**TUGAS BESAR**

**GRAFIKA KOMPUTER**

**“MEMBANGUN OBJEK 3D GEDUNG UNIKOM”**

***Diajukan untuk memenuhi***

***Tugas Mata Kuliah Grafika Komputer***

***Program Strata Satu Jurusan Teknik Informatika***

***Fakultas Teknik dan Ilmu Komputer***

***Universitas Komputer Indonesia***

**OLEH :**

**KELOMPOK 10**

**GURUH WIRAAJIPUTRO 10107349**

**INDRA ARIANSYAH 10107177**

****

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS KOMPUTER INDONESIA**

**BANDUNG**

**2013**

DAFTAR ISI

[DAFTAR ISI 1](#_Toc361426650)

[BAB I PENDAHULUAN 1](#_Toc361426651)

[I.1 Latar Belakang Masalah 1](#_Toc361426652)

[I.2 Identifikasi Masalah 2](#_Toc361426653)

[I.3 Maksud dan Tujuan 2](#_Toc361426654)

[I.4 Batasan Masalah 3](#_Toc361426655)

[BAB II LANDASAN TEORI 4](#_Toc361426656)

[II.1 Pengertian Grafika Komputer 4](#_Toc361426657)

[II.2 Elemen – Elemen Dasar Grafika Komputer 4](#_Toc361426658)

[II.3 Grafik Komputer 2D 5](#_Toc361426659)

[II.4 Grafik Komputer 3D 5](#_Toc361426660)

[II.5 Perbedaan Grafik 2D dan Grafik 3D 6](#_Toc361426661)

[II.6 Transformasi 6](#_Toc361426662)

[II.6.1 Jenis jenis dari transformasi dasar: 6](#_Toc361426663)

[II.7 *Color* (Warna) 9](#_Toc361426664)

[II.7.2 Pembagian Warna 9](#_Toc361426665)

[II.7.3 Lighting (pencahayaan) 10](#_Toc361426666)

[II.7.4 Bayangan 10](#_Toc361426667)

[II.7.5 Blending (Pencampuran) 11](#_Toc361426668)

[II.7.6 Texture Mapping 12](#_Toc361426669)

[II.7.7 Fog 14](#_Toc361426670)

[II.7.8 Pengertian OpenGl 14](#_Toc361426671)

[II.7.9 Dev C++ 14](#_Toc361426672)

[BAB III PERANCANGAN DAN IMPLEMENTASI 16](#_Toc361426673)

[III.1 Perancangan dan implementasi 16](#_Toc361426674)

[III.2 Source Code Program OpenGL 16](#_Toc361426675)

[III.3 Tampilan Program OpenGL 60](#_Toc361426676)

[BAB IV KESIMPULAN DAN SARAN 61](#_Toc361426677)

[IV.1 Kesimpulan 61](#_Toc361426678)

[IV.2 Saran 61](#_Toc361426679)

# PENDAHULUAN

## Latar Belakang Masalah

Universitas Komputer Indonesia (UNIKOM) secara resmi berdiri pada hari Selasa, tanggal 8 Agustus 2000 berdasarkan Surat Keputusan Menteri Pendidikan Nasional nomor 126/D/0/2000. Awalnya dimulai pada bulan Juli tahun 1994 ketika didirikan Lembaga Pendidikan Komputer Indonesia Jerman, disingkat LPKIG, bertempat di jalan Dipati Ukur 102 Bandung . Dengan 1 ruang kelas berkapasitas 50 orang dan 1 laboratorium komputer dengan 25 unit komputer, Lembaga ini membuka program pendidikan 1 tahun dengan 5 program studi yaitu Ahli Komputer Aplikasi Bisnis, Ahli Komputer Keuangan & Perbankan, Ahli Komputer Akuntansi & Perpajakan, Ahli Komputer Manajemen & Pemasaran dan Sekretaris Eksekutif. Jumlah peserta pendidikan pada tahun pertama ini sebanyak 233 siswa.

Pada tahun kedua, 1995, dibuka jenjang pendidikan 3 tahun untuk memenuhi animo siswa tahun pertama yang ingin memperdalam ilmunya, disamping pemikiran jangka panjang pengembangan institusi. Pada tahun ini juga dibuka program studi baru, meliputi : Ahli Komputer Teknik Informatika, Ahli Komputer Manajemen Informatika dan Sekretaris Eksekutif. Ruang kelas ditambah menjadi 2 buah dan laboratorium komputer menjadi 2 buah dengan jumlah siswa sebanyak 457 orang. Pada tahun ketiga, 1996, dilakukan penambahan gedung kuliah baru bertempat di jalan Dipati Ukur 116 (gedung FISIP sekarang), sekaligus pemindahan pusat administrasi dan perkantoran. Digedung baru ini dilakukan penambahan 1(satu) Lab. Komputer, 5(lima) Ruang Kuliah, Ruang Dosen dan Ruang Kemahasiswaan. Jumlah siswa dari tahun 1996 hingga tahun 1998 bertambah dari 632 orang menjadi 1184 orang. Pada tahun kelima, 1998, dimulai pembangunan Kampus baru (Gedung Rektorat /Kampus-1 sekarang) berlantai 6(enam) di jalan Dipati Ukur 114. Pembangunan Kampus baru ini dapat diselesaikan pada bulan Agustus 1999, sehingga pada awal perkuliahan bulan September 1999 telah dapat digunakan. Pada tahun 2012, dimulai pembangunan Kampus baru berlantai 13 (tiga belas) di jalan Dipati Ukur. Pembangunan Kampus baru ini masi berlangsung di pertengahan tahun 2013.

Pemodelan bangunan tidak terlepas dari sebuah grafis 3D, agar *user* tidak merasa bosan untuk melihat model tersebut, serta *user* akan tertarik mempelajari dari struktur bangunan tersebut. Banyak cara untuk memodelkan bangunan, salah satunya menggunakan OpenGL. OpenGL adalah spesifikasi standar yang mendefinisikan sebuah lintas-bahasa, lintas platform API untuk mengembangkan aplikasi yang menghasilkan grafis komputer 2D maupun3D.

Untuk itu memerlukan pemodelan dari Gedung Unikom sebagai bahan pembelajaran serta sebagai model untuk merepresentasikan struktur bangunan untuk bahan pengembangannya menggunakan OpenGL.

Untuk pembuatan tugas besar kali ini, penulis menyertakan beberapa objek dalam pembuatannya, diantaranya :

1. Adanya objek utama yaitu sebuah bentuk 3D gedung unikom baru.
2. Objek sekitar gedung, adanya halaman berupa objek latar belakang berupa rumput hijau, pepohonan dan jalan raya untuk mobil.
3. Pada objek gedung unikom akan menggunakan *lightning* dan pergerakan kamera menggunakan keyword untuk mengikuti objek dan kamera dengan posisi diam.
4. Semua objek disini hanya diam saja sesuai dengan alas halaman.

## Identifikasi Masalah

Berdasarkan uraian latar belakang diatas dapat diidentifikasikan masalah yaitu :

1. Bagaimana cara memodelkan Bangunan Unikom baru dengan melihat model paketnya ?
2. Keingintahuan masyarakat terhadap model gedung Unikom baru menjadikan perlunya pemodelan grafis untuk mengilustrasikan terhadap model yang sebenarnya.

## Maksud dan Tujuan

Berdasarkan permasalahan yang diteliti, maka maksud dari penulisan tugas besar ini adalah untuk membangun aplikasi pemodelan gedung Unikom baru dengan berbasis desktop. Sedangkan tujuan yang akan dicapai adalah :

1. Membangun aplikasi untuk memodelkan suatu tempat denganmelihat dari model paketnya.
2. Membangun suatu pemodelan untuk memodelkan bangunan Gedung Unikom baru.
3. Mengaplikasikan materi openGL dari mata kuliah Grafika Komputer.

## Batasan Masalah

Karena masalah yang harus dibahas luas, maka batasan-batasan adalah sebagai berikut:

1. Pembangunan pemodelan tersebut menerapkan bahasa OpenGL.
2. Pemodelan akan menerapkan konsep 3 Dimensi.
3. Pemodelan hanya akan memodelkan bentuk dari bangunan objek saja (Gedung Unikom Baru).

# 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.

## Elemen – Elemen Dasar Grafika Komputer

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.

## 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.

## 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.

## Perbedaan Grafik 2D dan Grafik 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.

## Transformasi

Transformasi dasar pada objek dua dimensi yang pertama adalah translasi (*translation*). Translasi berarti memindahkan suatu objek sepanjang garis lurus dari suatu lokasi koordinat tertentu ke lokasi yang lain. Transformasi skala (*scaling)* digunakan untuk mengubah ukuran suatu objek, sedangkan rotasi (*rotation)* adalah pemindahan objek menurut garis melingkar.

### Jenis jenis dari transformasi dasar:

1. Translasi

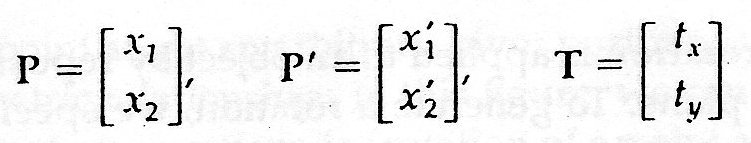
Translasi dilakukan dengan penambahan translasi pada suatu titik koordinat dengan translasi vektor atau shift vektor, yaitu (tx,ty), dimana tx adalah translation vektor menurut sumbu x, sedangkan ty adalah translation vektor menurut sumbu y. koordinat baru titik yang ditranslasi dapat diperoleh dengan

X’= x + tx

Y’= y + ty

Dimana (x,y) adalah koordinat asal suatu objek dan (x’ , y’) adalah koordinat baru objek tersebut setelah ditranslasi.

