

DIAN PRATIWI. ST, MTI

**PEMROGRAMAN  
GRAFIKA KOMPUTER  
BERBASIS OPENGL**

Diterbitkan melalui

**Nida Dwi Karya Publishing**

PEMROGRAMAN GRAFIKA KOMPUTER BERBASIS  
OPENGL

Oleh: *Dian Pratiwi, ST, MTI*

Copyright © 2015 by *Dian Pratiwi, ST, MTI*

**Penerbit**

*www.nulisbuku.com*

&

Nida Dwi Karya Publishing

**Desain Sampul:**

*Soikhu Rojib, SPd*

Diterbitkan melalui:

**Nida Dwi Karya Publishing**

# **BAB I**

## **Konsep Dasar Ilmu Grafika Komputer dan OpenGL**

### **1. Peranan Ilmu Grafika Komputer**

Grafika komputer (*Computer graphics*) merupakan salah satu cabang disiplin ilmu informatika yang mempelajari pembuatan dan manipulasi gambar dengan computer atau secara digital. Grafika komputer sering dikenal juga dengan istilah visualisasi data.

Grafika komputer adalah ilmu yang sangat cepat berkembang pada saat ini. Perkembangan ini didukung oleh munculnya processor-processor komputer yang cepat dan kartu grafis yang semakin canggih dan hebat.

Bentuk sederhana dari grafika komputer adalah grafika komputer 2D, yang kemudian dikembangkan menjadi grafika komputer 3D, pengolahan citra (*image processing*), dan pengenalan pola (*pattern recognition*). Dalam pengolahan citra, gambar input yang diinput untuk diolah lebih lanjut telah tersedia. Sedangkan di dalam grafika komputer, dilakukan proses untuk menciptakan gambar input dari awal.

Bagian dari ilmu grafika komputer meliputi :

- a) Geometri, yaitu mempelajari cara menggambarkan permukaan bidang
- b) Animasi, yaitu mempelajari cara menggambarkan dan memanipulasi gerakan, baik secara otomatis maupun manual.

- c) Rendering, yaitu mempelajari algoritma untuk menampilkan efek cahaya
- d) Citra (*imaging*), yaitu mempelajari cara pengambilan dan penyuntingan gambar.

## **2. Pengenalan OpenGL pada Ilmu Grafika Komputer**

OpenGL adalah sebuah teknik pemrograman interface dengan ANSI C dan FORTRAN yang dikembangkan oleh Mark Kilgard untuk pengelolaan windows dan animasi.

GLUT (*GL Utility Toolkit*) pada OpenGL mendukung fungsi berikut ini :

- Menyediakan sub menu API (*Application Programming Interface*)
- Didesain khusus untuk program rendering
- Mendukung format bitmap dan fungsi keyboard
- Memiliki fungsi mengatur sistem windows

Fungsi GLUT dapat diklasifikasikan ke dalam sub-API berdasarkan kegunaannya, diantaranya :

### **a. Inisialisasi**

Merupakan inisialisasi sistem window, dan berfungsi untuk mengatur pembuatan window.

Contoh kodenya :

```
void glutInit(int *argcp, char **argv);  
void glutInitWindowSize(int width, int height);  
void glutInitWindowPosition(int x, int y);
```

b. Proses atau event awal

Event yang menandakan pemrosesan GLUT terjadi. Umumnya tidak mengembalikan nilai, dan selalu memanggil GLUT *callback* secara berlanjut.

Contoh kodenya :

```
void glutMainLoop(void);
```

c. Manajemen window

Fungsi ini membuat dan mengontrol windows

Contoh kodenya :

```
int glutCreateWindow(char *name);
```

```
int glutCreateSubWindow(int win, int x, int y, int width, int height);
```

d. Manajemen overlay

Fungsi ini untuk menstabilkan dan mengatur *overlay* dalam windows. *Overlay* adalah suatu keadaan dimana bidang normal yang digunakan ditampilkan dengan menggunakan lapisan *buffer* atau *frame*, sehingga menghasilkan efek tertentu.

Contoh kodenya :

```
void glutEstablishOverlay(void);
```

```
void glutUseLayer(GLenum layer);
```

e. Manajemen menu

Fungsi ini membuat dan mengatur menu pop-up (menu yang dapat ditampilkan dengan penggunaan mouse pada window).

Contoh kodenya :

```
int glutCreateMenu(void (*func)(int value));
```

```
void glutDestroyMenu(int menu);
```

- f. Registrasi pemanggilan ulang (*callback*)  
Fungsi ini untuk memperbolehkan memanggil GLUT event proses *looping* pada saat terjadi *callback*.

Contoh kodenya :

```
void glutDisplayFunc(void (*func)(void));  
void glutReshapeFunc(void (*func)(int width, int  
height));
```

- g. Manajemen warna dan indeks warna (*color index*)

OpenGL mendukung warna RGBA dan rendering indeks warna. Fungsi GLUT color index ini untuk menulis dan membaca masukan-masukan dalam pemetaan indeks warna di window.

