

Pertemuan ke-5: Geometri II & Aplikasinya

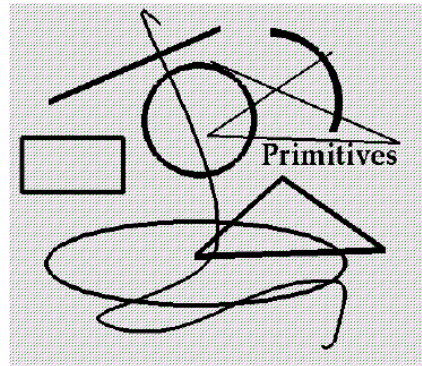


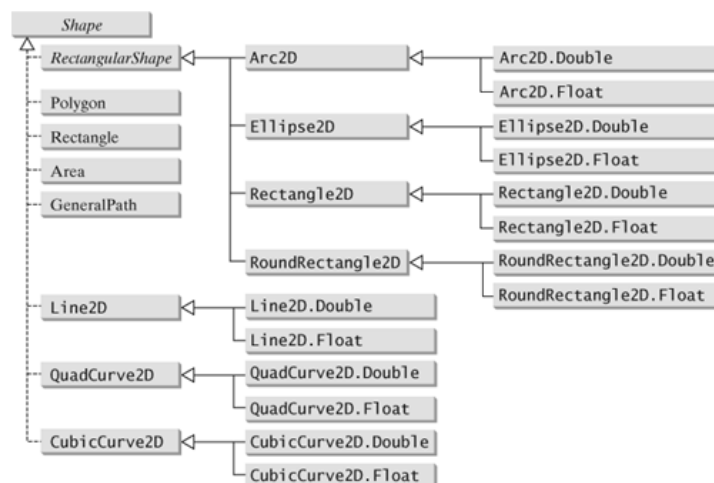
Image Sources:

<http://thewizardofhoz.com/graphics.html>

<http://www.siggraph.org/~rhyne/com97/com97-tut.html>

CG1: Universitas Pamulang

Komponen Geometri pada Graphics2D



CG1: Universitas Pamulang

Komponen-komponen Shape

- **Points**

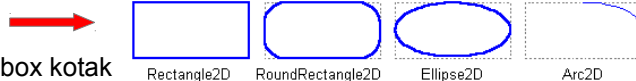
- Representasi lokasi (x,y) pada sistem koordinat. Tidak bisa di-render

- **Lines**

- Objek garis antara (x1, y1) dan (x2, y2)

- **Rectangular Shapes**

- Objek shape dengan bounding-box kotak dengan posisi (x, y) dan ukuran (width, height)



- **Quadratic and Cubic Curves**

- Objek kurva (quadratic dan cubic)



- **Arbitrary Shapes (*GeneralPath*)**

- Objek geometri sembarang yang terdiri dari garis dan kurva

- **Areas**

- Konstruksi objek geometri kompleks melalui operasi boolean seperti union, intersect dan subtract dari beberapa objek geometri.



Geometri dengan *GeneralPath*

- Class untuk mengkonstruksi objek geometri yang terdiri dari **Line** dan **Curve**

- **moveTo(x, y)**

- Memindahkan titik *Path* ke (x,y)

- **lineTo(x, y)**

- Membuat garis dari titik *Path* ke (x,y)

- **quadTo(x1, y1, x2, y2)**

- Membuat kurva dari titik *Path* ke (x2, y2) dengan (x1, y1) sebagai titik kontrol

- **curveTo(x1, y1, x2, y2, x3, y3)**

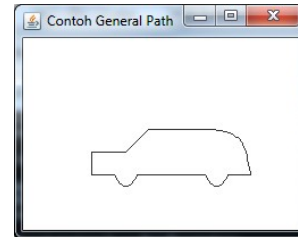
- Membuat kurva dari titik *Path* ke (x3, y3) dengan (x1, y1) dan (x2, y2) sebagai titik kontrol

- **closePath()**

- Menutup kurva dengan cara membuat garis dari titik *Path* ke posisi *moveTo* yang terakhir

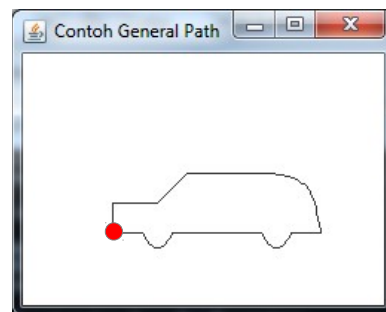
Contoh *GeneralPath*

```
GeneralPath gp = new GeneralPath();
gp.moveTo(60,120);
gp.lineTo(80,120);
gp.quadTo(90,140,100,120);
gp.lineTo(160,120);
gp.quadTo(170,140,180,120);
gp.lineTo(200,120);
gp.curveTo(195,100, 200,80,160,80);
gp.lineTo(110,80);
gp.lineTo(90,100);
gp.lineTo(60,100);
gp.lineTo(60,120);
g2.draw(gp);
```



