

Minggu ke : 13
Penyusun : Yudi Wibisono
Materi : 7.5 Membuat Tulisan
7.6 Pixel
7.7 Mengubah Koordinat Canvas
7.8 Membuat Grafik Fungsi
7.9 Tee Chart

URAIAN POKOK PERKULIAHAN

Membuat Tulisan

Gunakan method `TextOut` untuk menampilkan tulisan pada canvas.

Contoh berikut menampilkan tulisan “Hello World” pada posisi 10,10

```
Image1.Canvas.TextOut(10,10,'Hello World');
```

Untuk memotong tulisan sehingga tidak melewati suatu daerah tertentu, gunakan

```
Canvas.TextRect
```

Contoh:

```
Image1.Canvas.TextRect(Rect(10,10,50,20),10,10,'Hello world');
```

akan menghasilkan (perhatikan kata yang terpotong):

Helld wr

Untuk mengatur jenis dan ukuran, ubahlah property `Canvas.Font`. Contoh:

```
Image1.Canvas.Font.Color:=clBlue;           //warna  
Image1.Canvas.Font.Name:='Arial';           //jenis huruf  
Image1.Canvas.Font.Style:=[fsItalic,fsBold]; //bold dan italic  
Image1.Canvas.Font.Size:=20;                //ukuran font  
Image1.Canvas.TextOut(10,10,'Hello World'); //Arial biru, bold,italic
```

Gunakan method `Canvas.TextWidth` dan `Canvas.TextHeight` untuk mengetahui sebesar apa teks menggunakan tempat di canvas, sehingga kita bisa menyesuaikan posisi gambar lain.

Pixel

Pixel adalah property canvas untuk menggambar titik dilayar (bukan method!).

Kode berikut akan menggambar titik berwarna merah di posisi (10,10)

```
Image1.Canvas.Pixels[10,10]:=clRed;
```

Pixel bermanfaat saat digunakan dalam loop/pengulangan. Contoh berikut menggambarkan garis dengan persamaan $y = x$ untuk $x = 1$ sampai dengan 100

Hasilnya adalah sbb:

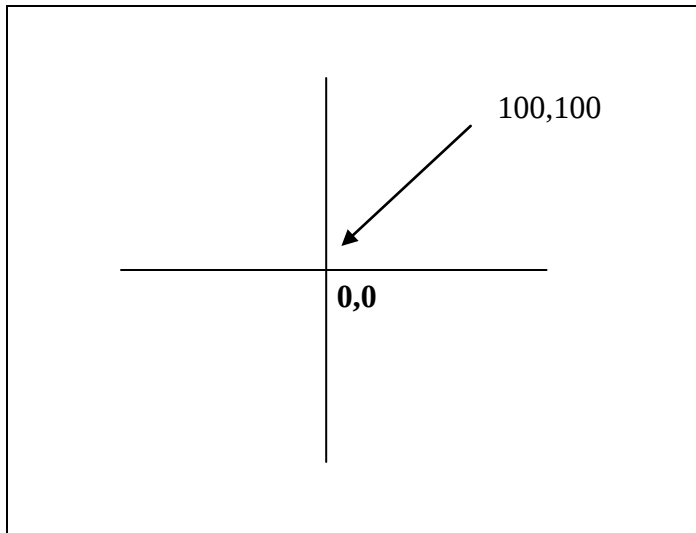
```
var
  i:integer;
begin
  For i:=1 to 100 do
    begin
      Image1.Canvas.Pixels[i,i]:=clRed;
    end;
  end;
```



Mengubah Koordinat Canvas Menjadi Koordinat Empat Kuadran

Telah kita ketahui bersama bahwa koordinat Canvas Delphi hanya memiliki satu kuadran (semua positif), dengan titik (0,0) berada di sudut kiri atas dan nilai sumbu y semakin kebawah semakin besar. Jika kita ingin mem-plot grafik sebuah fungsi, tentu lebih mudah untuk mengubah sistem koordinat menjadi sistem koordinat biasa dengan empat kuadran. Ini dilakukan dengan melakukan pemetaan dari koordinat logik (empat kuadran) ke koordinat fisik (canvas). Untuk lebih jelas, selanjutnya koordinat logik akan ditulis dengan huruf **tebal** dan koordinat fisik ditulis dengan huruf biasa.

Untuk memetakan koordinat logik ke koordinat fisik caranya mudah, pertama tentukan berapa koordinat fisik untuk posisi **0,0** koordinat logik, misalnya 100,100



Sekarang kita buat pemetaan untuk sumbu x. Dari gambar diatas, dapat dilihat bahwa saat kita memasukkan koordinat logik $x = 0$ maka posisi sebenarnya pada canvas adalah 100. Jika koordinat logik 50 maka koordinat canvas adalah 150.

