

# Implementasi Open Gl Untuk Pembuatan Objek 3d

<sup>1</sup>Muhammad Adnani, <sup>2</sup>Achmad Zakki Falani

<sup>1</sup> Fakultas Ilmu Komputer Jurusan Teknik Informatika, Universitas Narotama Surabaya

<sup>2</sup> Fakultas Ilmu Komputer Jurusan Teknik Informatika, Universitas Narotama Surabaya

<sup>1</sup> [muhammadadenanii@gmail.com](mailto:muhammadadenanii@gmail.com), [muhammadadnani.17@narotama.ac.id](mailto:muhammadadnani.17@narotama.ac.id), <sup>2</sup> [zakki\\_falani@yahoo.com](mailto:zakki_falani@yahoo.com),  
[achmad.zakki@narotama.ac.id](mailto:achmad.zakki@narotama.ac.id)

**Abstract** - Abstracts Computer Graphics is a part of computer science that deals with the creation and manipulation of (visual) images digitally. A simple form of computer graphics is 2D computer graphics which later evolved into 3D computer graphics, image-based (image processing), and pattern recognition (pattern recognition). Computer graphics are often also known as data visualization. Graphics include images and other imagery generated by the computer in the form of lines. OpenGL (Open Graphics Library) is an API standard that can be used to create graphics-based applications, both two-dimensional (2D) and three-dimensional (3D). OpenGL is cross-platform, meaning that it can be run on various operating system platforms that exist today. using OpenGL, we first need a conception of the interface in its implementation of object protection. One method that is commonly used is to create a window-based OpenGL. To be able to conceptualize windowing in OpenGL, we need a certain tool. What we are using this time is GLUT (OpenGL Utility Toolkit).

**Keywords** : Implementation, OpenGL, 3D & 2D Objects, Rendering, GLUT, and Computer Graphics

**Abstrak**—Abstrak Grafika Komputer adalah bagian dari ilmu komputer yang berkaitan dengan pembuatan dan manipulasi gambar (visual) secara digital. Bentuk sederhana dari grafika komputer adalah grafika komputer 2D yang kemudian berkembang menjadi grafika komputer 3D, pemrosesan citra (image processing), dan pengenalan pola (pattern recognition). Grafika komputer sering dikenal juga dengan istilah visualisasi data. Grafik meliputi gambar dan pencitraan lain yang dihasilkan oleh komputer berbentuk garis. 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. menggunakan OpenGL, terlebih dahulu kita membutuhkan suatu konsepsi interfacing dalam implementasinya pada proteksi objek. Salah satu cara yang sudah umum digunakan adalah dengan membuat window-based OpenGL. Untuk dapat membuat konsep windowing pada OpenGL, kita memerlukan tool tertentu. Yang kita gunakan kali ini adalah GLUT (OpenGL Utility Toolkit).

**Kata Kunci**—Komponen; Implementasi, OpenGL, Objek 3 Dimensi & 2 Dimensi, Rendering, GLUT, dan Grafika Komputer

## I. Pendahuluan

Perkembangan zaman yang disertai oleh perkembangan teknologi komputer grafika dan pencitraan tiga dimensi saat

ini telah mengalami kemajuan yang sangat pesat dengan tingkat kualitas dan pencapaian yang cukup signifikan dalam dekade terakhir. Grafika komputer merupakan teknik dalam ilmu komputer dan matematika untuk merepresentasikan dan memanipulasikan gambar menggunakan komputer. Salah satu aplikasi yang komplit dari grafika komputer adalah untuk visualisasi data dalam bentuk grafika dua dimensi (2D) atau tiga dimensi (3D) dilengkapi dengan animasi.

Perlu dikembangkan sebuah software yang mampu merepresentasikan visual dalam format tiga dimensi (3D) secara real time dengan memasukan data atau parameter – parameter yang diperoleh dari suatu alat eksternal. Tanpa perlu proses rendering kesebuah file, melainkan langsung bisa di displaykan pada layer monitor.

Menggunakan OpenGL (Open Graphic Library) merupakan library yang terdiri dari berbagai macam fungsi dan digunakan untuk menggambar beberapa objek 2D dan 3D. library – library ini mendefinisikan sebuah Bahasa, cross platform API (antar muka pemrograman aplikasi) untuk menulis aplikasi yang menghasilkan objek 2D dan 3D grafis. Bahasa pemrograman yang digunakan pada umumnya adalah Bahasa C++.

