo .

```
<HTML>
<TITLE>Life Cycle Demo</TITLE>
   <applet code="LifeCycleDemo" width=300 height=100>
   </applet>
</HTML>
```

11.3.2 Method *paint*

Method lain yang tidak kalah penting adalah method *paint*, yang mana class *Applet* menurunkannya dari class induknya yaitu class *Component*, yang meminta output applet setiap waktu yang diperlukan untuk dapat digambar kembali. Sebagai contoh dari setiap instance adalah ketika sebuah applet tersembunyi oleh window lain dapat dibuat terlihat lagi. Method ini selalu menolak ketika Anda ingin membuat bagaimana applet yang Anda buat harus terlihat seperti yang Anda inginkan. Pada contoh Hello World, applet memiliki string "Hello world!" pada background setelah menolak method *paint*.

11.3.3 Method *ShowStatus*

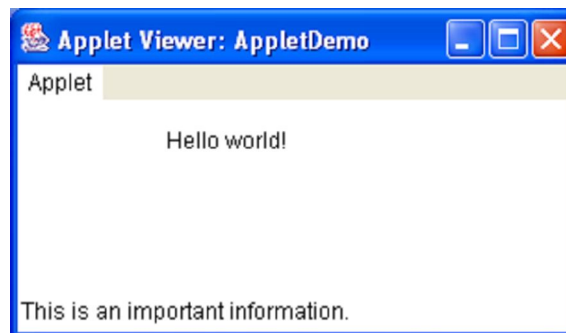
Applet memiliki window status, dimana memberi informasi kepada Anda tentang apa yang sebenarnya dilakukan applet. Jika Anda ingin memberi output ke window status, secara sederhana memanggil method *showStatus*.

Contoh berikut ini sama seperti contoh Hello World tapi dengan pernyataan tambahan yang memodifikasi isi dari window status.

```
import java.awt.*;
import java.applet.*;
/*
   <applet code="AppletDemo" width=300 height=100>
   </applet>
*/

public class AppletDemo extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello world!", 80, 25);
        showStatus("This is an important information.");
    }
}
```

Berikut ini adalah contoh hasil outputnya:



Gambar 1.2.3: contoh *showStatus()*

11.3.4 Memainkan Klip Audio

Applet juga menyediakan layanan melalui adanya suatu method yang memungkinkan Anda untuk memainkan file audio. Memainkan audio clips dalam sebuah applet melibatkan dua langkah dasar :

1. Dapatkan audio clip menggunakan method `getAudioClip`.
2. Untuk memainkan audio clip, menggunakan method `play` atau `loop` pada object audio clip. *play* memungkinkan Anda untuk memainkan audio satu kali mengingat *loop* berulang pada audio clip dan berhenti hanya ketika method *stop* dipanggil.

Contoh berikutnya memainkan file audio secara terus-menerus hingga method *stop* applet dipanggil.

```
import java.awt.*;
import java.applet.*;
/*
   <applet code="AudioApplet" width=300 height=100>
   </applet>
*/

public class AudioApplet extends Applet {
    AudioClip ac;
    public void init() {
        try {
            /*audio clip tersimpan dalam direktori yang sama
seperti kode javanya*/
            /* spaceMusic telah terdownload dari java.sun.com */
            ac = getAudioClip(getCodeBase(), "spaceMusic.au");
            ac.loop();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
    public void stop() {
        ac.stop();
    }
    public void paint(Graphics g) {
        g.drawString("Playing space music!", 80, 25);
    }
}
```

