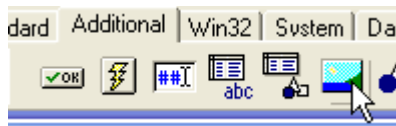


Minggu ke : 12
Penyusun : Yudi Wibisono
Materi : 7. Pengantar Fasilitas Grafik dalam Delphi
7.1 Koordinat, Canvas
7.2 Rectangle, Circle, MoveTo, LineTo, Polygon, RoundRect
7.3 Pen, Brush
7.4 FloodFill

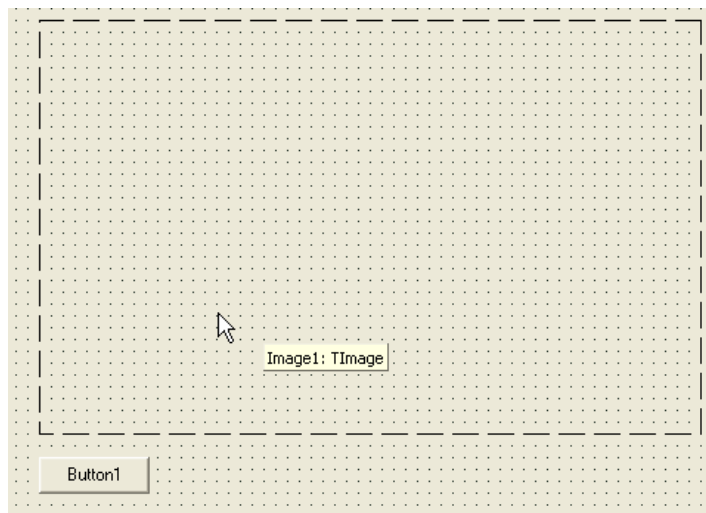
URAIAN POKOK PERKULIAHAN

Pendahuluan

Menggambar di Delphi dilakukan dengan memanipulasi canvas. Canvas dapat dianalogikan sebagai sebuah canvas pelukis yang dapat digambar sesuai dengan kehendak pelukisnya. Pada materi ini yang akan kita gunakan adalah canvas pada komponen TImage (ada di bagian Additional)



