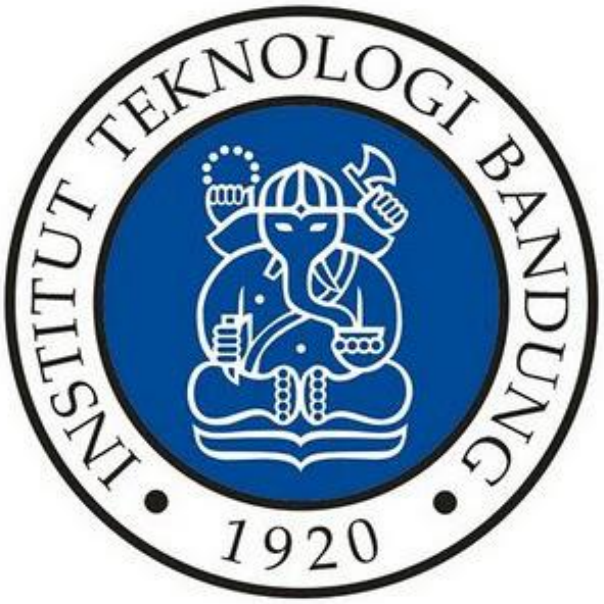
LAPORAN

Tugas Besar 1: 2D Web Based CAD
IF3260 Grafika Komputer

DISUSUN OLEH

13520124	Owen Christian Wijaya
13520137	Muhammad Gilang Ramadhan
13520151	Rizky Ramadhana Putra Kusnaryanto



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

2022

DAFTAR ISI

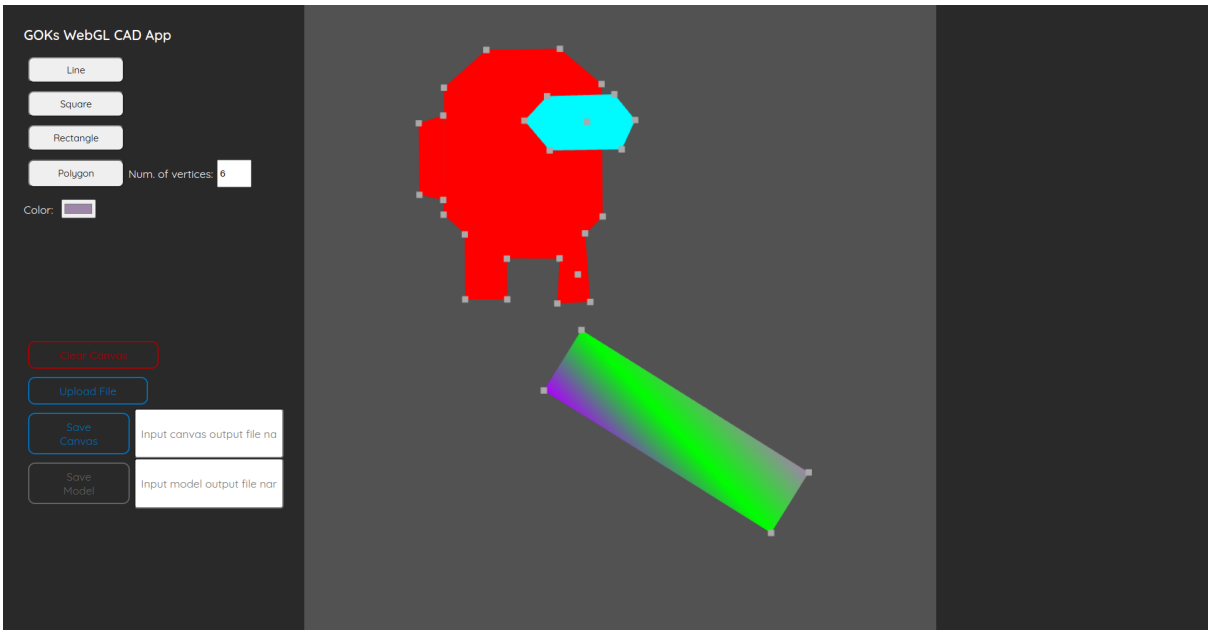
BAB I: DESKRIPSI

Tugas besar 1 mata kuliah IF3260 Grafika Komputer bersangkutan dengan WebGL sebagai pustaka grafika pada bidang *web* yang akan digunakan untuk membuat sebuah aplikasi *computer-aided design* (CAD). Adapun WebGL yang digunakan merupakan WebGL murni, tanpa *library/framework* tambahan. Umumnya dalam proyek WebGL, terdapat *library* yang berisi fungsi utilitas yang umumnya sudah disiapkan oleh WebGL itu sendiri. Namun, pada tugas besar kali ini, penulis membuat fungsi-fungsi tersebut sendiri.

Aplikasi yang dibuat harus bisa digunakan untuk menggambar bidang dua dimensi yaitu persegi, persegi panjang, dan polygon dengan fungsi utilitas untuk mengubah, dan memvisualisasi sejumlah model pada kanvas. Selain itu, pengguna harus dapat melakukan transformasi (translasi, rotasi, dilatasi), menggerakan salah satu titik sudut, mengubah warna titik sudut, dan menyimpan hasil gambar yang telah dibuat.

BAB II: HASIL IMPLEMENTASI

Pada tugas besar ini, *vertex* akan didefinisikan sebagai sebuah titik sudut dari bidang yang akan mengandung informasi mengenai posisi (*position*) dan warna (*color*). Sebuah model akan didefinisikan sebagai sebuah *object* yang menyimpan daftar dari *vertex* (disebut *vertices*). Program utama akan menyimpan sebuah *array* berisi *object-object* yang akan diolah bersamaan. *Array* inilah yang akan menjadi pedoman aplikasi untuk melakukan operasi selama berjalan.



