**LAPORAN TUGAS BESAR 2**

**IF2123 - ALJABAR GEOMETRI**

**“Simulasi Transformasi Linier pada Bidang 2D Dengan Menggunakan OpenGL API”**



disusun oleh:

**K03-Teknik Informatika 2016**

Michelle Eliza Gananjaya 13516015

Erma Safira Nurmasyita 13516072

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG**

**BAB I**

**DESKRIPSI MASALAH**

Pada tugas kali ini, mahasiswa diminta membuat program yang mensimulasikan transformasi linier untuk melakukan operasi translasi, refleksi, dilatasi, rotasi, dan sebagainya pada sebuah bidang 2D. Bidang dibuat dengan mendefinisikan sekumpulan titik sudut lalu membuat bidang dari titik-titik tersebut.  
 Program akan memiliki dua buah window, window pertama (command prompt) berfungsi untuk menerima input dari user, sedangkan window kedua (GUI) berfungsi untuk menampilkan output berdasarkan input dari user. Kedua window ini muncul ketika user membuka file executable.

Saat program baru mulai dijalankan, program akan menerima input N, yaitu jumlah titik yang akan diterima. Berikutnya, program akan menerima input N buah titik tersebut (pasangan nilai x dan y). Setelah itu program akan menampilkan output sebuah bidang yang dibangkitkan dari titik-titik tersebut. Selain itu juga ditampilkan dua buah garis, yaitu sumbu x dan sumbu y. Nilai x dan y memiliki rentang minimal - 500 pixel dan maksimum 500 pixel. Pastikan window GUI yang Anda buat memiliki ukuran yang cukup untuk menampilkan kedua sumbu dari ujung ke ujung.  
 Berikutnya, program dapat menerima input yang didefinisikan pada tabel dibawah.

|  |  |
| --- | --- |
| **Input** | **Keterangan** |
| translate <dx> <dy> | Melakukan translasi objek dengan menggeser nilai x sebesar *dx* dan menggeser nilai y sebesar *dy*. |
| dilate <k> | Melakukan dilatasi objek dengan faktor scaling *k*. |
| rotate <deg> <a> <b> | Melakukan rotasi objek secara berlawanan arah jarum jam sebesar *deg* derajat terhadap titik *a,b* |
| reflect <param> | Melakukan pencerminan objek. Nilai *param* adalah salah satu dari nilainilai berikut: **x, y, y=x, y=-x, atau (a,b)**. Nilai (a,b) adalah titik untuk melakukan pencerminan terhadap. |
| shear <param> <k> | Melakukan operasi *shea*r pada objek. Nilai *param* dapat berupa x (terhadap sumbu x) atau y (terhadap sumbu y). Nilai *k* adalah faktor *shear*. |
| stretch <param> <k> | Melakukan operasi *stretch* pada objek. Nilai *param* dapat berupa x (terhadap sumbu x) atau y (terhadap sumbu y). Nilai *k* adalah faktor *stretch*. |
| custom <a> <b> <c> <d> | Melakukan transformasi linier pada objek dengan matriks transformasi sebagai berikut:  a b  c d |
| multiple <n>  … // input 1  … // input 2  …  … // input n | Melakukan transformasi linier pada objek sebanyak n kali berurutan. Setiap baris input 1..n dapat berupa *translate, rotate, shear*, dll tetapi bukan *multiple, reset, exit*. |
| reset | Mengembalikan objek pada kondisi awal objek didefinisikan. |
| exit | Keluar dari program. |

**BAB II  
TEORI SINGKAT**

1. **Transformasi Linier**

Transformasi linier adalah fungsi yang memetakan struktur linier dari ruang vektor satu ke ruang vektor lainnya. *T : V1 → V2* adalah transformasi linier jika:

(i) T(**u** + **v**) = T**u** + T**v** untuk semua vektor **u, v** pada ruang *V1*;

(ii) T(c**u**) = cT**u** untuk semua vektor **u** pada ruang *V1* dan besaran skalar c.

