

IF3260 - Grafika Komputer
Laporan Tugas Besar 2 - 3D WebGL Hollow Object



Disusun oleh:

Dzaky Fattan Rizqullah	13520003
Muhammad Garebaldhie Er Rahman	13520029
Taufan Fajarama Putrawansyah R	13520031

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2023

1. Deskripsi Program

3D WebGL Hollow Object Viewer adalah sebuah program berbasis *web* jenis *3D Object Viewer* yang dibangun menggunakan kakas WebGL. WebGL adalah *library* yang digunakan untuk menampilkan *render* dari objek-objek sederhana pada *web*, salah satunya adalah objek tiga dimensi. Aplikasi dapat menampilkan model *3D Hollow* yang disediakan oleh aplikasi, yang dipilih oleh pengguna.

Model yang telah digambar pada *canvas* dapat ditransformasi, dalam hal ini melakukan translasi, rotasi dalam tiga sumbu X, Y, dan Z (*pitch, yaw, roll*), dan dilatasi. Model juga dapat ditentukan jenis *shading*-nya, dalam hal ini apakah menggunakan *flat shading* atau menggunakan *lighting*. Selain itu, pengguna dapat memanipulasi properti dari kamera yang menampilkan model *3D Hollow*. Manipulasi yang dapat dilakukan berupa translasi, rotasi dalam tiga sumbu X, Y, dan Z, perubahan nilai *Field Of View, theta, phi*, serta dapat mengubah jenis proyeksi, apakah menggunakan *orthographic, oblique, atau perspective*. Terakhir, aplikasi memiliki fitur menyimpan dan memuat model *3D Hollow*. Model disimpan menjadi file *.json* berisi informasi *vertices, faces*, dan warna dari model.

Berikut merupakan fitur yang diimplementasikan pada tugas besar ini

- Transformasi geometri berupa
 - ***Translate***
 - ***Scaling***
 - ***rotate***
- Proyeksi berupa
 - ***Orthographic***
 - ***Oblique***
 - ***Perspective***
- Teknik shading dengan ***directional lightning (lambert)***
- Pembentukan tiga objek
 - ***Kubus***
 - ***Tetrahedron***
 - ***Octahedron***
- Kamera dengan fitur
 - ***Translasi pada sumbu x, y, z (radius)***
 - ***Rotasi berupa pitch, yaw, roll***
- Load JSON object
- [Bonus] Save JSON objek yang sudah di transformasi
- [Bonus] Animasi

2. Hasil Implementasi

Untuk menampilkan kesan 3D, terdapat fitur yang harus ditampilkan pada WebGL yaitu **gl.DEPTH_TEST** serta **gl.CULL_FACE**. Depth test akan meng-enable depth buffer yang akan memberikan kesan 3D. Secara normal, WebGL hanya dapat menampilkan bidang yang memiliki arah normal keluar (inward faces) dan ketika bidang tersebut diputar maka WebGL tidak bisa merendernya. CULL_FACE merupakan fitur yang dapat diaktifkan untuk mengatasi masalah itu sehingga WebGL dapat merender (backward faces) juga. Jika bidang kita ingin dapat dilihat pada depan maupun belakang, maka kita harus membuat dua bidang yang memiliki arah berlawanan atau dapat dikatakan membuat bidang dengan inward dan backward faces.

Model *3D Hollow* ditampilkan dengan memanfaatkan informasi *vertices*, *faces*, *colors*, dan *normals* pada file. Model ditampilkan sedemikian rupa dengan **gl.drawElements()** dengan tipe **gl.TRIANGLES**. Kode program GL serta inisialisasinya diimplementasikan pada file `webgl.js`. Keseluruhan operasi proyeksi dan transformasi menggunakan perkalian matriks yang diimplementasikan pada file `matrices.js`

