LAPORAN PRAKTIKUM
GRAFIKA DAN KOMPUTASI VISUAL

Dibuat Untuk Memenuhi Tugas Besar Pembuatan Animasi Truk

Disusun Oleh :

1. Farrel Andhika Rizky Putra 24060120130071

2. Mohamad Fahrul Islami 24060120130074

3. Wasis Pambudi 24060120120015

PROGRAM STUDI S-1 INFORMATIKA KELAS B

FAKULTAS SAINS DAN MATEMATIKA

UNIVERSITAS DIPONEGORO

SEMARANG

2022

BAB I
PENDAHULUAN

1. Latar Belakang

Salah satu tujuan grafika komputer saat ini adalah menampilkan citra 3-Dimensi yang bagus dalam satuan waktu sekian milidetik pada layer komputer atau yang sering kita sebut dengan animasi. Untuk menghasilkan animasi yang realistik diperlukan banyak proses yang perlu dilakukan, contohnya pembuatan model, pencahayaan, *texturing* , pemberian efek khusus dan lainnya.

Dalam membuat sebuah animasi dibutuhkan OpenGL. OpenGL (Open Graphics Library) adalah standar API yang dapat digunakan untuk membuat aplikasi berbasis grafik, baik dua dimensi (2D) maupun tiga dimensi (3D). OpenGL ini bersifat cross-platform, artinya dapat dijalankan pada berbagai platform sistem operasi yang ada saat ini. Dalam mengoperasikan OpenGL, dapat ditambahkan GLUT yang merupakan sebuah library pada OpenGL dan berfungsi untuk mempermudah dalam penggunaan OpenGL.

Ada 3 file penting dalam OpenGL yang akan digunakan dalam sistem operasi MSWindows, yaitu glut.h, glut32.lib dan glut32.dll. Dimana masing-masing file OpenGL tersebut akan diinstal kedalam Windows. Salah satu penggunaan GLUT pada OpenGL yaitu dapat digunakan untuk membuat animasi truk gandeng yang dapat berinteraksi dengan objek lainnya. Dalam pembuatan animasi ini diperlukan library dan fungsi pendukung lainnya yang dimiliki oleh OpenGL. Untuk lebih jelas dalam memahami dan mendemokan animasi truk gandeng dapat dilihat pada laporan praktikum ini.

2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, didapatkan rumusan masalah sebagai berikut.

a. Bagaimana membuat animasi Truk Gandeng dengan interaksi objek, dan sebagainya menggunakan OpenGL?
b. Bagaimana hasil eksekusi program saat dijalankan?

3. Tujuan

Berdasarkan rumusan masalah, tujuan dari pembuatan laporan ini sebagai berikut.

a. Mahasiswa dapat membuat animasi Truk Gandeng dengan interaksi objek, dan sebagainya menggunakan OpenGL.
b. Mahasiswa dapat menampilkan hasil setelah program dijalankan.

BAB II
DASAR TEORI

1. OpenGL

OpenGL (Open Graphics Library) adalah standar API yang dapat digunakan untuk membuat aplikasi berbasis grafik, baik dua dimensi (2D) maupun tiga dimensi (3D). OpenGL ini bersifat cross-platform, artinya dapat dijalankan pada berbagai platform sistem operasi yang ada saat ini.

Keuntungan dari pendekatan ini adalah bahwa hal itu memungkinkan fleksibilitas yang besar dalam proses menghasilkan gambar. Aplikasi ini gratis untuk trade-off rendering kecepatan dan kualitas gambar dengan mengubah langkah-langkah di mana foto tersebut diambil. Cara termudah untuk menunjukkan kekuatan dari antarmuka prosedural adalah untuk dicatat bahwa antarmuka deskriptif dapat dibangun di atas antarmuka prosedural, tetapi tidak sebaliknya. Pikirkan OpenGL sebagai "bahasa assembly grafis": potongan-potongan fungsi OpenGL dapat dikombinasikan sebagai building blocks untuk menciptakan teknik inovatif dan menghasilkan kemampuan baru grafis.

