LAPORAN TUGAS BESAR II

IF2123 : ALJABAR GEOMETRI

SIMULASI TRANSFORMASI LINIER PADA BIDANG 2D DAN 3D DENGAN MENGGUNAKAN OPENGL API

DIBUAT OLEH :

ARIEL ANSA RAZUMARDI / 13517040

IGNATIUS TIMOTHY MANULANG / 13517044

VINSEN MARSELINO ANDREAS / 13517054

PROGRAM STUDI TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2018

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 objek **2D dan 3D**. Objek dibuat dengan mendefinisikan sekumpulan titik sudut lalu membuat bidang 2D/3D dari titik-titik tersebut. Contoh objek 2D: segitiga, segiempat, polygon segi-n, lingkaran, rumah, gedung, mobil, komputer, lemari, dsb. Contoh objek 3D: kubus, pyramid, silinder, terompet, dll.

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*.

Untuk objek 2D, 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 maksikum **500 pixel**. Pastikan window *GUI* yang Anda buat memiliki ukuran yang cukup untuk menampilkan kedua sumbu dari ujung ke ujung. Hal yang sama juga berlaku untuk objek 3D tetapi dengan tiga sumbu: x, y, dan z.

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 nilai-nilai berikut: **x**, **y**, **y=x**, **y=-x**, atau **(a,b)**. Nilai (a,b) adalah titik untuk melakukan pencerminan terhadap.
shear <param> <k>	Melakukan operasi *shear* 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: $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$
multiple <n> ... // input 1	Melakukan transformasi linier pada objek sebanyak *n* kali berurutan. Setiap baris input

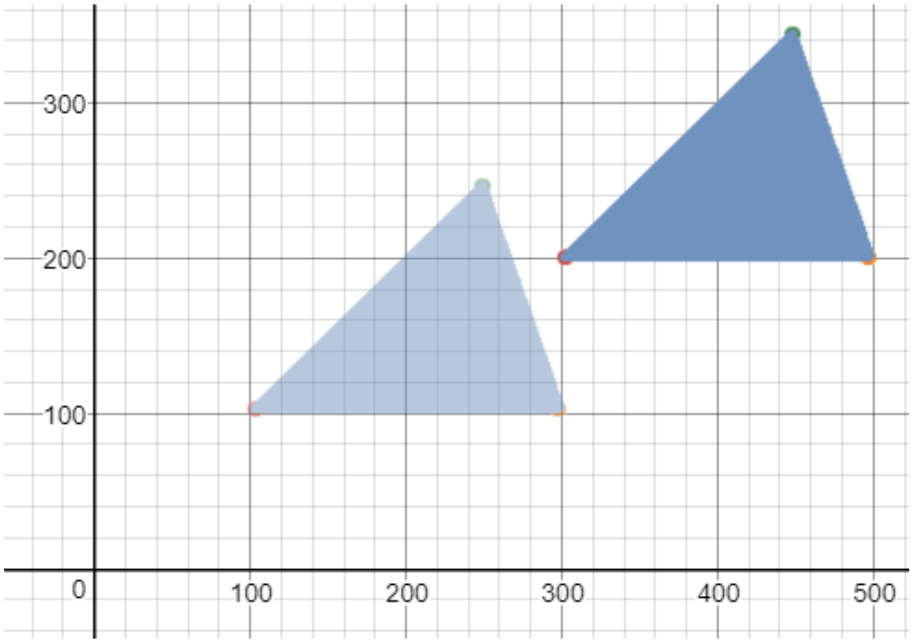
... // input 2 ... ... // input n	1..*n* dapat berupa *translate, rotate, shear, dll* tetapi bukan *multiple, reset, exit*.
reset	Mengembalikan objek pada kondisi awal objek didefinisikan.
exit	Keluar dari program.

BAB II. Teori Dasar

a. Transformasi linier adalah suatu proses pemetaan suatu titik terhadap suatu pengubah yang menghasilkan titik titik baru. Pengubah dalam kasus ini adalah dalam bentuk perkalian matriks dengan vektornya(titik). Hasil pemetaan dari transformasi ini ada bermacam-macam dan akan dijelaskan satu demi satu.

1. Translasi(translate)

   Transformasi jenis ini dilakukan dengan menggeser suatu titik ke titik selanjutnya. Parameter yang dipakai untuk transformasi jenis ini adalah banyaknya pergeseran yang diinginkan. Sebagai contoh, pada kasus dibawah, titik awal(100,100) digeser ke titik(300,200). Ini berarti translasi yang digunakan adalah (200,100).