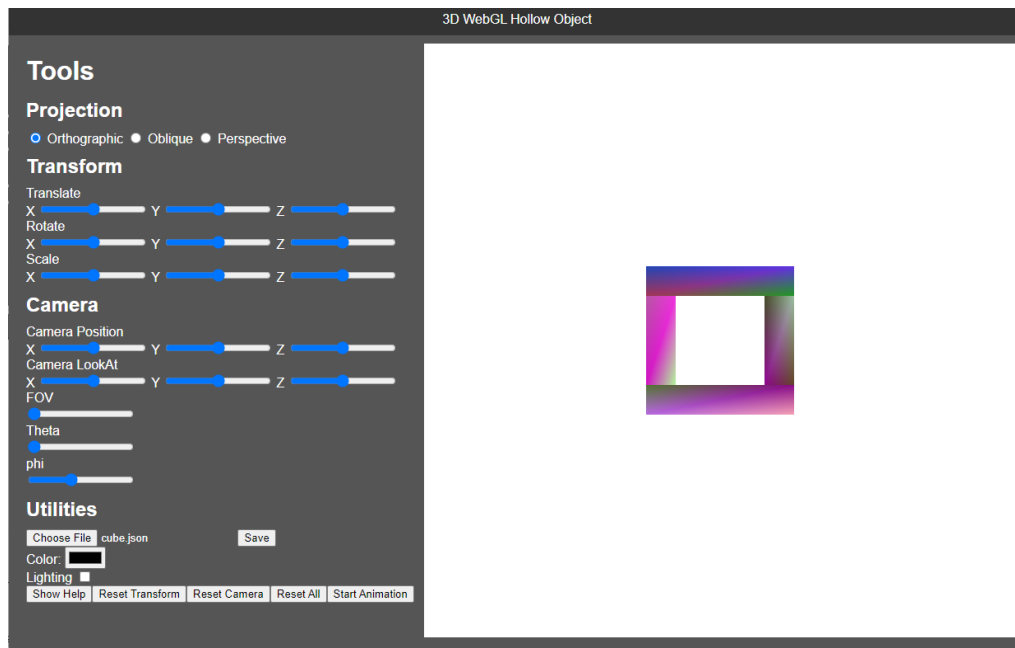
Pada tugas kecil ini, terdapat tiga buah model 3D Hollow, yaitu Kubus, Tetrahedron, dan Octahedron. Informasi terkait *vertices*, *faces*, *colors*, dan *normals* disimpan menjadi suatu javascript object, yang dapat disimpan menjadi file `.json`. File `.json` ini dapat dimuat kembali pada program. Fungsi menyimpan dan memuat objek diimplementasikan pada file `objHandler.js`.

Beberapa fungsi utilitas juga diimplementasikan untuk membantu keberjalanan program, yang keseluruhan implementasinya dibuat pada file `utils.js`. Fungsi-fungsi utilitas ini adalah sebagai berikut:

1. Kalkulasi *normals* pada *faces*,
2. Kalkulasi jarak *Euler*,
3. Pembentukan *faces* dan model *3D* dari *vertices*
4. Kalkulasi normalisasi nilai,
5. Pengurangan vektor serta *cross-product* dari vektor.

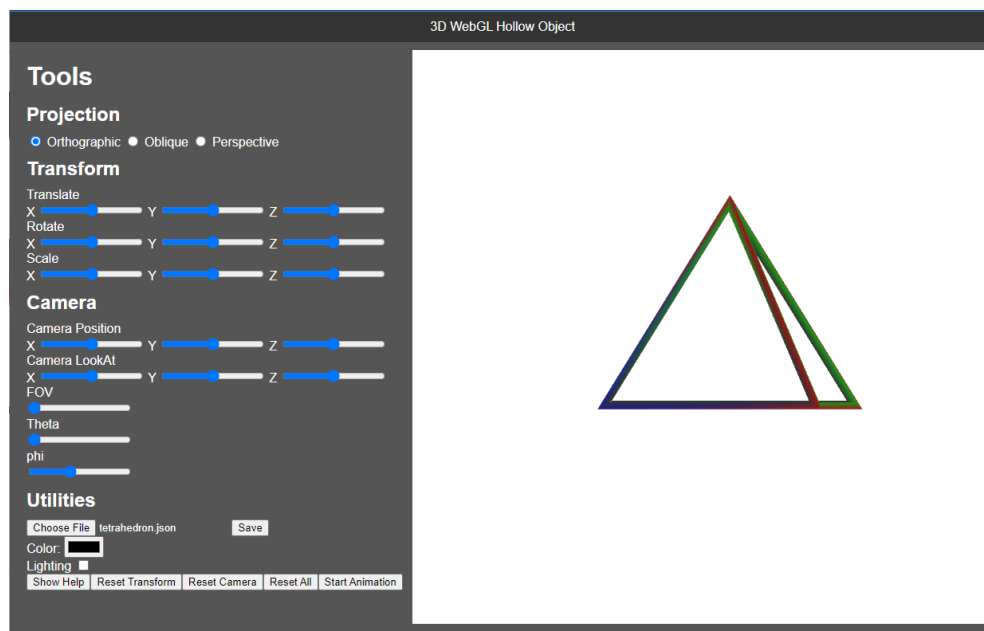
Hasil implementasi program ini ditampilkan pada gambar berikut.