2. Texture Mapping pada OpenGL

Tekstur adalah suatu metode umum yang digunakan untuk menambahkan suatu detail pada objek dengan memetakan suatu pola kedalam geometris dari objek yang didefinisikan. Pola tekstur didefinisikan dalam suatu array yang berisi nilai-nilai warna atau suatu prosedur yang mengubah warna objek. Metode ini untuk menambah detail objek pada layar yang biasa dikenal dengan texture mapping atau pattern mapping. Spesifikasi suatu tekstur yang menunjukkan pada suatu permukaan tekstur menggunakan koordinat tekstur yang mempunyai range nilai dari 0 sampai 1.0. Tekstur pada suatu permukaan benda/objek umumnya didefinisikan dengan pola warna berbentuk segiempat dan posisi tekstur ini pada suatu permukaan berbentuk koordinat dua dimensi (s,t). Spesifikasi dari masing – masing warna pada pola tekstur disimpan pada suatu array yang terdiri dari 3 komponen array.

Pada laporan kali ini, digunakan kombinasi dari teknik texture mapping dengan environment mapping. Texture Mapping merupakan teknik pemetaan sebuah tekstur pada

pola gambar wireframe, dimana wireframe yang telah dibuat akan ditampilkan memiliki kulit luar seperti tekstur yang diinginkan. Sedangkan di bidang komputer grafik, Environment Mapping merupakan teknik untuk mensimulasikan sebuah obyek agar dapat merefleksikan lingkungan sekitarnya. Teknik ini pertama kali diajukan oleh Blinn dan Newell pada tahun 1976. Pada bentuk yang paling sederhana, teknik ini biasanya memakai obyek yang permukaannya terlihat seperti krom. Konsep dari teknik ini ialah menggunakan beberapa gambar yang diambil dari lingkungan sekitarnya ataupun gambar rekaan untuk dijadikan lingkungan yang akan direfleksikan oleh obyek.

Ada beberapa teknik Environment Mapping, antara lain Sphere Mapping, Dual Paraboloid Mapping, dan Cube Mapping. Adapun yang akan dijelaskan lebih lanjut ialah teknik Cube Mapping karena disini kami menggunakan teknik tersebut.

Cube Mapping sebagai bagian dari metode Environment Mapping merepresentasikan lingkungan sekitarnya dengan cara"menempelkan" enam buah gambar-yang UerUeaa di keenam sisi obyek. Hal ini membuat obyek seolah memiliki enam sisi pantul3, yaitu depan, belakang, kanan, kiri, atas, dan bawah.

Cube Mapping muncul sebagai pengganti dua metode mapping sebelumnya. Hal-hal yang menjadi kelemahan dua metode terdahulu seperti ketergantungan sudut pandang (viei dependency), keterbatasan cangkupan tekstur (warping & distortion), dan kerumitan penerapan menjadi alasan beralihnya teknik mapping ke Cube Mapping. Dengan mentransformasikan tekstur ke dalam enam sisi kubus, Cube Mapping lebih menawarkan kemudahan implementasi karena pantulan pada permukaan obyek cukup dikonsentrasikan di keenam sisi obyek.

Tidak seperti Dual Paraboloid Mapping, teknik Cube Mapping hanya membutuhkan satu unit tekstur4 dan satu tahap rendering. Selain itu, teknik Cube Mapping tlAal mengurangi resolusi gambar (teknik Sphere Mapping dan Dual Paraboloid Mapping dapat mengurangi resolusi gambar sampai 78% dari resolusi semula). Secara konsep, Cube Mapping memang lebih "fo fhe point' dibandingkan dengan dua teknik lainnya. Namun, proses texturing pada Cube Mapping membutuhkan kemampuan yang lebih agar dapat mengakses enam gambar secara bersamaan.

3. Depth dan Lighting pada OpenGL

Depth testing digunakan untuk menghilangkan permukaan-permukaan yang tersembunyi pada saat program dieksekusi. Sedangkan lighting akan jauh lebih terasa jika semua objek telah terdefinisi dan objek sisanya merupakan objek yang berkerangka, tetapi tidak realistis. Untuk menggunakan depth testing, perlu ditambahkannya satu lagi konstanta ke glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH); Sekarang sebuah depth buffer telah dibuat dimana informasi dari depth telah tersimpan dalam tiap pixel. Hal berikunya yaitu mengaktifkan glEnable(GL_DEPTH_TEST); dan kita pun dapat membuat gambar-gambar seperti biasa.

