

LAPORAN TUGAS BESAR 2 ALJABAR GEOMETRI

LAPORAN

Diajukan untuk memenuhi Tugas Besar Aljabar Geometri

oleh:

ABRAM PERDANAPUTRA / 13516083
TIMOTHY THAMRIN A. H. S. / 13516090



**TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2017**

BAB I

DESKRIPSI MASALAH

Pada tugas besar ini, dibuat sebuah program berupa program yang dapat membantu pengguna dalam mengoperasikan operasi dalam matriks 2D. Program ini dibuat menggunakan bahasa Python. Program ini juga menggunakan PyOpenGL. Program ini dieksekusi menggunakan command line dan menggunakan PyOpenGL. Program ini memiliki mekanika sesuai algoritma yang sudah diberikan dalam spesifikasi, dan menggunakan berbagai tipe struktur data yang berbeda.

Dalam laporan ini akan dibahas lebih lengkap spesifikasi dari program yang dibuat, berbagai fungsi yang digunakan, penjelasan algoritma program yang telah dibuat, contoh implementasi program, pembagian tugas, dan lampiran-lampiran yang diperlukan.

Secara keseluruhan, pengerjaan tugas besar ini dilakukan berkelompok dan bersifat modular, dengan pengaplikasian beberapa struktur data yang dibutuhkan sesuai spek. Selain itu, tugas besar ini bermanfaat sebagai salah satu implementasi terhadap berbagai algoritma dan struktur data yang telah dipelajari di mata kuliah Aljabar Geometri.

Kami **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 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**.

BAB II

TEORI SINGKAT

Untuk memindahkan suatu titik atau bangun pada bidang dapat dilakukan dengan menggunakan transformasi. Transformasi geometri adalah bagian dari geometri yang membicarakan perubahan, baik perubahan letak maupun bentuk dan penyajiannya didasarkan dengan gambar dan matriks.

Operasi Matriks :

1. **Fungsi Translasi** : *def translate(self, dx, dy):*

Spesifikasi : Translasi (pergeseran) adalah pemindahan suatu objek sepanjang garis lurus dengan arah dan jarak tertentu.

$$T = \begin{pmatrix} a \\ b \end{pmatrix}$$

Jika translasi memetakan titik P (x, y) ke titik P'(x', y') maka $x' = x + a$ dan $y' = y + b$ atau P' (x + a, y + b) ditulis dalam bentuk :

$$T = \begin{pmatrix} a \\ b \end{pmatrix} : P(x, y) \rightarrow P'(x + a, y + b)$$

2. **Fungsi Dilatasi** : *def dilate(self, k):*

Spesifikasi : Dilatasi dengan pusat (a, b) dan faktor skala k

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

Rumus praktis dilatasi dengan faktor skala k dan pusat dilatasi O(0, 0):

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

3. Fungsi Rotasi : *def rotate(self, deg, a, b):*

Spesifikasi : Rotasi sejauh θ dengan pusat (a, b)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x-a \\ y-b \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix}$$

Rumus praktis untuk rotasi dengan pusat rotasi O(0, 0):

$$\begin{aligned} P(x, y) &\xrightarrow{R[0, 90^\circ]} P'(-y, x) \\ P(x, y) &\xrightarrow{R[0, -90^\circ]} P'(y, -x) \\ P(x, y) &\xrightarrow{R[0, 180^\circ]} P'(-x, -y) \end{aligned}$$

4. Fungsi Refleksi : *def reflect(self, param):*

Spesifikasi : Pencerminkan terhadap garis $x = a$ atau $y = b$

$$\begin{aligned} P(x, y) &\xrightarrow{M_{x=a}} P'(2a - x, y) \\ P(x, y) &\xrightarrow{M_{y=b}} P'(x, 2b - y) \end{aligned}$$

Pencerminkan terhadap sumbu x atau sumbu y

$$\begin{aligned} P(x, y) &\xrightarrow{M_x} P'(x, -y) \Rightarrow \text{matriks transformasi: } \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\ P(x, y) &\xrightarrow{M_y} P'(-x, y) \Rightarrow \text{matriks transformasi: } \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

