MENGGAMBAR TANGAN DENGAN *OPENGL*

****

**LAPORAN**

**Disusun untuk Memenuhi Tugas III Praktikum GKV**

**DISUSUN OLEH:  
ANANDA PRABU TRITYA VIJAYA  
24060117130048**

**PROGRAM STUDI STRATA I INFORMATIKA**

**DEPARTEMEN ILMU KOMPUTER/INFORMATIKA**

**FAKULTAS SAINS DAN MATEMATIKA**

**UNIVERSITAS DIPONEGORO**

**SEMARANG**

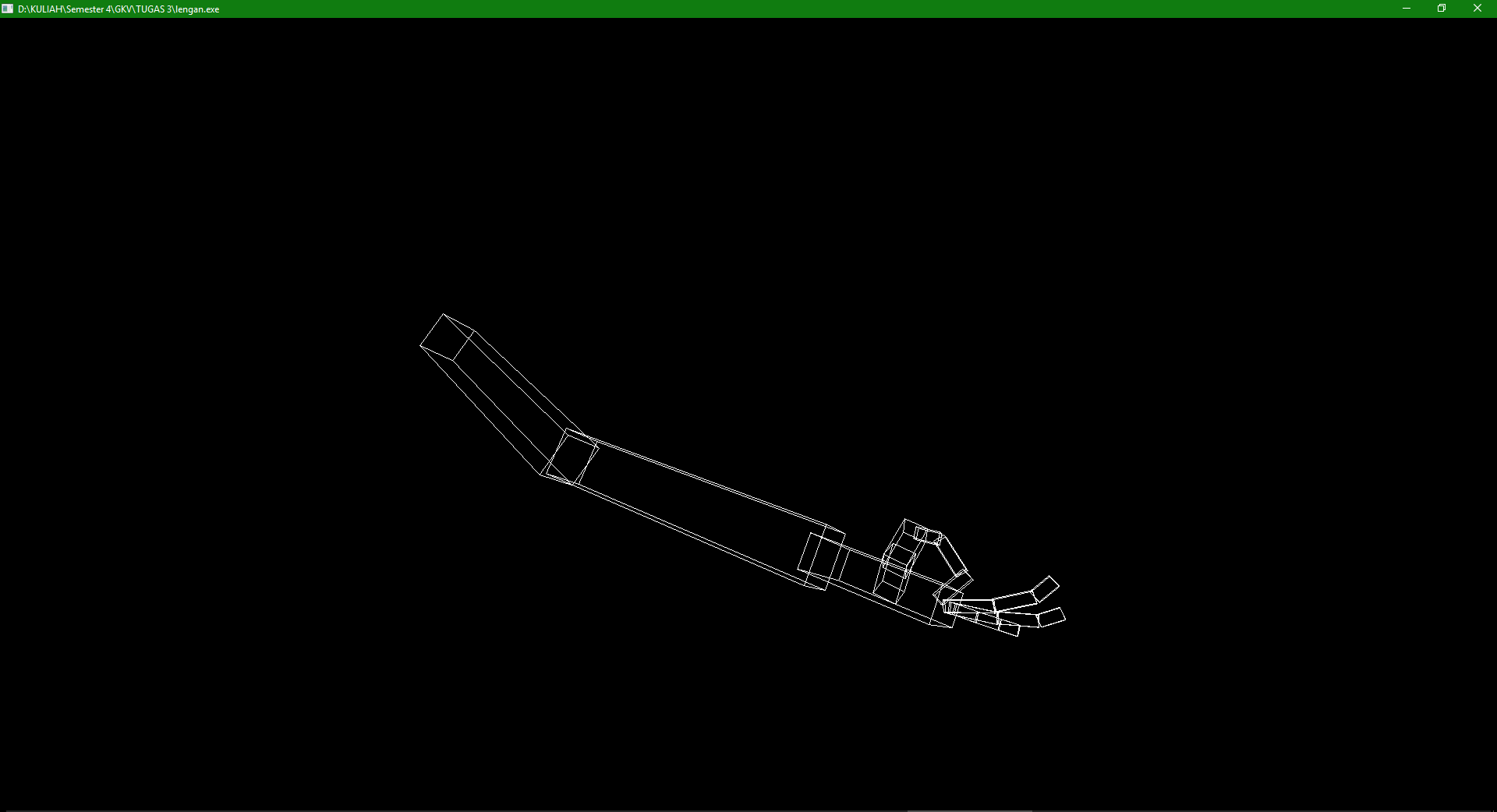
**2018**

**PEMBAHASAN**

1. Buatlah kerangka tangan menggunakan OpenGL!

|  |
| --- |
| #include <GL/glut.h>  static int shoulder1 = 0, shoulder2 = 0;  static int elbow = 0;  static int telapak1 = 0, telapak2 = 0, telapak3 = 0;  static int jempol1 = 0, jempol2 = 0, jempol3 = 0;  static int telunjuk1 = 0, telunjuk2 = 0, telunjuk3 = 0;  static int tengah1 = 0, tengah2 = 0, tengah3 = 0;  static int manis1 = 0, manis2 = 0, manis3 = 0;  static int kelingking1 = 0, kelingking2 = 0, kelingking3 = 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();  //lengan atas  glTranslatef(-2.5,.7,.0);  glRotatef((GLfloat) shoulder1,0,0,1);  glRotatef((GLfloat) shoulder2,0,1,0);  glTranslatef(.1,.0,.0);  glPushMatrix();  glTranslatef(1.0,.0,0);  glScalef(1.8,.3,.3);  glutWireCube(1);  glPopMatrix();  //lengan bawah  glTranslatef(1.8,.0,.0);  glRotatef((GLfloat) elbow,0,0,1);  glTranslatef(1,.0,.0);  glPushMatrix();  glScalef(1.8,.3,.3);  glutWireCube(1);  glPopMatrix();  //telapak tangan  glTranslatef(.9,.0,.0);  glRotatef((GLfloat) telapak1,0,1,0);  glRotatef((GLfloat) telapak2,0,0,1);  glRotatef((GLfloat) telapak3,1,0,0);  glTranslatef(.3,.0,.0);  glPushMatrix();  glScalef(.6,.17,.6);  glutWireCube(1);  glPopMatrix();  //Ibu Jari  glPushMatrix();  glTranslatef(.0,-.05,.3);  glRotatef((GLfloat) jempol1,1,0,0);  glRotatef((GLfloat) jempol2,0,1,0);  glTranslatef(.1,.05,.1);  glPushMatrix();  glScalef(.1,.1,.2);  glutWireCube(1);  glPopMatrix();  glTranslatef(-.1,-.05,.1);  glRotatef((GLfloat) jempol3,1,0,0);  glTranslatef(.1,.05,.1);  glPushMatrix();  glScalef(.1,.1,.2);  glutWireCube(1);  glPopMatrix();  glPopMatrix();  //Telunjuk  glPushMatrix();  glTranslatef(.28,.0,.23);  glRotatef((GLfloat) telunjuk1,0,0,1);  glTranslatef(.1,.0,.0);  glPushMatrix();  glScalef(.16,.05,.05);  glutWireCube(1);  glPopMatrix();  glTranslatef(.06,.0,.0);  glRotatef((GLfloat) telunjuk2,0,0,1);  