**LAPORAN**

**TUGAS BESAR GRAFIKA KOMPUTER**

**MEMBUAT MENARA KINCIR ANGIN BELANDA 3D**

*Diajukan untuk Memenuhi Tugas Besar*

*Mata Kuliah Grafika Komputer*

Dosen : Hendri Karisma, S. Kom

Oleh:

Abdul Rojak Bilal - 10109362

Firdaus Musyafi - 10109366

Candy Nursadi - 10109397

Kelas IF-9/VIII



**PROGRAM STUDI S1**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS KOMPUTER INDONESIA**

**2013**

**KATA PENGANTAR**



*Assalaamu’alaikum wr. wb,*

Puji dan syukur penulis panjatkan kepada Allah SWT atas rahmat dan karunia-Nya yang telah dilimpahkan, shalawat dan salam tidak lupa dicurahkan kepada Nabi Muhammad SAW, sehingga penulis dapat menyelesaikan penyusunan Tugas Besar ini tepat pada waktunya dengan judul “**MEMBUAT MENARA KINCIR ANGIN BELANDA 3D**”.

Penyusunan Tugas Besar ini diajukan untuk memenuhi mata kuliah Grafika Komputer Program Strata Satu Jurusan Teknik Informatika Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia Bandung.

Sebagai manusia biasa penulis menyadari masih terdapat banyak kekurangan di dalam penulisan laporan Tugas Besar ini karena keterbatasan pengetahuan penulis. Oleh karena itu, saran dan kritik sangat penulis harapkan. Semoga laporan ini dapat bermanfaat bagi semua pihak yang memerlukan.

*Wassalaamu’alaikum wr. wb.*

Bandung, Juli 2013

Penulis

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Kincir angin adalah sebuah [mesin](http://id.wikipedia.org/wiki/Mesin) yang digerakkan oleh tenaga [angin](http://id.wikipedia.org/wiki/Angin) untuk menumbuk biji-bijian. Kincir angin juga digunakan untuk memompa [air](http://id.wikipedia.org/wiki/Air) untuk mengairi [sawah](http://id.wikipedia.org/wiki/Sawah). Kincir angin modern adalah mesin yang digunakan untuk menghasilkan energi [listrik](http://id.wikipedia.org/wiki/Listrik), disebut juga dengan [turbin angin](http://id.wikipedia.org/wiki/Turbin_angin). Turbin angin kebanyakan ditemukan di [Eropa](http://id.wikipedia.org/wiki/Eropa) dan [Amerika Utara](http://id.wikipedia.org/wiki/Amerika_Utara).

Menara Kincir Angin Belanda memiliki ukuran menaranya dengan tinggi 20 meter, diatas menara terdapat 4 baling-baling yang masing-masing memilki panjang 10 meter dan lebar 1,5 meter.

Objek 3D yang akan dibangun terdiri dari bangunan menara kincir angin nantinya dapat berputar atau baling-balingnya akan berputar layaknya kincir angin pada umumnya. Terdapat bangunan menara kincir angin terdapat rumah disampingnya dan ada sedikit sentuhan rerumputan serta pepohonan yang berada di sekitar bangunan kincir angin tersebut.

Penggambaran lansung dari lokasi menara kincir angin Belanda yang nantinya dituangkan dalam bentuk 3D. Dimana di sekitar menara kincir angin akan terdapat sungai dan jalan. Kincir angin dapat berputar layaknya kincir angin pada umumnya. desain warna sesuai dengan keadaan yang sesungguhnya sehingga akan terlihat lebih seperti aslinya. Kesesuaian penglihatan kamera dikontrol menggunakan beberapa key yang ada di keyboard.

* 1. **Perumusan Masalah**

Bagaimana membangun suatu objek 3D Menara Kincir Angin yang baling-balingnya berputar disertai dengan beberapa pepohonan, rerumputan, jalan dan sungai yang ada disekitarnya.

* 1. **Maksud dan Tujuan**

**1.3.1 Maksud**

Membangun suatu objek 3D Menara Kincir Angin dengan baling-baling yang berputar atau bergerak dengan menggunakan OpenGL.

