

Simulasi Transformasi Linier pada Bidang 2D Dengan Menggunakan OpenGL API

Diajukan untuk memenuhi tugas mata kuliah IF2123 Aljabar Geometri
oleh

RABBI FIJAR MAYOZA 13516081

HAFIZH BUDIMAN 13516137



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

2017

BAB I

DESKRIPSI MASALAH

Tugas besar ini adalah pembuatan 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. Selanjutnya, program dapat menerima *input* **translate**, **dilate**, **rotate**, **reflect**, **shear**, **stretch**, **custom**, **multiple**, dan **exit**. Keterangan masing-masing *input* sesuai dengan yang tertera pada *handout*.

BAB II

TEORI SINGKAT

2.1 OpenGL API

Open Graphics Library (OpenGL) adalah API 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.

2.2 Transformasi Linier

Transformasi linier adalah proses mengubah setiap titik koordinat menjadi titik koordinat lain pada bidang tertentu. Transformasi bisa juga dilakukan pada kumpulan titik yang membentuk bidang/bangun tertentu. Jika terdapat sebuah titik A (x,y) kemudian ditransformasikan oleh transformasi T maka akan menghasilkan titik yang baru A' (x',y'). Secara matematis di tulis:

$$A(x, y) \xrightarrow{t} A'(x', y')$$

2.3 Matriks Transformasi

Jenis - jenis matriks transformasi:

1. Refleksi

Refleksi atau sering disebut dengan istilah pencerminan adalah suatu transformasi dengan memindahkan setiap titik pada bidang dengan menggunakan sifat-sifat pencerminan pada cermin datar. Berikut tabel transformasi pencerminan.

Pencerminan Terhadap	Pemetaan	Matriks Transformasi
Sumbu x	$(x,y) \rightarrow (x,-y)$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Sumbu y	$(x,y) \rightarrow (-x,y)$	$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$
Garis $x = y$	$(x,y) \rightarrow (y,x)$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Garis $x = -y$	$(x,y) \rightarrow (-y,-x)$	$\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$
Titik (0,0)	$(x,y) \rightarrow (-x,-y)$	$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$
Garis $x = k$	$(x,y) \rightarrow (2k-x,y)$	
Garis $y = k$	$(x,y) \rightarrow (x,2k-y)$	

2. Rotasi

Rotasi adalah memutar setiap titik pada bidang dengan menggunakan titik pusat tertentu yang memiliki jarak sama dengan setiap titik yang diputar (jari-jari). Rotasi tidak mengubah ukuran benda sama sekali. Ada dua macam rotasi, rotasi dengan titik pusat (0,0) dan rotasi dengan titik tertentu P (a,b).

Table 5

Operator	Illustration	Rotation Equations	Standard Matrix
Counterclockwise rotation about the origin through an angle θ		$w_1 = x \cos \theta - y \sin \theta$ $w_2 = x \sin \theta + y \cos \theta$	$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$

2. Rotasi dengan Titik Pusat (a,b) dengan Sudut Putar α

Jika sobat punya sebuah titik (x,y) yang diputar sebesar α derajat dengan titik pusat P (a,b) maka:

$$A(x,y) \xrightarrow{R[P,\alpha]} A'(x',y')$$

dimana

$$x' - a = (x-a) \cos \alpha - (y-b) \sin \alpha$$

$$y' - b = (x-a) \sin \alpha + (y-b) \cos \alpha$$

3. Translasi

Translasi atau pergeseran adalah transformasi yang memindahkan setiap titik pada bidang menurut jarak dan arah tertentu. Translasi hanya memindahkan tanpa mengubah ukuran tanpa memutar. Kata kuncinya transformasi ke arah yang sama dan ke jarak yang sama.

$$T \begin{pmatrix} x \\ y \end{pmatrix} \xrightarrow{\begin{bmatrix} a \\ b \end{bmatrix}} T' \begin{pmatrix} x+a \\ y+b \end{pmatrix}$$

$$(x',y') = (x+a, y+b)$$

4. Regangan

Regangan atau dalam bahasa Inggris disebut stretching artinya transformasi dengan menarik sebuah benda searah sumbu x atau sumbu y dengan skala tertentu. Sebuah titik (x,y) ditarik ke arah sumbu x atau y dengan skala k.

Regangan	Pemetaan	Matriks Transformasi
Searah Sumbu x	$(x', y') \rightarrow (kx, y)$	$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} k & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
Searah Sumbu y	$(x', y') \rightarrow (x, ky)$	$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & k \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

5. Gusuran

Gusuran artinya menggeser searah sumbu x atau sumbu y dengan faktor skala tertentu. Untuk gusuran menurut sumbu x \rightarrow Jika nilai k positif maka ke kanan, jika k negatif maka ke kiri. Untuk gusuran menurut sumbu y \rightarrow Jika nilai k positif maka ke atas, jika k negatif maka ke bawah.

