

# Tutorial 01. “Hello World” dengan GLUT

Tujuan dari bagian ini adalah untuk membuat OpenGL-based window. Ada banyak cara untuk membuat dan memperlihatkan suatu window dalam berbagai sistem window. Salah satunya adalah dengan menggunakan OpenGL Utility Toolkit. OpenGL Utility Toolkit (GLUT) menyediakan banyak fungsi yang dapat membuat window dengan cara yang independen terhadap sistem operasinya. Hal ini berarti program yang dibuat dengan GLUT dapat beroperasi pada sistem windowing yang berbeda tanpa merubah code secara manual.

GLUT adalah API (Application Programming Interface) dengan binding ANSI C untuk penulisan Sistem Windows program OpenGL. GLUT adalah buatan Mark J. Killgard, ketika bekerja di Silicon Graphics Inc. Walaupun dirancang untuk digunakan bersama-sama dengan kode OpenGL, GLUT dapat digunakan dengan atau tanpa OpenGL. Toolkit ini mendukung fungsionalitas sebagai berikut:

- Multiple windows untuk rendering OpenGL.
- Callback driven event processing.
- Sophisticated input devices.
- An “idle” routine and timers.
- A simple, cascading pop-up menu facility.
- Utility routines to generate various solid and wire frame objects.
- Support for bitmap and stroke fonts.
- Miscellaneous window management functions, including managing overlays.

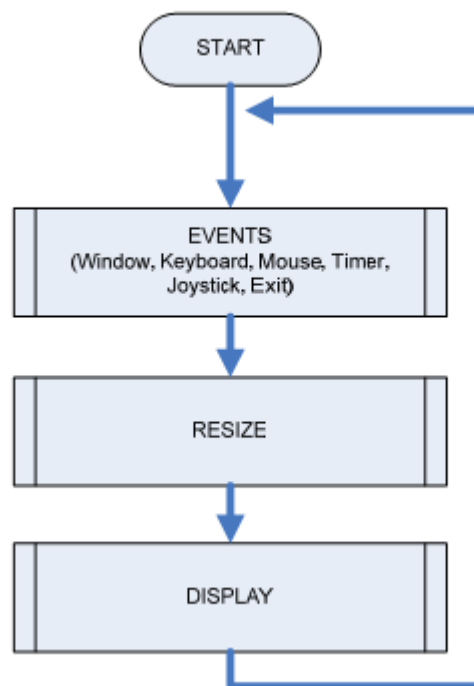
Walaupun secara fungsional mungkin jauh tertinggal dari multiplatform window system yang lain seperti Qt, namun kesederhanaan penggunaan serta hubungannya dengan API grafika komputer OpenGL membuat glut masih banyak digunakan terutama sebagai alat bantu pendidikan Grafika Komputer.

Struktur dari aplikasi berbasis GLUT akan terdiri atas beberapa langkah berikut, yaitu:

- Menetapkan konfigurasi windows, dan membuka windows
- Inisialisasi status OpenGL
- Registrasi callback functions (jika dibutuhkan)
  - o Render
  - o Resize
  - o Input
  - o Timer
  - o Idle
- Enter event processing loop.

Gambar 1 mengilustrasikan loop tersebut.

**Gambar 1. GLUT Event Processing Loop yang disederhanakan**



```
int main(int argc, char **argv) {

    // init GLUT and create window

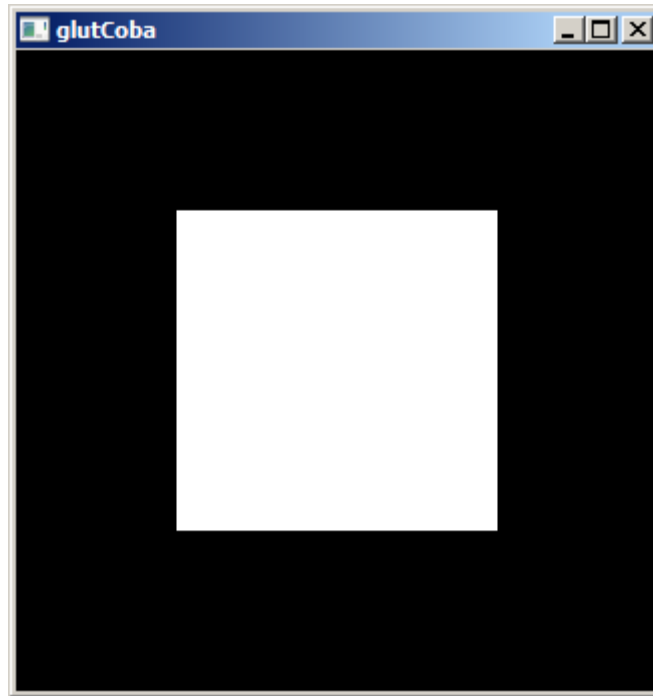
    // register callbacks

    // enter GLUT event processing cycle

}
```

```
1  #include <GL/glut.h>
2
3  void renderScene(void);
4
5  int main(int argc, char* argv[])
6  {
7      glutInit(&argc, argv);
8      glutInitDisplayMode(GLUT_DEPTH | GLUT_SINGLE | GLUT_RGBA);
9      glutCreateWindow("glutCoba");
10     glutDisplayFunc(renderScene);
11     glutMainLoop();
12
13     return 0;
14 }
15
16 void renderScene(void)
17 {
18     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
19     glBegin(GL_POLYGON);
20         glVertex2f(-0.5f, -0.5f);
21         glVertex2f(-0.5f, 0.5f);
22         glVertex2f(0.5f, 0.5f);
23         glVertex2f(0.5f, -0.5f);
24     glEnd();
25
26     glFlush();
27 }
28
```

renderScene() diregistrasi oleh glutDisplayFunc() sebagai fungsi yang dilaksanakan saat window digambar, yang biasanya adalah isi dari gambarnya. Hasilnya akan tampak seperti gambar berikut:



Coba periksa dimensi window seperti koordinat titik tengah, titik kiri atas, titik kiri bawah, titik kanan atas dan titik kanan bawah dengan merubah-rubah parameter fungsi glVertex2f().

Diperoleh Koordinat:

- Titik Tengah ( \_\_, \_\_)
- Titik Kiri Atas ( \_\_, \_\_), Titik Kiri Bawah ( \_\_, \_\_)
- Titik Kanan Atas ( \_\_, \_\_), Titik Kanan Bawah ( \_\_, \_\_)