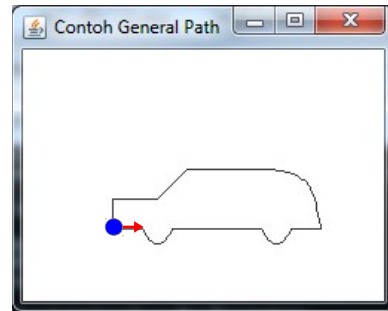
Ilustrasi *GeneralPath* - 1

```
GeneralPath gp = new GeneralPath();
gp.moveTo(60,120);
gp.lineTo(80,120);
gp.quadTo(90,140,100,120);
gp.lineTo(160,120);
gp.quadTo(170,140,180,120);
gp.lineTo(200,120);
gp.curveTo(195,100, 200,80,160,80);
gp.lineTo(110,80);
gp.lineTo(90,100);
gp.lineTo(60,100);
gp.lineTo(60,120);
g2.draw(gp);
```



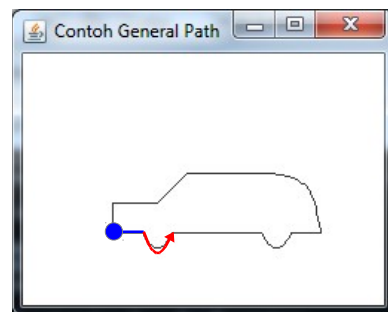
Ilustrasi *GeneralPath* - 2

```
GeneralPath gp = new GeneralPath();  
gp.moveTo(60,120);  
gp.lineTo(80,120);  
gp.quadTo(90,140,100,120);  
gp.lineTo(160,120);  
gp.quadTo(170,140,180,120);  
gp.lineTo(200,120);  
gp.curveTo(195,100, 200,80,160,80);  
gp.lineTo(110,80);  
gp.lineTo(90,100);  
gp.lineTo(60,100);  
gp.lineTo(60,120);  
g2.draw(gp);
```



Ilustrasi *GeneralPath* - 2

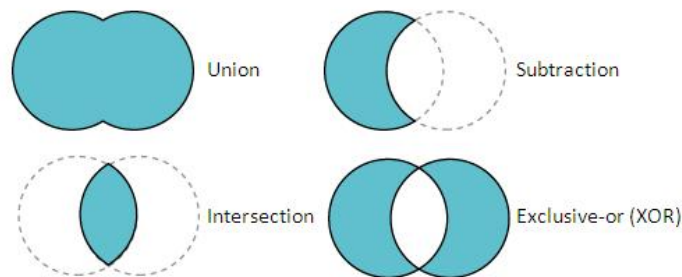
```
GeneralPath gp = new GeneralPath();  
gp.moveTo(60,120);  
gp.lineTo(80,120);  
gp.quadTo(90,140,100,120);  
gp.lineTo(160,120);  
gp.quadTo(170,140,180,120);  
gp.lineTo(200,120);  
gp.curveTo(195,100, 200,80,160,80);  
gp.lineTo(110,80);  
gp.lineTo(90,100);  
gp.lineTo(60,100);  
gp.lineTo(60,120);  
g2.draw(gp);
```



Dan seterusnya ...

Class Area

- Cara untuk membuat objek geometri yang kompleks melalui operasi boolean seperti *union*, *subtract*, *intersect*
 - `add(Area a)` : operasi union/gabungan
 - `intersect(Area a)` : operasi intersect/irisan
 - `subtract(Area a)` : operasi pengurangan
 - `exclusiveOr(Area a)` : operasi XOR



CG1: Universitas Pamulang

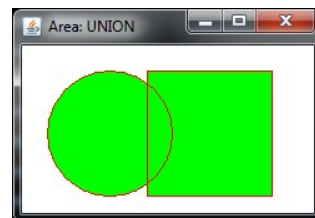
Source: <http://www.java-forums.org/blogs/java-awt/1045-constructing-complex-shapes-geometry-primitives.html>

Area: union/gabungan

```
//Definisikan lingkaran dan kotak
Shape lingkaran = new Ellipse2D.Double(20, 20, 100, 100);
Shape kotak = new Rectangle2D.Double(100, 20, 100, 100);
//Definisikan area untuk masing-masing shape
Area area = new Area(lingkaran);
Area areaKanan = new Area(kotak);
```

```
//Operasi UNION
area.add(areaKanan);
g2.setColor(Color.GREEN);
g2.fill(area);

g2.setColor(Color.RED);
g2.draw(lingkaran);
g2.draw(kotak);
```



