

# **IF3260 Grafika Komputer**

## **Tugas 1**

### **2D Web Based CAD (Computer-Aided Design)**



Oleh:

<b>Muhammad Fikri Ranjabi</b>	<b>13520002</b>
<b>Taufan Fajarama Putrawansyah R</b>	<b>13520031</b>
<b>Fadil Fauzani</b>	<b>13520032</b>

**K-01**

**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
2023**

# **Bab I**

## **Deskripsi**

Pada tugas 1 IF3260 grafika komputer ini kami membuat sebuah website yang menyediakan interaksi dua dimensi terhadap objek atau model, yaitu 2D Web Based CAD (Computer-Aided Design). Website ini dibuat menggunakan HTML, CSS, dan javascript dengan memanfaatkan kakas dari WebGL. WebGL merupakan kakas dengan spesialisasi ranah grafika yang dapat dengan mudah diintegrasikan dengan web dan dapat digunakan untuk mengimplementasikan website dengan fitur menggambar, mengedit, dan memvisualisasi sejumlah model pada kanvas.

Beberapa fitur utama yang kami implementasikan adalah sebagai berikut.

1. Implementasi model beserta metode spesialnya
  - a. Model garis dengan metode ubah panjang.
  - b. Model persegi dengan metode ubah panjang sisi.
  - c. Model persegi panjang dengan metode ubah panjang atau lebar.
  - d. Model polygon dengan metode penambahan dan penghapusan titik sudut.
2. Transformasi geometri translasi (garis, persegi, persegi panjang, polygon), dilatasi (garis, persegi), dan rotasi (garis, polygon)
3. Menggerakkan salah satu titik sudut dengan slider atau drag and drop
4. Mengubah warna salah satu atau semua titik sudut
5. Save dan load model yang telah dibuat
6. Integrasi animasi (garis dan polygon)

## Bab II

### Hasil

Berikut adalah hasil dari website yang kami buat dengan implementasi kakas WebGL.

#### 1. Tampilan utama

Tampilan utama dari website adalah sebuah canvas dan sidebar di bagian kanan berisi tombol-tombol atau slider. Canvas dibuat menggunakan elemen HTML canvas dengan tinggi dan lebar sebesar 50 view height. Misalkan pengguna memiliki tampilan layar pada browser dengan tinggi 1080 pixel, maka canvas akan memiliki dimensi 540 pixel x 540 pixel. Pada sidebar, terdapat tombol untuk menggambar setiap model, tombol untuk save dan load model, serta slider untuk properti yang terdapat di setiap model.

Pada file *index.html*, terdapat dua buah script yaitu *vertex-shader-2d* dan *fragment-shader-2d*. *Vertex-shader-2d* berfungsi untuk mengatur posisi objek dalam ruang tampilan yang ditentukan, menerima beberapa masukan dan menghitung koordinat titik dalam sistem koordinat. Beberapa variable di script ini memiliki fungsi sebagai berikut.

- *attribute vec2 a\_position* adalah input vertex untuk posisi objek dalam koordinat x dan y.
- *uniform vec2 u\_resolution* adalah variabel yang menyimpan ukuran resolusi layar atau tampilan yang digunakan.
- *varying vec4 fColor* digunakan untuk meneruskan warna dari vertex ke fragment shader.
- *attribute vec3 vColor* adalah variabel untuk menerima warna untuk setiap vertex.

Pada canvas, digunakan satuan pixel untuk manipulasi objek sehingga dilakukan konversi dari *clip space* ke satuan pixel dalam resolusi layar.

*Fragment-shader-2d* berfungsi untuk mengatur warna dan tampilan dari objek. Script fragment shader hanya menerima satu input variabel yaitu *varying vec4 fColor*; yang telah diteruskan dari vertex shader. Warna dari *fColor* diteruskan ke *gl\_FragColor* untuk menetapkan warna pada objek.

#### 2. Menggambar model

Proses penggambaran seluruh model menggunakan prinsip yang sama yaitu dengan memasukkan koordinat model ke dalam *buffer vertex position*. Terdapat tiga kelas yang merepresentasikan model yaitu kelas garis, rectangle (untuk implementasi persegi dan persegi panjang), serta polygon.

Saat user mengklik tombol model di sidebar, maka akan dihasilkan model pada canvas. Model akan dihasilkan dari posisi awal yaitu kiri atas dan untuk model

selanjutnya akan memiliki offset dari model sebelumnya untuk memisahkan posisi model yang sudah dihasilkan sebelumnya. Kelas model akan mencatat koordinat (x, y) dan menghasilkan model pada canvas dengan atribut awal yang sudah ditentukan seperti panjang/lebar model beserta warna random.