Pencerminkan terhadap titik (0, 0)

$$P(x, y) \xrightarrow{M_{(0,0)}} P'(-x, -y) \Rightarrow \text{matriks transformasi: } \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

Pencerminkan terhadap garis $y = x$ atau $y = -x$

$$P(x, y) \xrightarrow{M_{\Phi=x}} P'(y, x) \Rightarrow \text{matriks transformasi: } \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$P(x, y) \xrightarrow{M_{\Phi=-x}} P'(-y, -x) \Rightarrow \text{matriks transformasi: } \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

5. **Fungsi Shear** : *def shear(self, param, k):*

Spesifikasi : Untuk pemetaan geser (secara visual mirip dengan slanting), ada dua kemungkinan. Untuk paralel geser dengan sumbu x memiliki $x' = x + ky$ dan $y' = y$; matriks geser, yang diterapkan pada vektor kolom, adalah:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Sebuah paralel geser ke sumbu y memiliki $x' = x$ dan $y' = y + kx$, yang memiliki bentuk matriks:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

6. **Fungsi Stretch** : *def stretch(self, param, k):*

Spesifikasi : Matriks yang terkait dengan peregangan oleh faktor k sepanjang sumbu x diberikan oleh

$$T_x[k] = \begin{bmatrix} k & 0 \\ 0 & 1 \end{bmatrix}$$

Demikian pula, peregangan oleh faktor k sepanjang sumbu y memiliki bentuk $x' = x$; $y' = ky$, jadi matriks yang terkait dengan transformasi ini

$$T_y[k] = \begin{bmatrix} 1 & 0 \\ 0 & k \end{bmatrix}$$

BAB III

IMPLEMENTASI PROGRAM

3.1 File main.py

- a. `clear()` :
Membersihkan terminal.
- b. `drawCoord(width, height)` :
Menggambar sumbu koordinat pada layar.
- c. `drawPolygon(polygon)` :
Menggambar objek 2 dimensi pada layar.
- d. `refresh2d(width, height)` :
Melakukan pembaruan pada layar.
- e. `animate(polygon, next_polygon, duration, interval)` :
Melakukan animasi saat terjadi perubahan bentuk.
- f. `userInput()` :
Menerima dan memroses input dari pengguna.
- g. `executeCommand(query)` :
Menjalankan instruksi pada input multiple.
- h. `GLUT()` :
Melakukan inisialisasi pada OpenGL.

3.2 File GObject.py

Class `Object2d`

- a. `__init__(self, mat = np.zeros((0, 0)))` :
Melakukan inisialisasi objek dengan matriks nol jika tidak ada input matriks.
- b. `copyObject(self)` :
Mengembalikan salinan objek.

- c. `countVertex(self)` :
Mengembalikan jumlah titik pada objek.
- d. `vertices(self)` :
Mengembalikan titik-titik yang ada pada objek.
- e. `translate(self, dx, dy)` :
Mengembalikan objek yang sudah ditranslasi.
- f. `dilate(self, k)` :
Mengembalikan objek yang sudah didilatasi.
- g. `rotate(self, deg, a, b)` :
Mengembalikan objek yang sudah dirotasi.
- h. `reflect(self, param)` :
Mengembalikan objek yang sudah dicerminkan.
- i. `shear(self, param, k)` :
Mengembalikan objek yang sudah melalui proses *Shear*.
- j. `stretch(self, param, k)` :
Mengembalikan objek yang sudah diregangkan.
- k. `custom(self, a, b, c, d)` :
Mengembalikan objek yang sudah ditransformasi dengan matriks custom.

3.3 Jenis command

1. `draw2d` :

Meminta masukan jumlah titik dari user dan meminta masukan (x y) titik sebanyak jumlah titik. Mengembalikan objek yang dibentuk oleh titik - titik.

2. `translate <dx> <dy>` :

Mengembalikan objek yang sudah ditranslasi.

3. dilate <k> :

Mengembalikan objek yang sudah ditranslasi.

4. rotate <deg> <a> :

Mengembalikan objek yang sudah dirotasi.

5. reflect <param> :

Mengembalikan objek yang sudah dicerminkan.

6. shear <param> <k> :

Mengembalikan objek yang sudah melalui proses *Shear*.

7. stretch <param> <k> :

Mengembalikan objek yang sudah diregangkan.

8. custom <a> <c> <d> :

Mengembalikan objek yang sudah ditransformasi dengan matriks custom.

9. multiple <n> ;

Mengembalikan objek yang sudah melalui n proses. (Prosesnya sesuai dengan inputan user.

10.reset :

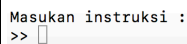
Mengembalikan objek seperti awal setelah input draw2d.

11.exit :

Keluar dari program.

EKSPERIMEN/CONTOH

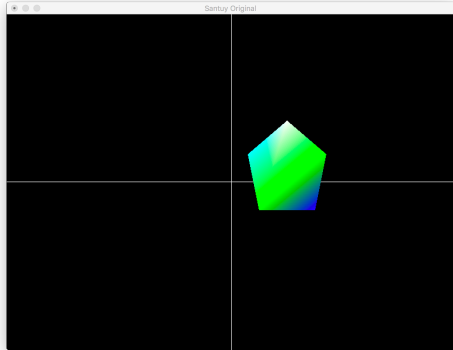
1. Tampilan awal :



2. Menggambar objek 2 dimensi

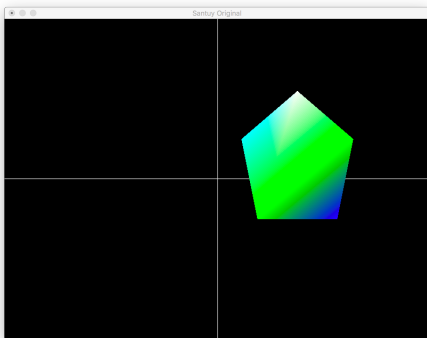


3. Melakukan translasi objek



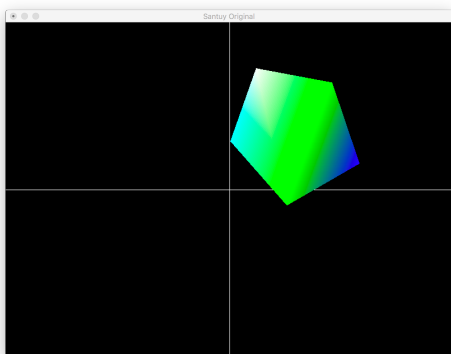
Masukan instruksi :
`>> translate 50 -50`

