LAPORAN PRAKTIKUM
GRAFIKA DAN KOMPUTASI VISUAL

Judul Praktikum:

Depth dan Lighting dalam OpenGL dan Realisasinya dalam Membuat Truk

Disusun oleh:

Nama : Mochammad Arya Jadmika

NIM : 24060121130085

Laboratorium : B1

INFORMATIKA

FAKULTAS SAINS DAN MATEMATIKA

UNIVERSITAS DIPONEGORO

SEMARANG

2022

PEMBAHASAN

Dalam pembuatan objek truk gandeng 3 dimensi ini, pertama yang harus dilakukan adalah mendefinisikan variabel-variabel yang akan digunakan sebagai berikut:

```
float angle=0.0, deltaAngle = 0.0, ratio;
float x=0.0f,y=1.75f,z=15.0f; // posisi awal kamera
float lx=0.0f,ly=0.0f,lz=-1.0f;
int deltaMove = 0,h,w;
int bitmapHeight=12;
```

Untuk rincian penjelasannya sebagai berikut:

a. angle = 0.0 : variabel untuk sudut, bertipe data float
b. x = 0.0f, y = 1.75f, z = 15.0f : posisi awal kamera
c. lx = 0.0f, ly = 0.0f, lz = -1.0f : definisi dan inisialisasi variabel lx, ly, lz
d. deltamove = 0,h,w : definisi dan inisialisasi variabel deltaMove
e. bitmapHeight = 12 : definisi dan inisialisasi variabel bitmapHeight

setelah semua variabel yang nantinya digunakan dalam semua prosedur dan fungsi dalam source code didefinisikan, maka akan dibahas satu per satu semua prosedur dan fungsi yang ada.

1) Prosedur Reshape

```
void Reshape(int w1, int h1)
{
    // Fungsi reshape
    if(h1 == 0) h1 = 1;
    w = w1;
    h = h1;
    ratio = 1.0f * w / h;
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glViewport(0, 0, w, h);
    gluPerspective(45,ratio,0.1,1000);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(
        x, y, z,
        x + lx,y + ly,z + lz,
        0.0f,1.0f,0.0f);
}
```

Fungsi *reshape* adalah sebuah fungsi callback yang jika digunakan, maka GLUT akan memberikan pointer pada fungsi *reshape*, lalu GLUT akan memanggil fungsi *reshape* tersebut kapanpun ukuran atau bentuk dari jendela aplikasi berubah. Fungsi *reshape* memiliki dua parameter yaitu lebar dan tinggi dari jendela yang bentuknya berubah.

2) Prosedur moveMeFlat

```
void moveMeFlat(int i)
{
    // Fungsi ini untuk maju mundur kamera
    x = x + i*(lx)*0.1;
    z = z + i*(lz)*0.1;
    glLoadIdentity();
    gluLookAt(x, y, z,
    x + lx,y + ly,z + lz,
    0.0f,1.0f,0.0f);
}
```

Prosedur moveMeFlat digunakan untuk mengatur kamera dan pengaturan tampilan ketika kamera digerakkan/ saat transformasi, baik itu transformasi rotasi maupun translasi.

3) Prosedur Grid

```
void Grid() {
    // Fungsi untuk membuat grid di "lantai"
    double i;
    const float Z_MIN = -50, Z_MAX = 50;
    const float X_MIN = -50, X_MAX = 50;
    const float gap = 1.5;
    glColor3f(0.5, 0.5, 0.5);
    glBegin(GL_LINES);
     for(i=Z_MIN;i<Z_MAX;i+=gap)
     {
        glVertex3f(i, 0, Z_MIN);
        glVertex3f(i, 0, Z_MAX);
     }

     for(i=X_MIN;i<X_MAX;i+=gap)
     {
        glVertex3f(X_MIN, 0, i);
        glVertex3f(X_MAX, 0, i);
     }
 glEnd();
}
```

Prosedur Grid digunakan untuk membuat pola garis kotak-kotak pada lantai. Pola garis kotak-kotak pada lantai akan digambar dengan memanggil GL_LINES yang menjadi parameter glBegin. glBegin merupakan fungsi untuk membatasi (awal) vertex dari sebuah primitive atau sekumpulan primitive, dalam hal ini GL_LINES yang digunakan untuk membuat garis. Kemudian digunakan juga glVertex3f yang merupakan fungsi untuk mengatur koordinat dari titik-titik pembentuk garis. Fungsi glVertex3f memiliki 3 parameter yang akan menentukan koordinat dari titik-titik pembentuk garis yaitu x, y, dan z.

