DASAR PEMPROGRAMAN GRAFIK PADA BAHASA C

Denny Charter
Irma Agtrisari
denny_charter@telkom.net / agtree80@yahoo.com
http://www.dennycharter.multiply.com / http://www.agtree.wordpress.com

Lisensi Dokumen:
Copyright © 2003-2008 IlmuKomputer.Com
Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Gambar bermakna sejuta kata. Kalimat tersebut tidak berlebihan, karena dengan gambar kita dapat memperlihatkan sesuatu tanpa perlu menjelaskan, dan orang pun langsung mengerti. Pada bagian ini, kita akan membahas mengenai dasar-dasar pemrograman grafik menggunakan bahasa C.

Ada dua syarat utama untuk memulai pemrograman grafik, yaitu meng-*include*-kan file header **<graphics.h>** dan melakukan proses inisialisasi mode grafik yang digunakan oleh komputer. Untuk inisialisasi mode grafik digunakan perintah **initgraph()** yang ada di file header graphics.h. Format umum dari perintah ini adalah :

```
initgraph(graphdriver, graphmode, direktori_bgi)
```

Variabel graphdriver digunakan untuk menyimpan nilai driver grafis yang ditentukan berdasarkan kartu grafik yang digunakan. Variabel graphmode digunakan untuk menyimpan nilai mode grafis, sedangkan direktori_bgi adalah tempat file .bgi yang digunakan dalam pemrograman grafik. Contoh listing program berikut ini akan menggambarkan persegi panjang dengan diagonal yang menutupi layar.

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int Xmax, Ymax;
int GraphDriver,GraphMode,ErrorCode;
main()
{
      GraphDriver = DETECT;
      initgraph(      &GraphDriver,      &GraphMode,
"c:\\tc" );
      ErrorCode = graphresult();
      if( ErrorCode != grOk ){
          printf(" Graphics System Error: %s\n",
                    grapherrormsg( ErrorCode ) );
          exit( 1 );
       }
       Xmax = getmaxx();
       Ymax = getmaxy();
       rectangle(0,0,Xmax,Ymax);
       line(0,0,Xmax,Ymax);
       line(0,Ymax,Xmax,0);
       getch();
       closegraph();
       return 0;
}
```

Program diatas dimulai dengan inisialisasi variabel **`GraphDriver = DETECT`**. Perintah ini berarti program meminta *autodeteksi,* sehingga Turbo C secara langsung akan mencocokkan driver dan mode grafik yang terdeteksi.

Proses berikutnya adalah inisialisasi mode grafik dengan perintah :

`initgraph( &GraphDriver, &GraphMode, "c:\\tc" );`

melalui perintah ini Turbo C akan menginisialisasi kartu grafik yang telah terdeteksi, kemudian kompiler akan menemukan dan memasukkan *graphic-driver* (file dengan akhiran **.bgi**). Begitu nama driver ditemukan, maka driver akan menangani rincian antara program grafik dengan perangkat keras.

Jika ternyata driver grafik tidak dapat ditemukan, maka initgraph tidak akan dapat membuat grafik apapun. Karena itu perintah **`ErrorCode = graphresult();`** digunakan untuk mengembalikan nilai inisialisasi grafik. Jika nilai yang dikembalikan 0 (grOK) maka kita dapat memulai program grafik. Jika tidak, maka kita harus memuat listing berikut untuk mendeteksi

Copyright © 2003-2007 IlmuKomputer.Com

kesalahan pada inisialisasi grafik.

