

Laporan Tugas Besar I IF3260
Implementasi Kakas Penggambaran Elemen 2D Berbasis WebGL



Disusun oleh
Kelompok 19 K-02 (HelloTriangle)

13521140 Ryan Samuel Chandra

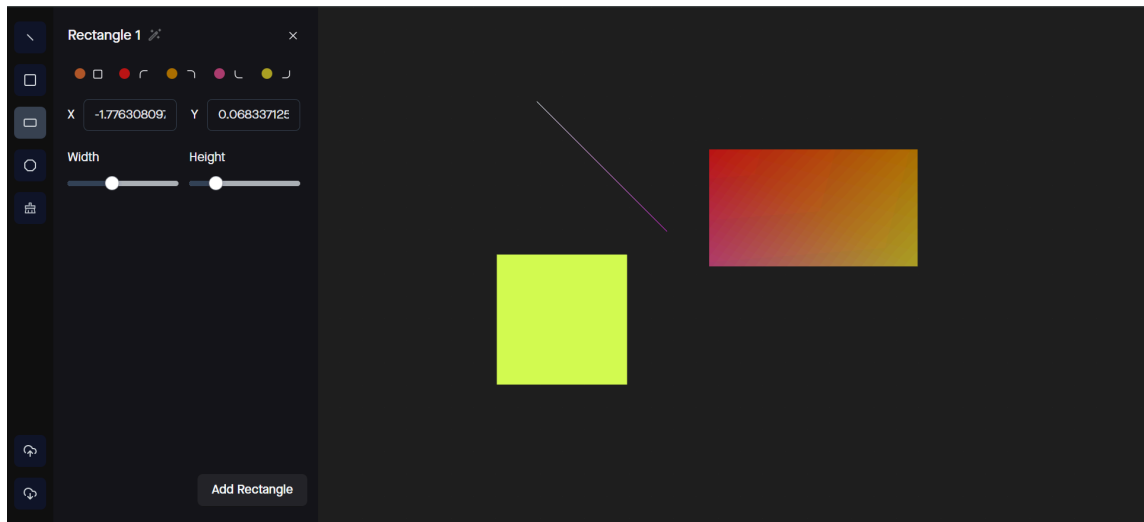
13521142 Enrique Alifio Ditya

13521170 Haziq Abiyyu Mahdy

Program Studi Teknik Informatika
Institut Teknologi Bandung

1. Deskripsi Program

Program ini merupakan implementasi kanvas pembuatan elemen primitif 2D sederhana berbasis WebGL. Terdapat empat elemen primitif yang didukung pada program ini, yaitu garis, persegi, persegi panjang, dan poligon. Program ini menyediakan fitur untuk membuat, memanipulasi, serta menghapus elemen primitif 2D. Selain itu, terdapat pula fitur untuk mengubah warna serta mengaplikasikan transformasi geometri pada elemen. Elemen-elemen yang sudah di-render dapat disimpan ke dalam file JSON yang kemudian dapat di-load untuk ditampilkan kembali. Program dapat dijalankan di *browser* yang mendukung WebGL. Bahasa pemrograman yang digunakan adalah TypeScript dengan kanvas React. Berikut adalah tampilan aplikasi.