Untuk mengaktifkan pencahayaan dilakukan dengan memanggil fungsi glEnable(GL_LIGHTING); kemudian dilanjutkan dengan mengaktifkan masing-masing lampu yang digunakan, misalnya glEnable (GL_LIGHT0);

Pencahayaan adalah topik yang sangat kompleks dan dapat mensimulasikan setiap kemungkinan cahaya dengan OpenGL. Untuk menetapkan sifat cahaya, harus menggunakan perintah glLight*(), di mana * diisi dengan I atau f, jika ingin menentukan nilai sebagai vektor, gunakan "v". Untuk menentukan posisi pada OpenGL, hal yang pertama kali dilakukan ialah mendefinisikan sebuah vektor empat-dimensi (x, y, z dan w). Nilai x, y dan z dibagi oleh w, itu berarti bahwa jika kita menetapkan nilai "w" sebesar 0.0, posisi akan meluas sangat jauh. Contoh untuk ini adalah matahari. Jika ingin menggunakan apa yang disebut "posisi" cahaya (yaitu w adalah nol), kita harus menetapkan w menjadi 1,0.

Contoh : static GLfloat LightPos[] = {0.5, 0.5, 1.0, 1.0};

Kemudian panggil glLightfv(GL_LIGHT0, GL_POSITION, LightPos); Tentu saja kita bisa menggunakan nilai-nilai yang lain untuk GL_LIGHT0.

4. Interaksi Objek OpenGL

Di era sekarang ini, hampir semua animasi maupun game pada komputer selalu menggunakan keyboard untuk mengontrol objek objek didalamnya. Dengan adanya fungsi fungsi pada keyboard ini dapat memudahkan pengguna dalam memainkan atau mengendalikan game tersebut. Interaksi yang ada tidak lain untuk mencapai scenario yang kita inginkan. Seperti Untuk mendorong objek dengan objek lainnya atau menabrak objek lainnya.

Sayangnya ada keterbatasan interaksi pada openGL dimana tidak terdapat pendeteksi untuk mengetahui apakah dua objek sedang saling berinteraksi/bersentuahn atau tidak. Hal ini tentu saja merugikan dalam mengembangkan skenario yang diinginkan. Sebagai solusi untuk tetap mendapatkan skenario yang diinginkan interaksi dilakukan dengan melakukan perkondisian dari koordinat lokasi objek-objek yang ingin diinteraksi. Seperti jika ingin objek A mendorong objek B maka dibuat permisalan jika koordinat sisi objek A telah menyentuk koordinat sisi objek B maka objek B akan mulai bergerak searah dengan objek A. Untuk lebih memahami konsep di atas, maka akan dijelaskan seperti di bawah ini.

```
GLUTAPI void APIENTRY glutKeyboardFunc(void (GLUTCALLBACK *func)(unsigned char key, int x, int y));
```

Dalam penggunaan `glutKeyboardFunc` dimungkinkan untuk mendeteksi input dari keyboard. Fungsi ini diletakkan pada fungsi main dari program, dan parameternya adalah callback

function yang telah didefinisikan berupa fungsi dengan 3 parameter, seperti contoh di bawah ini.

```
void myKeyboard(unsigned char key, int x, int y){ if(key == 'a') glTranslatef(4,0,0); //seleksi tombol yang ditekan }
void mySpecialKeyboard(int key, int x, int y){ switch(key){ case GLUT_KEY_??? : …; break; } }
```

Agar fungsi keyboard ini dapat dideteksi terus maka fungsi untuk animasi (update) harus telah disertakan.

Untuk fungsi callback yang memanggil tombol keyboard normal/biasa adalah `glutKeyboardFunc(myKeyboard);` //hanya memanggil fungsi myKeyboard, sedangkan untuk mendeteksi tombol-tombol keyboard yang bersifat spesial seperti tombol F1, arah panah, Home, Enter, dsb dapat menggunakan callback function `glutSpecialFunc(mySpecialKeyboard);` //hanya memanggil fungsi mySpecialKeyboard