Gusuran	Pemetaan	Matriks Transformasi
Menurut Sumbu x	$(x', y') \rightarrow (x + ky, y)$	$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
Menurut Sumbu y	$(x', y') \rightarrow (x, kx + y)$	$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

6. Dilatasi

Jika $k > 1$ atau $k < -1$ maka diperbesar

Jika $-1 < k < 1$ maka diperkecil

Jika $k = 1$ atau $k = -1$ maka bangun tidak mengalami perubahan ukuran

1. Dilatasi terhadap titik pusat O (0,0) Dilatasi dengan pusat O (0,0) dan faktor dilatasi K maka

$$A(x, y) \xrightarrow{[O, k]} A'(kx, ky)$$

$$A \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

2. Dilatasi terhadap titik pusat P (a,b) jika sebuah titik didilatasi dengan faktor dilatasi k dan titik pusat P (a,b) maka

$$A(x, y) \xrightarrow{[P, k]} A'(x', y')$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix} \begin{pmatrix} x - a \\ y - b \end{pmatrix}$$

dimana

$$x' - a = k(x - a)$$

$$y' - b = k(y - b)$$

7. Transformasi Sembarang (Custom)

Mentransformasikan suatu titik dengan sebuah matriks yang bernilai sembarang.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \times \begin{pmatrix} x \\ y \end{pmatrix}$$

BAB III

IMPLEMENTASI PROGRAM

3.1 Implementasi Program

Pada tugas besar ini kami menggunakan bahasa pemrograman Python 2.7. OpenGL API dapat berjalan dengan PyOpenGL dan PyOpenGL_accelerate. Selain itu kami menambahkan library “numpy” untuk komputasi matriks.

Dalam program kami, kami membuat fungsi dan prosedur untuk pembuatan window GUI dari OpenGL API, diantaranya **def** refresh2d(width,height); **def** draw_polygon(); **def** draw_axis(); **def** drawinput(). Lalu fungsi dan prosedur tersebut dipanggil dalam program utama dari window GUI **def** main(). Sedangkan untuk eksekusi selain menampilkan *output* berupa gambar bidang hasil transformasi kami juga membuat program kami dapat menampilkan animasi gerakan bidang ketika operasi transformasi sedang dieksekusi.

Kami membuat fungsi dan prosedur untuk operasi transformasi:

a. **def** translate(dx,dy);

```
165 ''' Melakukan translasi objek dengan menggeser nilai x sebesar dx dan menggeser nilai y sebesar dy. '''
166 def translate(dx,dy):
167     a = dx/20000.0
168     b = dy/20000.0
169     a1 , b1 = 0.0 , 0.0
170     while a1<=abs(dx/2.0) and b1<=abs(dy/2.0):
171         for i in range(0, sisi):
172             matrix[i][0] += a
173             matrix[i][1] += b
174             a1 = abs(a)+a1
175             b1 = abs(b)+b1
```

Melakukan translasi objek dengan menggeser nilai x sebesar dx dan menggeser nilai y sebesar dy.

b. **def dilate(k);**

```
177 '''Melakukan dilatasi objek dengan faktor scaling k.'''
178 def dilate(k):
179     target = deepcopy(matrix)
180     for i in range(0, sisi):
181         target[i][0] *= k
182         target[i][1] *= k
183
184     for i in range(0, sisi):
185         cc, dd = 0, 0
186         while cc <= abs(matrix[i][0] - target[i][0]) and dd <= abs(matrix[i][0] - target[i][0]):
187             c = abs((target[i][0] - matrix[i][0]) / 20000.0)
188             d = abs((target[i][0] - matrix[i][0]) / 20000.0)
189             for i in range(0, sisi):
190                 matrix[i][0] += (target[i][0] - matrix[i][0]) / 20000.0
191                 matrix[i][1] += (target[i][1] - matrix[i][1]) / 20000.0
192             cc += c
193             dd += d
```

Melakukan dilatasi objek dengan faktor scaling k.

c. **def shear(sumbu, x);**

```
195 '''Melakukan operasi shear pada objek. Nilai param dapat berupa x
196 (terhadap sumbu x) atau y (terhadap sumbu y). Nilai k adalah faktor
197 shear.'''
198 def shear(sumbu, x):
199     if sumbu == 'x':
200         target = deepcopy(matrix)
201         for i in range(0, sisi):
202             target[i][0] = target[i][0] + (x * target[i][1])
203     elif sumbu == 'y':
204         target = deepcopy(matrix)
205         for i in range(0, sisi):
206             target[i][1] = target[i][1] + (x * target[i][0])
207
208     for i in range(0, sisi):
209         c = sisi * 20000
210         d = sisi * 20000
211         cc = 0
212         dd = 0
213         while(cc <= c and dd <= d):
214             for i in range(0, sisi):
215                 matrix[i][0] += (target[i][0] - matrix[i][0]) / 20000.0
216                 matrix[i][1] += (target[i][1] - matrix[i][1]) / 20000.0
217             cc += 1
218             dd += 1
```