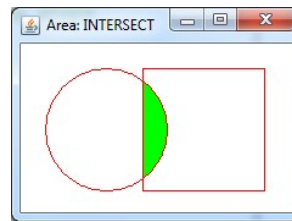
CG1: Universitas Pamulang

Area: intersect/irisan

```
//Definisikan lingkaran dan kotak
Shape lingkaran = new Ellipse2D.Double(20, 20, 100, 100);
Shape kotak = new Rectangle2D.Double(100, 20, 100, 100);
//Definisikan area untuk masing-masing shape
Area area = new Area(lingkaran);
Area areaKanan = new Area(kotak);

//Operasi INTERSECT
area.intersect(areaKanan);
g2.setColor(Color.GREEN);
g2.fill(area);

g2.setColor(Color.RED);
g2.draw(lingkaran);
g2.draw(kotak);
```

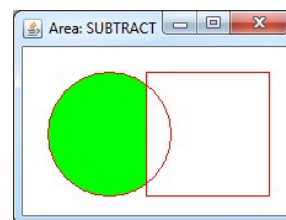


Area: subtract/pengurangan

```
//Definisikan lingkaran dan kotak
Shape lingkaran = new Ellipse2D.Double(20, 20, 100, 100);
Shape kotak = new Rectangle2D.Double(100, 20, 100, 100);
//Definisikan area untuk masing-masing shape
Area area = new Area(lingkaran);
Area areaKanan = new Area(kotak);

//Operasi SUBTRACT
area.subtract(areaKanan);
g2.setColor(Color.GREEN);
g2.fill(area);

g2.setColor(Color.RED);
g2.draw(lingkaran);
g2.draw(kotak);
```

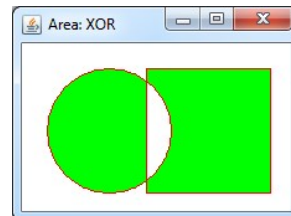


Area: XOR

```
//Definisikan lingkaran dan kotak
Shape lingkaran = new Ellipse2D.Double(20, 20, 100, 100);
Shape kotak = new Rectangle2D.Double(100, 20, 100, 100);
//Definisikan area untuk masing-masing shape
Area area = new Area(lingkaran);
Area areaKanan = new Area(kotak);

//Operasi XOR
area.exclusiveOr(areaKanan);
g2.setColor(Color.GREEN);
g2.fill(area);

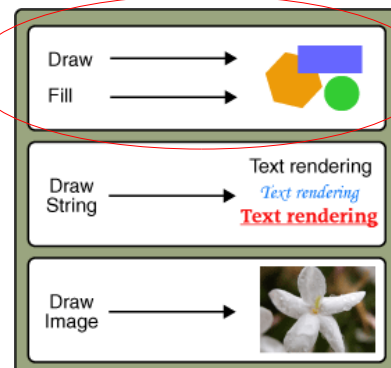
g2.setColor(Color.RED);
g2.draw(lingkaran);
g2.draw(kotak);
```



CG1: Universitas Pamulang

Method-method dalam Graphics

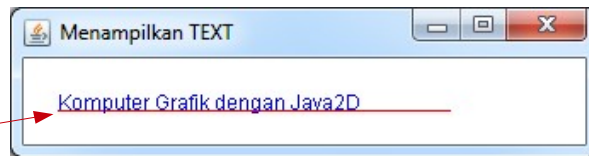
- Secara garis besar ada 2 group method dalam *Graphics*
 1. *draw* dan *fill* untuk merender shape, text, image
 2. Method untuk mengeset atribut dari objek grafik seperti *Font*, *Warna*, dsb
- Method untuk render



CG1: Universitas Pamulang Source <http://docs.oracle.com/javase/tutorial/2d/basic2d/index.html>

Menambahkan *Text*

- Untuk menambahkan *Text*, gunakan method
 - `drawString(text, x, y)` dimana *x, y* adalah posisi *baseline* dari text yang ingin ditampilkan



(20, 30) baseline

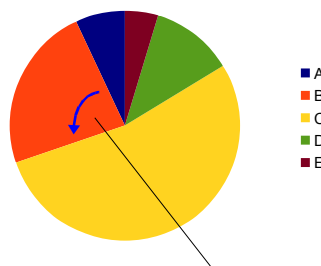
```
//Garis untuk menandai POSISI
g2.setColor(Color.RED);
g2.draw(new Line2D.Float(20, 30, 250, 30));

//Tampilkan text
g2.setColor(Color.BLUE);
g2.drawString("Komputer Grafik dengan Java2D", 20, 30);
```