BAB III

PEMBAHASAN

A. Source Code Program

```
#include <math.h>
#include <gl/glut.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <assert.h>
#include <fstream>
#include "imageloader.h"
#define EN_SIZE 15

GLuint _textureId, _textureIdDepan, _textureIdKanan,
_textureIdBelakang, _textureIdKiri, _textureIdAtas;
float angle=0.0, deltaAngle = 0.0, ratio;
float x=0.0f,y=20.0f,z=95.0f;
float lx=0.0f,ly=0.0f,lz=-1.0f;
int deltaMove = 0,h,w;
static int rotAngleX =0, rotAngleY =0, rotAngleZ =0;
int bitmapHeight=12;
float panjang, lebar, tinggi;
const double PI = 3.141592653589793;
int i,j,radius,jumlah_titik,x_tengah,y_tengah;
bool start = false;
float r[] = {0.1,0.4,0.0,0.9,0.2,0.5,0.0,0.7,0.5,0.0};
float g[] = {0.2,0.0,0.4,0.5,0.2,0.0,0.3,0.9,0.0,0.2};
float b[] = {0.4,0.5,0.0,0.7,0.9,0.0,0.1,0.2,0.5,0.0};
int tola[5000][5000];
float tX=0,tY=0,tZ=-8,rX=0,rY=0,rZ=14;
float tZ1=-20,tZ2=-40,tZ3=-60,tZ4=-80,tZ5=-100,tZ6=-120,tZ7=-140,tZ8=-
160,tZ9=-180,tZ10=-200;
float yappari = 15;
int belok = 0;
float angTrukDepan = 0.0, angTrukBelakang = 0;
```

```
float posXTruk = 0.0, posZTruk = 0.0;
float speed = 1;
float posZJalan = -100,posZPohon = -100,posZRumah = -250,posZMobil =
200; //Posisi awal tiap object
int ujung = 0, ujungP = 0, ujungR = 0, ujungM = 0;
bool touch = false;
```

B. Penjelasan Source Code Program

1. Library dan Deklarasi Variabel

Program Truk Gandeng 3D menggunakan beberapa library yang harus di-include, yaitu <math.h>, <GL/glut.h>, <GL/glu.h>, <GL/gl.h>, <stdio.h>, <string.h>, dan <stdlib.h>.

Setelah seluruh library ter-include selanjutnya kita mendeklarasikan variabel-variabel yang diperlukan. Variabel yang dibuat yaitu :

a. GLuint _textureId, _textureIdDepan, _textureIdKanan, _textureId Belakang, _textureIdKiri, _textureIdAtas; ID OpenGL untuk tekstur
b. float angle=0.0, deltaAngle = 0.0, ratio; untuk menentukan angle dari kamera
c. float x=0.0f,y=20.0f,z=95.0f; variabel untuk posisi awal kamera
d. float lx=0.0f,ly=0.0f,lz=-1.0f; variabel untuk arah kamera
e. int deltaMove = 0,h,w; variabel untuk pergerakan maju mundur kamera
f. static int rotAngleX =0, rotAngleY =0, rotAngleZ =0; variabel untuk merotasi kamera
g. int bitmapHeight=12;
h. float panjang, lebar, tinggi; variabel untuk membuat truk
i. const double PI = 3.141592653589793; variabel PI
j. int i,j,radius,jumlah_titik,x_tengah,y_tengah; variabel untuk membuat roda
k. float r[] = {0.1,0.4,0.0,0.9,0.2,0.5,0.0,0.7,0.5,0.0};
l. float g[] = {0.2,0.0,0.4,0.5,0.2,0.0,0.3,0.9,0.0,0.2};
m. float b[] = {0.4,0.5,0.0,0.7,0.9,0.0,0.1,0.2,0.5,0.0}; variabel warna rumah
n. int tola[5000][5000]; variabel tinggi rumah

o. float tX=0,tY=0,tZ=-8,rX=0,rY=0,rZ=14; variabel posisi rumah
p. float tZ1=-20,tZ2=-40,tZ3=-60,tZ4=-80,tZ5=-100,tZ6=-120,tZ7=-140,tZ8 =-160,tZ9=-180,tZ10=-200; variabel posisi rumah
q. float yappari = 15; variabel pengubah posisi truk
r. int belok = 0; variabel untuk belok truk
s. float angTrukDepan = 0.0, angTrukBelakang = 0; variabel angle dari truk
t. float posXTruk = 0.0, posZTruk = 0.0; variabel posisi truk
u. float speed = 1; variabel kecepatan
v. float posZJalan = -100,posZPohon = -100,posZRumah = -250,posZMobil = 200; variabel Posisi awal tiap object
w. int ujung = 0, ujungP = 0, ujungR = 0, ujungT = 0; variabel posisi ujung object
x. bool touch = false; variabel kondisi truk menabrak atau tidak.

