

Pemanfaatan Sintaks *OpenGL* dalam Menggambar Obyek Dua Dimensi

Jackri Hendrik¹⁾, Tintin Chandra²⁾

Teknik Informatika, STMIK TIME Medan¹⁾ Teknik Informatika, STMIK IBBI Medan²⁾

Jalan. Merbabu No. 32 AA-BB Medan¹⁾, Jalan. Jend. Gatot Subroto No. 130 Medan²⁾

Telp. 061-4561932¹⁾ Telp. 061-4519070²⁾

E-mail: jackrihendrik@stmik-time.ac.id¹⁾, E-mail : tinuhnsbm@gmail.com²⁾

Abstract

Next to nothing area none which get out of requirement of graph. Although initially application in science and engineering need equipments which costly enough. With picture prepare will make easier in reading information and also data compared to by reading data manually in a communications of visual. Which one programming of graphical desain is OpenGL. OpenGL accepted to become one of the standard in computer graphic and in this time have implementation in so many computer system.

Early graphical programming is turned into text mode is graphical mode by exploiting function of VGA card. Portability of OpenGL as corps of library conducive to be accessed to through language of programming by figuring in OpenGL (library in directory of library language. In making of object two dimension with OpenGL cannot be direct by calling command which have been provided in library, but require to be done by setting used programming. At programming base on done easier visual especially at Visual Microsoft C++ 2005, where there is file of glu. * placed to 3 folder that is in gl folder include, library, and system32.*

For coloration at object at OpenGL differ from result of yielded graphic of programming of PACK, because at OpenGL exploit colour of RGB (Red, Green, Blue), also can be made in the form of colour gradation.

Keyword: Object OpenGL, Syntax OpenGL, Object 2D.

1. Pendahuluan

Grafika komputer telah menunjukkan ke-majuan yang pesat dalam pengembangan ber-bagai aplikasi untuk menghasilkan gambar. Hampir tidak ada bidang satupun yang lepas dari kebutuhan grafik. Walaupun pada awalnya aplikasi dalam *engineering* dan sains memerlukan peralatan yang cukup mahal, namun dengan perkembangan teknologi komputer seka-rang ini memberikan suatu kemudahan peng-gunaan komputer sebagai alat bantu aplikasi grafik komputer interaktif. Dengan sajian gam-bar akan membuat lebih mudah dalam membaca data maupun informasi dibanding dengan mem-baca data secara manual dalam sebuah ko-munikasi visual.[1]

Dengan penggunaan grafis akan lebih mudah dimengerti dan dipahami dibanding den-gan sederetan kalimat. Melalui perkembangan perangkat lunak memudahkan para desain grafis menjadi lebih mudah

mulai dari pengolahan sampai dengan penyajian grafik termasuk pem-rograman dalam bidang grafis. Salah satu pemrograman desain grafis adalah *OpenGL*.

OpenGL merupakan kepanjangan dari *open graphic library*. *OpenGL* diproduksi oleh Silicon Graphics, Inc (SGI) dan pada awalnya ditujukan hanya untuk sistem komputer mereka, tetapi dalam perkembangannya, *OpenGL* diteri-ma menjadi salah satu bakuan (standard) dalam grafika komputer dan saat ini telah dimple-mentasikan dalam berbagai sistem komputer. *OpenGL* merupakan pustaka program (program *library*) yang menyediakan sejumlah perintah yang berhubungan dengan grafika.[2]

Awal pemrograman grafis adalah mode teks yang diubah menjadi mode grafis dengan memanfaatkan fungsi dari VGA *card*. Pada awal pemrograman grafis, warna yang bisa dimanfaatkan sebanyak 8 warna sampai dengan 16 warna. Dengan pemrograman *visual*, maka penggunaan warna menjadi lebih banyak (0 sampai dengan 255 atau 16 bit warna).

Dengan menggunakan rutin tersebut, para pemrogram komputer grafik dapat membuat grafis yang lebih realistis dari sekedar gambar 2D dan 3D biasa. Portabilitas *OpenGL* sebagai kumpulan *library* memungkinkan untuk diakses melalui bahasa pemrograman (dengan menyer-takan *library OpenGL* di direktori *library* bahasa tersebut) dan dijalankan pada *platform* komputer apapun.[3]

2. Metode

Dalam penggambaran pada visual ber-beda dengan penggambaran pemrograman DOS dimana pada DOS mode dasarnya adalah teks yang diubah menjadi mode grafis sedangkan untuk visual sudah memanfaatkan GUIs (*Graphic User Interfaces*). Untuk hasil tampilannya tentu akan lebih menarik dengan visual apabila bisa mengetahui kombinasi warna RGB.

Untuk perintah *OpenGL* dalam meng-gambar obyek dua dimensi dapat dilihat pada koding 1.

