

Praktikum 1

Instalasi OpenGL

Tujuan :

1. Dapat menginstal dan menjalankan program dengan library glut
2. Memahami area kerja dan bentuk fungsi OpenGL
3. Mengenal Grafik Promitif Dasar dan menggambarannya menggunakan OpenGL

Pelaksanaan :

1. Membuat fungsi OpenGL
2. Melakukan Configurasi OpenGL

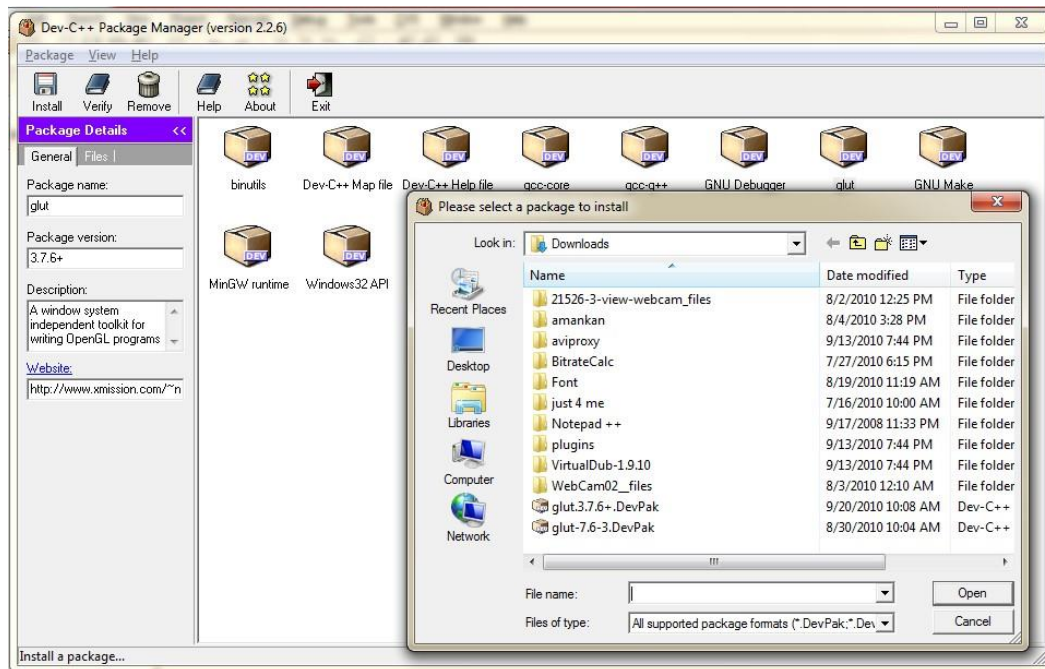
I. Instalasi OpenGL

OpenGL (Open Graphic Library) adalah sebuah library terdiri dari berbagai macam fungsi dan biasanya digunakan untuk menggambar sebuah objek 2D atau 3D. Untuk dapat menggunakan OpenGL dalam Dev C++, diperlukan library OpenGL. Library ini bisa didapatkan di www.opengl.org / www.devpack.org. Sebelum anda mulai menggunakan OpenGL, anda harus menginstall terlebih dahulu glut yang telah di download, versi terbaru saat modul ini ditulis adalah glut.3.7.6+.DevPak dan aplikasi Dev C++ versi 4.9.9.2 (5.0 beta). Berikut ini langkah – langkah untuk menginstall opengl :

