

PRAKTIKUM 2
PENGENALAN TOOLS : BELAJAR MENGENAI ROTASI,
TRANSLASI, DAN SCALING.

1. TUJUAN PRAKTIKUM

a. Tujuan Umum

Praktikum ini bertujuan untuk memberikan pemahaman dan pengalaman praktis kepada peserta praktikum tentang konsep dasar rotasi, translasi, dan scaling dalam grafika komputer.

b. Tujuan Khusus

- Memahami konsep rotasi dan cara mengaplikasikannya dalam grafika komputer.
- Mengerti konsep translasi dan bagaimana melakukan perpindahan objek.
- Memahami konsep scaling dan cara mengubah ukuran objek tanpa mengubah bentuknya.
- Melakukan percobaan dan latihan.

2. PENGENALAN

• Rotasi:

Rotasi adalah salah satu jenis transformasi dalam grafika komputer yang mengubah orientasi objek dalam ruang tiga dimensi. Dalam rotasi, objek dapat diputar sekitar sumbu tertentu (misalnya, sumbu x, y, atau z) dengan sudut tertentu. Hal ini digunakan untuk menciptakan efek perputaran atau perubahan orientasi objek.

• Translasi:

Translasi adalah jenis transformasi lain yang mengacu pada perpindahan atau pergeseran objek dalam ruang tiga dimensi. Dalam translasi, objek digeser dari satu lokasi ke lokasi lain tanpa mengubah bentuk, orientasi, atau ukuran objek tersebut. Ini digunakan untuk mengatur posisi objek dalam tampilan.

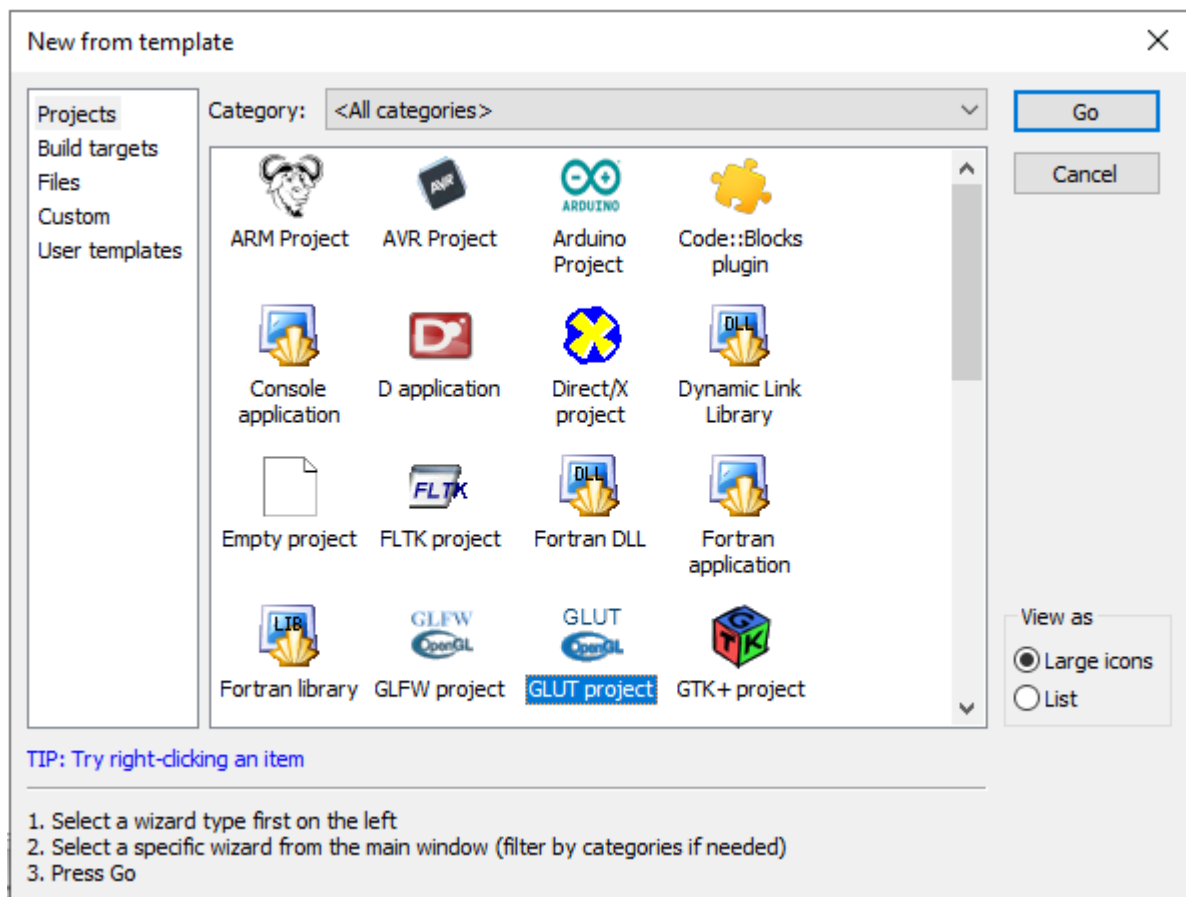
• Scaling :