```
if( ErrorCode != grOk ){
      printf(" Graphics System Error: %s\n",
                  grapherrormsg( ErrorCode )
);
      exit( 1 );
}
```

Setelah proses inisialisasi berhasil dilakukan, maka kita dapat mulai membuat grafik yang diinginkan. Pixel pada layar dinomori mulai dari sudut kiri atas layar (**0, 0**) sampai dengan sudut kanan bawah (**getmaxx(), getmaxy**).

Perintah untuk menggambar persegi panjang adalah **`rectangle(x1,y1,x2,y2);`** dimana x1 dan y1 adalah koordinat sudut kiri atas dari persegi panjang, dan x2, y2 adalah sudut kanan bawah dari persegi panjang.

Sedangkan perintah **`line(x1,y1,x2,y2);`** digunakan untuk menggambar garis dari koordinat (x1,y1) sampai koordinat (x2,y2).

Jika program selesai dilakukan maka kita harus memanggil fungsi **`closegraph();`** untuk menutup mode grafik.

FUNGSI-FUNGSI GAMBAR PADA TURBO C

Dengan menggunakan format program seperti contoh sebelumnya, kita dapat membuat gambar-gambar lain menggunakan fungsi-fungsi gambar yang disediakan oleh turbo C, diantaranya adalah :

- garis

Copyright © 2003-2007 IlmuKomputer.Com

Untuk menggambar garis digunakan fungsi **line(x1,y1,x2,y2)** yang berarti penggambaran garis dimulai dari koordinat awal (x1, y1) sampai koordinat akhir (x2,y2). Perintah lain untuk menggambar garis adalah **lineto(x,y)** dan **linerel(dx,dy)**.

Untuk menggunakan kedua perintah tersebut, kita harus mengerti terlebih dahulu konsep CP (*Current Position*) yang berarti posisi saat ini. Pada saat pertama kali memanggil fungsi **lineto(x,y)** dan **linerel(dx,dy)**CP terletak pada posisi koordinat (0,0) sehingga garis akan digambar dari posisi (0,0) sampai posisi (x,y) pada perintah lineto() dan posisi (dx,dy) pada perintah linerel(), selanjutnya CP akan diperbaharui pada titik akhir penggambaran garis.

Pada awal pemanggilan kedua perintah tersebut memang memiliki persamaan, yaitu menggambar garis dari posisi (0,0) sampai ke posisi pada parameter perintah. Perbedaanya adalah saat CP diperbaharui pada posisi akhir penggambaran garis (xCP,yCP). Jika kita menggunakan fungsi **lineto(x,y)**maka garis akan digambar dari (xCP,yCP) sampai koordinat (x,y), kemudian CP akan diperbaharui pada posisi (x,y). Pada penggunaan perintah **linerel(dx,dy)**, garis akan digambarkan dari koordinat (xCP,yCP) sampai koordinat (xCP+dx , yCP+dy).

Turbo C juga menyediakan bentuk-bentuk garis yang dapat diatur dengan perintah **setlinestyle(tipe_garis, pola_garis, tebal_garis).** Nilai untuk tipe garis diberikan dalam tabel berikut ini :

Nama Garis	Nomor Garis	Gambar
SOLID_LINE	0	
DOTTED_LINE	1	
CENTER_LINE	2	
DASHED_LINE	3	

Copyright © 2003-2007 IlmuKomputer.Com

Nilai untuk pola garis diabaikan sehingga kita dapat mengisikan nilai nol (0). Sedangkan ketebalan garis memiliki pilihan nilai sebagai berikut :

KETEBALAN	NILAI	KETERANGAN
NORM_WIDTH	1	Ketebalan 1 pixel
THICK_WIDTH	3	Ketebalan 3 pixel

Penggunaan perintah ini dapat dilakukan dengan langsung mengisikan tipe garis, atau dengan nomornya saja. Jadi perintah :

Setlinestyle(center_line,0,thick_width) sama dengan

Setlinestyle(2,0,3)

- segi empat

  Untuk menggambar segi empat, digunakan perintah **rectangle(x1,y1,x2,y2),** dimana x1 dan y1 adalah koordinat sudut kiri atas dari persegi panjang, dan x2, y2 adalah sudut kanan bawah dari persegi panjang. Perintah lain untuk menggambar segiempat adalah **bar(x1,y1,x2,y2)**. Perbedaan kedua perintah ini terletak pada pewarnaan (akan dijelaskan pada bagian warna grafik)

- lingkaran

  Untuk menggambar lingkaran digunakan perintah **circle(x,y,radius)** dimana (x,y) adalah koordinat titik pusat lingkaran, sedangkan radius adalah jari-jari lingkaran.

- Elips

  Untuk menggambar elips digunakan perintah **ellipse(x,y,alpha,beta,rx,ry)** dimana (x,y) adalah koordinat titik pusat elips, alpha adalah sudut awal ellipse, beta adalah sudut akhir ellipse, rx adalah jari-jari ellipse yang sejajar dengan sumbu x, ry adalah jari-jari ellipse yang sejajar dengan sumbu y.

  Ellips akan digambarkan tertutup atau terbuka sesuai dengan nilai alpha dan beta-nya, jika ellips ingin digambar tertutup maka alpha harus diisi dengan 0, dan beta harus diisi dengan 360. perintah ellipse() juga dapat digunakan untuk menggambar lingkaran yaitu dengan mengisi nilai alpha = 0, beta = 360 dan rx =

Copyright © 2003-2007 IlmuKomputer.Com

ry.

Jika ingin menggambarkan ellipse dengan pola isian warna, digunakan perintah **fillellipse(x,y,rx,ry)** dimana (x,y) adalah koordinat titik pusat ellipse, rx jari-jari terpanjang ellipse, dan ry jari-jari terpendek ellipse. Parameter alpha dan beta memang tidak terdapat pada perintah ini, karena ellipse akan diisi pola isian warna maka ellipse harus digambar tertutup dengan kata lain nilai alpha selalu 0 dan beta selalu 360 sehingga tidak perlu dituliskan. Pola isian warna akan dibahas berikutnya.

Contoh program menggunakan fungsi gambar diberikan pada listing berikut ini :

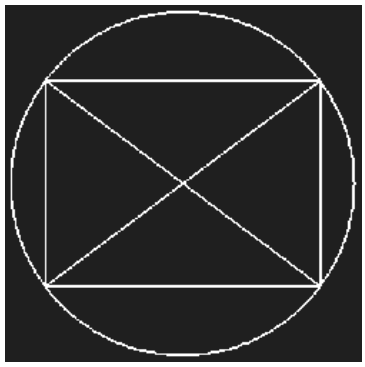
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
main()
{
      int x,y;
      int GraphDriver,GraphMode,ErrorCode;

      GraphDriver = DETECT;
      initgraph(     &GraphDriver,     &GraphMode,
"c:\\tc" );
      ErrorCode = graphresult();
      if( ErrorCode != grOk ){
          printf(" Graphics System Error: %s\n",
                   grapherrormsg( ErrorCode ) );
          exit( 1 );
       }

       x=getmaxx();
       y=getmaxy();
       rectangle(x/4,y/4,3*x/4,3*y/4);
       line(x/4,y/4,3*x/4,3*y/4);
       line(3*x/4,y/4,x/4,3*y/4);
       circle(x/2,y/2,200);
       getch();
       closegraph();
       return 0;
}
```