Tampilan program

Pengguna dapat menggambar empat buah tipe model pada *canvas* yang disediakan, yaitu model *line* (garis), *square* (persegi), *rectangle* (persegi panjang), dan *polygon*. Pengguna dapat mengklik tombol pada *sidebar* dan mulai menggambar model yang diinginkan. Untuk *polygon*, terdapat *field* tambahan untuk mengisikan jumlah *vertex* yang diinginkan.

Pengguna dapat mengubah ukuran garis, persegi, dan persegi panjang dengan mengklik salah satu *vertex* dan menggerakan *vertex* tersebut. Pengguna juga dapat menggerakkan model dengan mengklik model tersebut dua kali dan menggerakannya dengan *mouse*. Pengguna dapat mengganti warna salah satu *vertex* / seluruh *vertex* dengan meng-klik (untuk *vertex*) / meng-*double click* (untuk semua vertex) dan memilih warna baru di *color picker*. Pengguna juga dapat mengubah ukuran objek melalui *field* yang muncul di *sidebar* apabila sedang memilih objek tersebut. Pengguna dapat memilih untuk menyimpan sebuah model spesifik ketika men-*select* model tersebut, menyimpan *union* dari dua model, atau menyimpan keseluruhan *canvas*.

Website juga diterapkan dengan *live rendering*, yang berarti pergerakan kursor pengguna selama menggerakan / menggambar model akan terbaca dan langsung ditampilkan selama proses penggerakan. Hal ini mempermudah pengguna dalam menggunakan *website* dikarenakan tampilan langsung yang dapat dilihat. Selain itu, hal ini membantu pengguna menggambar *polygon*, karena pengguna dapat melihat secara langsung hasil gambar yang akan dibuat sebelum meletakkan *vertex* di sebuah titik.