Berdasarkan hal tersebut kita dapat membuat persamaan:

$$\text{KoordinatCanvasX} = \text{KoordinatLojikX} + 100;$$

```
Function CanvasX(LojikX:Double):Integer;
begin
    result:=Round(LojikX + 100);
end;
```

Pada parameter diatas, variabel LojikX dibuat bertipe double, ini memungkinkan kita menggunakan bilangan pecahan untuk koordinat logik. Sedangkan fungsi Round membulatkan ke bilangan terdekat.

Hal yang sama juga berlaku untuk sumbu y. Tapi karena canvas memiliki aturan yang terbalik (semakin ke bawah semakin besar) maka fungsi pemetaan harus disesuaikan.

```
Function CanvasY(LojikY:Double):Integer;
begin
    result:=Round(100 - LojikY);
end;
```

Sekarang contoh penggunaannya:

```
with Image1.canvas do
begin
    MoveTo(CanvasX(-100),CanvasY(0)); //sumbu x
```

```

    LineTo(CanvasX(100),CanvasY(0));

    MoveTo(CanvasX(0),CanvasY(-100)); //sumbu y
    LineTo(CanvasX(0),CanvasY(100));
end;

```

Membuat Grafik Fungsi

Berbekal fungsi pemetaan tadi, kita dapat membuat grafik fungsi sederhana.

Berikut adalah kode untuk membuat grafik persamaan $y = x^2$ mulai dari $x = -100$ sampai dengan $x = 100$.

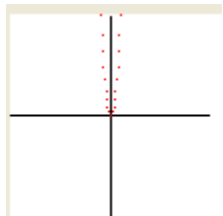
```

with Image1.canvas do
begin
    //gambar sumbu x dan sumbu y sesuai kode diatas

    //plot grafik fungsi
    For i:=-100 to 100 do
    begin
        Pixels[Canvasx(i),Canvasy(i*i)]:=clRed;
    end;
end;

```

Hasilnya akan muncul sebagai berikut:



Perhatikan interval yang besar pada titik-titik grafik. Ini disebabkan x yang kita gunakan adalah bilangan bulat dari -100 sampai dengan 100. Untuk memperhalus gambar, gunakan bilangan pecahan.

Ubah kode di atas menjadi:

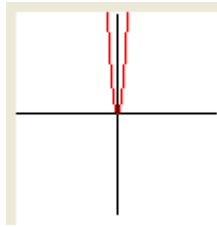
```

with Image1.canvas do
begin
    //gambar sumbu x dan sumbu y sesuai kode diatas

    //plot fungsi grafik
    posx:=-100; //inisialisasi
    while posx <= 100 do
    begin
        Pixels[Canvasx(posx),Canvasy(posx*posx)]:=clRed;
        posx:=posx+0.1; //ditambah 0.1
    end;
end;

```

Hasilnya adalah grafik yang lebih halus.



Untuk latihan cobalah bermain-main dengan mengganti konstanta titik pusat (ganti koordinat 100,100 menjadi koordinat yang lain). Kemudian cobalah membuat zoom in dan zoom out dengan mengalikan koordinat.

Contoh, untuk pemetaan sumbu x, ubah kode menjadi berikut:

```
result:=Round((LojikX + 200) * 0.5); //gambar digeser kekanan & diperkecil
```

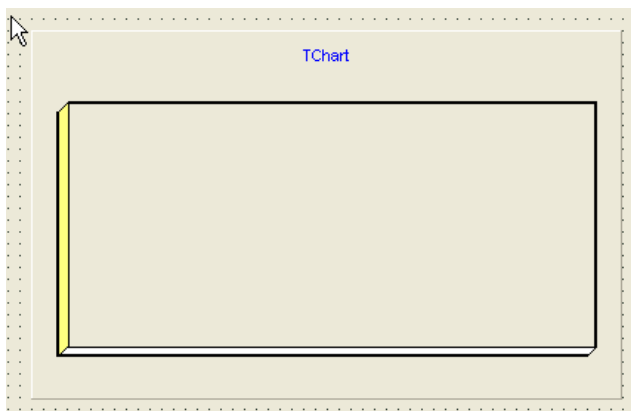
Menggunakan TeeChart

Teechart adalah komponen yang dapat kita gunakan untuk membuat grafik garis, batang, pie-chart, point, gantt dan bubble.

