Nama : Angel Mentari Bulawan P
NIM : 24060123140200
Kelas : A
Lab : A2

Pada praktikum ini, dilakukan simulasi pergerakan lengan manusia menggunakan OpenGL dengan GLUT. Simulasi ini mencakup pergerakan bahu, siku, pergelangan tangan, dan jari-jari tangan. Program ini memanfaatkan transformasi matriks, seperti translasi dan rotasi, untuk menggerakkan bagian-bagian lengan.

PENJELASAN PROGRAM

Program ini menggunakan GLUT untuk menggambar dan mengontrol pergerakan bagian lengan. Beberapa bagian utama dalam program ini adalah:

- **Fungsi *init()***: Mengatur warna latar belakang dan model shading.
- **Fungsi *display()***: Menggambar lengan dengan setiap bagian seperti bahu, siku, pergelangan tangan, dan jari-jari tangan.
- **Fungsi *reshape()***: Mengatur viewport dan proyeksi perspektif.
- **Fungsi *keyboard()***: Mengontrol pergerakan masing-masing bagian lengan dan jari dengan input dari keyboard.
- **Fungsi *main()***: Inisialisasi GLUT dan menjalankan loop utama program.

Kontrol Keyboard

- **'s' / 'S'**: Menggerakkan bahu
- **'e' / 'E'**: Menggerakkan siku
- **'w' / 'W'**: Menggerakkan pergelangan tangan
- **'1' - '5'**: Menggerakkan masing-masing jari
- **Esc**: Keluar dari program

SOURCE CODE

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

static int shoulder = 0, elbow = 0, wrist = 0, finger1 = 0, finger2 = 0, finger3 = 0, finger4 = 0, finger5 = 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
    glTranslatef(-1.0, 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
    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();

    // Wrist
    glTranslatef(1.0, 0.0, 0.0);
    glRotatef((GLfloat)wrist, 0.0, 0.0, 1.0);
    glTranslatef(0.5, 0.0, 0.0);
    glPushMatrix();
    glScalef(1.0, 0.3, 0.7);
    glutWireCube(1.0);
    glPopMatrix();

    // Fingers
    for (int i = 0; i < 5; i++) {
        glPushMatrix();
        glTranslatef(0.5, (i - 2) * 0.2, 0.0);
        glRotatef((GLfloat)(i == 0 ? finger1 : i == 1 ? finger2 : i == 2 ? finger3 : i == 3 ? finger4 : finger5), 0.0, 0.0, 1.0);
        glTranslatef(0.3, 0.0, 0.0);
        glPushMatrix();
        glScalef(0.6, 0.1, 0.2);
        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; break;
```

```
    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; break;
        case 'S': shoulder = (shoulder - 5) % 360; break;
        case 'e': elbow = (elbow + 5) % 360; break;
        case 'E': elbow = (elbow - 5) % 360; break;
        case 'w': wrist = (wrist + 5) % 360; break;
        case 'W': wrist = (wrist - 5) % 360; break;
        case '1': finger1 = (finger1 + 5) % 360; break;
        case '2': finger2 = (finger2 + 5) % 360; break;
        case '3': finger3 = (finger3 + 5) % 360; break;
        case '4': finger4 = (finger4 + 5) % 360; break;
        case '5': finger5 = (finger5 + 5) % 360; break;
        case 27: exit(0); break;
    }
    glutPostRedisplay();
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(700, 600);
    glutCreateWindow("Lengan dengan Jari");
    init();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}
```

PENJELASAN DETAIL KODE

1. Inisialisasi **GLUT** dan **OpenGL**

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

Fungsi ini mengatur latar belakang tampilan menjadi hitam *(glClearColor)* dan menggunakan model shading datar *(glShadeModel(GL_FLAT))*.

2. **Fungsi *display()*** - Menggambar Lengan

```
void display(void) {
    glClear(GL_COLOR_BUFFER_BIT);
    glPushMatrix();
```

Fungsi ini membersihkan layar dengan *glClear()*, lalu mulai menggambar objek menggunakan *glPushMatrix()*.

Menampilkan bahu:

```
14      // Shoulder
15      glTranslatef(-1.0, 0.0, 0.0);
16      glRotatef((GLfloat)shoulder, 0.0, 0.0, 1.0);
17      glTranslatef(1.0, 0.0, 0.0);
18      glPushMatrix();
19      glScalef(2.0, 0.4, 1.0);
20      glutWireCube(1.0);
21      glPopMatrix();
```

Bagian ini menerapkan transformasi untuk bahu dengan translasi *(glTranslatef)* dan rotasi *(glRotatef)*. Kubus digambar menggunakan *glutWireCube* sebagai representasi lengan atas.

Menampilkan siku dan pergelangan tangan:

```
23      // Elbow
24      glTranslatef(1.0, 0.0, 0.0);
25      glRotatef((GLfloat)elbow, 0.0, 0.0, 1.0);
26      glTranslatef(1.0, 0.0, 0.0);
27      glPushMatrix();
28      glScalef(2.0, 0.4, 1.0);
29      glutWireCube(1.0);
30      glPopMatrix();
31
32      // Wrist
33      glTranslatef(1.0, 0.0, 0.0);
34      glRotatef((GLfloat)wrist, 0.0, 0.0, 1.0);
35      glTranslatef(0.5, 0.0, 0.0);
36      glPushMatrix();
37      glScalef(1.0, 0.3, 0.7);
38      glutWireCube(1.0);
39      glPopMatrix();
```

Sama seperti bahu, bagian siku dan pergelangan tangan digambarkan dengan transformasi dan rotasi.

Menampilkan jari-jari:

```
41      // Fingers
42      for (int i = 0; i < 5; i++) {
43          glPushMatrix();
44          glTranslatef(0.5, (i - 2) * 0.2, 0.0);
45          glRotatef((GLfloat)(i == 0 ? finger1 : i == 1 ? finger2 : i == 2 ? finger3 : i == 3 ? finger4 : finger5), 0.0, 0.0, 1.0);
46          glTranslatef(0.3, 0.0, 0.0);
47          glPushMatrix();
48          glScalef(0.6, 0.1, 0.2);
49          glutWireCube(1.0);
50          glPopMatrix();
51          glPopMatrix();
52      }
```

Jari-jari dibuat dalam loop dengan posisi yang berbeda dan memiliki rotasi terpisah untuk masing-masing jari.

3. **Fungsi *reshape()*** - Mengatur Viewport

```
58  void reshape(int w, int h) {
59      glViewport(0, 0, (GLsizei)w, (GLsizei)h);
60      glMatrixMode(GL_PROJECTION);
61      glLoadIdentity();
62      gluPerspective(65.0, (GLfloat)w / (GLfloat)h, 1.0, 20.0);
63      glMatrixMode(GL_MODELVIEW);
64      glLoadIdentity();
65      glTranslatef(0.0, 0.0, -5.0);
66  }
```

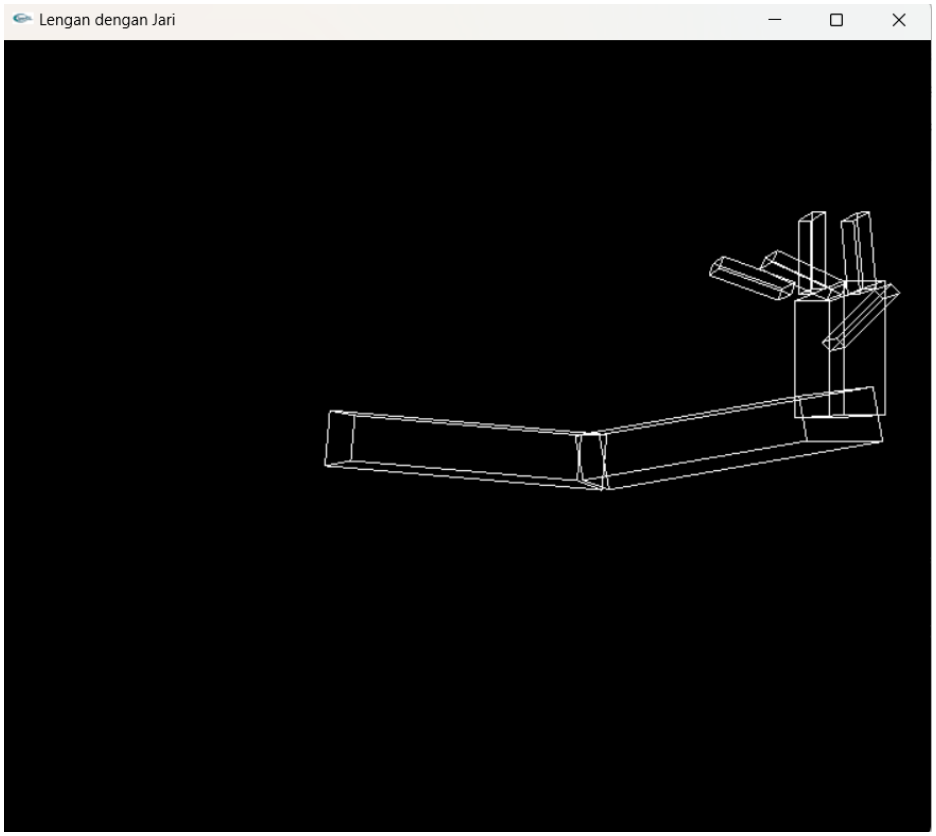
Fungsi ini digunakan untuk mengatur tampilan ketika jendela diubah ukurannya.

4. **Fungsi *keyboard()*** - Interaksi dengan Pengguna

```
68  void keyboard(unsigned char key, int x, int y) {
69      switch (key) {
70          case 's': shoulder = (shoulder + 5) % 360; break;
71          case 'S': shoulder = (shoulder - 5) % 360; break;
72          case 'e': elbow = (elbow + 5) % 360; break;
73          case 'E': elbow = (elbow - 5) % 360; break;
74          case 'w': wrist = (wrist + 5) % 360; break;
75          case 'W': wrist = (wrist - 5) % 360; break;
76          case '1': finger1 = (finger1 + 5) % 360; break;
77          case '2': finger2 = (finger2 + 5) % 360; break;
78          case '3': finger3 = (finger3 + 5) % 360; break;
79          case '4': finger4 = (finger4 + 5) % 360; break;
80          case '5': finger5 = (finger5 + 5) % 360; break;
81          case 27: exit(0); break;
82      }
83      glutPostRedisplay();
84  }
```

Fungsi ini memungkinkan pengguna mengontrol pergerakan lengan dengan menekan tombol di keyboard.

HASIL IMPLEMENTASI



KESIMPULAN

Dari hasil praktikum, dapat disimpulkan bahwa dengan menggunakan OpenGL dan transformasi matriks, kita dapat membuat simulasi pergerakan objek secara dinamis. Simulasi ini memungkinkan kita untuk memahami dasar-dasar animasi dan interaksi di dalam pemrograman grafis 3D.