Panduan penggunaan program yang lebih detail dapat dilihat pada **Bab III.**

Terdapat beberapa bonus yang diimplementasikan, yaitu:

1. **Algoritma *convex hull***

   Algoritma *convex hull* dibuat supaya hasil *polygon* yang terbentuk akan selalu membentuk *convex hull*. Algoritma ini memanfaatkan konsep *divide-and-conquer* untuk membagi sekumpulan *vertex* menjadi dua bagian, lalu menentukan *vertex* mana saja yang akan membentuk sebuah *convex hull*. Algoritma akan membagi kumpulan *vertex* menjadi bagian "kiri" dan "kanan", kemudian membandingkan *vertex* menggunakan determinan dan perhitungan matematis lainnya untuk menentukan apakah sebuah *vertex* harus dilibatkan dalam pembentukan *convex hull* atau tidak. Setelah itu, akan ada *vertex* yang membentuk *convex hull* dan tidak termasuk dalam pembentuk *convex hull* (berada di dalam *convex hull*).

   Untuk merealisasikan algoritma ini, terdapat sebuah batasan yang harus diimplementasikan. Pada pembentukan *convex hull*, terdapat kemungkinan adanya beberapa titik yang berada di dalam *convex hull*. Titik-titik tersebut akan di-*render* setelah titik-titik pembentuk *convex hull* di-*render*. Apabila titik-titik tersebut digerakkan ke luar *convex hull*, maka hasil yang dibentuk juga akan berbentuk *convex hull* baru dengan titik berbeda di dalamnya. Pada saat menggambar / menggerakkan *vertex* di *polygon*, algoritma ini akan menggambarkan *preview* dari *convex hull* yang akan terbentuk. Oleh karena itu, pengguna akan selalu mengetahui bentuk *convex hull* yang akan terbentuk.

2. ***Union* dari dua model yang *overlap***

   Union dalam konteks ini berarti pengguna dapat memilih dan memodifikasi dua buah objek bersamaan. Algoritma yang digunakan adalah algoritma Gilbert–Johnson–Keerthi (GJK) untuk menentukan apakah dua buah *convex hull* / model *overlap* satu sama lain. Algoritma GJK sendiri merupakan algoritma yang dapat menghitung jarak minimum antara 2 poligon convex. Dengan demikian pada program kami, algoritma ini didesain untuk mengecek apakah ada poligon lain yang menempel dengan poligon yang diklik sekarang atau dengan kata lain jarak minimum kedua poligon ialah nol.

Untuk skemanya, secara garis besar digunakan selisih *minkowski* antara 2 buah poligon dengan titik acuan tertentu yang digunakan untuk menghitung jarak antara 2 buah titik sudut antar *polygon*. Kemudian akan dicek melalui *dot product* terlebih dahulu apakah nilainya kurang dari sama dengan nol atau tidak. Jika ya maka pasti kedua *polygon* tidak menempel, kalau ya dilakukan pengecekan *triple product* pada pengurangan titik sebelumnya dan hasil dari *minkowski distance* sebelumnya, begitu seterusnya sampai dengan point ketiga, dihasilkan nilai *dot product* dengan skema yang sama yaitu > 0.

Untuk tata cara penggunaannya, apabila pengguna meng-*double click* sebuah model yang *overlap* dengan model lain sambil menekan tombol *Shift*, maka kedua model tersebut akan terpilih bersamaan (akan muncul indikator putih pada *vertex* kedua objek). Pengguna dapat menggerakkan kedua model tersebut dan merotasikan kedua model secara bersamaan. Pengguna juga dapat menyimpan kedua objek tersebut ke 1 *file* yang sama/

Hasil implementasi dapat dilihat pada: https://github.com/owencwijaya/IF3260-WebGL-CAD

BAB III: MANUAL FUNGSIONALITAS PROGRAM

3.1 Garis

Menggambar Garis

1. Klik tombol "*Line"* pada *sidebar*
2. Pilih titik bebas pada *canvas,* lalu klik kiri *mouse*
3. Pilih titik bebas lainnya, lalu klik kiri *mouse*
4. Garis baru akan terbentuk