Hasil eksekusi program diatas ditunjukkan oleh gambar berikut :

Copyright © 2003-2007 IlmuKomputer.Com



WARNA DAN POLA PADA GRAFIK

Warna grafik yang disediakan turbo C pada file header graphics.h adalah :

Kode Warna	Keterangan
0	BLACK
1	BLUE
2	GREEN
3	CYAN
4	RED
5	MAGENTA
6	BROWN
7	LIGHTGRAY
8	DARKGRAY
9	LIGHTBLUE
10	LIGHTGREEN
11	LIGHTCYAN
12	LIGHTRED
13	LIGHTMAGENTA
14	YELLOW
15	WHITE

Perintah-perintah yang berhubungan dengan pola dan warna diantaranya :

1. setcolor()

   Perintah setcolor() berfungsi untuk menentukan warna gambar yang aktif. Sintaks dari perintah ini adalah **setcolor(nomor_warna).** Contoh, jika kita menuliskan perintah: **setcolor(1);**

   rectangle(50,50,120,100);

   berarti kita menggambarkan segiempat dengan warna border

Copyright © 2003-2007 IlmuKomputer.Com

biru

2. setbkcolor()

   Perintah setbkcolor() berfungsi untuk mengatur warna background layar. Sintaks dari perintah ini adalah **`setbkcolor(nomor_warna)`**. Contoh, untuk membuat background berwarna kuning digunakan perintah `setbkcolor(14)`.

3. floodfill()

   Sintaks dari perintah ini adalah **`floodfill(x, y, warna_border)`**. Perintah floodfill() berfungsi untuk mengisi suatu wilayah dari posisi (x,y) sampai warna border yang sudah ditentukan. Contohnya, perintah `floodfill(100,50,4)`digunakan untuk mengisi warna dari posisi koordinat (100,50) sampai ditemukan warna border merah.

