**TECHNICAL REPORT**

**GRAFIKA KOMPUTER**

**MODUL 5**

**“Transformasi Geometri Lanjut”**

****

**Disusun Oleh:**

TGL PRAKTIKUM : 30 April 2018

NAMA : NATIQ HASBI ALIM

NRP : 160411100152

KELAS/KELOMPOK : B3

DOSEN : ACH. KHOZAIMI, S.KOM., M.KOM.

ASISTEN : WRIDA ADI WARDANA

DOSEN PENGAMPU :

|  |
| --- |
| Disetujui : ....../……./………../Bangkalan |
| WRIDA ADI WARDANA  150411100052 |

****

**LABORATORIUM BISNIS INTELIJEN SISTEM**

**PRODI INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

** 2017/ 2018**

**BAB I**

**PENDAHULUAN**

1. **Pengantar**

Salah satu kegunaan kombinasi Transformasi adalah untuk melakukan rotasi pada obyek terhadap arbitrary point/sembarang titik (bukan terhadap sumbu koordinar). Metode rotasi terhadap arbitrary point adalah pertama-tama mentranslasikan obyek untuk berhimpit dengan sumbu koordinat, diikuti dengan rotasi, dan terkahir men-translasikan kembali obyek pada posisinya semula.

Program 5.1 menunjukkan contoh kombinasi transformasi

|  |
| --- |
| void display(){  glClear(GL\_COLOR\_BUFFER\_BIT);  glColor3f(0.0,0.0,0.5);  // Gambar kotak pertama di sudut kiri bawah  glRecti(0,0, 10, 10);  //rotasi kotak kedua terhadap titik tengah kotak  glTranslated(25.0, 25.0, 0);  glRotated(45, 0, 0, 1.0);  glTranslated(-25.0, -25.0, 0);  glRecti(20, 20, 30, 30);  glFlush();  }  void myinit() {  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluOrtho2D(0.0,50.0,0.0,50.0);  glMatrixMode(GL\_MODELVIEW);  glClearColor(1.0,1.0,1.0,1.0);  glColor3f(0.0,0.0,0.0);  }  int main(int argc, char\* argv[]) {  glutInit(&argc,argv);  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  glutInitWindowSize(400,400);  glutInitWindowPosition(100,100);  glutCreateWindow("Transform");  glutDisplayFunc(display);  myinit();  glutMainLoop();  return 0;  } |

**Program 5.1 Rotasi terhadap sembarang titik**

Yang perlu pula diperhatikan di topik transformasi adalah Current Transformation Matrix (CTM). Perintah OpenGL yang berkaitan dengan CTM ini adalah: glPushMatrix(), glPopMatrix(), dan glLoadIdentity(). Berikut keterangan terkait ketiga perintah tersebut:

1. glPushMatrix() menduplikasi CTM dan meletakkannya di bagian teratas stack
2. glPopMatrix() pop matrix teratas dari stack
3. glLoadIdentity mengubah matrix teratas stack menjadi matrix identitas

Contoh :

1. Inisialisasi stack

glMatrix(GL\_MODEL\_VIEW);

I

glLoadIdentity();

1. Scale by 2

S(2)

glScaled(1.0, 2.0, 1.0);

1. Rotate terhadap x by 90 derajat

S(2).R(90)

glRotated(90, 1, 0, 0);

Push stack

S(2).R(90)

1. glPushMatrix();

S(2).R(90)

1. Translate by a

S(2).R(90)Tr(a)

glTranslatef(a.x, a.y, a.z);

S(2).R(90)

1. Menggambar point pada layar

glBegin(GL\_POINTS);

S(2).R(90)Tr(a)

glVertex3f(1,1,1);

S(2).R(90)

glEnd();

- proses perkalian matriksnya: S(2) \* R(90) \* Tr(a) \* (1,1,1)

1. Pop off stack

S(2).R(90)

glPopMatrix();

Stack transformasi ini berguna kalau kita ingin membuat hirarki transformasi. Contoh: kita hendak memodelkan tangan yang terdiri dari lengan atas, lengan bawah, telapak tangan, jari-jari. Untuk keseluruhan tangan, transformasinya kita simpan di stack nomor 2 dari bawah. Untuk telapak tangan dan jari-jari, transformasinya kita simpan di stack nomor 3 dari bawah. Sedangkan untuk jari-jari, transformasinya kita simpan di stack teratas. Dengan demikian kita jadi lebih fleksibel untuk mentrasformasi tiap bagian obyek.