Mengubah Panjang Garis

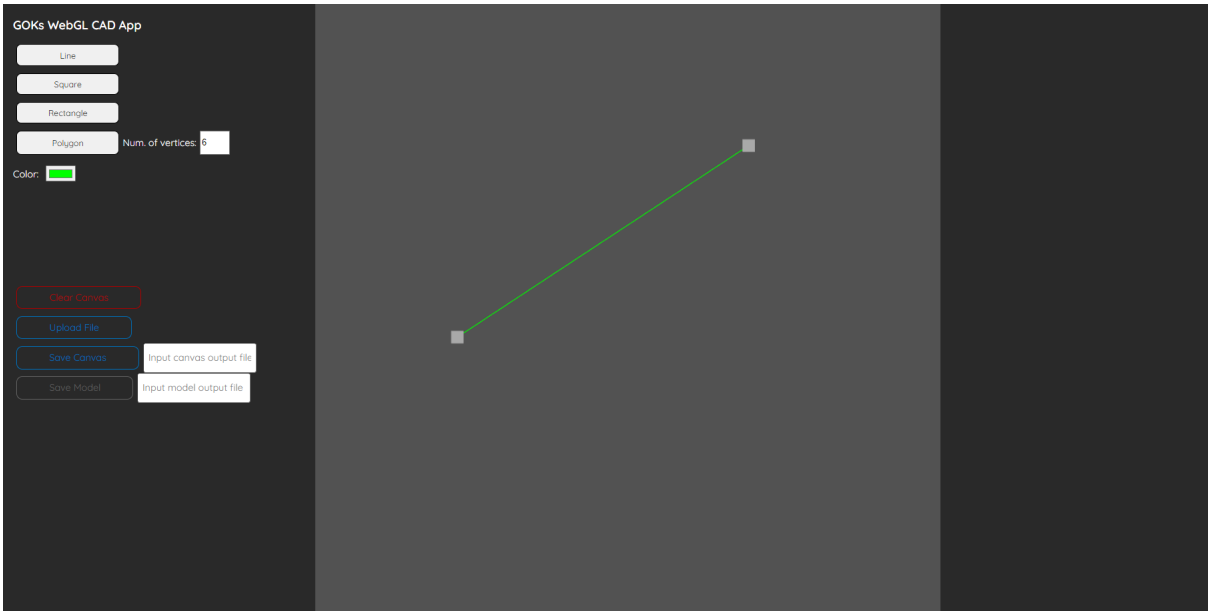
Mengubah panjang garis dapat dilakukan dengan **menggerakkan *vertex* atau mengisikan *field***

Melalui menggerakan vertex,

1. Klik salah satu *vertex* pada garis hingga indikator berubah warna
2. Gerakkan *mouse* Anda ke posisi baru
3. Lepaskan tombol *mouse* apabila kursor / *vertex* sudah berada di posisi yang diinginkan

Melalui *field*,

1. *Double-click* garis yang ingin diubah panjangnya
2. Akan muncul *field* untuk *resize* pada *sidebar*, lalu isikan persentase pertambahan panjang / lebar yang diinginkan (misal. '10' menandakan menambahkan panjang sepanjang 10%)



3.2 Persegi dan Persegi Panjang

Menggambar Persegi / Persegi Panjang

1. Klik tombol *"Square" / "Rectangle"* pada *sidebar*
2. Pilih titik bebas pada *canvas,* lalu klik kiri *mouse*
3. Persegi baru akan terbentuk