2. Prosedur dan Fungsi Dasar untuk Menjalankan OpenGL

Prosedur reshape berfungsi sebagai parameter fungsi glutReshapeFunc yang ada di main. Fungsinya adalah mengatur ulang callback dari window saat ini. Dalam prosedur reshape ini terdapat gluLookAt yang berfungsi untuk mengatur garis pandang kamera. Prosedur Reshape berguna untuk mengatur proyeksi objek yang ada, yaitu truk.

Prosedur orientMe dan moveMeFlat adalah prosedur untuk mengatur arah kamera. Prosedur orientMe berguna untuk mengatur kamera agar bisa tengok ke kiri dan ke kanan. Prosedur moveMeFlat berguna untuk mengatur kamera agar bisa bergerak maju dan mundur. Kedua prosedur ini menggunakan gluLookAt.

Prosedur display akan digunakan pada fungsi glDisplayFunc dan glIdleFunc pada main. Fungsinya untuk merealisasikan semua prosedur sebelumnya seperti prosedur TrukDepan, TrukBelakang, orientMe, dan moveMeFlat, Jalan, Pohon, Rumah, Mobil, dan dapat merotasi permukaan.

Prosedur pressKey dan releaseKey berguna sebagai prosedur yang dijalankan ketika user menekan key yang ada. Prosedur pressKey dijalankan ketika

user menekan key atas, bawah, kiri, kanan dan akan menjalankan kamera sesuai key yang ditekan. Selama key ditekan, maka prosedur akan terus dijalankan. Sedangkan prosedur releaseKey akan dijalankan ketika user tidak lagi menekan key yang sama.

Prosedur init berguna untuk mengaktifkan GL_DEPTH_TEST dan melakukan glPolygon Mode. Di main, semua program sebelumnya diterapkan. glutInit, glutInitDisplay Mode, glutInitWindowPosition, glutInitWindowSize, glutCreateWindow berfungsi sebagai awal membuat window tempat objek truk gandeng berada. glutIgnoreKeyRepeat berfungsi untuk mengabaikan key yang ditekan terus menerus. glutSpecialFunc berguna untuk memanggil void pressKey. glutSpecialUpFunc berguna untuk memanggil void releaseKey. GlutDisplayFunc berguna untuk memanggil void Display. Dan glutIdleFunc berguna agar void Display dipanggil terus menerus. glutReshapeFunc berguna untuk memanggil reshape. Lalu lighting untuk pencahayaan. Dan glutMainLoop untuk melakukan perulangan processing glut.

3. Pembuatan Objek Dua Truk Gandeng
    a. TrukDepan dan TrukBelakang

       Pembuatan truk dilakukan dengan membuat masing-masing bagian dari truk mulai dari kepala truk, bemper, roda hingga muatan truk. Truk dibagi menjadi dua yaitu bagian depan dan belakang. Hal ini dilakukan agar pada pembuatan animasi truk berbelok terlihat lebih realistis.
    b. Pohon1 dan 2

       Pembuatan pohon hampir sama seperti truk. Hanya saja dibuat agar menyerupai pohon.
    c. Rumah

       Pembuatan rumah menggunakan fungsi yang sama dengan pohon dan truk. Namun dengan konfigurasi yang berbeda sehingga menghasilkan bentuk rumah.
    d. Mobil

       Pembuatan mobil hanya menggunakan fungsi untuk membuat mobil sederhana saja.
    e. Awan

Pembuatan awan menggunakan fungsi yang sama dengan pohon dan rumah. Namun dengan konfigurasi yang berbeda sehingga menghasilkan bentuk awan.