Kadang-kadang transformasi dinyatakan dalam bentuk matriks, sehingga matriks tranformasi untuk translasi dapat dinyatakan sebagai berikut :



Dengan demikian translasi dua dimensi dapat ditulis dalam bentuk matriks :

|  |
| --- |
| P’ = P + T |

Disamping dinyatakan dalam vektor kolom, matriks transformasi dapat dituliskan dalam bentuk vektor baris, sehingga menjadi P = [ x y ] dan T = [ tx ty ]. Bentuk vektor kolom adalah standar dari symbol matematik, yang juga berlaku bagi notasi grafik seperti GKS dan PHIGS.

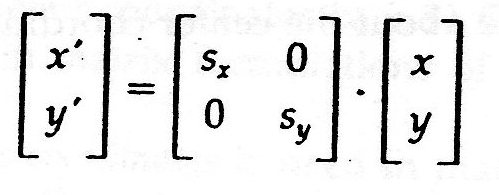
1. Skala

Transformasi skala adalah perubahan ukuran suatu objek. Koordinat baru dapat diperoleh dengan melakukan perkalian nilai koordinat dengan scaling factor, yaitu (sx , sy) ,dimana sx adalah scaling factor menurut sumbu x, sedangkan sy adalah scaling factor menurut sumbu y. koordinat baru titik yang diskala dapat diperoleh dengan

X’= x + sx

Y’ = y + sy

Dimana (x , y) adalah koordinat asal suatu objek dan (x’,y’) adalah koordinat setelah diskala. Matriks transformasi untuk skala dapat dinyatakan sebagai berikut:



Dengan demikian skala dapat juga dituliskan

P’ = S . P

Scaling factor sx dan sy dapat diberikan sembarang nilai positif. Nilai lebih dari 1 menyebabkan objek diperbesar, sebaliknya bila nilai lebih kecil dari 1, maka objek akan diperkecil. Bila sx dan sy mempunyai nilai yang sama, maka skala disebut uniform scaling. Nilai yang tidak sama dari sx dan sy menghasilkan differential scaling, yang biasa digunakan pada program aplikasi.

1. Rotasi

Rotasi dua dimensi pada suatu objek kan memindahkan objek tersebut menurut garis melingkar. Pada bidang xy. Untuk melakukan rotasi diperlukan sudut rotasi θ dan pivot point (xp’ yp ) atau rotasi point dimana objek di rotasi, seperti pada gambar 5-3 nilai positif dari sudut rotasi menentukan arah rotasi berlawanan dengan jarum jam, dan sebaliknya nilai negative akan memutar objek searah jarum jam.

Rotasi dapat dilakukan dengan pivot point yaitu titik pusat koordinat, seperti pada gambar 5-4. Pada betuk ini, r adalah jarak konstan dari titik pusat, sudut φ adalah sudut posisi suatu titik dengan sumbu horizontal, sedangkan θ adalah sudut rotasi. Menggunakan trigonometri, transformasi dapat dinyatakan dengan sudut θ dan φ sebagai berikut:

X’= r cos (φ + θ) = r cos φ cos θ – r sin φ sin θ

y’ = r sin (φ + θ) = r cos φ sin θ + r sin φ cos θ

sedangkan dengan koordinat polar diketahui bahwa

x = r cos φ, y = r sin φ

dengan melakukan substitusi, diperoleh rumus transformasi untuk rotasi suatu titik (x, y) dengan sudut rotasi θ sebagai berikut:

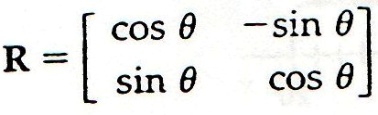
x’ = x cos θ – y sin θ

y’ = x sin θ – y cos θ

matriks transformasi untuk rotasi dapat dinyatakan sebagai berikut:

P’ = R . P

Rotasi dapat dinyatakan dalam bentuk lain, yaitu matriks. Matriks rotasi dapat dituliskan dengan



Rotasi suatu titik terhadap pivot point (xp’ yp ) seperti pada gambar 5-5, menggunakan bentuk trigonometri, secara umum dapat dituliskan sebagai berikut:

X’ = xp +(x - xp) cos θ – (y - yp) sin θ

Y’ = yp + (x – xp) sin θ + (y – yp) cos θ

## *Color* (Warna)

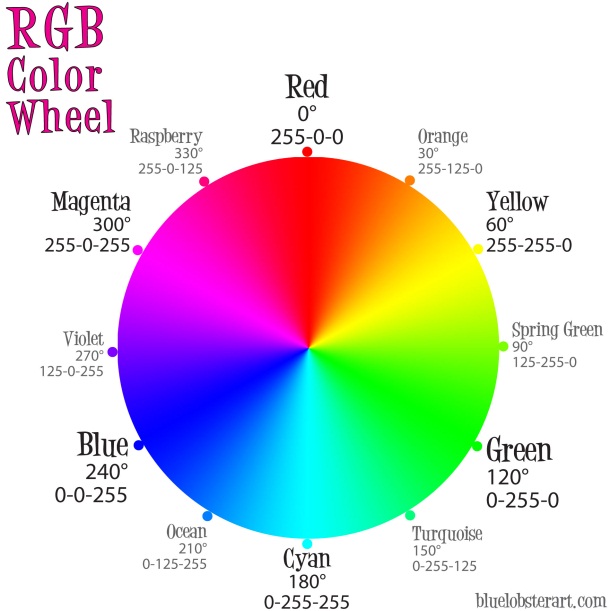
Bentuk gelombang elektromagnetik yang terkandung dalam cahaya yang berasal dari sumber cahaya. Spectrum warna memiliki panjang gelombang elektomagnetik antara 350-750 nanometer .

### Pembagian Warna

RGB (Red-Green-Blue) : warna dasar yang dijadikan patokan warna secara universal (primary colors)

CMYK (Cyan-Magenta-Yellow-Black) : Sistem representasi pada warna tinta yang diterapkan dalam dunia fotografi dan produksi grafika

Sistem Warna Lingkaran (the color wheel) : Digunakan untuk mengkombinasikan dan mengharmonikan warna pada karya seni dan design



Gambar 1 Pembagian warna

Warna dapat didefinisikan secara obyektif/fisik sebagai sifat cahaya yang dipancarkan, atau secara subyektif/psikologis sebagai bagian dari pengalaman indera pengelihatan. Secara obyektif atau fisik, warna dapat diberikan oleh panajang gelombang. Dilihat dari panjang gelombang, cahaya yang tampak oleh mata merupakan salah satu bentuk pancaran energi yang merupakan bagian yang sempit dari gelombang elektromagnetik.

Dari sekian banyak warna, dapat dibagi dalam beberapa bagian yang sering dinamakan dengan sistem warna Prang System yang ditemukan oleh Louis Prang pada 1876 meliputi :

1. Hue, adalah istilah yang digunakan untuk menunjukkan nama dari suatu warna, seperti merah, biru, hijau dsb.
2. Value, adalah dimensi kedua atau mengenai terang gelapnya warna. Contohnya adalah tingkatan warna dari putih hingga hitam.
3. Intensity, seringkali disebut dengan chroma, adalah dimensi yang berhubungan dengan cerah atau suramnya warna.

### Lighting (pencahayaan)

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

Beberapa cara mengatasi masalah pencahayaan, antara lain :

1. Mengerti persepsi dari cahaya (warna)
2. Membuat sebuah solusi untuk merepresentasikan dan menghasilkan warna menggunakan komputer.
3. 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.

### Blending (Pencampuran)

Pencampuran merupakan fungsi yang menggabungkan nilai warna dari sumber dan tujuan. Operasi campuran yaitu cara yang paling alami untuk mengetahui bahwa komponen RGB adalah suatu fragmen yang mewakili warna dan komponen alfa adalah suatu fragmen yang mewakili sifat tidak tembus cahaya.

1. Faktor sumber dan tujuan

Pada proses pencampuran, nilai cairan warna yang masuk fragmen (sumber) digabungkan dengan warna yang sesuai dengan nilai saat ini yang disimpan pada piksel (tujuan) dalam dua tahap proses. Yang pertama menghitung faktor sumber dan tujuan, factor-faktor tersebut adalah RGBA quadruplets yang masing-masing dikalian dengan komponen-komponen R, G, B dan nilai-nilai dari sumber dan tujuan. Kemudian komponen yang sesuai dalam dua set RGBA quadruplets. Secara sistematis, faktor sumber dan tujuan pencampuran (SR, Sg, Sb, Sa) dan (Dr, Dg, dB, Da) dan nilai RGBA ditandai dengan s atau d dan terakhir nilai RGBA dicampurkan yang diperoleh dengan (RsSr + RdDr, GsSg + GdDg, BsSb + BdDb, AsSa + Adda) dimana setiap komponen adalah quadruplets is eventually clamped to [0,1].

Dengan menggunakan glBlendFunc () untuk persediaan pada dua hal utama, yang pertama menentukan bagaimana faktor sumber dan tujuan harus dihitung dan yang kedua menunjukan bagaimana faktor sumber dan tujuan dihitung. Dan untuk proses pencampurannya harus ada faktor pengaktifannya menggunakan : glEnable (GL\_BLEND). Menggunakan glDisable () dengan GL\_BLEND untuk menonaktifkan Pencampuran dan menggunakan konstan GL\_ONE (sumber) dan GL\_ZERO (tujuan) memberikan hasil yang sama seperti ketika Pencampuran dinonaktifkan. Nilai-nilai ini bersifat default dengan void glBlendFunc (GLenum *sfactor,* GLenum *dfactor)*.

