**MEMBUAT MODEL 3D BANDARA**

Diajukan untuk Memenuhi Tugas Besar Mata Kuliah Grafika Komputer



**Oke Triyana 10109365**

**Asyer Yulian Kalo 10109378**

**Raju Riyanda 10109399**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS KOMPUTER INDONESIA**

**BANDUNG**

**2013**

# KATA PENGANTAR

*Bismillahirohmanirrohim*

Segala puji dan syukur yang tak terhingga penulis panjatkan kehadirat Allah SWT, yang selalu memberikan rahmat dan karunia-Nya kepada penulis sehingga proposal ini dapat selesai.

Laporan yang berjudul **” *Membuat Model 3D Bandara ”*.** Laporan ini juga dibuat untuk memenuhi tugas besar mata kuliah Komputer Grafika semester 8.

Penulis menyadari sepenuhnya laporan ini masih jauh dari sempurna, oleh karena itu penulis sangat mengharapkan kritik dan saran yang bersifat membangun untuk perbaikan di masa yang akan datang.

*Billahitaufiq Walhidayah, Wassalamu’alaikum Wr. Wb.*

Bandung, Juli 2013

Penulis

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang Masalah**

Bandara, dimana didalamnya terdapat bangunan bandara, landasan pesawat untuk lepas landas dan mendarat, lampu landasan yang ada dipinggiran landasan untuk menerangi landasan di malam hari, pohon yang berada di bandara agar suasana terlihat sejuk, dan langit yang cerah berawan dan matahari, dan pesawat yang akan lepas landas.

* 1. **Perumusan Masalah**

Berdasarkan pemaparan pada latar belakang masalah, maka dapat disimpulkan beberapa masalah :

1. Bagaimana memahami dan menerapkan konsep komputer grafika dalam aplikasi 3D.
2. Bagaimana memahami dan menerapkan fungsi-fungsi OpenGL dalam aplikasi 3D.
   1. **Maksud dan Tujuan**

Untuk mengimplementasikan konsep komputer grafika dan fungsi OpenGL, maka kami bermaksud membuat model 3D Bandara.

Tujuan dari aplikasi ini adalah :

1. Memahami konsep komputer grafika dan fungsi OpenGL dalam model 3D.
2. Menerapkan konsep komputer grafika dan fungsi OpenGL sehingga lebih mengerti.
   1. Batasan Masalah

Adapun batasan masalahnya sebagai berikut:

1. Bahasa yang digunakan bahasa pemrograman C++.
2. IDE yang digunakan Dev C++ dengan library openGL
3. Informasi yang disampaikan hanya mengenai bangunan bandara, landasan pesawat, pohon, awan, matahari, dan pesawat
   1. Object List
4. Bangunan Bandara : Bangunan akan dibuat menyerupai bangunan asli dari sebuah bandara.
5. Ladasan Pesawat : Landasan pesawat dibuat menyerupai aslinya.
6. Lampu Landasan : Lampu landasan dibuat dipinggiran landasan pesawat.
7. Pohon : Pohon akan dibuat di sekitar bangunan bandara.
8. Awan : Awan akan dibuat untuk menambah estetika dari sebuah konsep alam.
9. Matahari : Matahari akan dibuat untuk menyerupai keadaan sebenarnya.
10. Pesawat : Pesawat akan dibuat menyerupai aslinya dan akan bergerak untuk lepas landas.
    1. Objek Pembangun
11. Bangunan Bandara : Bangunan terdiri dari beberapa gabungan kubus sebagai tembok, jendela, dan pintu dari bangunan bandara dan kerucut sebagai atap dari sebuah bandara sehingga menyerupai bentuk aslinya.
12. Ladasan Pesawat : Landasan terdiri dari persegi panjang dan ditengahnya terdapat garis putih untuk menyerupai landasan pacu.
13. Lampu Landasan : Lampu landasan akan dibuat dengan menggunakan sphere berwarna oranye.
14. Pohon : Pohon terdiri dari tabung sebagai batang dan sphere berwarna hijau sebagai daun-daunnya.
15. Awan : Awan akan dibuat dengan menggunakan sphere berwarna biru muda.
16. Matahari : Matahari akan dibuat dengan menggunakan sphere yang berwarna kuning.
17. Pesawat : Pesawat dibuat dengan menggunakan campuran objek silinder untuk membuat badan pesawat dan bagian depan pesawat.Sedangkan sayap, ekor dan sejenisnya akan dibuat dengan menggunakan gabungan garis vertex.
    1. Teknologi