Karena dalam masalah ini kami menggunakan transformasi matriks, maka matriks transformasi yang digunakan untuk 2 dimensi adalah

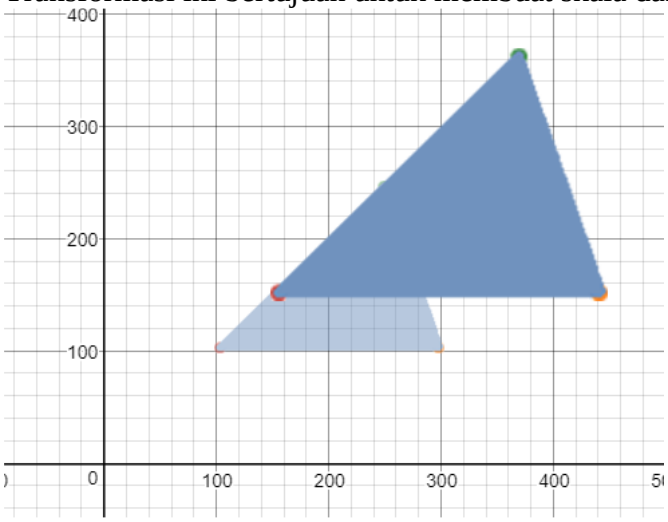
$$T(dx, dy) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix}$$

Sedangkan untuk 3 dimensi, matriks yang digunakan lebih kecil yaitu :

$$T(dx, dy, dz) = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Dilatasi(dilate)

Transformasi ini bertujuan untuk membuat skala dari objek yang sesungguhnya.



Dilatasi yang diimplementasikan adalah dilatasi dengan titik acuan (0,0,0). Matriks yang digunakan untuk dilatasi 2 dimensi adalah

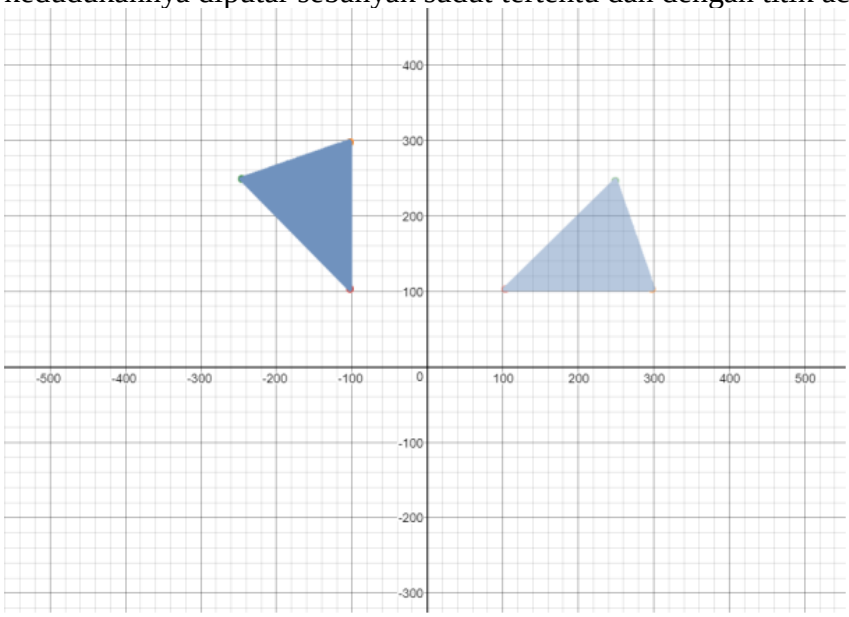
$$T(k) = \begin{bmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Sedangkan untuk dilatasi 3 dimensi

$$T(dx, dy, dz) = \begin{bmatrix} k & 0 & 0 & 0 \\ 0 & k & 0 & 0 \\ 0 & 0 & k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Rotasi(rotate)

Tujuan dari transformasi ini adalah memetakan suatu titik ke titik baru yang kedudukannya diputar sebanyak sudut tertentu dan dengan titik acuan tertentu,



Dalam hal ini, kesepakatannya adalah titik diputar berlawanan arah dengan jarum jam.

Matriks yang digunakan untuk transormasi ini dalam 2 dimensi adalah :

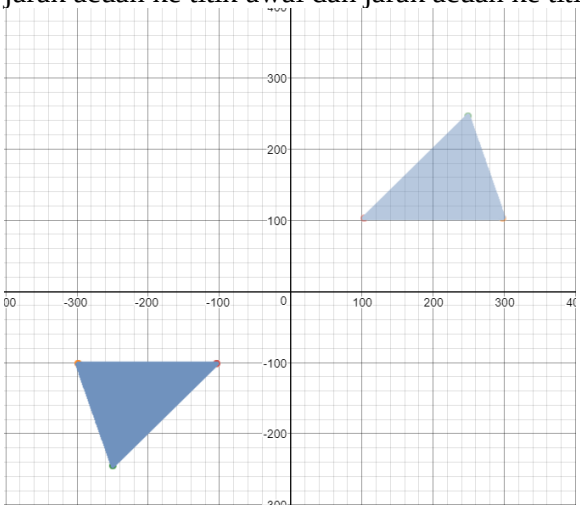
$$T(\theta, x, y) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & -x + x * \cos(\theta) - y * \sin(\theta) \\ \sin(\theta) & \cos(\theta) & -y + y * \cos(\theta) - x * \sin(\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