## A. Perumusan Masalah

Berdasarkan latar belakang, maka dalam penelitian ini dapat dirumuskan beberapa masalah sebagai berikut :

- Bagaimana membuat grafika tiga dimensi (3D) dari suatu objek yang dapat dilihat dari berbagai sudut pandang.
- Bagaimana cara implementasi OpenGL dengan Bahasa pemrograman C++.

## B. Batasan Masalah

Berdasarkan latar belakang dan perumusan masalah serta pembatasan masalah, maka dalam penelitian ini dapat dirumuskan beberapa masalah sebagai berikut :

- Membuat objek 3D (tiga dimensi) dengan cara memvisualisasikan grafika atau gambar kebentuk tiga dimensi (3D).
- Menggunakan OpenGL sebagai library berbasis grafika tiga dimensi yang menggunakan Bahasa pemrograman C++.

- c. Menggunakan bahasa OpenGL\_QUADS, |GL\_Poligon, dan GL\_Line.
- d. Membuat objek interior kamar.

**C. Tujuan**

Ada beberapa tujuan yang ingin dicapai dalam penelitian ini adalah mengetahui cara membuat objek grafika tiga dimensi (3D) dari suatu objek yang dapat dilihat dari sudut pandang dan implementasikan OpenGL dengan bahasa pemrograman C++, cara untuk implementasikan objek 3 dimensi bisa dipakai untuk desain pada penginapan maupun untuk yang lainnya.

**D. Manfaat**

Manfaat yang diharapkan dapat diperoleh dari penelitian ini adalah memperoleh referensi sebagai bahan pembahasan grafika komputer. Dapat membuat objek tiga dimensi (3D), menggunakan Bahasa pemrograman C++. Menerapkan prinsip dasar grafikas untuk keperluan visualisasi.

**II. Metode Penelitian**

**A. Metode Penelitian**

Untuk dapat mengimplementasikan sistem ini nantinya, maka diperlukan langkah-langkah sebagai berikut:

1. Perumusan Masalah Berdasarkan perumusan masalah yang telah di dapat dari Bab I, maka akan diuraikan secara lebih terperinci tentang perumusan masalah yang ditemukan dalam penelitian ini. Oleh karena itu langka awal yang dilakukan dalam perumusan masalah adalah dengan melakukan studi literatur.
2. Studi Literatur Studi literatur yang dilakukan dalam penelitian ini adalah dengan memberbantingkan jurnal dan buku penelitian terdahulu yang menjadi acuan refensi pada penelitian ini. Perbantingan yang dilakukan adalah dengan mempelajari bagaimana tahapan proses yang dilakukan oleh penelitaan dari awal sampai akhir. Dimana hasil analisis dari pengujian data tersebut menjadi nilai perbandingan dan acuan refensi pada penelitian ini.
3. Anilisi Sistem. Secara umum proses untuk menghasilkan visualisasi dan animasi tiga dimensi (3D) dari suatu kasus simulasi adalah dengan cara membangun model obyek tiga dimensi 3D di sebuah software tiga dimensi 3D authoring, menganimasikan menggunakan metode Keyframe atau Motion Capture sesuai data atau parameter simulasi yang ada, menrender menjadi gambar atau video.
4. Implementasi Setelah tahap analisis system dilakukan pada tahap Implentasi penelitan dengan melakukan proses dalam bentuk coding dari program yang digunakan untuk membuat objek tiga dimensi (3D).

**B. GRAFIKA KOMPUTER**

Menurut Suyoto (2003), Grafika komputer (Computer Graphic) dapat diartikan sebagai seperangkat alat yang terdiri dari hardware dan software untuk membuat gambar, grafik atau citra realistik untuk seni, game komputer, foto dan film animasi. Sistem grafika komputer dapat dijalankan dengan komputer pribadi (Personal Computer) atau workstation. Grafika komputer semakin lama semakin pesat perkembangannya sehingga definisi dari grafika komputer dapat diartikan sebagai suatu studi tentang bagaimana menggambar (membuta grafik) dengan menggunakan komputer dan manipulasinya (merubah sedikit/transformasi/ animasi).

1. Cabang ilmu Grafik Pemrosesan citra (image processing) adalah proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Pengenalan pola (pattern recognition) dapat dikatakan sebagai kemampuan manusia mengenali objek-objek berdasarkan ciri-ciri dan pengetahuan yang pernah diamatinya dari objek-objek tersebut. Tujuan dari pengenalan pola ini adalah mengklasifikasi dan mendeskripsikan pola atau objek kompleks melalui pengetahuan sifat-sifat atau ciri-ciri objek tersebut.