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 :

* Banyak memiliki operator untuk mengolah / memanipulasi data.
* Bahasa C termasuk sebagai bahasa yang terstruktur sehingga program dapat
* lebih mudah dipahami atau dikembangkan.
* Bahasa C lebih mudah dimengerti karena lebih mirip kepada bahasa manusia.
* Kecepatan eksekusi tinggi.
* Mengenal data pointer.
* Sedangkan kelemahan dari bahasa C adalah :
* Banyaknya operator atau cara penulisan program kadang menimbulkan
* kebingungan para pemakainya.
* Perlunya ketelitian dalam penulisan program karena perintah (*statement*)
* dalam bahasa C bersifat *case sensitiv* (huruf kapital dan huruf kecil dibedakan

## Open Gl

OpenGL adalah suatu graphic library yang sebagian bersifat open source, dipakai pada banyak platform (windows, linux) dan dapat digunakan pada berbagai jenis compiler seperti C++ atau Delphi. OpenGL bukanlah bahasa pemrograman tetapi merupakan suatu Application Programming Interface (API).

**BAB II**

**LANDASAN TEORI**

## 2.1 Objek 3 Dimensi

Objek tiga dimensi adalah sebuah model struktur data yang menyatakan suatu gambar 3D dibentuk dan disusun. Objek 3Ddidefinisikan dengan :

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

Dari definisi ini, dapat dikatakan bahwa objek 3D merupakan kumpulan titik-titik dan kumpulan face yang merupakan susunan dari titik-titik yang ditentukan. Seperti gambar kubus, kubus terdiri dari 8 titik dan 6 sisi/*face*.Dimana *face* merupakan polygon atau kumpulan titik-titik yang disusun urutannya. Dalam kubus, *face-*nya semua berupa bujursangkar dengan 4 titik.

## 2.2 Bitmap

Bitmap adalah representasi atau gambaran yang terdiri dari baris dan kolom pada titik image graphics di komputer. Nilai dari titik disimpan dalam satu atau lebih data bit. Tampilan dari bitmap atau raster, menggunakan titik-titik berwarna yang dikenal dengan sebutan pixel. Pixel-pixel tersebut ditempatkan pada lokasi-lokasi tertentu dengan nilai-nilai warna tersendiri, yang secara keseluruhan akan membentuk sebuah tampilan gambar. Tampilan bitmap mampu menunjukkan kehalusan gradasi bayangan dan warna dari sebuah gambar, karena itu bitmap merupakan media elelktronik yang paling tepat untuk gambar-gambar dengan perpaduan gradasi warna yang rumit seperti foto dan lukisan digital.

Struktur bitmap terdiri dari Header, Info Header dan Color Tabel. Header adalah bagian dari file bitmap yang berisi informasi header dari file gambar bitmap. Ukuran dari header ini 14 byte, masing-masing terdiri dari signature 2 bytes (berisi “BM” sebagai tanda gambar mempunyai format bmp), FileSize 4 bytes (besarnya ukuran gambar mempunyai satuan bytes), Reserved 4 bytes (tidak digunakan atau sama diisi dengan nilai nol) dan DataOffset 4 bytes (file offset untuk raster data). Info header adalah bagian dari header yang berisi informasi lebih detail dari file gambar bitmap. Color table adalah table yang berisi warna-warna yang ada pada gambar bitmap.

## 2.3 Lighting (Diffuse, Ambient dan Specular)

Proses menghitung intensitas cahaya terutama pada 3D point, biasanya diatas suatu permukaan :

* Bayangan

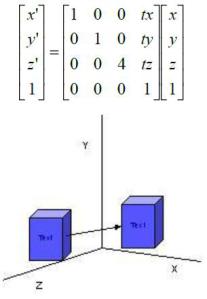
Bayangan akan muncul saat cahaya jatuh menyinari suatu objek.Pada dunia maya, layaknya cahaya, terdapat beberapa jenis bayangan yang dapat dihasilkan oleh komputer.

