A. Kompetensi
Setelah mengikuti mata kuliah ini, mahasiswa diharapkan memiliki kemampuan membuat applet dan grafik.

B. Dasar Teori

**Java applets**

The applet is responsible for the graphical components and user interaction in a rectangular section of the Web browser's screen. An applet object has many instance variables and methods. Most of these are defined in the JApplet class. To access these definitions, your program should import javax.applet.JApplet and java.awt.*.

- **Extending the JApplet class.**

```java
import javax.swing.JApplet;
import java.awt.*;
public class Hello extends JApplet
               {}
               
```

- **The setBackground() method**

Use the following method of the Graphics object to change the pen color:

```java
setColor( Color.something )
```

For example, if `gr` is a reference to the Graphics object, then this changes the pen color to blue:

```java
gr.setColor( Color.blue );
```

Several million colors are possible, but for now, use the pre-defined colors:

```java
Color.red     Color.orange     Color.yellow     Color.green
Color.blue    Color.magenta    Color.white      Color.black
Color.gray    Color.darkGray   Color.lightGray  Color.pink
Color.cyan
```

- **The paint() method.**

The class JApplet has a paint() method, but that method does little. The Web browser calls the paint() method when it needs to "paint" the section of the monitor screen devoted to an applet. Each applet that you write has its own paint() method.

```java
public void paint ( Graphics gr )
{
        
}
```
• The Graphics class.

Methods of the Graphics class:

- **drawString(String str, int x, int y)**
  - **str**: a String to place into the drawing area
  - **x**: horizontal distance in pixels from the left edge
  - **y**: vertical distance in pixels from the top edge

The x and y parameters tell where to place the string within the applet's area. This is like graph paper, except the (0,0) location is the upper left corner (of the applet's drawing area, not of the full screen). Increasing y values move down the area. The location (x, y) is where to place the lower left part of the first character of the string.

- **drawOval(int X, int Y, int width, int height)**

This draws a circle or an oval that fits within the rectangle specified by the X, Y, width and height arguments. The oval is drawn inside a rectangle whose upper left hand corner is at (X, Y), and whose width and height are as specified. The measurements are in terms of pixels. The (X, Y) position is relative to the upper left hand corner of the applet's drawing area. It gives the location of the upper left corner of a rectangle. The width and height arguments give the horizontal and vertical size of the rectangle. The drawOval() method draws the oval that fits inside the rectangle. (It does not actually draw the rectangle.)

You do not give the method the center and radius of the circle you want.
o \texttt{drawLine}(\texttt{int} \ x_1, \texttt{int} \ y_1, \texttt{int} \ x_2, \texttt{int} \ y_2))

This draws a line from \((x_1, y_1)\) to \((x_2, y_2)\) in the drawing area using the current pen color. As usual, the point \((0,0)\) is the upper left corner of the drawing area. It doesn't matter in which direction you draw the line. The end and the start of the line could be reversed.

o \texttt{drawRect}(\texttt{int} \ x, \texttt{int} \ y, \texttt{int} \ width, \texttt{int} \ height)

It draws the outline of a rectangle using the current pen color. The left and right edges of the rectangle are at \(x\) and \(x + \text{width}\) respectively. The top and bottom edges of the rectangle are at \(y\) and \(y + \text{height}\) respectively. This method is also used to draw a square. This applet draws a rectangle around the entire drawing area, then puts another rectangle in the center.

\section*{Running an JApplet}

An applet does not have a \texttt{main()} method and so cannot be started as an application. For the applet to do anything, some other application must be running and call the applet's methods. A Web browser can do this. A Web page is a text file that a Web browser can display. The browser follows instructions that are part of the text to nicely format the page. Here is a tiny Web page that asks for the applet \texttt{Hello}:

\begin{verbatim}
<html>
<body>
<applet code="Hello.class" width="300" height="150">
</applet>
</body>
</html>
\end{verbatim}

This tells the browser to run the applet \texttt{Hello.class}, and to use a 300 pixels wide and 150 high drawing area. Quote marks are required around all the values, including the numbers, but usually web browsers will work if you forget them. You can create this Web page by typing the above characters into a text editor (or copy and paste them). Then save the file as \texttt{Hello.html} in the same directory that has \texttt{Hello.class}.

\section*{C. Alat/ Bahan}

1. Lab. Sheet Pemrograman Java 2
2. PC / Laptop with OS installed
3. JDK 1.5 or latest
4. J-Creator or text editor

D. Langkah Kerja
1. Baca dan pahami dasar teori di atas.
2. Lakukan kompilasi dan eksekusi terhadap contoh-contoh source code atau program yang ada di dasar teori dan LAMPIRAN.

E. Tugas Individu
Buatlah sebuah applet yang menampilkan:

- 4 buah garis berbeda warna (merah, hijau, orange dan magenta). Ketebalan garis adalah 2x tebal garis pada umumnya.
- 50 buah lingkaran berwarna biru dengan kriteria:
  - Diameter semua lingkaran adalah sama
  - Sumbu X memiliki selisih 3 pixel dari Sumbu X sebelumnya.
  - Sumbu Y memiliki selisih 2 pixel dari Sumbu Y sebelumnya.
- Tambahkan Nama dan NIM anda masing-masing

Hasil susunan berbentuk seperti gambar di bawah ini (boleh menggunakan looping).
F. Lampiran

- **CH 36, 37, dan 38** (Introduction to Computer Science using Java, Java 5.0 version, January 2006, Bradley Kjell, Central Connecticut State University http://chortle.ccsu.edu/CS151/cs151java.html)