Program 5.2 menunjukkan contoh penggunaan glPushMatrix() dan glPopMatrix().

|  |
| --- |
| void display() {  glClear(GL\_COLOR\_BUFFER\_BIT);  glColor3f(0.0,0.0,1.0);  glLoadIdentity();  glPointSize(6);  //Point pertama  glBegin(GL\_POINTS);  glVertex2f(15,15);  glEnd();  glPushMatrix();  glScalef(2,1,1);  //Point pertama yang di Scale  glBegin(GL\_POINTS);  glVertex2f(15,15);  glEnd();  //Point kedua yang sudah di Scale  glColor3f(0.0,1.0,0.0);  glBegin(GL\_POINTS);  glVertex2f(10,25);  glEnd();  glPopMatrix();  //Point kedua tanpa Scale  glBegin(GL\_POINTS);  glVertex2f(10,25);  glEnd();  glRotatef(10, 0, 0, 1);  //Point pertama dengan rotate  glColor3f(1.0,0.0,0.0);  glBegin(GL\_POINTS);  glVertex2f(15,15);  glEnd();  glPushMatrix();  glTranslatef(5,0,0);  //Point pertama dengan rotate dulu dan setelah itu di translate  glBegin(GL\_POINTS);  glVertex2f(15,15);  glEnd();  glFlush();  }  void myinit() {  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluOrtho2D(0.0,50.0,0.0,50.0);  glMatrixMode(GL\_MODELVIEW);  glClearColor(1.0,1.0,1.0,1.0);  glColor3f(0.0,0.0,0.0);  }  int main(int argc, char\* argv[]) {  glutInit(&argc,argv);  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  glutInitWindowSize(400,400);  glutInitWindowPosition(100,100);  glutCreateWindow("Transform");  glutDisplayFunc(display);  myinit();  glutMainLoop();  return 0;  } |

**Program 5.2 Contoh penggunaan glPushMatrix() dan glPopMatrix()**

.

1. **Percobaan**
2. Untuk program 5.1, modifikasi parameter fungsi transformasi, lalu amati perubahan tampilannya. Amati juga efek urutan pemanggilan fungsi transformasi.
3. Berdasarkan program 5.1, buat sebuah program yang melakukan Scaling kotak kedua terhadap titik tengah kotak kedua, bukan terhadap titik pusat koordinat.
4. Untuk program 5.2, modifikasi transformasi dan push/pop matrix, lalu amati perubahan tampilannya.
5. Untuk program 5.2, hapus semua glPushMatrix() dan glPopMatrix, ganti dengan glLoadIdentity() sedemikian rupa sehingga tampilan program tetap sama. Posisi vertex-vertexnya harus tetap, sedangkan nilai transformasi boleh diubah.

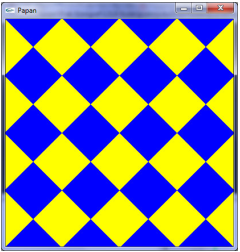
**BAB II**

**LAPORAN TUGAS**

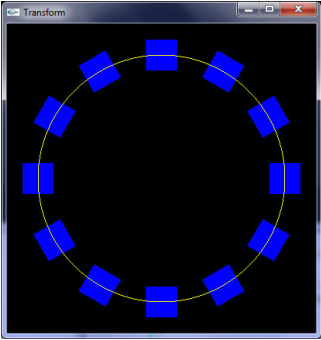
* 1. **TUGAS**

Tugas modul 5 sama dengan tugas modul 4. Bedanya, gunakan glPushMatrix() dan glPopMatrix:

1. Buat checker board (papan catur) miring sebagai berikut dengan menggunakan Transformasi.

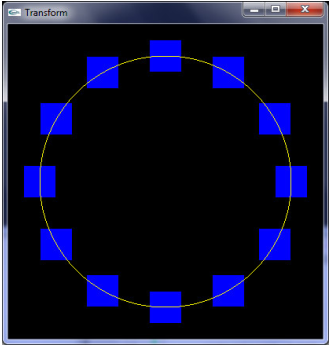


1. Buat kotak berputar berikut menggunakan transformasi.



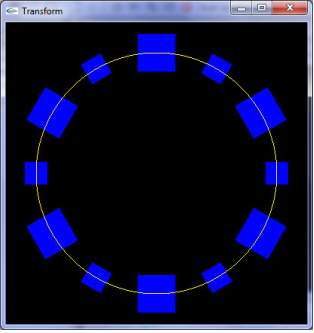
Gambar tidak perlu persis, asal cukup mirip.

1. Buat kotak berputar berikut menggunakan transformasi.



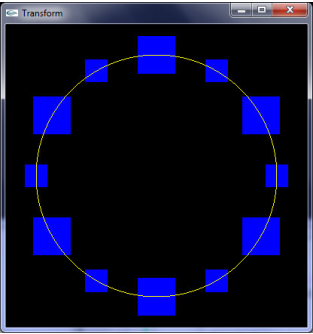
Gambar tidak harus persis, asal cukup mirip.

1. Buat variasi kotak berputar berikut menggunakan transformasi.



Gambar tidak harus persis, asal cukup mirip

1. Buat variasi kotak berputar beerikut menggunakan transformasi.



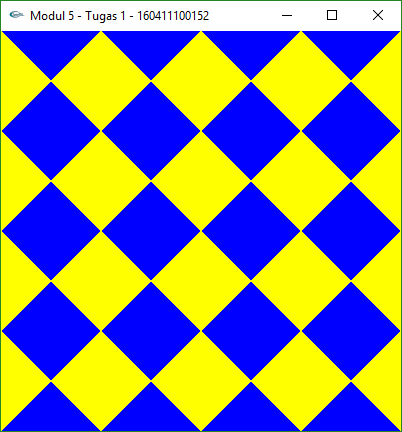
Gambar tidak harus persis, asal cukup mirip.

* 1. **PENYELESAIAN**

1. Membuat papan catur miring Menggunakan Transformasi

|  |
| --- |
| #include <windows.h>  #ifdef \_\_APPLE\_\_  #include <GLUT/glut.h>  #else  #include <GL/glut.h>  #endif  #include <stdlib.h>  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT);  glColor3d(1,1,0);  for (int x=0; x<4; x++){  if (x==0) {  glTranslated(0,0,0);  } else {  glTranslated(0,-14,0);  }  glPushMatrix();  for (int i=0; i<5; i++) {  if (i==0) {  glTranslated(0,0,0);  } else if (0>i<4){  glTranslated(14,0,0);  }  glPushMatrix();  glRotated(45,0,0,1);  glRectd(-5,-5,5,5);  glPopMatrix();  }  glPopMatrix();  }  glFlush();  }  void myinit() {  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluOrtho2D(-0.0, 56.0, -49.0, 7.0);  glMatrixMode(GL\_MODELVIEW);  glClearColor(0.0,0.0,1.0,1.0);  }  int main(int argc, char\* argv[]) {  glutInit(&argc,argv);  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  glutInitWindowSize(400,400);  glutInitWindowPosition(100,100);  glutCreateWindow("Modul 5 - Tugas 1 - 160411100152");  glutDisplayFunc(display);  myinit();  glutMainLoop();  return 0;  } |