**1.3.2 Tujuan**

Adapun tujuan dari pembuatan objek Menara Kincir Angin iniadalah sebagai berikut

1. Menampilkan bangunan menara kincir angin yang memiliki 4 baling-baling yang bisa berputar dan disekitarnya terdapat beberapa pepohonan, rumah, jalan, sungai dan jembatan.
2. Mengetahui fungsi-fungsi pada openGL dengan menggunakan empat unsur yaitu coloring, lighting, blending, dan mapping.
   1. **Batasan Masalah**

Agar pembahasan ,asalah tidak menyimpang dari objek pokok bahasan, maka batasan masalah dalam pembuatan perangkat lunak ini adalah sebagai berikut :

1. Hanya menampilkan menara kincir angin 3D,rumah, pepohonan, jalan, sungai dan jembatan.
2. Menampilkan pergerakan baling-baling kincir angin
3. Objek ditampilkan dalam 3 Dimensi
4. Menggunakan OpenGL.
5. Bahasa yang digunakan bahasa pemograman C++
   1. **Objek List**
6. Menara Kincir Angin : Bangunan akan dibuat menyerupai bangunan asli menara kincir angin yang ada di Belanda dan kincir angin bisa berputar
7. Rumah : Rumah dibuat disesuaikan dengan rumah aslinya berada tepat disamping menara kincir angin
8. Sungai : Sungai dibuat tepat pada didepan menara kincir angin dan rumah
9. Jembatan : Jembatan dibuat sesuai dengan aslinya yang memotong sungai atau menjadi akses penyebrangan jalan menuju kincir angin dan rumah
10. Jalan : Jalan dibuat selayaknya aslinya, jalan bersebelahan dengan sungai
11. Pohon : pohon akan dibuat ditepian jalan dan disekitar bangunan menara kincir angin
    1. **Objek Pembangun**

|  |  |
| --- | --- |
| **Bentuk Bangun** | **Penjabaran** |
| Bola | Bola digunakan untuk membuat pohon bagian atas atau bagian daun. Diibaratkan sebagai kumpulan daun yang bergumpal |
| Persegi | Persegi digunakan untuk membuat jendela |
| Persegi Panjang | Persegi panjang digunakan untuk membuat pintu, jembatan, jendela, dermaga, kincir angin, dll |
| Kerucut | Kerucut digunakan untuk membuat batang pohon |
| Balok | Balok digunakan untuk membuat Jembatan, rumah, atap rumah, menara kincir angin, dll |
| Kubus | Kubus digunakan untuk membuat menara kincir angin |

**BAB II**

**LANDASAN TEORI**

## Pengertian Grafika Komputer

Grafika komputer (*Computer graphics*) adalah bagian dari [ilmu komputer](http://id.wikipedia.org/wiki/Ilmu_komputer) yang berkaitan dengan pembuatan dan manipulasi gambar secara digital. Bentuk sederhana dari grafika komputer adalah [grafika komputer 2D](http://id.wikipedia.org/w/index.php?title=Grafika_komputer_2D&action=edit&redlink=1) yang kemudian berkembang menjadi [grafika komputer 3D](http://id.wikipedia.org/wiki/Grafika_komputer_3D), [pemrosesan citra](http://id.wikipedia.org/w/index.php?title=Pemrosesan_citra&action=edit&redlink=1) *(image processing),* dan [pengenalan pola](http://id.wikipedia.org/wiki/Pengenalan_pola) *(pattern recognition).* Grafika komputer sering dikenal juga dengan istilah [visualisasi](http://id.wikipedia.org/wiki/Visualisasi) data.

Bagian dari grafika komputer meliputi:

1. [Geometri](http://id.wikipedia.org/wiki/Geometri): mempelajari cara menggambarkan permukaan bidang
2. [Animasi](http://id.wikipedia.org/wiki/Animasi): mempelajari cara menggambarkan dan memanipulasi gerakan
3. [Rendering](http://id.wikipedia.org/w/index.php?title=Rendering&action=edit&redlink=1): mempelajari [algoritma](http://id.wikipedia.org/wiki/Algoritma) untuk menampilkan efek cahaya
4. [Citra](http://id.wikipedia.org/wiki/Citra) (Imaging) : mempelajari cara pengambilan dan penyuntingan gambar.