Transformasi linier *T : V1 → V2* dari ruang vektor *V1* ber-dimensi *m* ke ruang vektor *V2* ber-dimensi *n* dinyatakan dalam matriks *n x m*. Namun diperlukan pemilihan basis untuk *V1* dan *V2*, sedangkan transformasi linier tidak bergantung oleh basis. Transformasi digunakan untuk untuk memindahkan suatu titik atau bangun pada suatu bidang.

Terdapat beberapa jenis-jenis transformasi, yaitu:

1. Translasi

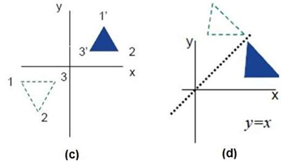
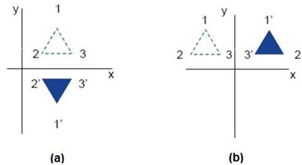
Translasi memindahkan objek ke posisi berbeda. Suatu titik dapat ditranslasikan dalam 2 dimensi dengan menambahkan koordinat translasi (tx, ty) kepada koordinat original (X, Y) untuk mendapatkan koordinat baru (X’, Y’).

1. Dilatasi

Dilatasi pada transformasi geometri mengakibatkan ukuran benda berubah, tetapi bentuknya tetap. Faktor yang menyebabkan perubahan ukuran benda disebut faktor dilatasi atau faktor pengali.

1. Pencerminan

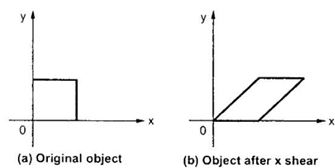
Transformasi yang memindahkan titik-titik dengan menggunakan sifat bayangan oleh suatu cermin.

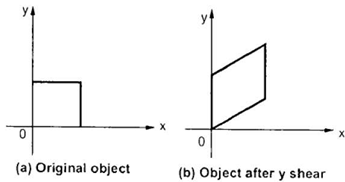


*(* *Pencerminan terhadap: (a)sumbu x; (b)sumbu y; (c)titik (0,0); (d)garis y=x )*

1. *Shear*

Transformasi *shear* adalah deformasi objek di mana suatu bidang sejajar tetap sejajar tetapi digeser sejauh faktor *shear* ke arah yang sejajar dengan dirinya sendiri.

****



1. *Stretch*

*Stretch* atau dalam Bahasa Indonesia disebut regangan adalah bentuk transformasi dengan menarik sebuah benda searah sumbu x atau sumbu y dengan skala tertentu.

1. *Custom*

Operasi *custom* melakukan transformasi linier pada objek dengan matriks transformasi sebagai berikut:

1. **Matriks Transformasi**

Matriks transformasi adalah matriks yang membuat sebuah obyek grafik mengalami perubahan baik berupa perubahan posisi maupun perubahan ukuran daripada obyek grafik tersebut. Untuk dua dimensi, matriks transformasi dinyatakan dalam ukuran 3 x 3 dengan kolom ke 3 dipakai sebagai tempat penyedia untuk proses transformasi (translasi, dilatasi, atau rotasi).

1. Translasi

Perpindahan titik-titik pada bidang dengan jarak dan arah tertentu yang diwakili oleh ruas garis berarah (vector)  atau dengan suatu pasangan bilangan .

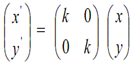
Translasi  memetakan titik P(x1 ,y1 ) ke titik P '( x1 + a, y1 + b ) yang dinotasikan dengan:



1. Dilatasi

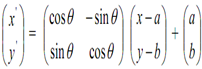
Transformasi yang mengubah jarak titik-titik dengan factor pengali tertentu terhadap suatu titik tertentu. Perkalian atau dilatasi ini ditentukan oleh factor skala (k) dan pusat dilatasi. Dilatasi terhadap titik pusat O(0,0) pemetaannya adalah: [O, k] : P(x,y) → P '(kx, ky).

