

Nama: Shofi Rasyida Nata

NIM: 24060123120007

LAB: GKV E1

1. Jelaskan Bagaimana cara kerja Pergerakan Tangan pada code diatas? (code asli dari modul, belum ada perubahan menambahkan telapak tangan dan jari)

Jawab:

Code ini mensimulasikan pergerakan tangan yang terdiri dari Shoulder (bahu) untuk menggerakkan bagian atas lengan (upper arm) dan Elbow (siku) untuk menggerakkan bagian bawah lengan (forearm). Setiap bagian digambarkan dengan kubus wireframe yang bisa berotasi terhadap titik sambungan. Pergerakan dikontrol lewat keyboard. Pergerakannya dengan transformasi yaitu translasi posisi awal, rotasi sesuai input (shoulder/elbow), translasi ke posisi bagian berikutnya, skala dan gambar kubus.

```
glTranslatef(-1.0, 0.0, 0.0);    // Geser ke posisi awal  
bahu
```

```
glRotatef((GLfloat)shoulder, 0.0, 0.0, 1.0); // Putar bahu
```

```
glTranslatef(1.0, 0.0, 0.0);    // Geser ke tengah upper arm
```

code tersebut untuk:

- Pertama, posisi digeser ke titik bahu.
- Lalu, lengan atas diputar sesuai nilai shoulder.
- Lalu digeser ke tengah lengan atas, dan digambar sebagai kubus diskalakan horizontal (2.0 x 0.4 x 1.0).

```
glTranslatef(1.0, 0.0, 0.0);    // Geser ke posisi siku
```

```
glRotatef((GLfloat)elbow, 0.0, 0.0, 1.0); // Putar siku
```

```
glTranslatef(1.0, 0.0, 0.0);    // Geser ke tengah forearm
```

Code tersebut untuk menggambar forearm (lengan bawah)

- Dari ujung lengan atas, geser ke titik siku.
- Lalu putar forearm sesuai nilai elbow.

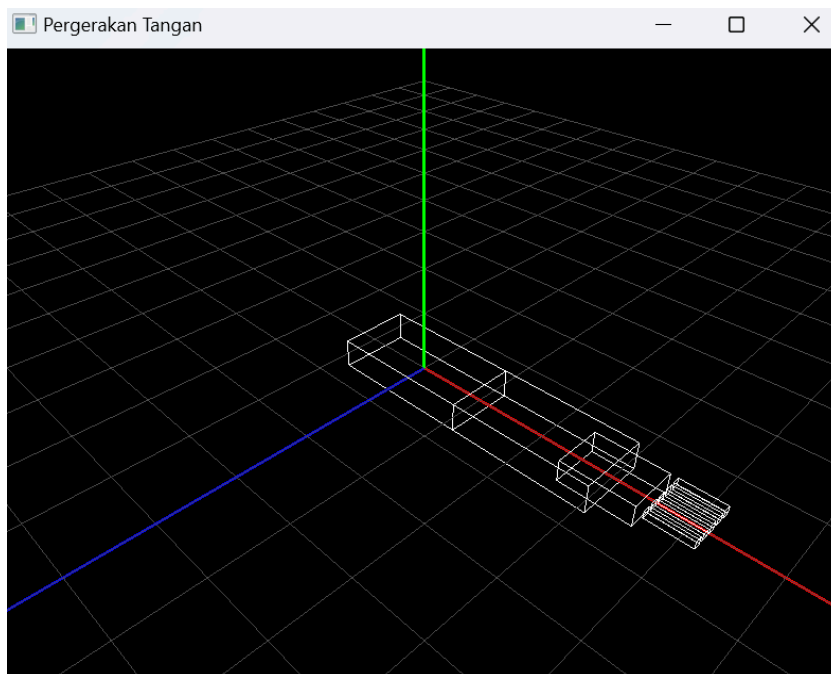
- Geser ke tengah forearm, dan gambar kubus sebagai forearm.