Teknik-teknik yang dipelajari dalam grafika komputer adalah teknik-teknik bagaimana membuat atau menciptakan gambar menggunakan komputer.Ada perbedaan yang sangat mendasar antara foto dan gambar, yaitu pada foto semua detail obyek terlihat sedangkan pada gambar (baik itu gambar manusia atau gambar komputer) tidak dapat memperlihatkan semua detail yang ada tetapi hanya detail-detail yang dianggap penting dalam menunjukkan pola suatu gambar.

## 2.2 Elemen Dasar Grafika

Ada beberapa elemen dasar dari grafika komputer antara lain:

1. Point

*Point* adalah sebuah titik yang digunakan untuk membangun obyek. Setiap titik dalam obyek 3 dimensi memiliki nilai dalam x, y dan z.

1. Polyline

*Polyline* adalah sebuah fungsi yang dibentuk dari beberapa garis yang saling berhubungan dan membentuk sebuh kurva yang terbuka.

1. Polygon

*Polygon* adalah suatu fungsi yang mirip dengan polyline hanya saja hasilnya adalah kurva tertutup, sedangkan *polyline* hasilnya kurva terbuka.

1. Filled Polygon ( Face )

*Filled Polygon* adalah sebuah *polygon* yang bagian dalamnya diwarnai atau dipenuhi dengan sebuah warna tertentu.*Filled polygon* biasanya digunakan sebagai *face* dari pembentukan obyek–obyek 3 Dimensi.

1. Gradate Polygon

*Gradate polygon* adalah sebuah *polygon* yang bagian dalamnya memiliki warna – warna yang bergradasi dari satu warna ke warna yang lainnya.

## 2.3 Grafik Komputer 2D

Grafik komputer 2D adalah pembuatan objek gambar dengan menggunakan 2 titik sebagai acuannya yaitu sumbu x dan y. Grafik 2D ini dapat digunakan dalam berbagai aplikasi yang pada awalnya dikembangkan pada teknologi cetak tradisional dan gambar, seperti tipografi, kartografi, gambar teknik, iklan, dan lain-lain.

Grafik komputer 2D ini merupakan langkah paling awal dalam membentuk model objek yang akan dibangun dalam grafik komputer 3D. Dalam aplikasi, gambar dua dimensi adalah bukan hanya representasi dari objek dunia nyata, tetapi sebuah artefak independen dengan nilai tambah semantik. Keseluruhan obyek 2D dapat dimasukkan dengan jumlah lebih dari satu, model yang akan dibentuk sesuai dengan kebutuhan. Tahap rekayasa hasil obyek 2D dapat dilakukan dengan aplikasi program grafis seperti Adobe Photoshop, Corel Draw, dan lain sebagainya.

## 2.4 Grafik Komputer 3D

Grafik komputer 3D merupakan representasi dari data geometrik 3 dimensi sebagai hasil dari pemrosesan dan pemberian efek cahaya terhadap grafik komputer 2D.hasilnya dapat ditampilkan secara real time untuk keperluan simulasi. Prinsip yang dipakai mirip dengan grafik komputer 2D dalam penggunaan algoritma, grafika vektor, model frame kawat (wire frame model), dan grafik rasternya.

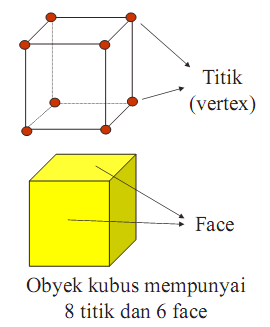
Grafik komputer 3D sering disebut sebagai model 3D. Namun, model 3D ini lebih menekankan pada representasi matematis untuk objek 3 dimensi. Obyek pada grafik 3D adalah sekumpulan titik-titik 3D (x,y,z) yang membentuk suatu face (bidang) yang digabungkan menjadi satu kesatuan. Face sendiri adalah gabungan titik-titik yang membentuk bidang tertentu. Data matematis ini belum bisa dikatakan sebagai gambar grafis hingga saat ditampilkan secara visual pada layar komputer atau printer. Proses penampilan suatu model matematis ke bentuk citra 2 D biasanya dikenal dengan proses 3D rendering.

