

Soal:

1. Jelaskan Bagaimana cara kerja Kode Lengan di atas!
2. Tambahkan telapak tangan beserta jari-jari dari lengan di atas. Telapak tangan dan jari-jari tersebut bisa digerakan menggunakan keyboard!
3. Simulasikan menggunakan sumbu x,y,z bagaimana operasi yang dilakukan ketika keyboard di ketik (dalam bentuk screenshot dan milimeter blok).

Jawaban:

1. a. Hierarki Transformasi (`glPushMatrix/glPopMatrix`): Kode menggunakan fungsi `glPushMatrix()` dan `glPopMatrix()` untuk menyimpan dan mengembalikan state matriks transformasi. Hal ini memungkinkan kita menerapkan transformasi secara hierarkis, sehingga rotasi pada sendi bahu mempengaruhi seluruh bagian lengan yang “terikat” padanya, sedangkan rotasi pada siku hanya mempengaruhi bagian bawah lengan.  
  
b. Transformasi Sendi Bahu: Pertama, sistem koordinat digeser menggunakan `glTranslatef(-1.0, 0.0, 0.0)` untuk menentukan titik awal. Kemudian, dilakukan rotasi menggunakan `glRotatef(shoulder, 0.0, 0.0, 1.0)`, yang mengatur pergerakan lengan bagian atas (upper arm) sesuai nilai variabel `shoulder`. Selanjutnya, dengan `glTranslatef(1.0, 0.0, 0.0)`, posisi dipindahkan ke titik di mana kubus yang menggambarkan upper arm akan digambar. Kubus tersebut digambar dengan `glutWireCube()` setelah skala diubah dengan `glScalef(2.0, 0.4, 1.0)` sehingga membentuk batang yang mewakili upper arm.  
  
c. Transformasi Siku dan Forearm Setelah menggambar upper arm, terjadi translasi lagi dengan `glTranslatef(1.0, 0.0, 0.0)` untuk mencapai titik sendi siku. Dilanjutkan dengan rotasi `glRotatef(elbow, 0.0, 0.0, 1.0)` yang mengubah orientasi forearm sesuai nilai variabel `elbow`. Kemudian, dengan `glTranslatef(1.0, 0.0, 0.0)` lagi, posisi dialihkan ke lokasi pusat kubus forearm. Kubus forearm juga digambar menggunakan `glutWireCube()` dengan skala yang sama seperti upper arm.  
  
d. Event Keyboard: Fungsi `keyboard()` mengatur input dari pengguna. Ketika tombol tertentu ditekan (misalnya 's' dan 'S' untuk mengubah nilai `shoulder`, 'e' dan 'E' untuk `elbow`), nilai variabel tersebut diubah (ditambah atau dikurangkan 5 derajat) dan fungsi `glutPostRedisplay()` dipanggil untuk merefresh tampilan sehingga pergerakan sendi terlihat.  
  
e. Proyeksi dan Tampilan: Fungsi `reshape()` memastikan tampilan tetap proporsional dengan mengatur viewport dan proyeksi perspektif. Selain itu, `glTranslatef(0.0, 0.0, -5.0)` pada `modelview` berfungsi memindahkan “kamera” ke belakang agar objek 3D dapat terlihat dengan jelas.  
  
Dengan teknik hierarkis ini, rotasi pada bahu akan membawa keseluruhan lengan (upper arm, forearm, dan selanjutnya tangan jika ditambahkan), sedangkan rotasi pada siku hanya mempengaruhi bagian bawah lengan.