Persamaan matriksnya:



1. Rotasi

Rotasi terhadap titik pusat P(a, b) (dilambangkan dengan R(O, θ ). Jika suatu titik P (x,y) diputar sejauh θ berlawanan dengan arah jam terhadap titik pusat A(a,b) maka bayangannya adalah P ' (x ', y ') dengan x ' - a = (x –a) cosθ - (y-b) sinθ y '- b = (x – a) sin θ + (y- b) cos θ. Persamaan matriknya:



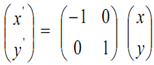
1. Pencerminan

Pencerminan terhadap sumbu X (dilambangkan dengan Mx) adalah Mx : P(x,y) → P '(x ', y ') = P '(x, -y)   
Persamaan matriksnya :



Pencerminan terhadap sumbu Y (dilambangkan dengan My) adalah My: P(x,y) → P '(x ', y ') = P '(-x, y)

Persamaan matriksnya :



Pencerminan terhadap titik asal O(0,0) (dilambangkan dengan M0) adalah M0: P(x,y) → P '(x ', y ') = P '(-x, -y)   
 Persamaan matriksnya :



Pencerminan terhadap garis y = x (dilambangkan dengan M(y=x)) adalah M(y=x): P(x,y) → P '(x ', y ') = P '(y, x)

Persamaan matriksnya :



Pencerminan terhadap garis y = -x (dilambangkan denganM(y=-x)) adalah M(y=-x): P(x,y) → P '(x ', y ') = P '(-y, -x)

Persamaan matriksnya :



Pencerminan terhadap titik (a,b) (dilambangkan dengan M(a,b)) adalah M(a,b): P(x,y) → P '(x ', y ') = P '( 2a-x, 2b - y)

1. *Shear*

Matriks transformasi dari perintah shear dengan konstanta k adalah:

Untuk parameter sumbu x:

1 k 0

0 1 0

0 0 1

Parameter sumbu y:

1 0 0

k 1 0

0 0 1

1. *Stretch*

Matriks transformasi dari perintah stretch dengan konstanta k adalah:

Untuk parameter sumbu x:

k 0 0

0 1 0

0 0 1

Parameter sumbu y:

1 0 0

0 k 0

0 0 1

1. *Custom*

Matriks transformasi dari perintah *custom* dengan input a, b, c, d adalah:

a b 0

c d 0

0 0 1

1. **OpenGL**

*Open Graphics Library* (OpenGL) adalah *aplication* *programming interface* (API) yang dapat dijalankan pada berbagai platform dan bahasa untuk menerjemahkan grafis vektor 2D dan 3D. OpenGL mulai dikembangkan pada tahun 1991. Aplikasi ini dipergunakan pada bidang grafika komputer, contohnya *computer-aided design* (CAD), teknologi *virtual reality*, visualisasi data dan informasi ilmiah, simulasi penerbangan, dan *video games*. Untuk melakukan proses *rendering* grafis, API ini berinteraksi dengan unit pemrosesan grafis (GPU) dan akselerasi *hardware*.

OpenGL terdefinisikan sebagai kumpulan fungsi dan konstanta dalam sebuah *library* yang dapat digunakan langsung oleh *client*. Walaupun definisi fungsi-fungsinya serupa dengan bahasa C, API ini tidak bergantung pada bahasa pemrograman. OpenGL memiliki keterikatan dengan berbagai bahasa, beberapa diantaranya adalah JavaScript terkait WebGL, C terkait WGL, GLX CGL, *binding* yang didukung oleh iOS dan Android.

OpenGL adalah API *open source* yang terus berkembang. Pengembangan spesifikasi OpenGL dirilis secara regular oleh Khronos Group, yang mendukung dan meningkatkanAPI dengan berbagai fitur. *Toolkit* versi yang pertama adalah OpenGL *Utility Toolkit* (GLUT), yang kemudian digantikan dengan *freeglut.* Kini dilanjutkan dengan *toolkit* GLFW.

**BAB III**

**IMPLEMENTASI PROGRAM DAN PEMBAGIAN TUGAS**

1. Implementasi Program

Pada pengerjaan program simulasi transformasi linier, kami menggunakan *toolkit* dari freeglut dan diimplementasikan dalam bahasa Python. Program kami terdiri dari dua modul, yaitu modul transformation dan main program. Pada modul transformation terdapat operasi matriks transformasi dan matriks awal. Sedangkan main program mengatur segala sesuatu yang ditampilkan pada jendela program, seperti pengaturan jendela, proses *rendering* grafis, input/output, hingga pilihan menu.