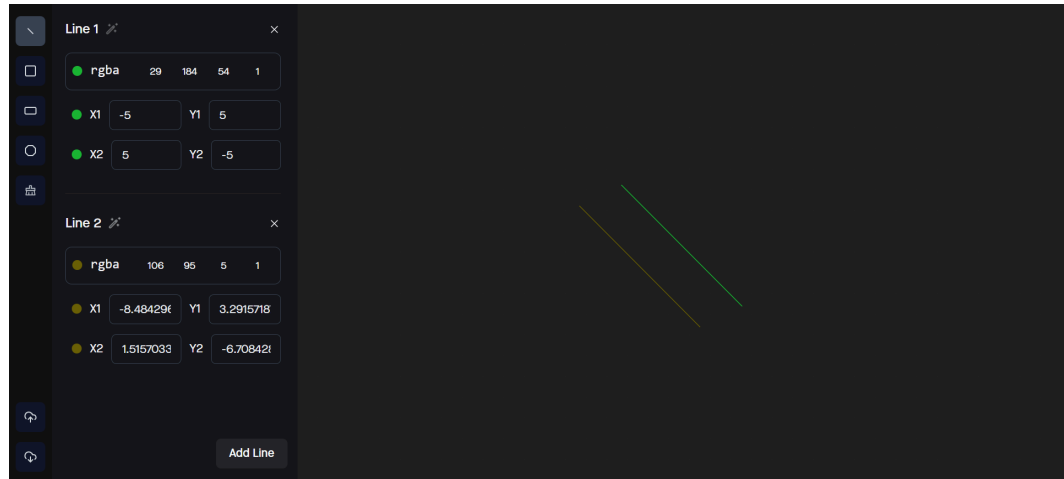


Secara umum, program menyimpan representasi elemen-elemen dalam tipe Shape, yang terdiri atas Line, Square, Rectangle, dan Polygon. Ketika pengguna memasukkan elemen baru dengan menekan tombol Add Line/Square/Rectangle/Polygon, maka program akan menambahkan elemen baru tersebut ke *array* Shapes yang merupakan *react state*. Apabila terjadi manipulasi terhadap *array* Shapes maupun elemen dari *array* Shapes itu sendiri, maka *useEffect* akan memanggil fungsi untuk me-render setiap elemen yang ada pada *array* tersebut, yaitu renderShape (renderLine/renderSquare/renderRectangle/renderPolygon). Fungsi render inilah yang melakukan pendefinisian *buffer*, *binding buffer* dengan data, serta menggambarkan *vertex/fragment* ke dalam *canvas* dengan menggunakan *vertex shader* dan *fragment shader* yang telah didefinisikan sebelumnya.

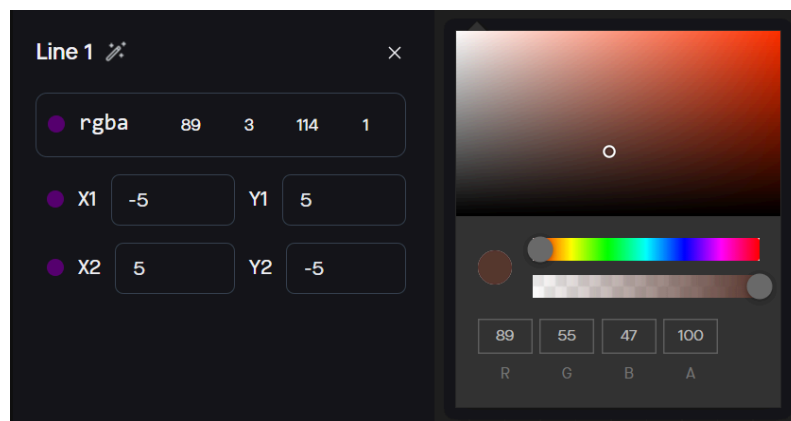
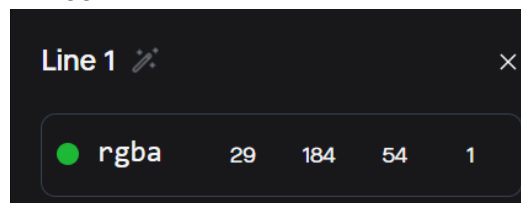
2. Fungsionalitas Program

a. Garis

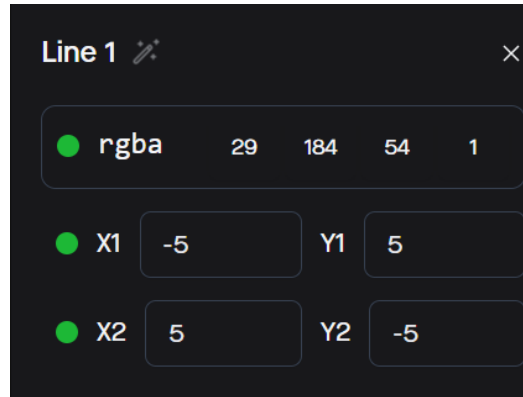
Berikut adalah tampilan fungsionalitas elemen garis.