Scaling adalah transformasi yang mengubah ukuran objek. Objek dapat diperbesar (scaling up) atau diperkecil (scaling down) sesuai dengan faktor penskalaan yang diterapkan. Scaling memungkinkan pengontrolan ukuran dan proporsi objek dalam tampilan grafis.

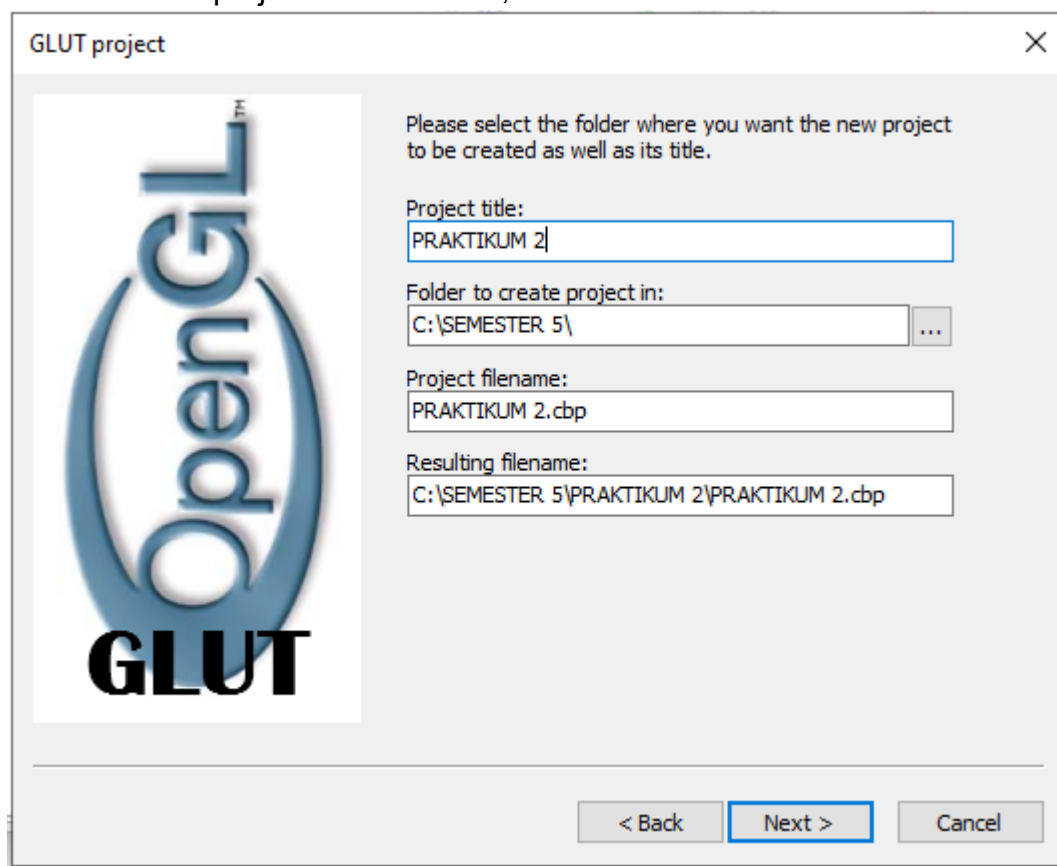
3. PELAKSANAAN PRAKTIKUM

Langkah – langkah praktikum :