1. Transformation

Modul ini terdiri dari fungsi-fungsi pengolah matriks. Terdapat fungsi serbaguna yaitu KaliMatriks untuk mengalikan dua matriks. Untuk mentransformasikan bidang terdapat enam fungsi, yaitu fungsi translasi, dilatasi, rotasi, *shear, stretch,* dan *custom*. Matriks transformasi dari masing-masing jenis transformasi dibentuk di dalam masing-masing fungsi yang kemudian akan dikalikan dengan matriks koordinat titik bidang. Pada modul ini juga terdapat fungsi multmenu yang mengatur pilihan *multiple*.

1. Main Program

Pada bagian ini, terdapat pengaturan jendela OpenGL dan program *rendering* grafis. Program memunculkan dua jendela, yaitu untuk menerima input dari user dan untuk memunculkan gambar. Karena kedua jendela tersebut dieksekusi secara paralel, diperlukan pengaturan *multi-threading* antara fungsi pemuncul jendela dan fungsi *mainmenu* yang meng-*update* hasil transformasi. Pengaturan *multithreading* diperoleh dari *library* \_thread.

Input titik disimpan di dalam sebuah matriks yang kemudian dioperasikan dengan matriks transformasi dengan memanfaatkan modul transformation. Kemudian setiap perubahan disimpan dalam matriks penyimpan koordinat bidang yang kemudian ditampilkan pada jendela grafis.

Animasi pergerakan bidang diimplementasikan pada fungsi pengupdate matriks *points* dengan melakukan transformasi secara bertahap. Agar animasi transformasi terlihat, transformasi dijalankan dalam *loop* dengan primitif *timesleep*. Setiap tahap transformasi disimpan pada matriks *points* yang di-*render* program secara terus menerus dengan selang beberapa milisekon. Pengaturan waktu diperoleh dari *library* *time.*

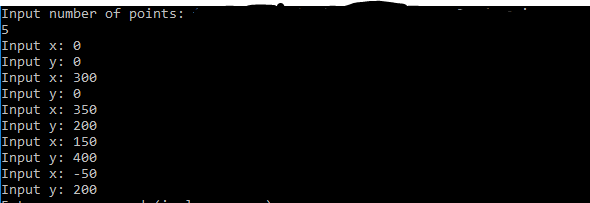
1. Pembagian Tugas

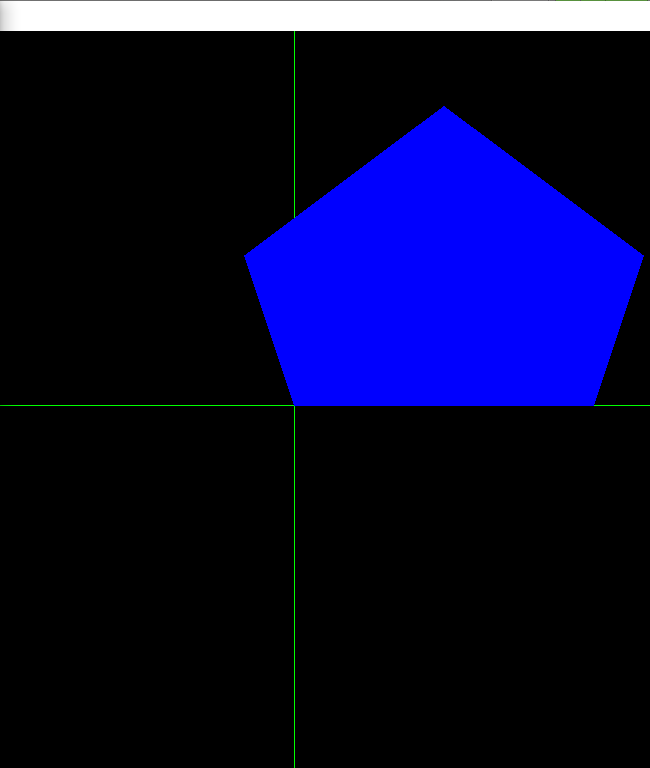
* Michelle Eliza G. (13516015): Program utama, fungsi multiple, laporan
* Erma Safira N. (13516072): Transformation, animasi, laporan