Terdapat suatu variable global bernama *objects* yang menampung seluruh objek yang dihasilkan. Kemudian seluruh koordinat pada objek akan di-*flatten* ke variable *flattenVertexPosition* dengan bentuk array satu dimensi yang akan digunakan ke dalam *bufferData* untuk menampung *vertex buffer*.

Fungsi *drawArrays* akan menggambar informasi yang terdapat pada buffer vertex dan color dengan primitif setiap objek. Primitif yang digunakan adalah POINTS untuk menggambar kotak sudut objek, TRIANGLES, dan LINES.

### 3. Metode spesial dari model

#### a. Garis, metode ubah panjang

Pada garis, metode ubah panjang diimplementasikan dengan melakukan *drag & drop* pada salah satu vertex point garis atau menggunakan slider. Kedua point pada ujung garis dapat digeser dan dimanfaatkan untuk mengubah panjang dari garis. Selain itu, terdapat juga slider untuk mengubah panjang garis.

#### b. Persegi, metode ubah panjang sisi

Pada persegi, metode ubah panjang sisi diimplementasikan dengan melakukan drag dari vertex point persegi. Vertex point yang dapat digeser adalah pada posisi kiri atas atau kanan bawah. Saat vertex point digeser, maka akan dilakukan kalkulasi untuk menghitung panjang baru dari sisi persegi. Panjang baru sisi persegi dihitung dengan selisih koordinat pada mouse dengan koordinat vertex persegi yang lama. Perubahan panjang sisi persegi menjaga kesebangunan.

#### c. Persegi panjang, metode ubah panjang atau ubah lebar

Pada persegi panjang, metode ubah panjang atau ubah lebar kurang lebih sama dengan implementasi metode ubah panjang sisi pada persegi. Perbedaannya adalah pada persegi panjang, sisi panjang dan lebar yang baru tidak perlu memiliki panjang yang sama.

#### d. Polygon, metode penambahan dan penghapusan titik sudut

Pada poligom, metode penambahan dan penghapusan titik sudut diimplementasikan membuat tombol “Add point” ketika objek yang terpilih adalah poligon. Jika tombol “Add point” ditekan, maka titik bisa ditambahkan dengan mengklik suatu titik pada canvas. Penambahan titik bisa dilakukan hingga tombol “Stop” ditekan. Sedangkan, penghapusan titik dilakukan dengan menambahkan tombol “Delete” pada tiap titik poligon pada tab properti. Penghapusan titik bisa dilakukan hingga titik habis, serta jika titik habis object tersebut akan dihapus dari pool objects.

#### **4. Transformasi geometri translasi, dilatasi, dan rotasi**

Transformasi geometri untuk model garis diterapkan translasi dan rotasi. Transformasi geometri untuk model persegi dan persegi panjang diterapkan translasi dan dilatasi. Transformasi geometri untuk model polygon diterapkan translasi dan rotasi. Ketiga transformasi geometri dilakukan dengan menggunakan slider (translasi dan rotasi) dan drag & drop (untuk dilatasi persegi/persegi panjang).

Implementasi translasi menggunakan informasi posisi x dan posisi y dari salah satu titik pada object saat ini, lalu dengan menggunakan slider, object dapat dilakukan translasi terhadap sumbu-x dan terhadap sumbu-y. Dilakukan perhitungan agar object tetap mempertahankan bentuknya saat berpindah tempat.

Implementasi dilatasi menggunakan informasi koordinat tiap titik dari persegi/persegi panjang dan akan melakukan update koordinat tersebut ketika dilakukan drag & drop. Perhitungan dilatasi untuk update koordinat dilakukan dengan tetap memperhatikan bentuk object.

Implementasi rotasi menggunakan informasi matriks posisi dari setiap titik di object. Matriks posisi tersebut akan dilakukan perkalian matriks dengan matriks rotasi sehingga akan dihasilkan matriks baru hasil rotasi. Perhitungan dilakukan dengan memanfaatkan informasi sinus cosinus rotasi dan matriks posisi object.

#### **5. Menggerakkan salah satu titik sudut dengan slider atau drag and drop**

Implementasi pergerakan salah satu titik sudut diawali dengan mencari vertex point terdekat dari posisi mouse, vertex point terdekat akan disimpan ke dalam variable *closestVertex*. Hal ini dilakukan menggunakan euclidean distance dari vertex point setiap objek dengan posisi mouse. Jika terdapat vertex point dengan euclidean distance kurang dari 20 pixel terhadap posisi mouse, maka objek yang memiliki vertex point tersebut akan ditambahkan ke dalam variable *closestObject*.