- Untuk menambahkan garis, klik panel Line pada *sidebar*, lalu klik tombol Add Line.
- Garis yang telah ditambahkan akan tercatat pada sisi kiri aplikasi, dengan ID sesuai urutan penambahan untuk mengidentifikasi garis.
- Warna garis dapat diubah pada field warna di bawah nama garis. Perubahan warna dapat dilakukan dengan mengisi kolom rgba atau menggunakan color picker.



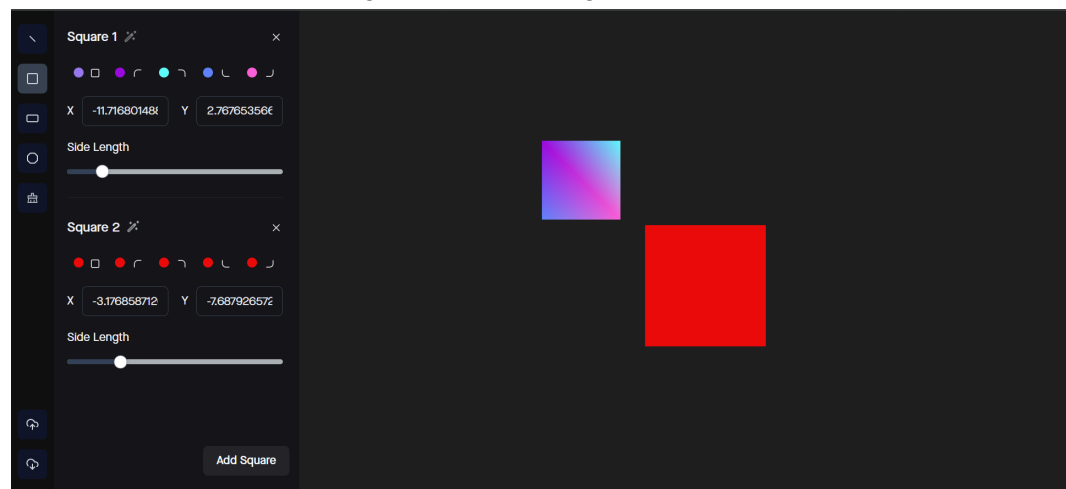
- Untuk mengubah posisi titik awal dan titik akhir dari garis, dapat dilakukan perubahan pada *field* X1, X2, Y2, atau Y2. Selain itu, warna titik juga dapat diubah pada tombol warna berbentuk lingkaran di sisi kiri X1 dan Y1.



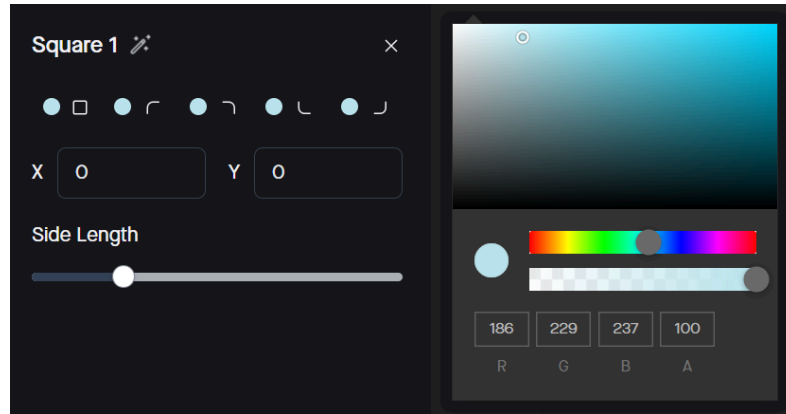
- Untuk menghapus garis, klik tombol silang pada sisi kanan panel garis

b. Persegi

Berikut adalah tampilan fungsionalitas persegi.