Mengontrol bagaimana nilai warna dalam fragmen yang diproses digabungkan dengan yang sudah disimpan dalam framebuffer (tujuan).Pendapat *sfactor* menunjukkan bagaimana untuk menghitung faktor sumber Pencampuran dan *dfactor* menunjukkan bagaimana untuk menghitung faktor tujuan Pencampuran. Campuran faktor yang diasumsikan terletak pada rentang [0,1]; setelah nilai warna dalam sumber dan tujuan digabungkan, setelah dihitung kisaran [0,1].

### 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
7. Wrapping , filtering, dsb.

Langkah-langkah dalam memulai mapping sebuah tekstur yakni dengan spesifikasi dibawah ini :

1. Menentukan Tekstur Image
2. Mendefinisikan tekstur image dari sebuah array teksel (element tekstur ) ke dalam memory cpu : Glubyte my\_texels[512][512];
3. Mendefinisikan seperti semua peta piksel yang lain
4. Gambar yang didefinisikan (baik secara manual maupun dengn suatu fungsi matematik tertentu).
5. Membangkitkan dengan kode aplikasi
6. Mengenablekan tekstur mapping

glEnable(GL\_TEXTURE\_2D)

OpenGL mendukung 1 sampai 4 dimensional tekstur mapping

Mendefinisikan gambar sebagai sebuah tekstur

glTexImage2D (target, level, components, w, h, border, format, type, texels );

Keterangan :

target: tipe dari teksture, e.g. GL\_TEXTURE\_2D

level: digunakan untuk *mipmapping*

components: element per texel

w, h: lebar dan tinggi dari texels pada pixels

border: digunakan untuk smoothing

format and type: menjelaskan texels

texels: pointer ke array texel

glTexImage2D(GL\_TEXTURE\_2D, 0, 3, 512, 512, 0,GL\_RGB, GL\_UNSIGNED\_BYTE, my\_texels);

Mengubah gambar tekstur

OpenGL meminta dimensi tekstur untuk menjadi dasar dari 2

Jika dimensi dari image bukan power ke 2, gluScaleImage (format,w\_in,h\_in,type\_in,\*data\_in,w\_out,h\_out,type\_out,\*data\_out);

data\_in adalah gambar inputan.

data\_out adalah gambar hasil

Mapping Tekstur :

Didasarkan pada koordinat tekstur parametric

glTexCoord\*() ditetapkan pada masing – masing vertex

### Fog

Fog adalah suatu istilah umum yang menggambarkan bentuk yang sama dari pengaruh atmosfer, yang digunakan untuk menirukan kabut, kabut tipis, asap, atau polusi. fog sangat penting di dalam aplikasi-aplikasi simulasi yang visual, yang dibatasi oleh jarak penglihatan untuk itu perlu lebih mendekati objek. Juga sering disatukan ke dalam tampilan simulator penerbangan.

Ketika fog memungkinkan, sudut pandang objek bersifat lebih jauh untuk memudarkan warna fog. Anda dapat mengendalikan kepadatan fog, yang mana menentukan tingkat objek memudar dengan jarak meningkat, seperti juga warna fog. Fog terdapat di dalam kedua gaya yaitu gaya RGBA dan indeks warna, meski perhitungannya sedikit berbeda di kedua gaya

### 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.

### Dev C++

Dev-C + + adalah sebuah lingkungan pengembangan terpadu gratis (IDE) didistribusikan di bawah GNU General Public License untuk pemrograman di C dan C + +. MinGW, sebuah kompiler gratis, dibundel dengan itu. IDE ditulis dalam Delphi.

Proyek ini diselenggarakan oleh SourceForge. Dev-C + + pada awalnya dikembangkan oleh programmer Colin Laplace. Dev-C + + berjalan secara eksklusif pada Microsoft Windows.

Pertumpahan darah Dev-C + + adalah fitur lengkap Integrated Development Environment (IDE) untuk C dan C + + bahasa pemrograman. Ia menggunakan MinGW atau TDM-GCC pelabuhan 64bit dari GCC sebagai compiler nya. Dev-C + + juga dapat digunakan dalam kombinasi dengan Cygwin atau kompiler berbasis GCC lainnya.

Salah satu aspek tambahan Dev-C + + adalah penggunaan DevPaks, dikemas ekstensi pada lingkungan pemrograman dengan library tambahan, template, dan utilitas. DevPaks sering mengandung, namun tidak terbatas pada, GUI utilitas, termasuk toolkit populer seperti GTK +, wxWidgets, dan FLTK. DevPaks lainnya termasuk perpustakaan untuk lebih penggunaan fungsi lanjutan.

Dev-C + + umumnya dianggap program Windows-only, tetapi ada upaya untuk membuat versi Linux: file header dan pembatas jalan yang switchable antara platform.

# LANGKAH PEMBUATAN DAN IMPLEMENTASI

## Langkah Pembuatan

Langkah-langkah pembuatan dimulai dari membuat sebuah persegi panjang yang disusun hingga menyerupai bentuk bangunan gedung. Setelah seluruh badan gedung selesai maka tahap selanjutnya gedung tersebut diberi texture berupa warna dari bangunan yang sebelumnya sudah dibuat dahulu.