Coba letakkan komponen Timage, letakkan satu tombol



Double klik button1, isi dengan

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    image1.Canvas.Rectangle(10,10,100,100);  
end;
```

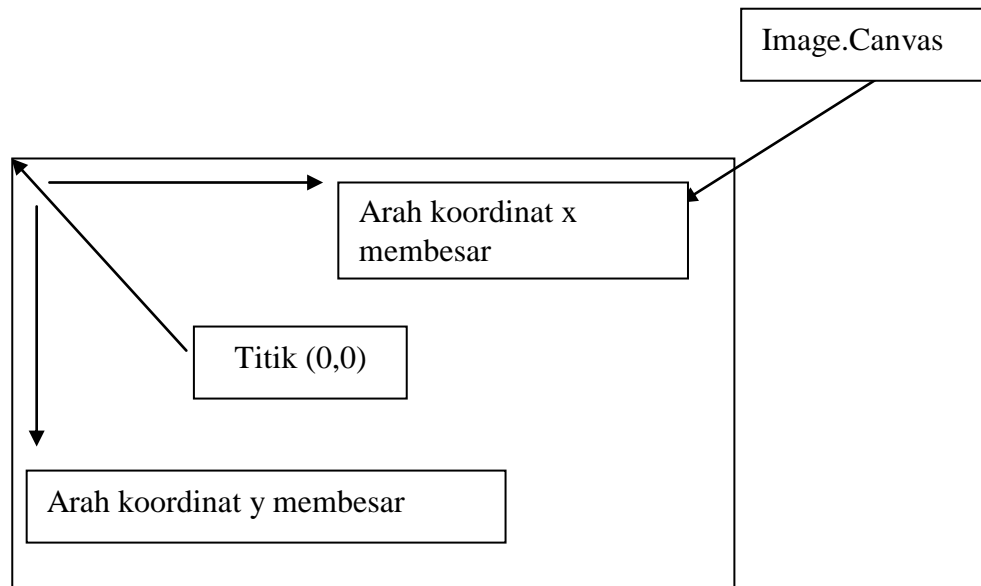
Eksekusi program (F9) maka sebuah kotak akan muncul di layar. Sangat mudah bukan?

Pada kode diatas yang kita lakukan adalah memerintahkan objek **image1.Canvas** untuk menggambar kotak dengan koordinat sudut kiri atas (10,10) dan sudut kanan bawah (100,100).

Method adalah fungsi atau procedure yang dimiliki sebuah objek. Jika kita analogikan manusia sebagai objek, maka contoh method untuk objek manusia adalah: berjalan, membaca dsb. Pada Delphi method ditulis dengan notasi titik: **manusia.berjalan, manusia.membaca**

Sistem Koordinat Canvas

Aturan koordinat pada canvas adalah sebagai berikut:



Semua koordinat bernilai positif dan merupakan bilangan bulat (tidak ada pecahan)

Berbagai Method Canvas untuk Menggambar

Objek canvas memiliki beberapa method lain untuk menggambar:

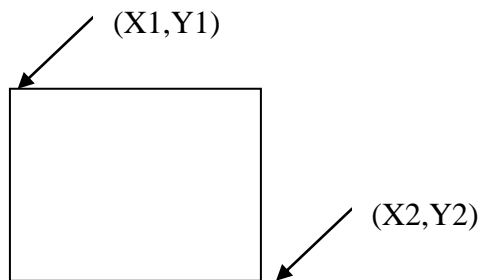
| Nama Method | Keterangan |
|-------------|---|
| Arc | Menggambar lengkungan |
| Ellipse | Menggambar lingkaran/ellips. |
| MoveTo | Menggeser posisi |
| LineTo | Menggambar garis (digunakan bersama MoveTo) |
| Rectangle | Membuat kotak |

| | |
|-----------|--|
| RoundRect | Membuat kotak yang memiliki sudut melengkung |
| Polygon | Menggambar poligon |

Berikut akan kita bahas secara lebih rinci beberapa diantaranya.

Rectangle

Rectangle digunakan untuk membuat kotak. Method Rectangle mempunyai empat parameter (X1,Y1,X2,Y2) . (X1,Y1) menyatakan koordinat **kiri-atas** kotak dan (X2,Y2) koordinat **kanan-bawah** kotak.



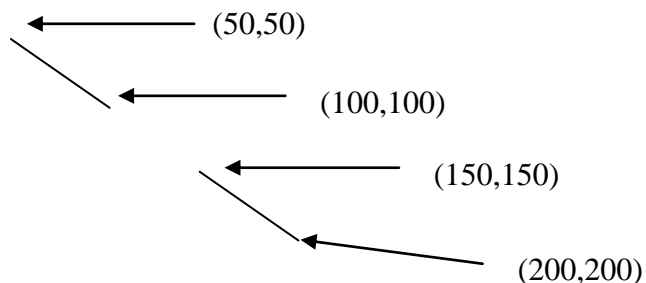
MoveTo dan LineTo

MoveTo dan LineTo digunakan secara bersama-sama untuk membuat garis. MoveTo adalah perintah untuk memindahkan posisi menggambar **tanpa** membuat garis. Sedangkan LineTo memindahkan posisi dengan **menggambar** garis.

Contoh:

```
with image1.Canvas do
begin
  MoveTo(50,50); //pindahkan posisi ke koordinat 50,50
  LineTo(100,100); //buat garis yang menghubungkan titik (50,50) ke (100,100)
  MoveTo(150,150); //pindahkan posisi ke(150,150). Tdk ada garis yg digambar!
  LineTo(200,200); //buat garis dari (150,150) ke (200,200)
end;
```

Hasilnya adalah sebagai berikut:



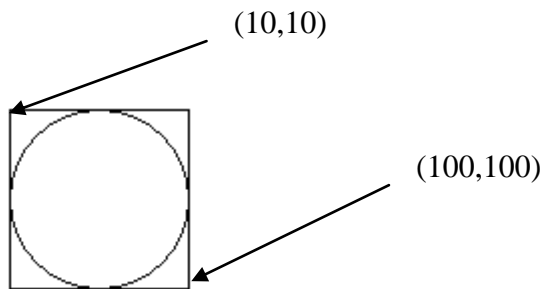
Ellipse

Ellipse digunakan untuk menggambar lingkaran dan elips.

Contoh:

```
Image1.Canvas.Rectangle(10,10,100,100);  
Image1.Canvas.Ellipse(10,10,100,100);
```

Akan menghasilkan:



Dari potongan program diatas, dapat dilihat bahwa perintah Ellipse memiliki parameter yang sama dengan rectangle (x1,y1, x2, y2). Ellipse akan menggambar lingkaran sesuai dengan kotak yang dibentuk oleh koordinat tersebut. Untuk membuat lingkaran sempurna, Anda harus membentuk koordinat kotak yang berbentuk bujur sangkar (semua sisinya sama panjang).

Coba tukar kedua baris pada program diatas, sehingga lingkaran digambar terlebih dulu.

```
Image1.Canvas.Ellipse(10,10,100,100);  
Image1.Canvas.Rectangle(10,10,100,100);
```

Apa hasilnya? lingkaran tidak muncul. Ini disebabkan setelah lingkaran digambar, lingkaran tersebut “tertimpa” oleh gambar kotak. Delphi menggambar di Canvas sesuai dengan urutan pada kode program.

Arc

Gunakan Arc untuk menggambar lengkungan.

Method Arc membutuhkan enam parameter (X1, Y1, X2, Y2, X3, Y3, X4, Y4).

Empat parameter pertama (x1,y1) dan (x2,y2) sama dengan parameter Method Ellipse yaitu mendefinisikan kotak. Lengkungan digambar mulai dari titik awal ke titik akhir **berlawanan dengan jarum jam**. Titik awal adalah titik potong antara ellips dengan garis yang menghubungkan antara pusat ellips dengan X3,Y3 . Titik akhir adalah titik potong antara ellips dengan garis yang menghubungkan antar pusat lingkaran dengan X4,Y4.

Contoh:

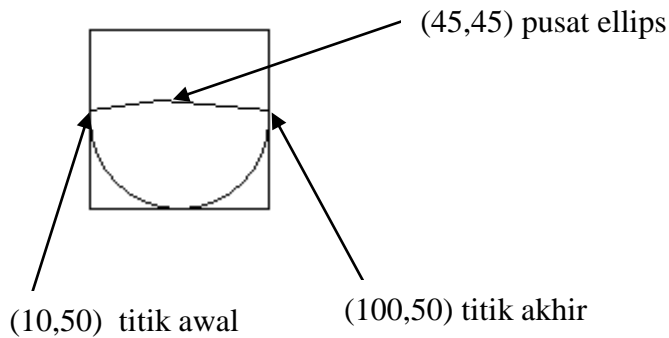
```
with image1.Canvas do
```


```

begin
  Rectangle(10,10,100,100);
  MoveTo(45,45); //titik pusat ellips
  LineTo(10,50); //titik mulai
  MoveTo(45,45); //titik pusat ellips
  LineTo(100,50); //titik akhir
  Arc(10,10,100,100,10,50,100,50); //lengkungan
end;

```

Akan menghasilkan:



Untuk membuat lingkaran yang melengkung cembung  pindahkan titik awal ke sebelah kanan dan titik akhir ke sebelah kiri. Ingat lengkungan digambar dari titik awal ke titik akhir **berlawanan dengan jarum jam**.

RoundRect

Gunakan RoundRect untuk menggambar kotak (seperti rectangle) tetapi dengan ujung-ujung tumpul.

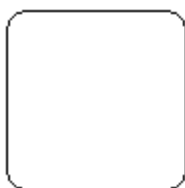
Contoh:

```

image1.Canvas.RoundRect(10,10,100,100,20,20);

```

Akan menghasilkan:



RoundRect menggunakan enam parameter (X1, Y1, X2, Y2, X3, Y3). Empat parameter pertama X1, Y1 dan X2, Y2 sama fungsinya dengan Rectangle, yaitu menyatakan sudut kiri atas dan sudut kanan bawah. Parameter X3, Y3 digunakan untuk mengatur lengkungan sudut. Sudut melengkung pada RoundRect sama dengan lengkungan ellips yang memiliki lebar X3 dan tinggi Y3.

Polygon

Poligon pada prinsipnya sama dengan kombinasi moveto dan lineto. Misalnya kita ingin menggambar sebuah segitiga berwarna merah. Dengan polygon kodenya sebagai berikut:

```
Image1.Canvas.Brush.Color:=clRed;  
Image1.Canvas.Polygon([Point(10,10),Point(100,100),Point(50,100)]);
```

Hasilnya akan muncul sebagai berikut:



Parameter method polygon adalah array bertipe point, oleh karena itu pada contoh diatas digunakan kurung siku.

Mengapa tidak menggunakan MoveTo dan LineTo saja? Alasan pertama, method Polygon lebih ringkas. Alasan kedua, property Brush (color, style dsb) tidak dapat digunakan jika kita menggunakan MoveTo dan LineTo.

Selain method yang sudah dibahas di atas (rectangle, ellipse, arc, roundrect, moveto, lineto, polygon) terdapat beberapa method lain seperti: Chord, FocusRect, Pie, PolyBezier dan Polyline. Silahkan pelajari melalui fasilitas Help pada Delphi.

Brush dan Pen

Brush dan Pen adalah property dari Canvas yang mendefinisikan bagaimana sebuah objek digambar.

Property adalah variabel yang dimiliki sebuah objek. Bedakan dengan method yang merupakan fungsi/prosedur objek.
Jika kita analogikan manusia sebagai objek, maka contoh property untuk objek manusia adalah: nama, umur dsb. Sama dengan method, property juga ditulis dengan notasi titik: **manusia.nama**, **manusia.umur**.

Cobalah lihat kotak dibawah. Garis merah yang membentuk kotak didefinisikan oleh objek Pen. Sedangkan isinya yang berwarna biru didefinisikan oleh objek Brush.



Dalam Delphi, kita tuliskan sebagai berikut:

```
Image1.Canvas.Pen.Color = clRed;  
Image1.Canvas.Brush.Color = clBlue;  
Image1.Canvas.Rectangle(10,10,100,100);
```

Dapat kita lihat bersama bahwa Pen dan Brush memiliki property Color. Ini adalah contoh property bertingkat (*nested*). Objek Image1 memiliki property Canvas, Canvas memiliki property Pen dan akhirnya Pen memiliki property Color.

Jika kode di atas ditambahkan kode untuk menggambar lingkaran.

```
Image1.Canvas.Pen.Color = clRed; //warna pen  
Image1.Canvas.Brush.Color = clBlue; //warna brush  
Image1.Canvas.Rectangle(10,10,100,100);  
Image1.Canvas.Ellipse(200,10,300,100); //tambahkan membuat lingkaran
```

Hasilnya adalah sebagai berikut.



Terlihat bahwa brush dan pen lingkaran juga mengikuti brush dan pen kotak. Artinya sekali diubah, Brush dan Pen berlaku untuk semua proses menggambar yang berlangsung berikutnya.

Contoh berikut akan memperjelas hal ini

```
with image1.Canvas do  
begin  
    Brush.Color:=clRed; //brush warna merah  
    Brush.Style:=bsCross; //bentuknya kotak  
    Pen.Style:=psDot; //pen-nya berbentuk titik-titik  
    Rectangle(10,10,100,100);  
  