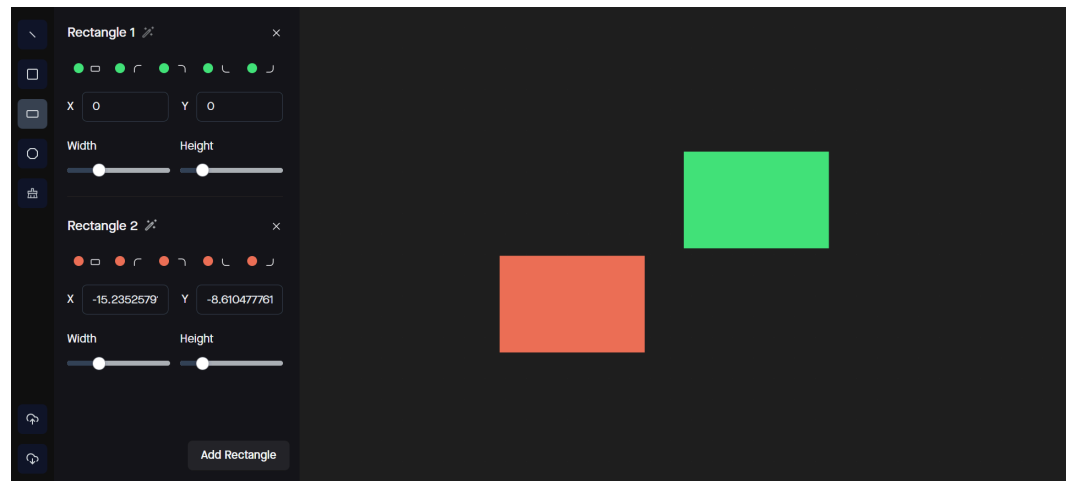
- Untuk menambahkan persegi, klik panel Square pada *sidebar*, lalu klik tombol Add Square.
- Persegi yang telah ditambahkan akan tercatat pada sisi kiri aplikasi, dengan ID sesuai urutan penambahan untuk mengidentifikasi persegi.
- Warna keseluruhan persegi dapat diubah dengan mengklik simbol persegi di bawah nama persegi yang akan memunculkan color picker. Apabila warna ini diubah, maka warna ini akan meng-*override* seluruh warna sudut persegi. Warna sudut juga dapat diubah dengan memilih mengklik ikon sudut yang akan diubah (kiri atas, kanan atas, kiri bawah, atau kanan bawah) yang akan memunculkan *color picker*.



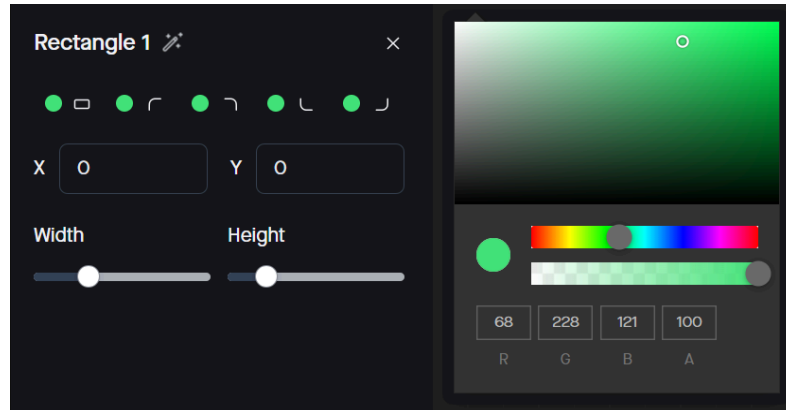
- Parameter yang dapat diatur adalah posisi sudut kiri bawah persegi melalui *field* X dan Y serta panjang sisi persegi melalui *slider* Side Length