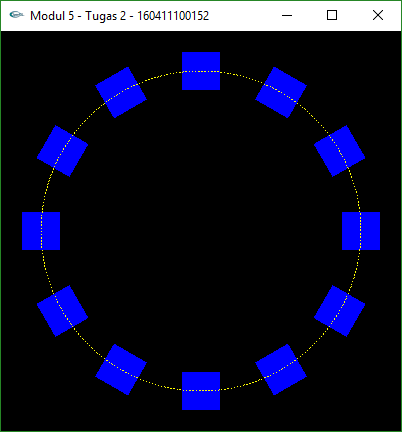
**Hasil Running**



1. Membuat Kotak berputar sesuai gambar soal 2

|  |
| --- |
| #include <windows.h>  #ifdef \_\_APPLE\_\_  #include <GLUT/glut.h>  #else  #include <GL/glut.h>  #endif  #include <stdlib.h>  #include <math.h>  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT);  glColor3d(0,0,1);  int derajat = 30;  for (int i=0; i<12; i++) {  glPushMatrix();  glRotated(derajat, 0, 0, 1);  glTranslated(0, 40, 0);  glScaled(0.95, 0.95, 0);  glRectd(-5, -5, 5, 5);  glPopMatrix();  derajat = derajat + 30;  }  glBegin(GL\_LINES);  glColor3f (1, 1.0, 0.0);  for(float i = 0; i<=3.14\*2; i+=0.01){  glVertex2f(40\*cos(i),40\*sin(i));  }  glEnd();  glFlush();  }  void myinit() {  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluOrtho2D(-50.0, 50.0, -50.0, 50.0);  glMatrixMode(GL\_MODELVIEW);  glClearColor(0.0,0.0,0.0,1.0);  }  int main(int argc, char\* argv[]) {  glutInit(&argc,argv);  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  glutInitWindowSize(400,400);  glutInitWindowPosition(100,100);  glutCreateWindow("Modul 5 - Tugas 2 - 160411100152");  glutDisplayFunc(display);  myinit();  glutMainLoop();  return 0;  } |

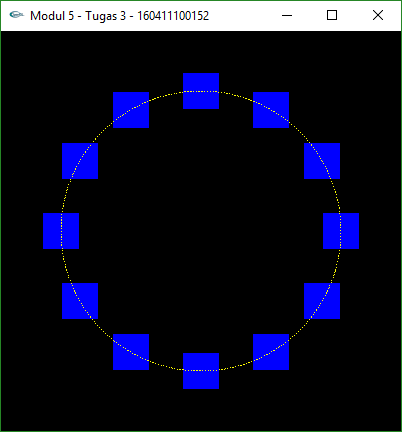
**Hasil Running**



1. Membuat kotak beputar sesuai gambar soal 3

|  |
| --- |
| #include <windows.h>  #ifdef \_\_APPLE\_\_  #include <GLUT/glut.h>  #else  #include <GL/glut.h>  #endif  #include <stdlib.h>  #include <math.h>  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT);  for (float i=0; i<3.14\*2; i+=6.28/12 ){  glColor3d(0,0,1);  glPushMatrix();  glTranslated(cos(i)\*3.5, sin(i)\*3.5, 0);  glScaled(0.9, 0.9, 0);  glRectf(-0.5, -0.5, 0.5, 0.5);  glPopMatrix();  }  glBegin(GL\_LINES);  glColor3d(1,1,0);  for (float i=0; i<3.14\*2; i+=0.01){  glVertex2f(cos(i)\*3.5, sin(i)\*3.5);  }  glEnd();  glFlush ();  }  void myinit() {  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluOrtho2D(-5.0, 5.0, -5.0, 5.0);  glMatrixMode(GL\_MODELVIEW);  glClearColor(0.0,0.0,0.0,1.0);  }  int main(int argc, char\* argv[]) {  glutInit(&argc,argv);  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  glutInitWindowSize(400,400);  glutInitWindowPosition(100,100);  glutCreateWindow("Modul 5 - Tugas 3 - 160411100152");  glutDisplayFunc(display);  myinit();  glutMainLoop();  return 0;  } |