Melakukan operasi shear pada objek. Nilai param dapat berupa x (terhadap sumbu x) atau y (terhadap sumbu y). Nilai k adalah faktor shear.

d. **def stretch(sumbu, x);**

```

220 '''Melakukan operasi stretch pada objek. Nilai param dapat berupa x
221 (terhadap sumbu x) atau y (terhadap sumbu y). Nilai k adalah faktor
222 stretch.'''
223 def stretch(sumbu,x):
224     target = deepcopy(matrix)
225     if sumbu == 'x':
226         for i in range(0, sisi):
227             target[i][0] = target[i][0]*x
228     elif sumbu == 'y':
229         for i in range(0, sisi):
230             target[i][1] = target[i][1]*x
231     for i in range(0,sisi):
232         c = sisi*20000
233         d = sisi*20000
234         cc = 0
235         dd = 0
236         while(cc<=c and dd<=d):
237             for i in range(0,sisi):
238                 matrix[i][0] += (target[i][0]-matrix[i][0])/20000.0
239                 matrix[i][1] += (target[i][1]-matrix[i][1])/20000.0
240             cc += 1
241             dd += 1

```

Melakukan operasi stretch pada objek. Nilai param dapat berupa x (terhadap sumbu x) atau y (terhadap sumbu y). Nilai k adalah faktor stretch.

e. **def rotate(deg,a,b);**

```

243 '''Melakukan rotasi objek secara berlawanan arah jarum jam sebesar deg
244 derajat terhadap titik a,b'''
245 def rotate(deg,a,b):
246     degr = radians(deg)
247     k = 0
248     d = degr/50000.0
249     for i in range(0, sisi):
250         while degr>=k :
251             for i in range(0,sisi):
252                 matrix[i][0] = ((matrix[i][0]-a)*cos(d)) - ((matrix[i][1]-b)*sin(d)) + a
253                 matrix[i][1] = ((matrix[i][0]-a)*sin(d)) + ((matrix[i][1]-b)*cos(d)) + b
254             k += degr/50000.0

```

Melakukan rotasi objek secara berlawanan arah jarum jam sebesar deg derajat terhadap titik a,b.

f. **def** custom(a,b,c,d);

```
256 '''Melakukan transformasi linier pada objek dengan matriks transformasi dengan elemen [a b][c d]'''
257 def custom(a,b,c,d):
258     target = deepcopy(matrix)
259     for i in range(0, sisi):
260         target[i][0] = matrix[i][0]*a + matrix[i][1]*c
261         target[i][1] = matrix[i][0]*b + matrix[i][1]*d
262
263     for i in range(0,sisi):
264         c = sisi*20000
265         d = sisi*20000
266         cc = 0
267         dd = 0
268         while(cc<=c and dd<=d):
269             for i in range(0,sisi):
270                 matrix[i][0] += (target[i][0]-matrix[i][0])/20000.0
271                 matrix[i][1] += (target[i][1]-matrix[i][1])/20000.0
272             cc += 1
273             dd += 1
```

Melakukan transformasi linier pada objek dengan matriks transformasi dengan elemen $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

g. **def** reflect(x);

```
293 '''Melakukan pencerminan objek. Nilai param adalah salah satu dari nilai-nilai
294 berikut: x, y, y=x, y=-x, atau (a,b). Nilai (a,b) adalah titik untuk
295 melakukan pencerminan terhadap.'''
296 def reflect(x):
297     target = deepcopy(matrix)
298     param = re.split(r'[, \s()]+', x)
299     if param[0] == 'y=x':
300         for i in range(0, sisi):
301             target[i][0] = matrix[i][1]
302             target[i][1] = matrix[i][0]
303     elif param[0] == 'y=-x':
304         for i in range(0, sisi):
305             target[i][0] = matrix[i][1]*(-1.0)
306             target[i][1] = matrix[i][0]*(-1.0)
307     elif param[0] == 'y':
308         for i in range(0, sisi):
309             target[i][0] = matrix[i][0]*(-1.0)
310     elif param[0] == 'x':
311         for i in range(0, sisi):
312             target[i][1] = matrix[i][1]*(-1.0)
313     else:
314         for i in range(0, sisi):
315             target[i][0] = -1*target[i][0] + float(param[1])
316             target[i][1] = -1*target[i][1] + float(param[2])
317
318     for i in range(0,sisi):
319         c = sisi*20000
320         d = sisi*20000
321         cc = 0
322         dd = 0
323         while(cc<=c and dd<=d):
324             for i in range(0,sisi):
325                 matrix[i][0] += (target[i][0]-matrix[i][0])/20000.0
326                 matrix[i][1] += (target[i][1]-matrix[i][1])/20000.0
327             cc += 1
328             dd += 1
```

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.