## 2. Source code :

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

```
/*
```

```
    Nama          : Moh Yusril Nur Syabani
```

```
    NIM           : 24060123140181
```

```
    Lab           : GKV C1
```

```
    Tanggal       : 17 Maret 2025
```

```
*/
```

```
static int shoulder = 0, elbow = 0, wrist = 0;
```

```
static int thumb = 0, index_finger = 0, middle_finger = 0, ring_finger = 0, pinky = 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();
```

```
    // Bahu
```

```
    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();


// Siku dan lengan

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();


// Pergelangan tangan dan telapak tangan

glTranslatef(1.0, 0.0, 0.0);

glRotatef((GLfloat) wrist, 0.0, 1.0, 0.0);

glTranslatef(0.5, 0.0, 0.0);

glPushMatrix();

glScalef(1.0, 1.0, 1.0);

glutWireCube(1.0);

glPopMatrix();


// Jempol

glPushMatrix();

glTranslatef(0.0, 0.7, 0.0);

glRotatef((GLfloat) thumb, 0.0, 0.0, 1.0);
```

```
glTranslatef(0.0, 0.25, 0.0);
```

```
glScalef(0.2, 0.5, 0.2);
```

```
glutWireCube(1.0);
```

```
glPopMatrix();
```

```
// Telunjuk
```

```
    glPushMatrix();
```

```
    glTranslatef(0.6, 0.4, 0.0);
```

```
    glRotatef((GLfloat) index_finger, 0.0, 1.0, 0.0);
```

```
    glTranslatef(0.25, 0.0, 0.0);
```

```
    glScalef(0.5, 0.2, 0.2);
```

```
    glutWireCube(1.0);
```

```
    glPopMatrix();
```

```
// Jari tengah
```

```
glPushMatrix();
```

```
glTranslatef(0.6, 0.1, 0.0);
```

```
glRotatef((GLfloat) middle_finger, 0.0, 1.0, 0.0);
```

```
glTranslatef(0.35, 0.0, 0.0);
```

```
glScalef(0.7, 0.2, 0.2);
```

```
glutWireCube(1.0);
```

```
glPopMatrix();
```

```
// Jari manis
```

```
glPushMatrix();
```

```
glTranslatef(0.6, -0.2, 0.0);
```

```

glRotatef((GLfloat) ring_finger, 0.0, 1.0, 0.0);

glTranslatef(0.3, 0.0, 0.0);

glScalef(0.6, 0.2, 0.2);

glutWireCube(1.0);

glPopMatrix();


// Kelingking

glPushMatrix();

glTranslatef(0.6, -0.5, 0.0);

glRotatef((GLfloat) pinky, 0.0, 1.0, 0.0);

glTranslatef(0.2, 0.0, 0.0);

glScalef(0.4, 0.2, 0.2);

glutWireCube(1.0);

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(-2.0, 0.0, -5.0);
}

void keyboard(unsigned char key, int x, int y) {
    switch (key) {
        // Kontrol bahu
        case 'a': shoulder = (shoulder + 5) % 360;
            glutPostRedisplay();
            break;
        case 'A': shoulder = (shoulder - 5) % 360;
            glutPostRedisplay();
            break;

        // Kontrol siku
        case 's': elbow = (elbow + 5) % 360;
            glutPostRedisplay();
            break;
        case 'S': elbow = (elbow - 5) % 360;
            glutPostRedisplay();
            break;

        // Kontrol pergelangan tangan
        case 'd': wrist = (wrist + 5) % 360;
            glutPostRedisplay();
            break;
        case 'D': wrist = (wrist - 5) % 360;

```

```
    glutPostRedisplay();

    break;

// Kontrol jari

case 'q': thumb = (thumb + 5) % 90;

    glutPostRedisplay();

    break;

case 'Q': thumb = (thumb - 5) % 90;

    glutPostRedisplay();

    break;


case 'w': index_finger = (index_finger - 5) % 90;

    glutPostRedisplay();

    break;

case 'W': index_finger = (index_finger + 5) % 90;

    glutPostRedisplay();

    break;


case 'e': middle_finger = (middle_finger - 5) % 90;

    glutPostRedisplay();

    break;

case 'E': middle_finger = (middle_finger + 5) % 90;

    glutPostRedisplay();

    break;


case 'r': ring_finger = (ring_finger - 5) % 90;
```

```

        glutPostRedisplay();

        break;

    case 'R': ring_finger = (ring_finger + 5) % 90;

        glutPostRedisplay();

        break;


    case 't': pinky = (pinky - 5) % 90;

        glutPostRedisplay();

        break;

    case 'T': pinky = (pinky + 5) % 90;

        glutPostRedisplay();

        break;


    case 27: exit(0); // ESC key

        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;
```

```
}
```

### 3. Ss program:





