

Pertemuan 12

Line 2D dan Rectangle2

- Line 2Dimensi
- Rectangle 2Dimensi

Class Line2D

Class Line2D

Line2D.Double dan **Line2D.Float** adalah 2 sub-classes dari **Line2D**. Kedua class tersebut sama-sama menggambar garis, tetapi berbeda dalam representasi nilai koordinatnya. Untuk membuat sebuah objek **Line2D** dengan tipe double, gunakan *constructor*:

```
Line2D line = new Line2D.Double(x1, y1, x2, y2);
```

Class Line2D

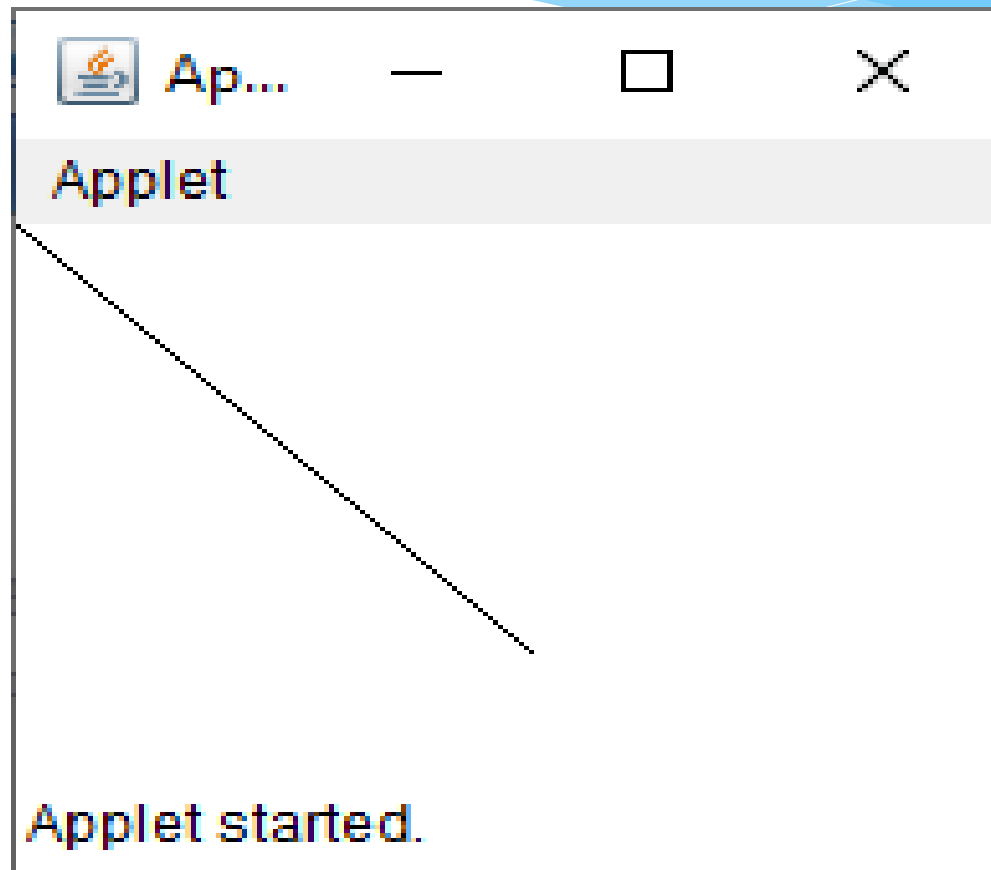
Contoh :

```
import java.awt.*;
import java.awt.geom.*;

public class SampleLine2D extends java.applet.Applet
{
    public void paint(Graphics g)
    {
        Graphics2D g2d = (Graphics2D)g;
        Shape s = new Line2D.Double(0, 0, 100, 100);
        g2d.draw(s); // outline
    }
}
```

Output Class Line2D

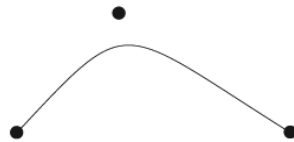
Hasil



QuadCurve2D

QuadCurve2D

QuadCurve2D merepresentasikan sebuah quadratic curve yang didefinisikan dengan tiga buah titik kontrol. Titik pertama dan terakhir merupakan titik-titik ujung kurva. Titik kedua biasanya tidak terletak pada kurva, tetapi menunjukkan kelengkungan kurva, seperti ditunjukkan pada gambar 3.2.