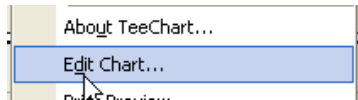
Komponen ini terletak di tab “Additional”



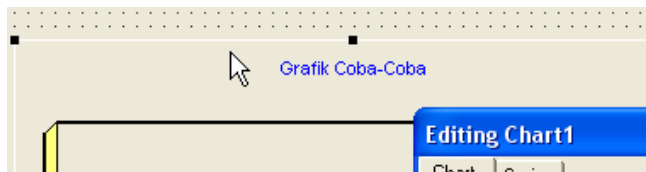
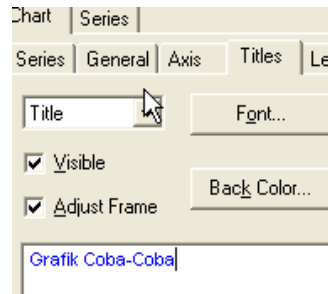
Letakkan komponen ini ke form.



Klik kanan komponen dan pilih

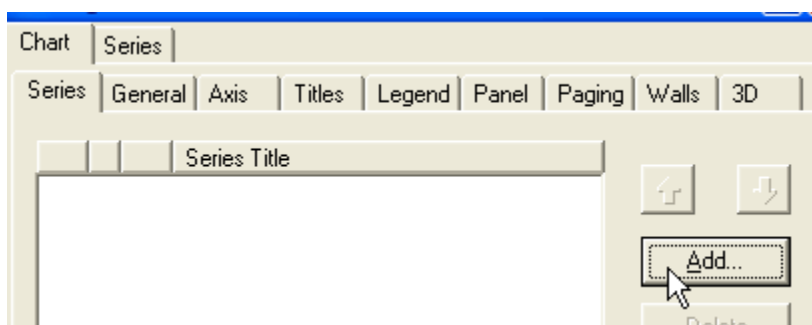


Sekarang kita akan mengganti judul grafik menjadi: “Grafik Coba-Coba”

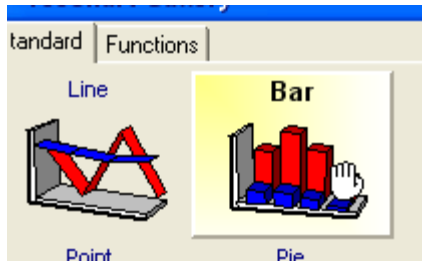


Lihat bahwa komponen pada form akan langsung berubah

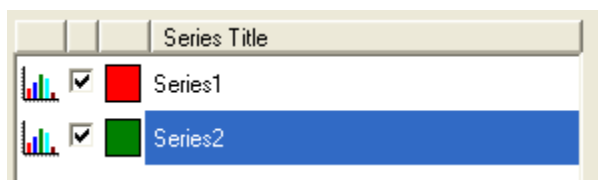
Sekarang tambahkan Series bertipe grafik batang. Pilih tab “Series” dan tekan tombol “Add”



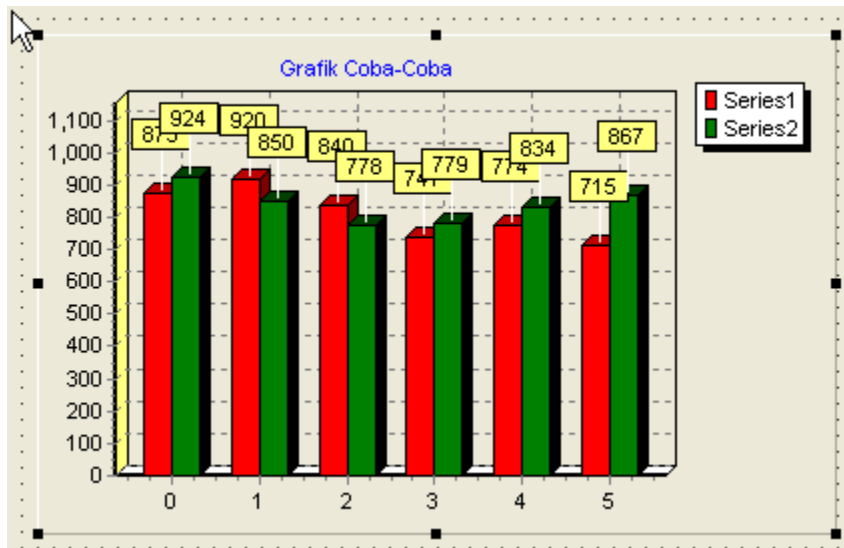
Pilih grafik bertipe “bar”



Tambahkan lagi satu series bertipe “bar”, sehingga terdapat dua series bertipe bar (Anda juga dapat mengkombinasikan beberapa series yang memiliki tipe berbeda)



Terlihat pada form, Delphi menggambar isi chart



Coba jalankan program dengan F9, lihat hasilnya. Kosong bukan? Ini karena grafik yang muncul di form tadi hanya merupakan contoh agar kita dapat melihat “preview” grafik. Selanjutnya kita akan mengisi kedua series ini.

