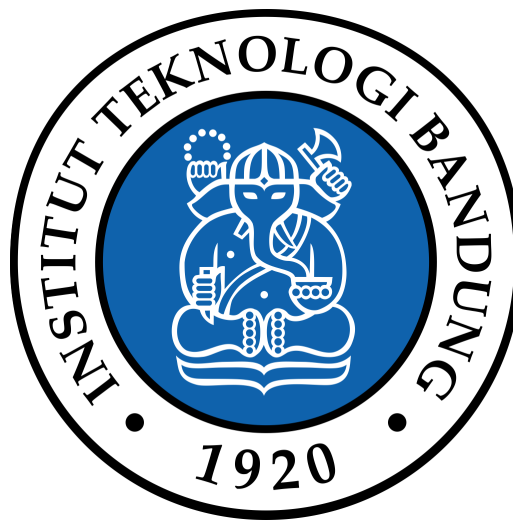


# **2D Web Based CAD**

## **(Computer-Aided Design)**

Ditujukan untuk memenuhi tugas besar 1 mata kuliah

IF3260 - Grafika Komputer



Disusun oleh:

Muhammad Ziad Rahmatullah	13518023
Samuel	13518041
Fatkhan Masruri	13518053

K2

Teknik Informatika

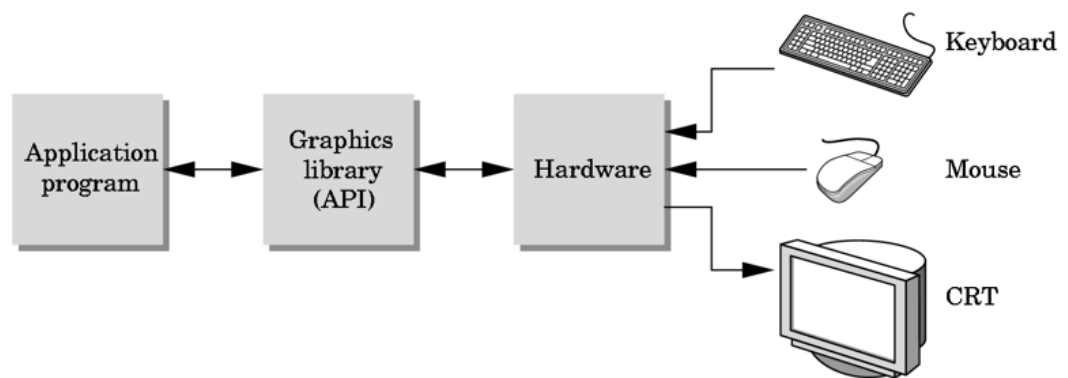
**SEKOLAH TEKNIK ELEKTRO DAN  
INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
BANDUNG  
2020**

## **1. Deskripsi**

Pada kesempatan kali ini, kami diminta untuk membuat aplikasi web yang dapat membantu dalam pembuatan denah arsitektur. Aplikasi web ini dibuat dengan menggunakan WebGL dan Vue.js.

WebGL adalah sebuah *graphic library* (GL) yang dibangun berdasarkan *graphic library* OpenGL ES 2.0. Namun, sejak Maret 2020, ada *major version* baru dari *graphic library* ini, yaitu WebGL2 yang dibangun berdasarkan *graphic library* OpenGL ES 3.0.

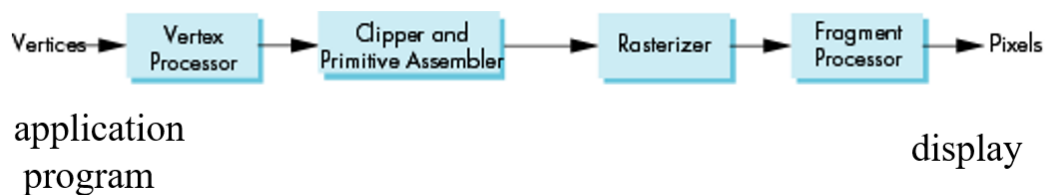
Jika OpenGL berjalan langsung di atas sistem operasi dari mesin yang digunakan dan kemudian berinteraksi langsung dengan driver GPU pada mesin tersebut, WebGL memiliki alur yang sedikit berbeda. Pada WebGL, kode tidak dibuat dengan menggunakan bahasa C seperti OpenGL, tetapi dibuat dengan menggunakan HTML dan JavaScript. Oleh karena itu, kode yang dijalankan sebenarnya berjalan di atas *browser* pengguna. Karena hal tersebut, salah satu kelemahan WebGL dibandingkan dengan OpenGL adalah kecepatannya yang lebih lambat. Selain itu, karena WebGL dibangun berdasarkan OpenGL ES 2.0/3.0, ada beberapa limitasi terhadap fitur-fitur WebGL dibandingkan dengan OpenGL. Walaupun begitu, salah satu keunggulan WebGL adalah *setup time* yang sangat mudah (bahkan hampir tidak ada) karena *library* ini digunakan langsung di *browser* dan tentunya seperti yang kita tahu, *browser* merupakan sebuah aplikasi *cross-platform*. Hal ini berbanding jauh dengan OpenGL yang sangat bergantung kepada OS dari mesin yang digunakan sehingga *setup* awal dari OpenGL cukup memakan waktu dan tenaga.