Sedangkan untuk rotasi 3 dimensi :

$$T(\theta, a, b, c) = \begin{bmatrix} \cos\theta + a^2(1-\cos\theta) & ab(1-\cos\theta) - c(\sin\theta) & ac(1-\cos\theta) + b(\sin\theta) & 0 \\ ab(1-\cos\theta) + c(\sin\theta) & \cos\theta + b(1-\cos\theta) & bc(1-\cos\theta) - a(\sin\theta) & 0 \\ ac(1-\cos\theta) - b(\sin\theta) & bc(1-\cos\theta) + a(\sin\theta) & \cos\theta + c^2(1-\cos\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. Reflect(refleksi)

Reflesi adalah transformasi yang memetakan suatu titik terhadap suatu acuan sehingga jarak acuan ke titik awal dan jarak acuan ke titik akhir adalah sama.



Titik acuan dalam refleksi ini ada bermacam-macam. Untuk 2 dimensi :

-Terhadap sumbu x

$$T(x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

-Terhadap sumbu y

$$T(y) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

-Terhadap garis y=x

$$T(y = x) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

-Terhadap garis y=-x

$$T(y = -x) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

-Terhadap titik (a,b)

$$T(a,b) = \begin{bmatrix} -1 & 0 & -2*a \\ 0 & -1 & -2*b \\ 0 & 0 & 1 \end{bmatrix}$$

Sedangakan untuk 3 dimensi

-Terhadap sumbu x

$$T(x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

-Terhadap sumbu y

$$T(y) = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

-Terhadap sumbu z

$$T(z) = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

-Terhadap bidang xy

$$T(xy) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

-Terhadap bidang xz

$$T(xz) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

-Terhadap bidang yz

$$T(yz) = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

-Terhadap titik(a,b,c)

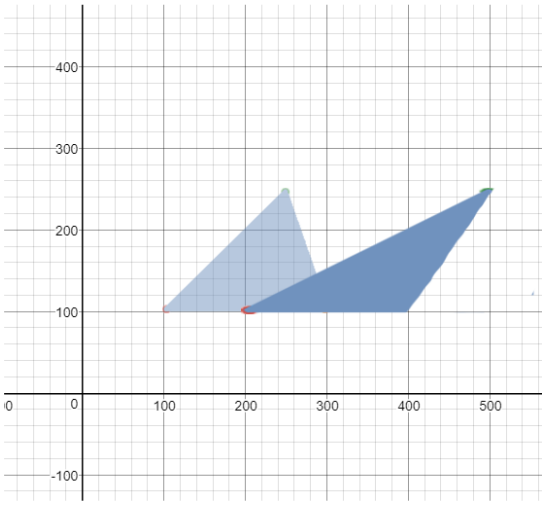
$$T(a,b,c) = \begin{bmatrix} -1 & 0 & 0 & -2*a \\ 0 & -1 & 0 & -2*b \\ 0 & 0 & -1 & -2*c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

-Terhadap bidang tertentu yang melewati titik(0,0,0) dengan a,b,c adalah vektor normal dari bidang

$$T(a,b,c) = \begin{bmatrix} 1-(2*a*a) & -2*a*b & -2*a*c & 0 \\ -2*a*b & 1-(2*b*b) & -2*b*c & 0 \\ -2*a*c & -2*b*c & 1-(2*c*c) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5. Shear

Transformasi shear adalah pergeseran dari salah satu sumbu. Atau bisa juga disebut sebagai distorsi dari bentuk suatu objek.



Matriks yang digunakan untuk transformasi 2 dimensi ada 2 macam, yaitu terhadap sumbu x

$$T(x,k) = \begin{bmatrix} 1 & k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Dan juga terhadap sumbu y

$$T(y,k) = \begin{bmatrix} 1 & 0 & 0 \\ k & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Untuk transformasi 3 dimensi, shear dapat dibedakan menjadi 6, yaitu terhadap xy, xz, yx, yz, zx, dan zy.

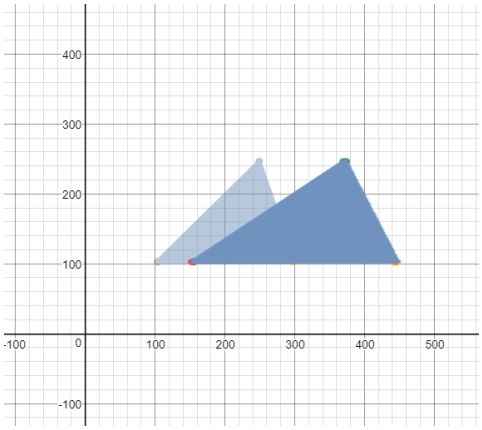
Secara umum, matriks dari transformasi tersebut adalah

$$T(k) = \begin{bmatrix} 1 & k_{yx} & k_{zx} & 0 \\ k_{xy} & 1 & k_{zy} & 0 \\ k_{xz} & k_{yz} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Saat shear dilakukan terhadap bidang xy, maka k_{xy}akan bernilai k, dan sisanya akan bernilai 0. Begitu juga bila shear dilakukan terhadap bidang yang lain

6. Strech(peregangan)

Transformasi stretch adalah memetakan suatu titik ke titik baru yang jaraknya *k* kali dari sumbu acuan stretch.