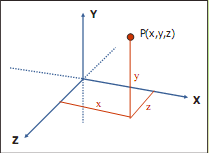
## 2.5 Perbedaan Grafik 2D dan 3D

Perbedaan yang paling mendasar dan terlihat dengan sangat jelas adalah tampilan gambarnya. Gambar 2D tampil flat adn frame tampilannya cenderung terbatas karena objek gambarnya disajikan hanya dengan sumbu x dan y. Sedangkan pada grafik 3D, gambar yang ditampilkan lebih hidup, membentuk ruang, tidak flat, serta framenya lebih luas yang dikarenakan gambar 3D disajikan dengan 3 sumbu, yaitu x, y, dan z.

* 1. **Definisi Objek 3 Dimensi**

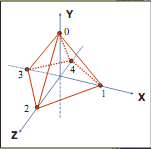
Obyek 3-D adalah sekumpulan titik-titik 3-D (x,y,z) yang membentuk luasan-luasan (face) yang digabungkan menjadi satu kesatuan. Face adalah gabungan titik-titik yang membentuk luasan tertentu atau sering dinamakan dengan sisi.





Sistem Koordinat 3 Dimensi

**Contoh pernyataan Objek : Limas Segiempat**

Titik-titik yang membentuk obyek:

Titik 0 (0,150,0)

Titik 1 (100,0,0)

Titik 2 (0,0,100)

Titik 3 (-100,0,0)

Titik 4 (0,0,-100)

## Pengertian OpenGl

OpenGL adalah API (Application Programing Interface) yang dikenalkan oleh SGI (Silicon Graphics Inc) kali pertama pada tahun 1980-an. Awalnya OpenGL diciptakan untuk mendukung proses rendering, pada saat itu hanya digunakan untuk mendukung proses redering yang dilakukan oleh komputer graphic produksi SGI. Namun akhirnya, OpenGL dijadikan standar oleh berbagai perusahaan software dan hardware.

open GL (program java for openGL). adalah bahasa yang dapat dijalankan dimanapun dan di sembarang platform apapun, diberagam lingkungan : internet, intranets, consumer electronic products, dan computer applications. Bahasa pemrograman berorientasi objek telah menjadi aliran utama (mainstream), java benar-benar berorientasi objek sejati, melebihi C++.Segala sesuatu dijava kecuali sedikit type dasar (int, float, double, char) adalah objek.

* 1. **Fungsi OpenGL 3D**

1. **Membersihkan *Windows***

Pada komputer, *memory* untuk menampilkan gambar biasanya diisi dengan gambar yang berasal dari perintah gambar paling akhir, jadi perlu dibersihkan dengan warna latar belakang sebelum digambar lagi. Contoh berikut ini perintah yang digunakan untuk membersihkan layar latar belakang dengan warna hitam dan *buffer* apa yang akan dibersihkan. Dalam hal ini, *buffer* warna yang akan dibersihkan karena *buffer* warna merupakan tempat gambar disimpan.

glClearColor 0, 0, 0, 0

glClear GL\_COLOR\_BUFFER\_BIT Or GL\_DEPTH\_BUFFER\_BIT

1. **Spesifikasi Warna**

Pada OpenGL mendeskripsikan objek dengan warna objek adalah proses yang berjalan sendiri-sendiri. Sebelum warna diubah maka semua objek yang digambar sesudah perintah tersebut akan menggunakan warna terakhir yang terdapat pada *coloring scheme.* Untuk warna digunakan perintah *glColor3f*. Contoh berikut menunjukkan urutan langkah dalam proses spesifikasi warna sebelum objek digambar. Warna yang ditampilkan adalah warna merah.

glColor3f 1, 0, 0

1. **Memaksa Proses Menggambar Sampai Selesai**

Kebanyakan sistem grafik modern sudah menggunakan sistem *graphics pipeline*. Dimana CPU utama memberikan *issue* perintah menggambar dan hardware lain yang melakukan *transformasi,* *clipping*, *shading*, *texturing* dan lain-lain. Pada arsitektur yang demikian, proses tidak dilakukan pada satu computer karena setiap komputer mempunyai tugas sendiri. CPU utama tidak harus menunggu proses pada masing-masing komputer tadi selesai, tapi bisa dengan memberikan *issue* perintah gambar yang berikutnya. Untuk inilah OpenGL menyediakan perintah *glFlush* yang memaksa *client* untuk segera mengirim paket *network* walaupun belum penuh. Program sebaiknya ditulis menggunakan perintah ini karena *glFlush* tidak memaksa proses gambar untuk selesai tetapi memaksa proses gambar untuk segera dieksekusi, sehingga dijamin semua perintah gambar yang sebelumnya akan segera dieksekusi dalam suatu waktu tertentu.

