

Laporan Tugas Besar Bagian 1 IF3260 Grafika Komputer
Semester II tahun 2023/2024
WebGL Part 1 : 2D Primitive Elements



Disusun oleh (Kelompok 4 - Grafkonn):

13521119	Muhammad Rizky Sya'ban
13521121	Saddam Annais Shaquille
13521166	Mohammad Rifqi Farhansyah

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2024**

Daftar Isi

Daftar Isi	2
Bab 1	3
Pendahuluan	3
A. Deskripsi Umum	3
B. Landasan Teori	3
Bab 2	5
Implementasi	5
A. Struktur Kode	5
B. Manual/Contoh Fungsionalitas Program	5
Lampiran	15
A. Pembagian Tugas	15
B. Pranala Terkait	15

Bab 1

Pendahuluan

A. Deskripsi Umum

WebGL merupakan kakas dengan spesialisasi pada ranah grafika yang dapat dengan mudah diintegrasikan pada sebuah *platform* berbasis web. Dengan menggunakan WebGL, kita dapat mengimplementasikan web dengan fitur menggambar, mengedit, serta memvisualisasikan sejumlah model pada kanvas HTML. Selanjutnya, WebGL juga mengutilisasikan GPU untuk melakukan *rendering* dan transformasi geometri. Oleh karena itu, aplikasi yang diimplementasikan pada tugas besar ini, yaitu: *2D Primitive Elements* menggunakan WebGL sebagai kakas untuk memodelkan berbagai bentuk geometri.



Gambar 1.1 Logo WebGL

Pada aplikasi *2D Primitive Elements* yang telah dikembangkan oleh penulis, pengguna dapat menambahkan berbagai bentuk geometris, seperti garis, persegi, persegi panjang, dan poligon. Kemudian, pengguna juga dapat melakukan transformasi geometri terhadap seluruh model yang telah ditampilkan di layar. Transformasi geometri yang dapat dilakukan oleh pengguna adalah translasi (pergerakan objek), dilatasi (perbesaran atau pengecilan objek agar tetap sebangun), resize (perbesaran salah satu pasangan sisi sejajar dari objek), rotasi (pemutaran objek), dan shear (pemindahan salah satu titik sudut dari objek). Tidak hanya transformasi geometri, pengguna juga dapat mengubah warna titik dari setiap bentuk geometris yang tercatat pada layar.

Berikut ini merupakan beberapa fitur yang telah berhasil diimplementasikan oleh penulis pada aplikasi *2D Primitive Elements*:

- Pemodelan garis
- Pemodelan bangun geometris persegi
- Pemodelan bangun geometris persegi panjang
- Pemodelan bangun geometris poligon
- Transformasi geometri : translasi, rotasi, dilatasi, resize, dan shear
- Penambahan dan pengurangan *vertex* pada bentuk geometris poligon
- Pewarnaan *vertex*
- Penyimpanan dan pengunggahan model

B. Landasan Teori

Dalam proses implementasi transformasi geometri dalam aplikasi ini, tentunya diperlukan pendekatan matematis agar proses transformasi geometri dapat terealisasi. Berikut ini adalah pendekatan matematis yang digunakan untuk setiap transformasi geometri:

- Translasi : perpindahan posisi suatu objek. Jika dinyatakan dalam koordinat kartesius, translasi merupakan perpindahan titik-titik koordinat suatu objek ke arah dan jarak tertentu. Apabila titik P yang memiliki koordinat (x,y) ditranslasikan sejauh (a,b), akan dihasilkan titik

P' dengan koordinat (x',y'). Secara matematis, perhitungan koordinat akhir pada proses translasi dinyatakan sebagai berikut:

$$P(x, y) \rightarrow P'((x + a), (y + b))$$

- Rotasi : perpindahan titik-titik suatu objek pada bidang geometri dengan cara memutarkannya sejauh sudut α . Apabila titik K yang memiliki koordinat (x,y) dirotasi sejauh α melalui titik pusat (0,0) hingga berada pada posisi K'(x',y'). Secara matematis, persamaan rotasi atas permasalahan tersebut dapat dinyatakan sebagai berikut:

$$K(x, y) \rightarrow K'(x', y')$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Dilatasi : perpindahan titik-titik suatu objek terhadap titik tertentu berdasarkan faktor pengali. Oleh karena terdapat faktor pengali, maka peristiwa dilatasi ini dapat mengakibatkan perubahan ukuran objek yang tetap sebangun. Apabila suatu titik M(x,y) mengalami dilatasi terhadap titik pusat (0,0) dengan faktor pengali k, maka akan dihasilkan koordinat M'(x',y'). Secara matematis, dapat dinyatakan sebagai berikut:

$$M(x, y) \rightarrow M'(x', y')$$

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

- Resize : konsep *resize* sebenarnya memiliki kesamaan dengan konsep dilatasi, akan tetapi pada *resize* hanya salah satu komponen pasangan sisi sejajar saja yang diubah titik koordinatnya. Apabila suatu titik M(x,y) mengalami proses *resize* untuk sisi horizontalnya terhadap titik pusat (0,0) dengan faktor pengali k, maka akan dihasilkan koordinat M'(x',y'). Secara matematis, dapat dinyatakan sebagai berikut:

$$M(x, y) \rightarrow M'(x', y')$$

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

- Shear : perpindahan koordinat dari salah satu titik sudut dari suatu bangun. Konsep perhitungan matematis dari *shear* sebenarnya sama dengan proses dilatasi dan *resize*, tetapi hanya dilakukan pada satu titik sudut saja.