Objek 1 (Kubus)

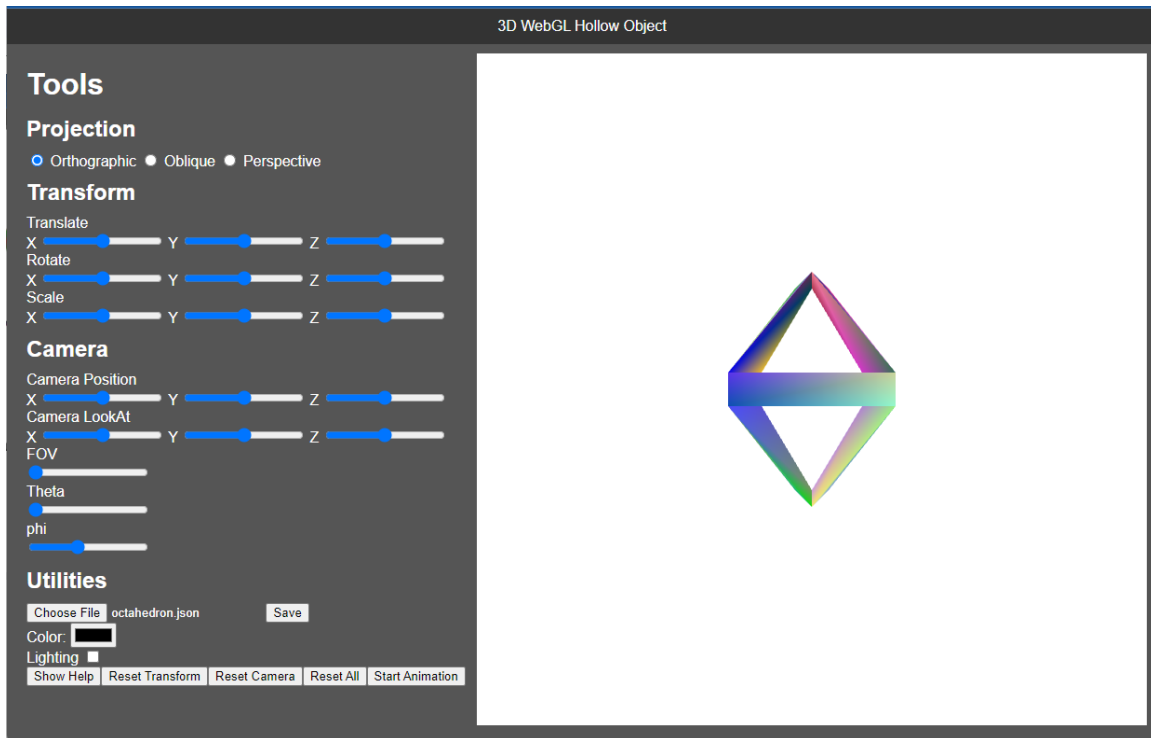


Kubus dibuat dengan cara membuat 12 balok kecil pada setiap sisinya

Objek 2 (Tetrahedral)



Objek 3 (Octahedral)