h. **def multiple(n);**

```

330 '''Melakukan transformasi linier pada objek sebanyak n kali berurutan.
331 Setiap baris input 1..n dapat berupa translate, rotate, shear, dll tetapi
332 bukan multiple, reset, exit.'''
333 def multiple(n):
334     Operasi = []
335     for i in range(n):
336         string = raw_input(">> Operasi %d = " %(i+1))
337         Operasi.append(string)
338
339     for string in Operasi:
340         pilih = []
341         pilih = string.split(" ")
342         if (pilih[0] == 'translate'):
343             x,y = float(pilih[1]),float(pilih[2])
344             translate(x,y)
345         elif (pilih[0] == 'dilate'):
346             k = float(pilih[1])
347             dilate(k)
348         elif (pilih[0] == 'rotate'):
349             deg,a,b = float(pilih[1]),float(pilih[2]),float(pilih[3])
350             rotate(deg,a,b)
351         elif (pilih[0] == 'reflect'):
352             reflect(pilih[1])
353         elif (pilih[0] == 'shear'):
354             sumbu,x = pilih[1],float(pilih[2])
355             shear(sumbu,x)
356         elif (pilih[0] == 'stretch'):
357             sumbu,x = pilih[1],float(pilih[2])
358             stretch(sumbu,x)
359         elif (pilih[0] == 'custom'):
360             a,b,c,d = float(pilih[1]),float(pilih[2]),float(pilih[3]),float(pilih[4])
361             custom(a,b,c,d)

```

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.

3.2 Pembagian Tugas

1. Rabbi Fijar Mayoza/13516081

- Menghias tampilan User Interface di CMD dan membuat User Interface menjadi lebih komunikatif
- Membuat main program
- Membuat fungsi rotate
- Membuat fungsi dilate
- Membuat fungsi multiple
- Membuat fungsi close
- Melakukan debugging
- Membuat laporan BAB I, II, III, IV, V
- Mencetak laporan
- Mengumpulkan tugas

2. Hafizh Budiman/13516137

- Membuat fungsi-fungsi pembuat grafik koordinat window GUI
- Membuat main program
- Membuat fungsi menu input
- Membuat fungsi translate
- Membuat fungsi shear
- Membuat fungsi stretch
- Membuat fungsi custom
- Membuat fungsi reflect
- Membuat fungsi reset
- Melakukan debugging
- Membuat laporan BAB III dan BAB IV
- Menyusun dokumentasi, readme.txt, dan executable file folder.

BAB IV

EKSPERIMEN

4.1. Contoh Menu Pembacaan Input Titik

Command Prompt

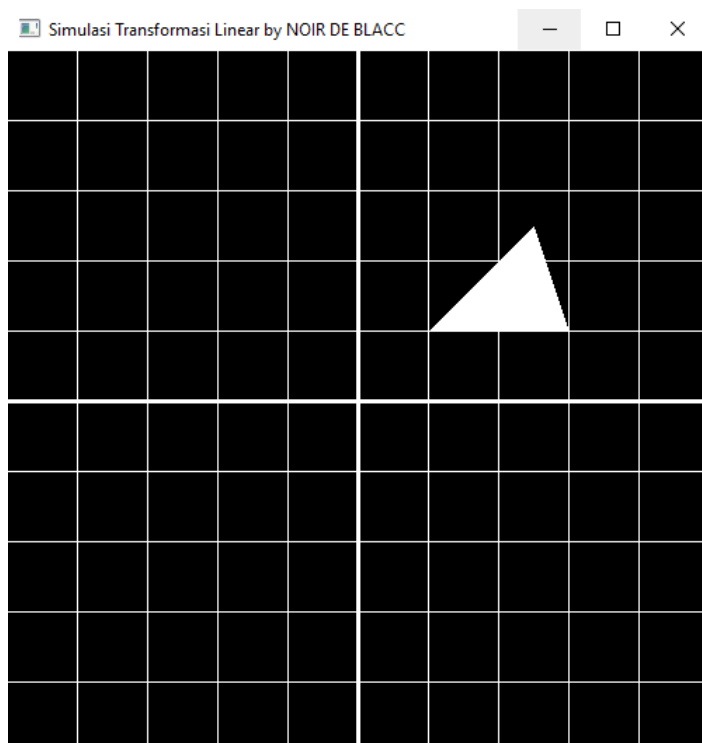
```
D:\MyCodes\Python\TubesAlgeo>python LinearTransformation.py

=====
==<
=====

-----by NOIR DE BLACC team-----
-----Rabbi Fijar Mayoza // 13516081-----
-----Hafizh Budiman // 13516137-----
-----

---SELAMAT DATANG DI PROGRAM TRANSLASI LINEAR KAMI---
>> Input jumlah sisi dari bidang = 3
>> titik x,y 1 : 100,100
>> titik x,y 2 : 250,250
>> titik x,y 3 : 300,100
```