4. Menambah Pencahayaan

   Pencahayaan dilakukan dengan terlebih dahulu membuat seta menentukan variabel sebagai konstanta dengan tipe data GLfloat. Variabel ini adalah light_ambient, light_diffuse, light_specular, light_position, mat_ambient, mat_diffuse, mat_specular, dan high_shininess. Kemudian, dibuat prosedur void lighting() untuk mengaktifkan pencahayaan berdasarkan variabel beserta nilai-nilainya yang telah ditentukan. Fungsi yang dipakai antara lain glEnable, glDepthFunc, glLightfv, dan glMaterialfv. Untuk membuat pencahayaan terlihat menyeluruh dan berasal dari matahari, digunakan static GLfloat LightPos dengan parameter w adalah 0 dan x,y,z merupakan titik arah matahari.

5. Prosedur Animasi Objek

   a. Animasi Truk

      Pembuatan prosedur TrukMove() dimulai dengan deklarasi variabel posXTruk dan posZTruk. Kedua variabel tersebut merupakan letak dimana truk berada berdasarkan posisi sumbu X dan Z. Kemudian untuk berbelok ditambahkan variabel Belok yang merupakan kondisi truk ketika berbelok. Jika belok bernilai 1 maka truk berbelok kiri, apabila bernilai -1 maka truk berbelok kanan.

   b. Animasi Jalan Tol, Pohon, Rumah, Mobil dan Awan

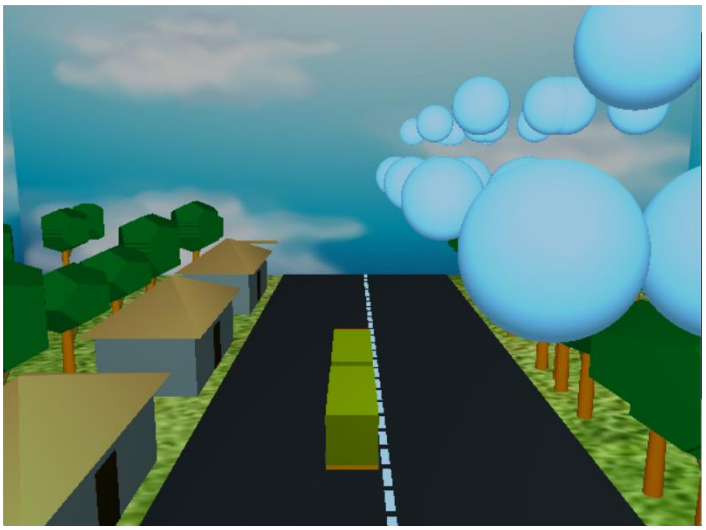
      Prosedur PergerakanJalan(), PohonMove(), RumahMove(), MobilMove() dan AwanMove() memiliki prinsip yang sama. Dimana ketiganya memiliki variabel posisi dan ujung, dimana ujung bernilai awal satu. Apabila ujung masih bernilai satu maka posisi akan terus dikurangi dengan nilai speed/kecepatan. Jika posisi sudah mencapai nilai tertentu maka ujung akan bernilai nol dan posisi diubah menjadi nilai awal posisi.

6. Prosedur Interaksi antar Objek

Prosedur ini dibuat dengan memperhatikan posisi truk dan/atau mobil yang akan mengubah nilai dari variabel touch. Apabila posisi truk sama dengan posisi mobil maka dinyatakan menabrak dan mengubah touch menjadi true. Hal ini juga berlaku apabila truk berbelok hingga keluar dari jalur kemudian menabrak rumah atau pohon sehingga touch berubah menjadi true. Jika touch bernilai true maka akan menuju ke tampilan “Cie Nabrak :v”. Kemudian jika pemain mengklik tombol ‘r’ pada keyboard, maka akan me-restart game.