    Brush.Style:=bsHorizontal; //bentuk brush garis-garis horizontal  
    Pen.Width:=5; //lebar pen 5 points  
    Pen.Style:=psSolid; //pen-nya berbentuk garis solid  
    Ellipse(200,10,300,100);
```

end;

Hasilnya sebagai berikut:



Ini terjadi karena setelah kotak digambar, brush dan pen diubah sehingga akan langsung berpengaruh terhadap gambar lingkaran.

Brush.Style dapat diisi dengan nilai-nilai:

| Value | Pattern | Value | Pattern |
|-------------|---------|--------------|---------|
| bsSolid | | bsCross | |
| bsClear | | bsDiagCross | |
| bsBDiagonal | | bsHorizontal | |
| bsFDiagonal | | bsVertical | |

Sedangkan Pen.Style dapat diisi dengan nilai-nilai:

psSolid: garis solid.

psDash: garis yang terdiri atas garis putus-putus

psDot: garis yang dibentuk oleh titik.

psDashDot: garis yang dibentuk dari urutan garis dan titik.

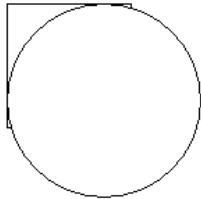
psDashDotDot: Kombinasi garis-titik-titik

psClear: Tidak terlihat (bedakan dengan garis berwarna putih)

Khusus untuk `Brush.Style` yang bernilai `bsClear` dapat kita analogikan sebagai kaca transparan.

Kode berikut akan menghasilkan gambar bertumpuk:

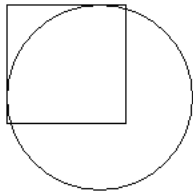
```
Image1.Canvas.Rectangle(10,10,100,100);  
Image1.Canvas.Ellipse(10,10,150,150);
```



Ubah `brush.style` menjadi `bsClear` sebelum menggambar lingkaran

```
Image1.Canvas.Rectangle(10,10,100,100);  
Image1.Canvas.Brush.Style:=bsClear;  
Image1.Canvas.Ellipse(10,10,150,150);
```

maka gambar lingkaran akan transparan



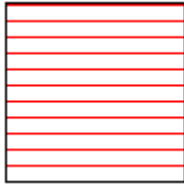
Mewarnai Daerah dengan FloodFill

`FloodFill` dapat digunakan untuk mewarnai suatu daerah yang dibatasi oleh garis. `FloodFill` ini bermanfaat saat `Brush` tidak dapat digunakan, misalnya pada bentuk yang dibuat oleh `MoveTo` dan `LineTo`. `Floodfill` juga dapat dimanfaatkan untuk mewarnai `Canvas`.

Misalkan kita akan mewarnai daerah yang dibentuk oleh sebuah kotak.

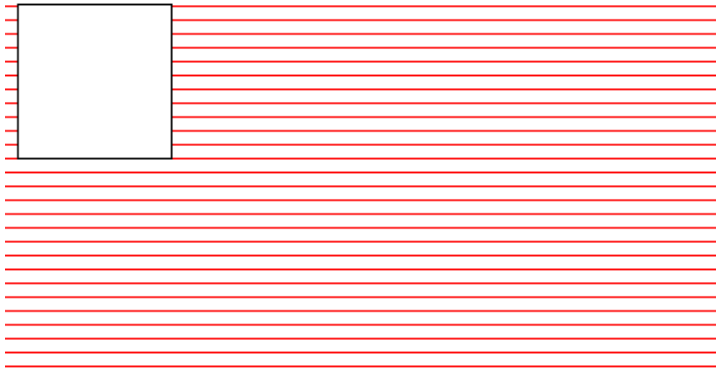
```
with Image1.Canvas do  
begin  
  Rectangle(10,10,100,100);           //buat kotak  
  Brush.Color:=clRed;                 //warna merah  
  Canvas.Brush.Style:=bsHorizontal;  //garis horizontal  
  Canvas.FloodFill(15,15,clBlack,fsBorder); //isi warna dg patokan 15,15  
end;
```

Hasilnya akan muncul sebagai berikut:



Sesuai dengan namanya, FloodFill “membangiri” daerah sesuai dengan Brush yang digunakan. Pada kode `Canvas.FloodFill(15,15,clBlack,fsBorder)`; “banjir” dimulai pada koordinat (15,15) yang terletak didalam kotak, “banjir” warna tersebut akan dibatasi oleh garis warna hitam.

Apa yang terjadi jika kita ubah koordinatnya menjadi (9,9) yang berada di luar lingkaran? Kode `Canvas.FloodFill(9,9,clBlack,fsBorder)`; akan menghasilkan:



Mengapa? karena Floodfill “membangiri” mulai pada posisi 9,9 yang berada diluar kotak dan “terbendung” dengan daerah yang dibatasi garis hitam. sendiri fitur TeeChart yang lain.