

## PENJELASAN KODE “lengan\_bergerak.cpp”

### lengan.bergerak.cpp

```
/* Nama File      : lengan_bergerak.cpp */
/* Deskripsi      : Membuat animasi pada objek representasi lengan dengan
modifikasi telapak tangan
                    dan jari-jari. */
/* Nama          : Aprilyanto Setiyawan Siburian */
/* NIM           : 24060121120022 */

#include <GL/glut.h>
static int shoulder = 0, elbow = 0, palm = 0,
thumbf = 0, indexf = 0, middlef = 0, ringf = 0, littlef = 0;

void init(void)
{
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glShadeModel (GL_FLAT);
}

void display(void)
{
    glClear (GL_COLOR_BUFFER_BIT);
    glPushMatrix();
    // shoulder = bahu
    glTranslatef (-2.0, 0.0, 0.0);
    glRotatef ((GLfloat) shoulder, 0.0, 0.0, 1.0);
    glTranslatef (1.0, 0.0, 0.0);
    glPushMatrix();
        glScalef (2.0, 0.4, 1.0);
        glutWireCube (1.0);
    glPopMatrix();

    // elbow = siku
    glTranslatef (1.0, 0.0, 0.0);
    glRotatef ((GLfloat) elbow, 0.0, 0.0, 1.0);
    glTranslatef (1.0, 0.0, 0.0);
    glPushMatrix();
        glScalef (2.0, 0.4, 1.0);
        glutWireCube (1.0);
    glPopMatrix();

    // palm = telapak tangan
    glTranslatef (1.0, 0.0, 0.0);
    glRotatef((GLfloat) palm, 0.0, 0.0, 1.0);
    glTranslatef (0.5, 0.0, 0.0);
    glPushMatrix();
        glScalef (1.0, 0.4, 2.0);
        glutWireCube (1.0);
    glPopMatrix();

    // thumbf = ibu jari
```

```

glPushMatrix();
    glTranslatef (0.0, 0.0, 1.1);
    glRotatef((GLfloat) thumbf, -1.0, 0.0, 0.0);
    glTranslatef (0.0, 0.0, 0.3);
    glPushMatrix();
        glScalef (0.5, 0.4, 0.8);
        glutWireCube (1.0);
    glPopMatrix();
glPopMatrix();

// indexf = jari telunjuk
glPushMatrix();
    glTranslatef (0.75, 0.0, -0.2);
    glRotatef((GLfloat) indexf, 0.0, 0.0, 1.0);
    glTranslatef (0.5, 0.0, -0.6);
    glPushMatrix();
        glScalef (1.5, 0.4, 0.3);
        glutWireCube (1.0);
    glPopMatrix();
glPopMatrix();

// middlef = jari tengah
glPushMatrix();
    glTranslatef (0.5, 0.0, -0.2);
    glRotatef((GLfloat) middlef, 0.0, 0.0, 1.0);
    glTranslatef (1.0, 0.0, -0.2);
    glPushMatrix();
        glScalef (2.0, 0.4, 0.3);
        glutWireCube (1.0);
    glPopMatrix();
glPopMatrix();

// ringf = jari manis
glPushMatrix();
    glTranslatef (0.75, 0.0, 0.2);
    glRotatef((GLfloat) ringf, 0.0, 0.0, 1.0);
    glTranslatef (0.5, 0.0, 0.2);
    glPushMatrix();
        glScalef (1.5, 0.4, 0.3);
        glutWireCube (1.0);
    glPopMatrix();
glPopMatrix();

// littlef = jari kelingking
glPushMatrix();
    glTranslatef (0.5, 0.0, 0.5);
    glRotatef((GLfloat) littlef, 0.0, 0.0, 1.0);
    glTranslatef (0.5, 0.0, 0.5);
    glPushMatrix();
        glScalef (1.0, 0.4, 0.3);
        glutWireCube (1.0);
    glPopMatrix();
glPopMatrix();