## Source Code Program OpenGL

|  |
| --- |
| #include <windows.h> //for Code::Blocks user  #include <GL/glut.h>  #include <stdio.h>  #include <gl\gl.h>  #include <gl\glu.h>  #include <stdlib.h>  GLuint texture[20];  struct Image {  unsigned long sizeX;  unsigned long sizeY;  char \*data;  };  typedef struct Image Image;  #define checkImageWidth 100  #define checkImageHeight 100  GLubyte checkImage[checkImageWidth][checkImageHeight][3];  char title[] = "Gedung Unikom Baru"; // penamaan judul window  int windowWidth = 1024; // mode lebar layar normal  int windowHeight = 768; // mode tinggi layar normal  int windowPosX = 50; // posisi tampilan layar pojok k1ri atas titik x  int windowPosY = 50; // posisi tampilan layar pojok k1ri atas titik y  bool fullScreenMode = false; // Full-screen or windowed mode?  float rotx , roty , tx=0 , ty, tz=0,y ,zz,p;  int i;  /\*----------------------------------------------------------------------------\*/  // Prosedure interaksi pada keyboard  void myKeyboard(unsigned char key, int x, int y)  {    switch (key) {  case 27: // ESC: exit atau keluar dari program  exit(0);  break;  case 'a': // geser kekiri  tx+=0.1;  break;  case 'd' : // geser kekanan  tx-=0.1;  break;  case 'w': // maju  tz+=0.1;  break;  case 's': // mundur  tz-=0.1;  break;  case 'y': // memutarkan kekiri  roty+=1;  break;  case 't': // memutar kekanan  roty-=1;  break;  case 'q': // memutar kekanan  p-=0.1;  break;  case 'e': // memutar kekanan  p+=0.1;  break;  default:  break;  }  }  // cek gambar  void makeCheckImage(void){  int i, j, c;  for (i = 0; i < checkImageWidth; i++) {  for (j = 0; j < checkImageHeight; j++) {  c = ((((i&0x8)==0)^((j&0x8)==0)))\*255;  checkImage[i][j][0] = (GLubyte) c;  checkImage[i][j][1] = (GLubyte) c;  checkImage[i][j][2] = (GLubyte) c;  }  }  }  // load gambar  int ImageLoad(char \*filename, Image \*image) {  FILE \*file;  unsigned long size; // size of the image in bytes.  unsigned long i; // standard counter.  unsigned short int planes; // number of planes in image (must be 1)  unsigned short int bpp; // number of bits per pixel (must be 24)  char temp; // temporary color storage for bgr-rgb conversion.  // make sure the file is there.  if ((file = fopen(filename, "rb"))==NULL){  printf("File Not Found : %s\n",filename);  return 0;  }  // seek through the bmp header, up to the width/height:  fseek(file, 18, SEEK\_CUR);  // read the width  if ((i = fread(&image->sizeX, 4, 1, file)) != 1) {  printf("Error reading width from %s.\n", filename);  return 0;  }  //printf("Width of %s: %lu\n", filename, image->sizeX);  // read the height  if ((i = fread(&image->sizeY, 4, 1, file)) != 1) {  printf("Error reading height from %s.\n", filename);  return 0;  }  //printf("Height of %s: %lu\n", filename, image->sizeY);  // calculate the size (assuming 24 bits or 3 bytes per pixel).  size = image->sizeX \* image->sizeY \* 3;  // read the planes  if ((fread(&planes, 2, 1, file)) != 1) {  printf("Error reading planes from %s.\n", filename);  return 0;  }  if (planes != 1) {  printf("Planes from %s is not 1: %u\n", filename, planes);  return 0;  }  // read the bitsperpixel  if ((i = fread(&bpp, 2, 1, file)) != 1) {  printf("Error reading bpp from %s.\n", filename);  return 0;  }  if (bpp != 24) {  printf("Bpp from %s is not 24: %u\n", filename, bpp);  return 0;  }  // seek past the rest of the bitmap header.  fseek(file, 24, SEEK\_CUR);  // read the data.  image->data = (char \*) malloc(size);  if (image->data == NULL) {  printf("Error allocating memory for color-corrected image data");  return 0;  }  if ((i = fread(image->data, size, 1, file)) != 1) {  printf("Error reading image data from %s.\n", filename);  return 0;  }  for (i=0;i<size;i+=3) { // reverse all of the colors. (bgr -> rgb)  temp = image->data[i];  image->data[i] = image->data[i+2];  image->data[i+2] = temp;  }  // we're done.  return 1;  }  //-----------------------lokasi ruang untuk teksture----------------------------  //rumput  Image \* loadTexture1(){  Image \*image1;  // allocate space for texture  image1 = (Image \*) malloc(sizeof(Image));  if (image1 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/rumput.bmp", image1)) {  exit(1);  }  return image1;  }  //tembok  Image \* loadTexture2(){  Image \*image2;  // allocate space for texture  image2 = (Image \*) malloc(sizeof(Image));  if (image2 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/dalam.bmp", image2)) {  exit(1);  }  return image2;  }  //tembok1  Image \* loadTexture3(){  Image \*image3;  // allocate space for texture  image3 = (Image \*) malloc(sizeof(Image));  if (image3 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/samping.bmp", image3)) {  exit(1);  }  return image3;  }  //tembok2  Image \* loadTexture4(){  Image \*image4;  // allocate space for texture  image4 = (Image \*) malloc(sizeof(Image));  if (image4 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/sebelah.bmp", image4)) {  exit(1);  }  return image4;  }  //tembok3  Image \* loadTexture5(){  Image \*image5;  // allocate space for texture  image5 = (Image \*) malloc(sizeof(Image));  if (image5 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/depan.bmp", image5)) {  exit(1);  }  return image5;  }  //tembok4  Image \* loadTexture6(){  Image \*image6;  // allocate space for texture  image6 = (Image \*) malloc(sizeof(Image));  if (image6 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/sebelahtinggi.bmp", image6)) {  exit(1);  }  return image6;  }  //tembok5  Image \* loadTexture7(){  Image \*image7;  // allocate space for texture  image7 = (Image \*) malloc(sizeof(Image));  if (image7 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/belakang.bmp", image7)) {  exit(1);  }  return image7;  }  //tembok6  Image \* loadTexture8(){  Image \*image8;  // allocate space for texture  image8 = (Image \*) malloc(sizeof(Image));  if (image8 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/depanatas.bmp", image8)) {  exit(1);  }  return image8;  }  //daun pohon  Image \* loadTexture9(){  Image \*image9;  // allocate space for texture  image9 = (Image \*) malloc(sizeof(Image));  if (image9 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/daun2.bmp", image9)) {  exit(1);  }  return image9;  }  //batang pohon  Image \* loadTexture10(){  Image \*image10;  // allocate space for texture  image10 = (Image \*) malloc(sizeof(Image));  if (image10 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/batang.bmp", image10)) {  exit(1);  }  return image10;  }  //background  Image \* loadTexture11(){  Image \*image11;  // allocate space for texture  image11 = (Image \*) malloc(sizeof(Image));  if (image11 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/pemandangan.bmp", image11)) {  exit(1);  }  return image11;  }  //parkiran  Image \* loadTexture12(){  Image \*image12;  // allocate space for texture  image12 = (Image \*) malloc(sizeof(Image));  if (image12 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/parkiran.bmp", image12)) {  exit(1);  }  return image12;  }  //Trotoar  Image \* loadTexture13(){  Image \*image13;  // allocate space for texture  image13 = (Image \*) malloc(sizeof(Image));  if (image13 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/trotoar.bmp", image13)) {  exit(1);  }  return image13;  }  //jalan  Image \* loadTexture14(){  Image \*image14;  // allocate space for texture  image14 = (Image \*) malloc(sizeof(Image));  if (image14 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/jalan.bmp", image14)) {  exit(1);  }  return image14;  }  //dalam  Image \* loadTexture15(){  Image \*image15;  // allocate space for texture  image15 = (Image \*) malloc(sizeof(Image));  if (image15 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/dalamtinggi.bmp", image15)) {  exit(1);  }  return image15;  }  //belakangsamping  Image \* loadTexture16(){  Image \*image16;  // allocate space for texture  image16 = (Image \*) malloc(sizeof(Image));  if (image16 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/dalamtinggi.bmp", image16)) {  exit(1);  }  return image16;  }  //atas samping  Image \* loadTexture17(){  Image \*image17;  // allocate space for texture  image17 = (Image \*) malloc(sizeof(Image));  if (image17 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/atassamping.bmp", image17)) {  exit(1);  }  return image17;  }  //atassebelah  Image \* loadTexture18(){  Image \*image18;  // allocate space for texture  image18 = (Image \*) malloc(sizeof(Image));  if (image18 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/atassebelah.bmp", image18)) {  exit(1);  }  return image18;  }  //atasdalam  Image \* loadTexture19(){  Image \*image19;  // allocate space for texture  image19 = (Image \*) malloc(sizeof(Image));  if (image19 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/dalamatas.bmp", image19)) {  exit(1);  }  return image19;  }  //sampingtinggi  Image \* loadTexture20(){  Image \*image20;  // allocate space for texture  image20 = (Image \*) malloc(sizeof(Image));  if (image20 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/belakangsamping.bmp", image20)) {  exit(1);  }  return image20;  }  //towerunikom  Image \* loadTexture21(){  Image \*image21;  // allocate space for texture  image21 = (Image \*) malloc(sizeof(Image));  if (image21 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/unikom.bmp", image21)) {  exit(1);  }  return image21;  }  //merah  Image \* loadTexture22(){  Image \*image22;  // allocate space for texture  image22 = (Image \*) malloc(sizeof(Image));  if (image22 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/merah.bmp", image22)) {  exit(1);  }  return image22;  }  //kaca  Image \* loadTexture23(){  Image \*image23;  // allocate space for texture  image23 = (Image \*) malloc(sizeof(Image));  if (image23 == NULL) {  printf("Error allocating space for image");  exit(0);  }  if (!ImageLoad("gambar/kaca.bmp", image23)) {  exit(1);  }  return image23;  }  /\*----------------------------------------------------------------------------\*/  void init(void)  { int n;  glClearColor(0.0, 0.0, 0.0, 0.0);  glEnable(GL\_DEPTH\_TEST);  glDepthFunc(GL\_LESS);  Image \*image1 = loadTexture1();  Image \*image2 = loadTexture2();  Image \*image3 = loadTexture3();  Image \*image4 = loadTexture4();  Image \*image5 = loadTexture5();  Image \*image6 = loadTexture6();  Image \*image7 = loadTexture7();  Image \*image8 = loadTexture8();  Image \*image9 = loadTexture9();  Image \*image10 = loadTexture10();  Image \*image11 = loadTexture11();  Image \*image12 = loadTexture12();  Image \*image13 = loadTexture13();  Image \*image14 = loadTexture14();  Image \*image15 = loadTexture15();  Image \*image16 = loadTexture16();  Image \*image17 = loadTexture17();  Image \*image18 = loadTexture18();  Image \*image19 = loadTexture19();  Image \*image20 = loadTexture20();  Image \*image21 = loadTexture21();  Image \*image22 = loadTexture22();  Image \*image23 = loadTexture23();    if(image1 == NULL||image2 == NULL||image3 == NULL||image4 == NULL||  image5 == NULL||image6 == NULL||image7 == NULL||image8 == NULL||  image9 == NULL||image10 == NULL||image11 == NULL||image12 == NULL||  image13 == NULL||image14 == NULL||image15 == NULL||image16 == NULL||  image17 == NULL||image18 == NULL||image19 == NULL||image20 == NULL||  image21 == NULL||image22 == NULL||image23 == NULL) {  printf("Image was not returned from loadTexture\n");  exit(0);  }  makeCheckImage();  glPixelStorei(GL\_UNPACK\_ALIGNMENT, 1);      //create texture kaca  glGenTextures(23, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[22]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image23->sizeX, image23->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image23->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  //create texture merah  glGenTextures(22, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[22]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image22->sizeX, image22->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image22->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  //create texture unikom  glGenTextures(21, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[21]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image21->sizeX, image21->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image21->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  //Create texture belakangsamping  glGenTextures(20, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[20]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image20->sizeX, image20->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image20->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture dalamatas  glGenTextures(19, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[19]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image19->sizeX, image19->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image19->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture Atassebelah  