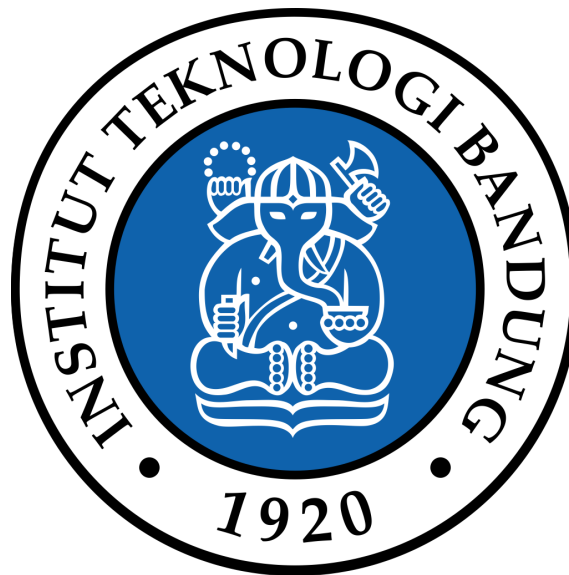


IF3260 - GRAFIKA KOMPUTER
LAPORAN TUGAS 1
2D Web Based CAD (Computer-Aided Design)



Kelompok 5 / Kelas 3
Jundullah - 13518027
Fikra Hadi Ramadhan - 13518036
Anna Elvira Hartoyo - 13518045

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2021

BAB I

DESKRIPSI

Pada tugas 1 kali ini, kami membuat sebuah website 2D CAD (Computer-Aided Design) dengan menggunakan WebGL murni tanpa library atau framework apa pun. Secara garis besar website ini terdiri dari beberapa tombol untuk melakukan perintah dan menjalankan fitur-fitur yang ada dan sebuah canvas tempat pengguna menggambar dan memodifikasi model yang telah digambar.

Website memiliki fitur-fitur dan interaksi sebagai berikut:

1. Menggambar atau membuat model

Model yang dapat digambar meliputi:

- Garis
- Persegi
- Poligon (dengan input jumlah sudut dari pengguna).

2. Menyimpan definisi model ke dalam sebuah file

Website memiliki fitur untuk menyimpan (save) seluruh model yang terdapat pada canvas dalam bentuk file json yang dapat dengan mudah diedit nilai dari setiap atribut objeknya.

3. Membuka file hasil penyimpanan model

Website dapat melakukan load terhadap file json hasil penyimpanan model. Setelah data berhasil di load, model akan ditampilkan pada canvas.

4. Menggeser titik kontrol atau simpul dengan mouse

Pengguna dapat menggeser model garis, persegi, maupun poligon yang telah digambar sehingga berpindah posisi sesuai keinginan pengguna.

5. Mengubah panjang garis

Sebuah model garis yang telah digambar dapat diubah panjangnya menjadi lebih panjang atau lebih pendek.

6. Mengubah panjang sisi persegi

Sebuah model persegi yang telah digambar dapat diubah panjang sisinya secara proporsional sehingga menghasilkan persegi yang lebih besar atau lebih kecil.

7. Mengubah warna poligon

Sebuah model poligon yang telah digambar dapat diubah warnanya menjadi sesuai input dari pengguna

8. Menu help

Menu help digunakan untuk memberi petunjuk (user manual) bagi pengguna baru untuk melakukan operasi dan memanfaatkan fitur-fitur yang ada tanpa perlu bertanya.

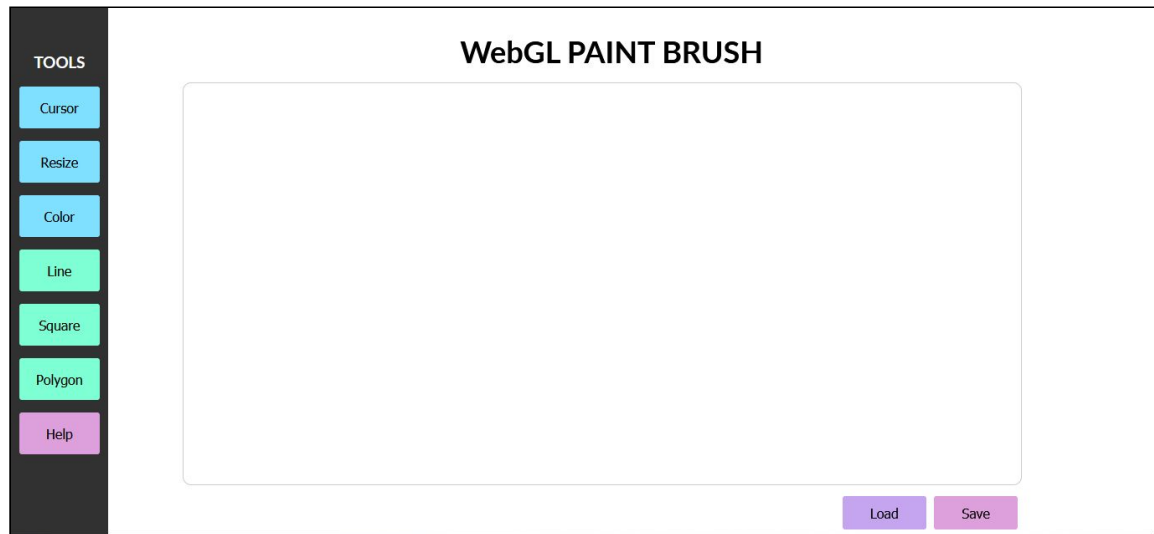
BAB II

HASIL

Berikut adalah hasil website 2D CAD yang telah kami buat.

1. Canvas kosong

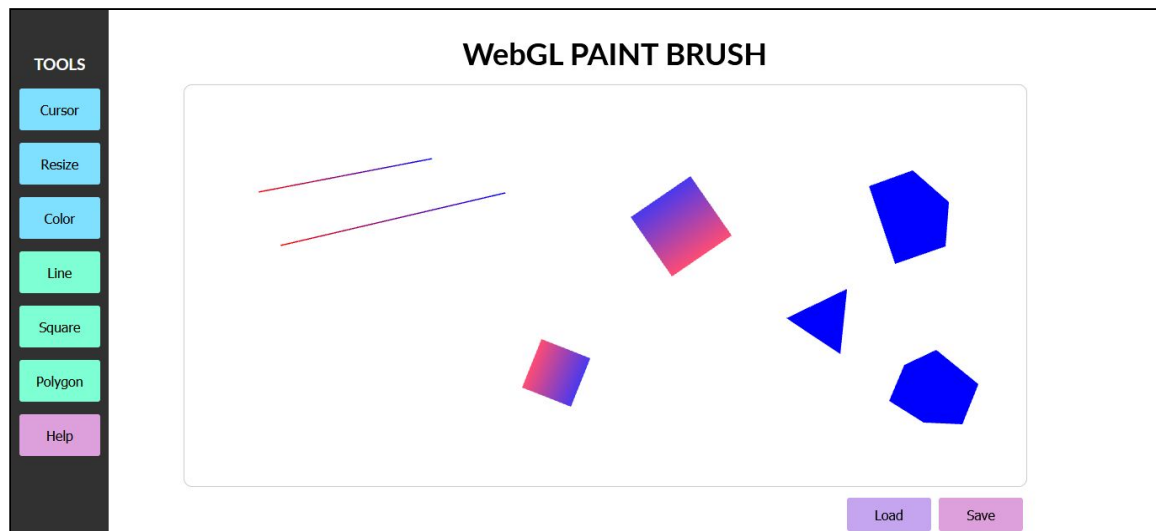
Berikut canvas kosong yang berhasil ditampilkan pada saat pertama kali membuka halaman web



Canvas dibuat menggunakan HTML5 canvas dengan ukuran lebar 1000px dan tinggi 480px. Canvas ini kemudian diinisialisasi dengan GL context sehingga proses penggambaran dan operasi webGL lainnya dapat dilakukan di atas canvas tersebut.

2. Menggambar model

Terdapat tiga jenis model yang dapat digambar pada web yang kami buat, yaitu garis (line), persegi (square), dan poligon. Berikut adalah hasil penggambaran model garis, square, dan poligon pada canvas.