```

```

    glPopMatrix();
    glutSwapBuffers();
}

void reshape (int w, int h)
{
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(65.0, (GLfloat) w/(GLfloat) h, 1.0, 20.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    glTranslatef (0.0, 0.0, -5.0);
}

void keyboard(unsigned char key, int x, int y)
{
    switch (key)
    {
        case 's': shoulder = (shoulder + 5) % 360;
            glutPostRedisplay();
            break;
        case 'S': shoulder = (shoulder - 5) % 360;
            glutPostRedisplay();
            break;
        case 'e': elbow = (elbow + 5) % 360;
            glutPostRedisplay();
            break;
        case 'E': elbow = (elbow - 5) % 360;
            glutPostRedisplay();
            break;
        case 'p': palm = (palm + 5) % 360;
            glutPostRedisplay();
            break;
        case 'P': palm = (palm - 5) % 360;
            glutPostRedisplay();
            break;
        case 't': thumbf = (thumbf + 5) % 360;
            glutPostRedisplay();
            break;
        case 'T': thumbf = (thumbf - 5) % 360;
            glutPostRedisplay();
            break;
        case 'i': indexf = (indexf + 5) % 360;
            glutPostRedisplay();
            break;
        case 'I': indexf = (indexf - 5) % 360;
            glutPostRedisplay();
            break;
        case 'm': middlef = (middlef + 5) % 360;
            glutPostRedisplay();
            break;
    }
}

```

```

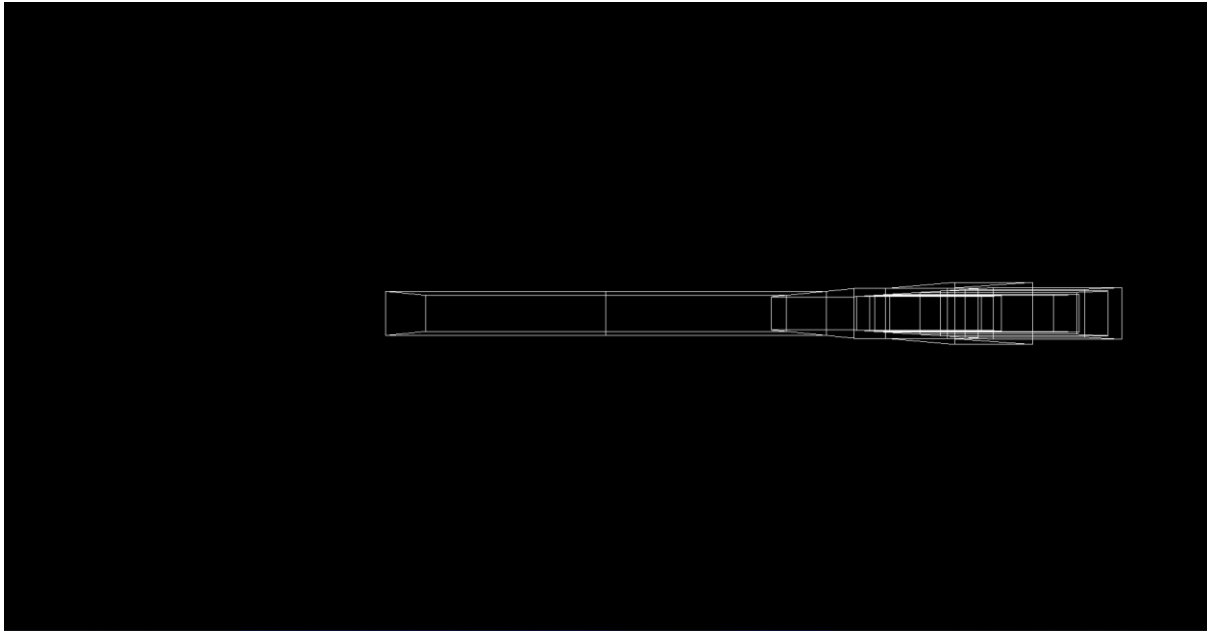
        case 'M': middlef = (middlef - 5) % 360;
                    glutPostRedisplay();
                    break;
        case 'r': ringf = (ringf + 5) % 360;
                    glutPostRedisplay();
                    break;
        case 'R': ringf = (ringf - 5) % 360;
                    glutPostRedisplay();
                    break;
        case 'l': littlef = (littlef + 5) % 360;
                    glutPostRedisplay();
                    break;
        case 'L': littlef = (littlef - 5) % 360;
                    glutPostRedisplay();
                    break;
        case 27: exit(0);
                    break;
        default:
                    break;
    }
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize (700, 600);
    glutInitWindowPosition (100, 100);
    glutCreateWindow (argv[0]);
    init();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutMainLoop();

    return 0;
}