Contoh kodenya :

```
void glutSetColor(int cell, GLfloat red, GLfloat  
green, GLfloat blue);  
void glutCopyColormap(int win);
```

- h. Menerima status

GLUT menyediakan fungsi untuk programmer agar dapat mengetahui atau menerima status program atau proses, meski tidak semua status dapat diterima.

Contoh kodenya :

```
int glutGet(GLenum state);  
int glutLayerGet(GLenum info);
```

- i. *Font rendering*

GLUT menyediakan dua tipe font rendering, yaitu :

- Stroke font
- Bitmap font

Bitmap font kurang fleksibel saat dirender menjadi sebuah bitmap, namun kecepatannya lebih tinggi jika dibandingkan dengan stroke font.

Contoh kodenya

```
void glutBitmapCharacter(void *font, int
character);
int glutBitmapWidth(GLUTbitmapFont font, int
character);
```

j. Bentuk objek rendering

GLUT merupakan salah satu fungsi yang memudahkan untuk pengenalan objek geometrik 3D. Fungsi GLUT ini dapat membentuk efek pencahayaan namun dengan tidak memperluas koordinat tekstur.

Contoh kodenya :

```
void glutSolidSphere(GLdouble radius, GLint
slices, GLint stacks);
void glutWireSphere(GLdouble radius, GLint
slices, GLint stacks);
```

### 3. Kode Perintah Pada OpenGL

Pada OpenGL, setiap pengkodeannya (*coding*) adalah dengan menggunakan prefix atau awalan “gl” yang kemudian diikuti kata berawalan huruf kapital sebagai nama perintahnya. Contoh, *glMatrixMode()*. Untuk mendefinisikan nilai konstan atau kontanta, OpenGL menuliskannya dengan mengawali “GL\_”, dan semuanya berhuruf kapital yang kemudian diikuti *underscore* untuk menghubungkan kata atau nama variabelnya. Contoh, *GL\_COLOR\_BUFFER\_BIT*.

Di dalam OpenGL, pengkodeannya juga ada beberapa yang berbeda. Perintahnya (*command*) juga ada yang terdapat perluasan detail mengenai perintahnya yang digunakan, umumnya untuk menyatakan jumlah argumen dan tipe datanya. Misalnya, pada perintah *glColor3f( )*. Angka „3” ini dimaksudkan sebagai banyaknya argumen yang ada dalam perintah *gl*-nya dan merupakan jumlah warnanya yaitu R, G, dan B. Sedangkan „f” di sini dimaksudkan sebagai tipe datanya yaitu tipe float.

Pada beberapa perintah OpenGL, tipe data yang dapat diterima untuk melengkapi argumennya yaitu sebagai berikut :

**Tabel 1.1 Perintah OpenGL dan Tipe Data Argumen**

Akhiran	Tipe data	Tipe data dalam bahasa C	Definisi tipe data OpenGL
B	8-bit integer	signed char	Glbyte
S	16-bit integer	Short	GLshort
I	32-bit integer	int atau long	Glint, GLsizei
F	32-bit float	Float	GLfloat, GLclampf
D	64-bit float	Double	Gldouble, GLclampd
Ub	8-bit unsigned integer	unsigned char	Glubyte, GLboolean
Us	16-bit integer	unsigned short	GLushort
Ui	32-bit integer	unsigned int atau unsigned long	GLuint, GLenum, GLbitfield



Misalnya dua perintah di bawah ini :

```
glVertex2i(1, 3);  
glVertex2f(1.0, 3.0);
```

adalah sama. Perbedaannya hanyalah pada tingkat presisinya, yaitu pada perintah pertama menggunakan 32-bit integer, dan pada perintah kedua menggunakan 32-bit float.

Pada beberapa perintah OpenGL juga terdapat akhiran „v“ pada penamaan fungsi gl-nya. Akhiran „v“ diidentifikasi sebagai perintah atau pointer yang menunjuk ke sebuah vektor atau array dari suatu nilai yang ada dalam argumen. Misalnya seperti contoh berikut ini :

```
glColor3f(1.0, 0.0, 0.0);
```

ekivalen dengan kedua perintah berikut :

```
GLfloat color_array[] = {1.0, 0.0, 0.0};  
glColor3fv(color_array);
```

#### **4. Bentuk-Bentuk Primitif Geometri OpenGL**

Untuk memulai membuat berbagai bentuk geometri, seperti polygon, segitiga, garis, segiempat, dan lain-lain, kode yang harus diletakkan setelah inisialisasi adalah *glBegin( )*. Kemudian buatlah perintah untuk membuat suatu bentuk geometri, dan diakhiri dengan *glEnd( )*.

Contohnya pada pembuatan polygon berikut ini :

```
glBegin(GL_POLYGON);  
glVertex2f(0.0, 0.0);  
glVertex2f(0.0, 3.0);  
glVertex2f(4.0, 3.0);  
glVertex2f(6.0, 1.5);  
glVertex2f(4.0, 0.0);  
glEnd();
```