Untuk contoh diatas, titik yang tadinya berada di (100,100) dipetakan ke titik (150,100). Ini berarti stretch dilakukan terhadap sumbu x sebanyak 1.5 kali.
Matriks transformasi untuk 2 dimensi :
-Terhadap sumbu x

$$T(x,k) = \begin{bmatrix} k & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

-Terhadap sumbu y

$$T(y,k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matriks transformasi untuk 3 dimensi :
-Terhadap sumbu x

$$T(x,k) = \begin{bmatrix} k & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

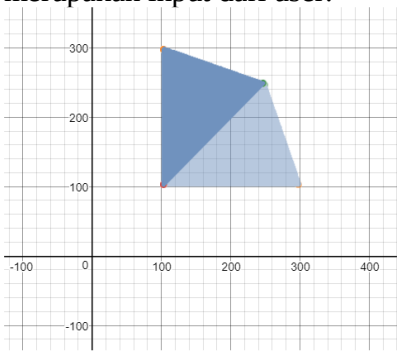
-Terhadap sumbu y

$$T(y,k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & k & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

-Terhadap sumbu z

$$T(z,k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

7. Custom
   Transformasi custom adalah transformasi menggunakan matriks sembarang yang merupakan input dari user.



   Jadi, apabila input user adalah a, b, c, d, maka matriks yang dibuat adalah

$$T(a,b,c,d) = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

   Hal tersebut juga berlaku untuk transformasi custom 3 dimensi.
8. Multiple
   Transformasi ini memungkinkan pengguna untuk melakukan beberapa transformasi sekaligus. Semua perintah transformasi dapat dilakukan dalam mode ini, kecuali multiple itu sendiri, reset, dan keluar program(exit) ditengah-tengah menjalankan multiple.

b. Library

Library adalah kumpulan dari fungsi dan prosedur yang sudah jadi. Kita dapat memakai library dan modul sesuai keperluan kita dan tidak perlu menambahkan semua library yang ada, sehingga program kita tidak terlalu berat. Library ini sangat memudahkan kita untuk membuat suatu program karena kita hanya tinggal memakai saja fungsi yang sudah ada tanpa perlu membuatnya lagi dari dasar.

1. OpenGL
   Open Graphics Library (*OpenGL*) adalah API (*Application Programming Interface*) yang berfungsi untuk melakukan *rendering* grafik 2D dan 3D. *OpenGL* bersifat *cross-language*, *cross-platform*, dan *open source*. *OpenGL* umumnya digunakan untuk melakukan interaksi dengan GPU (*graphics processing unit*) untuk mencapai hasil *render* yang diakselerasi dengan *hardware*. Anda diharapkan untuk melakukan eksplorasi penggunaan *OpenGL*. Berikut adalah contoh kode program yang menggunakan library *OpenGL*:

Kode Program (khusus untuk objek 2D)	**Keterangan**

GLfloat triangleVertices[] = {
100, 100, 0,
300, 100, 0,
250, 250, 0
};
	Mendefinisikan tiga buah titik, yaitu (100,100,0), (300,100,0), dan (250,250,0). Perhatikan nilai ketiga dari titik adalah nol supaya titik berupa 2D.

GLFWwindow *window; window = glfwCreateWindow (600, 600, "MyWindowName", NULL, NULL);	Membuat sebuah window yang akan Anda gunakan untuk menampilkan output program.

glEnableClientState(GL_VERTEX_ARRAY);
glVertexPointer(3, GL_FLOAT, 0, triangleVertices);
glDrawArrays(GL_POLYGON, 0, 3);
glDisableClientState(GL_VERTEX_ARRAY);
	Menggambar sebuah poligon sesuai titik-titik yang sudah didefinisikan pada `triangleVertices`.

2. PyGame
   Selain menggunakan OpenGL, kami juga menggunakan PyGame untuk interfacenya. Pygame merupakan modul dalam python yang biasanya digunakan untuk membuat desain sebuah video game yang dibuat dengan menggunakan bahasa pemrograman python.
3. Numpy
   Numpy adalah sebuah library yang memuat fungsi dengan keperluan utama menghitung. Dalam kasus matriks, numpy dapat digunakan untuk mengalikan matriks. Dalam kasus sudut, numpy dapat dik

BAB III. Implementasi Program dan Pembagian Tugas

Implementasi Program

Dalam membuat program ini, kami membaginya menjadi tiga buah file. Yang pertama adalah main.py(program utama), transformation.py, dan objects.py

- main.py
  Isi dari main.py adalah seluruh input output dari program. Input meliputi mode 2 dimensi atau 3 dimensi, jenis transformasi, dan parameter dari transformasi. Sedangkan output berisi seluruh perintah(command prompt) dan juga grafiknya(Pygame). Dalam main.py juga terdapat analisis kasus untuk setiap input dari pengguna.
- transformation.py

Seluruh matriks yang merupakan matriks transformasi ada di bagian ini. Dalam bagian ini juga terdapat seluruh fungsi untuk memetakan titik ke titik baru hasil perkalian dengan matriks transformasi yang sesuai dengan masukan pengguna.