GlFlush

1. **Fungsi untuk keperluan Transformasi**
2. **Fungsi Translasi (Translation)**

Translasi merupakan bentuk transformasi yang **memindahkan** posisi suatu objek, baik pada sumbu x, sumbu y, atau sumbu z. Fungsi yang digunakan untuk melakukan translasi adalah :

glTranslatef Tx, Ty, Tz

Parameter *Tx* digunakan untuk menentukan arah dan seberapa jauh suatu benda akan dipindahkan berdasarkan sumbu x. Parameter *Ty* digunakan untuk menentukan arah dan seberapa jauh suatu benda akan dipindahkan berdasarkan sumbu y. Sedangkan parameter *Tz* digunakan untuk menentukan arah dan seberapa jauh suatu benda akan dipindahkan berdasarkan sumbu z.

Contohnya :

glTranslatef 2, 2, 2

1. **Fungsi Rotasi (Rotation)**

Rotasi merupakan bentuk transformasi yang digunakan untuk **memutar** posisisuatu benda. Fungsi yang digunakan untuk melakukan rotasi ialah **glRotatef(θ, Rx, Ry, Rz)**

Parameter yang dibutuhkan pada fungsi tersebut ada 4 macam, yaitu parameter **θ** untuk besar sudut putaran, parameter ***Rx*** untuk putaran berdasarkan sumbu x, parameter ***Ry*** untuk putaran berdasarkan sumbu y, dan parameter ***Rz*** untuk putaran berdasarkan sumbu z. Jika parameter **θ** bernilai postif, maka objek akan diputar berlawanan arah jarum jam. Sedangkan jika parameter **θ** bernilai negatif, maka objek akan diputar searah jarum jam. Contohnya :

**glRotatef -30, 2, 2, 1**

1. **Fungsi Skala (Scalling)**

Skalasi merupakan bentuk transformasi yang dapat mengubah ukuran **(besar-kecil)** suatu objek. Fungsi yang digunakan untuk melakukan skalasi ialah :

**glScalef(Sx, Sy, Sz)**

Perubahan ukuran suatu objek diperoleh dengan mengalikan semua titik atau atau vertex pada objek dengan faktor skala pada masing-masing sumbu (parameter *Sx* untuk sumbu x, *Sy* untuk sumbu y, dan *Sz* untuk sumbu z).

Contohnya :

**glScalef(2, 2, 2)**

## Lighting (pencahayaan)

Lighting merupakan proses menghitung intensitas cahaya terutama pada 3-Dimensi point, biasanya diatas suatu permukaan.

Beberapa cara mengatasi masalah pencahayaan, antara lain :

* Mengerti persepsi dari cahaya (warna)
* Membuat sebuah solusi untuk merepresentasikan dan menghasilkan warna menggunakan komputer.
* Mengerti akan pengaruh cahaya dan objek

**` Bayangan**

* Bayangan akan muncul saat cahaya jatuh menyinari suatu objek.
* Pada dunia maya, layaknya cahaya, terdapat beberapa jenis bayangan yang dapat dihasilkan oleh komputer.

Bayangan bekerja sama dengan cahaya untuk memberi kesan natural atau realistic pada scene yang ada. Bayangan dapat membantu mendefinisikan posisi objek-objek, apakah berada di lantai atau melayang di udara.Bayangan yang dihasilkan bisa tajam dan solid namun bisa juga lembut dan buram (blurry).Keberadaan bayangan atau ketiadaannya dapat digunakan untuk memberi keseimbangan dan kontras pada objek-objek di dalam scene.

## Texture 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. Dalam pemberian tekstur, perlu diperhatikan dasarnya seperti:

1. Menentukan tekstur
2. Membaca atau membangkitkan tekstur.
3. Menandai tekstur.
4. Mengenablekan tekstur.
5. Menandai koordinat tekstur pada vertek.
6. Menentukan parameter tekstur seperti : Wrapping , filtering, dsb.
   1. **Definisi Dev C++**

Dev C++ adalah sebuah produk IDE untuk bahasa pemrograman C dan C++ yang dikembangkan Microsoft. Bahasa C atau C++ adalah suatu bahasa pemrograman. Bahasa C termasuk sebagai bahasa pemrograman tingkat menengah, maksudnya bahasa C bisa dipelajari dengan lebih mudah karena mudah dimengerti tetapi mempunyai kemampuan yang tinggi.