Fungsi KeyCallback mengatur input user. Tombol S untuk memutar bahu searah jarum jam, W untuk memutar bahu berlawanan jarum jam, E untuk memutar siku searah jarum jam, dan Q untuk memutar siku berlawanan jarum jam. Nilai shoulder dan elbow akan diputar dalam rentang 0° – 359° , karena $\% 360$.

2. Simulasikan menggunakan sumbu x, y, z bagaimana operasi yang dilakukan ketika keyboard diketik (dalam bentuk screenshot dan milimeter blok)

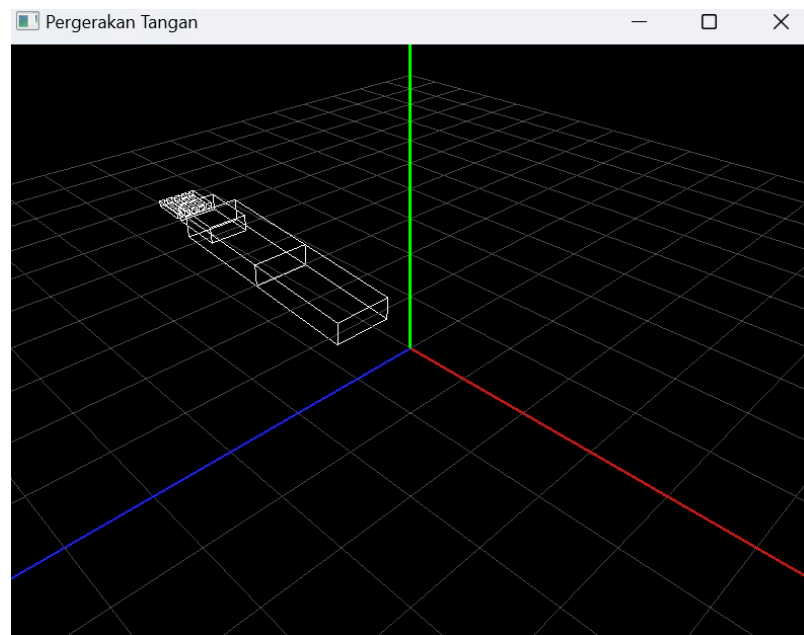
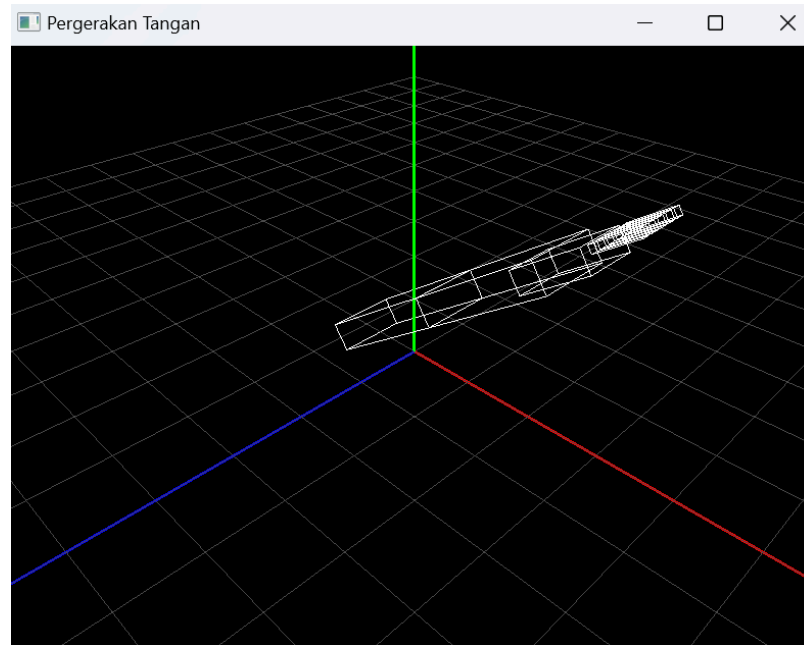
Jawab:

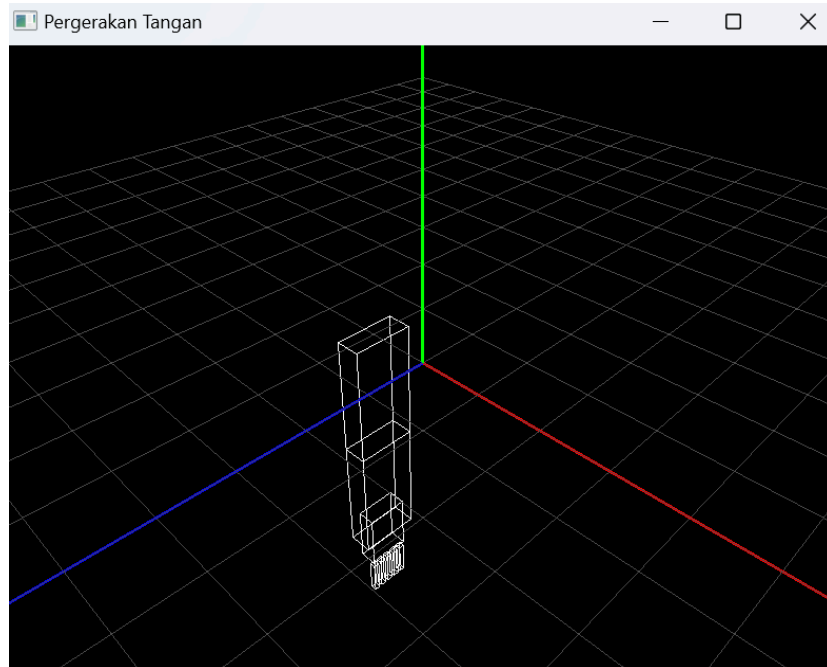
Berikut adalah hasil dari code saya menggunakan sumbu x, y, z dan terdapat milimeter blok atau grid nya.