- objects.py
  objects.py ini berisi benda benda yang ada untuk ditampilkan ke halaman pygame. Benda-benda ini meliputi garis sumbu, kubus dan bidang yang akan ditransformasi, dan bayangan saat transformasi.

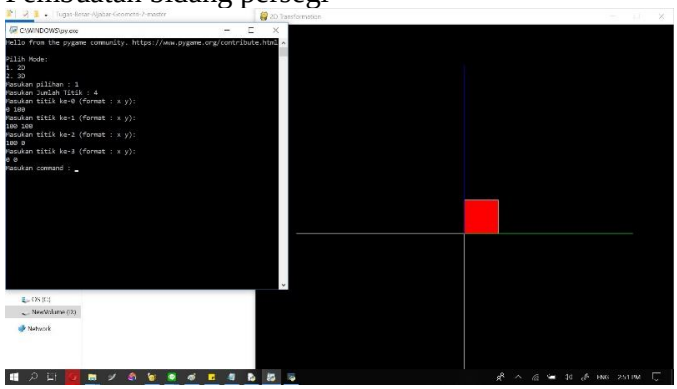
Pembagian Tugas

Anggota	Tugas
Ariel Ansa Razumardi / 13517040	Membuat dasar dari program, GUI, penggunaan OpenGL, membuat dasar animasi dan transformasi.
Ignatius Timothy Manullang / 13517044	Pembuatan fungsi transformasi 2D, mengetes program(untuk bagian eksperimen), membuat readme.
Vinsen Marselino Andreas / 13517054	Pembuatan fungsi transformasi 3D, dan membantu animasi, membuat laporan