JSON

Setiap objek yang dibuat, akan memiliki empat atribut yaitu *vertices*, *faces*, *colors*, dan *normals*. Ketika objek di save ataupun di load, semua objek harus memiliki keempat attribut ini agar dapat di render

- Vertices akan menyimpan koordinat koordinat vertex yang sudah dibuat dalam bentuk array dua dimensi
- Faces merupakan array dua dimensi yang akan menyimpan posisi index dari setiap sisi yang dibuat. Faces array akan berisi kumpulan backward dan inward faces dari objek yang dibuat
- Colors merupakan array dua dimensi yang menyimpan informasi warna dari setiap vertex yang telah dibuat
- Normals merupakan vector yang tegak lurus terhadap bidang. Fungsi dari vektor normal ini ialah pada proses shading. Ketika sedang dibuat suatu triangles, normal dapat dihitung dengan cara melakukan cross product antara kedua vector tersebut.

Berikut merupakan contoh tampilan objek yang dibuat

cubes.json

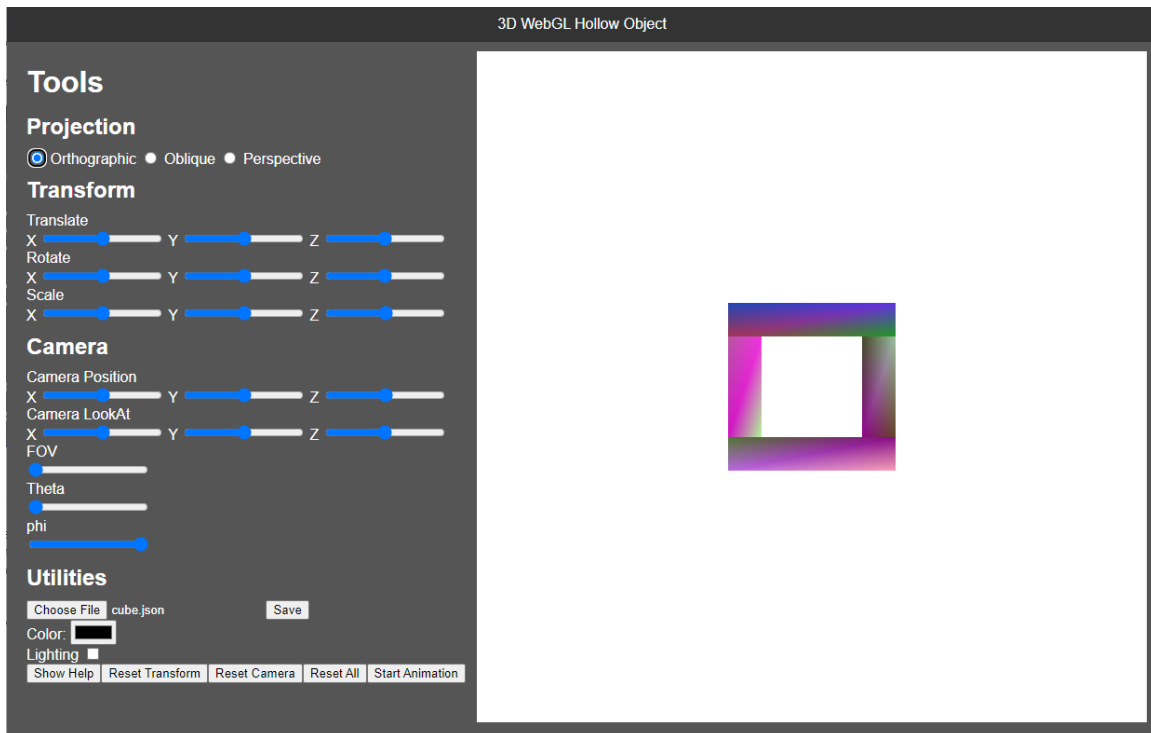
```
{
  "vertices": [
    [-0.5, -0.5, 1],
    [0.5, -0.5, 1],
    [0.5, -0.3, 1],
    ...snip...
  ],
  "faces": [
    [1, 2, 3],
    [1, 3, 2],
    [1, 3, 4],
    ...snip...
  ],
  "colors": [
    [0.7192185166388889, 0.4148960876362249, 0.8744833534407817],
    [0.9616128521822496, 0.6311746278786023, 0.7149131152478647],
    [0.5380442883714276, 0.05158987061656628, 0.5230237252200929],
    ...snip...
  ],
  "normals": [
    [0, 0, 1],
    [0, 0, 1],
    [0, 0, 1],
    ...snip...
  ],
}
```