Setelah didapat objek terdekat dari posisi mouse, maka terdapat fungsi *updateCoor* yang akan memperbarui salah satu titik sudut dari objek. Pergerakan salah satu titik sudut dengan drag and drop ini menjaga kesebangunan pada model persegi dan persegi panjang.

#### **6. Mengubah warna salah satu atau semua titik sudut**

Mirip dengan menggerakkan titik, untuk mengubah warna user harus memilih objek yang ingin diubah terlebih dahulu yang nantinya akan disimpan di variable *closestObject*. Setelah itu, pada tab properti akan muncul properti object tersebut salah satunya adalah warna. Properti warna tersebut bisa diganti baik pada tiap titik maupun untuk semua sudut dengan menggunakan tipe input color picker. Ketika warna pada properti tertentu diganti maka akan menyebabkan object mengganti warna pada titik yang bersangkutan dengan warna yang dipilih.

#### **7. Save dan load model yang telah dibuat**

Implementasi metode save dilakukan dengan pemanfaatan metode kelas yang dimiliki Javascript yaitu `JSON.stringify`, sehingga properti object yang ada di canvas kita bisa simpan dalam suatu file text dan mengunduh file tersebut. Sedangkan, metode load dilakukan dengan memanfaatkan `JSON.parse` sehingga properti objek yang ada di text kita bisa buat ulang objek-objeknya dengan menyalin properti yang ada.

#### 8. **Integrasi animasi**

Integrasi animasi dilakukan pada fitur rotasi di model line dan poligon dengan memanggil fungsi `'rotationAnim'` dengan parameter nilai `theta`. Fungsi ini akan dipanggil jika variable `'isObjectRotate'` bernilai `true` setelah mengklik tombol `'start rotation animation'` di sidebar. Nilai `theta` akan bertambah 1 setiap fungsi render frame dijalankan.

## Bab III

### Manual dan Contoh Fungsionalitas

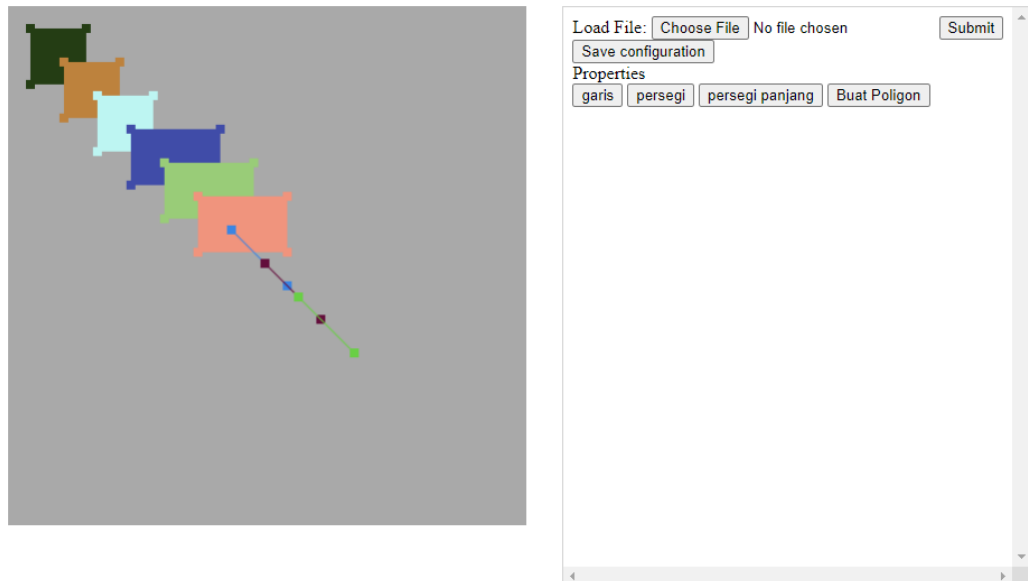
#### Instalasi:

1. Clone repository: <https://github.com/ranjabi/2d-web-based-cad>
2. Jalankan *index.html* di browser

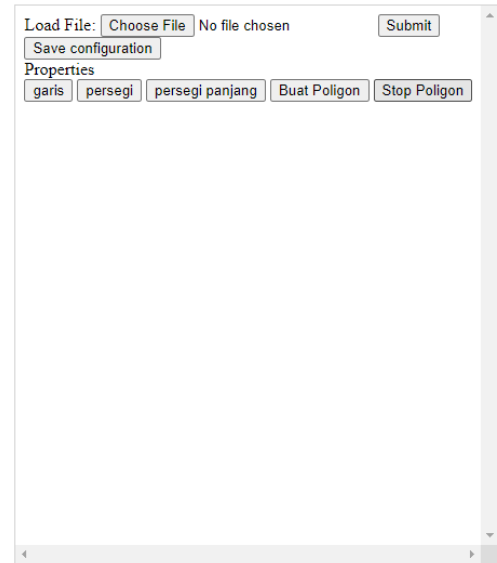
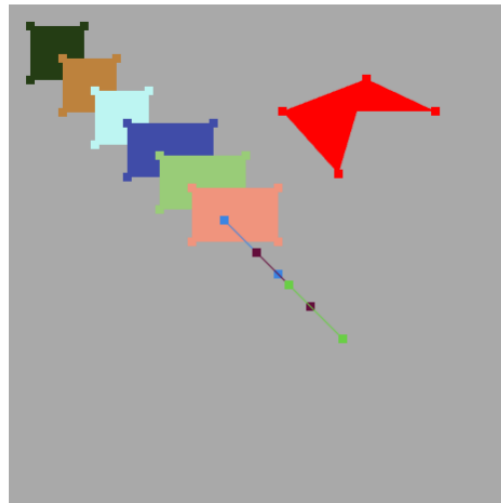
#### Langkah penggunaan dari setiap fitur:

- **Menggambar model**

1. Klik tombol setiap model pada sidebar untuk menggambar garis, persegi, dan persegi panjang. Model yang dipilih akan muncul di canvas.



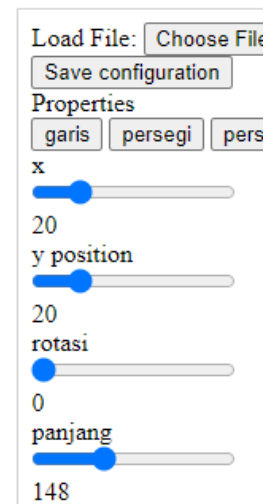
2. Untuk menggambar polygon, klik tombol 'Buat Poligon', kemudian klik sembarang titik di canvas, jika sudah selesai menggambar maka klik 'Stop Poligon'.



- **Metode spesial model**

### **Garis, ubah panjang**

1. Klik salah satu sudut pada garis, atur panjang garis pada slider bagian 'panjang'.



### **Persegi, ubah panjang sisi**

1. Klik salah satu sudut pada persegi, atur panjang sisi persegi pada slider bagian 'panjang'.





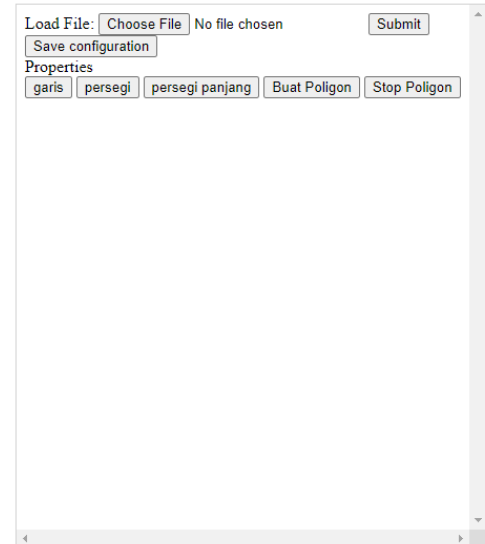
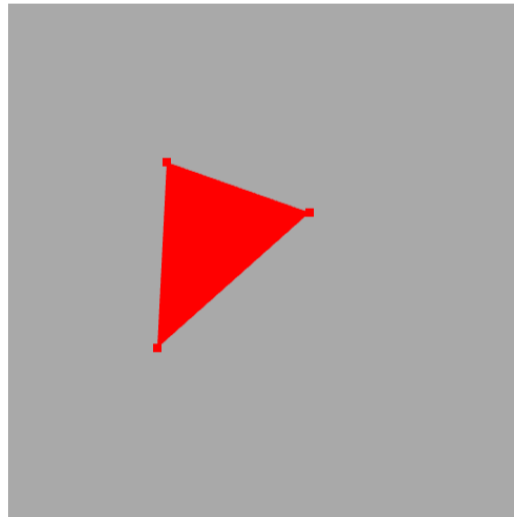
### Persegi panjang, ubah panjang dan lebar