glGenTextures(18, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[18]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image18->sizeX, image18->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image18->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture Atassamping  glGenTextures(17, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[17]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image17->sizeX, image17->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image17->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture Dalamtinggi  glGenTextures(16, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[16]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image16->sizeX, image16->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image16->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture dalamtinggi  glGenTextures(15, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[15]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image15->sizeX, image15->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image15->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture jalan  glGenTextures(14, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[14]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image14->sizeX, image14->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image14->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture trotoar  glGenTextures(13, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[13]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image13->sizeX, image14->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image13->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture parkiran  glGenTextures(12, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[12]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image12->sizeX, image12->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image12->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture background  glGenTextures(11, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[11]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image11->sizeX, image11->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image11->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture batang pohon  glGenTextures(10, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[10]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image10->sizeX, image10->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image10->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture daun pohon  glGenTextures(9, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[9]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image9->sizeX, image9->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image9->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture tembok  glGenTextures(8, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[8]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image8->sizeX, image8->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image8->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture tembok  glGenTextures(7, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[7]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image7->sizeX, image7->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image7->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture tembok  glGenTextures(6, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[6]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image6->sizeX, image6->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image6->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture tembok  glGenTextures(6, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[5]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image5->sizeX, image5->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image5->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture //tembok 2  glGenTextures(5, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[4]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image4->sizeX, image4->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image4->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Textur //Tembok 1  glGenTextures(4, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[3]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image3->sizeX, image3->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image3->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture tembok  glGenTextures(3, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[2]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image2->sizeX, image2->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image2->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  // Create Texture rumput  glGenTextures(2, texture);  glBindTexture(GL\_TEXTURE\_2D, texture[1]);  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MAG\_FILTER,GL\_LINEAR); //scale linearly when image bigger than texture  glTexParameteri(GL\_TEXTURE\_2D,GL\_TEXTURE\_MIN\_FILTER,GL\_LINEAR); //scale linearly when image smalled than texture  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, image1->sizeX, image1->sizeY, 0,  GL\_RGB, GL\_UNSIGNED\_BYTE, image1->data);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);    glBindTexture(GL\_TEXTURE\_2D, texture[0]);  glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_CLAMP);  glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_T, GL\_CLAMP);  glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);  glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_DECAL);  glTexImage2D(GL\_TEXTURE\_2D, 0, 3, checkImageWidth,  checkImageHeight, 0, GL\_RGB, GL\_UNSIGNED\_BYTE,&checkImage[0][0][0]);  glEnable(GL\_TEXTURE\_2D);  glShadeModel(GL\_FLAT);  }  /\*----------------------------------------------------------------------------\*/  void rumput(){  glBindTexture(GL\_TEXTURE\_2D, texture[1]);  glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-3, -0.5, -3);  glTexCoord2f(0.5, 0.0);glVertex3f(-3, -0.5, 1.02);  glTexCoord2f(0.5, 0.5);glVertex3f(3, -0.5, 1.02);  glTexCoord2f(0.0, 0.5);glVertex3f(3, -0.5, -3);  glEnd();  }  //----------------------Procedure dinding gedung--------------------------------  void dinding\_gedung()  {    //tembok bagian depan  //tembok depan kiri1  glBindTexture(GL\_TEXTURE\_2D, texture[3]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.8,0.4,0.5); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-1.0,0.4,0.5); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-1.0,-0.5,0.5); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.8,-0.5,0.5); //titik C  glEnd();    //tembok depan kiri1 dalam1  glBindTexture(GL\_TEXTURE\_2D, texture[2]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.8,0.4,0.48); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.8,0.4,0.5); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.8,-0.5,0.5); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-0.8,-0.5,0.48); //titik F  glEnd();    //tembok depan kiri1 dalam depan  glBindTexture(GL\_TEXTURE\_2D, texture[2]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.76,0.38,0.48); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.8,0.38,0.48); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.8,-0.5,0.48); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.76,-0.5,0.48); //titik C  glEnd();    //tembok depan kiri1 dalam2  glBindTexture(GL\_TEXTURE\_2D, texture[2]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.76,0.4,0.5); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.76,0.4,0.48); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.76,-0.5,0.48); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-0.76,-0.5,0.5); //titik F  glEnd();    //tembok depan kiri2  glBindTexture(GL\_TEXTURE\_2D, texture[4]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.4,0.4,0.5); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.76,0.4,0.5); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.76,-0.5,0.5); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.4,-0.5,0.5); //titik C  glEnd();    //tembok depan kiri2 dalam  glBindTexture(GL\_TEXTURE\_2D, texture[2]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.35,0.38,0.5); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.4,0.38,0.5); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.4,-0.5,0.5); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.35,-0.5,0.5); //titik C  glEnd();    //tembok depan kiri2 luar  glBindTexture(GL\_TEXTURE\_2D, texture[5]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.35,0.4,0.5); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.35,0.4,0.54); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.35,-0.5,0.54); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-0.35,-0.5,0.5); //titik F  glEnd();    //tembok depan tengah1  glBindTexture(GL\_TEXTURE\_2D, texture[5]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.35,0.4,0.54); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.35,0.4,0.54); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.35,-0.5,0.54); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.35,-0.5,0.54); //titik C  glEnd();    //tembok depan kanan2 luar  glBindTexture(GL\_TEXTURE\_2D, texture[5]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.35,0.4,0.5); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(0.35,0.4,0.54); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(0.35,-0.5,0.54); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.35,-0.5,0.5); //titik F  glEnd();    //tembok depan kanan2 dalam  glBindTexture(GL\_TEXTURE\_2D, texture[2]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.4,0.38,0.5); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.35,0.38,0.5); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.35,-0.5,0.5); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.4,-0.5,0.5); //titik C  glEnd();    //tembok depan kanan2  glBindTexture(GL\_TEXTURE\_2D, texture[4]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.4,0.4,0.5); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.76,0.4,0.5); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.76,-0.5,0.5); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.4,-0.5,0.5); //titik C  glEnd();    //tembok depan kanan1 dalam2  glBindTexture(GL\_TEXTURE\_2D, texture[2]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.76,0.4,0.5); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(0.76,0.4,0.48); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(0.76,-0.5,0.48); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.76,-0.5,0.5); //titik F  glEnd();    //tembok depan kanan1 dalam depan  glBindTexture(GL\_TEXTURE\_2D, texture[2]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.76,0.38,0.48); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.8,0.38,0.48); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.8,-0.5,0.48); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.76,-0.5,0.48); //titik C  glEnd();    //tembok depan kanan1 dalam1  glBindTexture(GL\_TEXTURE\_2D, texture[2]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.8,0.4,0.48); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(0.8,0.4,0.5); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(0.8,-0.5,0.5); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.8,-0.5,0.48); //titik F  glEnd();    //tembok depan kanan1  glBindTexture(GL\_TEXTURE\_2D, texture[3]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.8,0.4,0.5); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(1.0,0.4,0.5); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(1.0,-0.5,0.5); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.8,-0.5,0.5); //titik C  glEnd();    //------------------------------------------------------------------------------  //tembok bagian samping  //samping kanan  //samping kanan1  glBindTexture(GL\_TEXTURE\_2D, texture[3]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(1.0,0.4,0.3); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(1.0,0.4,0.5); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(1.0,-0.5,0.5); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(1.0,-0.5,0.3); //titik F  glEnd();    //samping kanan1 dalam  glBindTexture(GL\_TEXTURE\_2D, texture[2]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(1.0,0.4,0.3); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(0.95,0.4,0.3); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(0.95,-0.5,0.3); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(1.0,-0.5,0.3); //titik F  glEnd();    //samping kanan1 depan  glBindTexture(GL\_TEXTURE\_2D, texture[4]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.95,0.4,-0.1); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(0.95,0.4,0.3); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(0.95,-0.5,0.3); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.95,-0.5,-0.1); //titik F  glEnd();    //samping kanan2 tinggi  