Window GUI



Selanjutnya program akan menampilkan menu pilihan

```

>> Menu =
>> - translate <dx> <dy>
>>   Melakukan translasi objek dengan menggeser sejauh dx dan dy
>> - dilate <k>
>>   Melakukan dilatasi objek dengan faktor skala k
>> - rotate <deg> <a> <b>
>>   Melakukan rotasi objek berlawanan arah jarum jam sebesar <deg>
>>   derajat terhadap titik <a>,<b>
>> - reflect <param>
>>   Melakukan pencerminan objek terhadap <param>, <param> dapat berupa:
>>   x, y, y=x, y=-x, atau suatu titik (a,b)
>> - shear <param> <k>
>>   Melakukan operasi shear pada objek. Nilai <param> berupa
>>   sumbu x atau y, dan k adalah faktor shear.
>> - stretch <param> <k>
>>   Melakukan operasi stretch pada objek. Nilai <param> berupa
>>   sumbu x atau y, dan k adalah faktor stretch.
>> - custom <a> <b> <c> <d>
>>   Melakukan transformasi linier dengan objek matriks sebagai berikut :
>>
>>    $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 
>> - multiple <n>
>>   Melakukan transformasi linier pada objek sebanyak n kali berurutan
>> - reset
>>   Mengembalikan objek pada kondisi awal objek didefinisikan
>> - exit
>>   Keluar dari program
>> -----
>> Pilihan =

```

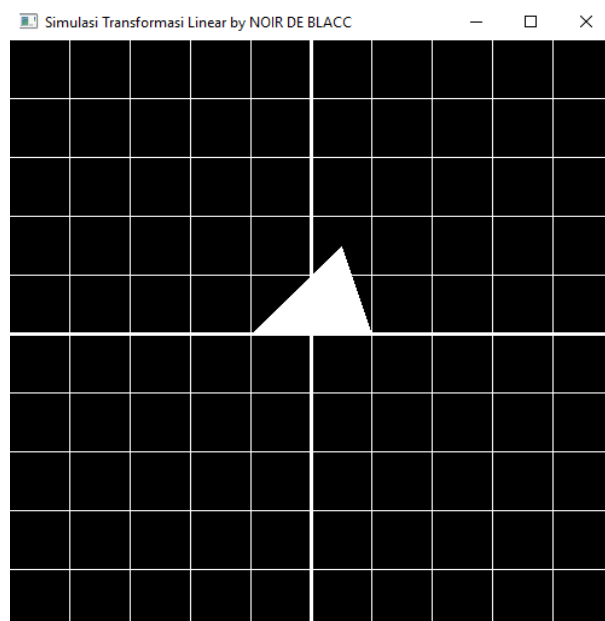
4.2. Contoh Transformasi Objek dan Hasil Eksekusi