**C. PENGENALAN OPENGL**

OpenGL adalah suatu spesifikasi grafikasi yang low level yang menyediakan fungsi untuk mempermudah pekerjaan atau untuk keperluan – keperluan pemrograman grafika (Graphics Programming / GP) Termasuk Grafika primitif (titik, garis, dan lingkaran) (Nicholas, 2011 : 541).

1. Singkatnya, Open Graphics Library, OpenGL menghilangkan kebutuhan untuk pemrogram untuk menulis ulang bagian grafis dari sistem operasi setiap kali sebuah bisnis akan diupgrade ke versi baru dari sistem. Fungsi dasar dari OpenGL adalah untuk mengeluarkan koleksi perintah khusus atau executable ke sistem operasi. PERINTAH DIOPENGL Perintah mengawali untuk menggambar. Mode merupakan konstanta yang menyatakan bagaimana opengl harus menghubungkan titik/vertex yang akan digambarkan.

Perintah	Arti
GL_POINTS	Titik
GL_LINE	Garis
GL_LINE_STRIP	Poligon
GL_LINE_LOOP	Polygon Tertutup
GL_TRIANGLES	Segitiga
GL_TRIANGLES_STRIP	Segitiga
GL_TRIANGLES_FAN	Segitiga
GL_QUADS	Segiti Empat
GL_QUAD_STRIP	Segiti Empat
GL_POLYGON	Polygon dengan pola tertentu
glVertex2i(x,y);	Lokasi titik berada di (x,y)

<code>glVertex2f(x,y);</code>	Lokasi titik berada di (x,y)
<code>glVertex3i(x,y,z);</code>	Lokasi titik berada di (x,y,z)
<code>glVertex3f(x,y,z);</code>	Lokasi titik berada di (x,y,z)
<code>glClearColor(R,G,B,a );</code>	Warna latar belakang
<code>glClearColor3f(R,G,B);</code>	Warna latar muka
<code>glColour4f(R,G,B,a );</code>	Warna latar muka
<code>glPointSize(k);</code>	Ukuran titik ke piksel
<code>glEnd();</code>	Akhir perintah OpenGL
<code>gl Lighting</code>	Cahaya Lokasi tertentu
<code>gl shading</code>	corak
<code>gl Proyeksi</code>	Sudut pandang terhadap objek tertentu

Tabel 2.2 Perinta OpenGL

#### D. Pengertian GLUT

Dikarenakan opengl tidak menyediakan interface sistem windwos atau pun input, maka untuk tujuan ini aplikasi harus menggunakan keseragaman spesifik platfrom. Dengan demikian jika aplikasi yang diinginkan tidak hanya berjalan di satu platfrom saja, maka perlu menggunkan GLUT. Glut merupakan pengembangan dari opengl yang didesain untuk aplikasi dengan level kecil hingga menengah dan menggunakan callback function untuk menambahkan interaksi dari user. Dua tujuan glut ialah menciptakan fleksibitas code flatfrom yang dapat dijalankan lebih dari satu sistem operasi (Windows, Linux, Mac OS X, FreeBSD, OpenBSD, NetBSD), dan untuk lebih

mudah mempelajari opengl. Dengan menggunakan Glut hanya memerlukan sedikit code dibantingkan dengan OpenGL, tanpa mengetahui spesifikasiopersai, dikarenakan opengl adalah sistem mesin.

#### E. PEMROGRAMAN BERBASIS EVENT

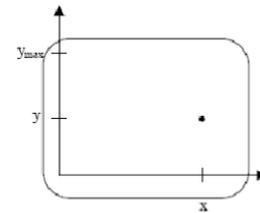
Glut pada dasar dibangun untuk menciptakan grafika menggunakan pemrograman yang bersifat procedural. Di dalam terdapat fungsi main loop yang diatur GLUT dan looping yang terjadi dalam aplikasi bertujuan untuk penanganan fungsi call back sebagai input dari user seperti fungsi redraw, mouse, keyboard, joystick, dan lain – lain Dibawah ini ialah tahap -tahap untuk pemrograman OpenGL menggunakan C++, diperlukan library tambahan yaitu :

1. Glut.h yang dicopy ke drive:\Program file (86X)\DevCpp\MinGW32\include\GL
2. Glut.lib yang dicopy ke drive;\Program file (86X)\DevCpp\MinGW32\lib.
3. Glut32.dll yang dicopy ke drive:\Windwos\system32.