Gambar 3.2 QuadCurve2D didefinisikan dengan 3 titik kontrol

Sebuah objek QuadCurve2D dapat diciptakan dengan *constructor*:

```
QuadCurve2D quad = new QuadCurve2D.Double(x1, y1, x2, y2, x3, y3);
```

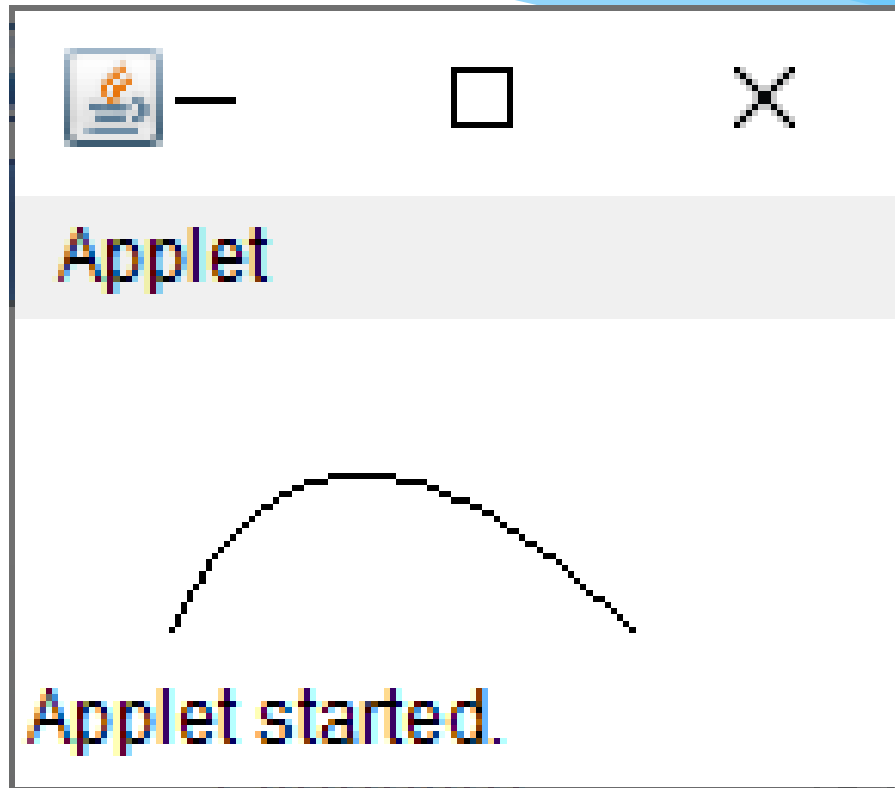
QuadCurve2D

Contoh :

```
import java.awt.*;
import java.awt.geom.*;
public class SampleQuarCurve2D extends java.applet.Applet
{
    public void paint(Graphics g)
    {
        // Cast Graphics to Graphics2D
        Graphics2D g2d = (Graphics2D)g;
        Shape s = new QuadCurve2D.Double(25, 50,50,0,100, 50);
        g2d.draw(s); // outline
        //g2d.fill(s); // solid
    }
}
```


Output QuadCurve2D

Hasil :



CubicCurve2D

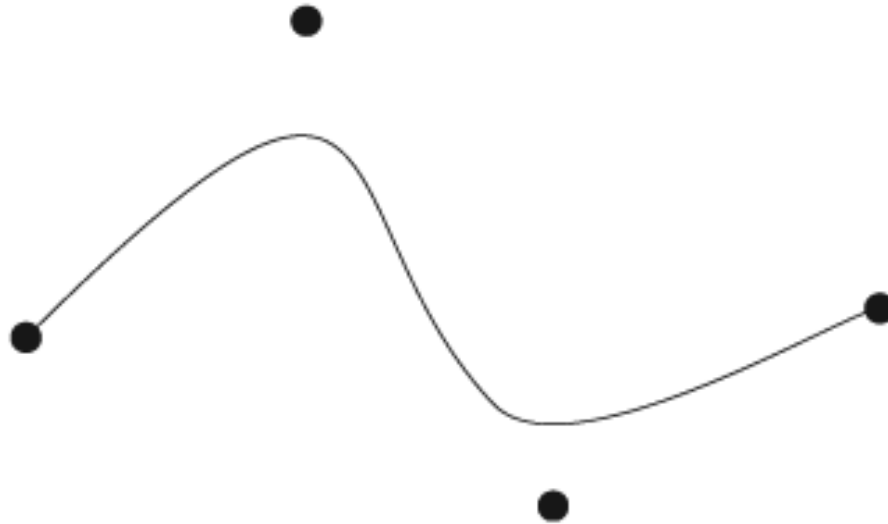
CubicCurve2D

CubicCurve2D merepresentasikan kurva *cubic Bezier* yang didefinisikan dengan **4 titik kontrol**. Sama seperti *quadratic curve*, titik kontrol pertama dan terakhir adalah titik ujung kurva.

Dua titik kontrol di antaranya mendefinisikan bentuk kelengkungan kurva, dan tidak terletak pada kurva, seperti ditunjukkan pada **gambar 3.3**.