4) Prosedur Truk

Berikut akan dibahas mengenai proses pembuatan objek truk.

a) glPushMatrix()

Pada pembentukan objek Truk akan dibutuhkan proses transformasi, seperti translasi dan rotasi, yang mana proses transformasi tersebut membutuhkan matriks-matriks transformasi. OpenGL akan menyimpan setumpuk matriks transformasi ini agar dapat dengan cepat diterapkan dan dihapus transformasinya. Dalam hal ini, glPushMatrix akan menyalin matriks paling atas (matriks transformasi yang berada dibawah glPushMatrix) dan mendorongnya ke tumpukan matriks

dengan glPushMatrix() yang mendorong matriks, pengguna dapat mengontrol transformasi apa yang diterapkan pada objek apa, serta menerapkan ttransformasi ke objek tertentu dan dengan mudah menghapus transformasinya tanpa mempengaruhi objek lain .

b) glPopMatrix()

glPopMatrix memunculkan (pops) matriks teratas dari tumpukan.

c) glBegin()

glBegin adalah fungsi untuk membatasi (awal) simpul (vertex) dari sebuah primitive atau sekumpulan primitive, di antaranya GL_QUADS, GL_LINES, dan primitive lainnya.

d) glEnd()

glEnd merupakan fungsi untuk membatasi (akhir) vertex dari sebuah primitive atau sekumpulan primitive.

e) glTranslatef()

glTranslatef digunakan untuk melakukan translasi terhadap matriks saat ini dengan matriks translasi. Faktor translasi dispesifikasikan dalam parameternya, yaitu

x : faktor translasi koordinat x
y : faktor translasi koordinat y
z : faktor translasi koordinat z

f) glRotated()

glRotated digunakan untuk melakukan transformasi rotasi terhadap matriks dengan sudut dan faktor rotasi yang menjadi parameternya, yaitu *angle* : sudut rotasi.

g) glColor3f (red, green, blue)

GL menyimpan indeks warna yang bernilai tunggal dan warna RGBA yang mempunyai 4 nilai. glColor mempunyai dua varian, yaitu glColor3f dan glColor4f. Pada pembuatan truk ini menggunakan varian glColor3f yang mana didominasi atau kebanyakan dengan warna hijau (0.0, 0.5, 0.0) dan abu-abu gelap (0.1, 0.1, 0.1)

h) glVertex3f(x, y, z)

Pada fungsi ini digunakan untuk mengatur koordinat dari vertex (simpul). Fungsi glVertex3f dipanggil di dalam pasangan glBegin dan glEnd. Fungsi glVertex3f memiliki 3 parameter yang akan menentukan koordinat dari titik-titik pembentuk garis. Berikut parameternya: glVertex3f(x, y, z) dimana

x : menentukan posisi absis dari titik
y : menentukan posisi ordinat dari titik
z : menentukan posisi koordinat z dari titik

Untuk *source code* yang dibutuhkan dalam pembuatan bagian-bagian dari sebuah Truk nya bisa dilihat pada file (.cpp). Penulis sudah menuliskan keterangan-keterangan yang dibutuhkan didalamnya.