Koding1. Perintah Menggambar Obyek Dua Dimensi Tanpa Warna.

```
#include "windows.h"
#include "stdafx.h"
#include "gl/glut.h"
void myInit(void)
{
    glClearColor(1.0,1.0,1.0,0.0);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPointSize(4.0);
    glMatrixMode(GL_PROJECTION);
```

```

    glLoadIdentity();
    gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}

void myDisplay(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_LINE_LOOP);
        glVertex2i(100,40);
        glVertex2i(100,100);
        glVertex2i(160,160);
        glVertex2i(220,100);
        glVertex2i(220,40);
    glEnd();

    glBegin(GL_LINE_LOOP);
        glVertex2i(110,40);
        glVertex2i(110,80);
        glVertex2i(140,80);
        glVertex2i(140,40);
    glEnd();

    glBegin(GL_LINE_LOOP);
        glVertex2i(170,50);
        glVertex2i(170,80);
        glVertex2i(210,80);
        glVertex2i(210,50);
    glEnd();

    glFlush();
}

void main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE|
GLUT_RGB); // set modus display
    glutInitWindowSize(640,480);
    glutInitWindowPosition(100, 150);
    glutCreateWindow("Gambar 2D ");
    glutDisplayFunc(myDisplay);
    myInit();
    glutMainLoop();
}

```

Tampilan dari perintah OpenGL pada koding 1., dapat dilihat pada gambar 1.

Setelah melihat tampilan output dari koding 1, maka gambar obyek tersebut kurang menarik karena obyek dua dimensi hanya berupa kum-pulan garis yang membentuk sebuah obyek dalam hal ini adalah sebuah rumah sederhana.

Untuk dapat mewarnai dan diberi gradasi warna seperti pada gambar 2., maka dapat dilihat pada koding 2.

Koding 2. Perintah Menggambar Obyek Dua Dimensi Dengan Warna

```

#include "windows.h"
#include "stdafx.h"

```

```

#include "gl/glut.h"
void myInit(void)
{
    glClearColor(1.0,1.0,1.0,0.0);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPointSize(4.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}

void myDisplay(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_POLYGON);
        glColor3f (1.0, 0.8, 0.6);
        glVertex2i(100,40);
        glColor3f (0.2, 0.8, 0.6);
        glVertex2i(100,100);
        glColor3f (0.4, 0.66, 0.5);
        glVertex2i(160,160);
        glVertex2i(220,100);
        glVertex2i(220,40);
    glEnd();

    glBegin(GL_POLYGON);
        glColor3f (1.0, 0.6, 0.3);
        glVertex2i(110,40);
        glVertex2i(110,80);
        glVertex2i(140,80);
        glVertex2i(140,40);
    glEnd();

    glBegin(GL_LINE_LOOP);
        glVertex2i(170,50);
        glVertex2i(170,80);
        glVertex2i(210,80);
        glVertex2i(210,50);
    glEnd();

    glFlush();
}

void main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE |
GLUT_RGB);
    glutInitWindowSize(640,480);
    glutInitWindowPosition(100, 150);
    glutCreateWindow("Gambar Obyek 2D");
    glutDisplayFunc(myDisplay);
    myInit();
}

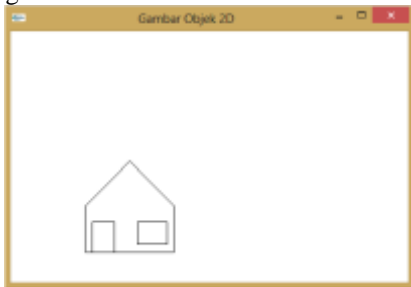
```

```
glutMainLoop();  
}
```

Dengan mengganti perintah *OpenGL* yaitu *glBegin(GL_LINE_LOOP)* diganti dengan *glBegin(GL_POLYGON)*, kemudian pada setiap bagian penempatan koordinat diberi warna dengan perintah *glColor3f* yang merupakan pemberian warna dalam bentuk RGB dengan interval pecahan desimal. Apabila gambar obyek dua dimensi tersebut hanya perlu satu warna, maka penempatan perintah *glColor3f* hanya perlu dilakukan diawal yaitu pada saat penempatan koordinat.

3. Diskusi

Hasil dari penggambaran terlihat bahwa gambar 1., merupakan menggunakan perintah *glBegin(GL_LINE_LOOP)*, sedangkan untuk penggunaan perintah *glBegin(GL_POLYGON)* dapat dilihat pada gambar 2.

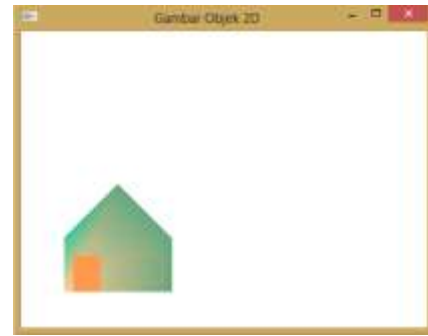


Gambar 1. Menggambar Obyek 2D tanpa warna.