4.2.1. Translasi

Command Prompt

```
>> Pilihan = translate -200 -100
```

Window GUI

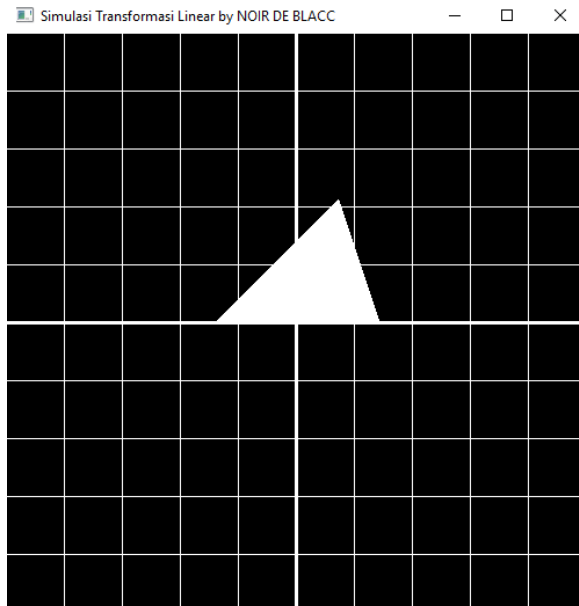


4.2.2. Dilatasi

Command Prompt

```
>> Pilihan = dilate 1.5
```

Window GUI

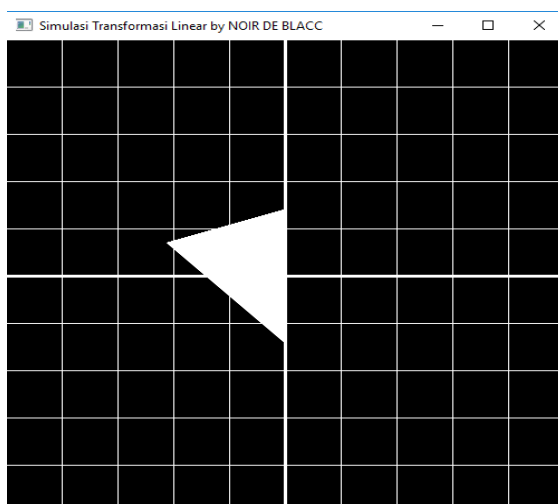


4.2.3. Rotasi

Command Prompt

```
>> Pilihan = rotate 90 0 0
```

Window GUI

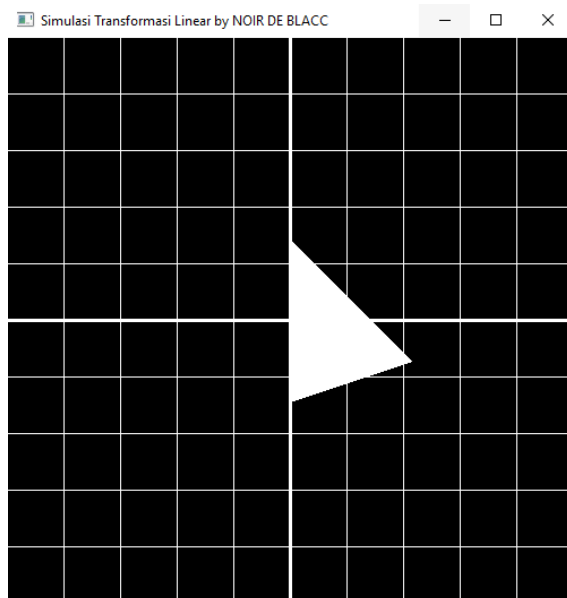


4.2.4 Refleksi

Command Prompt

```
>> Pilihan = reflect (0,0)
```

Window GUI

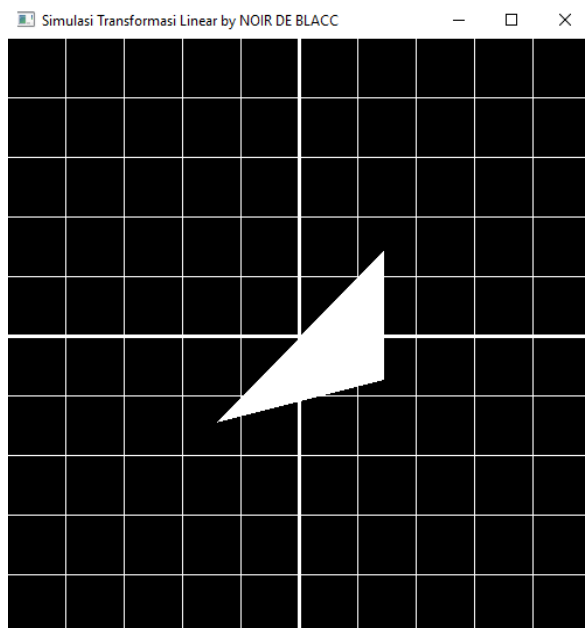


4.2.5 Gusuran (*Shear*)

Command Prompt

```
>> Pilihan = shear x 1
```

Window GUI

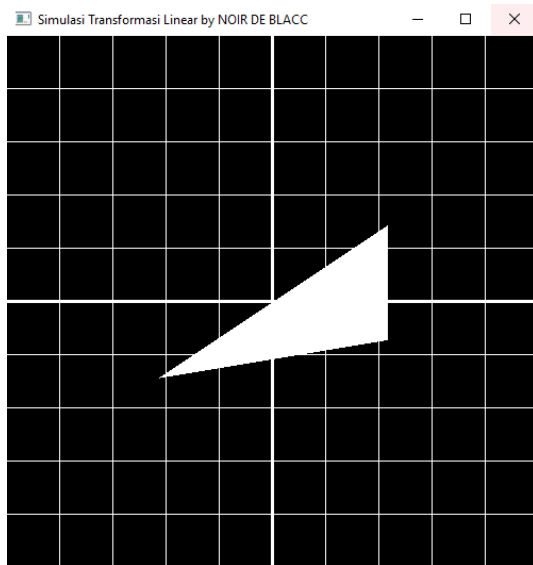


4.2.6 Regangan (*Stretch*)

Command Prompt

```
>> Pilihan = stretch x 1.5
```

Window GUI

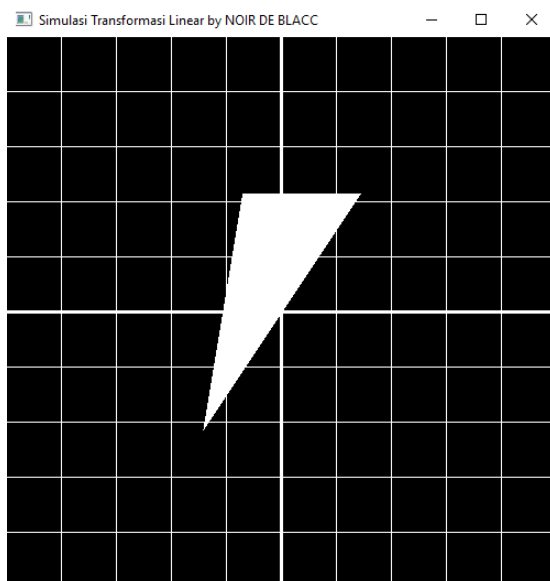


4.2.7. Custom

Command Prompt

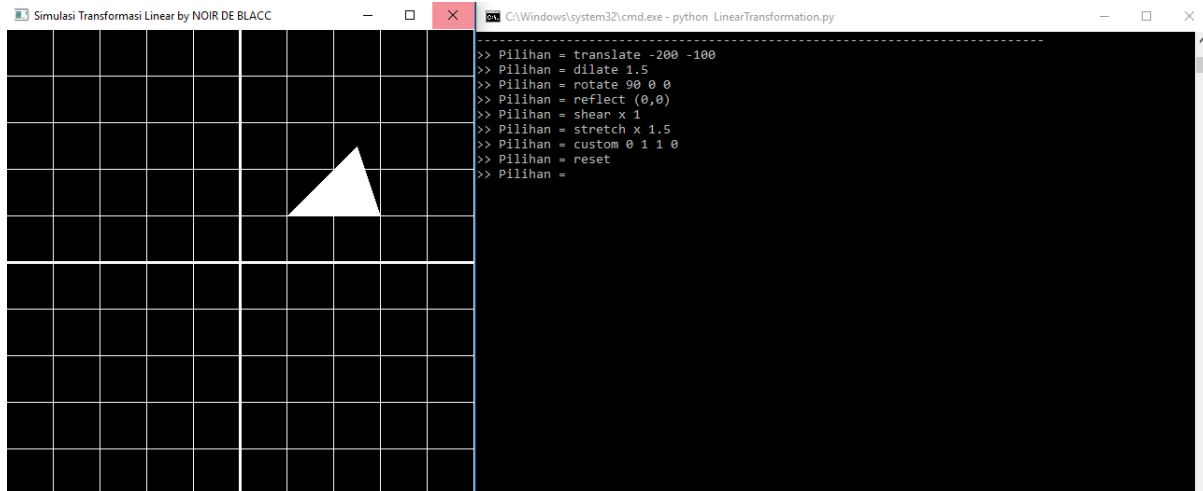
```
>> Pilihan = custom 0 1 1 0
```

Window GUI



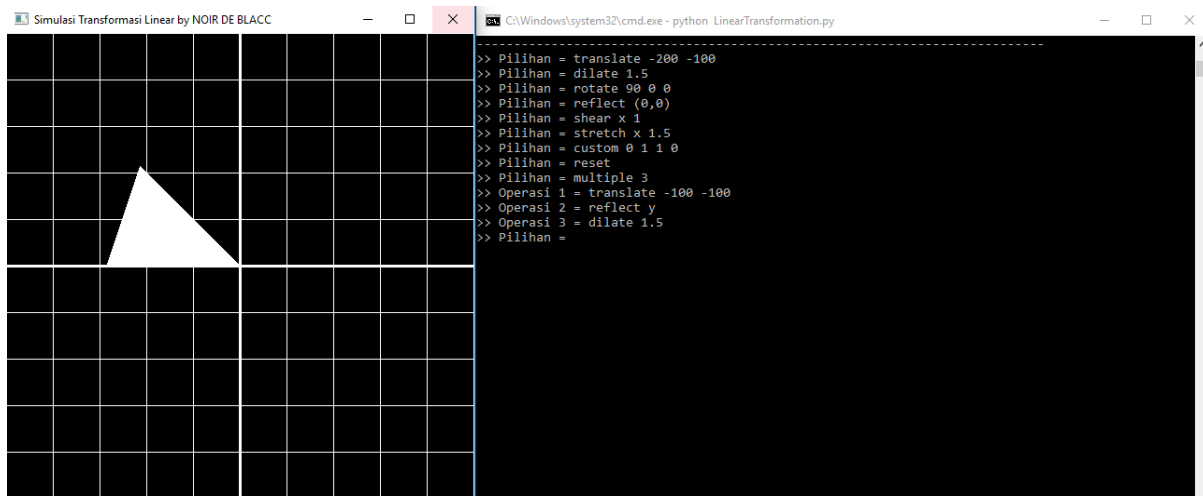
