| Nama | Adidarma Putra Harjuna |
| --- | --- |
| NRP | 5221600027 |
| Tanggal | 07/09/2022 |
| Praktek | Primitive Drawing |

Kerjakan Contoh Program berikut:

| 1 | Struktur dasar program OpenGL. Coba program berikut kemudian berikan penjelasan terkait fungsi - fungsi openGL yang digunakan pada baris program. |
| --- | --- |
|  |  |
|  | Output Program |
|  | Penjelasan:   * Fungsi void userdraw(void)digunakan untuk tempat menggambar * Kode dibawah ini digunakan untuk membuat Window yang akan menampilkan Gambar   glutInitWindowPosition(100,100);  glutInitWindowSize(640,480);  glutCreateWindow ("Test");   * Kode dibawah ini digunakan untuk memberi warna pada window(dalam hal ini berwarna putih)   glClearColor(0.0, 0.0, 0.0, 0.0);   * Kode dibawah ini adalah sistem koordinat yang akan digunakan di dalam window   gluOrtho2D(-320., 320., -240.0, 240.0); |
| 2 | **Menggambar Titik.**  Modifikasi program 1 agar dapat mengimplementasikan fungsi gambar titik berikut |
|  |  |
|  | Capture Output Program: |
|  | Kesimpulan:   * Jadi, titik yang dihasilkan bisa di edit ukurannya menggunakan perintah   glPointSize(1); |
| 3 | **Menggambar Garis.**  Modifikasi program 1 agar dapat mengimplementasikan fungsi gambar garis berikut: |
|  | **Capture Output Program** |
|  | **Kesimpulan**   * Jadi, untuk membuat sebuah garis dibutuhkan 2 titik di tempat yang berbeda. |
| 4 | **Menggambar Polyline**  Modifikasi program 1 agar dapat mengimplementasikan fungsi gambar polyline berikut: |
|  | **Capture Output Program** |
|  | **Kesimpulan**   * Jadi untuk menggambar polyline dibutuhkan perintah   glBegin(GL\_LINE\_STRIP); |
| 5 | **Menggambar Polygon**  Modifikasi program 1 agar dapat mengimplementasikan fungsi gambar poligon berikut: |
|  | **Capture Output Program** |
|  | **Kesimpulan**   * Jadi untuk menggambar polygon yang akan otomatis kembali ke titik awal, dibutuhkan perintah   glBegin(GL\_LINE\_LOOP); |
| 6 | **Mengubah Warna**  Modifikasi program 2 agar dapat mengubah warna titik berikut: |
|  | **Capture Output Program** |
|  | **Kesimpulan**   * Jadi, untuk mewarnai sebuah dot, bisa menggunakan perintah   glColor3f(0, 1, 0); |
| 7 | **Membuat Tampilan Televisi Rusak**  Modifikasi program 1 agar dapat mensimulasikan tampilan tv yang rusak: |
|  | **Capture Output Program** |
|  | **Kesimpulan**   * Jadi, untuk membuat efek seperti televisi rusak tersebut dibutuhkan titik titik yang cukup banyak dan di loop secara random. |

Kerjakan Soal Latihan Berikut:

| 1 | Buatlah program untuk membuat lingkaran. (Gunakan primitif GL\_LINE\_LOOP)  Capture Source Code:  #include <GL/glu.h>  #include <GL/gl.h>  #include <GL/glut.h>  #include <math.h>  void circle(void)  {  GLfloat x, y, angle;  glClear(GL\_COLOR\_BUFFER\_BIT);  glBegin(GL\_LINE\_LOOP);  for (angle = 0.0f; angle <= (2.0f \* 3.14); angle += 0.01f)  {  glColor3f(0, 1, 0);  x = 150.0f \* sin(angle);  y = 150.0f \* cos(angle);  glVertex3f(x, y, 0.0f);  }  glEnd();  }  void userdraw(void)  {  circle();  }  void display(void)  {  glClear(GL\_COLOR\_BUFFER\_BIT);  userdraw();  glutSwapBuffers();  }  int main(int argc, char \*\*argv)  {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);  glutInitWindowPosition(100, 100);  glutInitWindowSize(640, 480);  glutCreateWindow("Test");  glClearColor(0.0, 0.0, 0.0, 0.0);  gluOrtho2D(-320., 320., -240.0, 240.0);  glutIdleFunc(display);  glutDisplayFunc(display);  glutMainLoop();  return 0;  }  Capture Output Program: |
| --- | --- |
| 2 | Buatlah polygon dengan tampilan warna, seperti contoh berikut.    Capture Source Code:  #include <GL/glu.h>  #include <GL/gl.h>  #include <GL/glut.h>  #include <math.h>  void drawkotak()  {  glBegin(GL\_POLYGON);  glColor4f(1.0f, 0.0f, 0.0f, 0.0f);  glVertex2f(0, 0);  glColor3f(0.0f, 1.0f, 0.0f);  glVertex2f(200, 0);  glColor3f(0.0f, 0.0f, 1.0f);  glVertex2f(200, 200);  glColor4f(1.0f, 1.0f, 0.0f, 0.0f);  glVertex2f(0, 200);  glEnd();  }  void userdraw(void)  {  drawkotak();  }  void display(void)  {  glClear(GL\_COLOR\_BUFFER\_BIT);  userdraw();  glutSwapBuffers();  }  int main(int argc, char \*\*argv)  {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);  glutInitWindowPosition(100, 100);  glutInitWindowSize(640, 480);  glutCreateWindow("Test");  glClearColor(0.0, 0.0, 0.0, 0.0);  gluOrtho2D(-320., 320., -240.0, 240.0);  glutIdleFunc(display);  glutDisplayFunc(display);  glutMainLoop();  return 0;  }  Capture Output Program: |
| 3 | Buatlah Garis, Segitiga, dan Segiempat dengan variasi ukuran dan warna.  Capture Source Code:  #include <GL/glu.h>  #include <GL/gl.h>  #include <GL/glut.h>  #include <math.h>  void drawkotak(float x1, float y1, float x2, float y2, float x3, float y3, float x4, float y4)  {  glBegin(GL\_POLYGON);  glColor4f(1.0f, 0.0f, 0.0f, 0.0f);  glVertex2f(x1, y1);  glColor3f(0.0f, 1.0f, 0.0f);  glVertex2f(x2, y2);  glColor3f(0.0f, 0.0f, 1.0f);  glVertex2f(x3, y3);  glColor4f(1.0f, 1.0f, 0.0f, 0.0f);  glVertex2f(x4, y4);  glEnd();  }  void drawLine(float x1, float y1, float x2, float y2){  glBegin(GL\_LINES);  glVertex2f(x1, y1);  glVertex2f(x2, y2);  glEnd();  }  void drawsegitiga(float x1, float y1, float x2, float y2, float x3, float y3)  {  glBegin(GL\_POLYGON);  glColor4f(1.0f, 0.0f, 0.0f, 0.0f);  glVertex2f(x1, y1);  glColor3f(0.0f, 1.0f, 0.0f);  glVertex2f(x2, y2);  glColor3f(0.0f, 0.0f, 1.0f);  glVertex2f(x3, y3);  glEnd();  }  void userdraw(void) {  drawkotak(0, 0, -200, 0, -200, -200, 0, -200);  drawLine(100, 100, 200, 150);  drawsegitiga(0, 200, -200, 50, 0, 50);  }  void display(void) {  glClear( GL\_COLOR\_BUFFER\_BIT);  userdraw();  glutSwapBuffers();  }  int main(int argc, char \*\*argv){  glutInit(&argc,argv);  glutInitDisplayMode ( GLUT\_DOUBLE | GLUT\_RGB );  glutInitWindowPosition(100,100);  glutInitWindowSize(640,480);  glutCreateWindow ("Test");  glClearColor(0.0, 0.0, 0.0, 0.0);  gluOrtho2D(-320., 320., -240.0, 240.0);  glutIdleFunc(display);  glutDisplayFunc(display);  glutMainLoop();  return 0;  }  Capture Output Program: |