glTranslatef(.1,.0,.0);  glPushMatrix();  glScalef(.16,.05,.05);  glutWireCube(1);  glPopMatrix();  glTranslatef(.08,.0,.0);  glRotatef((GLfloat) telunjuk3,0,0,1);  glTranslatef(.05,.0,.0);  glPushMatrix();  glScalef(.1,.05,.05);  glutWireCube(1);  glPopMatrix();  glPopMatrix();  //tengah  glPushMatrix();  glTranslatef(.3,.0,.10);  glRotatef((GLfloat) tengah1,0,0,1);  glTranslatef(.1,.0,.0);  glPushMatrix();  glScalef(.2,.05,.05);  glutWireCube(1);  glPopMatrix();  glTranslatef(.08,.0,.0);  glRotatef((GLfloat) tengah2,0,0,1);  glTranslatef(.1,.0,.0);  glPushMatrix();  glScalef(.16,.05,.05);  glutWireCube(1);  glPopMatrix();  glTranslatef(.08,.0,.0);  glRotatef((GLfloat) tengah3,0,0,1);  glTranslatef(.05,.0,.0);  glPushMatrix();  glScalef(.1,.05,.05);  glutWireCube(1);  glPopMatrix();  glPopMatrix();  //manis  glPushMatrix();  glTranslatef(.3,.0,-.05);  glRotatef((GLfloat) manis1,0,0,1);  glTranslatef(.1,.0,.0);  glPushMatrix();  glScalef(.2,.05,.05);  glutWireCube(1);  glPopMatrix();  glTranslatef(.08,.0,.0);  glRotatef((GLfloat) manis2,0,0,1);  glTranslatef(.1,.0,.0);  glPushMatrix();  glScalef(.16,.05,.05);  glutWireCube(1);  glPopMatrix();  glTranslatef(.08,.0,.0);  glRotatef((GLfloat) manis3,0,0,1);  glTranslatef(.05,.0,.0);  glPushMatrix();  glScalef(.1,.05,.05);  glutWireCube(1);  glPopMatrix();  glPopMatrix();  //kelingking  glPushMatrix();  glTranslatef(.25,.0,-.2);  glRotatef((GLfloat) kelingking1,0,0,1);  glTranslatef(.1,.0,.0);  glPushMatrix();  glScalef(.1,.05,.05);  glutWireCube(1);  glPopMatrix();  glTranslatef(.05,.0,.0);  glRotatef((GLfloat) kelingking2,0,0,1);  glTranslatef(.05,.0,.0);  glPushMatrix();  glScalef(.1,.05,.05);  glutWireCube(1);  glPopMatrix();  glTranslatef(.04,.0,.0);  glRotatef((GLfloat) kelingking3,0,0,1);  glTranslatef(.05,.0,.0);  glPushMatrix();  glScalef(.08,.05,.05);  glutWireCube(1);  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 'q': shoulder1 = (shoulder1 + 2) % 360;  if(shoulder1 > 90) shoulder1 = 90;  glutPostRedisplay(); break;  case 'Q': shoulder1 = (shoulder1 - 2) % 360;  if(shoulder1 < -90) shoulder1 =-90;  glutPostRedisplay(); break;  case 'w': shoulder2 = (shoulder2 + 2) % 360;  if(shoulder2 > 90) shoulder2 = 90;  glutPostRedisplay(); break;  case 'W': shoulder2 = (shoulder2 - 2) % 360;  if(shoulder2 < -70) shoulder2 =-70;  glutPostRedisplay(); break;  case 'e': elbow = (elbow + 2) % 360;  if(elbow > 130) elbow = 130;  glutPostRedisplay(); break;  case 'E': elbow = (elbow - 2) % 360;  if(elbow < -5) elbow = -5;  glutPostRedisplay(); break;  case 'r' : telapak1 = (telapak1 + 2) % 360;  if(telapak1 > 5) telapak1 = 5;  glutPostRedisplay(); break;  case 'R' : telapak1 = (telapak1 - 2) % 360;  if(telapak1 < -20) telapak1= -20;  glutPostRedisplay(); break;  case 't' : telapak2 = (telapak2 + 2) % 360;  if(telapak2 > 90) telapak2 = 90;  glutPostRedisplay(); break;  case 'T' : telapak2 = (telapak2 - 2) % 360;  if(telapak2 < -20) telapak2 = -20;  glutPostRedisplay(); break;  case '5' : telapak3 = (telapak3 + 2) % 360;  if(telapak3 > 0) telapak3 = 0;  glutPostRedisplay(); break;  case '%' : telapak3 = (telapak3 - 2) % 360;  if(telapak3 < -90) telapak3= -90;  glutPostRedisplay(); break;  case 'y' : jempol1 = (jempol1 + 2) % 360;  if(jempol1 > 0) jempol1 = 0;  glutPostRedisplay(); break;  case 'Y' : jempol1 = (jempol1 - 2) % 360;  if(jempol1 < -100) jempol1 = -100;  glutPostRedisplay(); break;  case 'u' : jempol2 = (jempol2 + 2) % 360;  if(jempol2 > 60) jempol2 = 60;  glutPostRedisplay(); break;  case 'U' : jempol2 = (jempol2 - 2) % 360;  if(jempol2 < 0) jempol2 = 0;  glutPostRedisplay(); break;  case 'i' : jempol3 = (jempol3 - 2) % 360;  if(jempol3 < -90) jempol3 = -90;  glutPostRedisplay(); break;  case 'I' : jempol3 = (jempol3 + 2) % 360;  if(jempol3 > 2) jempol3 = 2;  glutPostRedisplay(); break;  case 'o' : telunjuk1 = (telunjuk1 + 2) % 360;  if(telunjuk1 > 90) telunjuk1 = 90;  glutPostRedisplay(); break;  case 'O' : telunjuk1 = (telunjuk1 - 2) % 360;  if(telunjuk1 < -2) telunjuk1 = -2;  glutPostRedisplay(); break;  case 'p' : telunjuk2 = (telunjuk2 + 2) % 360;  if(telunjuk2 > 90) telunjuk2 = 90;  glutPostRedisplay(); break;  case 'P' : telunjuk2 = (telunjuk2 - 2) % 360;  if(telunjuk2 < -4) telunjuk2 = -4;  glutPostRedisplay(); break;  case '[' : telunjuk3 = (telunjuk3 + 2) % 360;  if(telunjuk3 > 75) telunjuk3 = 75;  glutPostRedisplay(); break;  case '{' : telunjuk3 = (telunjuk3 - 2) % 360;  if(telunjuk3 < 0) telunjuk3 = 0;  glutPostRedisplay(); break;  case 'a' : tengah1 = (tengah1 + 2) % 360;  