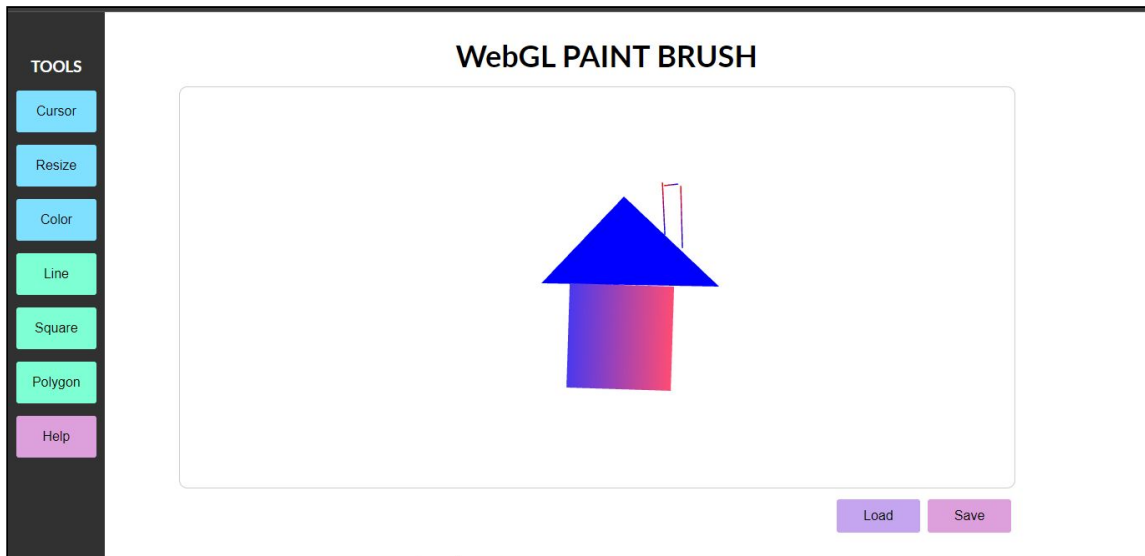
Proses penggambaran setiap model pada prinsipnya sama, yaitu dengan cara mendeteksi posisi koordinat x dan y ketika mouse mengklik suatu titik pada canvas. **Koordinat** tersebut kemudian disimpan dalam sebuah **array vertices** dan warna dari setiap vertex disimpan dalam sebuah array colors. Setiap **objek** yang digambar juga disimpan dalam sebuah array of objects dengan atribut objek terdiri atas nama, gl mode, offset, dan count.

Ketika vertex pada suatu objek selesai didefinisikan, array vertices dan array colors tersebut akan dikirimkan ke buffer untuk kemudian digambarkan pada canvas menggunakan gl mode yang sesuai, yaitu gl.LINES untuk garis, gl.TRIANGLE_STRIP untuk persegi, dan gl.TRIANGLE_FAN untuk poligon.

3. Menyimpan model ke dalam sebuah file

Seluruh objek yang sudah digambar pada canvas dapat disimpan ke dalam sebuah file dengan format .json yang dapat didownload ke local storage komputer pengguna. Informasi yang terdapat pada file tersebut adalah jenis objek (line/square/polygon), posisi titik koordinat, dan warna setiap titik koordinat. Selanjutnya file ini dapat diedit nilai atributnya menggunakan text editor.

Berikut adalah contoh hasil objek yang digambar pada canvas dan file .json yang disimpan.



```

1  [
2  {
3    "name": "square",
4    "vertices": [
5      {
6        "xAxis": -0.2326875000000005,
7        "yAxis": -0.09583333333333344,
8        "color": {
9          "r": 0.29,
10         "g": 0.223,
11         "b": 0.937
12       }
13     },
14     {
15       "xAxis": -0.03268749999999998,
16       "yAxis": -0.12916666666666665,
17       "color": {
18         "r": 1,
19         "g": 0.301,
20         "b": 0.458
21       }
22     },
23     {
24       "xAxis": -0.21668750000000012,
25       "yAxis": 0.32083333333333334,
26       "color": {
27         "r": 0.29,
28         "g": 0.223,
29         "b": 0.937
30       }
31     },
32     {
33       "xAxis": -0.01668750000000004,
34       "yAxis": 0.28750000000000002,
35       "color": {
36         "r": 1,
37         "g": 0.301,
38         "b": 0.458
39       }
40     }
41   ]
42 }

```

```

41   ],
42 },
43 {
44   "name": "polygon",
45   "vertices": [
46     {
47       "xAxis": -0.2866875,
48       "yAxis": 0.32916666666666667,
49       "color": {
50         "r": 0,
51         "g": 0,
52         "b": 1
53       }
54     },
55     {
56       "xAxis": 0.05931250000000001,
57       "yAxis": 0.2875,
58       "color": {
59         "r": 0,
60         "g": 0,
61         "b": 1
62       }
63     },
64     {
65       "xAxis": -0.07868750000000002,
66       "yAxis": 0.7333333333333334,
67       "color": {
68         "r": 0,
69         "g": 0,
70         "b": 1
71       }
72     }
73   ]
74 },
75 {
76   "name": "line",
77   "vertices": [
78     {
79       "xAxis": -0.26668749999999997,
80       "yAxis": 0.675,

```

```

{
  "name": "line",
  "vertices": [
    {
      "xAxis": -0.26668749999999997,
      "yAxis": 0.675,
      "color": {
        "r": 1,
        "g": 0,
        "b": 0
      }
    },
    {
      "xAxis": 0.1933125,
      "yAxis": 0.8041666666666667,
      "color": {
        "r": 0,
        "g": 0,
        "b": 1
      }
    }
  ]
}

```

Proses yang dilakukan adalah mengubah data dari array of objects, array vertices, dan array colors ke dalam bentuk yang lebih mudah dibaca kemudian menyimpannya sebagai objects baru dalam file .json.

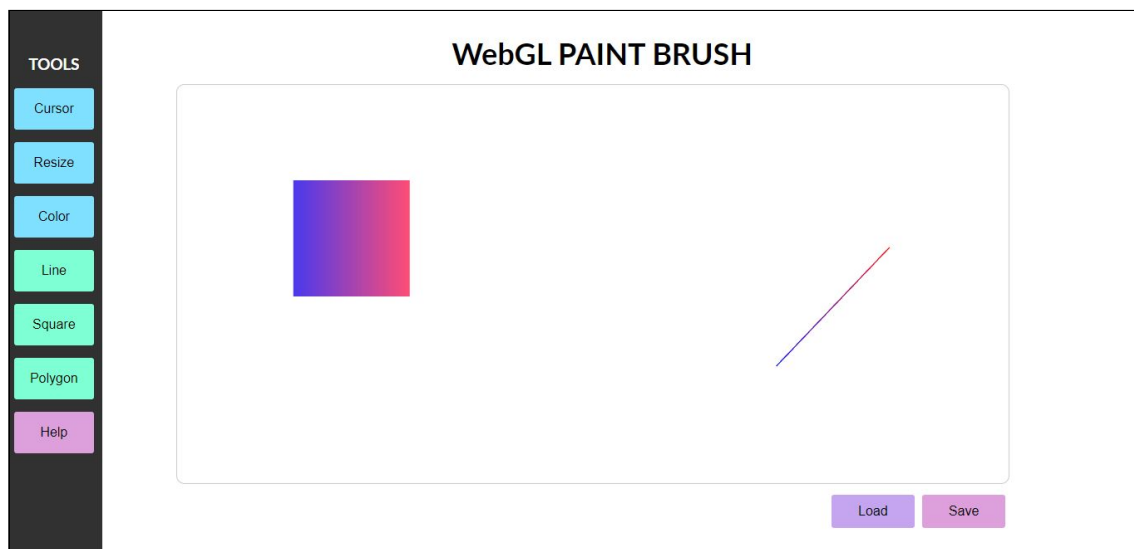
4. Membuka file hasil penyimpanan model

Web dapat menerima input file dengan format .json seperti pada fitur nomor 3, kemudian menampilkan isi file tersebut ke dalam penggambaran di canvas.