**Gambar 1.** Alur proses pengiriman data dari aplikasi hingga hardware

Dalam tugas ini kami diminta untuk membuat sebuah objek yang dapat melakukan *scaling*, rotasi, dan translasi dengan suatu besaran tertentu sesuai dengan keinginan pengguna. Hal ini diperlukan karena untuk membantu pengguna dalam membangun sebuah desain dengan ukuran dan model yang diinginkan. Objek yang dapat digambarkan terbatas pada bidang 2 dimensi dan garis.

Bidang dua dimensi tersebut dibuat dengan cara mendefinisikan koordinat-koordinatnya di dalam sebuah *vertex list* terlebih dahulu. Setelah itu, *vertex list* ini akan dikirim WebGL ke GPU untuk diolah dan diproses sehingga bisa menghasilkan grafis ke layar pengguna. Untuk memahami proses tersebut lebih dalam dan lebih baik lagi, kita dapat melihat urutan dari proses-proses tersebut pada gambar di bawah ini.



**Gambar 2.** Pipeline proses pengubahan dari *vertices* hingga menjadi *pixels*

Karena objek yang digambarkan hanya terbatas pada bidang 2 dimensi, transformasi yang dilakukan juga hanya terbatas pada transformasi pada bidang 2 dimensi, yaitu translasi, rotasi terhadap sumbu X atau sumbu Y, dan *scaling*. Namun, pada transformasi-transformasi ini, kita tidak benar-benar mengubah posisi dari objek

yang diletakkan. Jika kita melakukan transformasi terhadap objek tersebut secara langsung,

## 2. Hasil

### a. Garis

Untuk menggambar garis kita menggunakan input array of vertex dengan 4 buah angka float yang merepresentasikan point di sumbu x dan y. Kemudian untuk rendering kita perlu memberikan input bertipe LINE.



Gambar 3. Contoh garis pada aplikasi

### i. *Scaling*

Fitur scaling adalah pengguna dapat melakukan resize yaitu memanjang atau memendekkan garis dengan cara menggeser dan menekan mouse di canvas. Untuk melebarkan maka perlu menggeser mouse ke arah mana saja sehingga bisa membuat gradien dari garis tersebut berubah bersamaan dengan panjangnya..

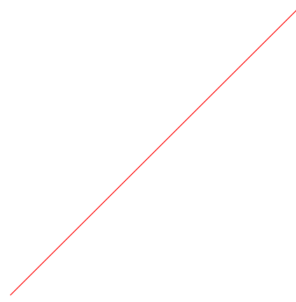
Contoh Demo :

- Memendekan garis dengan gradien 0



Gambar 4. Contoh garis yang dipendekkan

- Memanjangkan garis dengan gradien 1



**Gambar 5. Contoh garis yang diperpanjang**

## **ii. Menggeser**

Fitur menggeser adalah pengguna dapat melakukan pergeseran titik sumbu misalnya awalnya titik sumbu dari objek adalah 0,0 maka setelah dilakukan pergeseran ke sumbu x positif dan y positif menjadi ke titik 3,3. Untuk memindahkan garis ke posisi atas maka pengguna perlu menekan dan menggeser mouse ke atas dan sebaliknya. Sedangkan untuk menggeser garis ke samping maka pengguna perlu melakukan menggeser mouse dan menekannya ke samping. Berikut ini contoh melakukan pergeseran pada garis.



**Gambar 6. Contoh garis yang dipindahkan**

## **iii. Rotasi**

Rotasi pada garis hampir sama dengan scaling namun bedanya, pada rotasi, panjang dari garis tidak akan berubah saat mouse digerakan. Rotasi adalah melakukan perputaran objek dengan poros titik tengah. Objek akan diputar sesuai dengan sudut perputaran. Mouse tinggal digerakan ke kanan, maka garis akan rotate *anticlockwise*, dan saat digerakan ke kiri maka garis akan rotate *clockwise*



**Gambar 7.** Contoh garis yang dirotasi secara *clockwise*

#### **b. Persegi**

Untuk menggambar persegi kita perlu melakukan input array of vertex dengan 8 buah angka float yang merepresentasikan point di sumbu x dan y. Kemudian untuk rendering kita perlu memberikan input bertipe TRIANGLE\_FAN.