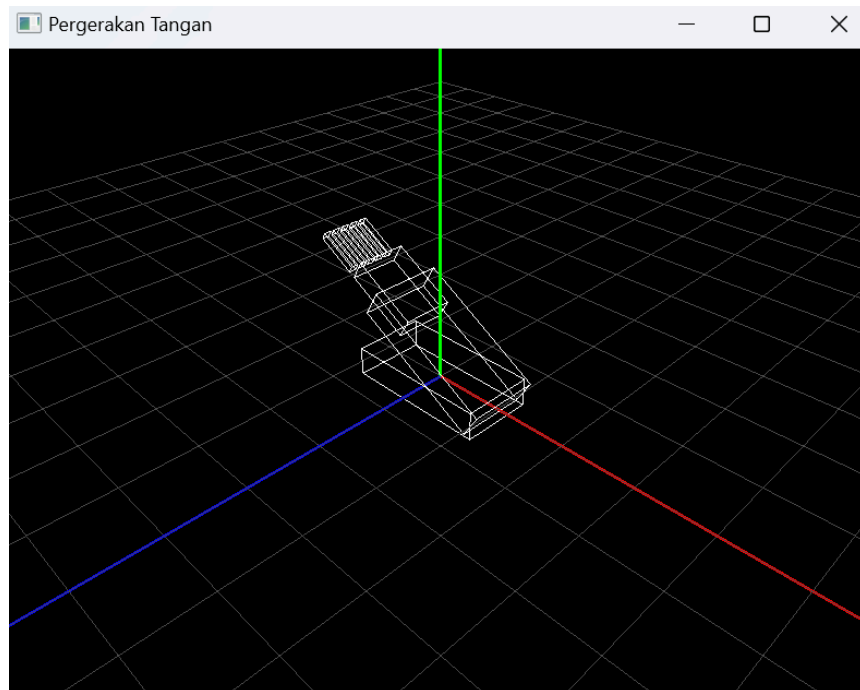
a. Gerakan bahu

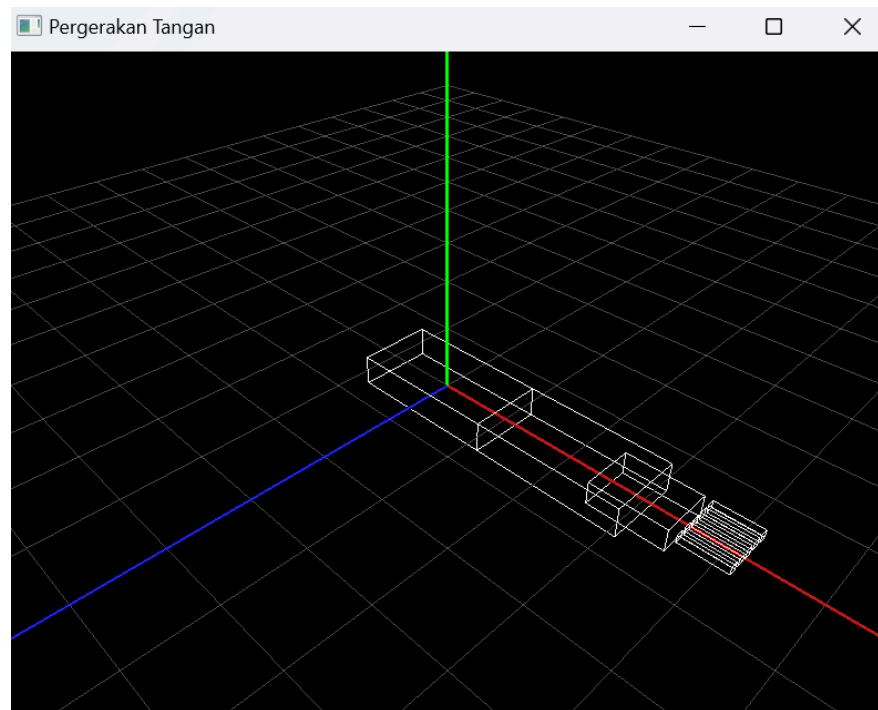
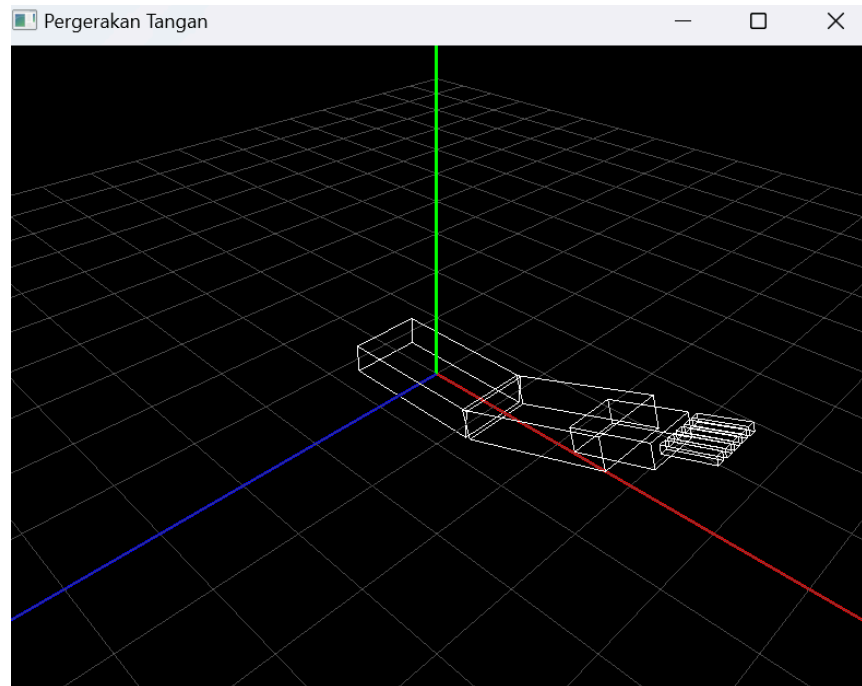
Dengan menekan tombol keyboard S/W