Bahasa C bisa digunakan untuk merekayasa program untuk segala kebutuhan, baik untuk aplikasi bisnis, matematis atau bahkan game. Semua bahasa mempunyai kelemahan atau kelebihan sendiri-sendiri. Begitu juga dengan bahasa C. Adapun sebagian kelebihan dari bahasa C adalah sebagai berikut :

1. Banyak memiliki operator untuk mengolah / memanipulasi data.
2. Bahasa C termasuk sebagai bahasa yang terstruktur sehingga program dapat
3. lebih mudah dipahami atau dikembangkan.
4. Bahasa C lebih mudah dimengerti karena lebih mirip kepada bahasa manusia.
5. Kecepatan eksekusi tinggi.
6. Mengenal data pointer.

Sedangkan kelemahan dari bahasa C adalah :

1. Banyaknya operator atau cara penulisan program kadang menimbulkan
2. kebingungan para pemakainya.
3. Perlunya ketelitian dalam penulisan program karena perintah (*statement*)
4. dalam bahasa C bersifat *case sensitiv* (huruf kapital dan huruf kecil dibedakan

**BAB III**

**IMPLEMENTASI dan HASIL**

* 1. **Implementasi**

Pembuatan objek yang kami lakukan dengan pemanggilan fungsi-fungsi umum yang sudah tersedia di library OpenGL itu sendiri. Berikut objek-objek yang kami panggil :

* + 1. **Rincian Aplikasi**
       1. Terain yang dibuat terdiri dari terrain tanah, terrain air dan terrain jalan raya
       2. Menara kincir angin dibuat dengan mengabungakan kubus (*cube*) dengan balok yang *solid* di transformasikan dan di rotasi *.* Sama halnya dengan kincir angin sendiri menggunakan kubus balok tetapi dengan di *rotasi.* kincir bisa berputar ada *fuction* nya yang menjadi sudut rotasi berubah ubah
       3. Rumah, jembatan, dermaga juga dibuat dengan mengabungakan kubus (*cube*) dengan balok yang *solid* yang di transformasikan dan di rotasi sedemikian rupa
       4. Pohon-pohonan menggabungkan kerucut (*cone*) sebagai batang pohon dan lingkaran (*sphere*) sebagai dedaunananya yang di transformasikan dan di rotasi sedemikian rupa sehingga terlihat seperti pohon
       5. Batu dibuat dengan lingkaran (*sphere*)
    2. **Perangkat Lunak Pendukung**

Perangkat lunak yang dipasang pada sistem komputer yang digunakan untuk membangun aplikasi ini adalah sebagai berikut :

Sistem Operasi *Windows* 7.