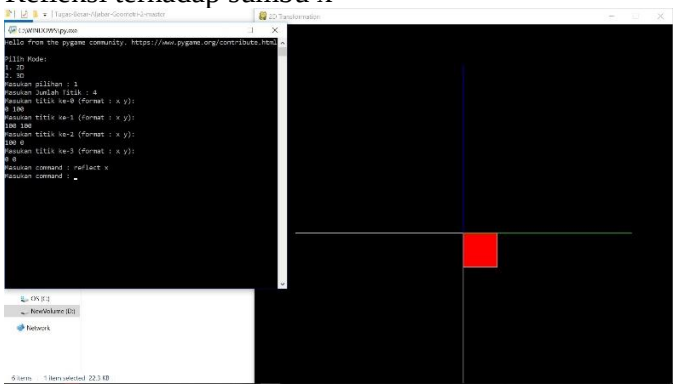
BAB IV. Eksperimen

- 2 Dimensi

Pembuatan bidang persegi

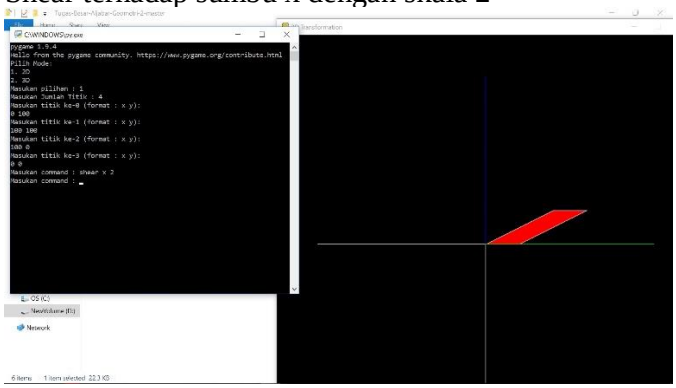


Refleksi terhadap sumbu x

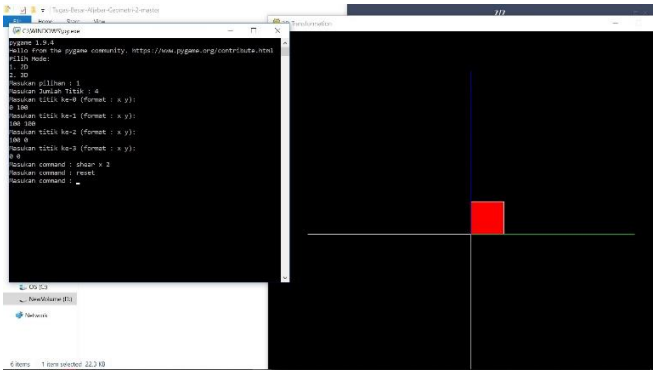


Refleksi terhadap sumbu y

Shear terhadap sumbu x dengan skala 2

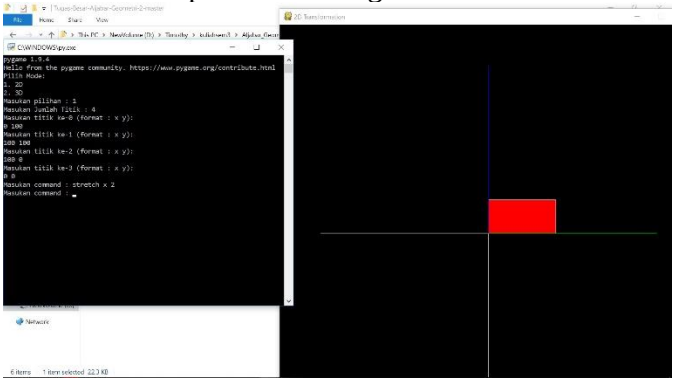


Reset ke keadaan semula

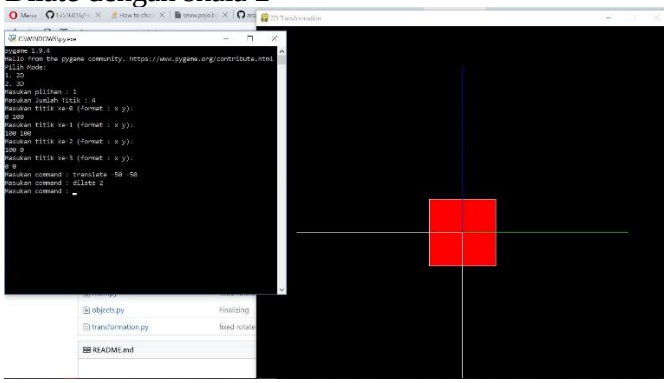


Translate sebanyak -50 -50(ke kiri 50 ke bawah 50)

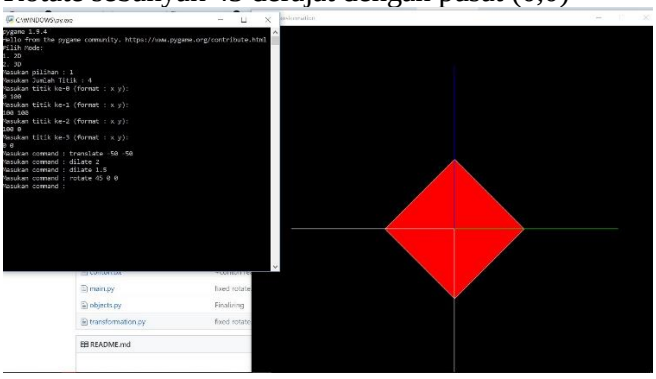
Stretch terhadap sumbu x dengan skala 2



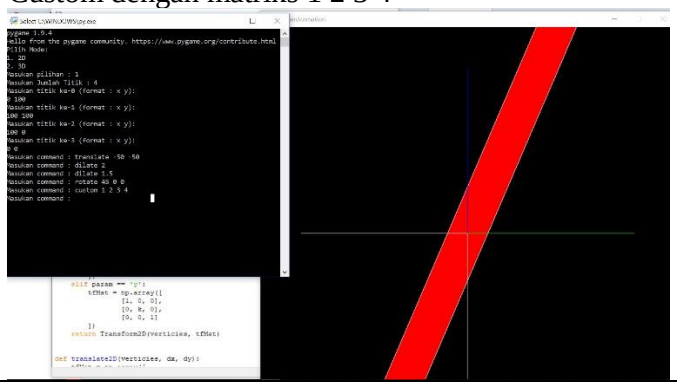
Dilate dengan skala 2



Rotate sebanyak 45 derajat dengan pusat (0,0)