#### F. SISTEM KORDINAT

Untuk menghasilkan suatu gambar pada paket pemrograman, yang pertama dibutuhkan untuk dilakukan adalah mendeskripsikan geometrik dari objek yang akan ditampilkan. Deskripsi ini menentukan lokasi dan bentuk dari objek. Untuk mendeskripsikannya dibutuhkan suatu standard yang dipakai kemudian ditetapkan standart yang dipakai menggunakan koordinat Cartesian. Koordinat Cartesian mereferensikan posisi koordinat pada kuadrant

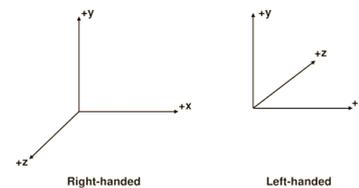
pertama dari dimensi dua. Yang menjadi titik acuan dari koordinat ini adalah pada sudut kiri bawah seperti pada gambar 2.2 Sedangkan koordinat yang dipakai pada layar adalah



Gambar 1 Kordinat Cartesian yang dimulai dari sudut kanan atas.

Dalam objek OpenGL, dikenal istilah transformasi. Transformasi merupakan suatu metode untuk mengubah lokasi suatu titik pembentukobjek, sehingga objek tersebut mengalami perubahan. Perubahan objek dengan mengubah koordinat dan ukuran suatu objek disebut dengan transformasi geometri. Dalam Transformasi dasar yang akan dibahas meliputi translasi, skala, dan rotasi. Pada grafika komputer, sistem koordinat suatu objek dapat di transformasi. Transformasi ini memungkinkan terjadinya perpindahan posisi suatu objek tanpa harus membuat model objek yang baru. Jenis-jenis transformasi yang sering digunakan pada grafika komputer dibagi menjadi 3 macam, yaitu translasi, rotasi, dan skalasi.

Pada sistem koordinat 3D harus diperhatikan orientasi dari sumbu x,y dan z. Grafika 3D mengenal 2 buah sistem koordinat, yaitu: a. Right-handed coordinat system (Sistem Koordinat Tangan Kanan) b. Left-handed coordinat system (Sistem Koordinat Tangan Kiri).

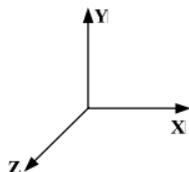


Gambar 2 Right-Handed dan Left-Handed.

Yang membedakan 2 dimensi dan 3 dimensi adalah kedalaman. Kedalaman didefinisikan sebagai jarak antara viewer terhadap benda yang dia lihat. Ini berarti berbeda dengan 2 dimensi yang hanya menggunakan 2 ukuran, yaitu panjang dan lebar, maka 3 dimensi menggunakan 3 ukuran, yaitu panjang, lebar dan kedalaman. Secara geometri ketiga ukuran tersebut disimbolkan dengan sumbu x, y, dan z.

digambar akan membentuk suatu sumbu koordinat mendarat yang sering dikenal dengan sumbu X dan sumbu koordinat tegak yang sering dikenal dengan

sumbu Y. Sedangkan dalam sistem koordinat 3 dimensi terdapat satu sumbu lain selain sumbu X dan sumbu Y, yaitu sumbu Z yang arahnya tegak lurus terhadap sumbu X dan sumbu Y. Untuk lebih jelasnya, bisa melihat gambar dibawah ini.



Gambar 3 sistem koordinat kartesian 3 dimensi

### G. Rendering

Untuk menghasilkan sebuah citra 2D dan 3D dibutuhkan sebuah proses dengan nama rendering. Tujuan dari proses ini adalah untuk memberikan visualisasi pada user mengenai data 3D tersebut melalui monitor atau pencetak yang hanya dapat menampilkan data 2D.

Gambar yang dibuat melalui proses rendering digital adalah gambar digital atau raster image, jenis gambar yang sama dengan yang biasa kita lihat sehari-hari pada desktop komputer atau wallpaper. Gambar digital tersebut dibuat melalui proses rendering digital sebagai langkah besar terakhir sebelum disusun menjadi animasi. Animasi sebagai tujuan akhir biasa digunakan dalam film, video game, permainan komputer, simulator, dan untuk efek khusus di televisi. Masing-masing menggunakan proses rendering digital yang menggunakan fitur dan teknik berbeda untuk mencapai hasil yang diinginkan. Secara umum, proses untuk menghasilkan rendering dua dimensi dari objek-objek 3D melibatkan 5 komponen utama yaitu :

1. Kamera
2. Cahaya
3. Karakteristik
4. Algoritma Rendering



Gambar 2.6 Skema Proses Rendering

### III. Hasil dan Pembahasan

Pada tahap ini adalah menerapkan semua proses yang telah dibahas pada pembahasan sebelumnya. Seperti halnya melakukan pengkodean pada coding opengl dengan menggunakan Library OpenGL dengan memasukan fungsi-fungsi bawaan yang telah disediakan. Seperti halnya pada gambar 3 berikut ini.