1. Klik salah satu sudut pada persegi panjang, atur panjang dan lebar pada slider bagian 'panjang' dan 'lebar'.

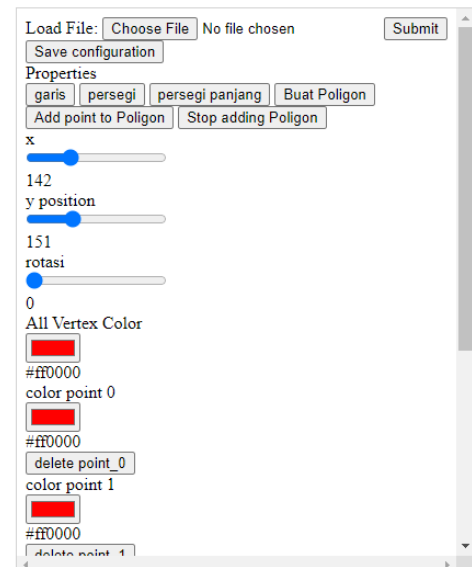
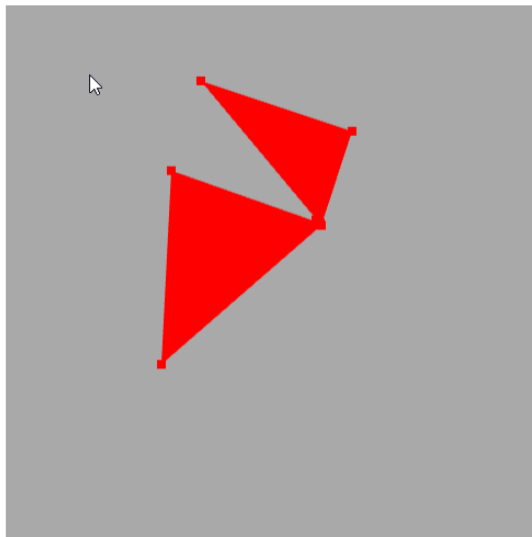


### Poligon, menambahkan dan menghapus titik

1. Buat Poligon dasar dengan tombol Buat poligon dan pilih titik pada canvas dengan klik mouse, jika sudah klik tombol stop poligon

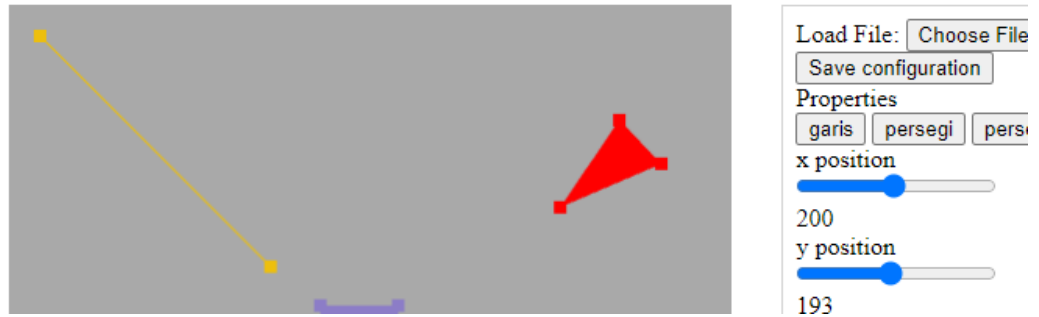


2. Untuk menambahkan titik pada poligon, pilih poligon dengan mengklik area yang dekat dengan salah satu titik pada poligon. Lalu pilih tombol add point to poligon dan pilih titik pada canvas dengan klik mouse, jika sudah klik tombol stop adding poligon.

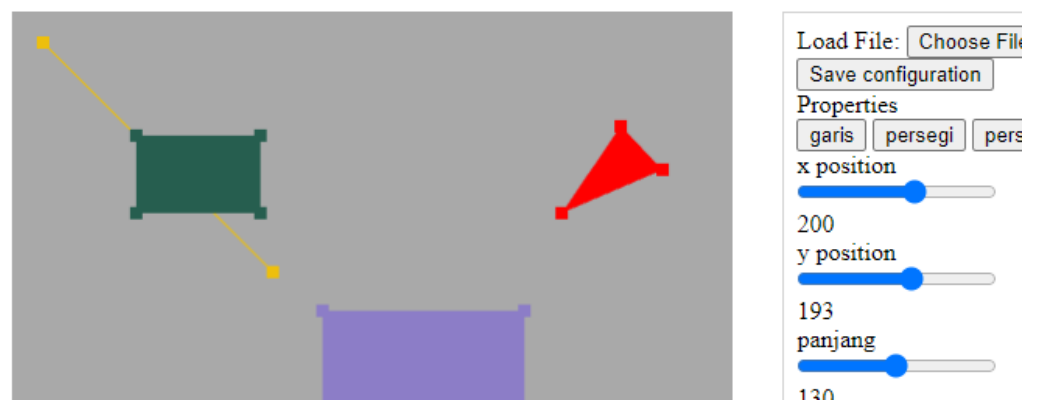


- **Transformasi geometri translasi, dilatasi, dan rotasi**

1. Translasi dapat dilakukan dengan mengklik salah satu sudut dari objek dan mengatur nilai 'x position' dan 'y position' di sidebar. Hal ini dapat dilakukan di garis, persegi, persegi panjang, dan polygon.