**Gambar 8. Contoh segiempat pada aplikasi**

**i. *Scaling***

Fitur scaling adalah pengguna dapat melakukan resize yaitu mengecilkan atau membesarkan bidang persegi dengan cara menggeser dan menekan mouse di kanvas. Untuk melebarkan maka perlu menggeser mouse ke sumbu x dengan x positif akan melebarkan dan x negatif akan mengecilkan. Sedangkan untuk memanjang persegi kita perlu menekan dan menggeser mouse ke sumbu y dengan y positif akan memanjangkan dan y negatif untuk mengecilkan.

Contoh demo:

- menggeser mouse ke sumbu y positif:



**Gambar 9. Contoh segiempat yang di-scale ke arah sumbu y positif**

- Menggeser mouse ke sumbu x positif



**Gambar 10. Contoh segiempat yang di-scale ke arah sumbu x positif**

**ii. Menggeser**

Fitur menggeser adalah pengguna dapat melakukan pergeseran titik sumbu misalnya awalnya titik sumbu dari objek adalah 0,0 maka setelah dilakukan pergeseran ke sumbu x positif dan y positif menjadi ke titik 3,3. Untuk memindahkan persegi ke posisi atas maka pengguna perlu menekan dan menggeser mouse ke atas dan sebaliknya. Sedangkan untuk menggeser persegi ke samping maka pengguna perlu melakukan menggeser mouse dan menekannya ke samping. Berikut ini contoh melakukan pergeseran pada bidang persegi.



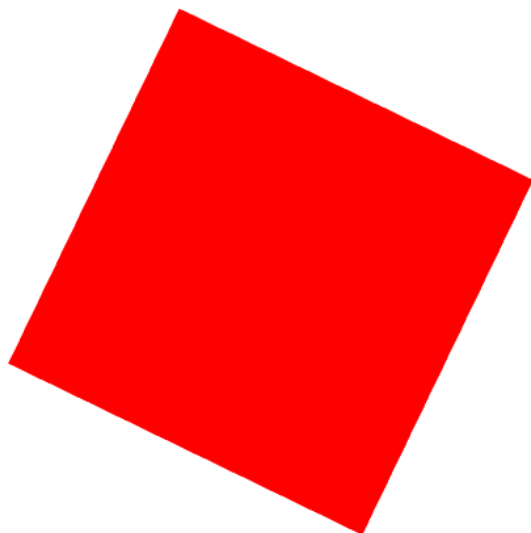




**Gambar 11. Contoh segiempat yang digeser**

**iii. Rotasi**

Rotasi adalah melakukan perputaran objek dengan poros titik tengah. Objek akan diputar sesuai dengan sudut perputaran. Pada implementasi tugas ini untuk menggeser objek/persegi pengguna perlu menggeser dan menahan cursor ke kanan atau ke kiri. Pengguna menggeser ke kanan untuk merotasi persegi searah jarum jam dan menggeser ke kiri untuk merotasi persegi berlawanan arah jarum jam.



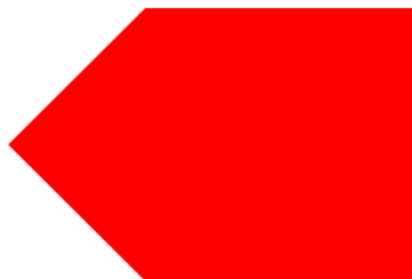
**Gambar 12. Contoh segiempat yang telah dirotasi**

### c. Poligon

Poligon merupakan sebuah bangun datar yang terdiri dari beberapa sisi. Sebenarnya, segiempat sendiri merupakan sebuah poligon. Oleh karena itu, apa yang akan Anda lihat di bawah ini, seharusnya tidak berbeda dengan apa yang Anda telah lihat pada bagian sebelumnya, yaitu segiempat. Poligon ini diimplementasi dengan menggunakan koordinat-koordinat dari setiap titik sudutnya. Setelah itu, kita akan mengirimkannya ke *shaders* untuk digambar dengan memanfaatkan tipe `TRIANGLE_FAN`. *Vertex-vertex* yang kurang akan di-*generate* sendiri oleh *shaders* tersebut.