c. Persegi panjang

Berikut adalah tampilan fungsionalitas persegi panjang



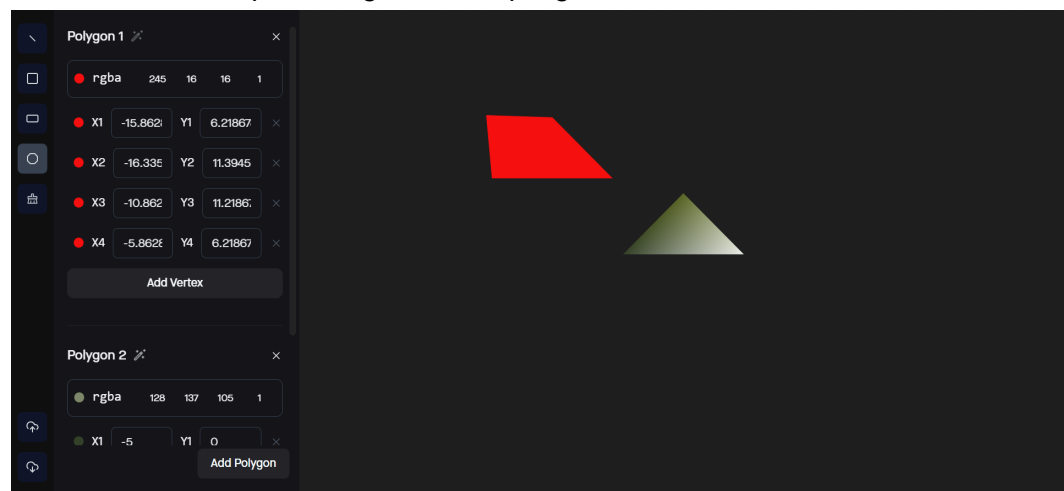
- Untuk menambahkan persegi panjang, klik panel Rectangle pada *sidebar*, lalu klik tombol Add Rectangle.
- Persegi panjang yang telah ditambahkan akan tercatat pada sisi kiri aplikasi, dengan ID sesuai urutan penambahan untuk mengidentifikasi persegi panjang.
- Warna keseluruhan persegi panjang dapat diubah dengan mengklik simbol persegi di bawah nama persegi panjang yang akan memunculkan color picker. Apabila warna ini diubah, maka warna ini akan meng-*override* seluruh warna sudut persegi panjang. Warna sudut juga dapat diubah dengan memilih mengklik ikon sudut yang akan diubah (kiri atas, kanan atas, kiri bawah, atau kanan bawah) yang akan memunculkan *color picker*.



- Parameter yang dapat diatur adalah posisi sudut kiri bawah persegi melalui *field* X dan Y serta lebar (*width*) dan tinggi (*height*) persegi melalui *slider*.

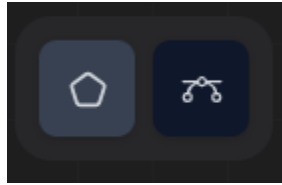
d. Poligon

Berikut adalah tampilan fungsionalitas poligon.