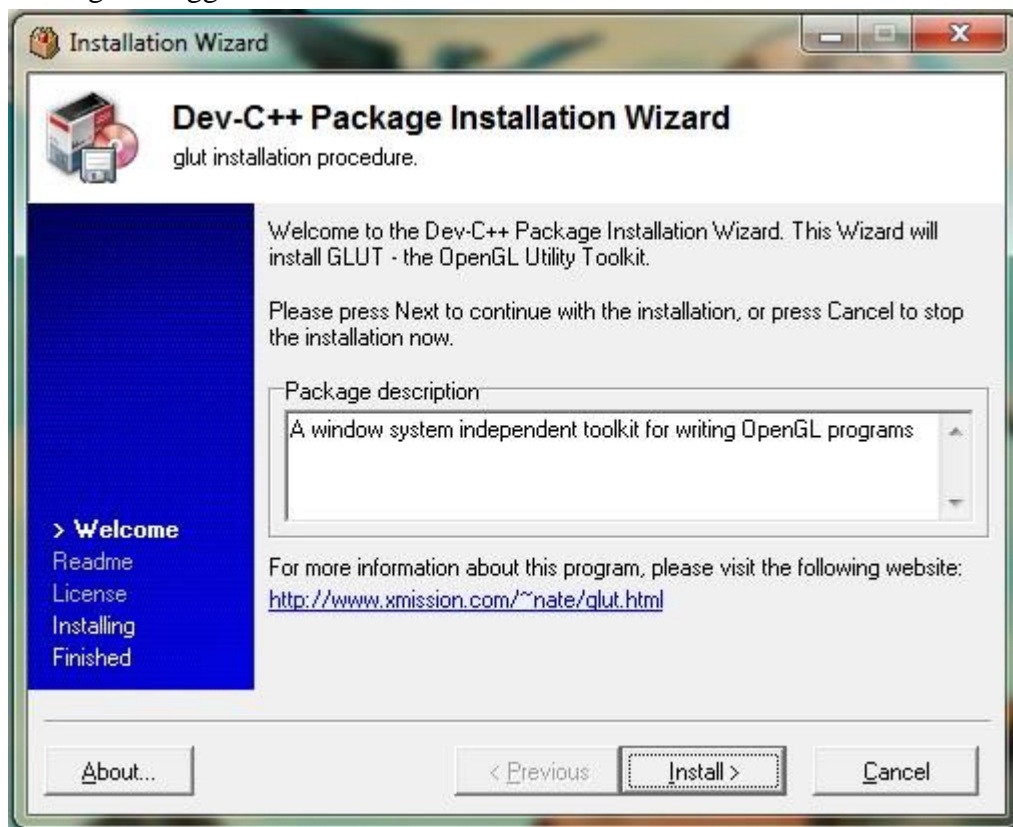
1. Setelah Dev C++ terinstal, jalankan Dev C++ lalu klik *Tools*, pilih *Package Manager*.



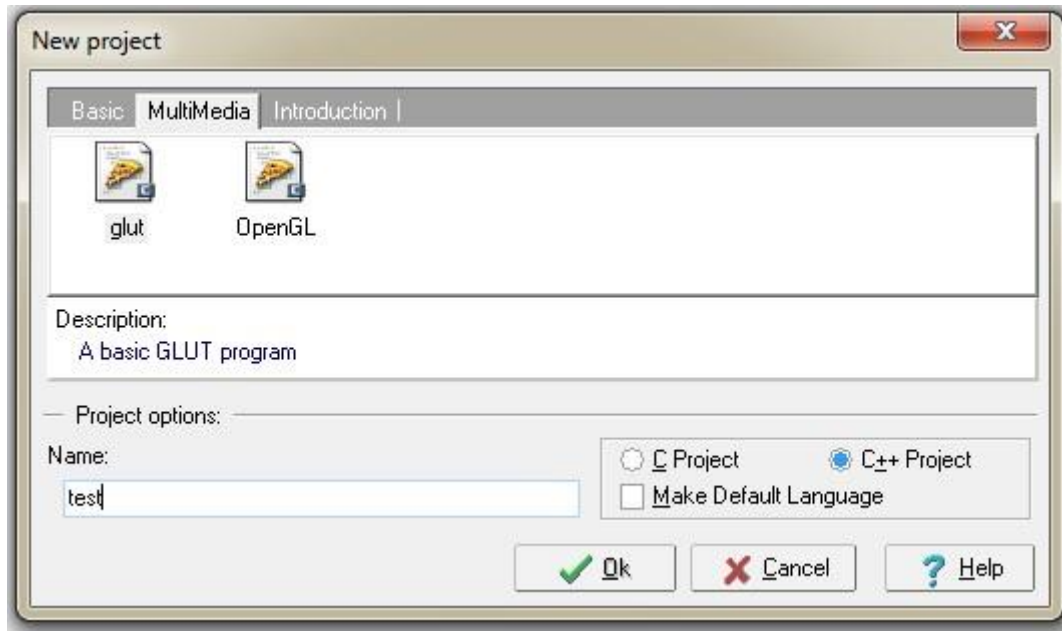
2. Jendela *Package Manager* akan terbuka, pilih glut kemudian klik icon Install di pojok kiri atas, Pilih file glut.3.7.6+.DevPak yang sudah Anda download lalu pilih open.