```

## Hasil kompilasi



Kode “*lengan\_bergerak.cpp*” menggabungkan antara proses translasi dan rotasi dengan objek 3D dengan memanfaatkan sumbu z sehingga objek dapat berbentuk 3D. Selain memanfaatkan sumbu z, benda juga digerakkan melalui suatu proses / perhitungan tertentu yang diterapkan pada objek oleh perintah keyboard, menghasilkan animasi benda berupa representasi lengan yang digerakkan oleh keyboard.

Representasi lengan oleh objek yang terdapat pada kode ini adalah *shoulder* (bahu), *elbow* (siku), *palm* (telapak tangan), *thumbf* (ibu jari), *indexf* (jari telunjuk), *middle* (jari tengah), *ringf* (jari manis), *littlef* (jari kelingking). Masing-masing bagian tersebut dapat digerakkan dengan keyboard. Kata kunci keyboard yang digunakan pada masing-masing bagian tersebut adalah huruf depan dari nama bagian. Contoh: key ‘s’ dan ‘S’ untuk *shoulder*, key ‘e’ dan ‘E’ untuk *elbow*. Setiap perubahan pada objek akibat kata kunci keyboard bernilai 5°. Semua objek dinaikkan/diturunkan berdasarkan sumbu z, kecuali *thumbf* yang dinaikkan/diturunkan berdasarkan sumbu x. Penggunaan key secara lengkap tertera pada tabel berikut.

Keyboard	Fungsi
‘s’	Menaikkan <i>shoulder</i> berdasarkan sumbu z sebanyak 5°
‘S’	Menurunkan <i>shoulder</i> berdasarkan sumbu z sebanyak 5°
‘e’	Menaikkan <i>elbow</i> berdasarkan sumbu z sebanyak 5°
‘E’	Menurunkan <i>elbow</i> berdasarkan sumbu z sebanyak 5°
‘t’	Menaikkan <i>thumbf</i> berdasarkan sumbu x sebanyak 5°
‘T’	Menurunkan <i>thumbf</i> berdasarkan sumbu x sebanyak 5°
‘i’	Menaikkan <i>indexf</i> berdasarkan sumbu z sebanyak 5°
‘I’	Menurunkan <i>indexf</i> berdasarkan sumbu z sebanyak 5°

‘m’	Menaikkan <i>middlef</i> berdasarkan sumbu z sebanyak $5^\circ$
‘M’	Menurunkan <i>middlef</i> berdasarkan sumbu z sebanyak $5^\circ$
‘r’	Menaikkan <i>ringf</i> berdasarkan sumbu z sebanyak $5^\circ$
‘R’	Menurunkan <i>ringf</i> berdasarkan sumbu z sebanyak $5^\circ$
‘l’	Menaikkan <i>litlef</i> berdasarkan sumbu z sebanyak $5^\circ$
‘L’	Menurunkan <i>litlef</i> berdasarkan sumbu z sebanyak $5^\circ$
‘esc’	Keluar