* Jenis Bayangan

Pada Maya, suatu sumber cahaya bisa tidak menghasilkan bayangan (*default*) atau bisa menghasilkan bayangan *depthmap* maupun *raytraced*. Anda dapat mengombinasikan kedua jenis bayangan ini *depthmap* maupun *raytraced* pada *scene* Anda. Dengan mengatur atribut bayangan *depth map* atau *raytraced*, Anda dapat mensimulasikan Bayangan yang dihasilkan oleh berbagai tipe cahaya di dunia nyata. Bayangan *depth map* maupun *raytraced*memberikan efek yang hamper sama, namun bayangan depth map waktu *render*-nya lebih cepat. Umumnya kebanyakan orang akan memilih bayangan depth map kecuali jika tipe bayangan tersebut tidak dapat membantu mencapai visualisasi yang diinginkan.

## 2.4 Translation

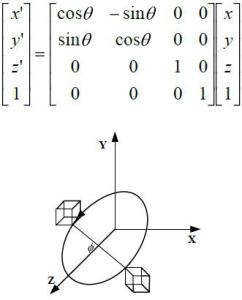
Sebuah objek dalam tiga dimensi di translasikan dengan mengubah setiap setiao point yang mendefinisikan objek. Untuk sebuah objek yang dibangun atau direpresentasikan dengan sebuah set dari permukaan – permukaan poligon , translasi dilakukan pada setiap permukaan dan menggambar kembali permukaan poligon pada posisi baru.



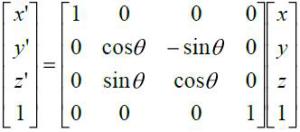
## 2.5 Rotation

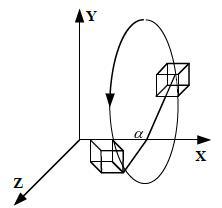
Untuk membangkitkan rotasi pada objek 3D kita harus membuat aksis dari rotasi dan jumlah sudut rotasi . Tidak seperti melakukan rotasi pada objek 2D yang semua proses transformasi dilakukan di koordinat xy , sebuah rotasi objek tiga dimensi bisa dilakukan di *space* manapun. Dengan menggunakan notasi matrix, maka besaran R bisa dikatakan sbb:

Rotasi Berdasarkan Sumbu Z :

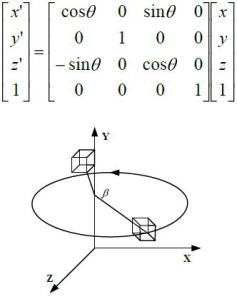


Rotasi Berdasarkan Sumbu X :



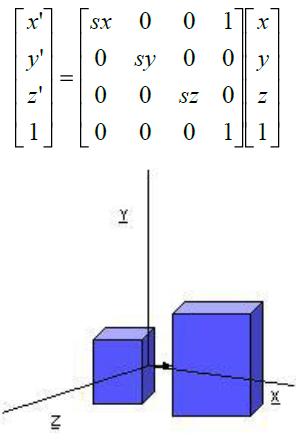


Rotasi Berdasarkan Sumbu Y :



## 2.6 Scaling

Penskalaan pada objek 3D dengan transformasi mengubah ukuran dan posisi objek relatif terhadap koordinat asli. Jika tidak semua parameternya sama dimensi – dimensi relative pada objek akan diubah. Matriks transformasi untuk skala dapat juga dinyatakn sebagai berikut :



**BAB III**

**ANALISIS DAN PERANCANGAN**

* 1. **Analisis Sistem**

Sistem yang akan dibangun adalah sebuah aplikasi yang menghasilkan efek – efek dari *lighting, color, tranformasi, dan mapping*. Pada bab I telah dijelaskan mengenai batasan masalah di dalam pengimplementasian aplikasi yang dimaksud.