4. setfillstyle()

   Perintah setfillstyle() berfungsi untuk mengatur warna dan pola isian. Sintaks dari perintah ini adalah **`setfillstyle(nomor_pola, nomor_warna)`**. Jenis-jenis pola isian diberikan oleh tabel berikut :

Nama Pola	Nomor Pola	Gambar
EMPTY_FILL	0	
SOLID_FILL	1	
LINE_FILL	2	
LTSLASH_FILL	3	

Copyright © 2003-2007 IlmuKomputer.Com

SLASH_FILL	**4**	
BKSLASH_FILL	**5**	
LTBKSLASH_FILL	**6**	
HATCH_FILL	**7**	
XHATCH_FILL	**8**	
INTERLEAVE_FILL	**9**	
WIDE_DOT_FILL	**10**	
CLOSE_DOT_FILL	**11**	

TEKS DALAM MODE GRAFIK

Untuk menampilkan teks dalam mode grafis digunakan perintah **`outextxy(x,y,str)`**, dimana (x,y) adalah koordinat awal penulisan teks dan str adalah teks yang akan dituliskan. Fungsi outtextxy akan selalu menuliskan teks berdasarkan koordinat tetapi tidak memperbaharui CP (*Current Position*)

Copyright © 2003-2007 IlmuKomputer.Com

Fungsi lain untuk menuliskan teks adalah perintah **outtext(str)**. Fungsi outtext() akan menuliskan teks dari posisi(0,0) saat pemanggilan pertama, kemudian memperbaharui CP pada pemanggilan berikutnya berdasarkan posisi CP sebelumnya.

Untuk mengatur jenis dan ukuran font, digunakan perintah settextstyle(). Sintaks dari perintah ini adalah:

```
settextstyle(jenis, arah, ukuran)
```

Jenis-jenis font yang dapat digunakan dalam bahasa C diberikan dalam tabel berikut ini :

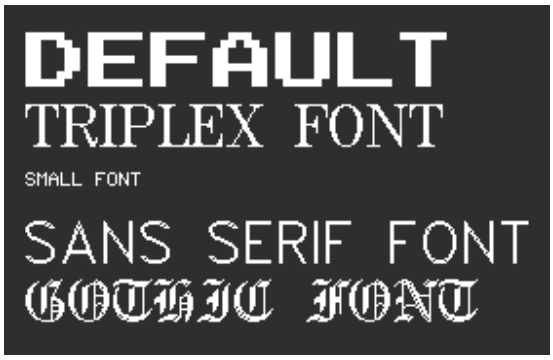
Nama Font	Nilai	Keterangan
DEFAULT_FONT	0	Font 8x8 bit-map
TRIPLEX_FONT	1	Font triplex
SMALL_FONT	2	Small font
SANS_SERIF_FONT	3	Font sans serif
GOTHIC_FONT	4	Font gothic

Arah penulisan tabel terdiri dari dua jenis yaitu dari kiri ke kanan atau dari atas ke bawah. Parameter arah diberikan dalam tabel berikut ini :

Direction	Nilai	Keterangan
HORIZ_DIR	0	Dari kiri ke kanan
VERT_DIR	1	Dari atas ke bawah

Sedangkan ukuran font ditentukan dari nilai 1 sampai 10. Angka 1 menggambarkan teks dengan ukuran huruf 8x8 pixel, 2 adalah 16x16 pixel demikian seterusnya.Contoh bentuk dari jenis-jenis font diatas dapat dilihat berikut ini :

Copyright © 2003-2007 IlmuKomputer.Com



Berikut ini akan diberikan contoh listing program yang menampilkan teks dan grafik :

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int x,y;
int GraphDriver,GraphMode,ErrorCode;
main()
{
      GraphDriver = DETECT;
      initgraph(      &GraphDriver,      &GraphMode,
"c:\\tc" );
      ErrorCode = graphresult();
      if( ErrorCode != grOk ){
          printf(" Graphics System Error: %s\n",
                   grapherrormsg( ErrorCode ) );
          exit( 1 );
       }
       settextstyle(3,0,4);
       outtextxy(25,50,"HAI FRIEND,");
       outtextxy(25, 100,"NAMA SAYA MR. SMILE");

       x = getmaxx()/2;
       y = getmaxy()/2;

       bar(x-75,y-50,x+75,y+100);
       setcolor(0);
       setfillstyle(9,14);
       fillellipse(x,y+25,60,60);

       setfillstyle(10,0);
       fillellipse(x-25,y+10,5,5);
       fillellipse(x+25,y+10,5,5);

       setcolor(0);
       setlinestyle(0,0,3);
       ellipse(x,y+40,180,360,20,10);
```


Copyright © 2003-2007 IlmuKomputer.Com

```
        getch();
        closegraph();
        return 0;
}
```

Tentang Penulis

DENNY CHARTER (www.dennycharter.wordpress.com)
IRMA AGTRISARI (www.agtree.wordpress.com)

Copyright © 2003-2007 IlmuKomputer.Com