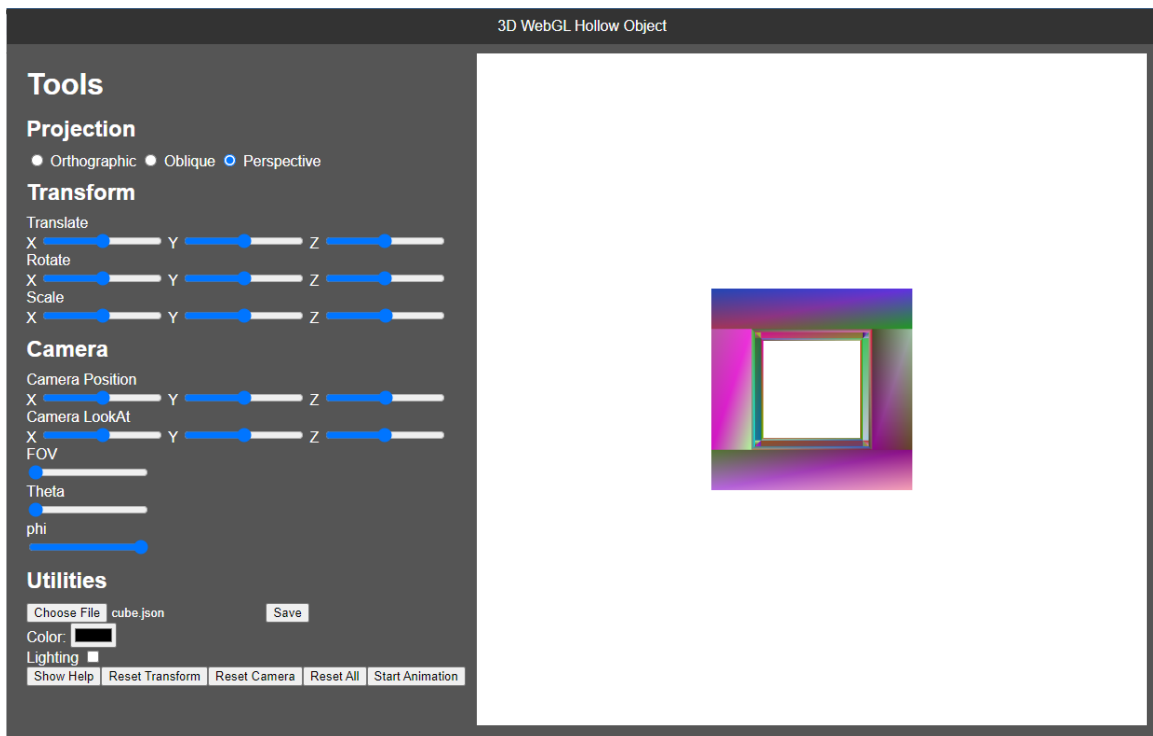
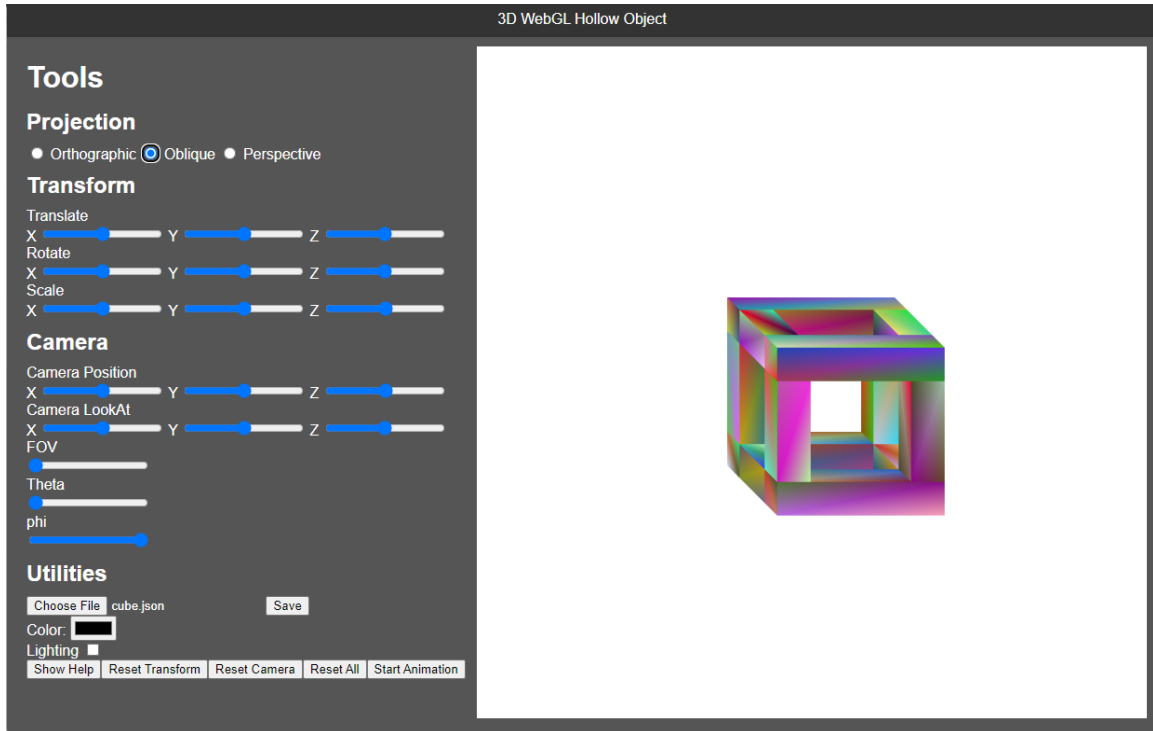
3. Fungsionalitas Program

