

## Graphical User Interface (GUI)

### Pendahuluan

Unit standar Graph menyediakan suatu pustaka lebih dari 50 buah rutin grafik yang dapat dipergunakan untuk keperluan pembuatan grafik

Untuk membuat grafik dengan fasilitas ini, maka unit standar Graph harus disebutkan dalam program.

Grafik driver menunjukkan *graphics adapter* yang dipergunakan untuk monitor.

Tiap - tiap driver berisi kode dan data yang diperlukan supaya grafik dapat ditampilkan di layar, karena tiap -tiap layar yang menggunakan graphics adapter tertentu mempunyai karakteristik yang berbeda.

### Menggunakan grafik

Untuk memulai menggunakan grafik, gunakan prosedur standar InitGraph :

```
InitGraph(var GraphDriver : Integer;
          var GraphMode : Integer;
          DriverPath      : String);
```

Graph driver merupakan driver yang dipergunakan pada komputer anda.

Graph Driver pada unit standar graph :

Detect =0, CGA =1, MCGA=2, EGA=3, EGA64=4, EGAMono=5, RESERVED=6, HercMono=7, ATT400=8, VGA=9, PC3270=10

Kalau anda menggunakan Color Graphics Adapters(CGA), maka graphics driver yang harus disebutkan adalah CGA atau nilai konstanta 1.

### Berpindah dari mode grafik ke mode teks dan sebaliknya

Dari keadaan mode grafik bila akan berpindah ke mode teks, maka dapat digunakan dengan prosedur standar **RestoreCrtMode** dan untuk kembali dari ke keadaan mode grafik, dapat dilakukan dengan prosedur standar **SetGraphMode**

Contoh:

```
Uses Graph;
```

```
Var
```

```
    DriverGrafik, ModeGrafik : integer;
```

```
Begin
```

```
    DriverGrafik:=Detect;
```

```
    InitGraph(DriverGrafik, ModeGrafik,"");
```

```
    (menampilkan grafik)
```

```
    .
```

```
    .
```

```
    (kembali ke mode teks)
```

```
    RestoreCrtMode;
```

```
    .
```

```
    .
```

```
(kembali lagi ke mode grafik)
setGraphMode(ModeGrafik);
```

End

**Mengakhiri penggunaan grafik**

Untuk mengakhiri penggunaan grafik dan menyebabkan keadaan kembali pada mode layar semula sebelum mode grafik dipergunakan adalah : **CloseGraph**

Contoh:

```
Uses Graph;
```

```
Var
```

```
DriverGrafik, ModeGrafik : integer;
```

```
Begin
```

```
DriverGrafik:=CGA;
```

```
ModeGrafik:=CGAC1;
```

```
InitGraph(DriverGrafik, ModeGrafik,'');
```

```
.
```

```
.
```

```
closeGraph;
```

```
End.
```

**Sistem Koordinat**

Pada mode grafik, ujung kiri atas dari layar (0,0). Nilai dari X atau kolom bergerak ke arah kanan dan nilai Y bergerak ke arah bawah. Jumlah kolom dan baris tergantung dari mode grafik yang dipergunakan untuk tiap-tiap graphics adapter. Misalnya untuk mode grafik CGAC1 mempunyai jumlah kolom sebanyak 320 pixel dan baris sebanyak 200 pixel(300x200). Untuk CGAC2 juga 320x200 dan untuk CGAHi sebanyak 640x200, fungsi standar GetMaxX dapat digunakan untuk mendapatkan nilai jumlah dari kolom dan fungsi GetMaxY untuk mendapatkan jumlah dari baris.



**Pengaturan warna**

4 hal yang perlu diketahui tentang pengaturan warna di mode grafik :

1. Warna yang tersedia adalah sebanyak 16 warna di mode grafik yaitu : Black=0, Blue=1, Green=2, Cyan=3, Red=4, Magenta=5, Brown=6, LightGray=7, DarkGray=8, LightBlue=9, LightGreen=10, LightCyan=11, LightRed=12, LightMagenta=13, Yellow=14, White=15

Kalau anda melukis gambar dengan menggunakan cat air, maka warna-warna ini dapat diibaratkan dengan cat-cat air yang masih ada dikotaknya dan dapat dipilih untuk dipergunakan.