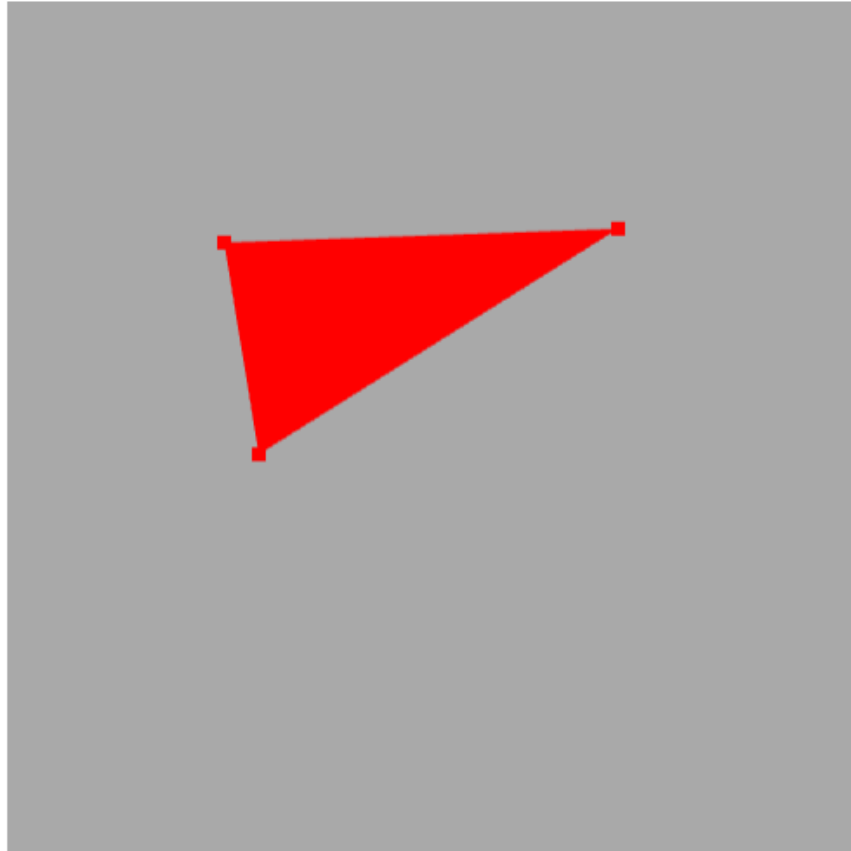
2. Dilatasi dapat dilakukan dengan mengatur properti 'panjang' pada garis dan persegi. Implementasi ini merupakan bagian dari 'ubah panjang' pada metode spesial model.



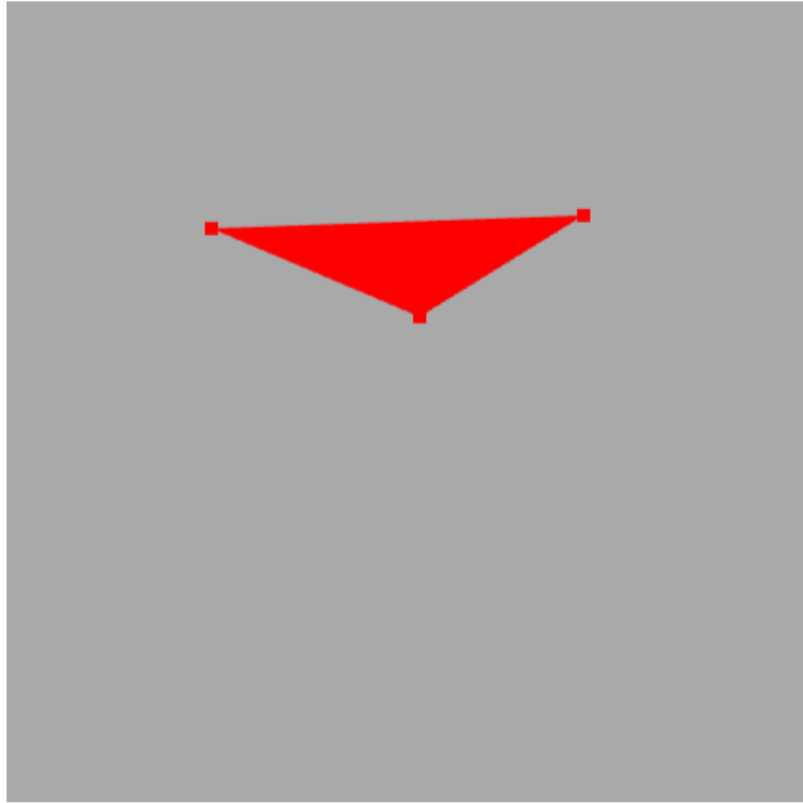
3. Rotasi dapat dilakukan dengan mengklik salah satu sudut dari objek dan mengatur nilai rotasi di sidebar. Hal ini dapat dilakukan di garis dan polygon.