Bab 2

Implementasi

A. Struktur Kode

Berikut ini merupakan struktur program yang telah diimplementasikan oleh penulis:

```
|—public
|—img
|—doc
|—src
|   App.css
|   App.tsx
|   index.css
|   main.tsx
|   vite-env.d.ts
|   webgl.ts
|
|—assets
|   react.svg
|   thumb.png
|   └─toolbar-icon
|       delete.png
|       line.png
|       load.png
|       move.png
|       polygon.png
|       rectangle.png
|       save.png
|       select.png
|       square.png
|
|—components
|   sidebar.tsx
|   toolbar.tsx
|
|—mandatory
|   addVertex.ts
|   changeColor.ts
|   create.ts
|   moveVertex.ts
|   removeVertex.ts
|   resize.ts
|   resizeLine.ts
```



Gambar 2.1 Struktur Implementasi Program

Dalam pengimplementasiannya, penulis menggunakan bahasa TypeScript untuk mengakses fungsionalitas dari fitur WebGL yang disediakan oleh *browser* serta melakukan manipulasi DOM (*Document Object Model*) dari halaman HTML sehingga *website* dapat bersifat interaktif. Selain itu, penulis juga menggunakan kakas React dan TailwindCSS dalam pembentukan tampilan antarmuka aplikasi.

B. Manual/Contoh Fungsionalitas Program

Program akan diawali dengan sebuah tampilan antarmuka sebagai berikut:



Gambar 2.2 Tampilan Antarmuka Program

Ketika hendak menggambar sebuah bangun geometri, maka tekan *icon* pada *toolbar* (bagian kiri layar) terlebih dahulu, sesuai dengan bentuk geometri yang diinginkan, misalnya jika hendak membuat sebuah persegi / *square* maka lakukan langkah berikut:



Gambar 2.3 Ilustrasi Pemilihan Icon Shape pada Toolbar

Selanjutnya tekan *canvas* di koordinat tertentu, maka bangun geometri akan terbentuk. Berikut ini adalah contoh pembentukan untuk tiap bangun geometri.

1. Membuat Garis

Klik *button* dengan *icon* garis/*line* kemudian klik koordinat tertentu dalam *canvas*. Titik koordinat tersebut akan digunakan sebagai nilai panjang dari garis.



Gambar 2.2.1 Ilustrasi Pembuatan Garis

2. Membuat Persegi

Klik *button* dengan *icon* persegi/*square* kemudian klik koordinat tertentu dalam *canvas*. Titik koordinat tersebut akan digunakan sebagai nilai ukuran dari persegi.



Gambar 2.2.2 Ilustrasi Pembuatan Persegi

3. Membuat Persegi Panjang

Klik *button* dengan *icon* persegi panjang/*rectangle* kemudian klik koordinat tertentu dalam *canvas*. Titik koordinat tersebut akan digunakan sebagai nilai ukuran dari persegi panjang.