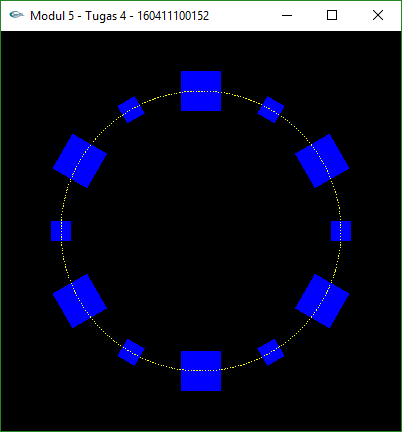
**Hasil Running**



1. Membuat kotak beputar sesuai gambar soal 4

|  |
| --- |
| #include <windows.h>  #ifdef \_\_APPLE\_\_  #include <GLUT/glut.h>  #else  #include <GL/glut.h>  #endif  #include <stdlib.h>  #include <math.h>  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT);  int hitung = 0;  for (float i=0; i<3.14\*2; i+=(3.14\*2)/12) {  glColor3d(0,0,1);  glPushMatrix();  glRotated(i\*360/6.28, 0, 0, 1);  if (hitung%2 == 0) {  glScaled(1, 1, 1);  glTranslated(0, 3.5, 0);  } else {  glScaled(0.5, 0.5, 1);  glTranslated(0, 7, 0);  }  glRectf(-0.5,-0.5,0.5,0.5);  glPopMatrix();  hitung++;  }  glBegin(GL\_LINES);  glColor3d(1,1,0);  for (float i=0; i<3.14\*2; i+=0.01){  glVertex2f(cos(i)\*3.5, sin(i)\*3.5);  }  glEnd();  glFlush ();  }  void myinit() {  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluOrtho2D(-5.0, 5.0, -5.0, 5.0);  glMatrixMode(GL\_MODELVIEW);  glClearColor(0.0,0.0,0.0,1.0);  glColor3d(1,1,1);  }  int main(int argc, char\* argv[]) {  glutInit(&argc,argv);  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  glutInitWindowSize(400,400);  glutInitWindowPosition(100,100);  glutCreateWindow("Modul 5 - Tugas 4 - 160411100152");  glutDisplayFunc(display);  myinit();  glutMainLoop();  return 0;  } |

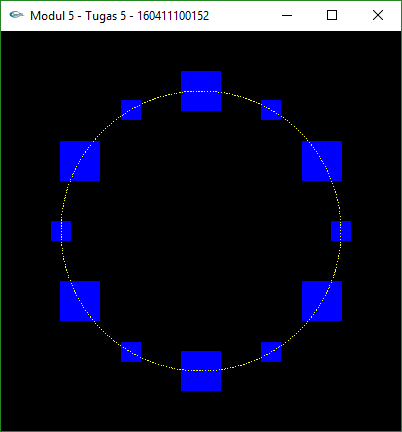
**Hasil Running**



1. Membuat kotak beputar sesuai gambar soal 5

|  |
| --- |
| #include <windows.h>  #ifdef \_\_APPLE\_\_  #include <GLUT/glut.h>  #else  #include <GL/glut.h>  #endif  #include <stdlib.h>  #include <math.h>  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT);  int hitung = 0;  for (float i=0; i<3.14\*2; i+= 6.28/12) {  glColor3d(0,0,1);  glPushMatrix();  glTranslated(3.5\*cos(i), 3.5\*sin(i), 0);  if (hitung%2==1) {  glScaled(1,1,1);  } else {  glScaled(0.5, 0.5, 1);  }  glRectf(-0.5,-0.5,0.5,0.5);  glPopMatrix();  hitung++;  }  glBegin(GL\_LINES);  glColor3d(1,1,0);  for (float i=0; i<3.14\*2; i+=0.01){  glVertex2f(cos(i)\*3.5, sin(i)\*3.5);  }  glEnd();  glFlush ();  }  void myinit() {  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluOrtho2D(-5.0, 5.0, -5.0, 5.0);  glMatrixMode(GL\_MODELVIEW);  glClearColor(0.0,0.0,0.0,1.0);  }  int main(int argc, char\* argv[]) {  glutInit(&argc,argv);  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  glutInitWindowSize(400,400);  glutInitWindowPosition(100,100);  glutCreateWindow("Modul 5 - Tugas 5 - 160411100152");  glutDisplayFunc(display);  myinit();  glutMainLoop();  return 0;  } |

**Hasil Running**



1. **PENJELASAN PROGRAM**
2. Program pertama adalah membuat papan catur miring diagonal variasi warna biru dan kuning dengan memanfaatkan rotasi dan translasi. Pembuatan papan catur kali ini kami menggunakan dua perulangan. Perulangan pertama untuk melakukan duplikasi kotak secara diagonal ke kanan hingga 5 kali. Didalam perulangan pertama kami membuat sebuah kotak yang kemudian di rotate 45 derajat pada sumbu z, kemudian kami translasikan ke sumbu x dengan nilai 0 untuk perulangan pertama dan 14 untuk perulangan kedua dan seterusnya.