Berikut contoh file .json yang di load beserta hasil penggambarannya pada canvas

```
[
  {
    "name": "square",
    "vertices": [
      {
        "xAxis": -0.7206874999999999,
        "yAxis": -0.0625,
        "color": {
          "r": 0.29,
          "g": 0.223,
          "b": 0.937
        }
      },
      {
        "xAxis": -0.4406875,
        "yAxis": -0.0625,
        "color": {
          "r": 1,
          "g": 0.301,
          "b": 0.458
        }
      },
      {
        "xAxis": -0.7206874999999999,
        "yAxis": 0.5208333333333331,
        "color": {
          "r": 0.29,
          "g": 0.223,
          "b": 0.937
        }
      },
      {
        "xAxis": -0.4406875,
        "yAxis": 0.5208333333333331,
        "color": {
          "r": 1,
          "g": 0.301,
          "b": 0.458
        }
      }
    ]
  },
  {
    "name": "line",
    "vertices": [
      {
        "xAxis": 0.7133125,
        "yAxis": 0.18333333333333335,
        "color": {
          "r": 1,
          "g": 0,
          "b": 0
        }
      },
      {
        "xAxis": 0.4413125,
        "yAxis": -0.41250000000000001,
        "color": {
          "r": 0,
          "g": 0,
          "b": 1
        }
      }
    ]
  }
]
```

```
[
  {
    "name": "square",
    "vertices": [
      {
        "xAxis": -0.7206874999999999,
        "yAxis": -0.0625,
        "color": {
          "r": 0.29,
          "g": 0.223,
          "b": 0.937
        }
      },
      {
        "xAxis": -0.4406875,
        "yAxis": -0.0625,
        "color": {
          "r": 1,
          "g": 0.301,
          "b": 0.458
        }
      },
      {
        "xAxis": -0.7206874999999999,
        "yAxis": 0.5208333333333331,
        "color": {
          "r": 0.29,
          "g": 0.223,
          "b": 0.937
        }
      },
      {
        "xAxis": -0.4406875,
        "yAxis": 0.5208333333333331,
        "color": {
          "r": 1,
          "g": 0.301,
          "b": 0.458
        }
      }
    ]
  },
  {
    "name": "line",
    "vertices": [
      {
        "xAxis": 0.7133125,
        "yAxis": 0.18333333333333335,
        "color": {
          "r": 1,
          "g": 0,
          "b": 0
        }
      },
      {
        "xAxis": 0.4413125,
        "yAxis": -0.41250000000000001,
        "color": {
          "r": 0,
          "g": 0,
          "b": 1
        }
      }
    ]
  }
]
```

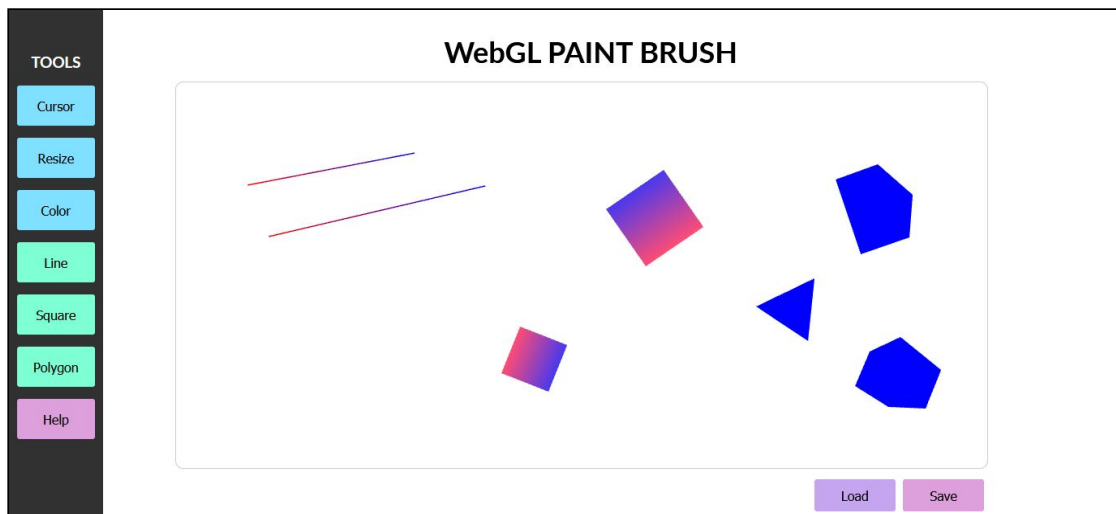


Proses yang dilakukan adalah menerima data masukan .json kemudian mengubah data tersebut menjadi bentuk yang sesuai dan dapat dibaca oleh WebGL. Selanjutnya dilakukan proses pengisian array of objects, array vertices dan colors. Array vertices dan colors dikirimkan ke framebuffer, sementara informasi pada array of objects digunakan untuk menggambarkan objek dari titik-titik pada array vertices.

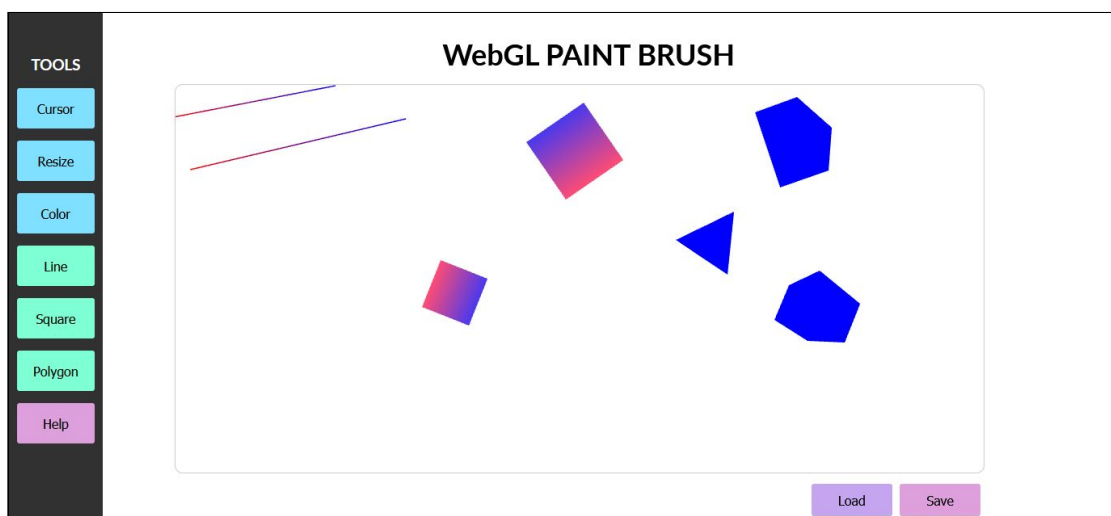
5. Menggeser titik kontrol atau simpul dengan mouse

- Seluruh model

Berikut adalah posisi awal model yang terdefinisi pada canvas.

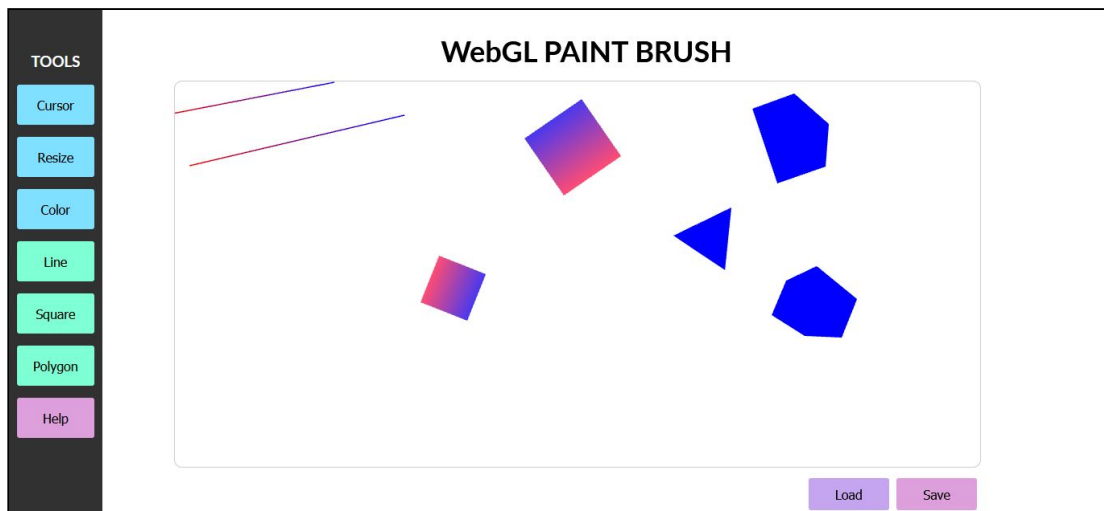


Dilakukan proses pemindahan titik kontrol atau simpul menggunakan mouse untuk seluruh model dengan cara melakukan drag and drop pada bagian canvas yang kosong, sehingga posisi model setelah dipindahkan menjadi sebagai berikut.