Gambar 2.2.3 Ilustrasi Pembuatan Persegi Panjang

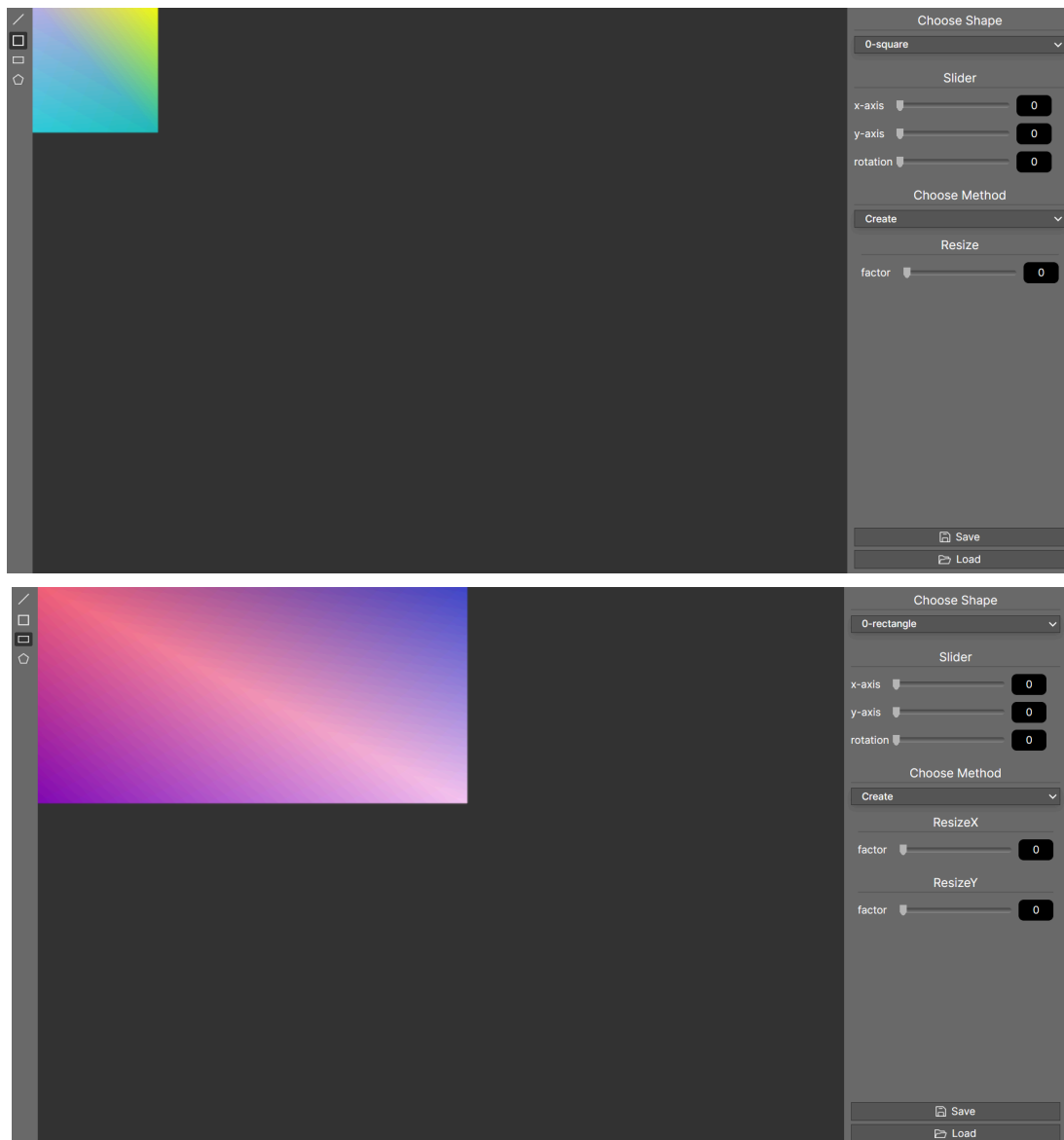
4. Membuat Poligon (implementasi *convex hull*)

Gambar 2.2.4 Ilustrasi Pembuatan Poligon

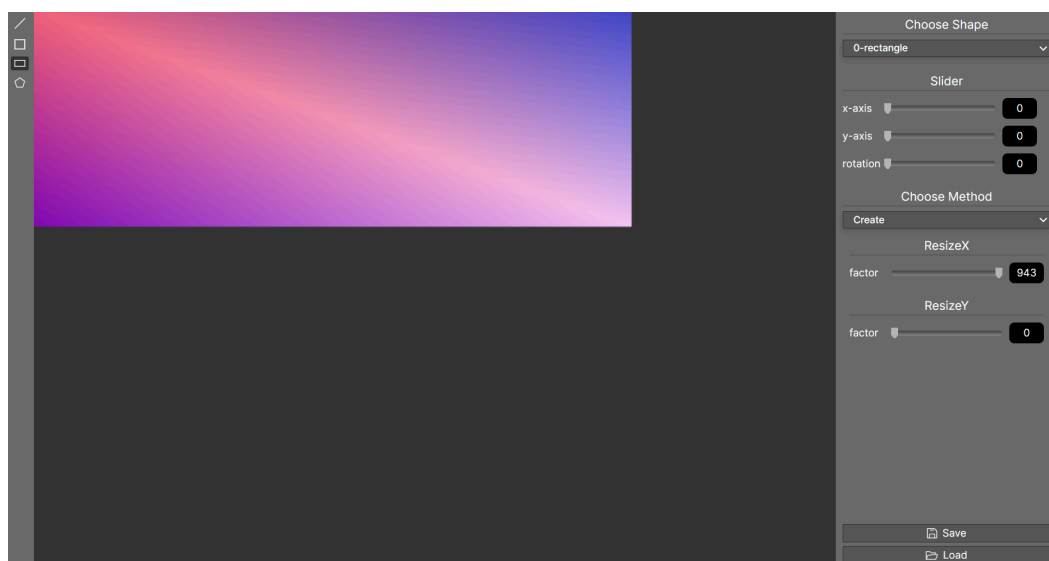
5. Mengubah ukuran Garis, Persegi dan Persegi Panjang

Buat gambar geometri dalam canvas, kemudian gunakan slider untuk melakukan *resize* dari bangun yang dipilih.





Gambar 2.2.5.1 Ilustrasi Pengubahan Ukuran (sebelum)



Gambar 2.2.5.2 Ilustrasi Pengubahan Ukuran (sesudah)