Mengubah Panjang Sisi

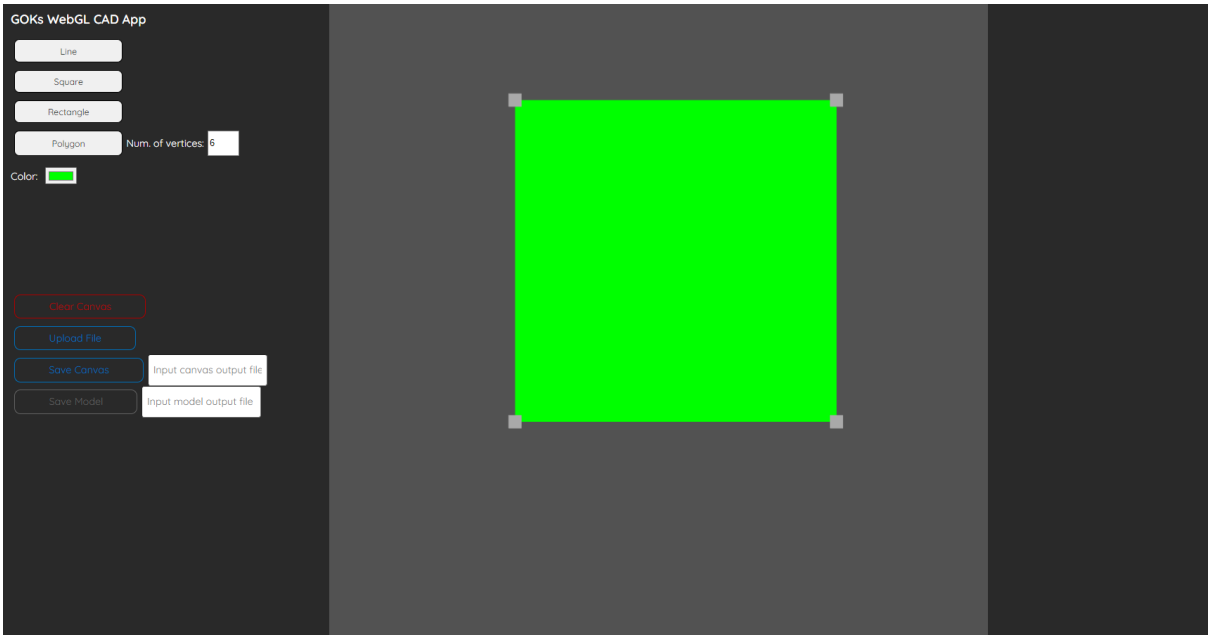
Mengubah panjang sisi dapat dilakukan dengan **menggerakkan *vertex* atau mengisikan *field***