C. Hasil Eksekusi

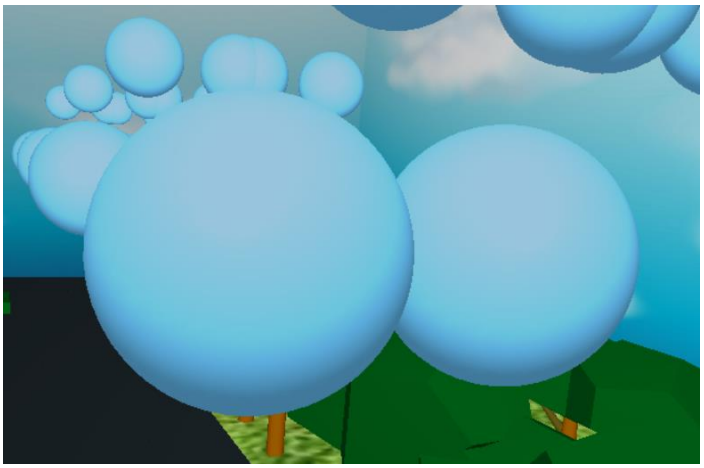
1. Tampilan awal



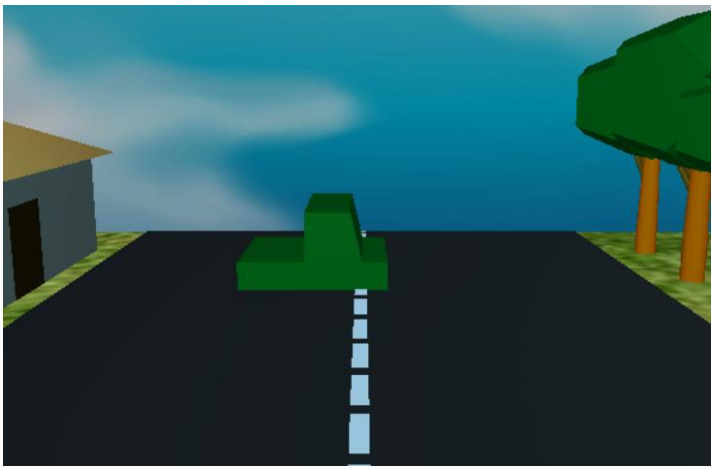
2. Tampilan Truck

3. Tampilan Rumah dan Pohon

4. Tampilan awan



5. Tampilan Mobil



6. Tampilan Ketika Nabrak

Link Youtube : https://youtu.be/ijiWGDce_u0

BAB IV
PENUTUP

A. Kesimpulan

Berdasarkan dari praktikum dan tugas yang telah kami lakukan, dapat disimpulkan sebagai berikut.

1. Pembuatan truk animasi ini dilakukan sepenuhnya dengan menggunakan OpenGL.
2. OpenGL merupakan standar API yang dapat digunakan untuk membuat grafis 2D maupun 3D.
3. Sementara GLUT merupakan library tambahan pada OpenGL yang digunakan untuk membuat truk animasi pada percobaan ini.
4. Pada pembuatan animasi ini, digunakan juga aspek aspek yang terdapat pada OpenGL seperti depth dan lighting, texture mapping, dan juga interaksi objek pada OpenGL.

B. Saran

Berdasarkan hasil praktikum ini, penyusun memiliki saran. Sebaiknya sebelum membuat animasi truk gandeng ini, harus terlebih dahulu memahami fungsi-fungsi yang terdapat pada OpenGL. Selain itu juga, pendeklarasian variabel pada percobaan ini sangat penting. Dengan adanya pendeklarasian variabel, program dapat dengan mudah memanggil nilai dari variabel tersebut. Harus memahami juga aspek-aspek lain seperti depth, lighting, dan object interaction secara matang untuk membantu pembuatan animasi lain menggunakan OpenGL nantinya.

LAMPIRAN

Pembagian Tugas :

1. Farrel Andhika Rizky Putra - 24060120130071

- render object (rumah, mobil, awan, pohon) + interaksi objek

2. Mohamad Fahrul Islami - 24060120130074

- animasi Truck + interaksi objek + laporan

3. Wasis Pambudi - 24060120120015

- laporan + interaksi objek + pergerakan lingkungan

DAFTAR PUSTAKA

Suhardiman, Deddy, dkk. 2012 " Pembuatan Simulasi Pergerakan Objek 3D (Tiga Dimensi) Menggunakan OpenGL ". https://ejournal.unsrat.ac.id/index.php/elekdankom/article/viewFile/595/467, diakses pada 21 Mei 2022 pukul 15.15.

Hidayatullah, Arif Rahman. 2014. " Praktikum Komputer Grafik ", https://www.scribd.com/doc/228980953/Laporan-Praktikum-Komputer-Grafik-dengan-OpenGL, diakses pada 19 Mei 2022 pukul 23.34.