Berikut merupakan penjelasan dari fungsi-fungsi yang dimiliki program.

1. Projection (Orthographic, Oblique, Perspective)

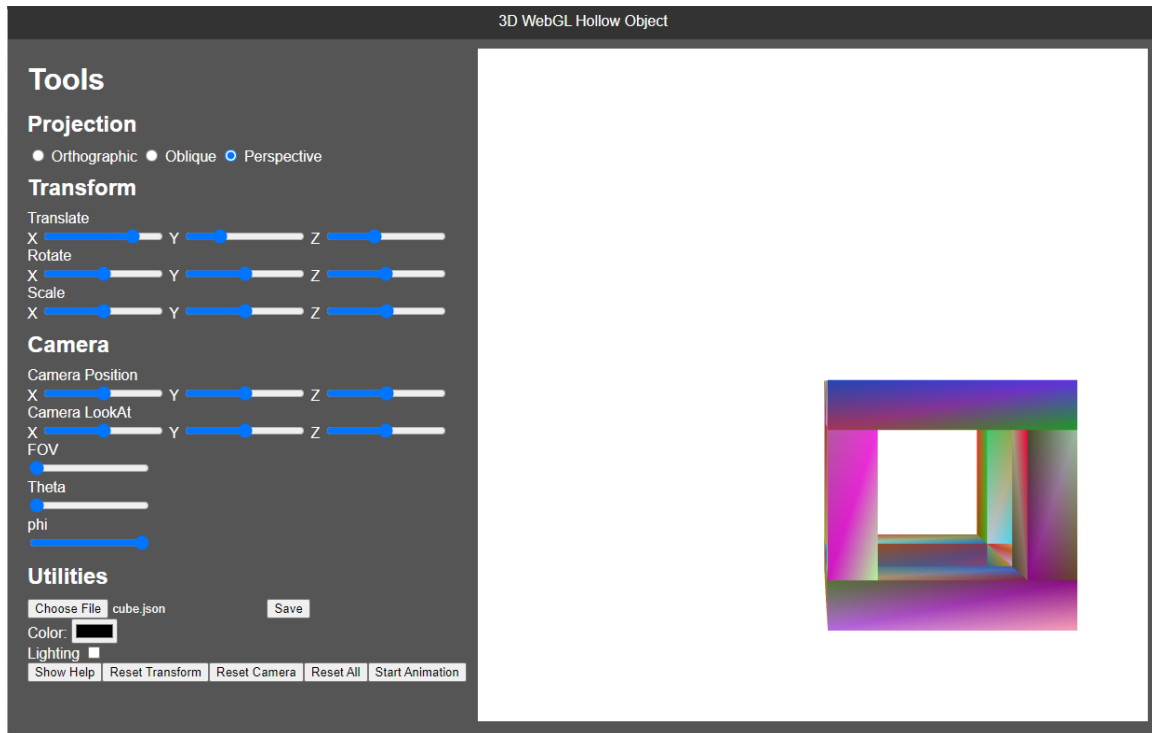
Program mengimplementasikan 3 jenis projection pada canvas WebGL yang digunakan. Projection diimplementasikan dengan matriks 4x4 sesuai projection yang dipilih dan kemudian digunakan dalam projection matrix WebGL.