**BAB IV**

**EKSPERIMEN**

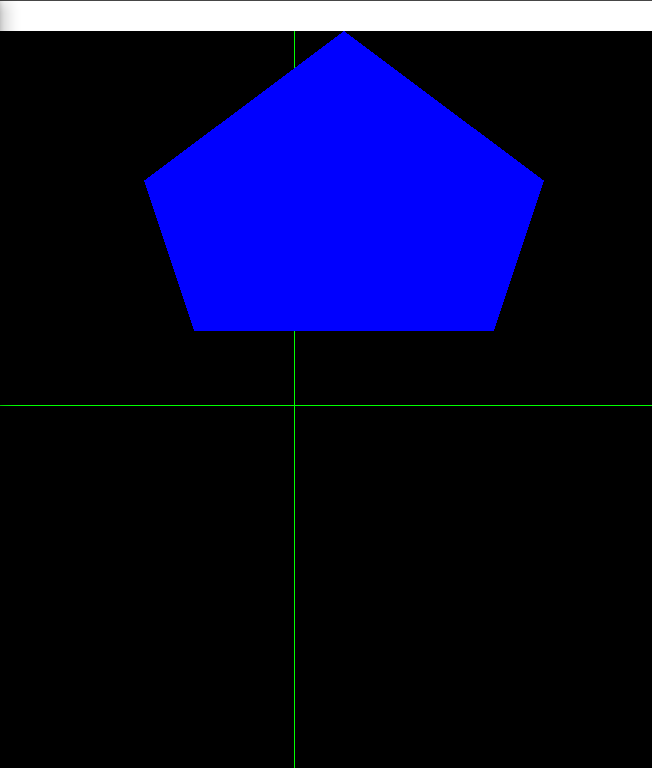
1. Menerima input titik dari pengguna:





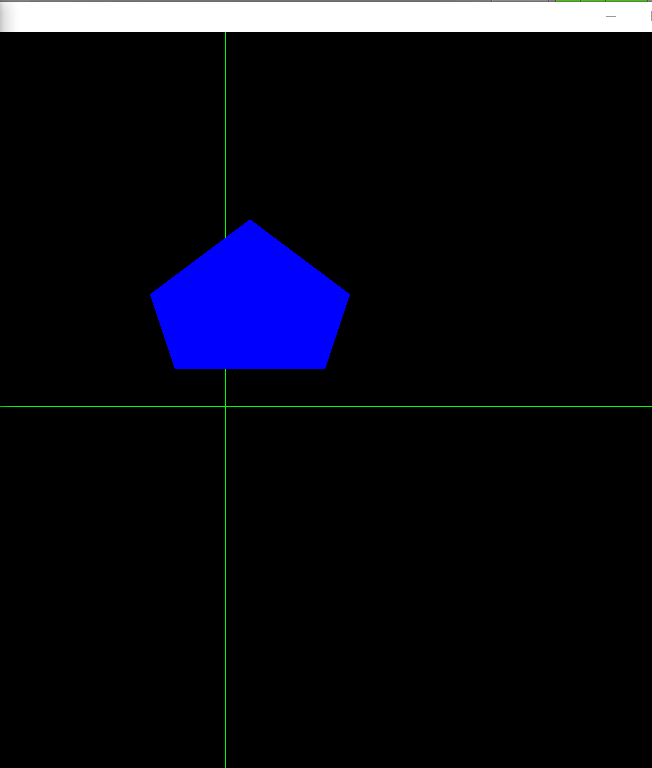
1. Test Case 1: translate

Command: translate -100 100



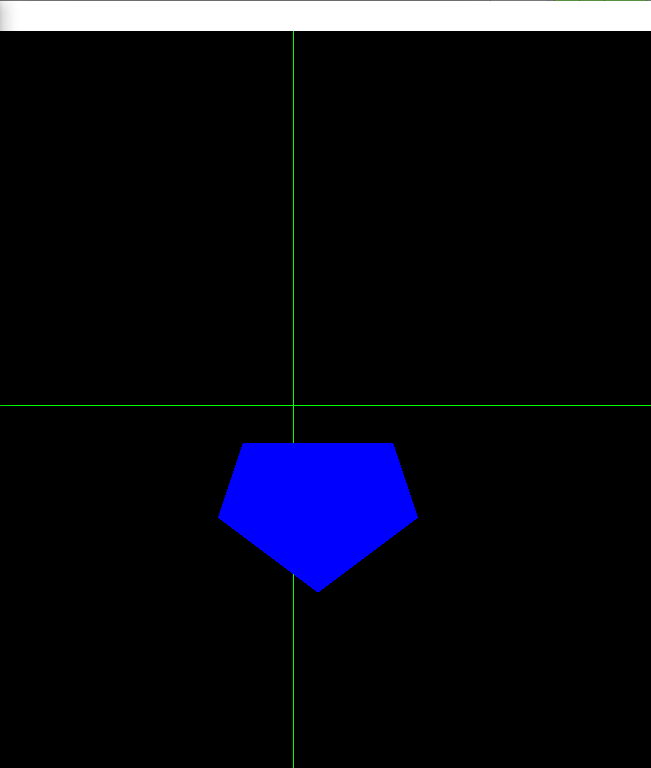
1. Test Case 2: dilate

Command: dilate 0.5



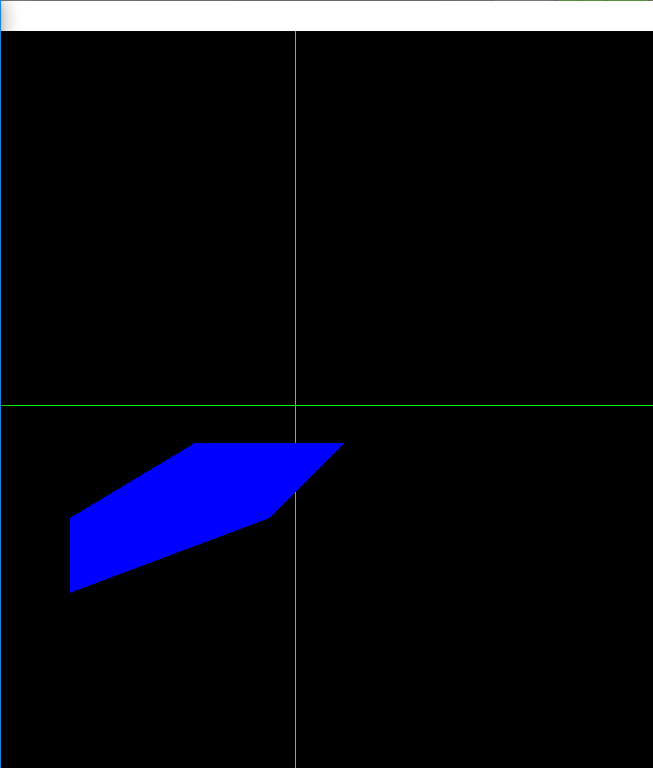
1. Test Case 3: reflect

Command: reflect x



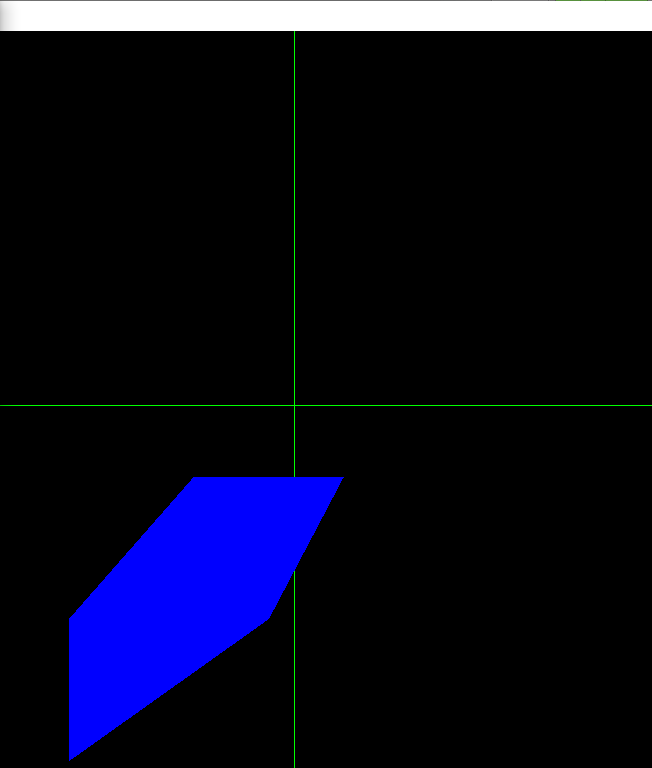
1. Test Case 4: shear

Command: shear x 1



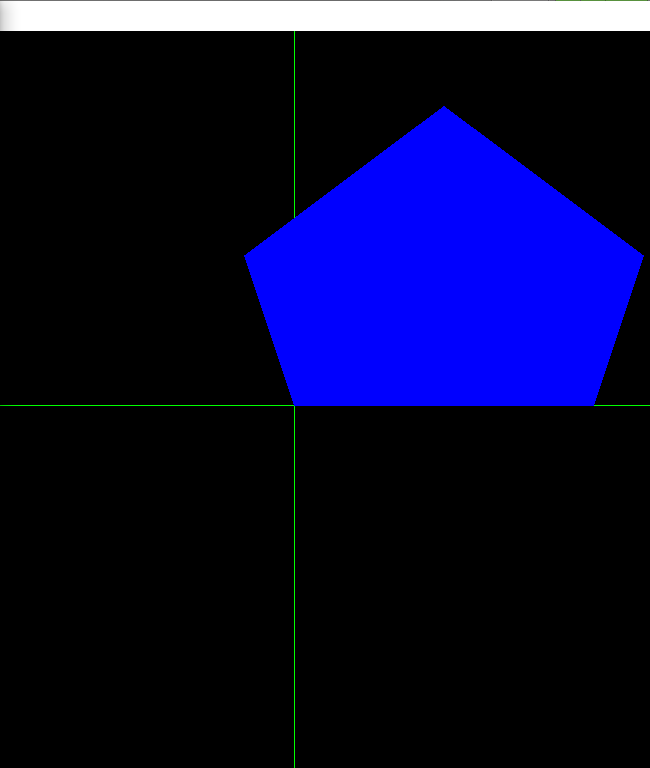
1. Test Case 5: stretch

Command: stretch y 2



1. Test Case 6: reset

Command: reset

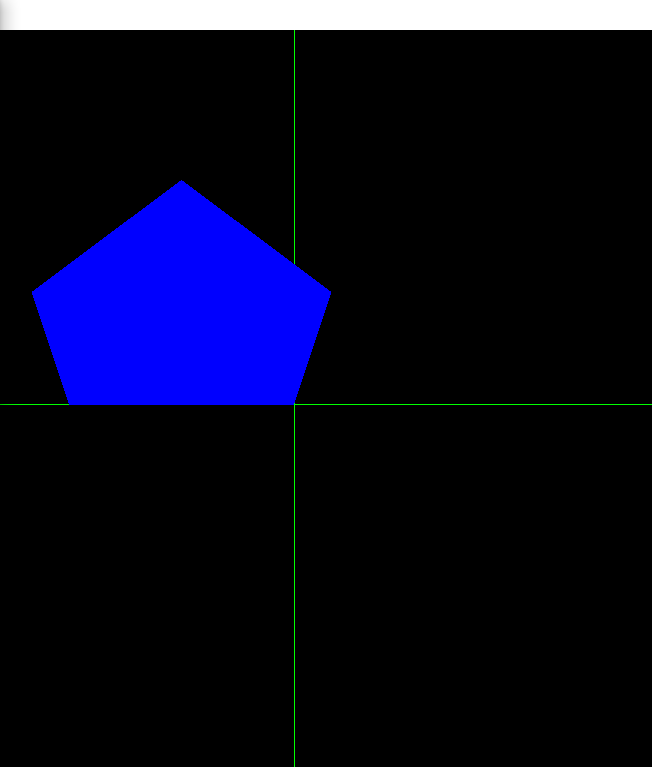


1. Test Case 7: multiple

Command: multiple 2

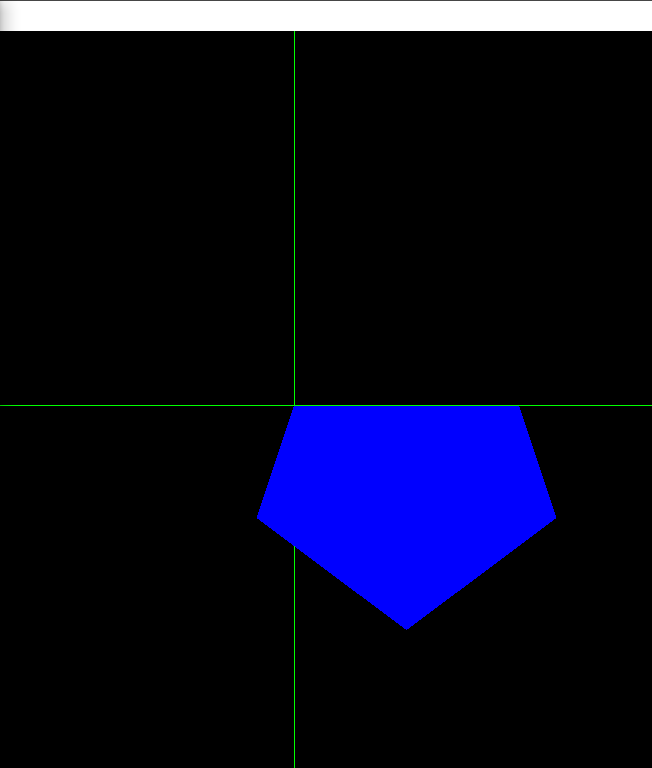
stretch x -1

dilate 0.75



1. Test Case 8: rotate

Command: rotate 180 0 0



**BAB V**

**KESIMPULAN DAN SARAN**

1. Kesimpulan

1.1 Untuk mengolah transformasi bidang digunakan matriks transformasi yang kemudian dikalikan dengan matriks koordinat asal.