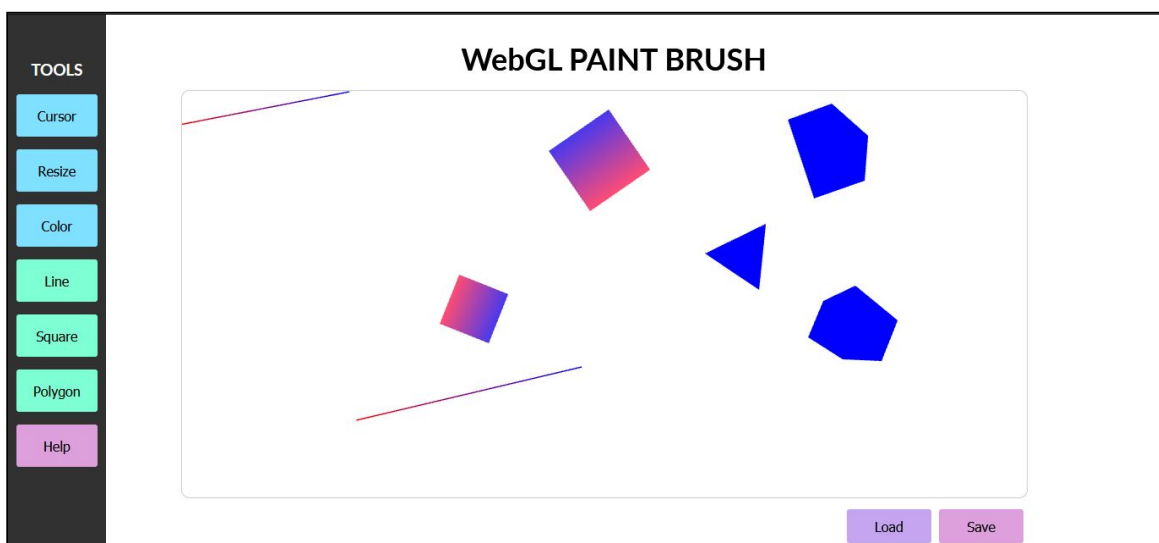


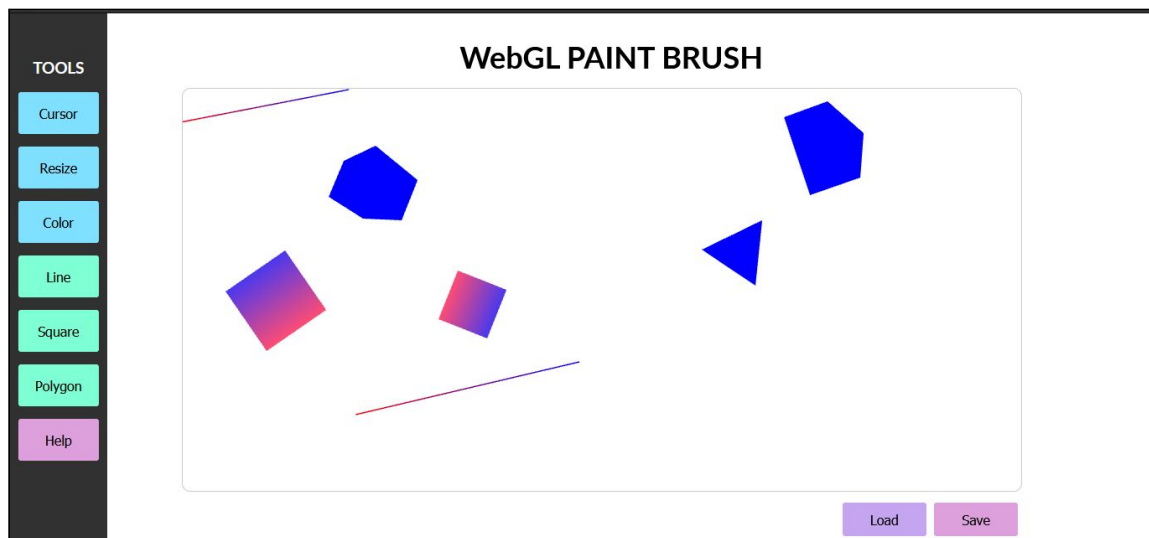
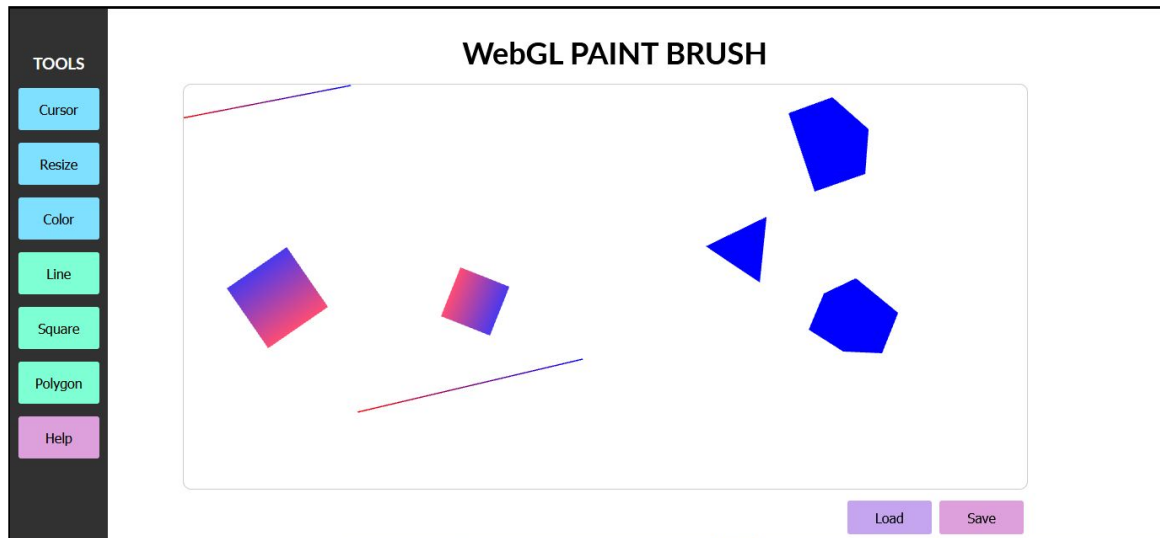
- Satu buah model

Berikut adalah posisi awal model yang terdefinisi pada canvas.



Dilakukan pemindahan titik kontrol atau simpul menggunakan mouse untuk salah satu objek square, poligon, dan line dengan cara melakukan drag and drop salah satu titik pada objek, sehingga diperoleh hasil seperti tiga gambar di bawah.





Proses yang dilakukan adalah ketika mouse di drag, program akan mencatat seberapa jauh titik koordinat tempat mouse berada sekarang (V_T). Kemudian akan diperoleh nilai deltaX dan deltaY sebagai banyaknya perpindahan suatu titik dengan formula sebagai berikut:

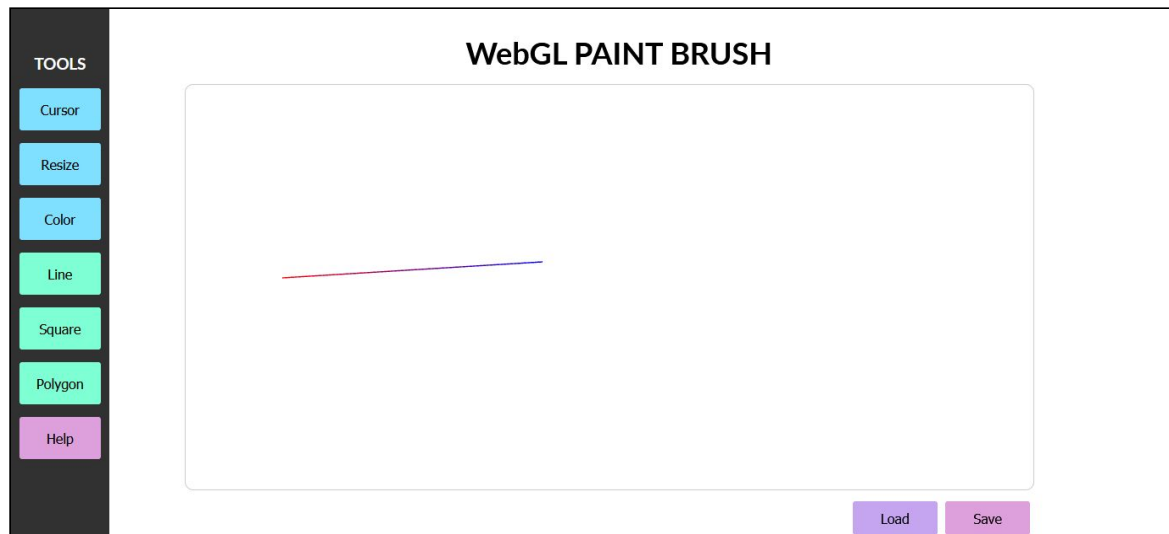
$$\text{deltaX} = V_{Tx} - V_{0x}$$

$$\text{deltaY} = V_{Ty} - V_{0y}$$

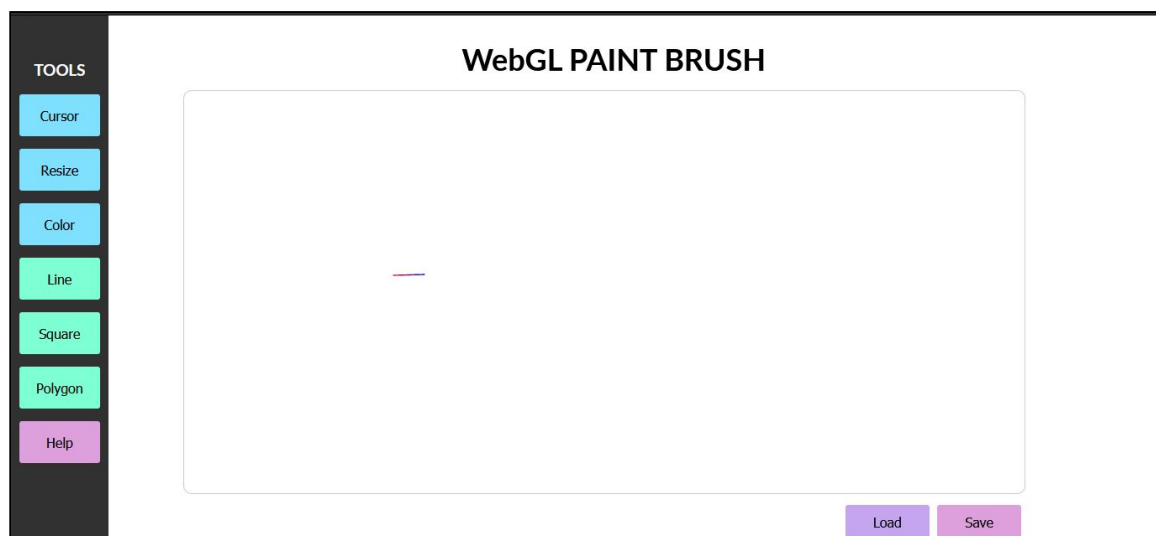
Selanjutnya seluruh titik pada model akan tambah sebesar nilai ΔX dan ΔY lalu data titik pada array vertices akan diupdate dan kemudian dikirimkan ke framebuffer.

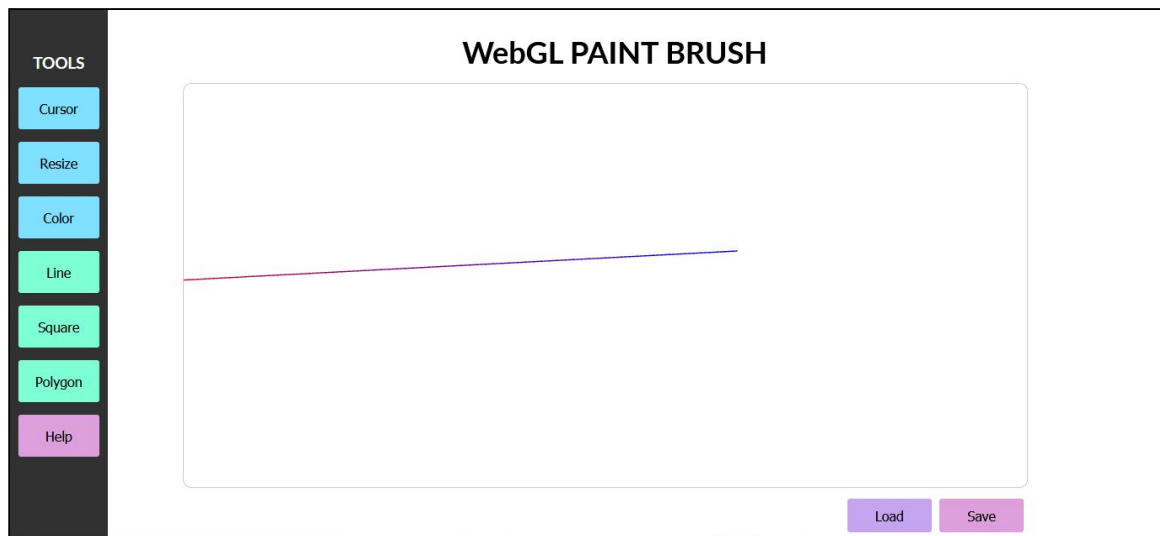
6. Mengubah panjang garis

Berikut adalah posisi awal model yang terdefinisi pada canvas.



Kemudian dilakukan perubahan panjang garis dengan cara melakukan drag and drop salah satu titik pada garis. Perubahan panjang dapat dilakukan untuk mengurangi panjang garis maupun menambah panjang garis seperti yang terlihat pada kedua gambar hasil berikut.





Proses yang dilakukan adalah dengan mencari koordinat titik tengah untuk model garis yang dipilih. Kemudian ketika mouse di drag, program akan mencatat seberapa jauh titik koordinat tempat mouse berada sekarang dengan titik tengah model garis lalu membaginya dengan jarak antara titik awal dengan titik tengah model, sehingga diperoleh nilai skala sebagai berikut:

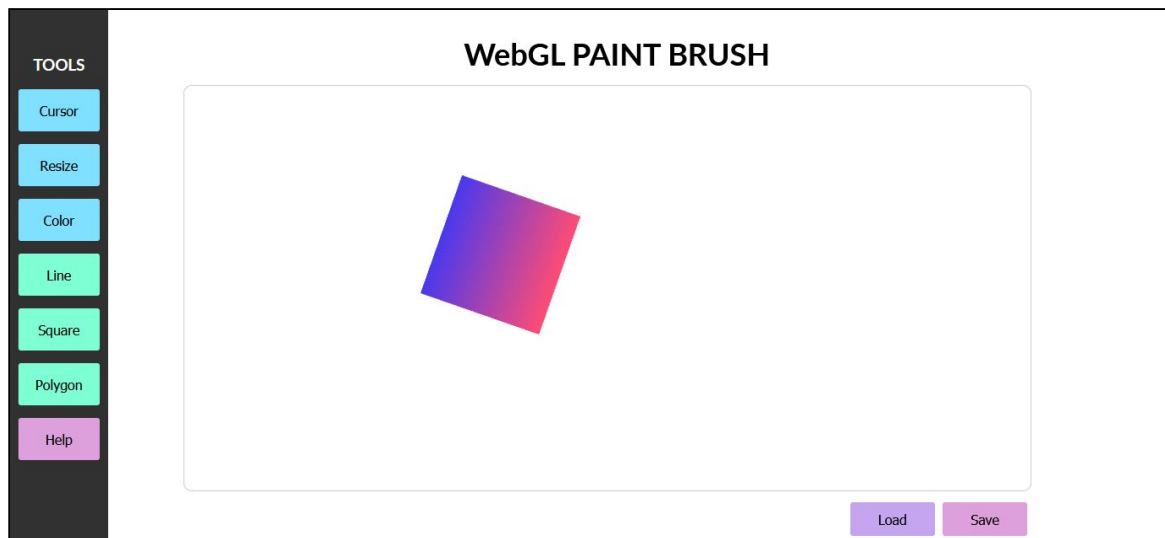
$$scaleX = \frac{V0x - centerX}{VTx - centerX}$$

$$scaleY = \frac{V0y - centerY}{VTy - centerY}$$

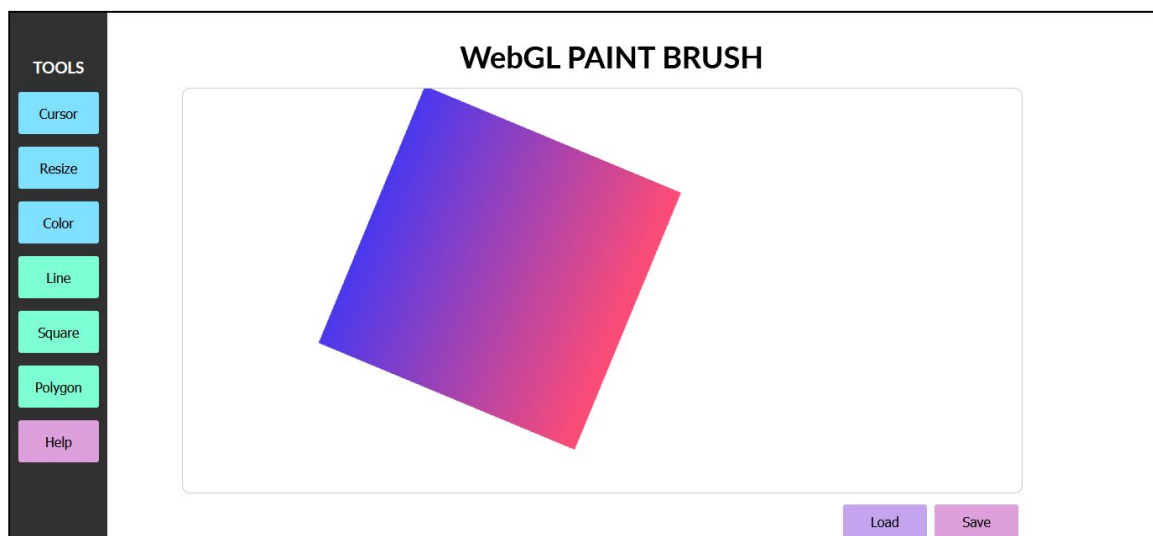
Selanjutnya seluruh titik pada model akan digeser sehingga titik tengah menjadi berada pada koordinat (0,0) kemudian baru dilakukan perkalian dengan nilai scale X untuk koordinat X dan scale Y untuk koordinat Y. Setelah di-scale barulah titik pusat model dikembalikan ke asal lalu data titik pada array vertices akan di update dan kemudian dikirimkan ke framebuffer.