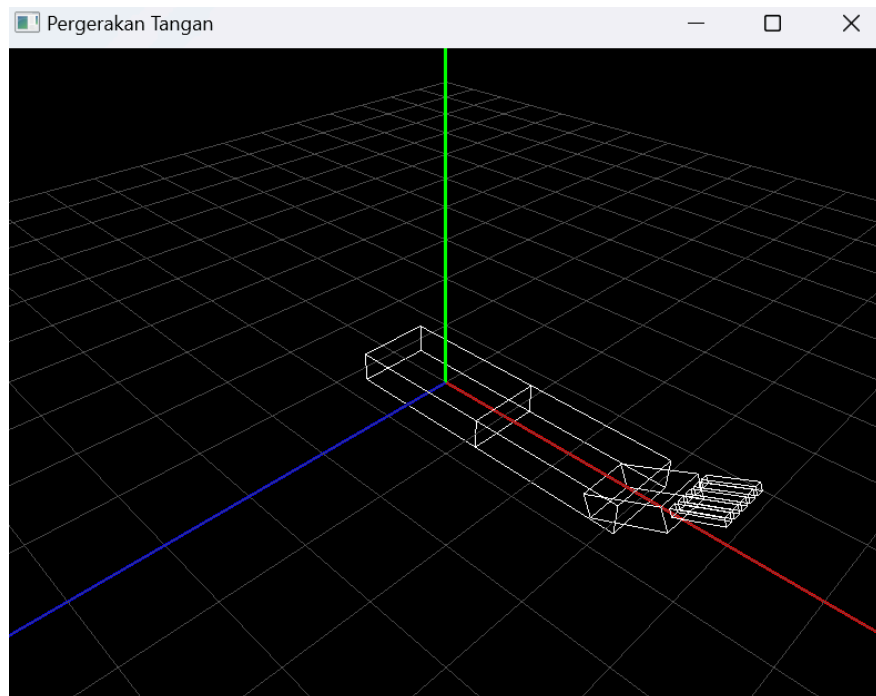
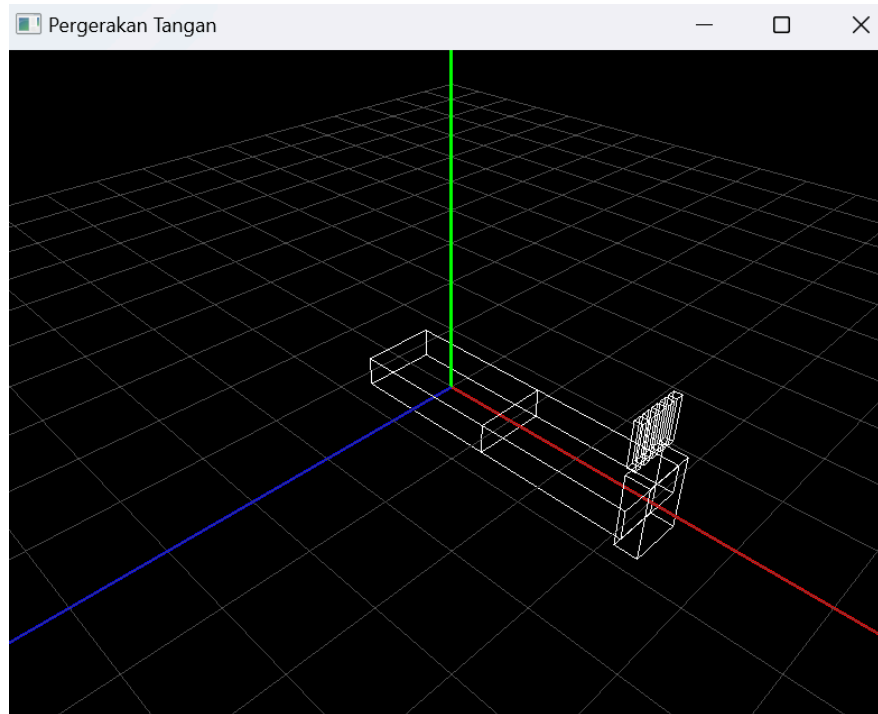