glBindTexture(GL\_TEXTURE\_2D, texture[6]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.95,1.0,-0.7); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(0.95,1.0,-0.1); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(0.95,-0.5,-0.1); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.95,-0.5,-0.7); //titik F  glEnd();    //samping kiri  glBindTexture(GL\_TEXTURE\_2D, texture[3]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-1.0,0.4,0.3); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-1.0,0.4,0.5); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-1.0,-0.5,0.5); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-1.0,-0.5,0.3); //titik F  glEnd();    //samping kiri1 dalam  glBindTexture(GL\_TEXTURE\_2D, texture[2]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-1.0,0.4,0.3); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.95,0.4,0.3); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.95,-0.5,0.3); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-1.0,-0.5,0.3); //titik F  glEnd();    //samping kiri1 depan  glBindTexture(GL\_TEXTURE\_2D, texture[4]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.95,0.4,-0.1); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.95,0.4,0.3); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.95,-0.5,0.3); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-0.95,-0.5,-0.1); //titik F  glEnd();    //samping kiri2 tinggi  glBindTexture(GL\_TEXTURE\_2D, texture[6]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.95,1.0,-0.7); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.95,1.0,-0.1); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.95,-0.5,-0.1); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-0.95,-0.5,-0.7); //titik F  glEnd();  //-----------------------------------------------------------------------------  //bagian belakang  //belakang kiri1 depan  glBindTexture(GL\_TEXTURE\_2D, texture[20]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.95,1.0,-0.7); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.7,1.0,-0.7); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.7,-0.5,-0.7); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-0.95,-0.5,-0.7); //titik F  glEnd();    //belakang kiri1 dalam  glBindTexture(GL\_TEXTURE\_2D, texture[16]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.7,1.0,-0.7); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.7,1.0,-0.9); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.7,-0.5,-0.9); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-0.7,-0.5,-0.7); //titik F  glEnd();    //belakang kiri2 depan  glBindTexture(GL\_TEXTURE\_2D, texture[20]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.7,1.0,-0.9); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.45,1.0,-0.9); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.45,-0.5,-0.9); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-0.7,-0.5,-0.9); //titik F  glEnd();    //belakang kiri2 dalam  glBindTexture(GL\_TEXTURE\_2D, texture[15]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.45,1.0,-0.9); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.45,1.0,-1.0); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.45,-0.5,-1.0); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-0.45,-0.5,-0.9); //titik F  glEnd();    //belakang kanan1 depan  glBindTexture(GL\_TEXTURE\_2D, texture[20]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.95,1.0,-0.7); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(0.7,1.0,-0.7); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(0.7,-0.5,-0.7); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.95,-0.5,-0.7); //titik F  glEnd();    //belakang kanan1 dalam  glBindTexture(GL\_TEXTURE\_2D, texture[16]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.7,1.0,-0.7); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(0.7,1.0,-0.9); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(0.7,-0.5,-0.9); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.7,-0.5,-0.7); //titik F  glEnd();    //belakang kanan2 depan  glBindTexture(GL\_TEXTURE\_2D, texture[20]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.7,1.0,-0.9); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(0.45,1.0,-0.9); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(0.45,-0.5,-0.9); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.7,-0.5,-0.9); //titik F  glEnd();    //belakang kanan2 dalam  glBindTexture(GL\_TEXTURE\_2D, texture[15]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.45,1.0,-0.9); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(0.45,1.0,-1.0); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(0.45,-0.5,-1.0); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.45,-0.5,-0.9); //titik F  glEnd();    //belakang tengah  glBindTexture(GL\_TEXTURE\_2D, texture[7]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.45,1.0,-1.0); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.45,1.0,-1.0); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.45,-0.5,-1.0); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.45,-0.5,-1.0); //titik F  glEnd();  //-----------------------------------------------------------------------------  //tembok bagian atas depan  //tembok atas depan kiri1  glBindTexture(GL\_TEXTURE\_2D, texture[18]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.8,1.0,-0.1); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.95,1.0,-0.1); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.95,0.38,-0.1); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.8,0.38,-0.1); //titik C  glEnd();    //tembok atas depan kiri1 dalam1  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.8,1.0,-0.18); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.8,1.0,-0.1); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.8,0.38,-0.1); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-0.8,0.38,-0.18); //titik F  glEnd();    //tembok atas depan kiri1 dalam depan  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.76,0.98,-0.18); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.8,0.98,-0.18); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.8,0.38,-0.18); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.76,0.38,-0.18); //titik C  glEnd();    //tembok atas depan kiri1 dalam2  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.76,1.0,-0.1); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.76,1.0,-0.18); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.76,0.38,-0.18); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-0.76,0.38,-0.1); //titik F  glEnd();    //tembok atas depan kiri2  glBindTexture(GL\_TEXTURE\_2D, texture[17]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.4,1.0,-0.1); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.76,1.0,-0.1); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.76,0.38,-0.1); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.4,0.38,-0.1); //titik C  glEnd();    //tembok atas depan kiri2 dalam  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.35,0.98,-0.1); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.4,0.98,-0.1); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.4,0.38,-0.1); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.35,0.38,-0.1); //titik C  glEnd();    //tembok atas depan kiri2 luar  glBindTexture(GL\_TEXTURE\_2D, texture[8]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.35,1.0,-0.1); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(-0.35,1.0,-0.04); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(-0.35,0.38,-0.04); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(-0.35,0.38,-0.1); //titik F  glEnd();    //tembok atas depan tengah1  glBindTexture(GL\_TEXTURE\_2D, texture[8]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.35,1.0,-0.04); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.35,1.0,-0.04); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.35,0.38,-0.04); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.35,0.38,-0.04); //titik C  glEnd();    //tembok atas depan kanan2 luar  glBindTexture(GL\_TEXTURE\_2D, texture[8]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.35,1.0,-0.1); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(0.35,1.0,-0.04); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(0.35,0.38,-0.04); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.35,0.38,-0.1); //titik F  glEnd();    //tembok atas depan kanan2 dalam  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.4,0.98,-0.1); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.35,0.98,-0.1); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.35,0.38,-0.1); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.4,0.38,-0.1); //titik C  glEnd();    //tembok atas depan kanan2  glBindTexture(GL\_TEXTURE\_2D, texture[17]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.4,1.0,-0.1); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.76,1.0,-0.1); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.76,0.38,-0.1); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.4,0.38,-0.1); //titik C  glEnd();    //tembok atas depan kanan1 dalam2  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.76,1.0,-0.1); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(0.76,1.0,-0.18); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(0.76,0.38,-0.18); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.76,0.38,-0.1); //titik F  glEnd();    //tembok atas depan kanan1 dalam depan  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.76,0.98,-0.18); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.8,0.98,-0.18); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.8,0.38,-0.18); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.76,0.38,-0.18); //titik C  glEnd();    //tembok atas depan kanan1 dalam1  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.8,1.0,-0.18); //titik E  glTexCoord2f(0.0, 1.0);glVertex3f(0.8,1.0,-0.1); //titik D  glTexCoord2f(0.0, 0.0);glVertex3f(0.8,0.38,-0.1); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.8,0.38,-0.18); //titik F  glEnd();    //tembok atas depan kanan1  glBindTexture(GL\_TEXTURE\_2D, texture[18]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.8,1.0,-0.1); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.95,1.0,-0.1); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.95,0.38,-0.1); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.8,0.38,-0.1); //titik C  glEnd();    //--------------------------Procedure atap--------------------------------------  //atapbawah    //atapdepanbawahtengah  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.35,0.4,-0.04); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.35,0.4,0.54); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.35,0.4,0.54); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.35,0.4,-0.04); //titik C  glEnd();    //atapdepanbawahtengahkiri  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.35,0.38,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.35,0.38,0.5); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.35,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.35,0.4,-0.102); //titik C  glEnd();    //atapdepanbawahtengahkanan  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.35,0.38,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.35,0.38,0.5); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.35,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.35,0.4,-0.102); //titik C  glEnd();    //atapdepanbawahtengahkiridalam  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.35,0.38,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.35,0.38,0.5); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.4,0.38,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.4,0.38,-0.102); //titik C  glEnd();    //atapdepanbawahtengahkanandalam  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.35,0.38,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.35,0.38,0.5); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.4,0.38,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.4,0.38,-0.102); //titik C  glEnd();    //atapdepanbawahtengahkiridalamluar  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.4,0.38,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.4,0.38,0.5); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.4,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.4,0.4,-0.102); //titik C  glEnd();    //atapdepanbawahtengahkanandalamluar  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.4,0.38,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.4,0.38,0.5); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.4,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.4,0.4,-0.102); //titik C  glEnd();    //atapdepanbawahtengahkiridalamluarbanget  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.4,0.4,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.76,0.4,-0.102); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.76,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.4,0.4,0.5); //titik C  glEnd();    //atapdepanbawahtengahkanandalamluarbanget  