7. Mengubah panjang sisi persegi

Berikut adalah posisi awal model yang terdefinisi pada canvas.



Kemudian dilakukan perubahan panjang sisi persegi dengan cara melakukan drag and drop salah satu titik pada persegi. Perubahan panjang sisi dapat dilakukan untuk mengurangi panjang sisi maupun menambah sisi persegi seperti yang terlihat pada kedua gambar hasil berikut.

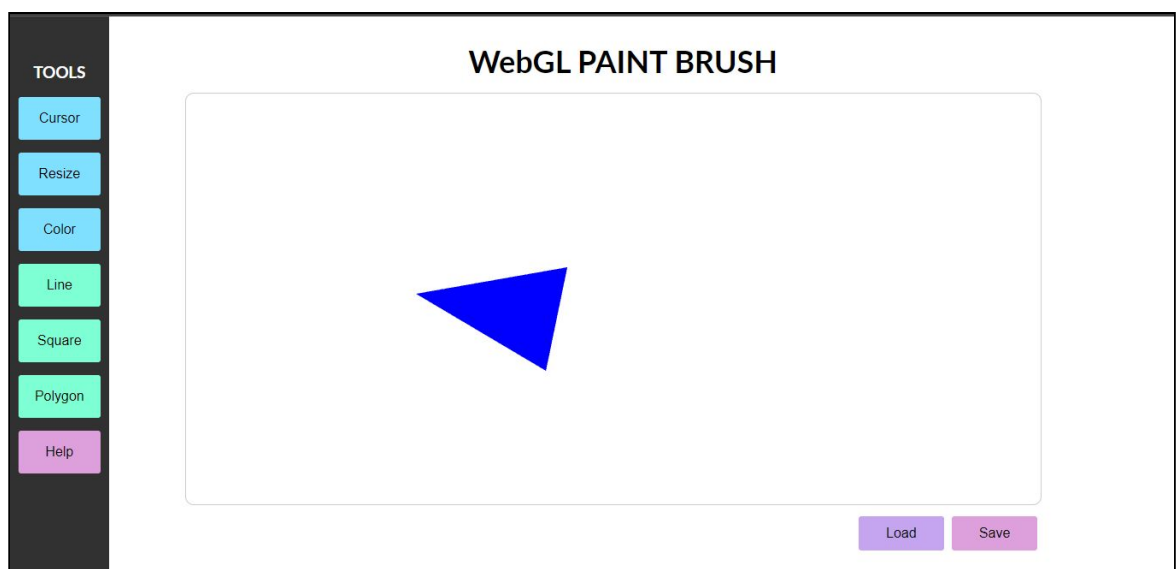




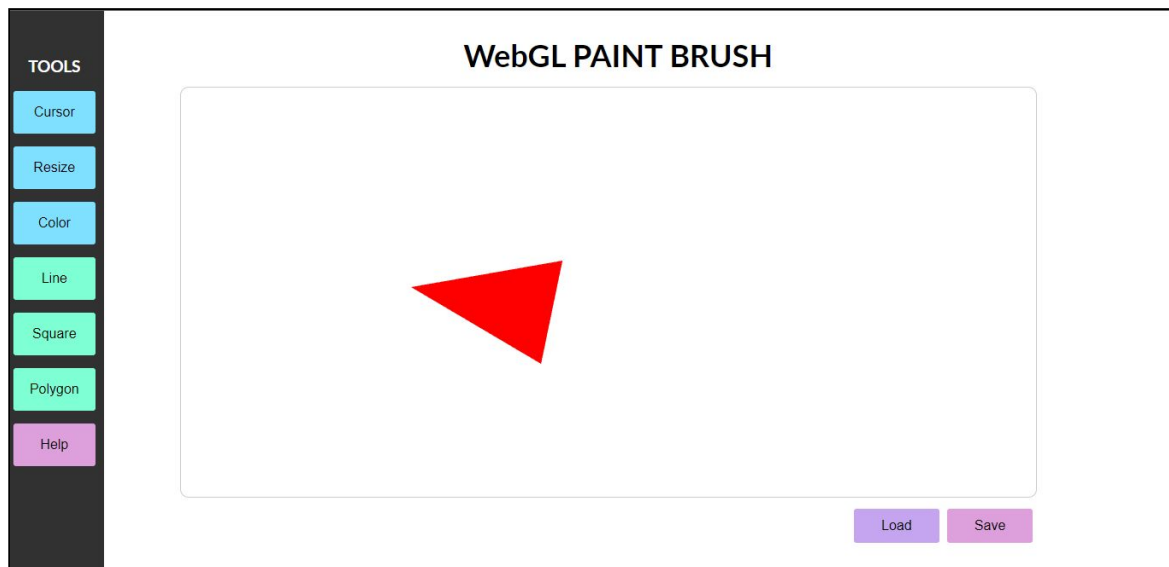
Proses yang dilakukan sama seperti proses pada scaling garis. Namun hanya dua titik saja yang dilakukan scaling lalu kedua titik tersebut akan digunakan untuk menghitung posisi dua titik lainnya pada persegi.

8. Mengubah warna poligon

Berikut adalah warna awal ketika poligon digambar, yaitu dengan value Red = 0, Green = 0, dan Blue = 255.



Kemudian dilakukan perubahan warna dengan menu color dan memilih salah satu titik sudut poligon. Setelah pengguna menginputkan warna baru dalam format RGB, misalnya Red = 255, Green = 0, Blue = 0, maka diperoleh hasil perubahan warna poligon sebagai berikut.



Proses yang dilakukan adalah dengan mengupdate value pada array colors dari objek poligon yang dipilih menjadi sesuai dengan value RGB yang diinputkan oleh pengguna.

9. Menu help

Berikut tampilan hasil halaman help ketika menu help pada toolbar ditekan.

Petunjuk Penggunaan

Tombol	Deskripsi dan Penggunaan
Cursor	Menggeser objek pada canvas dengan mouse.
Resize	Mengubah panjang garis/persegi dengan klik dan tahan salah satu sudut lalu geser sesuai ukuran yg diinginkan.
Color	Mengubah warna dari polygon yang dibuat dengan cara pilih salah satu sudut polygonnya, lalu pilih warna yang diinginkan dengan angka RGB.
Line	Membuat garis dengan 2 klik. klik pertama ke klik kedua merupakan panjang garis yang dibuat.
Square	Membuat persegi dengan 2 klik. klik pertama ke klik kedua merupakan panjang sisi yang dibuat.
Polygon	Membuat segi banyak. Silahkan isi banyaknya sisi yang diinginkan, lalu arahkan mouse ke papan, lalu silahkan klik sebanyak sisi yang dimasukkan. Klik disini berarti titik sudut setiap sisinya.
Load	Mengupload file sebelumnya yg sudah dibuat.
Save	Menyimpan hasil kerja pada canvas yang sudah dibuat.

BAB III

MANUAL DAN CONTOH FUNGSIONALITAS

Prerequisites

- Web browser yang mendukung WebGL, HTML5, dan javascript.

Installation

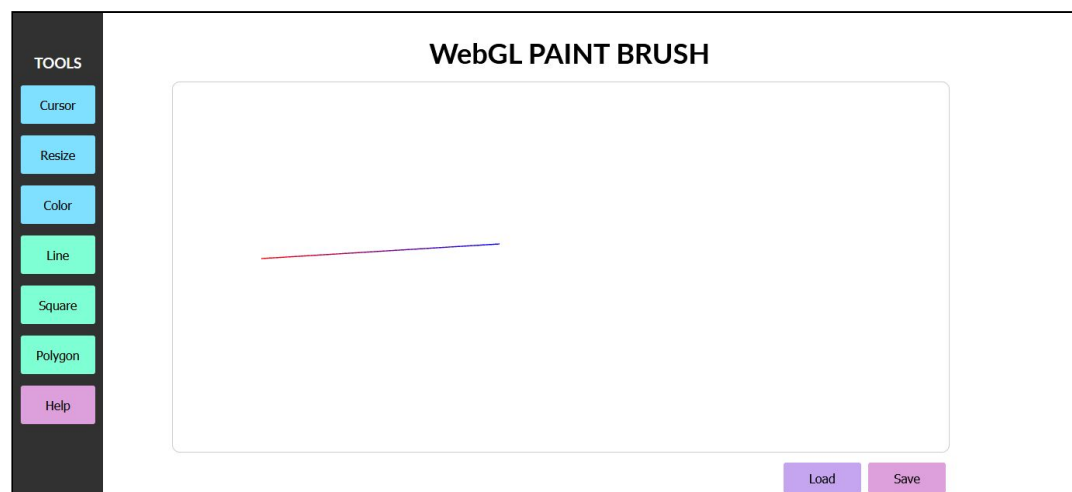
1. Clone repository berikut <https://github.com/annaelvira24/grafkom-project1>
2. Buka file index.html yang terdapat pada folder src menggunakan web browser Anda.
3. Pastikan canvas berhasil ditampilkan pada halaman web dan tidak ada pesan error.