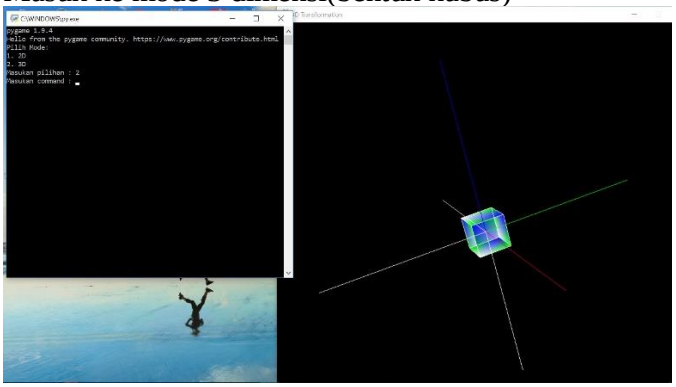


Custom dengan matriks 1 2 3 4

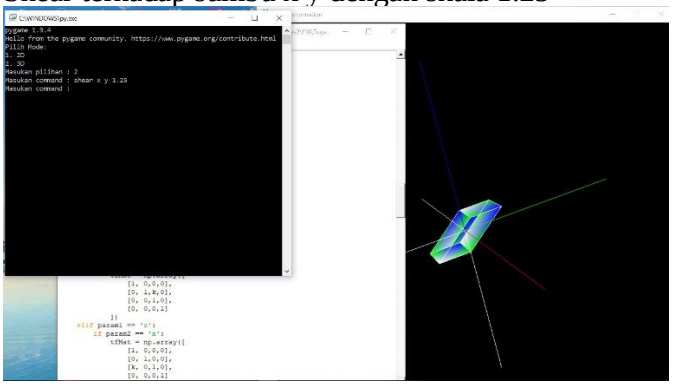


- 3 Dimensi

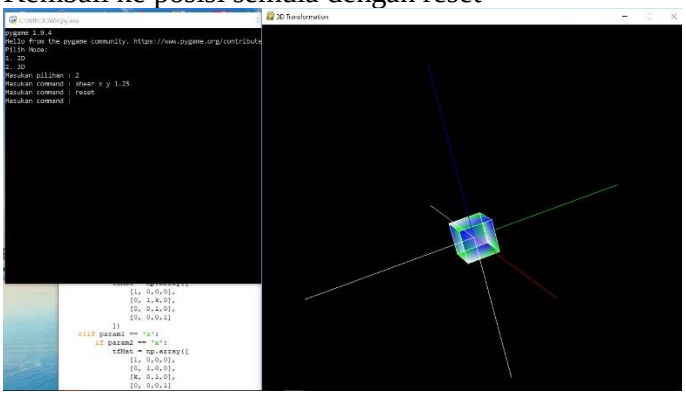
Masuk ke mode 3 dimensi(bentuk kubus)



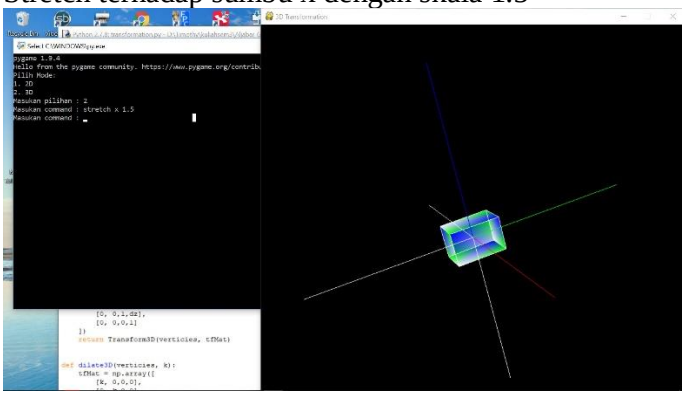
Shear terhadap sumbu x y dengan skala 1.25



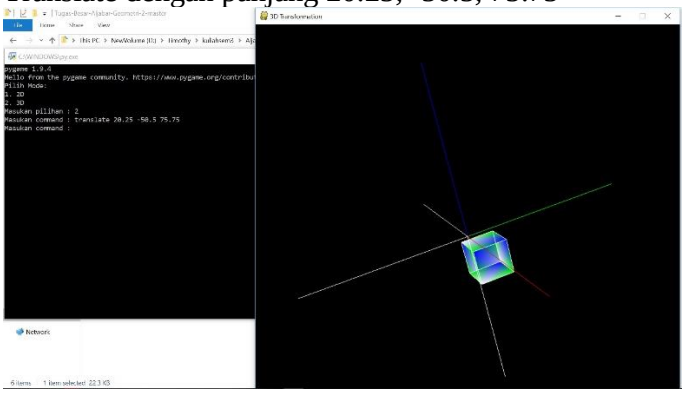
Kembali ke posisi semula dengan reset



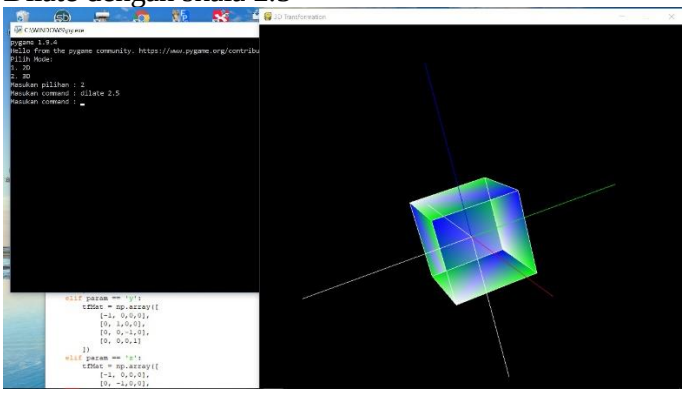
Stretch terhadap sumbu x dengan skala 1.5



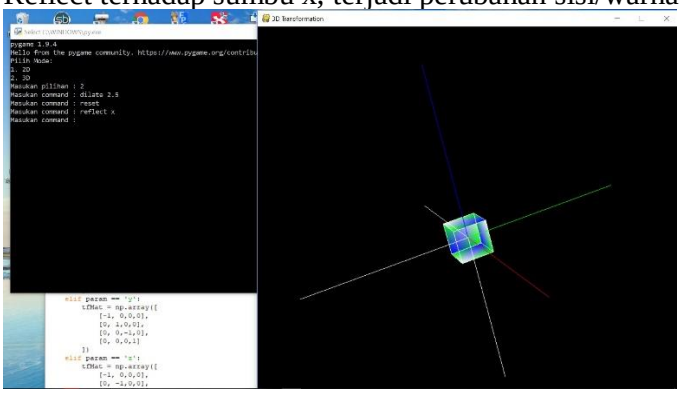
Translate dengan panjang 20.25, -50.5, 75.75