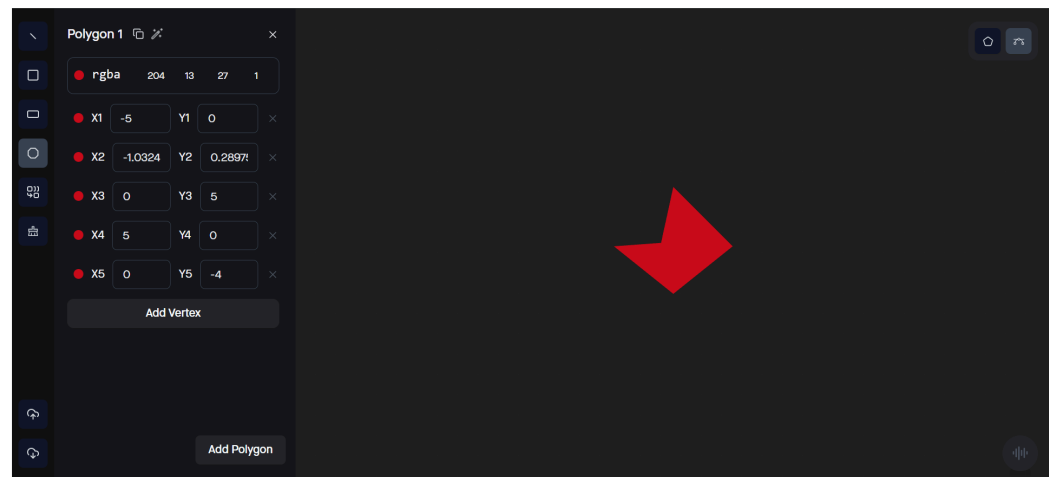


- Untuk menambahkan poligon, klik panel Polygon pada *sidebar*, lalu klik tombol Add Polygon.
- Poligon yang telah ditambahkan akan tercatat pada sisi kiri aplikasi, dengan ID sesuai urutan penambahan untuk mengidentifikasi poligon.
- Warna keseluruhan poligon dapat diubah dengan yang akan memunculkan color picker. Apabila warna ini diubah, maka warna ini akan meng-*override* seluruh warna sudut persegi panjang. Warna sudut juga dapat diubah dengan memilih mengklik ikon sudut yang akan diubah (kiri atas, kanan atas, kiri bawah, atau kanan bawah) yang akan memunculkan *color picker*.
- Terdapat dua opsi poligon, yaitu *convex* dan *free*. Untuk opsi *convex*, poligon yang ditampilkan selalu merupakan *convex hull* dari tiap *vertex*-nya, sedangkan pada opsi *free*, poligon yang ditampilkan tidak

harus berbentuk *convex*. Untuk mengakses fitur tersebut, klik opsi poligon pada sisi kanan atas layar.



- Berikut adalah tampilan poligon yang *convex* dan yang *free* secara berurutan.



e. *Drag and drop*

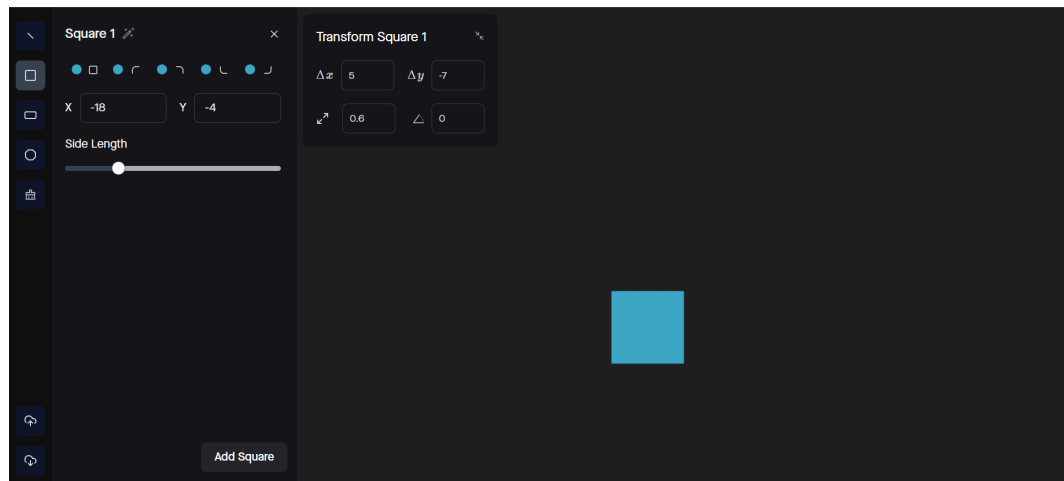
Elemen dapat dipindahkan dengan *drag and drop*. Untuk memindahkan elemen, dekatkan kursor ke elemen hingga kursor berbentuk tangan, lalu klik dan pindahkan elemen ke posisi yang diinginkan.

f. Penggeseran titik sudut dengan *mouse*

Titik sudut pada elemen juga dapat dipindahkan tanpa memindahkan keseluruhan elemen. Mekanismenya kurang lebih sama dengan *drag and drop*, namun pastikan meng-*hover* kursor ke dekat sudut/*vertex* dari elemen tersebut.

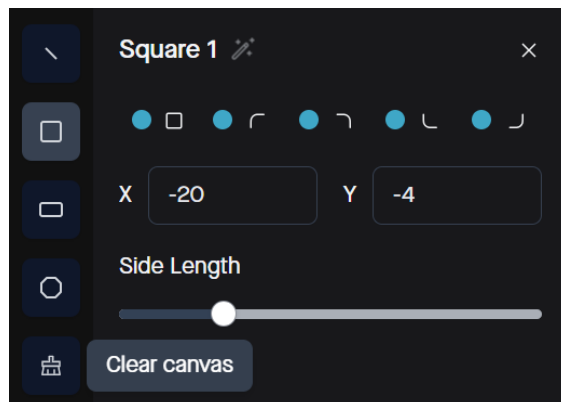
g. Transformasi geometri

Setiap elemen dapat dilakukan transformasi geometri berupa translasi, dilatasi, dan rotasi. Untuk mengakses fitur transformasi, klik tombol berbentuk tongkat sulap di samping nama elemen. Kemudian, isi parameter yang akan diubah berupa Δx dan Δy untuk translasi, skala untuk dilatasi, serta sudut untuk rotasi. Elemen akan ditransformasi tepat setelah terjadi perubahan pada salah satu parameter. Elemen yang ditampilkan adalah elemen hasil transformasi.



h. Clear canvas

Terdapat fitur untuk mengosongkan *canvas*, yang dapat diakses dengan tombol sapu pada *navbar*.