Contoh: *Pie Chart*

- Menampilkan data dalam lingkaran yang dibagi dalam beberapa sektor, dimana besarnya sudut menunjukkan prosentase nilai suatu variabel dari nilai total
- Contoh: menampilkan distribusi nilai ujian

Nilai	Jumlah
A	3
B	10
C	23
D	5
E	2
Total	43



Besar Sudut: $(360 \text{ derajat}) * 10 / 43$

Langkah Pembuatan - 1

- Import *package-package* Java2D yang diperlukan
- Buat *class* dengan nama *PieChart*
- Tambahkan field untuk menyimpan data nilai, jumlah mahasiswa dan warna tiap-tiap variabel

```
//Buat class, nama disesuaikan dengan project
public class PieChart extends JPanel
{
    //Definisikan field untuk menyimpan data
    String [] nilaiUjian = {"A", "B", "C", "D", "E"};
    int [] jumlahMahasiswa = {3, 10, 23, 5, 2};
    Color [] warna = {Color.BLUE, Color.ORANGE,
        Color.YELLOW, Color.GREEN, Color.RED};
}
```

Langkah Pembuatan - 2

- Definisikan *constructor*, method *paintComponent* dan *main*
- Dalam *paintComponent* tambahkan kode untuk setting *Antialias* dan membuat *Frame*

```
//Gunakan antialias, agar hasil bagus
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
    RenderingHints.VALUE_ANTIALIAS_ON);

//Frame background
Shape bg = new Rectangle2D.Float(10, 10, 250, 200);
g2.setColor(Color.LIGHT_GRAY);
g2.fill(bg);
g2.setColor(Color.GRAY);
g2.draw(bg);
```

Langkah Pembuatan - 3

- Tambahkan kode untuk menghitung total jumlah mahasiswa (atau variabel serupa pada kasus lain)

```
//Hitung total
float total = 0.0f;
for(int k = 0; k < jumlahMahasiswa.length; k++)
    total += jumlahMahasiswa[k];
```

- Tambahkan kode untuk menampilkan chart:
 - Hitung Sudut
 - Tampilkan PIE
 - Tampilkan LEGEND

Langkah Pembuatan - 4

- Menghitung sudut/prosentasi tiap-tiap sektor

```
float sudut, awal = 90; //Sudut PIE
float lx = 220, ly = 70, lw = 10, lh = 10; //Legend
for (int k = 0; k < jumlahMahasiswa.length; k++) {
    //Hitung besarnya sudut tiap-tiap sektor
    sudut = 360.0f * jumlahMahasiswa[k] / total;
    Shape sektor = new Arc2D.Float(30, 30, 150, 150,
        awal, sudut, Arc2D.PIE);
```

- Tampilkan (Render) PIE

```
//Tampilkan
g2.setColor(warna[k]);
g2.fill(sektor);
awal += sudut;
```

Langkah Pembuatan - 5

- Tampilkan LEGEND disebelah kanan

```
//Legend
g2.fill(new Rectangle2D.Float(lx, ly, lw, lh));
g2.setColor(Color.BLACK);
g2.drawString(nilaiUjian[k], lx + lw + 5, ly + lh);
ly += (lh + 5);
}
```

- Tambahkan signatur (pilihan)

```
//Signature
g2.setColor(Color.GRAY);
g2.drawString("Created by: I Putu Susila", 15, 205);
```

CG1: Universitas Pamulang

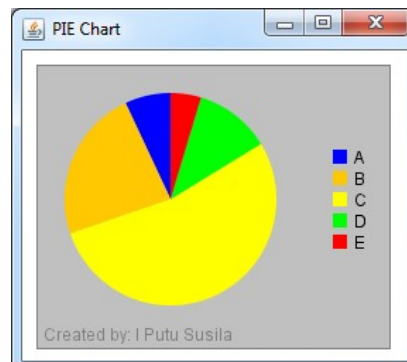
Langkah Pembuatan - 6

- Method *main*

```
public static void main(String args[]) {
    //Buat frame dengan title sesuai kebutuhan
    JFrame frame = new JFrame("PIE Chart");
    frame.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e)
        {System.exit(0);}
    });

    //tambahkan objek dari PieChart
    PieChart canvas = new PieChart();
    frame.getContentPane().add(canvas);

    frame.pack();
    frame.setVisible(true);
}
```



CG1: Universitas Pamulang