5) Prosedur Display

```
void display() {
    usleep(1);
    // Kalau move dan angle tidak nol, gerakkan kamera...
     if (deltaMove)
        moveMeFlat(deltaMove);
     if (deltaAngle) {
        angle += deltaAngle;
        orientMe(angle);
     }
     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    // Gambar grid
    Grid();

    // Gambar objek di sini...
    TrukGandeng();
    glutSwapBuffers();
    glFlush();
}
```

Prosedur Display berperan sebagai handler kejadian. Ketika prosedur dipanggil dalam fungsi main, maka prosedur-prosedur yang didefinsikan di dalamnya dapat dieksekusi. Pada prosedur display di atas, terdiri atas tiga bagian, yaitu:

a) Untuk transformasi kamera

Terdapat fungsi if yang memanggil prosedur moveMeFlat, dimana jika memenuhi kondisi maka kamera akan digerakkan. Kemudian, buffer akan dibersihkan dengan memanggil glClear yang merupakan operator bitwise OR yang mengidentifikasikan bahwa buffers akan dibersihkan. Di mana pada glClear menggunakan parameter konstan, yaitu GL_COLOR_BUFFER_BIT yang menjelaskan bahwa buffers dapat digunakan untuk penulisan warna dan GL_DEPTH-BUFFER_BIT untuk membersihkan buffer kedalaman (depth). Selanjutnya dengan memanggil glPushMatrix maka tercipta matriks 1, dimana matriks ini selanjutnya akan dirotasikan dengan sudut rotasi yang ada dalam variabel rotAngle terhadap sumbu y, kemudian akan dirotasikan dengan sudut rotasi sesuai yang dispesifikasikan dalam variabel rotAngle1 terhadap sumbu x.

b) Memanggil prosedur Grid

Menghandle atau menangani prosedur Grid sehingga prosedur display ketika dipanggil dalam fungsi main, maka prosedur Grid dapat dieksekusi dan menghasilkan output lantai dengan garis kotak-kotak.

c) Memanggil prosedur Truk

Meng*handle* atau menangani prosedur Truk sehingga prosedur display dipanggil dalam fungsi main, prosedur Truk dapat dieksekusi dan menghasilkan output Truk. Kemudian, prosedur-prosedur yang dipanggi; dalam prosedur display agar segera dieksekusi, maka dapat menggunakan glFlush untuk membatasi waktu eksekusi atau memaksa agar program segera dieksekusi.

6) Prosedur pressKey

```
void pressKey(int key, int x, int y) {
    // Fungsi ini akan dijalankan saat tombol keyboard ditekan dan belum dilepas
    // Selama tombol ditekan, variabel angle dan move diubah => kamera bergerak
    switch (key) {
     case GLUT_KEY_LEFT : deltaAngle = -0.01f;break;
     case GLUT_KEY_RIGHT : deltaAngle = 0.01f;break;
     case GLUT_KEY_UP : deltaMove = 1;break;
     case GLUT_KEY_DOWN : deltaMove = -1;break;
 }
}
```

Prosedur ini akan dieksekusi jika tombol keyboard ditekan dan belum dilepas, yaitu selama tombol ditekan, variabel angle (sudut rotasi) dan move akan berubah, artinya kamera akan bergerak.

7) Prosedur releaseKey

```
void releaseKey(int key, int x, int y) {
    // Fungsi ini akan dijalankan saat tekanan tombol keyboard dilepas
    // Saat tombol dilepas, variabel angle dan move diset nol =>kamera berhenti
    switch (key) {
     case GLUT_KEY_LEFT :
        if (deltaAngle < 0.0f)
            deltaAngle = 0.0f;
        break;
     case GLUT_KEY_RIGHT : if (deltaAngle > 0.0f)
        deltaAngle = 0.0f;
        break;
     case GLUT_KEY_UP : if (deltaMove > 0)
        deltaMove = 0;
        break;
     case GLUT_KEY_DOWN : if (deltaMove < 0)
        deltaMove = 0;
        break;
 }
}
```

Prosedur ini akan dieksekusi atau dijalankan jika tombol keyboard sudah tidak ditekan atau dilepas. Sehingga saat tombol dilepas, variabel angle dan move menjadi nol, artinya kamera berhenti bergerak.