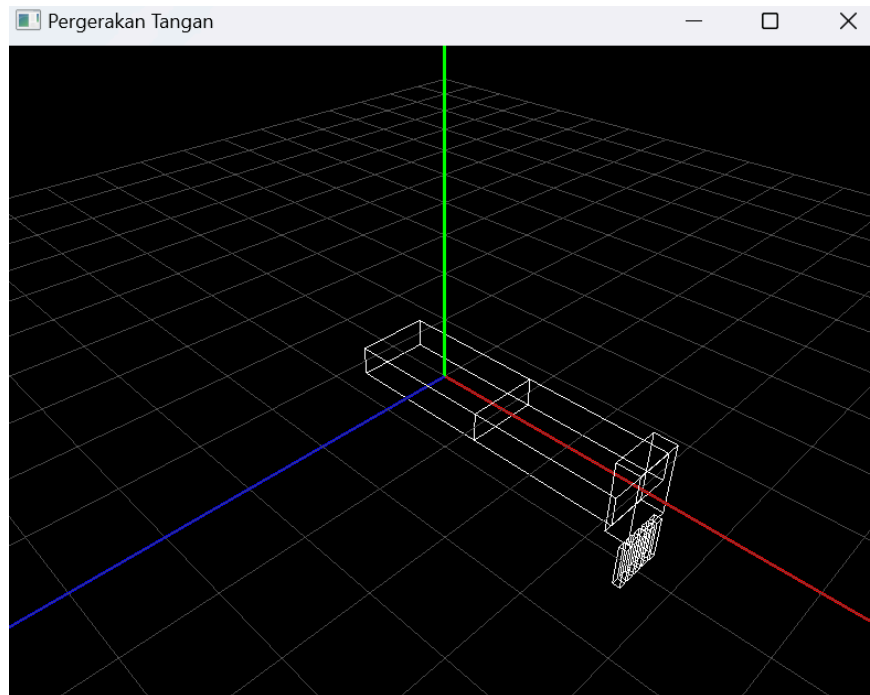
- b. Gerakan siku
Dengan menekan tombol keyboard E/Q



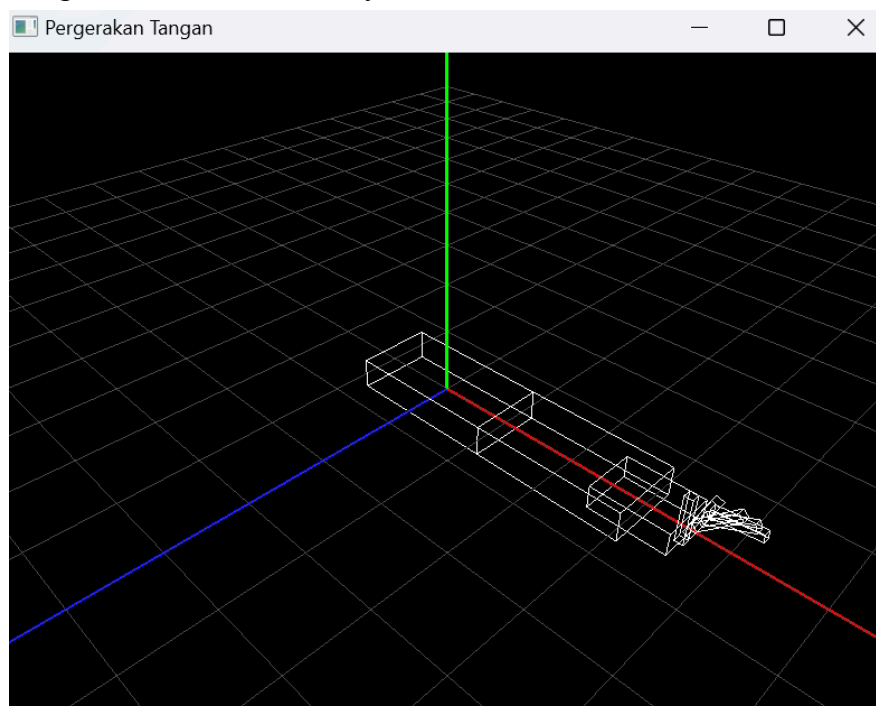


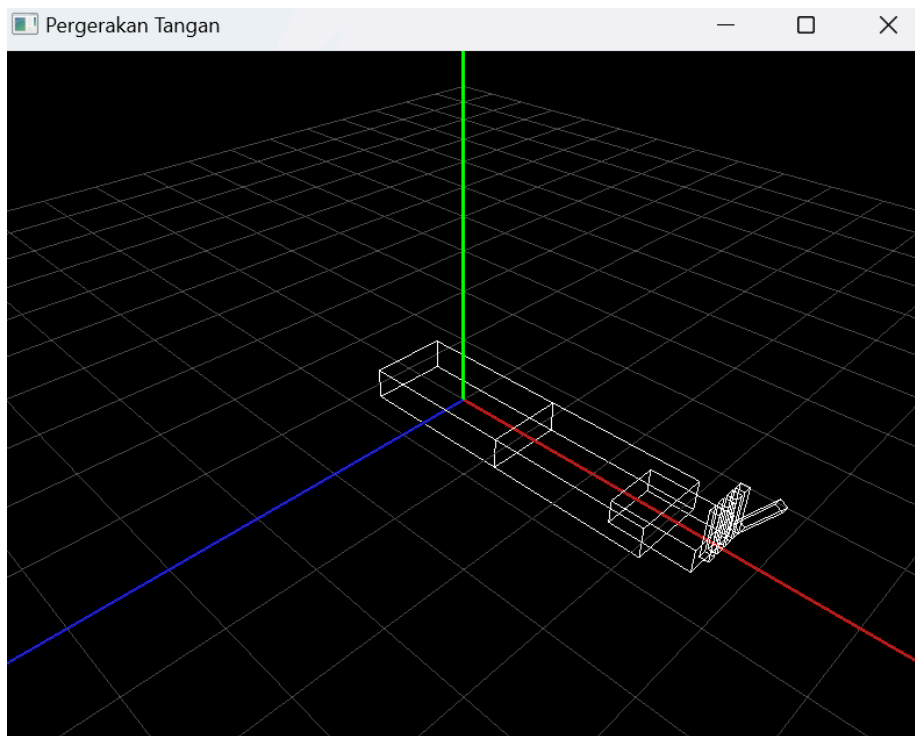
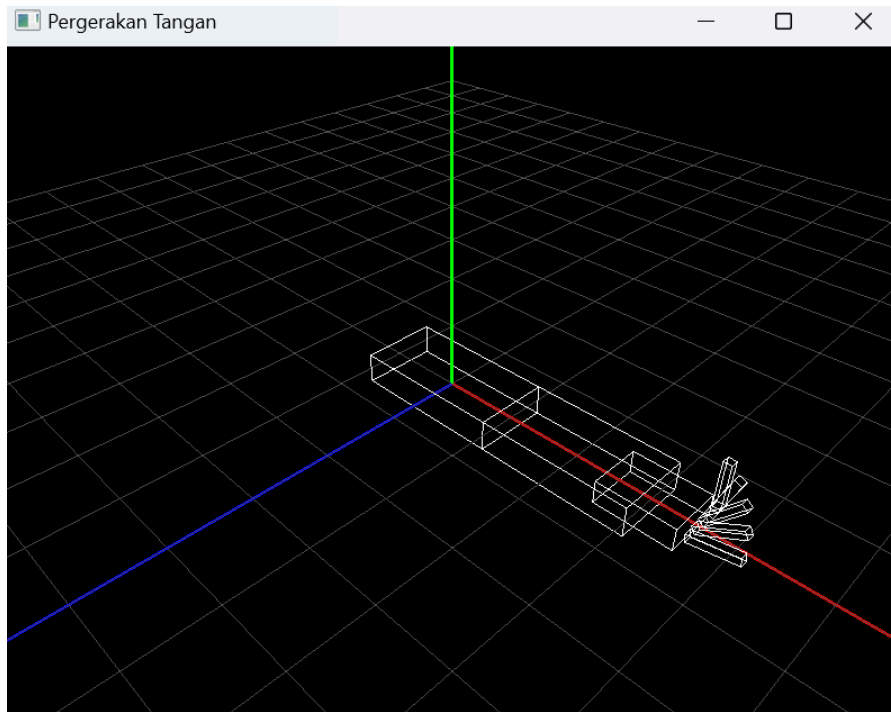
- c. Gerakan pergelangan/telapak tangan
Dengan menekan tombol keyboard R/F