if(tengah1 > 90) tengah1 = 90;  glutPostRedisplay(); break;  case 'A' : tengah1 = (tengah1 - 2) % 360;  if(tengah1 < -2) tengah1 = -2;  glutPostRedisplay(); break;  case 's' : tengah2 = (tengah2 + 2) % 360;  if(tengah2 > 90) tengah2 = 90;  glutPostRedisplay(); break;  case 'S' : tengah2 = (tengah2 - 2) % 360;  if(tengah2 < -4) tengah2 = -4;  glutPostRedisplay(); break;  case 'd' : tengah3 = (tengah3 + 2) % 360;  if(tengah3 > 75) tengah3 = 75;  glutPostRedisplay(); break;  case 'D' : tengah3 = (tengah3 - 2) % 360;  if(tengah3 < 0) tengah3 = 0;  glutPostRedisplay(); break;  case 'f' : manis1 = (manis1 + 2) % 360;  if(manis1 > 90) manis1 = 90;  glutPostRedisplay(); break;  case 'F' : manis1 = (manis1 - 2) % 360;  if(manis1 < -2) manis1 = -2;  glutPostRedisplay(); break;  case 'g' : manis2 = (manis2 + 2) % 360;  if(manis2 > 90) manis2 = 90;  glutPostRedisplay(); break;  case 'G' : manis2 = (manis2 - 2) % 360;  if(manis2 < -4) manis2 = -4;  glutPostRedisplay(); break;  case 'h' : manis3 = (manis3 + 2) % 360;  if(manis3 > 75) manis3 = 75;  glutPostRedisplay(); break;  case 'H' : manis3 = (manis3 - 2) % 360;  if(manis3 < 0) manis3 = 0;  glutPostRedisplay(); break;  case 'j' : kelingking1 = (kelingking1 + 2) % 360;  if(kelingking1 > 90) kelingking1 = 90;  glutPostRedisplay(); break;  case 'J' : kelingking1 = (kelingking1 - 2) % 360;  if(kelingking1 < -2) kelingking1 = -2;  glutPostRedisplay(); break;  case 'k' : kelingking2 = (kelingking2 + 2) % 360;  if(kelingking2 > 90) kelingking2 = 90;  glutPostRedisplay(); break;  case 'K' : kelingking2 = (kelingking2 - 2) % 360;  if(kelingking2 < -4) kelingking2 = -4;  glutPostRedisplay(); break;  case 'l' : kelingking3 = (kelingking3 + 2) % 360;  if(kelingking3 > 75) kelingking3 = 75;  glutPostRedisplay(); break;  case 'L' : kelingking3 = (kelingking3 - 2) % 360;  if(kelingking3 < 0) kelingking3 = 0;  glutPostRedisplay(); break;  case 27: exit(0); break;  default: break;  }  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutInitDisplayMode (GLUT\_DOUBLE | GLUT\_RGB);  glutInitWindowSize (900, 900);  glutInitWindowPosition (100, 100);  glutCreateWindow (argv[0]); init();  glutDisplayFunc(display);  glutReshapeFunc(reshape);  glutKeyboardFunc(keyboard);  glutMainLoop(); return 0;  } |