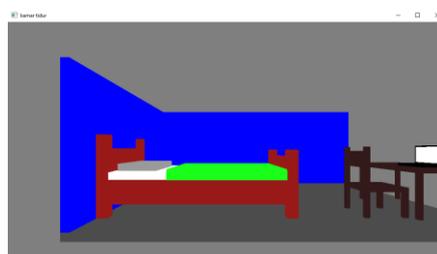
```

1 #include <cstdlib>
2 #include <iostream>
3 #include <windows.h>
4 #include <stdio.h>
5 #include <GL/gl.h>
6 #include <GL/glu.h>
7 #include <GL/glut.h>
8 #include <math.h>

```

Gambar 3.1. Pengkodean menggunakan OpenGL

Adapun pada tahap ini adalah menampilkan hasil yang telah diperoleh dari implementasi yang telah dilakukan. Bisa di lihat hasil dari gambar 3.2



Gambar 3.2 Hasil

Untuk Tampilan coding keseluruhan :

```

void display()
{
glClear(GL_COLOR_BUFFER_BIT
GL_DEPTH_BUFFER_BIT);
glLoadIdentity ();
glEnable(GL_DEPTH_TEST);
//perpindahan
glTranslatef(a,b,c);
//putaran
glRotatef(xpos,1,0,0);
glRotatef(ypos,0,1,0);
glRotatef(zpos,0,0,1);
//lantai
glPushMatrix();
glBindTexture(GL_TEXTURE, texture[1]);
glBegin(GL_QUADS);
glColor3f(0.3f,0.3f,0.3f);
kotak(0,0,18,25,0.5,0);
glEnd();
glPopMatrix();
//dinding kiri
glPushMatrix();
glBindTexture(GL_TEXTURE, texture[2]);
glBegin(GL_QUADS);
glColor3f(0.0f,0.0f,1.0f);
kotak(0,0.5,18,0.5,10,0);
glEnd();
glPopMatrix();
//dinding belakang
glPushMatrix();