2. Pallette, merupakan warna - warna yang dipilih dari seluruh 16 warna yang tersedia.

Kalau anda menggunakan mode grafik CGAC1, maka hanya 4 macam warna saja yang dapat dipergunakan dalam satu saat sebagai pallette, defaultnya adalah warna Black, Cyan, Magenta dan White. Demikian juga dengan mode grafik CGAC2, hanya 4 macam warna yang dapat dipergunakan sebagai pallette, dengan default *Black, Green, Red, Yellow*. Untuk mode grafik EGA atau VGA warna default di pallette dapat dirubah dengan menggunakan prosedur standar *SetPalette* atau *SetAllPalette*. (untuk mode grafik CGA yang dapat dirubah hanya warna palette ke 0 saja, yaitu warna background).

3. Warna Background, merupakan warna dasar layar. Default dari warna background adalah hitam. Warna background layar dapat diubah dengan warna yang tersedia menggunakan Prosedur standar : **SetBkColor**. Misalnya **SetBkColor(0)** akan membuat warna background layar adalah warna hitam, **SetBkColor(1)** akan membuat warna background layar adalah warna biru dan seterusnya.
4. Warna penggambaran grafik merupakan warna yang diambil dari warna-warna yang tersedia di pallette dan dapat ditentukan dengan Prosedur standar : **SetColor(warna)**. Misalnya pada mode grafik CGAC1 dengan warna pallette Black, Cyan, Magenta dan White, maka **SetColor(0)** akan menggambar grafik dengan grafik dengan warna Black, **SetColor(1)** warna Cyan, **SetColor(2)** warna Magenta dan **SetColor(3)** warna White. Default untuk mode grafik CGAC1 adalah warna White. Untuk mode grafik CGAC2 dengan warna pallette Black, Green, Red dan Yellow, maka **SetColor(0)** akan menggambar grafik dengan warna Black, **SetColor(1)** warna Green, **SetColor(2)** warna Red dan **SetColor(3)** warna Yellow. Default untuk mode grafik CGAC2 adalah warna Yellow.

### Membuat garis

Membuat garis dari koordinat (X1,Y1) sampai dengan (X2,Y2) :

**Line(X1,Y1,X2,Y2:integer);**

Untuk membuat garis dari posisi yang terakhir kali sampai di koordinat tertentu :

**LineTo(X,Y:integer);**

Menambah garis secara relatif dari titik terakhir kali:

**LineRel(Dx,Dy : integer);**

Menuju suatu titik tertentu tanpa menggambarinya:

**MoveTo(X,Y :integer);**

Contoh:

Line(70,50,100,20)

Dapat juga ditulis dengan:

MoveTo(70,50); LineTo(100,20)

Uses Graph;

Var

DriverGrafik, ModeGrafik : integer;

KodeSalah : Integer;

Begin

DriverGrafik:=CGA;

ModeGrafik:=CGAC1:

InitGraph(DriverGrafik, ModeGrafik,"");

Line(70,80,130,80)

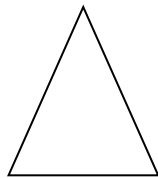
MoveTo(130,80)

LineTo(100,20)

LineRel(-30,60)

End.

Bila Program dijalankan akan didapatkan gambar Segitiga:



### **Membuat kotak**

Untuk membuat suatu gambar kotak sebenarnya dapat dilakukan dengan 4 buah garis dengan prosedur standar Line atau LineTo atau LineRel.

Prosedur standar yang digunakan untuk membuat kotak adalah : **Rectangle(x1,y1,x2,y2 : Integer);**

Contoh:

Uses Graph;

Var

DriverGrafik, ModeGrafik : integer;

KodeSalah : Integer;

Begin

DriverGrafik:=CGA;

ModeGrafik:=CGAC1:

InitGraph(DriverGrafik, ModeGrafik,"");

SetColor(1);

Rectangle(10,65,85,125);

End.

### **Membuat Diagram Batang**

Prosedur yang dapat digunakan untuk membuat diagram batang, yaitu prosedur standar Bar dan prosedur standar Bar 3D.

Prosedur Bar hampir sama dengan prosedur standar Rectangle, yaitu membuat kotak, hanya bedanya pada prosedur standar Bar ini, kotak yang dihasilkan sudah diberi arsir warna.

Bilamana batang yang diinginkan dalam bentuk 3 dimensi, maka dapat dipergunakan prosedur standar Bar3D.