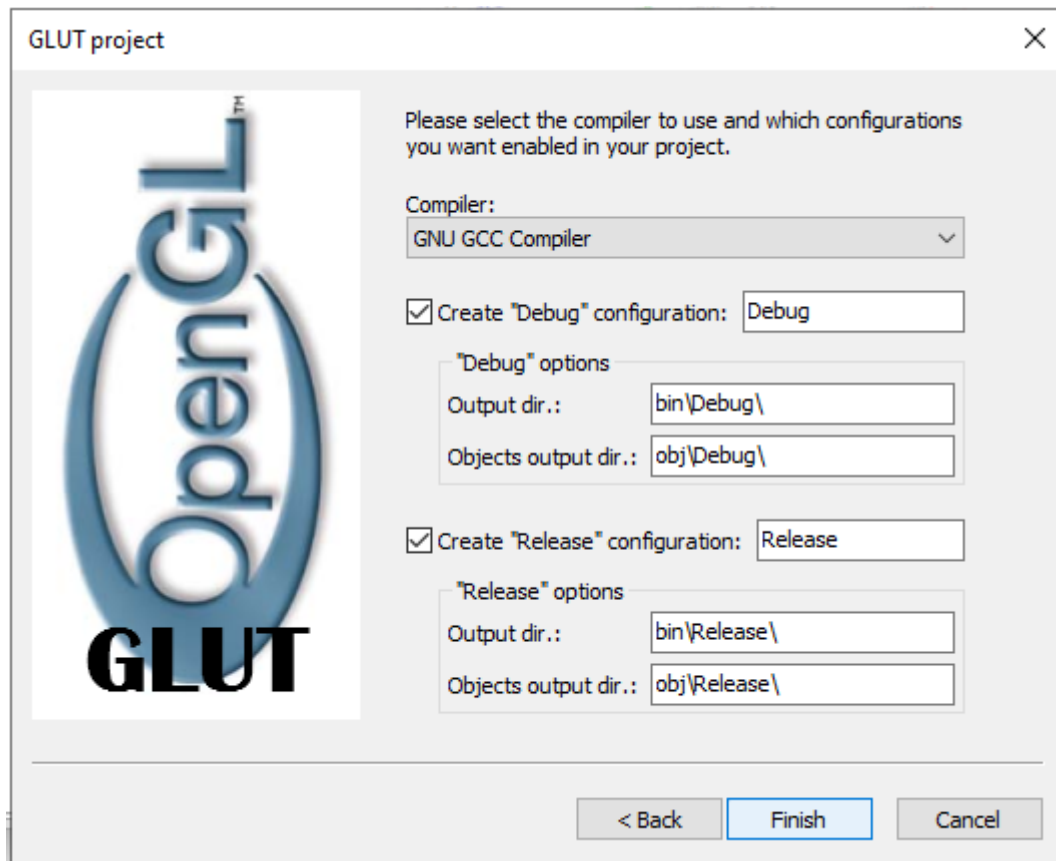
1. Buat Project GLUT dengan nama 'PRATIKUM 2'
Pilih kategori GLUT, lalu klik go



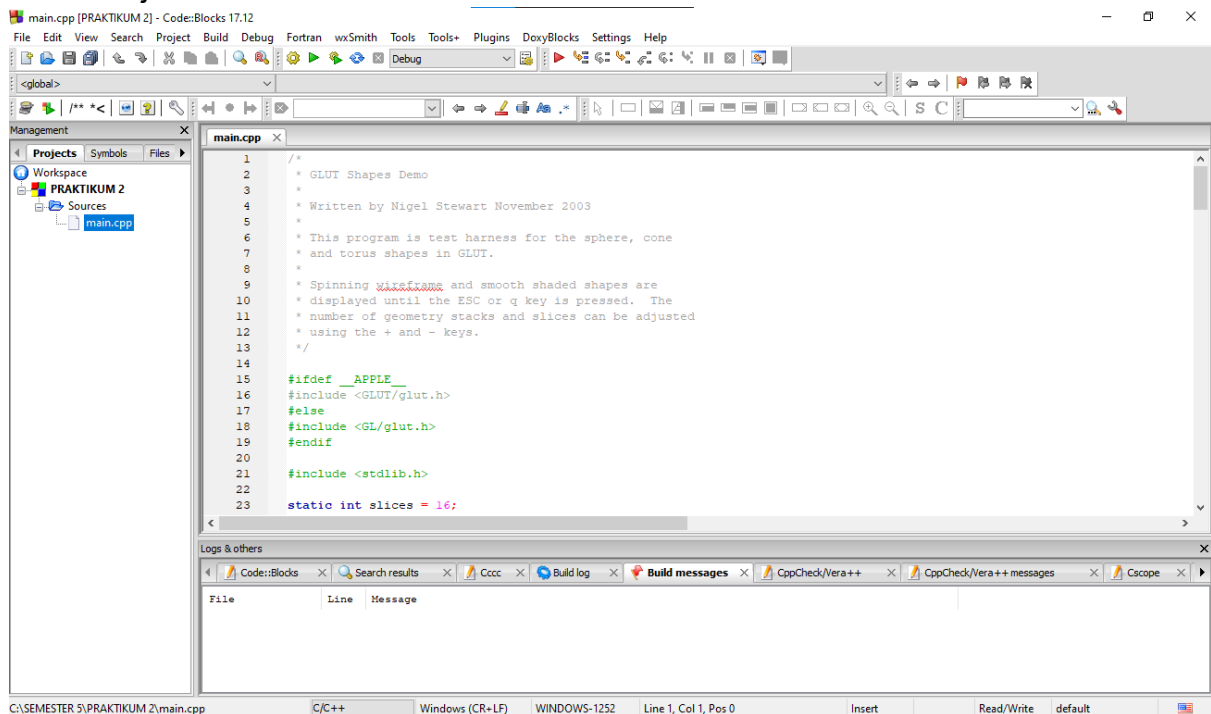
lalu beri nama project 'Praktikum 2', lalu klik next