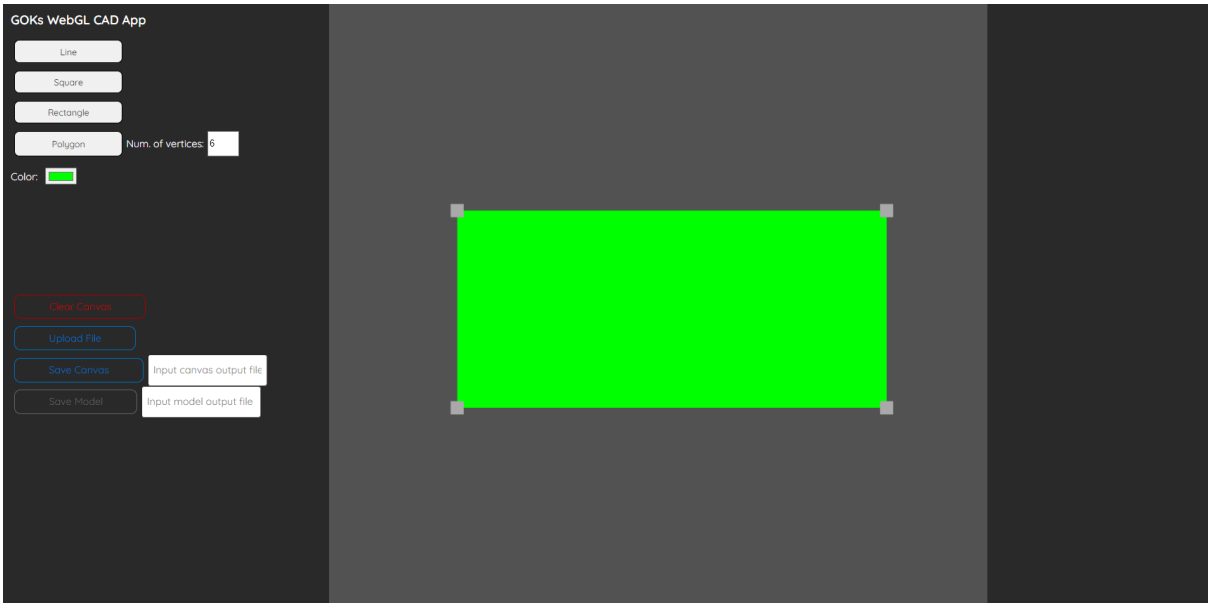
Melalui menggerakan vertex,

1. Klik salah satu *vertex* pada garis hingga indikator berubah warna
2. Gerakkan *mouse* Anda dan ukuran persegi akan berubah
    - Untuk persegi panjang, ukuran panjang dan lebar akan berubah sesuai kursor, sementara untuk persegi, ukuran sisi akan tetap sama

Melalui *field*,

1. *Double-click* persegi / persegi panjang yang ingin diubah
2. Akan muncul *field* untuk *resize* pada *sidebar*, lalu isikan persentase pertambahan panjang / lebar yang diinginkan (misal. '10' menandakan menambahkan panjang sepanjang 10%)
3. Untuk persegi, akan ada 1 *field* yang muncul untuk menandakan pergantian sisi, sementara untuk persegi panjang, akan ada 2 *field* untuk menandakan pergantian panjang (*length*) dan lebar (*width*)





3.3 *Polygon*

Menggambar *Polygon*

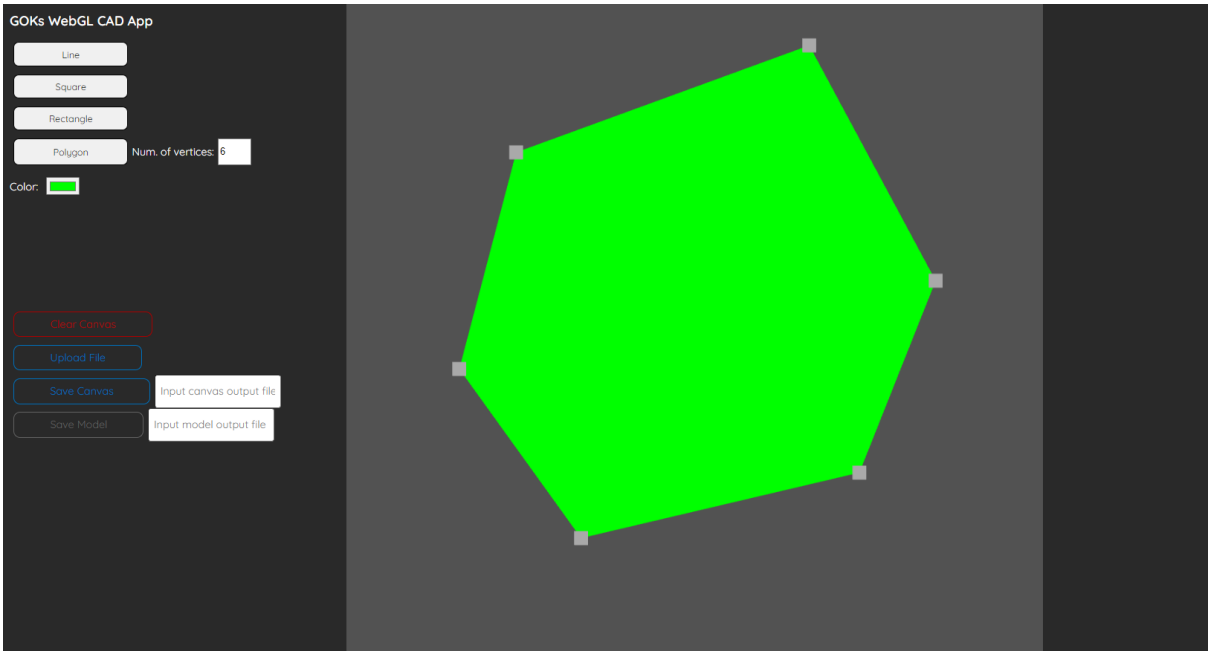
1. Terdapat *field* di bawah tombol untuk menandakan jumlah *vertex*; pastikan *field* sudah terisi (secara *default* berisi 6)
2. Klik tombol *"Polygon"* pada *sidebar*
3. Pilih titik bebas pada *canvas* lalu klik
4. Ulangi proses 3 hingga semua *vertex* sudah tergambar