- **Menggerakkan salah satu titik sudut**
  1. Buat bangun datar yang diinginkan

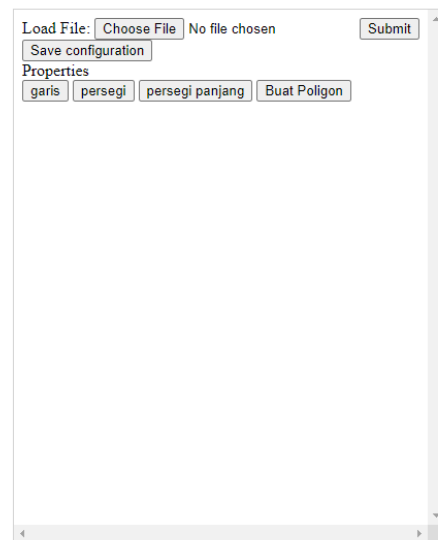
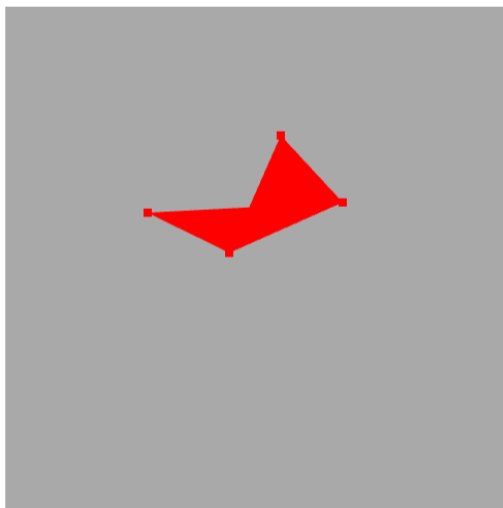


2. Pada salah satu titik bangun datar tersebut lakukan klik, tahan, lalu *drag*, titik tersebut akan berubah mengikuti pergerakan kursor sampai mouse sudah tidak ditahan lagi.

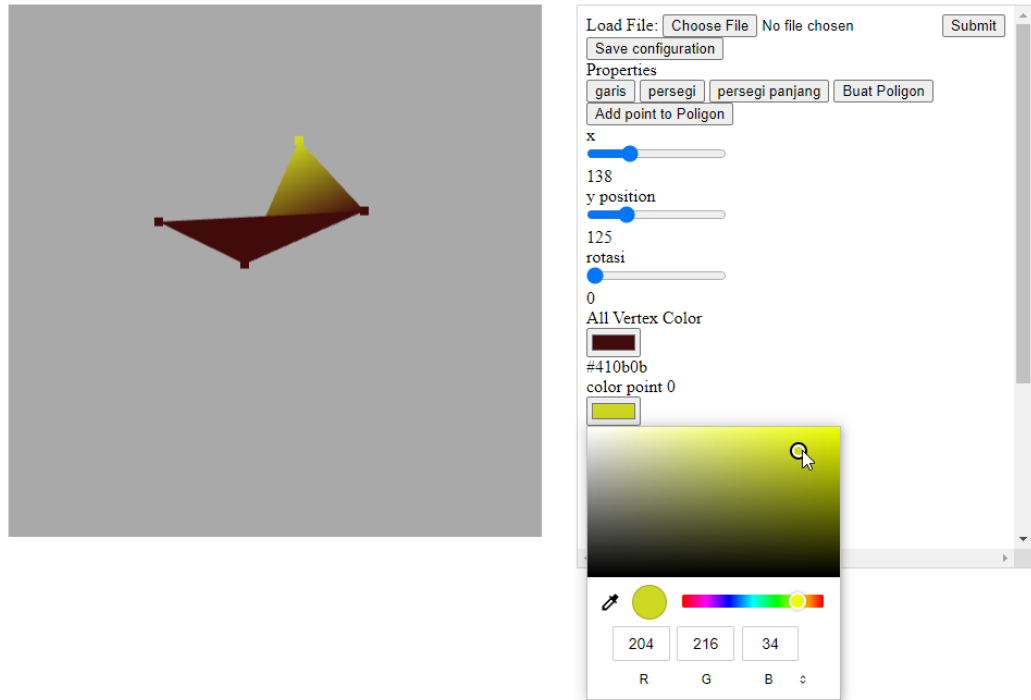


- **Mengubah warna salah satu atau semua titik sudut**

1. Buat bangun datar yang diinginkan



2. Pada tab properti akan ada warna warna tiap titik pada bangun datar. Untuk mengganti warna nya klik salah satu warna titik, lalu pilih warna sesuai keinginan