Manual penggunaan fitur

1. Menggambar atau membuat model

- **Garis**

- 1) Klik menu “Line” yang terdapat pada toolbar di sebelah kiri.
- 2) Arahkan kursor ke area canvas.
- 3) Klik pada area canvas. Titik yang Anda klik akan menjadi titik awal dari garis yang akan Anda buat
- 4) Klik lagi pada area kanvas berbeda. Titik kedua tersebut akan menjadi titik akhir garis, sehingga akan terlihat sebuah garis yang telah tergambar pada canvas.



- **Persegi**

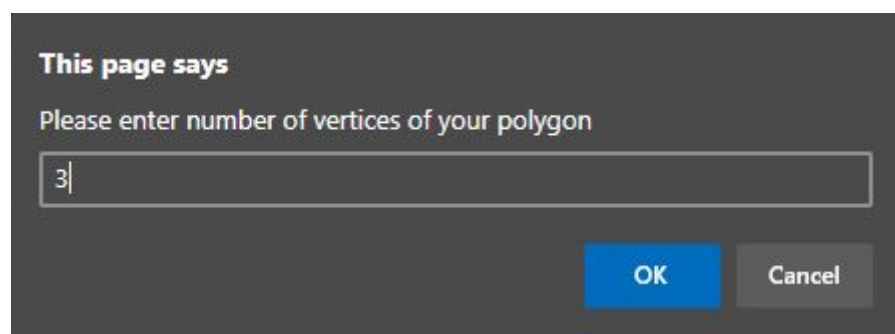
- 1) Klik menu “Square” yang terdapat pada toolbar di sebelah kiri.
- 2) Arahkan kursor ke area canvas.

- 3) Klik pada area canvas. Titik yang Anda klik akan menjadi salah satu sudut pada persegi yang akan Anda buat
- 4) Klik lagi pada area kanvas berbeda. Titik kedua tersebut menjadi sudut lain pada persegi. Kedua titik yang Anda klik tadi akan membentuk sebuah sisi persegi lalu secara otomatis ketiga sisi lainnya juga akan terbentuk, sehingga akan terlihat sebuah persegi (bujur sangkar dengan panjang keempat sisi sama) yang telah tergambar pada canvas.

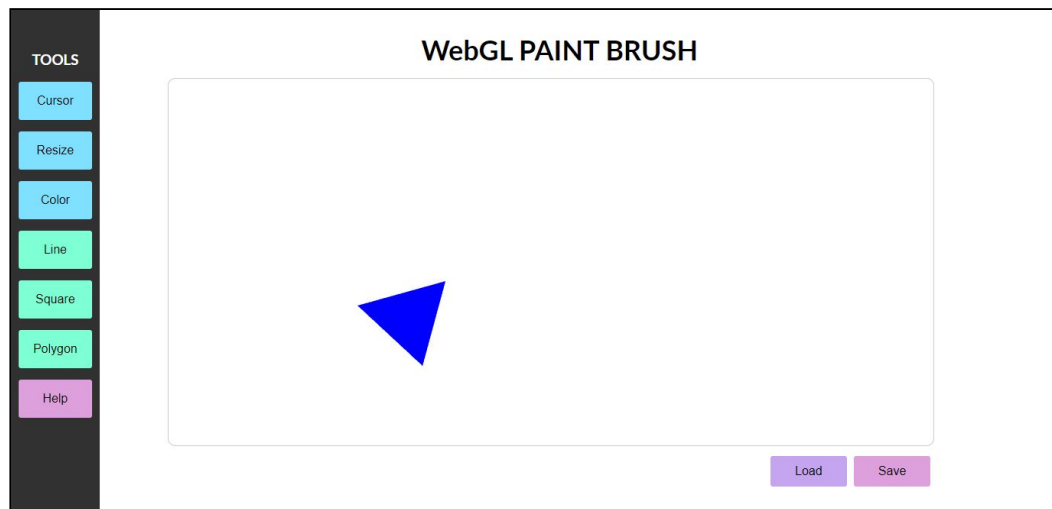


- **Poligon**

- 1) Klik menu “Polygon” yang terdapat pada toolbar di sebelah kiri.
- 2) Masukkan jumlah sudut dari poligon yang ingin Anda gambar pada pop up windows yang muncul lalu klik OK.



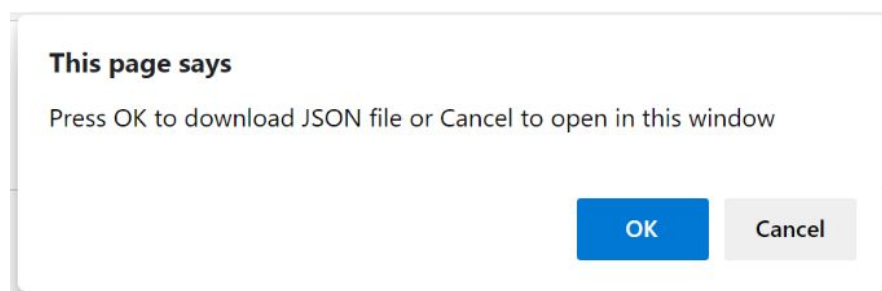
- 3) Arahkan kursor ke area canvas.
- 4) Klik pada area canvas berbeda sebanyak jumlah sudut yang Anda masukkan pada langkah 2. Setiap titik yang Anda klik akan menjadi salah satu sudut pada poligon yang akan Anda buat.
- 5) Akan ditampilkan hasil penggambaran poligon dengan jumlah sudut sesuai masukkan Anda.



2. Menyimpan model ke dalam sebuah file

- **Download file**

- 1) Klik menu “Save” yang ada di bawah canvas.
- 2) Akan muncul pop up seperti ini



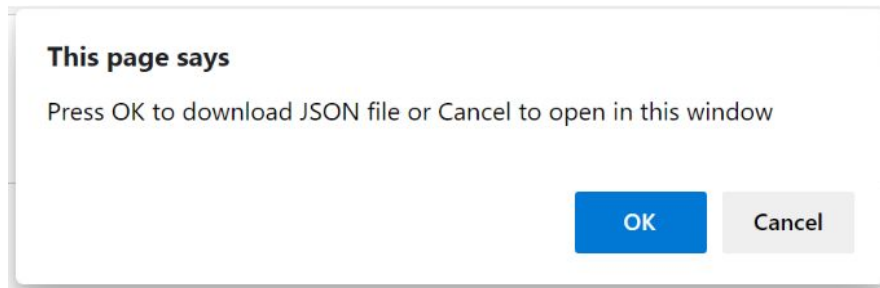
- 3) Pilih “OK”
- 4) Masukkan nama file yang diinginkan (tidak perlu menambahkan ekstensi, hasil upload adalah file .json)



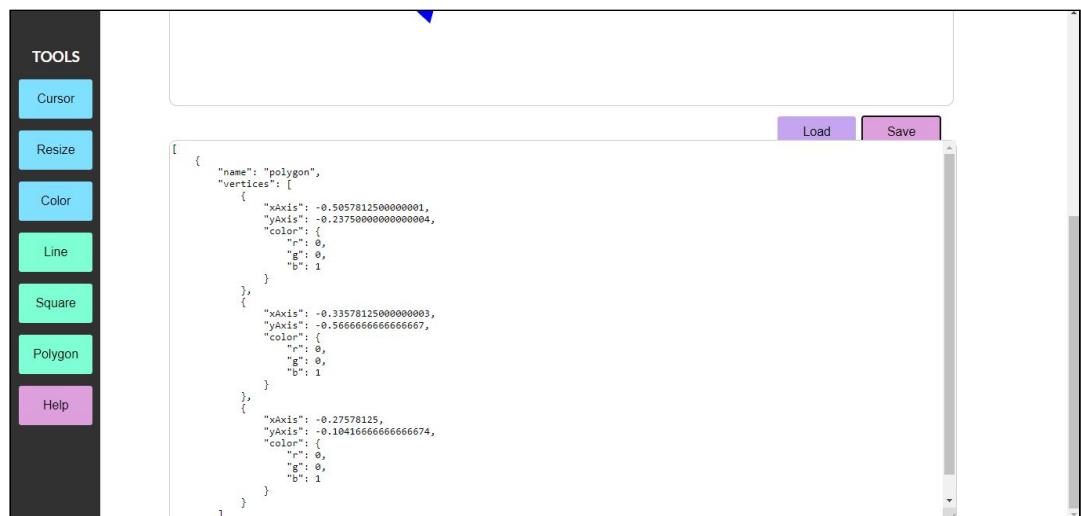
- 5) Pilih “OK” sekali lagi
- 6) File json akan terdownload pada path folder yang Anda pilih.