Setelah tercipta 5 kotak diagonal ke kanan. Kami menggunakan perulangan kedua untuk menduplikasi kelima kotak tadi ke bawah sebanyak 4 kali. Maka disini akan tercipta 20 kotak. Setelah itu supaya papan catur terlihat rapi dan sama seperti di gambar soal, kami memainkan ukuran ortho dengan cara memotong/mengecilkan ortho kiri-kanan-bawah-atas dengan rincian 0.0, -56, 49, 7. Jadi posisi papan catur berada pada kuadran IV.

1. Program kedua memanfaatkan fungsi rotate dan translate. Pertama-tama kami membuat kotak di tengah-tengah sumbu koordinat menggunakan glRect(-5, 5, 5, 5);. Kemudian kotak yang telah dibuat tadi ditranslasikan pada sumbu y ke posisi 40 yaitu keatas, setelah itu kami melakukan rotasi sebesar 30 derajat. Kami melakukan perulangan rotasi dan translasi sebanyak 12 kali sebesar 30 derajat tiap kotak hingga terbentuk 12 kotak yang berjejer melengkung searah jarum jam.

Ada masalah ketika kami melakukan perulangan transformasi, objek yang telah dibuat tadi berubah menjadi tidak teratur, namun dengan adanya perintah glPushMatrix() dan glPopMatrix(), masalah tersebut dapat teratasi

1. Program ketiga mirip seperti soal 2 namun jika dalam soal 2 beberapa kotak menjadi miring akibat dirotasi, pada program ini kami harus membuatnya tetap tegak. Pembuatan soal 3 ini dengan cara membuat satu kotak posisi di tengah-tengah sumbu x,y kemudian kami translasi dengan memanfaatkan rumus lingkaran . Perulangan dilakukan sebanyak 12 kali untuk menciptakan kotak sebanyak 12 kali juga.
2. Program mirip seperti soal 2 juga, namun tiap jeda satu kotak, kotak harus dibuat lebih kecil dengan urutan besar-kecil dengan jumlah 12 kotak. Pembuatan soal 4 ini dengan cara membuat satu kotak di tengah sumbu x y. Kemudian dirotasi 12 kali searah jarum jam dengan memanfaatkan rumus lingkaran . Disini kami menggunakan counter untuk mengecek apakah kotak ingin diperkecil atau diperbesar. Jika hitungan ganjil maka kotak diperbesar (sesuai ukuran asli saat pembuatan kotak pertama di tengah). Jika hitungan genap maka kotak akan diperkecil setengah kali nya. Jadi nantinya kotak yang tercipta akan terlihat besar-kecil dan terus terulang hingga tercipta 12 kotak.
3. Program kelima sama seperti soal 4, di rotasi dengan ukuran besar-kecil, namun kita harus membuat kotaknya untuk tegak. Pembuatan soal 5 ini dimulai dengan membuat kotak di tengah-tengah sumbu xy kemudian kam translasi dengan memanfaatkan rumus lingkaran . Selain itu untuk membuat kotak menjadi kecil dan besar kami menggunakan counter, jika dia ganjil maka kami buat ukurannya menjadi besar, jika hitungan genap maka kami buat kotaknya kecil.

**BAB III**

**PENUTUP**

* 1. **KESIMPULAN**

Dengan menggunakan OpenGL, kita dapat menggunakan transformasi geometri, kita dapat mentranslasi obyek, merotasi dan memperbesar/memperkecil ukuran obyek. Dengan memanfaatkan transformasi ini, kita dapat dengan mudah menggambar tanpa harus repot menentukan koordinat. glPushMatrix() dan glPopMatrix() juga sangat membantu untuk menggambar objek.

* 1. **SARAN**

Pemahaman tentang kombinasi transformasi akan sangat membantu menggambar obyek, karena pengaplikasinnya sedikit membingungkan. Selain itu contoh-contoh referensi yang lebih banyak bagi mahasiswa juga sangat membantu untuk mengimplementasikan fungsi-fungsi-fungsi dasar OpenGL dengan baik.