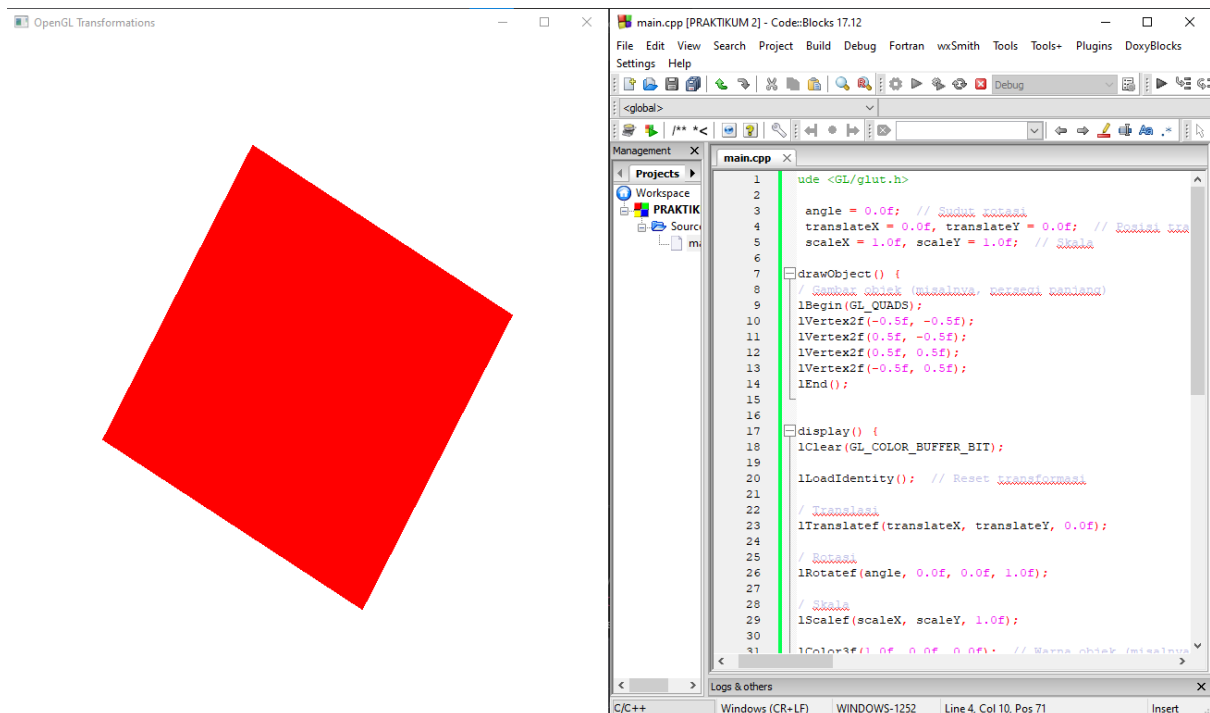
lalu klik klik finish



Maka Project sudah terbuat



2. Mencoba Code C++ Rotasi, Translasi dan Scaling
Berikut adalah output dari code nya :



Mari kita jelaskan secara detail setiap bagian dari kode tersebut:

1. Header dan Deklarasi Variabel:

```
#include <GL/glut.h>
```

```
float angle = 0.0f; // Sudut rotasi
float translateX = 0.0f, translateY = 0.0f; // Posisi translasi
float scaleX = 1.0f, scaleY = 1.0f; // Skala
```

- **#include <GL/glut.h>**: Memasukkan header file GLUT yang diperlukan untuk membuat aplikasi OpenGL menggunakan windowing system.
- **float angle, float translateX, float translateY, float scaleX, float scaleY**: Variabel-variabel yang digunakan untuk menyimpan nilai sudut rotasi, posisi translasi, dan skala.