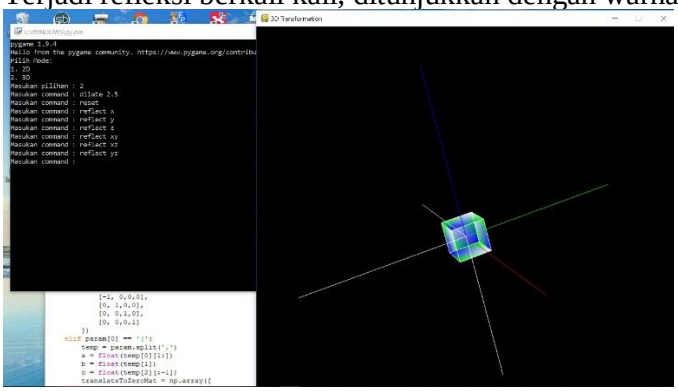
Dilate dengan skala 2.5



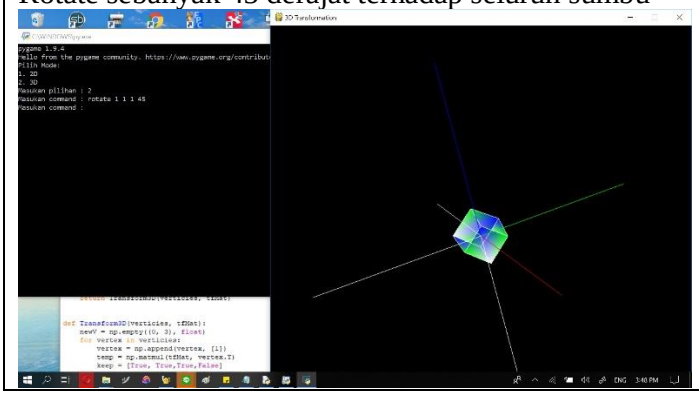
Reflect terhadap sumbu x, terjadi perubahan sisi/warna



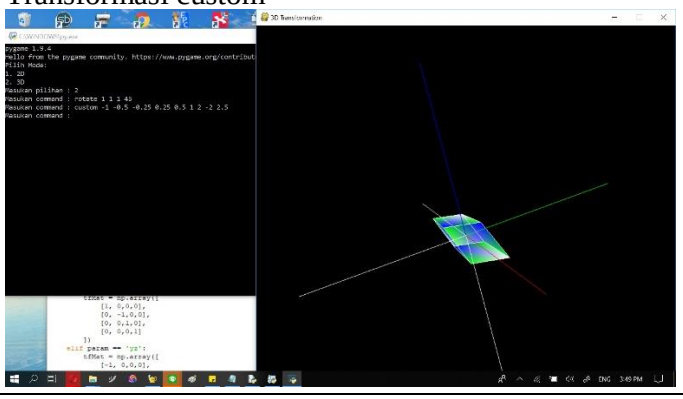
Terjadi refleksi berkali kali, ditunjukkan dengan warna



Rotate sebanyak 45 derajat terhadap seluruh sumbu



Transformasi custom



BAB V. Kesimpulan, saran, dan Refleksi

1. Kesimpulan

Kelompok kami sudah dapat membuat program untuk transformasi linier dengan baik dan sesuai harapan. Kelompok kami juga sudah dapat mengerti cara penggunaan OpenGL sebagai library yang digunakan untuk merender grafik. Selain OpenGL, kami juga menggunakan modul Pygame.Animasi sudah dapat kami lakukan dengan cara merender terus menerus, dengan transformasi yang sedikit sehingga dapat seolah-olah 'bergerak'. Kami juga sudah dapat membuat program membuka 2 tampilan window, yaitu command prompt untuk input dan window Pygame untuk grafik.

2. Saran

Untuk pengembangan aplikasi selanjutnya, dapat memperbaiki tampilan grafik, seperti warna dari objek yang lebih menarik, dan sumbu yang lebih jelas dengan *grid* supaya dapat lebih jelas. Selain itu, dapat pula dikembangkan supaya grafik dapat di *drag* supaya tampilan dapat lebih fleksibel. Diharapkan juga supaya program dapat melakukan validasi terlebih dahulu.

3. Refleksi

Tugas ini sudah cukup baik, karena memberikan peluang bagi mahasiswa untuk mengeksplor sendiri library dan bahasa pemrograman yang baru. Membuat program grafik pun merupakan hal yang baru diperkenalkan dan sangat menarik untuk dieksplor.

Daftar Pustaka

http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2018-2019/Tubes2-Algeo-2018.pdf

https://www.khronos.org/opengl/wiki/Getting_Started

https://en.wikipedia.org/wiki/Rotation_matrix#cite_note-5

https://www.tutorialspoint.com/computer_graphics/3d_transformation.htm