

PRAKTIKUM I

PENGENALAN *OPENGL*

A. Tujuan

- Mampu melakukan instalasi *Microsoft Visual Studio (Visual C++)*
- Mampu melakukan konfigurasi *Library OpenGL* di *Microsoft Visual Studio*
- Mampu memahami inisialisasi fungsi *OpenGL*

B. Materi

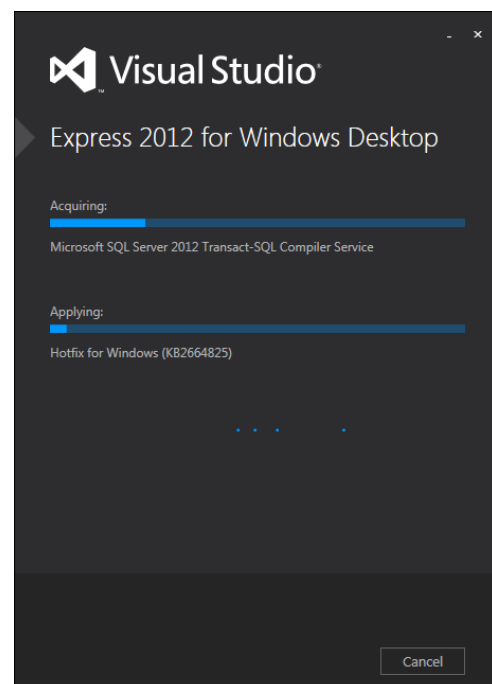
- Instalasi *Microsoft Visual Studio (Visual C++)*
- Konfigurasi *Library OpenGL*
- Inisialisasi fungsi *OpenGL*

C. Perangkat Yang Digunakan

- Komputer
- Sistem operasi *Windows*
- Microsoft Visual Studio (Visual C++)*
- Library OpenGL*

D. Instalasi *Microsoft Visual Studio (Visual C++)*

Pada modul praktikum ini menggunakan *Microsoft Visual Studio Express 2012*, Anda dapat menggunakan versi di atasnya. Proses instalasi *Microsoft Visual Studio Express 2012* ditampilkan pada gambar berikut:





Microsoft Visual Studio Express 2012 berhasil diinstal, klik Launch untuk menjalankan program.

E. OpenGL

OpenGL adalah sebuah program aplikasi interface yang digunakan untuk mendefinisikan komputer grafis 2D dan 3D. Program lintas-*platform* API ini umumnya dianggap ketetapan standar dalam industri komputer dalam interaksi dengan komputer grafis 2D dan juga telah menjadi alat yang biasa untuk digunakan dengan grafis 3D. Singkatnya, *Open Graphics Library*, *OpenGL* menghilangkan kebutuhan untuk pemrogram untuk menulis ulang bagian grafis dari sistem operasi setiap kali sebuah bisnis akan di*upgrade* ke versi baru dari sistem.

Fungsi dasar dari *OpenGL* adalah untuk mengeluarkan koleksi perintah khusus atau *executable* ke sistem operasi. Dengan demikian, program ini bekerja dengan perangkat keras grafis yang ada yang berada pada *hard drive* atau sumber tertentu lainnya. Setiap perintah dalam dirancang untuk melakukan tindakan tertentu, atau memulai efek khusus tertentu yang terkait dengan grafis.

Membuat perintah dalam *OpenGL* dapat terjadi dalam dua cara yang berbeda. Pertama, adalah mungkin bagi *programmer* untuk membuat dan menyimpan daftar perintah yang dapat dieksekusi secara berulang. Ini adalah salah satu cara yang lebih rutin untuk program *interface* yang digunakan. Seiring dengan berkembangnya kelompok perintah yang kurang lebih permanen, maka memungkinkan untuk membuat dan menjalankan salah satu perintah dalam batas-batas waktu dari komputer grafis.

Seiring dengan kemampuan *interface* dari sistem operasi, *OpenGL* juga menyediakan beberapa *built-in* protokol yang mungkin berguna bagi pengguna akhir. Di antaranya fitur alat seperti *alpha blending*, pemetaan tekstur, dan efek atmosfer. Alat ini dapat berinteraksi dengan sistem operasi yang sedang digunakan.