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.4,0.4,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.76,0.4,-0.102); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.76,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.4,0.4,0.5); //titik C  glEnd();    //atapdepanbawahtengahkiridalamluarbangetbanget  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.76,0.38,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.76,0.38,0.5); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.76,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.76,0.4,-0.102); //titik C  glEnd();    //atapdepanbawahtengahkanandalamluarbangetbanget  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.76,0.38,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.76,0.38,0.5); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.76,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.76,0.4,-0.102); //titik C  glEnd();    //atapdepanbawahtengahkiridalamluarbangetsekali  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.76,0.38,-0.18); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.8,0.38,-0.18); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.8,0.38,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.76,0.38,0.5); //titik C  glEnd();    //atapdepanbawahtengahkanandalamluarbangetsekali  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.76,0.38,-0.18); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.8,0.38,-0.18); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.8,0.38,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.76,0.38,0.5); //titik C  glEnd();    //atapdepanbawahtengahkiridalamluarbangetbangetluarbinasa  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.8,0.38,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.8,0.38,0.5); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.8,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.8,0.4,-0.102); //titik C  glEnd();    //atapdepanbawahtengahkanandalamluarbangetbangetluarbinasa  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.8,0.38,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.8,0.38,0.5); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.8,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.8,0.4,-0.102); //titik C  glEnd();    //atapdepanbawahtengahkiridalamluarbangetwahkeren  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.95,0.4,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.8,0.4,-0.102); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.8,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.95,0.4,0.5); //titik C  glEnd();    //atapdepanbawahtengahkanandalamluarbangetwahkeren  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.95,0.4,-0.102); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.8,0.4,-0.102); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.8,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.95,0.4,0.5); //titik C  glEnd();    //atapdepanbawahtengahkiridalamluarbangetwahkerenbreww  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(1.0,0.4,0.3); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.95,0.4,0.3); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.95,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(1.0,0.4,0.5); //titik C  glEnd();    //atapdepanbawahtengahkanandalamluarbangetwahkerenbreww  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-1.0,0.4,0.3); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.95,0.4,0.3); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.95,0.4,0.5); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-1.0,0.4,0.5); //titik C  glEnd();    //-------------------------------atapatas---------------------------------------  //atapdepanatastengah  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.35,1.0,-1.0); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.35,1.0,-0.04); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.35,1.0,-0.04); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.35,1.0,-1.0); //titik C  glEnd();    //atapdepanatastengahkiri  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.35,0.98,-0.04); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.35,0.98,-1.0); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.35,1.0,-1.0); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.35,1.0,-0.04); //titik C  glEnd();    //atapdepanatastengahkanan  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.35,0.98,-0.04); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.35,0.98,-1.0); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.35,1.0,-1.0); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.35,1.0,-0.04); //titik C  glEnd();    //atapdepanatastengahkiridalam  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.35,0.98,-0.1); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.35,0.98,-1.0); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.4,0.98,-1.0); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.4,0.98,-0.1); //titik C  glEnd();    //atapdepanatastengahkanandalam  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.35,0.98,-1); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.35,0.98,-0.1); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.4,0.98,-0.1); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.4,0.98,-1); //titik C  glEnd();    //atapdepanatastengahkiridalamluar  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.4,0.98,-0.1); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.4,0.98,-1); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.4,1,-1); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.4,1,-0.1); //titik C  glEnd();    //atapdepanatastengahkanandalamluar  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.4,0.98,-0.1); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.4,0.98,-1); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.4,1,-1); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.4,1,-0.1); //titik C  glEnd();    //atapdepanatastengahkiridalamluarbanget  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.4,1,-0.1); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.76,1,-0.1); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.76,1,-0.7); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.4,1,-0.7); //titik C  glEnd();    //atapdepanatastengahkiridalamluarbangetwow  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.4,1,-0.7); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.7,1,-0.7); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.7,1,-0.9); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.4,1,-0.9); //titik C  glEnd();    //atapdepanatastengahkiridalamluarbanget  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.4,1,-0.9); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.45,1,-0.9); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.45,1,-1); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.4,1,-1); //titik C  glEnd();    //atapdepanatastengahkiridalamluarbangetwow  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.4,1,-0.7); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.7,1,-0.7); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.7,1,-0.9); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.4,1,-0.9); //titik C  glEnd();    //atapdepanatastengahkiridalamluarbanget  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.4,1,-0.9); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.45,1,-0.9); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.45,1,-1); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.4,1,-1); //titik C  glEnd();    //atapdepanatastengahkanandalamluarbanget  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.4,1,-0.1); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.76,1,-0.1); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.76,1,-0.7); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.4,1,-0.7); //titik C  glEnd();    //atapdepanatastengahkiridalamluarbangetbanget  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.76,0.98,-0.1); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.76,0.98,-0.7); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.76,1,-0.7); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.76,1,-0.1); //titik C  glEnd();    //atapdepanatastengahkanandalamluarbangetbanget  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.76,0.98,-0.1); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.76,0.98,-0.7); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.76,1,-0.7); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.76,1,-0.1); //titik C  glEnd();    //atapdepanatastengahkiridalamluarbangetsekali  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.76,0.98,-0.7); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.8,0.98,-0.7); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.8,0.98,-0.18); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.76,0.98,-0.18); //titik C  glEnd();    //atapdepanatastengahkanandalamluarbangetsekali  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.76,0.98,-0.7); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.8,0.98,-0.7); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.8,0.98,-0.18); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.76,0.98,-0.18); //titik C  glEnd();    //atapdepanatastengahkiridalamluarbangetbangetluarbinasa  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.8,0.98,-0.7); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.8,0.98,-0.18); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.8,1,-0.14); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.8,1,-0.7); //titik C  glEnd();    //atapdepanatastengahkanandalamluarbangetbangetluarbinasa  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.8,0.98,-0.7); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.8,0.98,-0.18); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.8,1,-0.14); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.8,1,-0.7); //titik C  glEnd();    //atapdepanatastengahkiridalamluarbangetwahkeren  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(0.95,1,-0.7); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(0.8,1,-0.7); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.8,1,-0.1); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(0.95,1,-0.1); //titik C  glEnd();    //atapdepanatastengahkanandalamluarbangetwahkeren  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-0.95,1,-0.7); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.8,1,-0.7); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(-0.8,1,-0.1); //titik B  glTexCoord2f(0.0, 1.0);glVertex3f(-0.95,1,-0.1); //titik C  glEnd();    //----------------------------tower--------------------------------------------  //towerkiriatas  glBindTexture(GL\_TEXTURE\_2D, texture[21]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.02,1.1,-0.8); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.02,1.1,-1); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.02,1.05,-1); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.02,1.05,-0.8); //titik C  glEnd();    //towerkananatas  glBindTexture(GL\_TEXTURE\_2D, texture[21]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.02,1.1,-1); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.02,1.1,-0.8); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.02,1.05,-0.8); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.02,1.05,-1); //titik C  glEnd();    //tower tengahdepanatas  glBindTexture(GL\_TEXTURE\_2D, texture[22]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.02,1.1,-0.8); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.02,1.1,-0.8); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.02,1.05,-0.8); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.02,1.05,-0.8); //titik C  glEnd();    //tower tengahbelakangatas  glBindTexture(GL\_TEXTURE\_2D, texture[22]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.02,1.1,-1); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.02,1.1,-1); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.02,1.05,-1); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.02,1.05,-1); //titik C  glEnd();    //towerkiriatas  glBindTexture(GL\_TEXTURE\_2D, texture[21]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.02,1.1,-0.8); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.02,1.1,-1); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.02,1.05,-1); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.02,1.05,-0.8); //titik C  glEnd();    //towerkananatas  glBindTexture(GL\_TEXTURE\_2D, texture[21]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.02,1.1,-1); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.02,1.1,-0.8); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.02,1.05,-0.8); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.02,1.05,-1); //titik C  glEnd();    //tower tengahdepanatas  