- d. Gerakan jari
Dengan menekan tombol keyboard 1-5





3. Jelaskan struktur dari tugas yang kalian buat dalam proyeksi, proyeksi projection dan modelview dalam !

Jawab:

Struktur tugas yang saya buat menggunakan dua matriks utama dalam OpenGL, yaitu proyeksi (projection) dan modelview.

Projection(proyeksi)

Dilakukan di fungsi Reshape():

```
glMatrixMode(GL_PROJECTION);  
glLoadIdentity();  
gluPerspective(45.0, (float)width / height, 1.0, 100.0);
```

Penjelasan:

- `glMatrixMode(GL_PROJECTION)`
Memberitahu OpenGL bahwa kita akan mengatur matrix proyeksi, yaitu cara objek 3D diproyeksikan ke layar 2D.
- `glLoadIdentity()`
Mengatur matrix ke identitas (kosongkan dari transformasi sebelumnya).
- `gluPerspective(fov, aspect, zNear, zFar)`
Mengatur proyeksi perspektif
- `45.0`
Field of View (FOV) vertikal dalam derajat (seberapa lebar pandangannya).
- `(float)width / height`
Rasio aspek dari jendela (penting agar tidak melebar/menyempit)
- `1.0 dan 100.0`
Jarak terdekat dan terjauh yang terlihat kamera (near/far clipping plane).

Bagian ini bertanggung jawab untuk menentukan bagaimana ruang 3D akan dipetakan menjadi 2D di layar monitor, menciptakan efek perspektif seperti di dunia nyata.

ModelView

ModelView mengatur posisi kamera dan objek.

```
glMatrixMode(GL_MODELVIEW);  
glLoadIdentity();  
gluLookAt(6, 6, 6, 0, 0, 0, 0, 1, 0);
```

Penjelasan:

- `glMatrixMode(GL_MODELVIEW)`
Mengatur matrix model & view (gabungan dari posisi kamera dan transformasi objek).
- `glLoadIdentity()`
Mengosongkan transformasi sebelumnya.
- `gluLookAt(...)`
Menentukan posisi kamera
- `(6, 6, 6)` Posisi kamera di ruang 3D.
- `(0, 0, 0)` Titik yang dilihat oleh kamera (pusat koordinat).
- `(0, 1, 0)` Arah atas kamera (arah sumbu Y).

Bagian ini menentukan dari mana kita melihat (kamera), arah pandang, dan transformasi objek seperti rotasi (`glRotatef`) dan translasi (`glTranslatef`) yang membentuk lengan dan jarinya.

Pada bagian proyeksi, saya menggunakan fungsi `gluPerspective()` untuk membuat efek perspektif agar simulasi lengan robot terlihat realistis dalam ruang 3D. Proyeksi ini diatur setelah mengaktifkan mode `GL_PROJECTION`, dan digunakan untuk menentukan sudut pandang, rasio layar, serta jarak dekat dan jauh kamera. Setelah itu, beralih ke `GL_MODELVIEW` untuk mengatur posisi kamera menggunakan `gluLookAt()`, sekaligus menerapkan transformasi terhadap objek-objek dalam simulasi, seperti rotasi dan translasi setiap bagian dari lengan robot. Modelview dengan `glPushMatrix()` dan `glPopMatrix()` agar setiap sendi-seperti bahu, siku, pergelangan, dan jari dapat bergerak saling bergantung satu sama lain secara terstruktur. Dengan ini, simulasi dapat menampilkan pergerakan yang realistis dan terkontrol mengikuti sumbu koordinat yang ditampilkan di layar.