6. Mengubah jumlah titik sudut pada bentuk Poligon

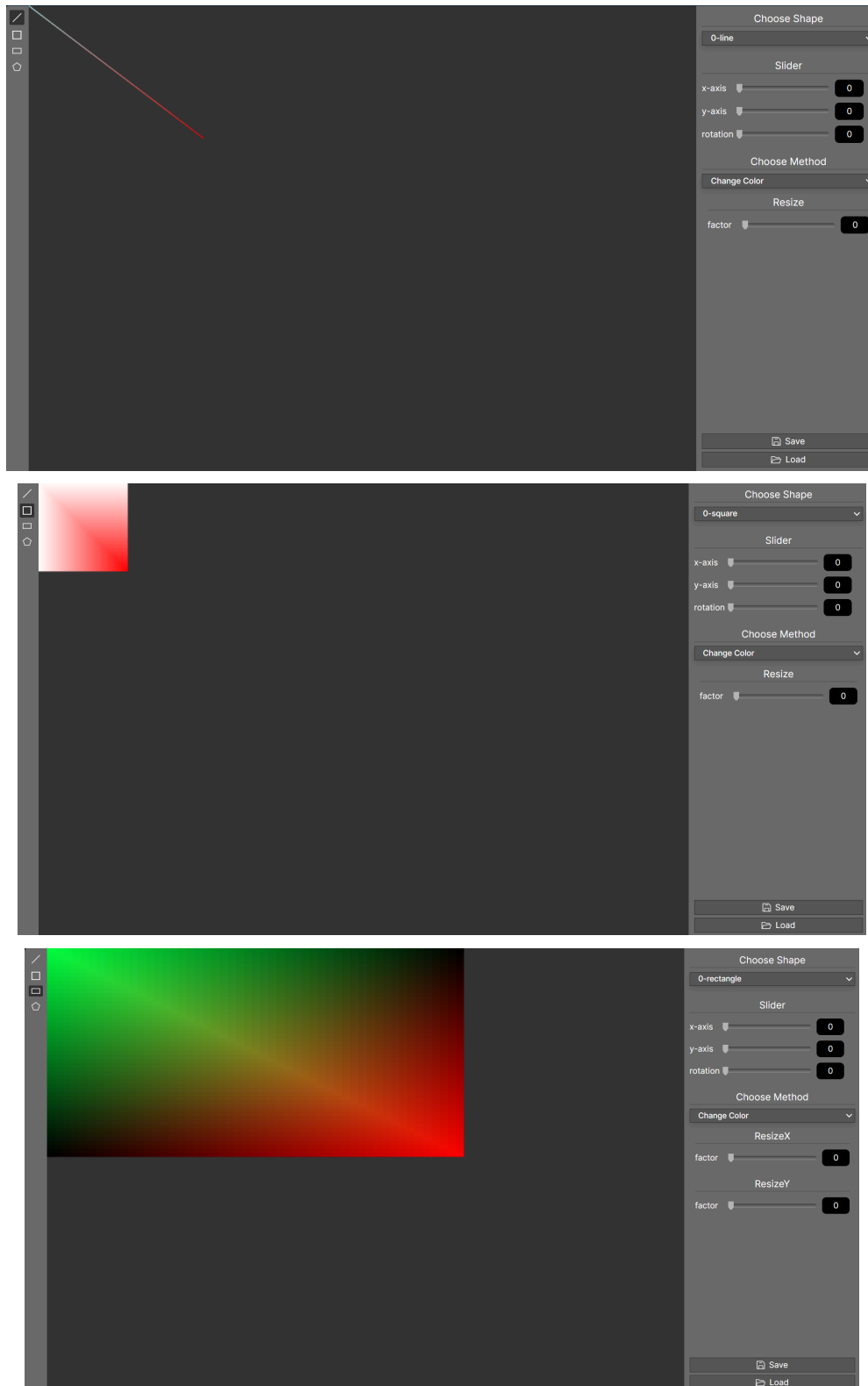
Gambar 2.2.6.1 Ilustrasi Pengubahan Jumlah Titik Sudut Poligon (sebelum)

Gambar 2.2.6.2 Ilustrasi Pengubahan Jumlah Titik Sudut Poligon (sesudah)

7. Mengubah warna titik sudut pada Garis, Persegi, Persegi Panjang, dan Poligon
Pilih bangun yang telah digambar di canvas, ganti *method* menjadi opsi *change color*. Pilih titik sudut yang hendak diubah warnanya, gunakan *color picker* untuk memilih warna yang sesuai.



Gambar 2.2.7.1 Ilustrasi Pengubahan Warna Titik Sudut (sebelum)



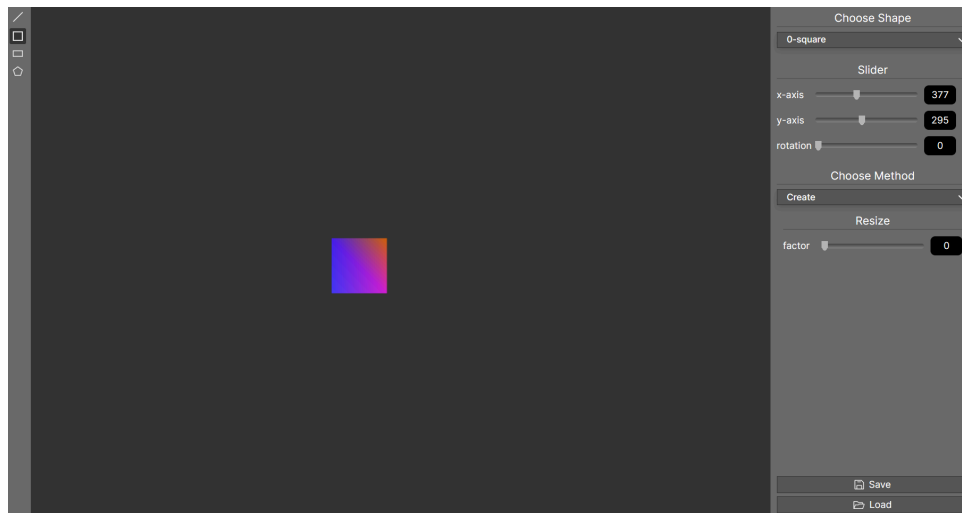
Gambar 2.2.7.2 Ilustrasi Pengubahan Warna Titik Sudut (sesudah)

8. Translasi (pergerakan objek)

Pilih bangun yang hendak ditranslasikan. Gunakan slider translasi untuk memindahkan koordinat dari suatu objek tertentu.

