

PRAKTIKUM 4

MENGENAL PEMETAAN TEKSTUR DAN TEKSTUR TRANSPARAN

1. TUJUAN PRAKTIKUM

a. Tujuan Umum

Praktikum ini bertujuan untuk memberikan pemahaman dan pengalaman praktis kepada peserta praktikum tentang pemetaan tekstur dan penggunaan tekstur transparan dalam grafika komputer.

b. Tujuan Khusus

- Memahami konsep pemetaan tekstur dalam grafika komputer.
- Mengerti cara mengimplementasikan tekstur pada objek yang akan dirotasi, diterjemahkan, dan di-scaled.
- Mengetahui konsep dan implementasi tekstur transparan untuk menciptakan efek visual yang menarik.

2. PENGENALAN

- Pemetaan Tekstur:

Pemetaan tekstur adalah proses menempatkan gambar (tekstur) pada permukaan objek dalam grafika komputer. Pada program ini, kita akan mempelajari cara mengaplikasikan pemetaan tekstur pada sebuah kubus yang akan mengalami transformasi.

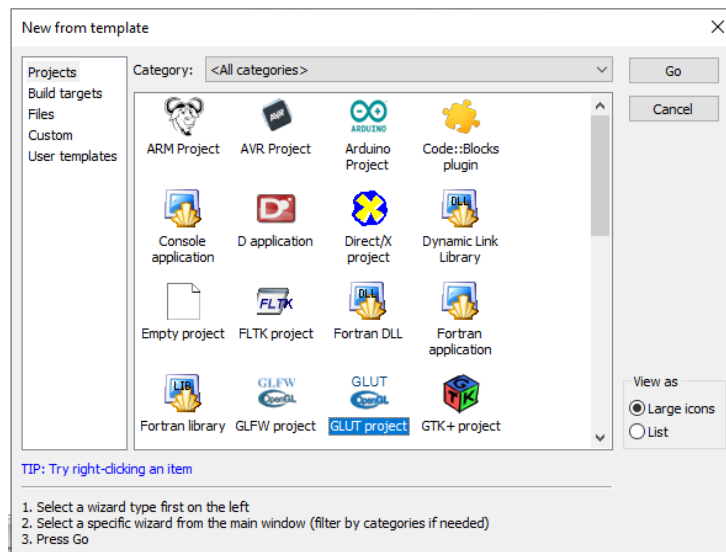
- Tekstur Transparan:

Tekstur transparan adalah penggunaan gambar dengan tingkat transparansi sehingga objek yang menggunakan tekstur tersebut dapat terlihat sebagian atau sepenuhnya transparan. Pada program ini, kita akan menciptakan efek transparansi pada beberapa bagian dari kubus untuk menciptakan tampilan yang menarik.


3. PELAKSANAAN PRAKTIKUM

Langkah – langkah praktikum :

1. Buat Project GLUT baru



GLUT project



Please select the folder where you want the new project to be created as well as its title.

Project title:


Folder to create project in:
 ...

Project filename:

Resulting filename:

< Back Next > Cancel

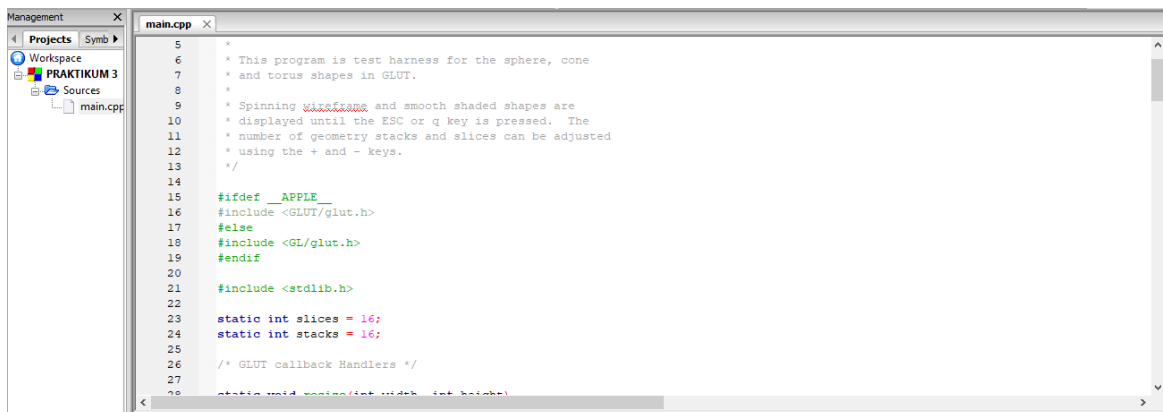
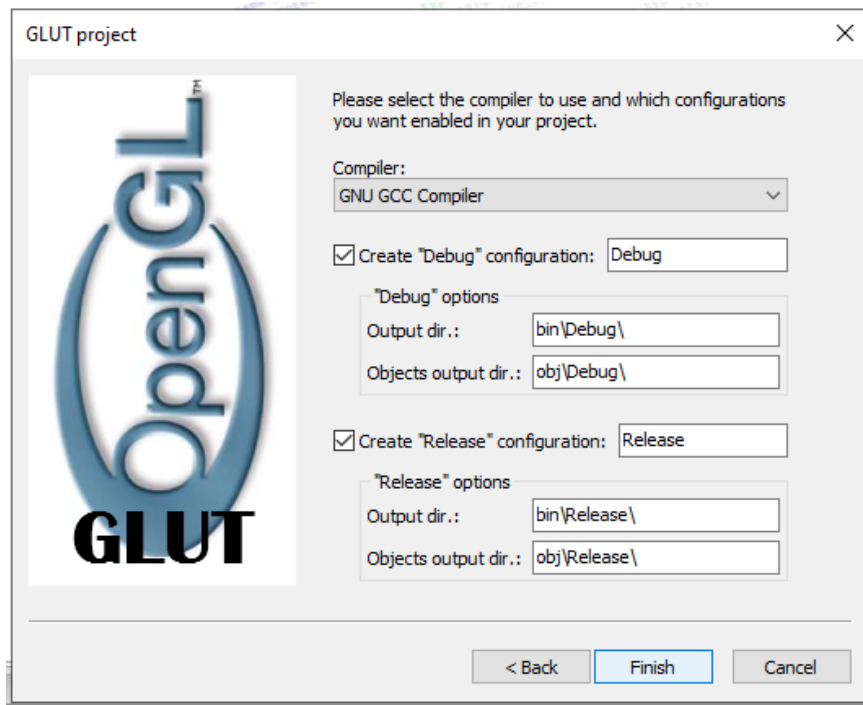
GLUT project



Please select the location of GLUT on your computer.
This is the top-level folder where GLUT was installed (unpacked).
To help you, this folder must contain the subfolders "include" and "lib".

Please select GLUT's location:
 ...

< Back Next > Cancel



2. Edit code pada Main.cpp
 - a. Inisialisasi Library OpenGL dan GLUT.

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

```
GLfloat angle = 0.0; // Sudut rotasi kubus
```

```
void init() {
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_BLEND);
    glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
}
```

Penjelasan:

- Mendeklarasikan variabel angle untuk menyimpan sudut rotasi.
- Menginisialisasi lingkungan OpenGL dengan memberikan warna latar belakang dan mengaktifkan tes kedalaman serta blending untuk tekstur transparan.