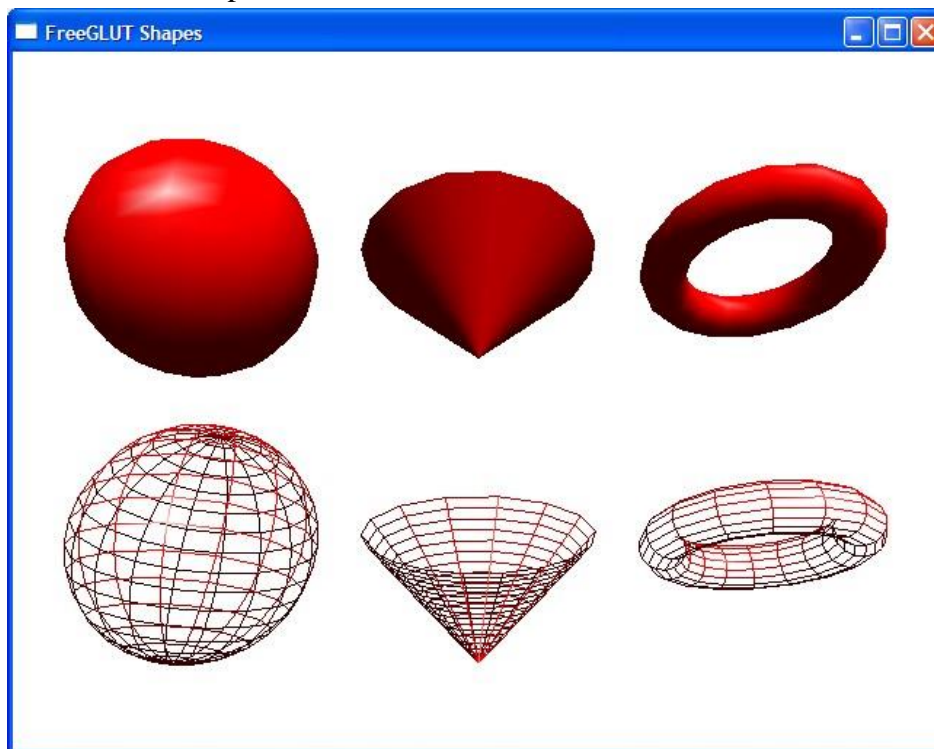
3. Install glut hingga selesai



4. Setelah menginstal lalu lakukan test apakah glut sudah terinstal dengan benar. Pilih File, New, Project. Pilih tab Multimedia, dan pilih glut, beri nama project di bagian bawah.



5. Simpan file project dan file code-nya. Setelah menyimpan lalu tekan F9. Jika berhasil maka akan muncul tampilan berikut.



Instalasi untuk GLUT

Download file di http://bit.ly/instalasi_glut

Langkah 1:

1. Download file glut-7.6.3 kemudian ekstrak .rar nya (terserah dimana aja boleh)
2. Buka program devc++, kemudian klik pada menu Tools kemudian pilih Package Manager
3. Klik menu Install kemudian masukkan file glut-7.6.3 yang tadi sudah di ekstrak
4. Kemudian install file glut-7.6.3 tadi ikuti hingga finish

Langkah 2:

1. Download file glutming kemudian ekstrak .rar nya (terserah dimana aja boleh)
2. Setelah file diekstrak, pindahkan file glut32.dll ke dalam folder sistem windows.
Ke dalam salah satu folder sesuai Bit nya.
 - Untuk windows 32 bit pindahkan ke direktori C:\Windows\System32
 - Untuk windows 64 bit pindahkan ke direktori C:\Windows\SysWOW64
3. Selanjutnya copy folder "lib" pada ekstrak tadi ke dalam folder DevC++ dan paste-kan pada folder :
 - C:\Dev-Cpp\ (paste disini)
4. Kemudian masuk pada folder "lib" tadi yang berisi 4 file kemudian copy 4 file tersebut.
dan paste-kan pada folder :
 - C:\Dev-Cpp\MinGW32\lib (paste disini)
5. Selanjutnya copy folder "include" pada ekstrak tadi ke dalam folder DevC++
 - C:\Dev-Cpp\ (paste disini)
6. Kemudian masuk pada folder "include" tadi yang berisi 2 folder kemudian copy 2 folder tersebut.
dan paste-kan pada folder :
 - C:\Dev-Cpp\MinGW32\include\ (paste disini)

Langkah 3 :

1. Setelah melakukan settingan tambahan di atas, proses instalasi sudah selesai 100%
Klik menu File -> New -> Project.
2. Klik pada tab menu yang kosong (tab antara Win32 dan Console),
lalu klik tab Misc. Pilih glut lalu klik Ok.

PRAKTIKUM 2

Primitif Drawing

Tujuan :

1. Mengetahui Grafik Primitif Dasar dan menggambarannya menggunakan OpenGL
2. Membandingkan pembuatan grafik primitif dasar dengan C++

Pelaksanaan :

1. Mengetahui fungsi dan penggunaan OpenGL
2. Melaksanakan proyek penggambaran menggunakan Dev C++

Primitif Drawing

Penghasilan citra pada komputer grafik menggunakan primitif grafik dasar. Primitif ini memudahkan untuk merender (menggambar pada layar monitor) sebagaimana penggunaan persamaan geometrik sederhana. Contoh primitif grafik dasar adalah Titik, Garis, Segiempat, Kurva, Lingkaran, Ellipse, Kurva Bezier, Text, Fill Area.

Sebelum membuat objek primitif, kita buat contoh program sederhana menggunakan OpenGL. Program berikut hanya akan memunculkan satu layar biru kosong, bila program berikut ini dapat berjalan dengan baik maka kita sudah siap memulai membuat grafik dengan OpenGL.

```
#include <gl/glut.h> void
Jendela(void)
{
    //membersihkan layar dan men-set dalam format warna
    glClear(GL_COLOR_BUFFER_BIT);
    //memastikan seluruh perintah tereksekusi
    glFlush();
} int main(int argc, char*
argv[])
{
    glutInit(&argc, argv);
    glutInitWindowSize(640,480);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
    glutCreateWindow("biru");
    glutDisplayFunc(Jendela);    glClearColor (0.0f,
0.0f, 1.0f, 1.0f);    glutMainLoop();    return
0;
}
```

Membuat Titik

Untuk membuat titik, anda harus menggunakan prosedur (GL_POINTS)

```
#include <gl/glut.h> void
point(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glPointSize(5.0f);
    glBegin(GL_POINTS);
    glColor3f(1.0f, 0.0f, 0.0f);
    glVertex3f(0.25, 0.25, 0.0);
    glEnd();    glFlush();
}

int main(int argc, char*
argv[])
{
    glutInit(&argc, argv);
    glutInitWindowSize(640,480);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
    glutCreateWindow("Titik");
    glutDisplayFunc(point);    glClearColor(0.0f,
0.0f, 1.0f, 1.0f);    glutMainLoop();    return
0;
}
```

glPointSize(5.0f) digunakan untuk mengatur besar titik yang akan digambar. Di sini, kita akan menggambar titik dengan ukuran 5.0. glBegin(GL_POINTS) digunakan untuk menggambar titik. Posisi titik yang akan digambar harus ditulis antara glBegin(GL_POINTS) dan glEnd(). glColor3f(1.0f, 0.0f, 0.0f) adalah warna dari titik yang akan digambar. Format yang digunakan adalah format RGB, jadi warna titik adalah merah. glVertex3f(-10.0f, 5.0f, -20.0f) berarti posisi titik adalah 10 satuan ke kiri, 5 satuan ke atas, dan 20 satuan menjauh dari layar. Untuk koordinat, posisi tengah (posisi 0,0,0) berada tepat di tengah-tengah layar. Sumbu X negatif berarti ke kiri, positif berarti ke kanan. Sumbu Y positif berarti ke atas, negatif ke bawah. Sedangkan untuk sumbu Z, tanda positif berarti di depan layar (di luar layar), jadi tidak terlihat di layar. Jadi sumbu Z harus menggunakan tanda negatif agar objek dapat dilihat.