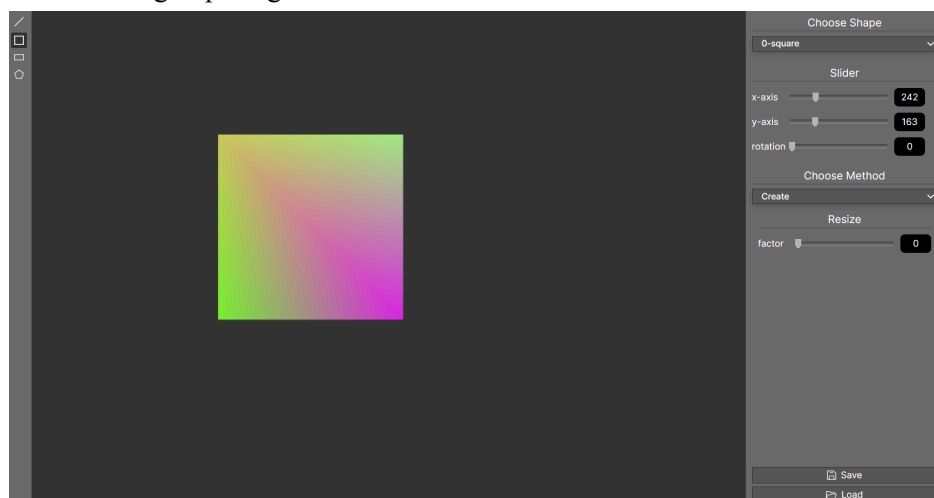


Gambar 2.2.8.1 Ilustrasi Translasi (sebelum)

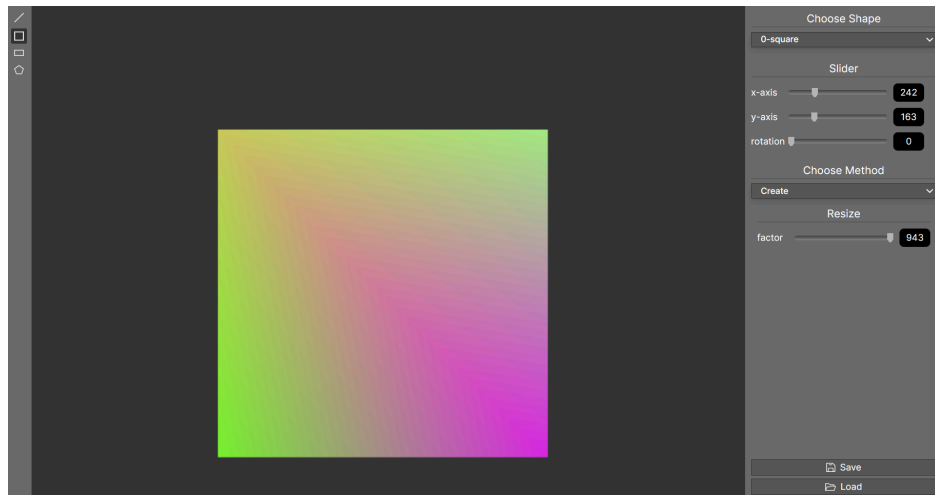


Gambar 2.2.8.2 Ilustrasi Translasi (sesudah)

9. Dilatasi (perbesaran atau pengecilan objek agar tetap sebangun)
Pilih *square* atau persegi yang hendak dilatasi. Gunakan slider *resize* untuk memperbesar ukuran dari bangun persegi.



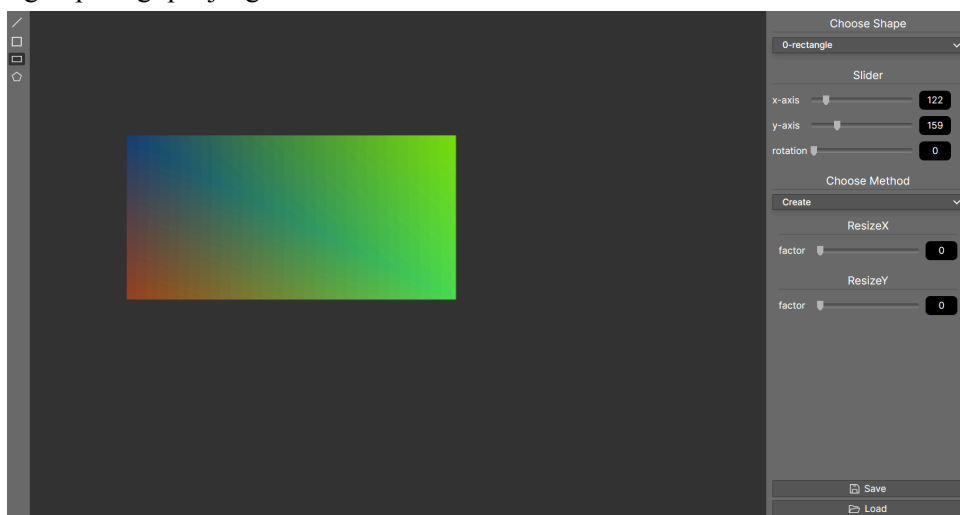
Gambar 2.2.9.1 Ilustrasi Dilatasi(sebelum)