```

```

glBindTexture(GL_TEXTURE, texture[2]);
glBegin(GL_QUADS);
glColor3f(0.0f,0.0f,1.0f);
kotak(0.5,0.5,0.5,25,10,0.0);
glEnd();
glPopMatrix();
//meja
//atas
glPushMatrix();
glBindTexture(GL_TEXTURE, texture[2]);
glBegin(GL_QUADS);
glColor3f(0.2f,0.1f,0.1f);
kotak(21,3.5,16,24.8,3.8,9.5);
//kaki kanan belakang
kotak(24.3,0.5,15.9,24.6,3.6,15.6);
//kaki kiri belakang
kotak(24.3,0.5,9.9,24.6,3.6,9.6);
//kaki kiri depan
kotak(21.1,0.5,9.9,21.4,3.6,9.6);
//kaki kanan depan
kotak(21.1,0.5,15.9,21.4,3.6,15.6);
glEnd();
glPopMatrix();

//pc
glPushMatrix();
glBindTexture(GL_TEXTURE, texture[2]);
glBegin(GL_QUADS);
glColor3f(0.0f,0.0f,0.0f);

//Monitor lcd
kotak(23.3,3.8,15.0,23.4,5.3,13.0);
//keyboard
kotak(23.3,3.8,15.0,22.4,13.0);
glEnd();
glPopMatrix();
//layar lcd
glPushMatrix();
glBindTexture(GL_TEXTURE, texture[8]);
glBegin(GL_QUADS);
glColor3f(1.1f,1.0f,1.0f);
kotak(23.29,4.05,15.0,23.30,5.2,13.1);
glEnd();
glPopMatrix();
//kursi
//alas
glPushMatrix();
glBindTexture(GL_TEXTURE, texture[3]);
glBegin(GL_QUADS);
glColor3f(0.2f,0.1f,0.1f);
kotak(18,2.2,15.5,20.8,2.5,12.8);
//kiri depan
kotak(20.5,0.5,15.5,20.8,2.5,15.2);
//kanan depan
kotak(20.5,0.5,13.1,20.8,2.5,12.8);
//kanan belakang
kotak(18,0.5,13.1,18.3,5.2,12.8);
//kiri belakang
kotak(18,0.5,15.5,18.3,5.2,15.2);
//senderan
kotak(18,3.8,15.5,18.2,4.9,12.8);
glEnd();
glPopMatrix();

//kasur
//rangka alas
glPushMatrix();
glBindTexture(GL_TEXTURE, texture[2]);
glBegin(GL_QUADS);
glColor3f(0.6f,0.1f,0.1f);
kotak(1.1,1.4,15.5,13.4,3.8,7);
//kaki 1
kotak(0.5,0.5,15.6,1.2,6,14.9);
//kaki 2
kotak(0.5,0.5,9.4,1.1,6,8.7);
//kaki 3
kotak(12.9,0.5,9.4,13.6,5,8.7);
//kaki 4
kotak(12.9,0.5,15.6,13.6,5,14.9);
//penyangga depan
kotak(0.6,3,15.4,1.5,1,8.9);
//penyangga belakang
kotak(12.9,1.4,15.4,13.6,4.5,8.9);
glEnd();
glPopMatrix();
//kasurnya
glPushMatrix();
glBindTexture(GL_TEXTURE, texture[1]);
glBegin(GL_QUADS);
glColor3f(1.1f,1.1f,1.1f);
kotak(1.1,2.5,15.4,13.4,3.5,9);
//bantal
glColor3f(0.6f,0.6f,0.6f);
kotak(1.3,3.5,14.4,4.4,10);
//selimut
glColor3f(0.1f,1.1f,0.1f);
kotak(5,3.5,15.4,13.4,3.7,9);
//kanan
kotak(5,3,9,13.4,3.6,8.9);
//kiri
kotak(5,3,15.5,13.4,3.6,15.4);
glEnd();
glPopMatrix();

```

#### IV. Kesimpulan

Kesimpulan dari hasil penelitian dijelaskan pada bagian ini

1. Untuk bias menampilkan interior kamar menggunakan tiga dimensi (3D) tidak selamanya

---

diperlukan proses rendering ke file terlebih dahulu, Implementasi Open GL bias langsung ditampilkan secara real time dengan bantuan library Open GL, Directx/Direct 3D. pada umumnya, hamper semua sofrware autoring Tiga Dimensi 3D memerlukan proses rendering yang cukup lama.

2. Dengan bantuan OpenGL atau DirectX/Direct3D bisa membantu mempercepat proses pengambilan keputusan dengan melihat visualisasi secara menyeluruh secara real time dalam format tiga dimensi (3D).
3. OpenGL merupakan status mesin dalam proses rendering dan atribut-atribut diubah melalui pemanggilan prosedur. Didesain untuk mengakomodasikan teknik rendering grafis tingkat lanjut.

## V. Daftar Pustaka

- [1] A. S. Riyadi, "Implementasi Open Gl32 Untuk Memanipulasi Gambar Segitiga Dan Segiempat," *Komput. Teknol. Inf.*, vol. 1, no. 1, 2016.
- [2] D. Suhardiman *et al.*, "Pembuatan Simulasi Pergerakan Objek 3D ( Tiga Dimensi ) Menggunakan OpenGL," *J. Inform.*, vol. 2, no. 1, hal. 1–6, 2015.
- [3] T. Ismail, S. Tinggi, M. Informatika, D. A. N. Komputer, dan S. A. Yogyakarta, "Visualisasi Tiga Dimensi ( 3D ) Real Time," 2010, [Daring]. Tersedia pada:  
[https://drive.google.com/file/d/0B3veF\\_xJ1onYS29wdmJUvG92bkU/view?pref=2&pli=1](https://drive.google.com/file/d/0B3veF_xJ1onYS29wdmJUvG92bkU/view?pref=2&pli=1).
- [4] M. B. Priyantono dan A. A. Rachmawan, "Implementasi Sistem Simulasi Penampilan Tata Surya Berbasis 3D Menggunakan Opengl," *J. Teknol. Inf.*, vol. 4, no. 1, 2020.
- [5] D. S. Hadi, *Modul grafika komputer*, no. November. 2014.
- [6] J. Simarmata dan T. Chandra, "Grafika Komputer," *Yogyakarta Penerbit Andi*, 2007.