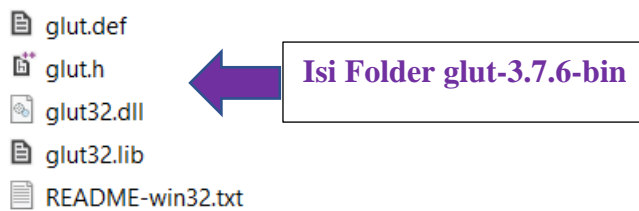
Awalnya dikembangkan oleh Silicon Graphics, *OpenGL* kini dianggap standar industri. *Interface* program aplikasi yang aktif didukung oleh Microsoft ini, menawarkan *download* gratis daftar perintah *OpenGL* untuk digunakan pada sistem Windows. *OpenGL* juga bekerja sangat baik dengan Inventor Open, sebuah pemrograman berorientasi obyek alat juga diciptakan oleh Silicon Graphics.

OpenGL lebih mengarah pada prosedural daripada sebuah deskriptif API grafis. Untuk mendeskripsikan *scene* dan bagaimana penampilannya, sebenarnya *programmer* lebih tau untuk menentukan hal-hal yang dibutuhkan untuk menghasilkan efek yang diinginkan.

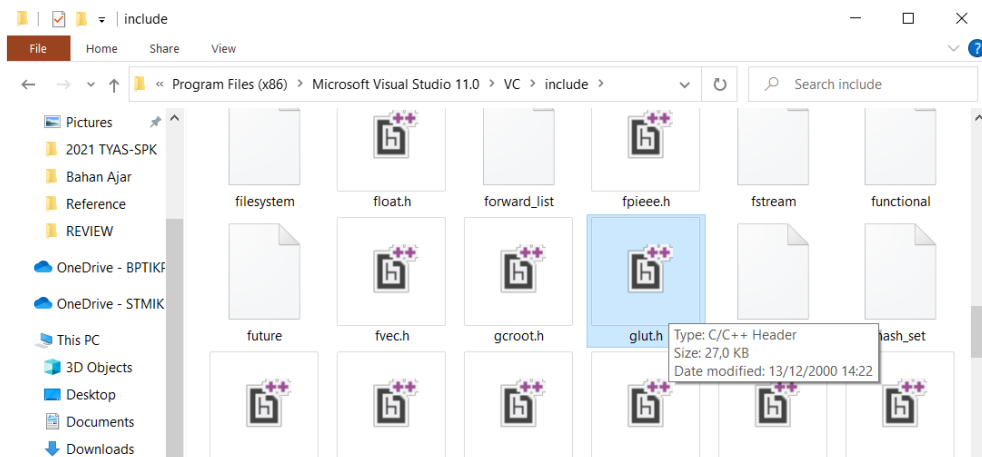
Langkah tersebut termasuk memanggil banyak perintah *OpenGL*, perintah tersebut digunakan untuk menggambarkan grafis primitif seperti titik, garis dan poligon dalam tiga dimensi. Sebagai tambahan, *OpenGL* mendukung *lighting*, *shading*, *texture mapping*, *blending*, *transparency*, dan banyak kemampuan efek khusus lainnya.

F. Konfigurasi *Library OpenGL*

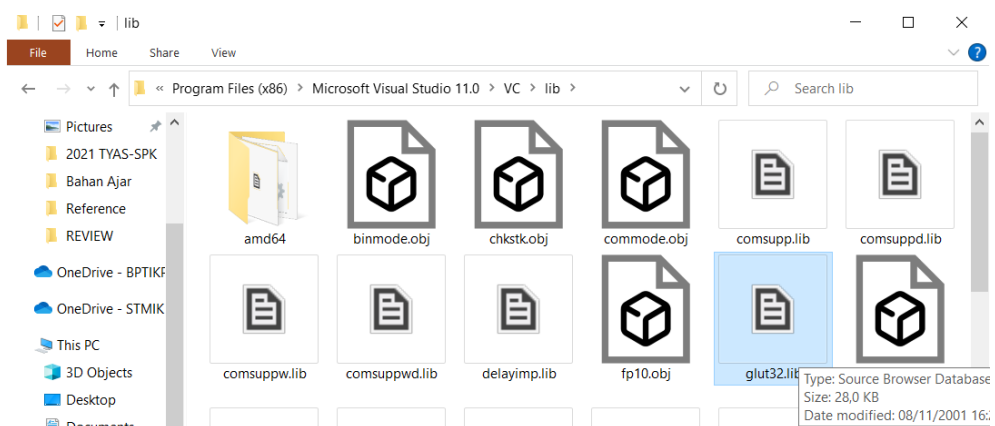
GLUT (*GL Utility Toolkit*) merupakan pengembangan dari *OpenGL* yang didesain untuk aplikasi dengan level kecil hingga menengah dan menggunakan *callback functions* untuk menambahkan interaksi dari user. Untuk pemrograman *OpenGL* menggunakan C++, diperlukan *library* tambahan yaitu:



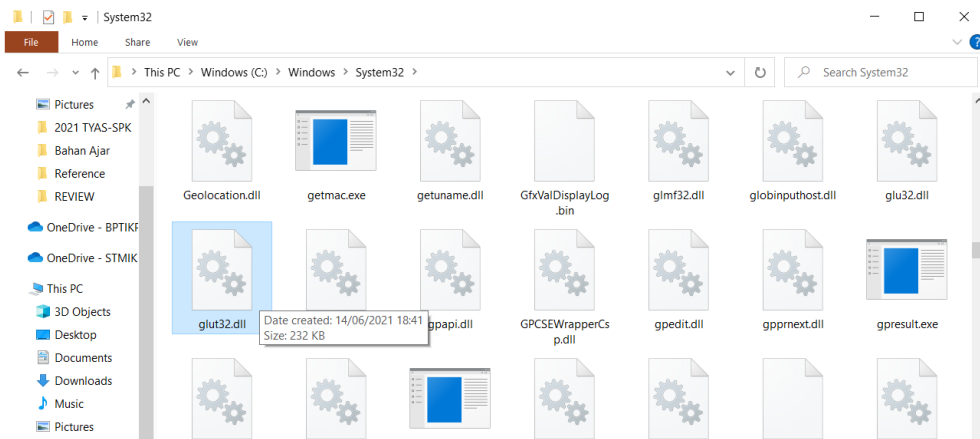
- a. **glut.h** yang dicopy ke direktori **C:\Program Files\Microsoft Visual Studio\VC\include**



- b. **glut32.lib** yang dicopy ke direktori **C:\Program Files\Microsoft Visual\VC\lib**



c. **glut32.dll** yang dicopy ke direktori **C:\Windows\System32**



G. Inisialisasi Awal

Inti dari tahapan ini adalah mengatur *view port* dan *persepektif* untuk penampilan obyek ke dalam layar monitor, *viewport* adalah besarnya layer monitor (*image*) yang dipakai untuk menampilkan obyek, sedangkan *persepektif* yang dimaksud adalah pengaturan sumbu z dalam penampilan obyek 3 dimensi, sehingga *user* dapat melihat obyek seolah-olah dalam bidang 3 dimensi (X-Y-Z), selain itu penggambaran obyek yang dilakukan oleh *programmer* juga dapat menggunakan koordinat 3 dimensi. Selain ke dua tujuan di atas pada tahap ini juga dilakukan koneksi awal dengan *library OpenGL*, koneksi ini dilakukan supaya fungsi-fungsi yang di sediakan *OpenGL* dapat digunakan. Fungsi/ prosedur yang digunakan:

Fungsi/ Prosedur	Pengertian/ Kegunaan
LoadGlut(glut32.dll) “if you no include gl in system”	Pemanggilan <i>library OpenGL</i>
InitGL	Inisialisasi <i>OpenGL</i> awal yang harus dilakukan
glViewport	Untuk pengaturan <i>viewport</i>
glMatrixMode	Untuk pengaturan <i>viewport</i>

a. Fungsi Pembuatan Gambar

Fungsi / Prosedur	Pengertian/ Kegunaan
GL_QUARDS	Untuk menggambar segi empat
GL_LINES	Untuk menggambar sebuah garis
glVertex3f	Untuk menentukan titik 3 dimensi
glColor3f	Untuk penentuan suatu warna
glLineWidth	Untuk mengatur tebalnya garis
glPointSize	Untuk mengatur besarnya suatu objek,

gluOrtho2D	Untuk mengatur proyeksi hasil eksekusi dan mendefinisikan besarnya sistem koordinat dengan urutan kirikanan dan bawah-atas.
glClearColor	Menandakan warna apa yang digunakan sebagai background. Warna dinyatakan dalam bentuk RGBA, angka pertama memwakil warna merah, kedua hijau, ketiga biru, dan keempat adalah alpha atau tingkat transparansi. Tiap warna berkisar antara 0 hingga 1.
glClear	Untuk menghapus window dan memberikan warna yang telah kita definisikan sebelumnya dengan menggunakan glClearColor
glPushMatrix	Untuk menyimpan koordinat Matrik yang ada
glPopMatrix	Untuk memanggil suatu fungsi yang telah disimpan pada glPushMatrix
glBegin(GL_Lines)	Untuk memulai menggambar dalam bentuk garis. Setiap pasangan verteks sebagai segemen garis independen
glEnd	Untuk mengakhiri penggambaran sebuah garis
SwapBuffers(hdc)	Untuk menukar bagian belakang buffer menjadi buffer screen
glflush()	Rendering