Membuat Garis

```
#include <gl/glut.h> void
Garis(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glLineWidth(2.0f);    glBegin(GL_LINES);

    glColor3f(1.0f, 1.0f, 1.0f);
    glVertex3f( 0.00, 0.20, 0.0);
    glVertex3f( 0.00, -0.20, 0.0);
    glEnd();    glFlush();
}

int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutInitWindowSize(640,480);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
    glutCreateWindow("Membuat Garis");
    glutDisplayFunc(Garis);    glClearColor(0.0f,
0.0f, 1.0f, 1.0f);    glutMainLoop();    return
0;
}
```

glLineWidth(2.0f) digunakan untuk menentukan lebar garis yang akan digambar. Di sini, kita akan menggambar garis dengan lebar 2. Garis bisa memiliki warna degradasi. Ingat bahwa garis terdiri dari dua titik. Bila titik pertama berwarna hijau dan titik kedua berwarna biru, warna garis akan berdegradasi dari hijau ke biru. Lihat bagian garis kedua untuk melihat warna degradasi.

Menggambar Segitiga

```
#include <gl/glut.h> void
Segitiga(void)
{
    glClear(GL_COLOR_BUFFER_BIT);    glBegin(GL_TRIANGLES);
    glColor3f(1.0f, 0.0f, 0.0f); glVertex3f( -0.10,-0.10, 0.00);
    glColor3f(0.0f, 1.0f, 0.0f); glVertex3f( 0.10,-0.10, 0.00);
    glColor3f(0.0f, 0.0f, 1.0f); glVertex3f( 0.00, 0.10, 0.00);
    glEnd();    glFlush();
}

int main(int argc, char*
argv[])
{
    glutInit(&argc, argv);
    glutInitWindowSize(640,480);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
    glutCreateWindow("Membuat Segitiga");    glutDisplayFunc(Segitiga);

    glClearColor(0.0f, 0.0f, 1.0f, 1.0f);
    glutMainLoop();    return 0;
}
```

Sekarang kita masuk ke bagian menggambar segitiga. Segitiga digambar dengan perintah `glBegin(GL_TRIANGLES)`. Perhatikan bahwa segitiga juga bisa memiliki warna degradasi.

Menggambar Segiempat Penuh

```
#include <gl/glut.h> void
SegiEmpat(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0f, 0.0f, 0.0f);
    glRectf(-0.18, 0.18, 0.18, -0.18);
    glFlush();
}

int main(int argc, char*
argv[])
{
    glutInit(&argc, argv);
    glutInitWindowSize(640,480);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutCreateWindow("Segi
Empat");
    glutDisplayFunc(SegiEmpat);
    glClearColor(0.0f, 0.0f,
1.0f, 1.0f);
    glutMainLoop();
    return 0;
}
```

Pertanyaan :

1. Apa fungsi `glColor3f(1.0f, 0.0f, 0.0f)`? dan apa fungsi f dibelakang valuenya ? jelaskan gabungan warna untuk menghasilkan warna hijau menggunakan fungsi `glColor3f`?
2. Apakah didalam openGL terdapat fungsi `glColor4f`)? Jika ada warna apa saja yang ada didalamnya?
3. Apa fungsi `glVertex3f(-0.10,-0.10, 0.00)`?
4. Apa fungsi `glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA)` ?

Tugas :

1. Selain membuat titik menggunakan `GL_POINT`, garis menggunakan `GL_LINES`,
2. buatlah program yang memanfaatkan fungsi `GL_LINE_STRIP`, `GL_LINE_LOOP`, `GL_TRIANGLE_FAN`, `GL_TRIANGLE_STRIP`, `GL_QUADS`, dan `GL_QUAD_STRIP`.
3. Buat Gambar Kubus bertingkat dari objek primitif sekreatif mungkin. (minimal 5 kubus)

Contoh :