#### i. *Scaling*

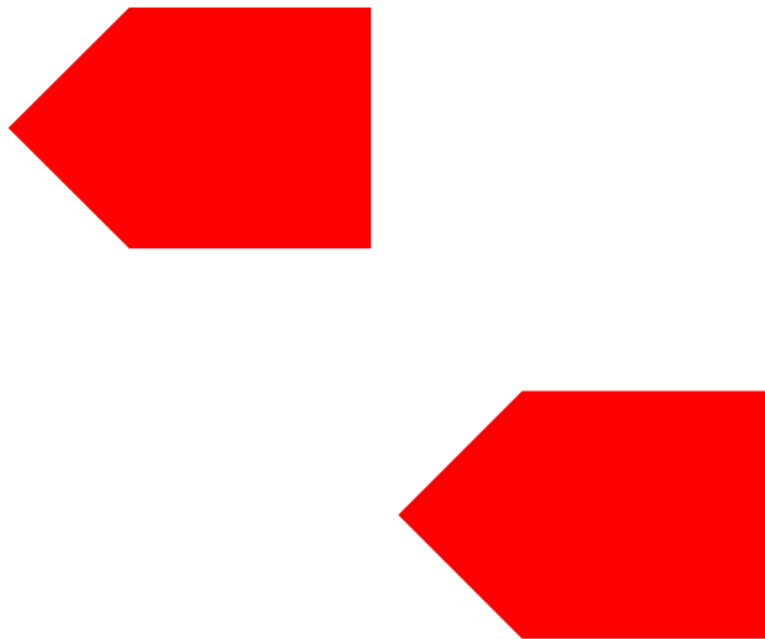
Seperti yang sudah dijelaskan sebelumnya, fitur *scaling* adalah fitur *resize* yang dapat digunakan untuk mengubah ukuran dari bangunan poligon yang ada di *canvas*. Fitur *scaling* ini bisa digunakan untuk memperbesar ataupun memperkecil berbagai bagian bangunan ini.



**Gambar 13. Contoh poligon (trapesium) yang di-scale**

**ii. Menggeser**

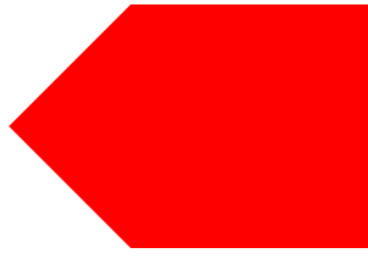
Untuk fitur ini, pengguna bisa melakukan translasi terhadap bidang yang diinginkan. Translasi yang dilakukan bisa terhadap sumbu X, sumbu Y, ataupun ke sumbu imajinatif yang berada di antara sumbu X dan sumbu Y. Harap perhatikan gambar berikut ini.



**Gambar 14. Contoh poligon (trapesium) yang ditranslasikan terhadap sumbu X**

**iii. Rotasi**

Pada fitur ini, poligon bisa melakukan transformasi rotasi di *canvas* yang ada di *browser*. Rotasi yang dilakukan bisa secara *clockwise* ataupun juga *counter-clockwise*. Namun, pengguna harus berhati-hati dalam melakukan rotasi agar rotasi yang dilakukan tidak berlebihan sehingga objek menghilang ke luar *canvas*. Agar Anda dapat lebih memahami proses ini, harap perhatikan gambar di bawah ini.



Gambar 15. Contoh poligon (trapesium) yang dirotasi dengan arah *clockwise*

### 3. Manual Fungsionalitas Program

Pada 1 halaman web, terdapat 3 canvas yang digunakan, pada canvas pertama, dapat melakukan scalling, menggeser dan rotasi pada sebuah garis, pada canvas kedua dapat melakukan *scaling*, menggeser dan rotasi pada sebuah persegi, dan pada canvas ketiga dapat melakukan scalling, menggeser dan rotasi pada sebuah poligon. Untuk saat ini, aplikasi ini tidak bisa menerima *input* dari file yang diinginkan pengguna. Oleh karena itu, poligon yang ada di layar hanyalah poligon yang sudah di-*hard-code* oleh kami ke dalam program. Poligon itu saat ini adalah trapesium.

Di bawah setiap *canvas*, ada tiga *radio button* untuk pemilihan mode transformasi yang ingin dilakukan pengguna. Apakah translasi (geser), atau rotasi, atau juga *scaling* benda yang diinginkan.

☐ Geser ☒ Rotate ☐ Scale

**Gambar 16. Contoh tombol pilihan mode yang ada di bawah setiap *canvas***