Dev C++ v.4.9.9.2

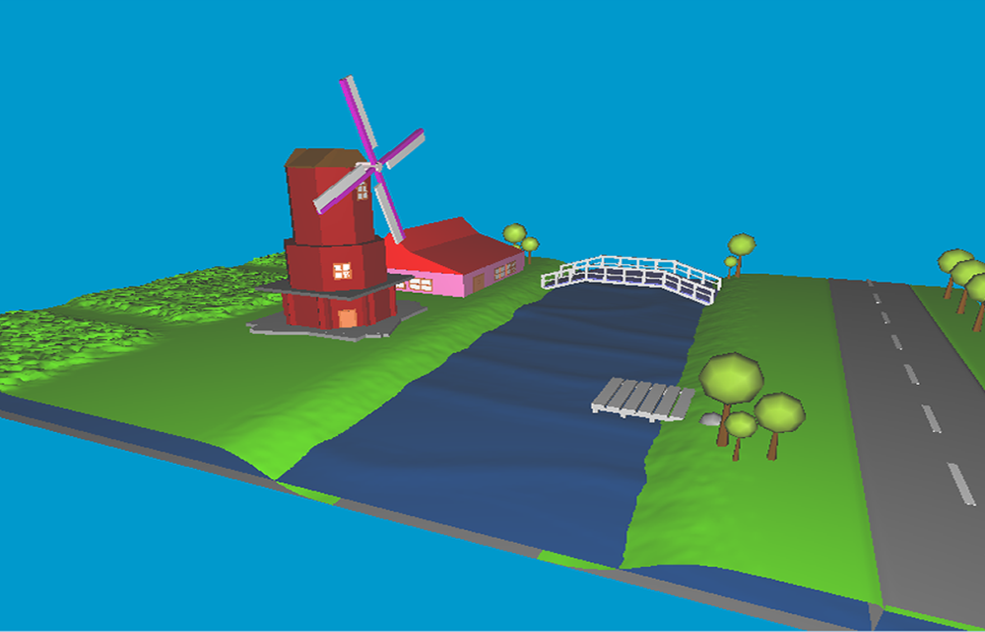
Glut32 dll

* + 1. **Perangkat Keras Pendukung**

Kebutuhan minimal perangkat keras (*hardware*) yang diperluka untuk mengimplementasikan program aplikasi ini adalah sebagai berikut :

* + - 1. Prosesor 1.60 Ghz
      2. Ram 2 Gb
      3. VGA 128MB
      4. Keyboard dan Mouse
  1. **Hasil Tampilan Aplikasi**

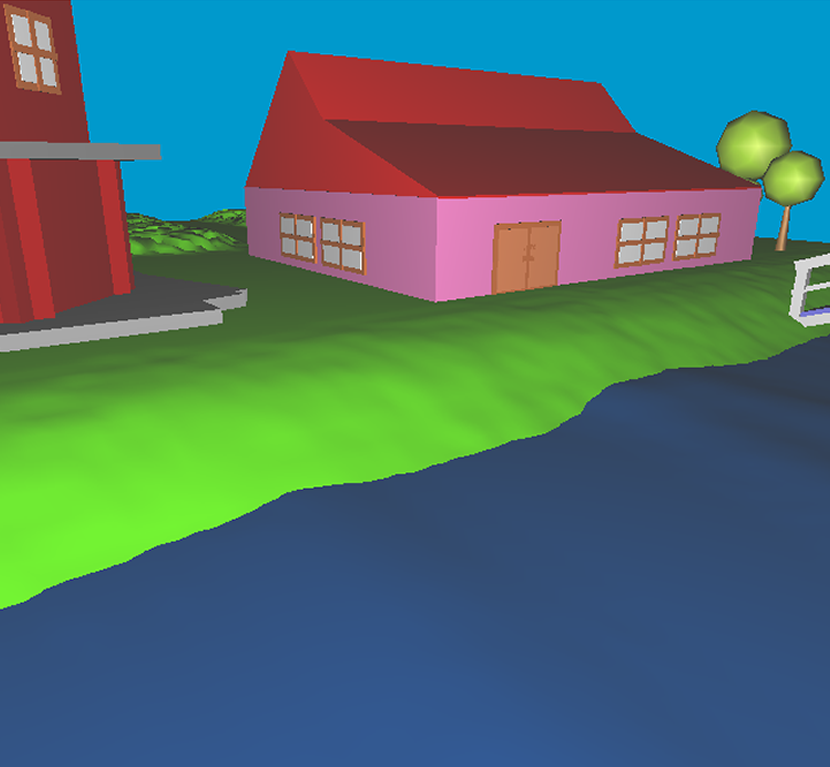
**3.2.1 Gambar Tampilan Awal Aplikasi Menara Kincir Angin Belanda**