b. Fungsi Menggambar Kubus dengan Pemetaan Tekstur dan Transparansi

```
void drawCube() {  
    // Depan  
    glBegin(GL_QUADS);  
    glColor4f(1.0, 0.0, 0.0, 0.5); // Merah, transparan  
    glVertex3f(-0.5, -0.5, 0.5);  
    glVertex3f(0.5, -0.5, 0.5);  
    glVertex3f(0.5, 0.5, 0.5);  
    glVertex3f(-0.5, 0.5, 0.5);  
    glEnd();
```

```
    // Belakang  
    glBegin(GL_QUADS);  
    glColor3f(0.0, 1.0, 0.0); // Hijau  
    glVertex3f(-0.5, -0.5, -0.5);  
    glVertex3f(0.5, -0.5, -0.5);  
    glVertex3f(0.5, 0.5, -0.5);  
    glVertex3f(-0.5, 0.5, -0.5);  
    glEnd();
```

```
    // Sisi Kiri  
    glBegin(GL_QUADS);  
    glColor3f(0.0, 0.0, 1.0); // Biru  
    glVertex3f(-0.5, -0.5, 0.5);  
    glVertex3f(-0.5, 0.5, 0.5);  
    glVertex3f(-0.5, 0.5, -0.5);  
    glVertex3f(-0.5, -0.5, -0.5);  
    glEnd();
```

```
    // Sisi Kanan  
    glBegin(GL_QUADS);  
    glColor3f(1.0, 1.0, 0.0); // Kuning  
    glVertex3f(0.5, -0.5, 0.5);  
    glVertex3f(0.5, 0.5, 0.5);  
    glVertex3f(0.5, 0.5, -0.5);  
    glVertex3f(0.5, -0.5, -0.5);  
    glEnd();
```

```
    // Atas  
    glBegin(GL_QUADS);  
    glColor3f(0.0, 1.0, 1.0); // Cyan  
    glVertex3f(-0.5, 0.5, 0.5);  
    glVertex3f(0.5, 0.5, 0.5);  
    glVertex3f(0.5, 0.5, -0.5);  
    glVertex3f(-0.5, 0.5, -0.5);  
    glEnd();
```

```
    // Bawah
```

```

glBegin(GL_QUADS);
glColor3f(1.0, 0.0, 1.0); // Magenta
glVertex3f(-0.5, -0.5, 0.5);
glVertex3f(0.5, -0.5, 0.5);
glVertex3f(0.5, -0.5, -0.5);
glVertex3f(-0.5, -0.5, -0.5);
glEnd();
}

```

Penjelasan:

- Membuat fungsi drawCube() untuk menggambar kubus dengan enam sisi berbeda, masing-masing diisi dengan warna dan tekstur transparan.

c. Fungsi Tampilan Utama

```

void display() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();

    glTranslatef(0.0, 0.0, -5.0);
    glRotatef(angle, 1.0, 1.0, 0.0);

    drawCube();

    glutSwapBuffers();
}

```

Penjelasan:

- Membuat fungsi display() untuk menampilkan kubus dengan pemetaan tekstur dan transparansi.

d. Fungsi Reshape

```

void reshape(int width, int height) {
    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45.0, (double)width / (double)height, 1.0, 100.0);
    glMatrixMode(GL_MODELVIEW);
}

```

Penjelasan:

- Membuat fungsi reshape() untuk menanggapi perubahan ukuran jendela dan melakukan penyesuaian viewport.

e. Fungsi Timer

```

void timer(int value) {
    angle += 2.0;
}

```

```

    if (angle > 360.0) {
        angle -= 360.0;
    }
    glutPostRedisplay();
    glutTimerFunc(16, timer, 0);
}

```

Penjelasan:

- Membuat fungsi timer() untuk mengatur rotasi otomatis kubus.

f. Fungsi Utama dan Inisialisasi GLUT

```

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

    init();

    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutTimerFunc(25, timer, 0);

    glutMainLoop();

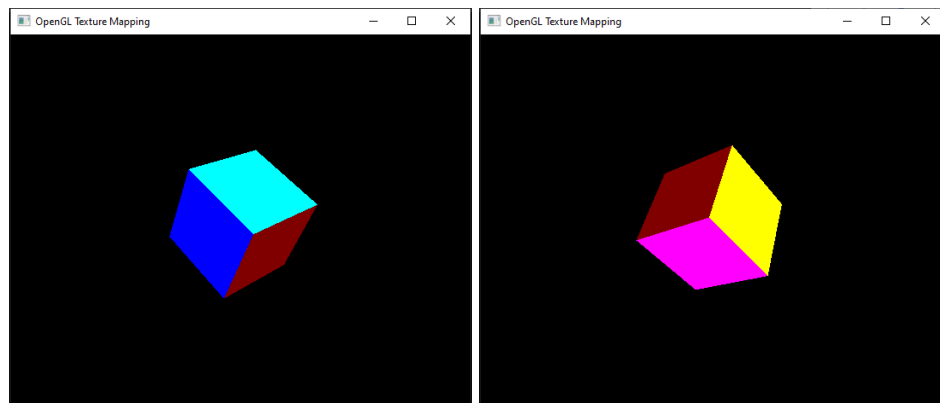
    return 0;
}

```

Penjelasan:

- Membuat fungsi main() sebagai fungsi utama yang menjalankan program.
- Inisialisasi GLUT dan menetapkan fungsi-fungsi callback.

g. OUTPUT



4. Latihan

Gantilah warna dan tingkat transparansi pada sisi depan kubus agar sesuai dengan Warna Putih.