Menambah Titik Sudut

1. *Double–click polygon* hingga semua indikator *vertex* berwarna putih
2. Klik titik bebas di luar *polygon*
3. *Vertex* baru akan tergambar

Menghapus Titik Sudut

1. *Double–click polygon* hingga semua indikator *vertex* berwarna putih
2. Klik kanan *vertex* yang ingin dihapus
3. *Vertex* tersebut akan dihapus



3.4 Semua Model

Unselect Model

Untuk batal memilih model, klik kanan pada *canvas* bagian manapun, atau *double click* model lainnya

- **Untuk polygon**, hanya terbatas untuk klik kanan, karena mengklik kiri pada objek polygon berfungsi untuk menambahkan *vertex* baru

Memilih Dua / Beberapa Model *Overlap*

1. Tahan tombol *Shift*
2. Klik model yang diinginkan
3. Model yang *overlap* dengan model tersebut dapat digerakkan, dirotasi bersama, atau di-*resize* bersama
    - **Untuk polygon**, apabila *overlap* dan dilakukan *resize*, tidak akan ter-*resize*

Translasi

1. *Double–click polygon* hingga semua indikator *vertex* berwarna putih
2. Tahan *mouse*, lalu gerakkan model yang diinginkan
3. Model akan bergerak ke koordinat baru

Rotasi

1. *Double–click polygon* hingga semua indikator *vertex* berwarna putih
2. Akan muncul *slider* pada *sidebar*, gerakkan *slider* untuk memutar model
3. Model akan berputar sesuai sudut yang diinginkan

Menggerakan *Vertex*

1. Klik *vertex* hingga indikator berwarna putih
2. Tahan *mouse* pada vertex, lalu gerakkan *mouse*
3. *Vertex* akan bergerak ke posisi yang diinginkan
    - **Untuk model garis,** panjang garis akan berubah secara langsung

- **Untuk model persegi**, sisi persegi akan berubah secara langsung, namun panjang keempat sisi akan dipertahankan
- **Untuk model persegi panjang**, panjang dan lebar akan berubah secara langsung tergantung gerakan *mouse*
- **Untuk model *polygon*,** akan muncul *preview* dari *convex hull* yang terbentuk secara langsung

Mengubah Warna Salah Satu Titik Sudut

1. Klik *vertex* hingga indikator berwarna putih
2. Klik *color picker* di *sidebar*, lalu pilih warna baru
3. Warna *vertex* tersebut akan berubah

Mengubah Warna Semua Titik Sudut

1. *Double–click polygon* hingga semua indikator *vertex* berwarna putih
2. Klik *color picker* di *sidebar*, lalu pilih warna baru
3. Warna model akan berubah
    - **Untuk mengubah warna model sebelum digambar**, pilih warna *color picker* sebelum memilih model yang ingin digambar

***Save* Keseluruhan Canvas**

1. Isikan field nama file sebelum menyimpan
2. Klik tombol *Save File*

***Save* Salah Satu Model**

1. *Double click* model yang ingin di-save
2. Isikan field nama file sebelum menyimpan
3. Klik tombol *Save Model*

Load Save File

1. Klik tombol *Upload File*
2. Pilih file yang telah di-*save* sebelumnya (dapat berupa file model atau file canvas)
3. Canvas akan di-*load*
    - Contoh *savefile* model dapat dilihat pada file **test-polygon.json** dan **test-square.json**
    - Contoh *savefile canvas* dapat dilihat pada file **test-canvas.json** dan **test-amongus.json**