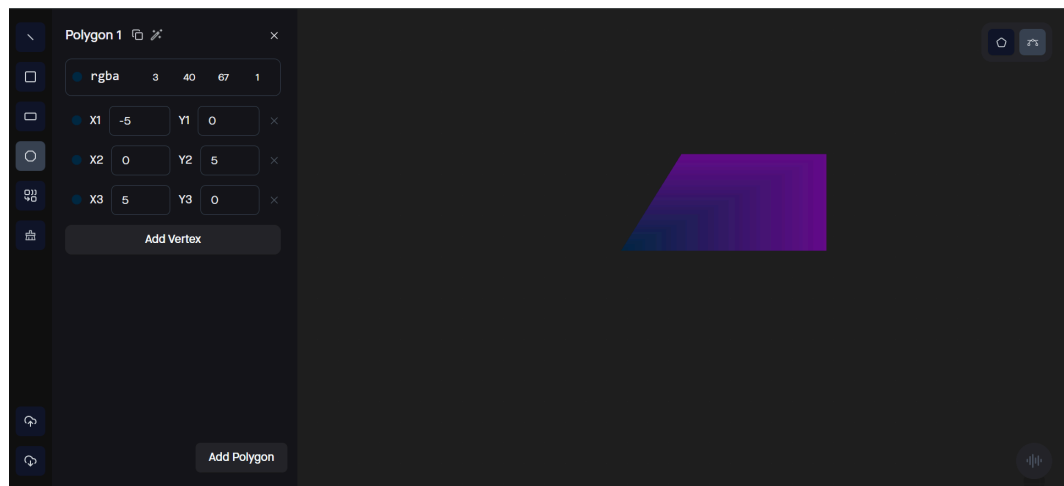
i. Load dan save model dalam format json

Di bagian bawah *navbar*, terdapat tombol untuk *load* dan *save* model dalam format json.



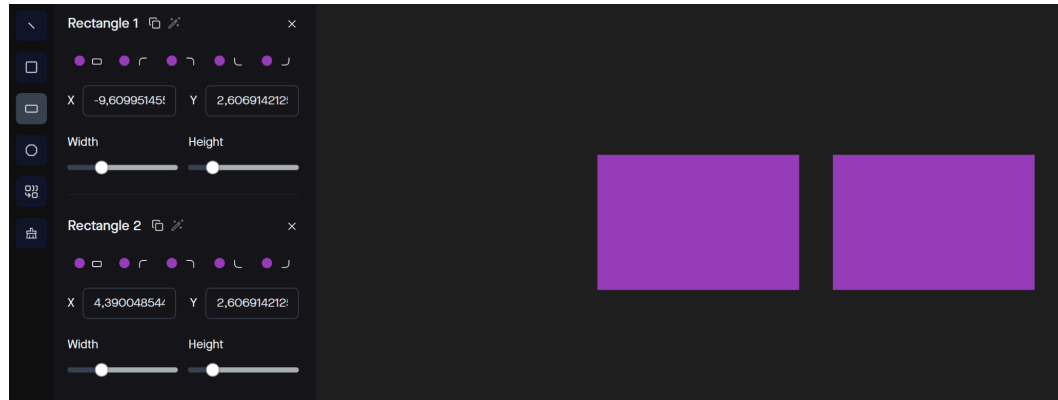
j. *Combine shapes*

Fitur ini dapat menggabungkan seluruh elemen menjadi satu seperti pada gambar berikut. Cara menggunakannya adalah dengan menekan tombol *combine shapes* pada panel di bagian kiri layar



k. *Duplicate shape*

Fitur ini adalah fitur tambahan yang berfungsi untuk menduplikasi suatu elemen. Fitur ini dapat digunakan untuk setiap jenis *shape*, di mana tombol untuk melakukan duplikasi terletak di sebelah kanan nama elemen.