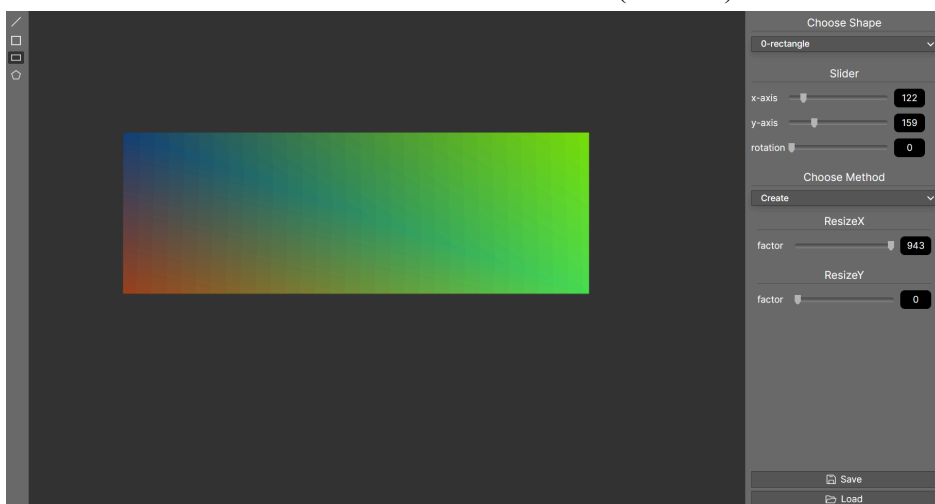
Gambar 2.2.9.2 Ilustrasi Dilatasi(sesudah)

10. Resize (perbesaran salah satu pasangan sisi sejajar dari objek)

Pilih *rectangle* atau persegi panjang yang hendak dilatasi. Gunakan slider *resizeX* untuk memperbesar panjang dari bangun persegi panjang atau *sliderY* untuk memperbesar lebar dari bangun persegi panjang.



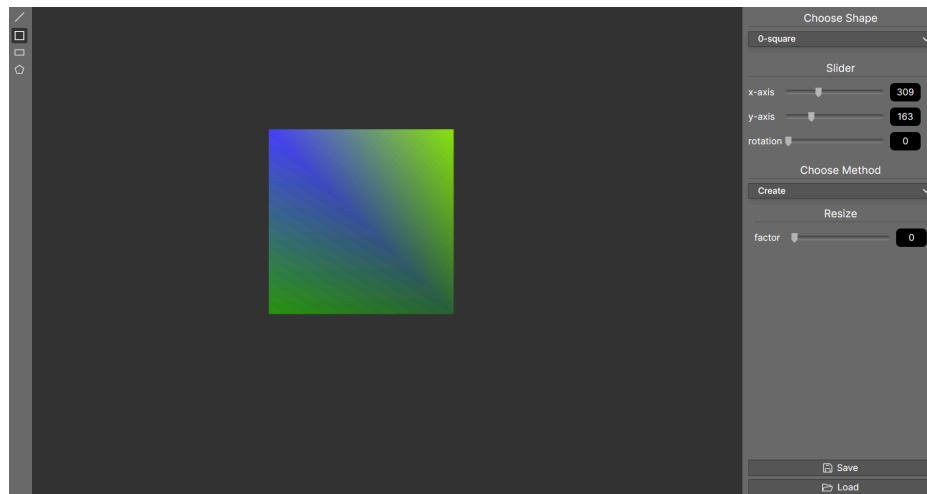
Gambar 2.2.10.1 Ilustrasi Resize (sebelum)



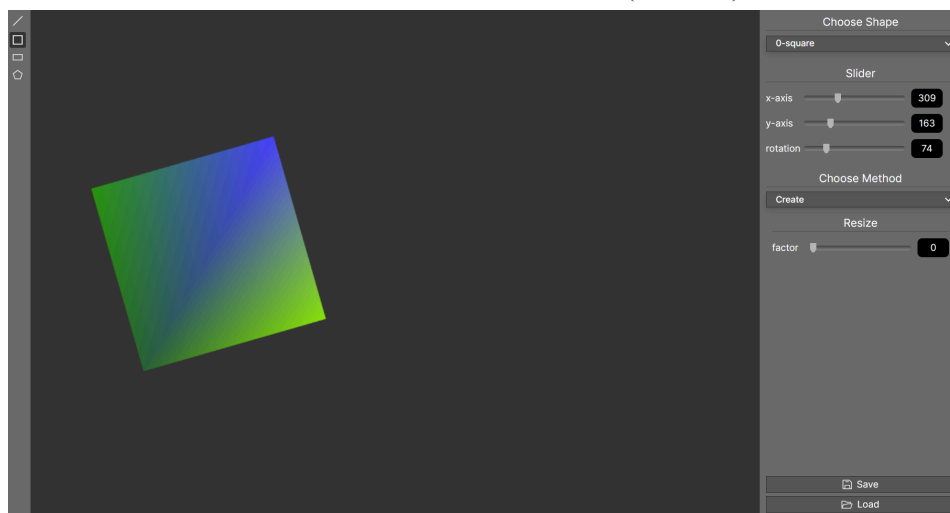
Gambar 2.2.10.2 Ilustrasi Resize (sesudah)