Karena proses penggambaran dilakukan oleh hadware, maka dibutuhkan bahasa pemrograman yaitu menggunakan bahas pemrograman Dev C++. Adapun kebutuhan software dan hadwarenya akan dijelaskan lebih lanjut.

### 3.2 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non fungsional dilakukan untuk menghasilkan spesifikasi kebutuhan non fungsional. Spesifikasi kebutuhan non fungsional adalah spesifikasi yang rinci tentang hal-hal yang dibutuhkan oleh system sebelum system diimplementasikan.

#### 3.2.1 Analisis Hardware

Space untuk menjalankan aplikasi ini kami merekomendasikan kebutuhan hardware sebagai berikut :

* Prosesor 1.60 Ghz
* Ram 1 Gb
* Keyboard dan Mouse
* VGA 128MB

#### 3.2.2 Anaisis Software

Dan untuk software kami menggunakan software sebagai berikut :

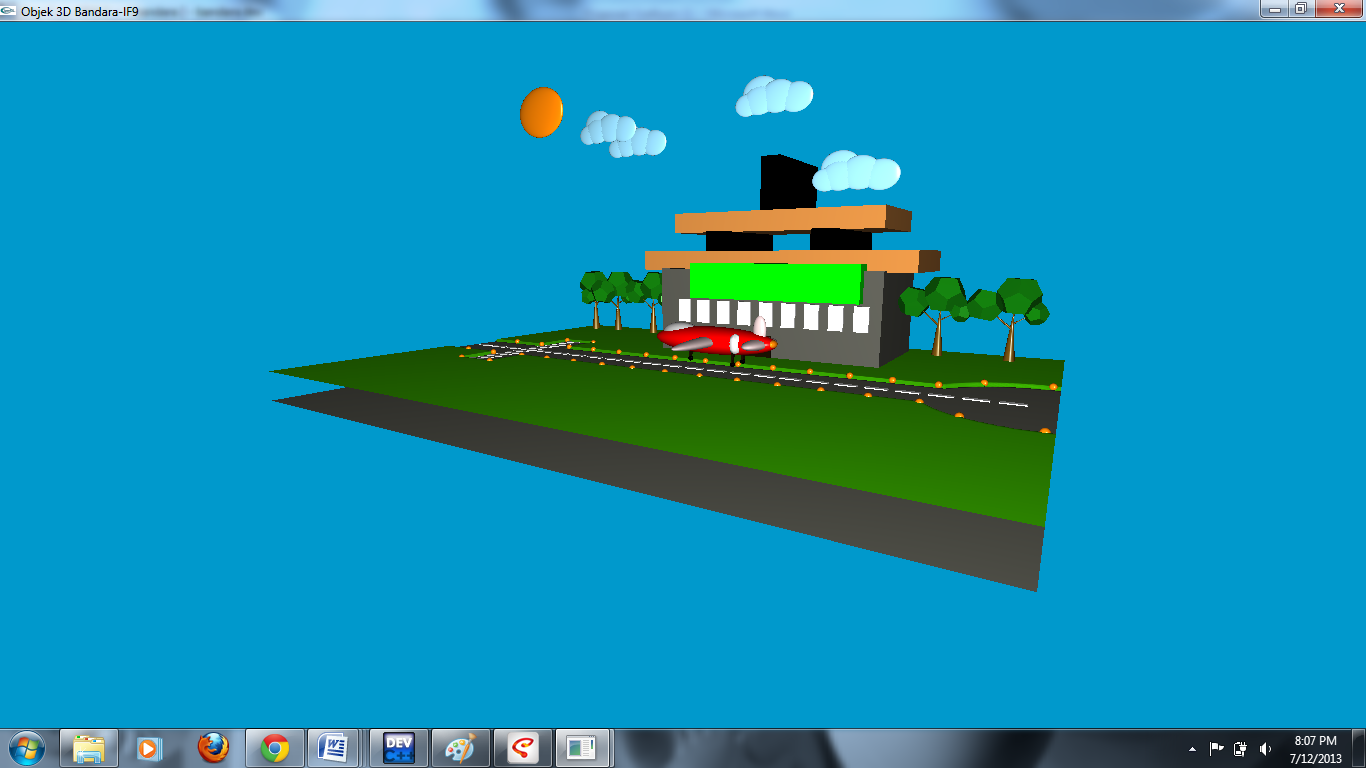
* Sistem Operasi (Microsoft Windows 7)
* Dev C ++ v.6.0
* Glut32.dll

### 3.3 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk menghasilkan spesifikasi kebutuhan fungsional. Spesifikasi kebutuhan fungsional adalah spesifikasi yang rinci tentang hal-hal yang akan dilakukan sistem ketika diimplementasikan.