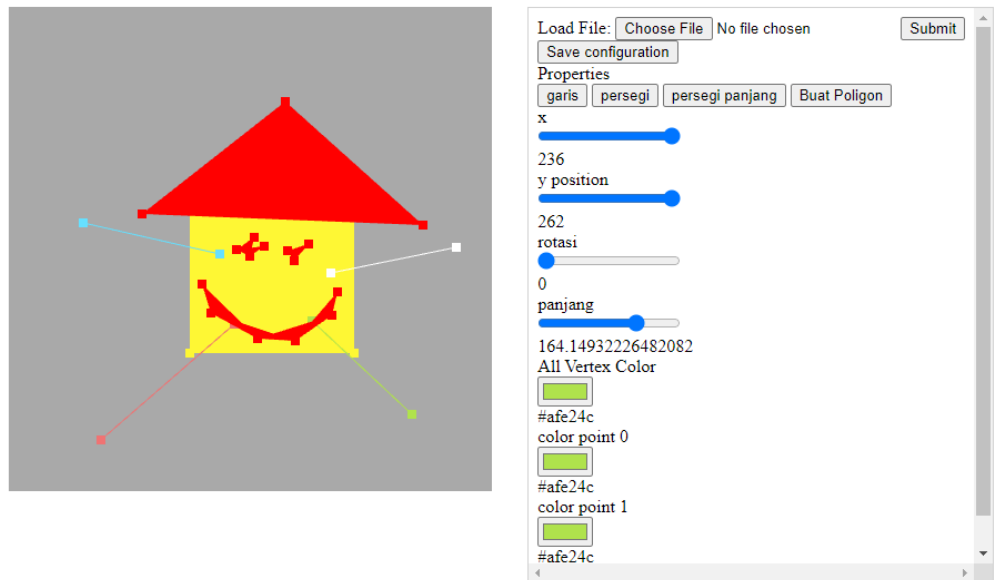
- **Save dan load model yang telah dibuat**

1. Buat model yang di inginkan



2. Untuk Save tekan tombol save configuration, browser akan otomatis mendownload file konfigurasi model yang telah dibuat
3. Untuk melakukan load, tekan Choose file lalu pilih file konfigurasi yang telah didownload sebelumnya. Setelah itu klik tombol submit.

Telah disediakan 2 buah test model pada folder ‘test’. Test model 1 memiliki bentuk sebagai berikut.

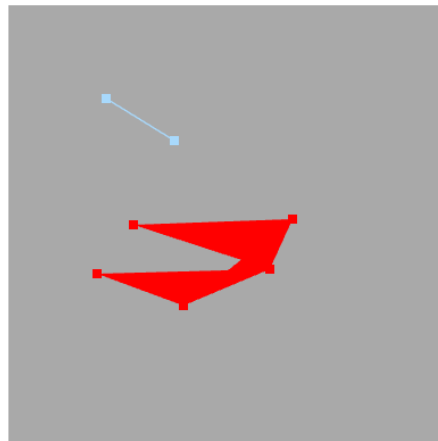


Test model 2 memiliki bentuk sebagai berikut.



- **Menjalankan animasi**

1. Klik salah satu sudut pada objek garis atau polygon
2. Klik tombol ‘start rotation animation’ untuk memulai animasi dan klik tombol ‘stop rotation animation’ untuk memberhentikan animasi.



Load File:  No file chosen

Properties

x

78

y position

188

rotasi

0

All Vertex Color

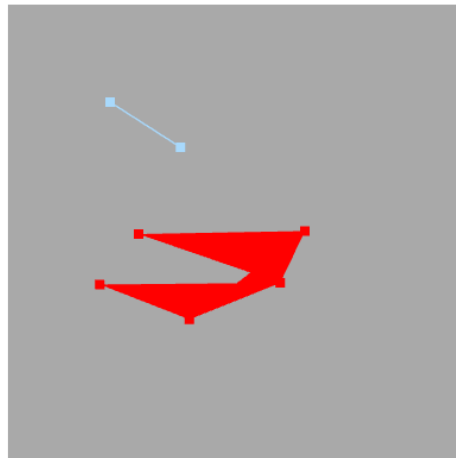
#ff0000

color point 0

#ff0000

color point 1

#ff0000



Load File:  No file chosen

Properties

x

78

y position

188

rotasi

0

All Vertex Color

#ff0000

color point 0

#ff0000

color point 1

#ff0000