b. Fungsi Dasar Menggambar Titik

Fungsi / Prosedur	Pengertian/ Kegunaan	Contoh
glVertex2i(x,y)	Yaitu suatu fungsi untuk menggambar titik pada koordinat x dan y, dengan nilai satuan berupa integer.	glVertex2i(10,10)
glVertex2f(x,y)	Yaitu suatu fungsi untuk menggambar titik pada koordinat x dan y, dengan nilai satuan berupa float.	glVertex2f(10.0,10.0)
glVertex2d(x,y)	Yaitu suatu fungsi untuk menggambar titik pada koordinat x dan y, dengan nilai satuan berupa double.	glVertex2d(10.0,10.0)

c. Fungsi Dasar Menggambar Primitive Objek

Fungsi / Prosedur	Pengertian/ Kegunaan
GL_POINTS	Untuk menggambar suatu titik
GL_LINES	Untuk menggambar suatu garis.
GL_LINE_LOOP	Untuk menarik sekelompok verteks yang terhubung menjadi segmen garis dari titik pertama sampai terakhir, kemudian kembali ke verteks yang pertama.
GL_LINE_STRIP	Untuk menggambar sekelompok verteks yang terhubung menjadi segitiga. Sebuah segitiga baru didefinisikan untuk setiap titik yang diberikan setelah dua verteks pertama dan seterusnya.
GL_TRIANGLES	Untuk menggambar triangle atau segitiga, yaitu adalah tiga buah titik yang terhubung menjadi suatu segitiga dengan blok di tengahnya.
GL_TRIANGLES_STRIP	Untuk menggambar pada triangles strip jumlah vertex yang dipergunakan adalah 3 buah vertex.
GL_TRIANGLES_FAN	Untuk menggambar triangles fan, yaitu pembuatan suatu objek dengan menggunakan segitiga dimana hanya menggunakan 1 titik pusat saja.
GL_QUADS	Untuk menggambar Quad atau segempat, yaitu empat buah titik yang terhubung menjadi quat segi empat dengan blok di tengahnya.
GL_QUADS_STRIP	Untuk menggambar pada quads strip 4 buah vertex merupakan 1 kelompok.
GL_POLYGON	Untuk menggambar polygon, dimana merupakan suatu fungsi yang mirip dengan polyline, tetapi menghasilkan kurva tertutup dengan blok warna (fill).

d. Fungsi Pewarnaan

Fungsi	Warna
glColor3f(0,0,0)	Black
glColor3f(0,0,1)	Blue
glColor3f(0,1,0)	Green
glColor3f(0,1,1)	Cyan
glColor3f(1,0,0)	Red
glColor3f(1,0,1)	Magenta

<code>glColor3f(1,1,0)</code>	Yellow
<code>glColor3f(1,1,1)</code>	White

e. Fungsi Transformasi Objek

Fungsi / Prosedur	Pengertian/ Kegunaan
<code>glTranslatef(x,y,z);</code>	<ul style="list-style-type: none"> • Parameter pertama menunjukkan translasi berdasarkan sumbu x. • Parameter ke dua menunjukkan translasi terhadap sumbu y. • Parameter ke tiga menunjukkan translasi terhadap sumbu z.
<code>glRotatef(sudut,x,y,z);</code>	<ul style="list-style-type: none"> • Parameter pertama merupakan nilai perputaran sudut yang diinginkan. • Parameter ke dua adalah perputaran terhadap sumbu x, parameter ke tiga merupakan perputaran terhadap sumbu y. • Parameter ke empat adalah perputaran terhadap sumbu z. • Parameter ke dua, ke tiga, dan ke empat harus bernilai 1 untuk menjadi sumbu putar sempurna. Dan untuk pemberian nilainya akan menyebabkan perputaran searah dengan jarum jam ketika diberi nilai negatif, dan sebaliknya.
<code>glScalef(x,y,z);</code>	<ul style="list-style-type: none"> • Parameter pertama, kedua, dan ke tiga masing-masing menunjukkan nilai skala untuk obyek berdasarkan sumbu x, y, dan z. Jika nilai skala adalah 1, maka ukuran obyek tersebut adalah tetap.