1.2 Pemodelan grafis transformasi linier suatu bidang diaplikasikan dengan *toolkit* OpenGL.

1.3 Terdapat dua algoritma yang dijalankan secara paralel, yaitu program grafis dan program peng-*update* matriks koordinat titik-titik bidang. Untuk menjalankan keduanya digunakan *multithreading.*

1. Saran

Disarankan untuk menggunakan bahasa pemrograman Python agar lebih mudah dalam mengimplementasikan library OpenGL dan juga Multithreading.

1. Komentar/Refleksi

Dengan adanya tugas ini, kemampuan eksplorasi dan belajar mandiri kami diuji. Mempelajari hal-hal baru seperti OpenGL merupakan tantangan tersendiri di tengah padatnya kegiatan akademik. Proses belajar pun menjadi tidak monoton dari materi yang diajarkan saja. Dan setidaknya kehadiran tugas ini membuat mata kuliah Aljabar Geometri lebih **menyenangkan**. Hasil dari pengerjaan tugas ini, kami memperoleh *insight* baru yang lebih praktikal dalam bidang grafika komputer dan juga *multi-threading.*

**DAFTAR REFERENSI**

Anton, Howard dan Chris Rorres. 2015. *Elementary Linear Algebra - with Supplemental Applications*. Edisi ke-11. Singapore: John Wiley & Sons.

*Linear Transformations*. Brilliant.org. Diakses 7 November 2017, pada <https://brilliant.org/wiki/linear-transformations/>

*2D Transformation.* Diakses 8 November 2017, pada <https://www.tutorialspoint.com/computer_graphics/2d_transformation.htm>

OpenGL. Diakses 7 November 2017, pada <https://en.wikipedia.org/wiki/OpenGL>

*Introduction to C++ OpenGL Programming.* Diakses 7 November 2017, pada <https://www.cprogramming.com/tutorial/opengl_introduction.html>

*Python OpenGL Tutorial.* Diakses 8 November 2017, pada <https://noobtuts.com/python/opengl-introduction>

*3D Graphics with OpenGL Basic Theory.* Diakses 8 November 2017, pada <https://www.ntu.edu.sg/home/ehchua/programming/opengl/CG_BasicsTheory.html>