4. Melakukan dilatasi objek



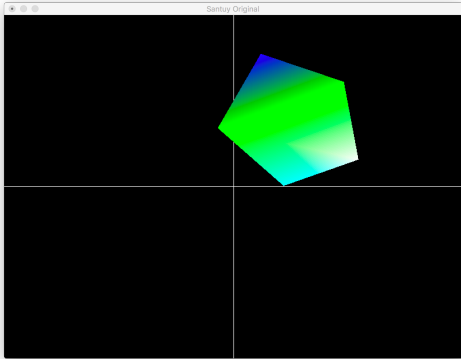
Masukan instruksi :
`>> dilate 1.5`

5. Melakukan rotasi objek



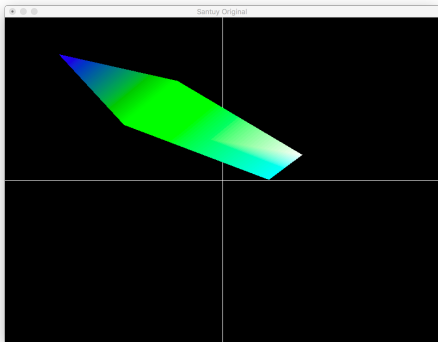
Masukan instruksi :
`>> rotate 30 0 0`

6. Melakukan pencerminan objek



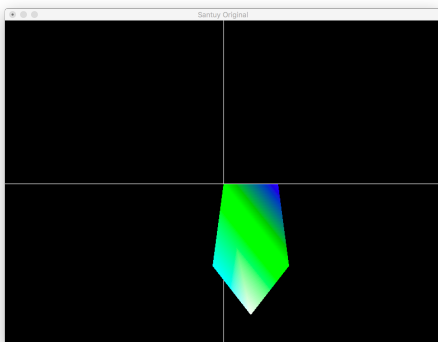
Masukan instruksi :
`>> reflect y=x`

7. Melakukan *Shear* pada objek



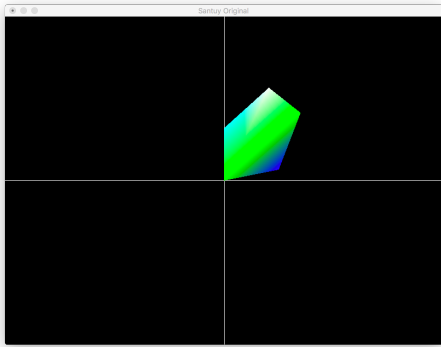
Masukan instruksi :
`>> shear x -1.5`

8. Melakukan peregangan objek



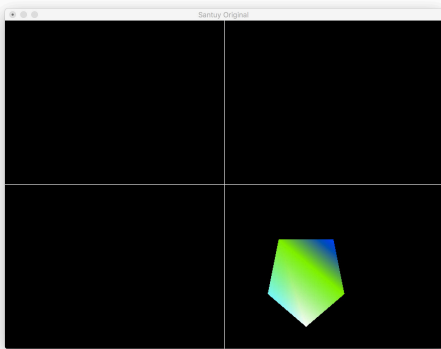
Masukan instruksi :
`>> stretch y -1.5`

9. Melakukan transformasi dengan matriks



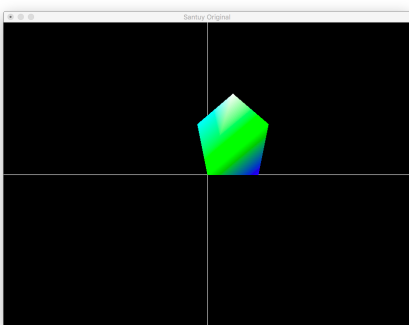
Masukan instruksi :
>> custom 1 0.2 0.2 1

10. Melakukan beberapa instruksi sekaligus



Masukan instruksi :
>> multiple 2
translate 100 100
reflect x

11. Mengembalikan objek ke posisi awal



Masukan instruksi :
>> reset

BAB V

KESIMPULAN

Dapat dibuat program dengan algoritma untuk melakukan operasi pada bidang dalam bentuk matriks yang direpresentasikan atau divisualisasikan dalam OpenGL. Untuk mengoperasikan aljabar pada matriks dapat menggunakan perkalian matriks dengan matriks yang memiliki bentuk dan karakteristik tersendiri.

Dari tugas besar ini dapat disimpulkan bahwa permasalahan yang bersifat matematis dapat diselesaikan dengan bantuan komputer jika digunakan algoritma yang benar sesuai dengan permasalahan yang ingin diselesaikan dalam program tersebut.

Hasil yang didapatkan dari permasalahan ini yaitu dapat menyelesaikan berbagai permasalahan lebih lanjut, seperti penyelesaian soal ujian anak SMA atau SMP, dan memvisualisasikan langsung dengan apa yang terjadi

Pengembangan lebih lanjut yang dapat diimplementasikan dalam program adalah dengan mencari dan mengaplikasikan algoritma yang lebih efisien untuk mengurangi memori dan waktu penyelesaian yang dibutuhkan komputer untuk menyelesaikan permasalahan khususnya dalam tugas ini.

Daftar Pustaka

<http://www.opengl-tutorial.org/>

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

<https://www.opengl.org/sdk/libs/>

<https://en.wikipedia.org/wiki/OpenGL>

<http://www.pyinstaller.org>

<https://docs.python.org/3/library>

<https://docs.scipy.org/doc/numpy-1.13.0/reference/>

<https://noobtuts.com/python/opengl-introduction>