11. Rotasi (pemutaran objek)

Pilih bangun yang hendak dirotasikan. Gunakan slider rotasi untuk mengubah orientasi sudut dari bangun.



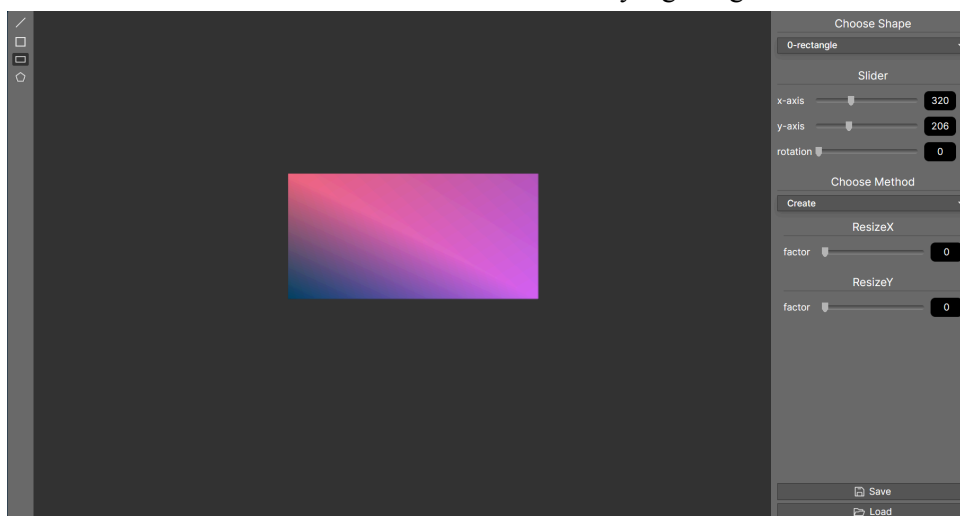
Gambar 2.2.11.1 Ilustrasi Rotasi (sebelum)



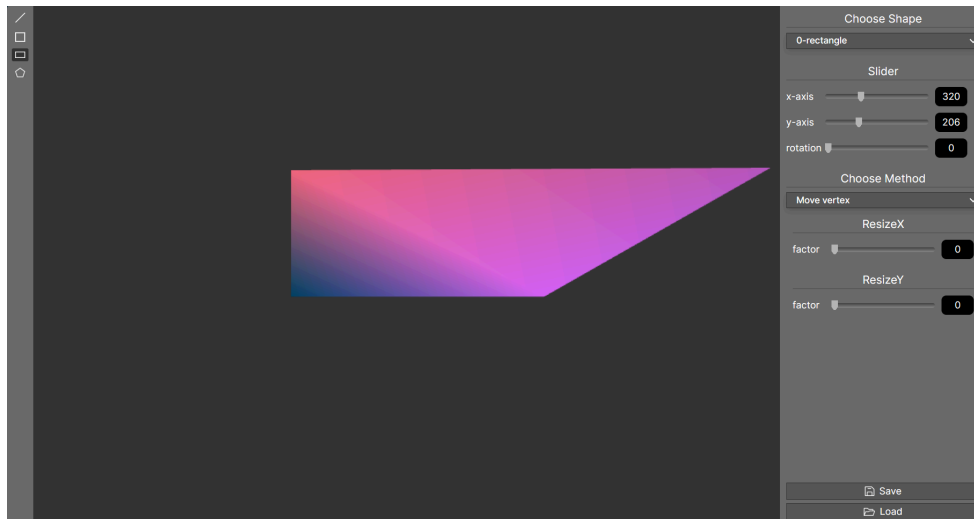
Gambar 2.2.11.2 Ilustrasi Rotasi (sesudah)

12. Shear (pemindahan salah satu titik sudut dari objek)

Pilih bangun yang hendak di-shear. Pilih metode *move vertex* dan tentukan titik sudut yang hendak diubah. Tarik titik sudut tersebut ke koordinat yang diinginkan.