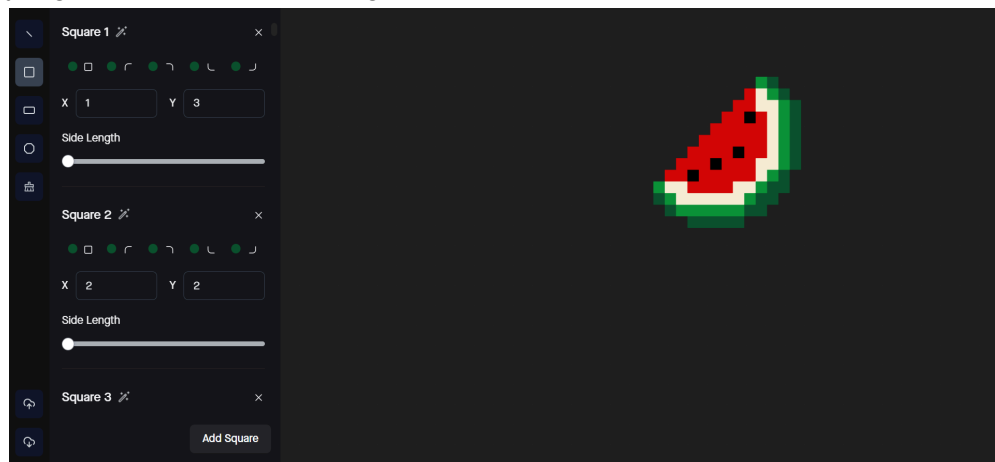


Catatan: Untuk seluruh *field* yang dapat diubah menggunakan *keyboard*, perubahan *field* tersebut juga dapat dilakukan dengan menekan tombol *up* (↑) untuk menambahkan 1 pada *field* tersebut dan tombol *down* (↓) untuk mengurangi 1 pada *field* tersebut

3. Contoh Hasil Input

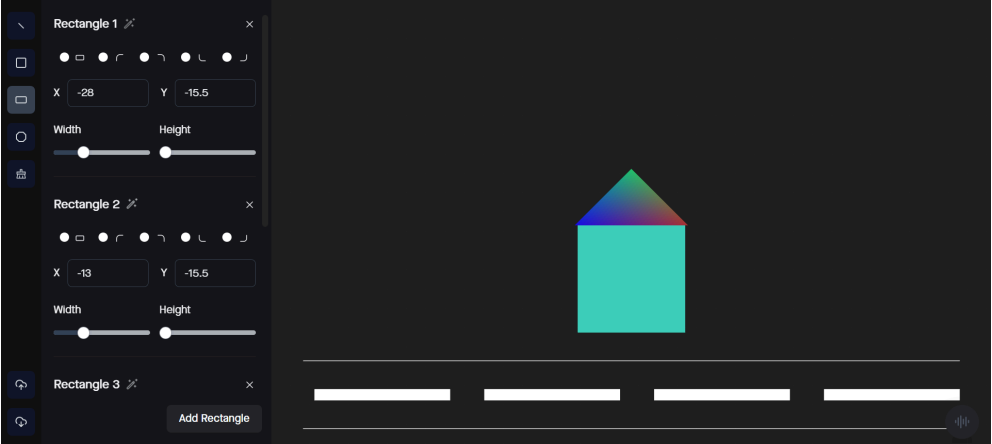
a. semangka.json

Merupakan *pixel art* berbentuk semangka yang terdiri atas sekitar 104 kotak yang direpresentasikan dengan elemen *square*



b. rumah.json

Berisi gambar rumah yang terdiri atas bangunan dengan elemen *square*, atap dengan elemen *polygon*, dan jalan dengan elemen *line* dan *rectangle*.



4. LAMPIRAN

Deployed Website: <https://hello-triangle.vercel.app/>

Repository: <https://github.com/GAIB20/tugas-besar-grafkom-1-hellotriangle.git>