11.4 Tag-tag Applet HTML

Pada contoh sebelum-sebelumnya, Anda sudah melihat bagaimana tag-tag applet HTML digunakan dalam dokumen HTML atau source code java. Sekarang, Anda akan dikenalkan pada versi tag-tag applet HTML yang lebih lengkap.

```
<APPLET
  [CODEBASE = codebaseURL]
  CODE = appletFile
  [ATL = alternateText]
  [NAME = appletInstanceName]
  WIDTH = widthInPixels HEIGHT = heightInPixels
  [ALIGN = alignment]
  [VSPACE = vspaceInPixels] [HSPACE = hspaceInPixels]
>
[<PARAM NAME = parameterName1 VALUE = parameterValue1>]
[<PARAM NAME = parameterName2 VALUE = parameterValue2>]
...
[<PARAM NAME = parameterNamen VALUE = parameterValuen>]
[HTML that will be displayed in the absence of Java]
</APPLET>
```

Kata kunci Applet HTML Tag	
CODEBASE	
	Direktori dimana class applet diletakkan. Untuk dokumen HTML, direktori URL sesuai dengan setting awalnya/defaultnya.
CODE	
	Nama file yang berisi kode applet. Dengan atau tanpa nama ekstensi <i>.java</i> atau <i>.class</i> .
ALT	
	Text ditampilkan jika browser mengerti applet tags tapi applet tidak dapat dieksekusi secara langsung. Mungkin terjadi jika Javanya disabled.
NAME	
	Nama dari applet. Digunakan untuk memungkinkan applet yang lain untuk berkomunikasi dengan applet ini dengan menunjukkan suatu applet berdasarkan namanya.
WIDTH, HEIGHT	
	Width dan height dari window applet. Dinyatakan dalam pixel.
ALIGN	

Kata kunci Applet HTML Tag
Alignment atau pengaturan posisi dari applet. Satu diantara "left", "right", "top", "bottom", "middle", "baseline", "texttop", "absmiddle", atau "absbottom". Peletakan posisi secara Default tergantung pada lingkungan.
"top" – posisi atas dari applet diratakan dengan item tertinggi dalam baris yang ada.
"bottom", baseline – posisi bawah dari applet diratakan dengan bawah dari content lain dalam baris yang sama.
"middle" – tengah dari applet diratakan dengan bawah dari content yang lain dalam baris yang sama.
"texttop" – posisi atas dari applet diratakan dengan posisi atas dari applet diratakan dengan posisi tertinggi dari posisi atas pada baris yang sama.
"absmiddle" – tengah dari applet diratakan dengan vertical middle dari content lain pada baris yang sama.
"absbottom" – posisi bawah dari applet diratakan dengan posisi bawah dari content lain dalam baris yang sama.
VSPACE, HSPACE
Spasi diatas dan dibawah (VSPACE) dan pada sisi (HSPACE) dari applet..
PARAM NAME, VALUE
Untuk mengelompokkan parameter yang dapat menampilkan applet; applet dapat meminta method <i>getParameter(String paramName)</i> .

Table 1.3: Applet HTML Tags

Contoh dibawah ini mendemokan bagaimana untuk mengakses parameter tertentu pada HTML tag.

```
import java.awt.*;
import java.applet.*;
/*
   <applet code="ParamDemo" width=300 height=100>
   <param name="myParam" value="Hello world!">
   </applet>
*/

public class ParamDemo extends Applet {
    public void paint(Graphics g) {
        g.drawString(getParameter("myParam"), 80, 25);
    }
}
```

Output dari program ini sama seperti applet Hello World.

11.5 Latihan

11.5.1 Tic-Tac-Toe Applet satu-player

Buat game satu-player Tic-Tac-Toe. User memainkannya melawan komputer. Untuk setiap giliran, pemain harus menggeser kotak dari papan. Sekali sebuah kotak terpilih, kotak tersebut ditandai oleh symbol pemain (O dan X yang selalu digunakan sebagai simbol). pemain yang berhasil mengatasi 3 kotak membentuk baris horizontal, vertical atau diagonal memenangkan game ini. Game ini berakhir ketika pemain menang atau ketika semua kotak sudah berhasil dibentuk. Desain dan gerakan komputer seakan-akan user akan memenangkan pertandingan melawan komputer.