8) Prosedur lighting

```
// Variable untuk pencahayaan
const GLfloat light_ambient[] = { 0.5f, 0.5f, 0.5f, 0.0f };
const GLfloat light_diffuse[] = { 1.0f, 1.0f, 1.0f, 1.0f };
const GLfloat light_specular[] = { 1.0f, 1.0f, 1.0f, 1.0f };
const GLfloat light_position[] = { 0.0f, 20.0f, 10.0f, 1.0f };
const GLfloat mat_ambient[] = { 0.7f, 0.7f, 0.7f, 1.0f };
const GLfloat mat_diffuse[] = { 0.8f, 0.8f, 0.8f, 1.0f };
const GLfloat mat_specular[] = { 1.0f, 1.0f, 1.0f, 1.0f };
const GLfloat high_shininess[] = { 100.0f };
```

```
void lighting(){
 // Fungsi mengaktifkan pencahayaan
 glEnable(GL_DEPTH_TEST);
 glDepthFunc(GL_LESS);
 glEnable(GL_LIGHT0);
 glEnable(GL_NORMALIZE);
 glEnable(GL_COLOR_MATERIAL);
 glEnable(GL_LIGHTING);
```

```
glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
glMaterialfv(GL_FRONT, GL_SHININESS, high_shininess);
}
```

Prosedur lighting digunakan untuk mengatur pencahayaan. Untuk dapat menerapkan pencahayaan pada suatu objek, perlu untuk mengaktifkannya menggunakan glEnable, dimana ketika GL_LIGHTING diaktifkan, objek tidak hanya akan diwarnai berdasarkan warna apa yang dispesifikasikan dalam glColor, tetapi perlu memberikan spesifikasi lebih seberapa bersinar objeknya nanti, bagaimana objek terkait bercahaya didalam kegelapan, bagaimana objeknya memancarkan cahaya, dan spesifiksi lainnya. Dalam OpenGL memodelkan empat mekanisme ketika cahaya mencapai mata, yaitu ada Ambient, Diffuse, Spekular, dan Emisi.

9) Prosedur init

```
void init(void)
{
    glEnable (GL_DEPTH_TEST);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
}
```

Pada prosedur init ketika depth test diaktifkan maka OpenGL menguji nilai kedalaman dari suatu fragmen terhadap konten atau isi dari buffer kedalaman. OpenGL melakukan pengujian kedalaman dan jika lulus uji, maka fragmen akan direnderdan buffer kedalaman diperbarui dengan nilai kedalaman yang baru. Jika tes gagal, maka fragmen akan dibuang. Kemudian akan dipanggil glPolygonMode yang digunakan untuk memilih mode rsterisasi polygon. glPolygonMode memiliki dua parameter, yaitu:

a) Face
   Yaitu menentukan polygon mana yang nantinya akan diterapkan mode. GL_FRONT_AND_BACK untuk polygon dengan muka depan dan muka belakang.
b) Mode
   Yaitu menentukan bagaimana polygon akan dirasterisasi. Nilai yang diterima GL_POINT, GL_LINE, dan GL_FILL. Jika GL_FILLL maka interior polygon akan terisi.

10) Prosedur Main

a) glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
   glutInitDisplayMode digunakan untuk menginisialisasi mode display dan mempunyai parameter mode untuk mode displaynya, yaitu
   GLUT_DEPTH : memilih jendela dengan buffer kedalaman
   GLUT_DOUBLE : memilih jendela dengan buffer ganda
   GLUT_RGBA : memilih jendela mode RGBA
b) glutInitWindowPosition(100,100); yaitu menginisialisasi posisi jendela
c) glutInitWindowSize(640,480); yaitu menginisialisasi ukuran jendela
d) glutCreateWindow("3D Lighting"); yaitu membuat jendela
e) glutIgnoreKeyRepeat(1); yaituMengabaikan key repeat (saat tombol keyboard dipencet terus)

f) glutSpecialFunc(pressKey); yaitu mengatur callback keyboard khusus untuk jendela saat ini dan juga dipicu saat fungsi keyboard atau tombol arah ditekan.
g) glutSpecialUpFunc(releaseKey); yaitu mengatur callback keyboard khusus untuk jendela saat ini ketika tombol keyboard dilepas (sudah ditekan)
h) glutDisplayFunc(display); yaitu mengatur callback prosedur display untuk jendela saat ini
i) glutIdleFunc(display); yaitu prosedur display akan dipanggil terus menerus.
j) glutReshapeFunc(Reshape); yaitu untuk menyetel callback dari reshape untuk jendela saat ini.
k) lighting();
l) init();
m) glutMainLoop();