- **Tampilkan isi model di browser editor**

- 1) Klik menu “Save” yang ada di bawah canvas.
- 2) Akan muncul pop up seperti ini



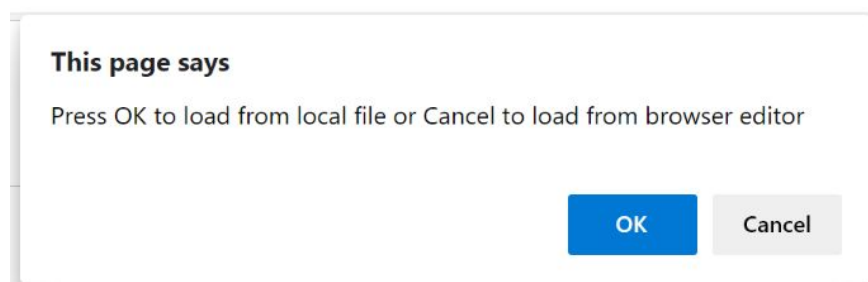
- 3) Pilih “Cancel”
- 4) Akan muncul sebuah textarea yang menampilkan model dalam bentuk json dan dapat diedit



3. Membuka file hasil penyimpanan model

- **Buka file dari local storage**

- 1) Klik menu “Load” yang ada di bawah canvas.
- 2) Jika muncul pop up seperti gambar dibawah ini, tekan “Ok”



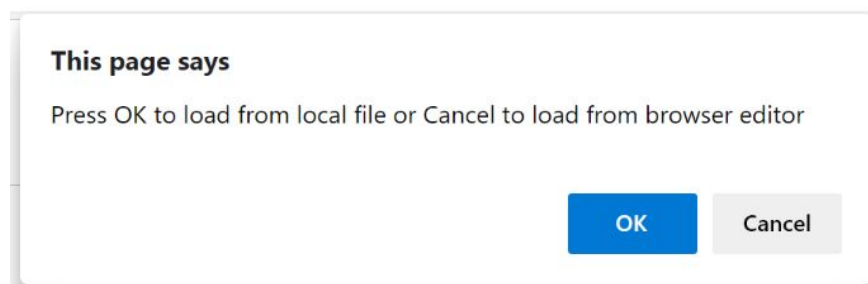
- 3) Pilih file dalam format .json yang akan di buka lalu klik “Open”.
Contoh format file .json adalah sebagai berikut.

```
[
  {
    "name": "line",
    "vertices": [
      {
        "xAxis": 0.20021874999999999,
        "yAxis": 0.5083333333333333,
        "color": {
          "r": 1,
          "g": 0,
          "b": 0
        }
      },
      {
        "xAxis": 0.20421874999999999,
        "yAxis": 0.19583333333333333,
        "color": {
          "r": 0,
          "g": 0,
          "b": 1
        }
      }
    ]
  }
]
```

4) Isi file akan dibaca dan ditampilkan penggambarannya pada canvas.

- **Load dari browser editor**

- 1) Pastikan browser editor sudah terbuka (jika belum terbuka, klik menu “Save” dan pilih “Cancel” ketika muncul pop up).
- 2) Klik menu “Load” yang ada di bawah canvas.
- 3) Akan muncul pop up seperti ini.



4) Pilih “Cancel”

4. Menggeser titik kontrol atau simpul dengan mouse

- **Seluruh model**

- 1) Pastikan pada canvas telah ada paling tidak satu model.
- 2) Klik menu “Cursor” yang terdapat pada toolbar di sebelah kiri.
- 3) Arahkan kursor ke area yang kosong pada canvas

- 4) Klik lalu drag mouse sampai Anda memperoleh posisi yang sesuai untuk seluruh model yang terdapat pada canvas
- 5) Drop titik pada posisi tersebut sehingga seluruh model pada canvas akan berpindah secara permanen ke posisi tersebut.

- **Satu buah model**

- 1) Pastikan pada canvas telah ada paling tidak satu model.
- 2) Klik menu “Cursor” yang terdapat pada toolbar di sebelah kiri.
- 3) Arahkan kursor ke salah satu titik pada model.
- 4) Klik lalu drag titik tersebut sampai Anda memperoleh posisi yang sesuai untuk model tersebut
- 5) Drop titik pada posisi tersebut sehingga model akan berpindah secara permanen ke posisi tersebut.

5. Mengubah panjang garis

- 1) Pastikan pada canvas telah ada paling tidak satu model garis.
- 2) Klik menu “Resize” yang terdapat pada toolbar di sebelah kiri.
- 3) Arahkan kursor ke salah satu titik ujung garis.
- 4) Klik lalu drag titik tersebut sampai Anda memperoleh ukuran panjang garis sesuai yang Anda inginkan
- 5) Drop titik pada posisi tersebut sehingga panjang garis akan berubah.

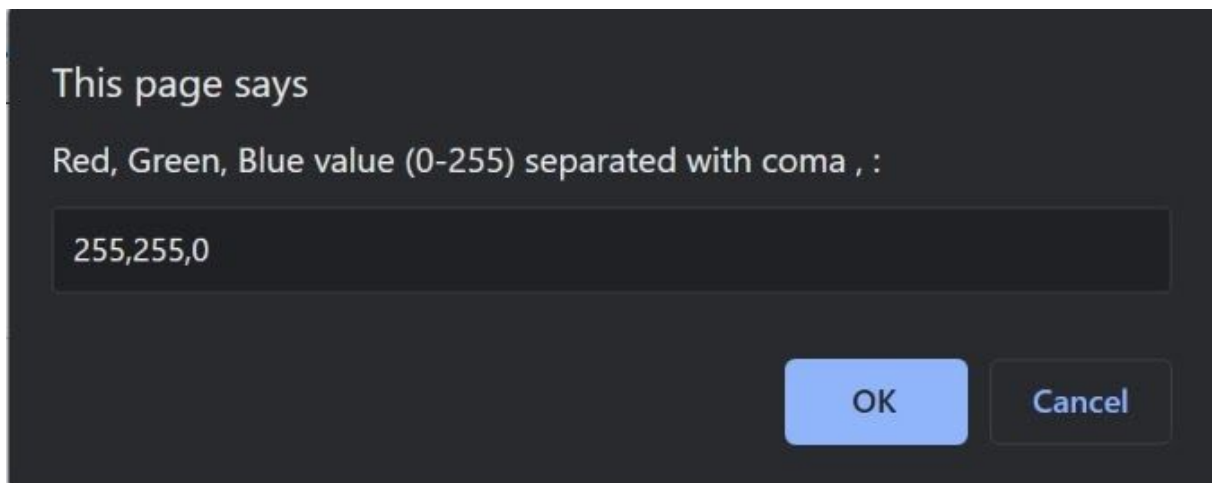
6. Mengubah panjang sisi persegi

- 1) Pastikan pada canvas telah ada paling tidak satu model persegi.
- 2) Klik menu “Resize” yang terdapat pada toolbar di sebelah kiri.
- 3) Arahkan kursor ke salah satu titik sudut persegi.
- 4) Klik lalu drag titik tersebut sampai Anda memperoleh ukuran panjang sisi persegi sesuai yang Anda inginkan
- 5) Drop titik pada posisi tersebut sehingga panjang garis akan berubah.

7. Mengubah warna poligon

- 1) Pastikan pada canvas telah ada paling tidak satu model polygon.
- 2) Klik menu “Color” yang terdapat pada toolbar di sebelah kiri.
- 3) Arahkan kursor ke salah satu titik sudut poligon.

- 4) Isi angka RGB (0-255) sesuai warna yang diinginkan.



- 5) Setelah itu warna poligon akan berubah.

8. Menu help

- 1) Klik menu "Help" yang terdapat pada toolbar di sebelah kiri.
- 2) Lalu akan muncul page Help di tab baru