Screenshot Hasil:



Pembahasan:

1. Pembuatan rangka

Kita ambil contoh 1 rangka (karena pada umumnya seluruh rangka cara pembuatannya hampir sama):

|  |
| --- |
| glTranslatef(1.8,.0,.0);  glRotatef((GLfloat) elbow,0,0,1);  glTranslatef(1,.0,.0);  glPushMatrix();  glScalef(1.8,.3,.3);  glutWireCube(1);  glPopMatrix(); |

1. glTranslatef yang pertama digunakan untuk mengatur tata letak objek
2. glRotatef digunakan untuk mengatur arah gerak objek jika digerakkan sesuai dengan koordinat x, y, dan z
3. glTranslatef yang kedua digunakan untuk mengatur jarak pergerakan suatu objek. Ini dapat mengubah tata letak objek juga, jadi disarankan untuk mengatur lagi glTranslatef yang pertama agar sesuai dengan yang diinginkan
4. glScalef digunakan untuk meng-skala suatu objek dengan perbesaran di koordinat x, y, dan z
5. glutWireCube digunakan untuk membuat objek kubus
6. *Keyboard binding* untuk pergerakan objek

|  |
| --- |
| …..  switch (key){  case 'q': shoulder1 = (shoulder1 + 2) % 360;  if(shoulder1 > 90) shoulder1 = 90;  glutPostRedisplay(); break;  case 'Q': shoulder1 = (shoulder1 - 2) % 360;  if(shoulder1 < -90) shoulder1 =-90;  ………..  ……….. |

Dengan membuat fungsi keyboard, kita dapat membuat suatu objek bergerak, seperti *script code* di atas dimana jika menekan q suatu objek akan bergerak dengan pergerakan maksimal 90 derajat ke koordinat yang sesuai dengan yang sudah diatur dengan fungsi glRotatef sebelumnya.

1. Main program function

|  |
| --- |
| int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutInitDisplayMode (GLUT\_DOUBLE | GLUT\_RGB);  glutInitWindowSize (900, 900);  glutInitWindowPosition (100, 100);  glutCreateWindow (argv[0]); init();  glutDisplayFunc(display);  glutReshapeFunc(reshape);  glutKeyboardFunc(keyboard);  glutMainLoop(); return 0;  } |

1. glutInit digunakan untuk menginisialisasi library GLUT
2. glutInitDisplayMode digunakan untuk menginisialisasi display mode. Disini Display mode yang digunakan adalah *Double buffered Window* dan juga *RGB mode Window*
3. glutInitWindowSize digunakan untuk mengatur besar window yang digunakan untuk display
4. glutInitWindowPosition digunakan untuk menginisialisasi posisi window awal
5. glutCreateWindow digunakan untuk membuat window dengan *setting*-an yang sudah di atur sebelumnya
6. glutDisplayFunc digunakan untuk memanggil fungsi display dengan parameter nama fungsi
7. glutReshapeFunc digunakan untuk memanggil fungsi reshape dengna parameter nama fungsi
8. glutKeyboardFunc digunakan untuk memanggil fungsi keyboard dengan parameter nama fungsi
9. glutMainLoop digunakan untuk menandakan pemrosesan glut terjadi