****

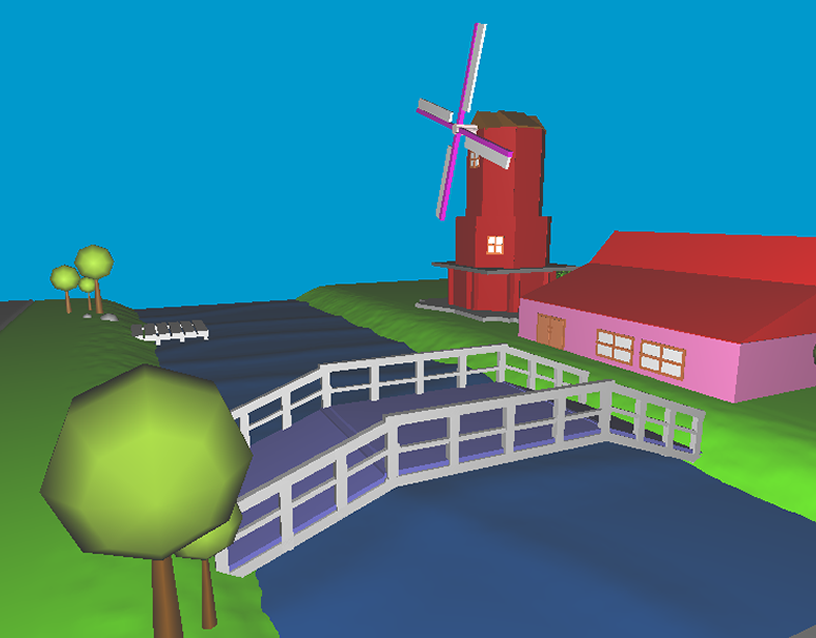
**3.2.2 Gambar Tampilan Kincir Angin Belanda**

****

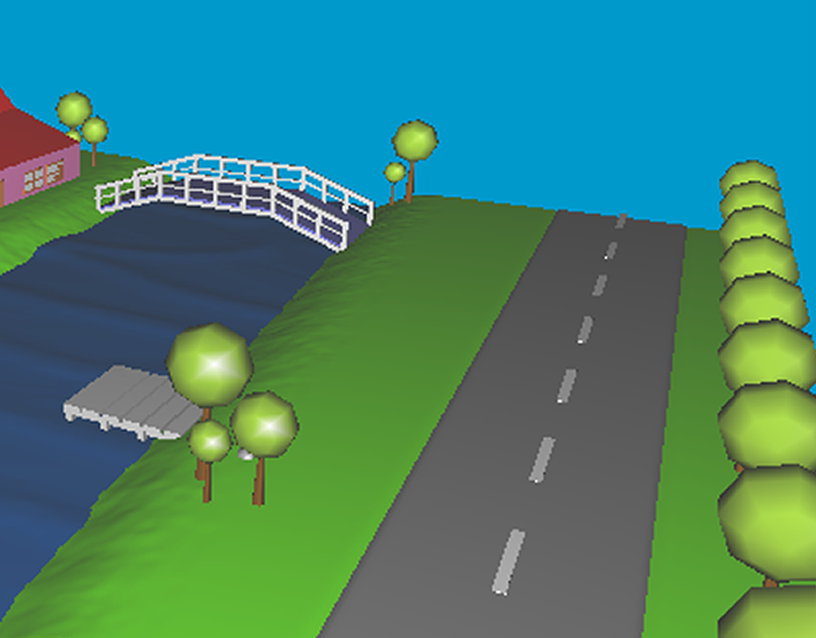
**3.2.3 Gambar Tampilan Rumah**

****

**3.2.4 Gambar Tampilan Jembatan**

****

**3.2.5 Gambar Tampilan Jalan dan Pohon**

****

**BAB IV**

**PENUTUP**

## Kesimpulan

**Berdasarkan sistem yang telah dikembangkan dan hasil pengujian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut :**

1. Dengan menggunakan Dev C++ dan library OpenGL dapat membantu membuat animasi Menara Kincir Angin Belanda 3D dengan baik
2. Mengetahui fungsi-fungsi pencampuran lighting, color, transformasi dan mapping.
3. Dapat melihat secara langsung perubahan yang terjadi pada objek.
4. Format gambar yang dapat digunakan untuk texturing adalah BMP

## Saran

**Ada beberapa hal yang dapat dijadikan saran dalam proses pembuatan animasi pesawat terbang, antara lain :**

1. Pemanfaatan Lightning dengan lebih baik, agar object dapat terlihat dengan lebih baik
2. Penggunaan gambar selain bitmap, agar file yang digunakan bisa lebih ringan
3. Pengefisienan coding agar lebih mudah dalam mengembangkan objek

**DAFTAR PUSTAKA**

<http://id.wikipedia.org/wiki/Grafika_komputer_3D>

<http://risaagustiana.blogspot.com/2012/10/seputar-grafik-komputer.html>

[http://badcoding.wordpress.com](http://badcoding.wordpress.com/)

<http://aflah7.wordpress.com/2010/10/14/konsep-pemodelan-grafik-2d-dan-3d/>