Prosedur standar yang digunakan untuk membuat diagram batang adalah :

**Bar(x1,y1,x2,y2 : Integer);**

dan

**Bar3D(x1,y1,x2,y2 : integer; Depth :word;  
Top : boolean);**

### **Mewarnai Obyek**

Prosedur standar yang digunakan untuk mewarnai obyek adalah :

**FloodFill(X,Y : integer ;Border:word);**

X dan Y adalah titik koordinat sembarang yang berada didalam area obyek. Bila X dan Y berada diluar area obyek gambar, maka yang akan dicat dengan warna atau pattern adalah daerah luar gambarnya.

Warna yang digunakan oleh prosedur standar FloodFill ini adalah warna default, yaitu untuk mode grafik CGAC1 berwarna putih(White) dan untuk mode grafik CGAC2 berwarna kuning(Yellow)

Untuk mewarnai dengan arsir pada pembuatan diagram batang(dengan prosedur standar Bar dan Bar3D) dan pembuatan bagan pastel(dengan prosedur PieSlice) serta untuk mewarnai obyek lainnya, bentuk pengarsiran, dirubah dengan menggunakan prosedur standar :

**SetFillStyle(Pattern : word; Color:word);**

### **Membuat Tulisan**

Prosedur standar yang khusus digunakan untuk menampilkan teks atau tulisan di mode grafik dengan suatu font yang tertentu, yaitu prosedur standar OutText dan OutTextXY

Prosedur standar OutText untuk menampilkan tulisan mulai pada posisi koordinat yang terakhir adalah :

**OutText(TextString : string);**

Sedang prosedur standar OutTextXY digunakan untuk menampilkan teks di posisi koordinat yang tertentu dengan sintak:

**OutTextXY(X,Y:integer; TextString:string);**

Kelebihan lain dari penggunaan prosedur standar OutText dan OutTextXY adalah dapat dipergunakannya beberapa bentuk huruf atau font, arah dari tulisan dapat diatur(kearah horisontal atau kearah vertikal) serta besarnya tulisan juga dapat ditentukan. Pemilihan

font, pengaturan arah tulisan dan besarnya tulisan dapat diatur ditentukan dengan menggunakan prosedur standar `SetTextStyle`

Menentukan jenis font :

```
SetTextStyle(Font,Direction : word;  
CharSize : word)
```

### **Membuat Titik**

Prosedur standar yang digunakan untuk membuat titik adalah :

```
PutPixel(X,Y : integer ; Pixel : word);
```

(X,Y) adalah posisi koordinat layar dan pixel adalah warna dari titik yang diinginkan

Untuk mengetahui warna dari suatu titik pada suatu koordinat tertentu :

```
GetPixel(X,Y : integer);
```

### **Membuat Lingkaran**

Prosedur standar yang digunakan untuk membuat lingkaran adalah :

```
Circle(X,Y : integer ; Radius : word);
```

(X,Y) adalah posisi koordinat titik pusat lingkaran dan radius adalah jari-jari lingkarannya.

### **Membuat Busur**

Prosedur standar yang digunakan untuk membuat busur adalah :

```
Arc(X,Y : integer ; StAngle,EndAngle, Radius : word);
```

(X,Y) adalah koordinat titik pusat dari busur. `StAngle` adalah sudut awal busur dimulai dari `EndAngle` adalah sudut akhir dari busur. `StAngle` dan `EndAngle` dalam suatu derajat(dari 0 sampai dengan 360). Bila `StAngle` adalah 0 dan `EndAngle` adalah 360, maka berarti akan dibuat lingkaran penuh, sehingga sama dengan prosedur standar `Circle`. Radius adalah jari-jari.

Contoh:

```
Uses Graph;
```

```
Var
```

```
    DriverGrafik, ModeGrafik : integer;
```

```
    KodeSalah : Integer;
```

```
Begin
```

```
    DriverGrafik:=CGA;
```

```
    ModeGrafik:=CGAC1;
```

```
    InitGraph(DriverGrafik, ModeGrafik,"");
```

```
    Circle(160,100,50);
```

```
    Circle(140,85,10);           (Mata Kiri)
```

```
    Circle(180,85,10);         (mata Kanan)
```