CubicCurve2D

```
CubicCurve2D quad = new  
CubicCurve2D.Double(x1, y1, x2, y2, x3, y3, y4, y5);
```



Gambar 3.3 CubicCurve2D didefinisikan dengan 4 titik kontrol

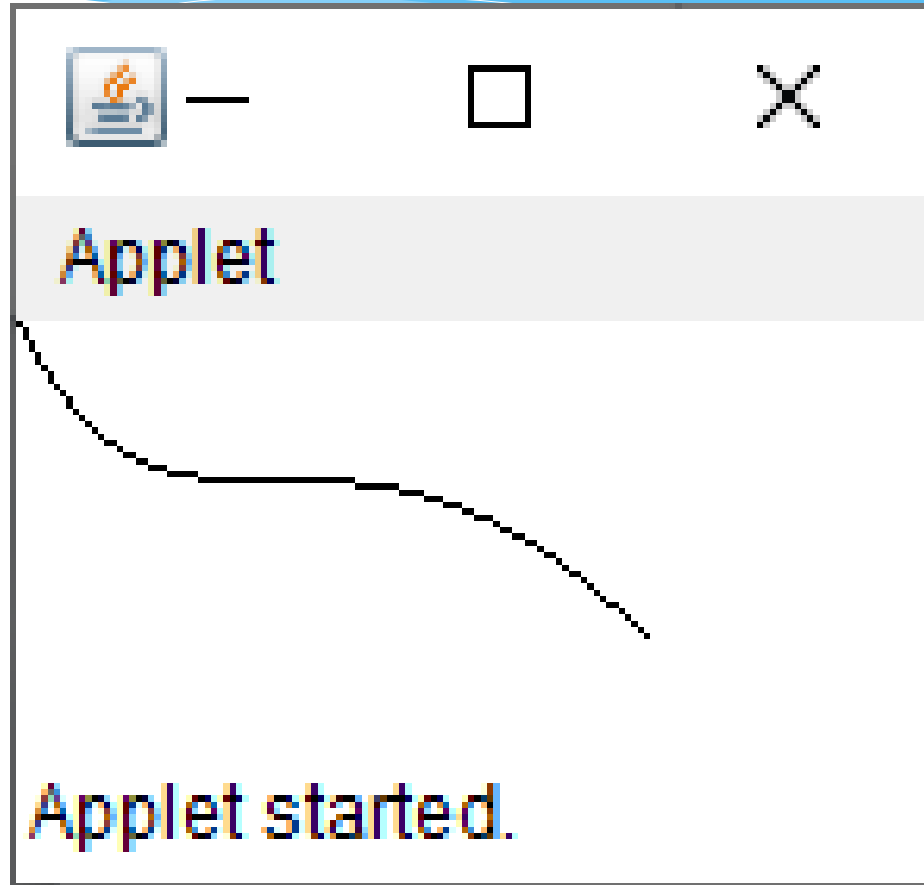
CubicCurve2D

Contoh :

```
import java.awt.*;
import java.awt.geom.*;

public class SampleCubicCurve2D extends java.applet.Applet
{
    public void paint(Graphics g)
    {
        Graphics2D g2d = (Graphics2D)g;
        Shape s = new CubicCurve2D.Double (0,0,25,50, 50,0,100, 50);
        g2d.draw(s); // outline
        //g2d.fill(s); // solid
    }
}
```

Output CubicCurve2D



Rectangle2D

Rectangle2D

Rectangle2D mendefinisikan bentuk segi empat, didefinisikan dengan sebuah titik ujung pertama dan panjang dan lebar dari bentuk segi empat tersebut.

Bentuk ini diciptakan dengan *constructor*:

```
Rectangle2D ri = new  
Rectangle(x0, y0, length, width);  
Rectangle2D rd = new  
Rectangle2D.Double(x0.0, y0.0, length.0, width.0);  
Rectangle2D rf = new  
Rectangle2D.Float((x0) f, (y0) f, (length) f, (width) f  
);
```


Rectangle2D

Contoh :

```
import java.awt.*;
import java.awt.geom.*;

public class SampleRectangle2D extends
java.applet.Applet
{
    public void paint(Graphics g)
        {
            Graphics2D g2d = (Graphics2D)g;
            Shape s = new Rectangle2D.Double(0,0,100, 50);
            g2d.draw(s); // outline
            //g2d.fill(s); // solid
        }
}
```

Output Rectangle2D



Latihan

Soal :

1. Buatlah Persegi pajang ke bawah dengan Baris dengan Metode Rectangle2D
2. Buatlah Persegi pajang ke Samping dengan Baris dengan Metode Rectangle2D
3. Buatlah Persegi pajang ke Samping dengan Baris dengan Metode Rectangle2D dengan jumlah 5 Buah

Terima Kasih