4.2.8 Reset

Segitiga akan kembali ke keadaan mula-mula

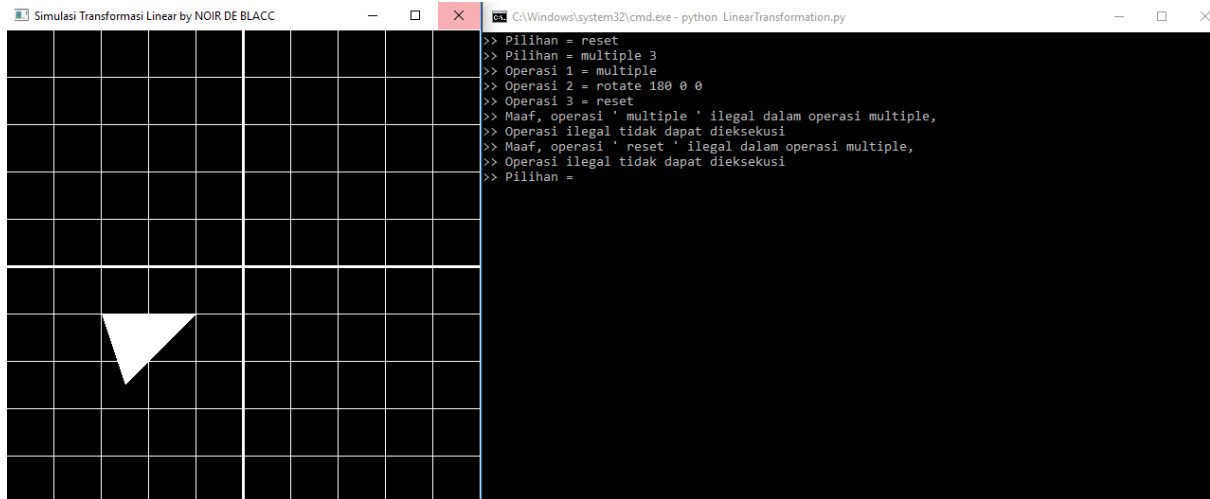


4.2.9 Multiple

Jika semua masukan benar, program akan mengeksekusi transformasi bidang secara berurutan mulai dari operasi-1 sampai operasi $-n$ sesuai urutan operasi yang di-input user.

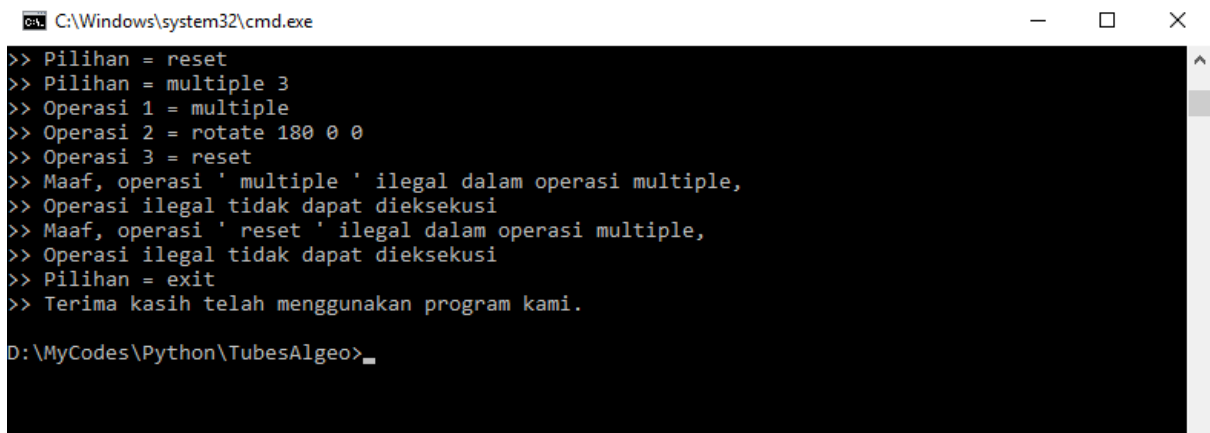


Jika ada masukan yang salah, program hanya akan mengeksekusi transformasi bidang yang masukannya benar saja dan tetap secara berurutan mulai dari operasi-1 sampai operasi $-n$ sesuai urutan operasi yang di-input user. Setelah itu akan ditampilkan pesan kesalahan.



4.2.10 Exit

Mengakhiri program dan menutup window GUI



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan uji coba yang telah kami lakukan, program yang kami buat dapat mengeksekusi berbagai jenis transformasi linier bidang 2 dimensi dan menampilkan hasil eksekusi beserta animasinya dalam window GUI dengan menggunakan OpenGL API.

5.2. Saran

Dari apa yang telah kami kerjakan, terdapat hambatan dalam mengerjakan tugas ini karena kurangnya pengetahuan kami tentang OpenGL. Dengan mengeksplorasi OpenGL dan Python lebih dalam lagi, program kami bisa dikembangkan agar dapat mensimulasikan transformasi linier bidang 3 dimensi dan menampilkan animasinya yang tentunya akan lebih kompleks namun akan lebih keren.

DAFTAR REFERENSI

- Sam Hokevar. 2004. "Opengl-tutorial". Diakses dari <http://www.opengl-tutorial.org/> pada tanggal 29 Oktober 2017.
- Noobtuts Team, "Python OpenGL-Introduction". Diakses dari <https://noobtuts.com/python/opengl-introduction> pada tanggal 30 Oktober 2017
- Rumus Hitung. 2015. "Rumus Transformasi Geometri". Diakses dari <http://rumushitung.com/2015/04/18/rumus-transformasi-geometri/> pada tanggal 4 November 2017.
- Anton, Howard. 2014. *Elementary Linear Algebra*. New Jersey: Wiley.