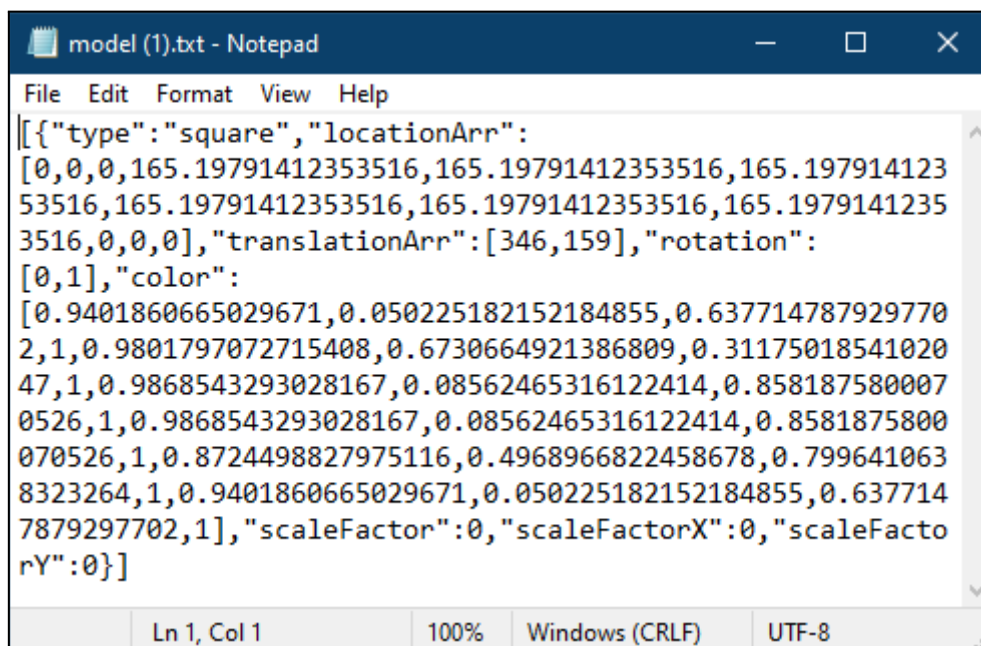
Gambar 2.2.12.1 Ilustrasi Shear(sebelum)



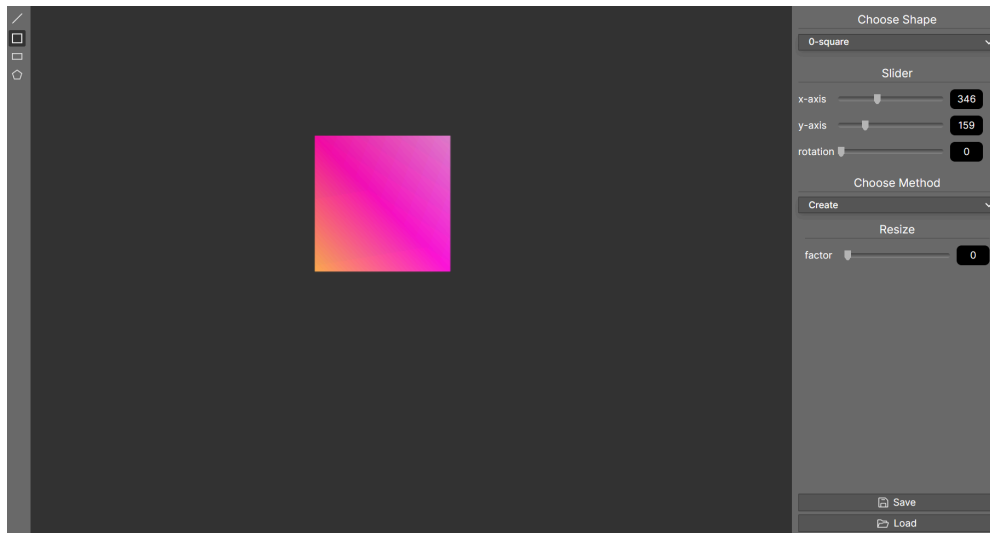
Gambar 2.2.12.2 Ilustrasi Shear(sesudah)

13. Export dan Import File

Tekan tombol *save* pada *sidebar* untuk mendownload model yang telah tergambar di *canvas*. Gunakan tombol *load* untuk memilih model tadi, dan hasil *load* gambar akan ditampilkan di *canvas*.



Gambar 2.2.13.1 Ilustrasi Export



Gambar 2.2.13.2 Ilustrasi Import

Lampiran

A. Pembagian Tugas

Berikut ini adalah daftar pembagian tugas yang dilakukan oleh kelompok Kami:

No	NIM	Nama	Tugas
1	13521119	Muhammad Rizky Sya'ban	<ul style="list-style-type: none">- Geometri Line- Geometri Polygon
2	13521121	Saddam Annais Shaquille	<ul style="list-style-type: none">- Init architecture- Geometri Square- Translasi- Rotasi- Change color- Move vertec- Choose Method- Choose Shape- Choose Icon Tools
33	13521166	Mohammad Rifqi Farhansyah	<ul style="list-style-type: none">- Init Frontend- Design UI- Frontend- Geometri Rectangle- Resize sebangun- Resize X and Y- Save & Load- Choose Icon Tools- Laporan

B. Pranala Terkait

Berikut ini adalah daftar pranala terkait pengerjaan tugas oleh kelompok kami:

- Pranala *design* UI :
<https://www.figma.com/file/QhDzOLPVtjWnDPlbajIXna/GrafKONNN?type=design&node-id=0%3A1&mode=design&t=KnDkXZo1sdJIswpZ-1>
- Pranala *repository* :
[GAIB20/tugas-besar-grafkom-1-grafkonn: tugas-besar-grafkom-1-grafkonn created by GitHub Classroom](#)