2. Fungsi drawObject:

```
void drawObject() {
    glBegin(GL_QUADS);
    glVertex2f(-0.5f, -0.5f);
    glVertex2f(0.5f, -0.5f);
    glVertex2f(0.5f, 0.5f);
    glVertex2f(-0.5f, 0.5f);
    glEnd();
}
```

- Fungsi ini digunakan untuk menggambar objek. Dalam contoh ini, objek yang digambar adalah persegi panjang dengan sudut di $(-0.5, -0.5)$, $(0.5, -0.5)$, $(0.5, 0.5)$, dan $(-0.5, 0.5)$.

3. Fungsi display :

```
void display() {  
    glClear(GL_COLOR_BUFFER_BIT);  
  
    glLoadIdentity(); // Reset transformasi  
  
    // Translasi  
    glTranslatef(translateX, translateY, 0.0f);  
  
    // Rotasi  
    glRotatef(angle, 0.0f, 0.0f, 1.0f);  
  
    // Skala  
    glScalef(scaleX, scaleY, 1.0f);  
  
    glColor3f(1.0f, 0.0f, 0.0f); // Warna objek (misalnya, merah)  
    drawObject();  
  
    glutSwapBuffers();  
}
```

- **glClear(GL_COLOR_BUFFER_BIT)** : Membersihkan buffer layar.
- **glLoadIdentity()** : Mereset transformasi menjadi identitas.
- **glTranslatef(translateX, translateY, 0.0f)** : Mentranslasikan objek berdasarkan nilai `translateX` dan `translateY`.
- **glRotatef(angle, 0.0f, 0.0f, 1.0f)** : Merotasi objek sebesar `angle` derajat searah jarum jam di sekitar sumbu z.
- **glScalef(scaleX, scaleY, 1.0f)** : Membesarkan atau mengecilkan objek berdasarkan nilai `scaleX` dan `scaleY`.
- **glColor3f(1.0f, 0.0f, 0.0f)** : Mengatur warna objek menjadi merah.
- **drawObject()** : Memanggil fungsi untuk menggambar objek.
- **glutSwapBuffers()** : Menukar buffer dan menampilkan hasilnya di layar.

4. Fungsi update :

```
void update(int value) {  
    angle += 2.0f;  
  
    if (angle > 360) {  
        angle -= 360;  
    }  
  
    glutPostRedisplay(); // Meminta untuk me-redraw window  
    glutTimerFunc(16, update, 0); // Mengatur timer untuk update setiap 16ms  
}
```

- Fungsi ini dipanggil secara berulang menggunakan `glutTimerFunc` untuk membuat animasi rotasi objek.

- `angle += 2.0f` : Menambahkan sudut rotasi sebesar 2 derajat pada setiap pemanggilan fungsi.
- Jika sudut rotasi melebihi 360 derajat, maka direset ke 0 derajat.
- `glutPostRedisplay()` : Meminta agar fungsi `display` dipanggil kembali untuk me-redraw window.

5. Fungsi main :

```
int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutCreateWindow("OpenGL Transformations");

    glOrtho(-1, 1, -1, 1, -1, 1); // Set window coordinates

    glutDisplayFunc(display);
    glutTimerFunc(25, update, 0); // Timer untuk update setiap 25ms

    glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Warna background putih

    glutMainLoop();

    return 0;
}
```

- Inisialisasi GLUT dan membuat window dengan judul "OpenGL Transformations".
- `glOrtho(-1, 1, -1, 1, -1, 1)` : Mengatur koordinat window.
- `glutDisplayFunc(display)` : Mengatur fungsi `display` sebagai fungsi untuk menampilkan konten di window.
- `glutTimerFunc(25, update, 0)` : Mengatur timer untuk memanggil fungsi `update` setiap 25ms.
 - `glClearColor(1.0f, 1.0f, 1.0f, 1.0f)` : Mengatur warna background window menjadi putih.
 - `glutMainLoop()` : Memulai event loop GLUT untuk menangani event dan menjalankan program secara terus-menerus.