Gambar Tampak Depan



Gambar Terlihat dari Sudut Kanan



Gambar Terlihat dari Sudut Kiri

|  |  |
| --- | --- |
| **Gambar** | **Source** |
|  | void matahari(void)  {  glPushMatrix();  glTranslatef(-120,120,-100);  glColor3ub(255, 253, 116);  glColor3f(1.0000, 0.5252, 0.0157);  glutSolidSphere(20, 60, 60);  glPopMatrix();  glEndList();  } |
|  | Void awan (void)  {  glPushMatrix();  glTranslatef(50,99,-2);  glutSolidSphere(9, 50, 50);  glPopMatrix();  } |
|  | void pohon(void){  //batang  GLUquadricObj \*pObj;  pObj =gluNewQuadric();  gluQuadricNormals(pObj, GLU\_SMOOTH);    glPushMatrix();  glColor3ub(104,70,14);  glRotatef(270,1,0,0);  gluCylinder(pObj, 4, 0.7, 30, 25, 25);  glPopMatrix();  }  void ranting(void){  GLUquadricObj \*pObj;  pObj =gluNewQuadric();  gluQuadricNormals(pObj, GLU\_SMOOTH);  glPushMatrix();  glColor3ub(104,70,14);  glTranslatef(0,27,0);  glRotatef(330,1,0,0);  gluCylinder(pObj, 0.6, 0.1, 15, 25, 25);  glPopMatrix();  //daun  glPushMatrix();  glColor3ub(18,118,13);  glScaled(5, 5, 5);  glTranslatef(0,7,3);  glutSolidDodecahedron();  glPopMatrix();  } |
|  | void body(void){  glPushMatrix();  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(153, 223, 255);  glColor3f(1.0,0.0,0.0);  glutSolidSphere(10, 50, 50);  glPopMatrix();  }    void depan(void){  glPushMatrix();  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(153, 223, 255);  glColor3f(0.8,0.8,0.8);  glutSolidSphere(8, 25, 10);  glPopMatrix();    }    void sayap(void){  glPushMatrix();  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(153, 223, 255);  glColor3f(0.9,0.8,0.8);  glutSolidSphere(8, 25, 10);  glPopMatrix();  }    void sayapbelakang(void){  glPushMatrix();  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(153, 223, 255);  glColor3f(0.9,0.8,0.8);  glutSolidSphere(8, 25, 10);  glPopMatrix();  }    void sayapatas(void){  glPushMatrix();  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(153, 223, 255);  glColor3f(0.9,0.8,0.8);  glutSolidSphere(10,100, 80);  glPopMatrix();  }    void roda(void){  glPushMatrix();  glColor3f(0.0980, 0.0608, 0.0077);  glScaled(1.5,10,3);  glutSolidCube(1);  glPopMatrix();  }  void roda1(void){  glPushMatrix();  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(153, 223, 255);  glColor3f(0.0,0.0,0.0);  glutSolidSphere(2, 50, 4);  glPopMatrix();  }    void knalpot(void) {  ////////////////////////////////////////////////////////////Knalpot/////////////////////////////////////////  glPushMatrix();  glTranslatef(1,0.6,2);  glutSolidSphere(5, 50, 50);  glPopMatrix();  } |
|  | void markajalan(void) {  glPushMatrix();  glScaled(1, 0.05,0.3);  glTranslatef(2.4,2.5,67);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1,1,1);  glutSolidCube(5.0);  glPopMatrix();  } |
|  | void lampu(void) {  glPushMatrix();  glTranslatef(1,0.6,2);  glutSolidSphere(5, 50, 50);  glPopMatrix();  } |