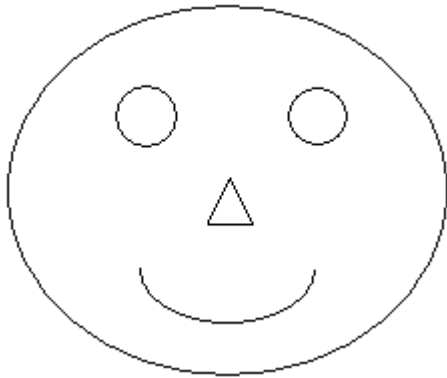
```
    Line(160,95,165,105);
```

```
    Line(165,105,155,105);
```

```

Line(155,105,160,95);      (Hidung)
Arc(160,110,200,340,25);  (Mulut)
End.
    
```

Bila program ini dijalankan, akan didapatkan :



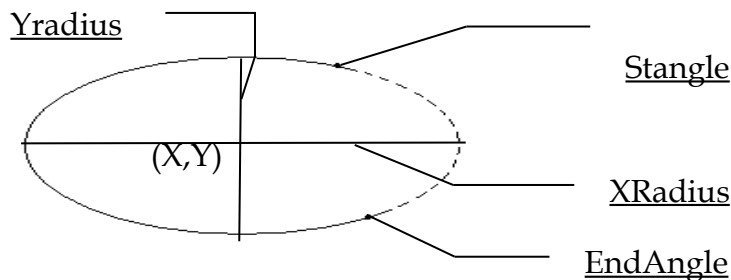
**Membuat ellipse**

Prosedur standar yang digunakan untuk membuat ellipse adalah :

```

Ellipse(X,Y : integer ; StAngle,EndAngle:word;
XRadius, YRadius : word);
    
```

X,Y adalah koordinat titik pusat ellipse, StAngle dan EndAngle adalah sudut awal dan akhir dari busur ellipse dengan satuan derajat(dari 0 sampai dengan 360). Xradius adalah jari-jari ellipse arah horisontal dan Yradius adalah jari-jari ellipse arah vertikal.



**Membuat Pie Chart**

Prosedur standar yang digunakan untuk membuat pie chart adalah :

```

PieSlice(X,Y : integer ; StAngle,EndAngle, Radius : word);
    
```

X,Y adalah koordinat titik pusat *pie chart*, StAngle dan EndAngle adalah sudut awal dan akhir dari busur *Pie Chart* dengan satuan derajat(dari 0 sampai dengan 359). Radius adalah jari-jari dari *Pie Chart*.

**Membuat Poligon**

Prosedur standar yang digunakan untuk membuat poligon (segi banyak) adalah :

```

DrawPoly(NumPoint : word; var PolyPoints);
    
```

NumPoint adalah jumlah dari koordinat titik poligon. PolyPoints adalah variabel larik yang berisi titik koordinat poligon dengan tipe PointType yang sudah didefinisikan sebagai berikut:

Type

```
PointType = record
    X,Y:Integer;
End
```

Contoh:

Uses Graph;

Var

```
DriverGrafik, ModeGrafik : integer;
```

Const

```
Poligon : Array[1..6] of PointType =
    ((X: 25; Y:50),
    (X: 50; Y:100),
    (X: 75; Y:60),
    (X: 100; Y:90),
    (X: 125; Y:30),
    (X: 150; Y:75));
```

Begin

```
DriverGrafik:=CGA;
ModeGrafik : CGAC1;
InitGraph(DriverGrafik, ModeGrafik,"");
```

```
DrawPoly(6,Poligon);
```

End.

Bila Program dijalankan, akan didapatkan hasil:



### **Membuat View**

Suatu jendela atau kotak pandangan(view) dapat dibuat di mode grafik dengan prosedur standar SetViewPort dengan sintak:

```
SetViewPort(x1,y1,x2,y2 : integer ; Clip : boolean);
```

X1,y1 adalah koordinat ujung kiri atas dari kotak view dan x2,y2 adalah koordinat ujung kanan bawah dari kotak view. Clip digunakan untuk menentukan apakah penggambaran

grafik didalam kotak view dapat melewati batas kotaknya atau tidak. Dua buah konstanta telah didefinisikan untuk clip yaitu:

```
Clipoff = false;  
Clipon = true;
```

Untuk nilai clip adalah clipoff, bila terjadi penggambaran dikotak view yang melebihi batas kotaknya, penggambaran akan terus dilakukan diluar kotak view menabrak batas kotaknya. Sebaliknya bila bernilai Clipon, bila penggambaran melebihi batas kotak view, maka akan dihentikan atau dipotong (clip) sampai dengan batas kotaknya.