Tambahkan satu button, double klik button ini dan isi kode berikut:

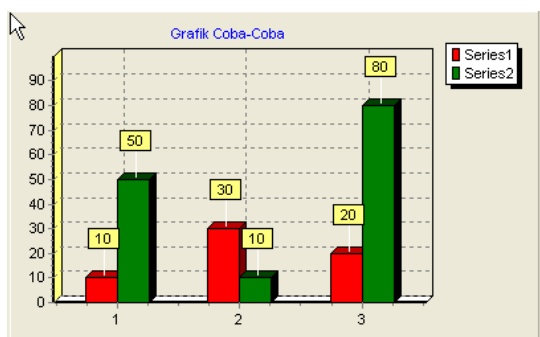
```
procedure TForm1.Button1Click(Sender: TObject);
begin
```

```
//isi series pertama
Chart1.Series[0].AddXY(1,10);
Chart1.Series[0].AddXY(2,30);
Chart1.Series[0].AddXY(3,20);

//isi series kedua, menggunakan with agar lebih singkat
with Chart1.Series[1] do
begin
    AddXY(1,50);
    AddXY(2,10);
    AddXY(3,80);
end;

end;
```

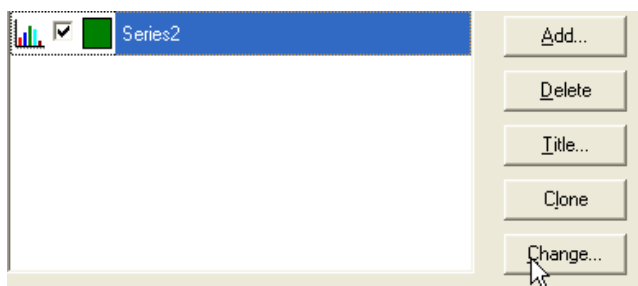
Hasilnya adalah sebagai berikut:



Sangat mudah bukan? Anda dapat menggunakan loop untuk mengisi series.

Sekarang kita akan mengganti jenis series2 menjadi “point” dan mengganti judul “Series1” dan “Series2” menjadi “Murid Kelas 1” dan “Murid Kelas 2”

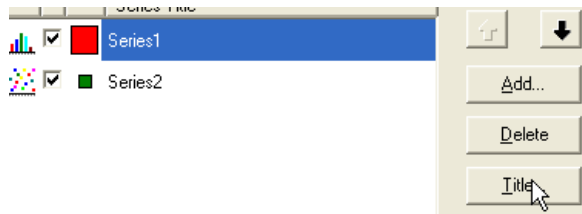
Edit kembali chart dengan klik kanan → Edit Chart



Pilih series2 dan klik “Change”, ganti jenisnya dengan “point”



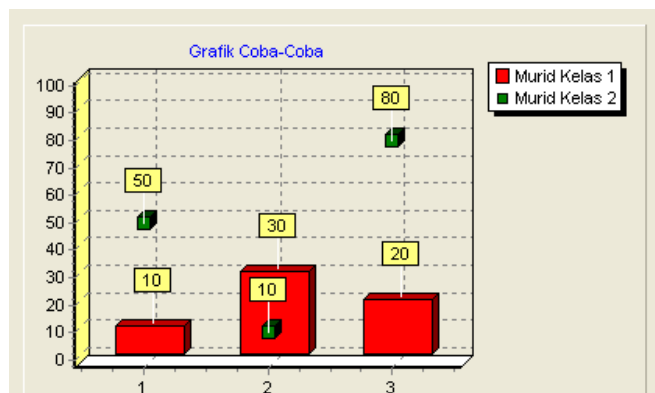
Kemudian untuk mengganti judul, pilih series1, tekan “Title”



Ganti dengan “Series1” dengan “Murid Kelas 1” dan lakukan hal yang sama untuk “Series2” sehingga judul series menjadi sebagai berikut:



Tekan close dan jalankan lagi program, maka akan muncul grafik sebagai berikut:



Demikian sekilas penggunaan komponen Teechart. Materi yang baru disampaikan mengenai TeeChart hanya mencakup sekitar 10% kemampuan TeeChart. Masih banyak lagi yang dapat kita lakukan dengan komponen ini. Silahkan mengeksplorasi