Tampilan gambar 1., menunjukkan warna *glBegin(GL_LINE_LOOP)* akan menggambar-garis sesuai dengan koordinat yang diberikan yaitu dengan perintah *glVertex2i*. Huruf *i* dibelakang angka dua menunjukkan bahwa garis yang digunakan dalam bentuk *integer* (bulat desimal). Untuk menyelesaikan penggambaran satu obyek yang diinginkan diakhir dengan *glEnd()*;

Banyaknya perintah *glBegin (GL_LINE_LOOP)* dan *glEnd()*; tergantung kepada banyaknya obyek dua dimensi yang digambarkan, misalnya pada gambar obyek dua dimensi pada gambar 1., ada sebanyak tiga, maka perintah *glBegin (GL_LINE_LOOP)* dan *glEnd()*; terdapat sebanyak tiga. Untuk bisa menampilkan ke layar setelah obyek dua dimensi ditentukan koordinatnya maka ditambahkan satu perintah *OpenGL* yaitu *glFlush()*; yang berfungsi mengirim semua *output* ke layar.

Untuk mewarnai obyek seperti pada gambar 2., ditambahkan perintah *glColor3f*.



Gambar 2. Menggambar Obyek 2D Dengan Gradasi Warna.

Tampilan pada gambar 2., adalah hasil memodifikasi koding 1., dengan penambahan warna baik dalam bentuk warna tunggal (untuk pintu) maupun dalam bentuk gradasi warna (untuk dinding) pada obyek rumah dua dimensi.

Pemberian angka pada perintah *glColor3f (1.0, 0.8, 0.6)*; akan memberikan spesifikasi warna yang lebih baik. Warna yang diberikan berisi nilai antara 0.0 sampai dengan 1.0. Nilai tersebut nantinya akan dikonversi menjadi rentang nilai 0 sampai dengan 255 pada pewarnaan RGB.

Pemberian *glColor3f(R.r, G.g, B.b)*; Perintah *glVertex* akan memberikan efek pewarnaan untuk setiap bagian obyek yang digambarkan. Walaupun demikian, agar dapat diwarnai maka perintah *glBegin(GL_LINE_LOOP)*; harus diganti dengan *glBegin (GL_POLYGON)*; Pada gambar obyek dua dimensi pada gambar 2., menggunakan penggunaan perintah *glBegin (GL_POLYGON)*; untuk dinding depan dan pintu, sedangkan untuk *glBegin (GL_LINE_LOOP)*; dapat dilihat pada bagian obyek jendela.

4. Hasil

Untuk menggambar objek dua dimensi mulai dari yang mudah sampai dengan yang kompleks, maka perlu penguasaan dasar koordinat kartesius. Penggambaran obyek dimulai dari sudut kiri bawah (0, 0). Dalam menggambar obyek dua dimensi tersebut dapat menggunakan kertas matematika untuk memudahkan penggambaran sehingga tidak menebak koordinat dari obyek dua dimensi yang akan digambarkan. Ukuran layar yang digunakan dapat ditentukan dengan melakukan *setting-an* dengan perintah *OpenGL*.

Dalam pembuatan obyek dua dimensi dengan *OpenGL* tidak bisa langsung dengan memanggil perintah yang sudah disediakan dalam *library*-nya, tetapi perlu dilakukan *setting* pada pemrograman yang digunakan. Pada pemrograman berbasis visual lebih mudah dilakukan terutama pada Microsoft Visual C++ 2005, dimana ada file dari *glu*.** yang ditempatkan kepada 3 folder yaitu di include folder *gl*, *library*, dan *system32*. Penempatan *file* tersebut tidak boleh sembarang karena akan mengakibatkan obyek yang dibuat tidak akan tampil di layar.

Untuk pewarnaan pada obyek selain memanfaatkan warna RGB (*Red, Green, Blue*), juga dapat dibuat dalam bentuk gradasi warna. Gradasi warna akan membuat obyek yang digambar menjadi lebih menarik.

5. Daftar Pustaka

- [1] Mufadhol, Visualisasi Obyek Pada Grafika Komputer Dengan Menggunakan *Processing*, Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang, Jurnal Transformatika, Volume 7, No.2, Januari 2010:77-84.
- [2] Kardian, A. R., Bheta A. Wardijono, Pewarnaan dan Perputaran *Polygon* Untuk Obyek Gambar Segi Tiga dan Segi Empat Menggunakan Program *OpenGL* 32, STMIK JAKARTA STI&K, Jurnal Komputasi, Volume 10 Nomor: 2 Desember 2011 ISSN: 1412-9434
- [3] Prasetyo, E., Dian Kusuma Ningtyas, Prasetyo, Teknik *Cube Mapping* Dengan *OpenGL*, Program Sarjana Magister Universitas Gunadarma, Jurnal Informatika, Volume 4 Nomor 1, April 2008.