glBindTexture(GL\_TEXTURE\_2D, texture[22]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.02,1.1,-0.8); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.02,1.1,-0.8); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.02,1.05,-0.8); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.02,1.05,-0.8); //titik C  glEnd();    //tower tengahbelakangatas  glBindTexture(GL\_TEXTURE\_2D, texture[22]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.02,1.1,-1); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.02,1.1,-1); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.02,1.05,-1); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.02,1.05,-1); //titik C  glEnd();    //towerkiriatasabu  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.02,1.05,-0.8); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.02,1.05,-1); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.02,1,-1); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.02,1,-0.8); //titik C  glEnd();    //towerkananatasabu  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.02,1.05,-1); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.02,1.05,-0.8); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.02,1,-0.8); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.02,1,-1); //titik C  glEnd();    //towerkiriatasmerah  glBindTexture(GL\_TEXTURE\_2D, texture[22]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.02,1.05,-0.75); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.02,1.05,-0.8); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.02,1,-0.8); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.02,1,-0.75); //titik C  glEnd();    //towerkananatasmerah  glBindTexture(GL\_TEXTURE\_2D, texture[22]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.02,1.05,-0.8); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.02,1.05,-0.75); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.02,1,-0.75); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.02,1,-0.8); //titik C  glEnd();    //tower tengahdepanatas  glBindTexture(GL\_TEXTURE\_2D, texture[22]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.02,1.05,-0.75); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.02,1.05,-0.75); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.02,1,-0.75); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.02,1,-0.75); //titik C  glEnd();    //tower tengahdepanatastutup  glBindTexture(GL\_TEXTURE\_2D, texture[22]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.02,1.05,-0.8); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.02,1.05,-0.8); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.02,1.05,-0.75); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.02,1.05,-0.75); //titik C  glEnd();    //tower tengahbelakangatas  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.02,1.05,-1); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.02,1.05,-1); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.02,1,-1); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.02,1,-1); //titik C  glEnd();    //tower tengahdepanatastutupmerah  glBindTexture(GL\_TEXTURE\_2D, texture[22]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.02,1.1,-0.8); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.02,1.1,-1); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.02,1.1,-1); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.02,1.1,-0.8); //titik C  glEnd();    //--------------------------teras kaca------------------------------------------  //teras kiri  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.19,-0.35,0.56); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.2,-0.35,0.56); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.2,-0.5,0.56); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.19,-0.5,0.56); //titik C  glEnd();  //teras kanan  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.19,-0.35,0.56); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.2,-0.35,0.56); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.2,-0.5,0.56); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.19,-0.5,0.56); //titik C  glEnd();  //teras kiri depan  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.19,-0.35,0.8); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(-0.2,-0.35,0.8); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(-0.2,-0.5,0.8); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.19,-0.5,0.8); //titik C  glEnd();  //teras kanan depan  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(0.19,-0.35,0.8); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.2,-0.35,0.8); //titik A  glTexCoord2f(0.0, 0.0);glVertex3f(0.2,-0.5,0.8); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.19,-0.5,0.8); //titik C  glEnd();  //teras atap  glBindTexture(GL\_TEXTURE\_2D, texture[19]);    glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-0.2,-0.35,-0.8); //titik D  glTexCoord2f(0.0, 1.0);glVertex3f(0.2,-0.35,-0.8); //titik A  glTexCoord2f(1.0, 1.0);glVertex3f(0.2,-0.35,0.8); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(-0.2,-0.35,0.8); //titik C  glEnd();    }  void tower(){  GLUquadricObj \*p = gluNewQuadric();  //batang  glPushMatrix();  glBindTexture(GL\_TEXTURE\_2D, texture[19]);  glTranslatef(0.0,-0.5,0.0);  glRotatef(180, 0.0, 1.0, 1.0);  glScalef(0.1, 0.1, 5);  gluCylinder(p, 0.04, 0.04, 0.04, 32, 32);  glPopMatrix();  }  //--------------------------Procedure pohon-------------------------------------  void daun()  {  //bawah  glBegin(GL\_POLYGON);  glVertex3f(-0.15,-0.3,0.2); //titik A  glVertex3f(0.15,-0.3,0.2); //titik B  glVertex3f(0.2,-0.3,0.0); //titik C  glVertex3f(0.15,-0.3,-0.2); //titik D  glVertex3f(-0.15,-0.3,-0.2); //titik E  glVertex3f(-0.2,-0.3,0.0); //titik F  glEnd();    //depan  glBindTexture(GL\_TEXTURE\_2D, texture[9]);  glBegin(GL\_POLYGON);  glTexCoord2f(0.5, 1.0);glVertex3f(0,0.2,0); //titik G  glTexCoord2f(0.0, 0.0);glVertex3f(-0.15,-0.3,0.2); //titik A  glTexCoord2f(1.0, 0.0);glVertex3f(0.15,-0.3,0.2); //titik B  glEnd();    //kanan depan  glBindTexture(GL\_TEXTURE\_2D, texture[9]);  glBegin(GL\_POLYGON);  glTexCoord2f(0.5, 1.0);glVertex3f(0,0.2,0); //titik G  glTexCoord2f(0.0, 0.0);glVertex3f(0.15,-0.3,0.2); //titik B  glTexCoord2f(1.0, 0.0);glVertex3f(0.2,-0.3,0.0); //titik C  glEnd();    //kanan belakanggl  glBegin(GL\_POLYGON);  glTexCoord2f(0.5, 1.0);glVertex3f(0,0.2,0); //titik G  glTexCoord2f(0.0, 0.0);glVertex3f(0.2,-0.3,0.0); //titik C  glTexCoord2f(1.0, 0.0);glVertex3f(0.15,-0.3,-0.2); //titik D  glEnd();    //belakang  glBindTexture(GL\_TEXTURE\_2D, texture[9]);  glBegin(GL\_POLYGON);  glTexCoord2f(0.5, 1.0);glVertex3f(0,0.2,0); //titik G  glTexCoord2f(0.0, 0.0);glVertex3f(0.15,-0.3,-0.2); //titik D  glTexCoord2f(1.0, 0.0);glVertex3f(-0.15,-0.3,-0.2); //titik E  glEnd();    //kiri belakang  glBindTexture(GL\_TEXTURE\_2D, texture[9]);  glBegin(GL\_POLYGON);  glTexCoord2f(0.5, 1.0);glVertex3f(0,0.2,0); //titik G  glTexCoord2f(0.0, 0.0);glVertex3f(-0.15,-0.3,-0.2); //titik E  glTexCoord2f(1.0, 0.0);glVertex3f(-0.2,-0.3,0.0); //titik F  glEnd();    //kiri depan  glBindTexture(GL\_TEXTURE\_2D, texture[9]);  glBegin(GL\_POLYGON);  glBindTexture(GL\_TEXTURE\_2D, texture[9]);  glTexCoord2f(0.5, 1.0);glVertex3f(0,0.2,0); //titik G  glTexCoord2f(0.0, 0.0);glVertex3f(-0.2,-0.3,0.0); //titik F  glTexCoord2f(1.0, 0.0);glVertex3f(-0.15,-0.3,0.2); //titik A  glEnd();  }  void pohon(){  GLUquadricObj \*p = gluNewQuadric();  //batang  glPushMatrix();  glBindTexture(GL\_TEXTURE\_2D, texture[10]);  glTranslatef(0.0,-0.5,0.0);  glRotatef(180, 0.0, 1.0, 1.0);  glScalef(1, 1, 5);  gluCylinder(p, 0.04, 0.04, 0.04, 32, 32);  glPopMatrix();  //daun  daun();  y=0.12;  for(i=1;i<2;i++){  glPushMatrix();  glTranslatef(0.0,y, 0.0);  daun();  glPopMatrix();  y=y+0.12;  }  }  //--------------------------Procedure jalan -----------------------------------  void jalan()  {  glBindTexture(GL\_TEXTURE\_2D, texture[14]);  glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-3,-0.49,0.56); //atas kiri  glTexCoord2f(1.0, 0.0);glVertex3f(3,-0.49,0.56); //atas kanan  glTexCoord2f(1.0, 1.0);glVertex3f(3,-0.49,1);//kanan bawah  glTexCoord2f(0.0, 1.0);glVertex3f(-3,-0.49,1); //kiri bawah  glEnd();  }  void trotoar()  {  glBindTexture(GL\_TEXTURE\_2D, texture[13]);  glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-3,-0.49,0.54); //atas kiri  glTexCoord2f(1.0, 0.0);glVertex3f(3,-0.49,0.54); //atas kanan  glTexCoord2f(1.0, 1.0);glVertex3f(3,-0.49,0.56);//kanan bawah  glTexCoord2f(0.0, 1.0);glVertex3f(-3,-0.49,0.56); //kiri bawah  glEnd();    glBindTexture(GL\_TEXTURE\_2D, texture[13]);  glBegin(GL\_QUADS);  glTexCoord2f(0.0, 0.0);glVertex3f(-3,-0.49,1); //atas kiri  glTexCoord2f(1.0, 0.0);glVertex3f(3,-0.49,1); //atas kanan  glTexCoord2f(1.0, 1.0);glVertex3f(3,-0.49,1.02);//kanan bawah  glTexCoord2f(0.0, 1.0);glVertex3f(-3,-0.49,1.02); //kiri bawah  glEnd();  }  //---------------------------Procedure Parkiran---------------------------------  void parkiran()  {  glBindTexture(GL\_TEXTURE\_2D, texture[12]);  glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(1,-0.49,0.54); //atas kiri  glTexCoord2f(0.0, 1.0);glVertex3f(1.7,-0.49,0.54); //atas kanan  glTexCoord2f(0.0, 0.0);glVertex3f(1.7,-0.49,-0.54);//kanan bawah  glTexCoord2f(1.0, 0.0);glVertex3f(1,-0.49,-0.54); //kiri bawah  glEnd();  }  //---------------------------procedure background-------------------------------  void background(){  glBindTexture(GL\_TEXTURE\_2D, texture[11]);  glBegin(GL\_QUADS);  glTexCoord2f(1.0, 1.0);glVertex3f(-3,5,-3); //atas kiri  glTexCoord2f(0.0, 1.0);glVertex3f(3,5,-3); //atas kanan  glTexCoord2f(0.0, 0.0);glVertex3f(3,-0.5,-3);//kanan bawah  glTexCoord2f(1.0, 0.0);glVertex3f(-3,-0.5,-3); //kiri bawah    glEnd();  }  //--------------------------Procedure kamera------------------------------------  void kamera(){  gluLookAt (0 ,1 , 5 ,0 , p , 1 , 0 , 4 , 1 ) ;  glTranslatef( tx , ty , tz ) ;  glRotatef(roty , 0 , 1 , 0 ) ;  }  /\*----------------------------------------------------------------------------\*/  void display(void){  glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);  glEnable(GL\_COLOR\_MATERIAL); //mengaktifkan fungsi untuk memberikan warna pada material  glColorMaterial(GL\_FRONT, GL\_DIFFUSE); //parameter-parameter fungsi warna material  // Set the camera orientation  glMatrixMode(GL\_MODELVIEW) ;  glLoadIdentity ( ) ;  kamera();    rumput();  dinding\_gedung();  //tower  glPushMatrix();  glTranslatef(0,1.6,-0.98);  tower();  glPopMatrix();    //pohon kiri  glPushMatrix();  glTranslatef(-1.5,0,0.35);  pohon();  glPopMatrix();  //pohon kanan  glPushMatrix();  glTranslatef(-1.8,0,0.35);  pohon();  glPopMatrix();    //jalan  jalan();  trotoar();  //parkiran  parkiran();    //gambar background  background();  glutSwapBuffers();    }  void resize(int w1, int h1){  glViewport(0,0,w1,h1);  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluPerspective(45.0,(float) w1/(float) h1, 1.0,300.0);  glMatrixMode(GL\_MODELVIEW);  glLoadIdentity();  }    //special-key event untuk fullscreen atau normal  void specialKey(int key, int x, int y) {  switch (key) {  case GLUT\_KEY\_F1: // F1: Toggle between full-screen and windowed mode  fullScreenMode = !fullScreenMode; // Toggle state  if (fullScreenMode) { // Full-screen mode  windowPosX = glutGet(GLUT\_WINDOW\_X); // Save parameters  windowPosY = glutGet(GLUT\_WINDOW\_Y);  windowWidth = glutGet(GLUT\_WINDOW\_WIDTH);  windowHeight = glutGet(GLUT\_WINDOW\_HEIGHT);  glutFullScreen(); // Switch into full screen  } else { // Windowed mode  glutReshapeWindow(windowWidth, windowHeight); // Switch into windowed mode  glutPositionWindow(windowPosX, windowPosX); // Position top-left corner  }  break;  default:  break;  }  }  /\*----------------------------------------------------------------------------\*/  //program utama  int main (int argc, char \*\*argv){  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_DEPTH | GLUT\_RGBA);  glutInitWindowSize(windowWidth, windowHeight); // Initial window width and height  glutInitWindowPosition(windowPosX, windowPosY); // Initial window top-left corner (x, y)  glutCreateWindow(title); // Create window with the given title  glutFullScreen(); // Pull into full screen  glutDisplayFunc(display);  glutReshapeFunc(resize);  glutKeyboardFunc(myKeyboard); // memanggil Prosedure interaksi pada keyboard  glutSpecialFunc(specialKey); // memanggil special-key event  glutIdleFunc(display);  init();  glutMainLoop();  return 0;  } |

## Tampilan Program OpenGL



Gambar 2 Tampilan Program

# KESIMPULAN DAN SARAN

## Kesimpulan

Berdasarkan permasalahan yang telah diteliti, maka kesimpulan dari penulisan tugas besar ini adalah :

1. Dapat membangun aplikasi untuk memodelkan suatu tempat yang memiliki nilai konsep yang baik.
2. Dapat membangun suatu pemodelan untuk memodelkan bangunan Gedung Unikom yang baru.
3. Juga dapat memahami dan mengaplikasikan materi openGL dari mata kuliah Grafika Komputer dengan baik.

## Saran

Besar harapan kami agar saran dan kritik dari pembaca dapat membangun dan memperbaiki segala kekurangan, agar kami dapat menghasilkan projek yang lebih baik lagi.