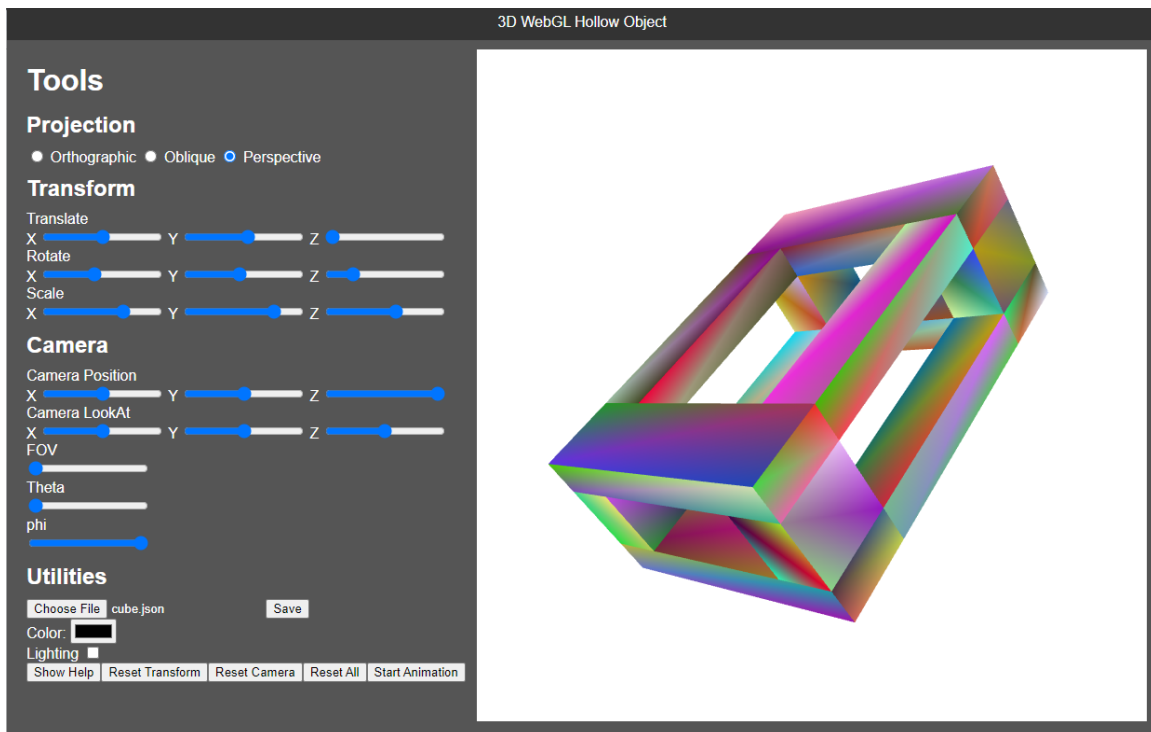
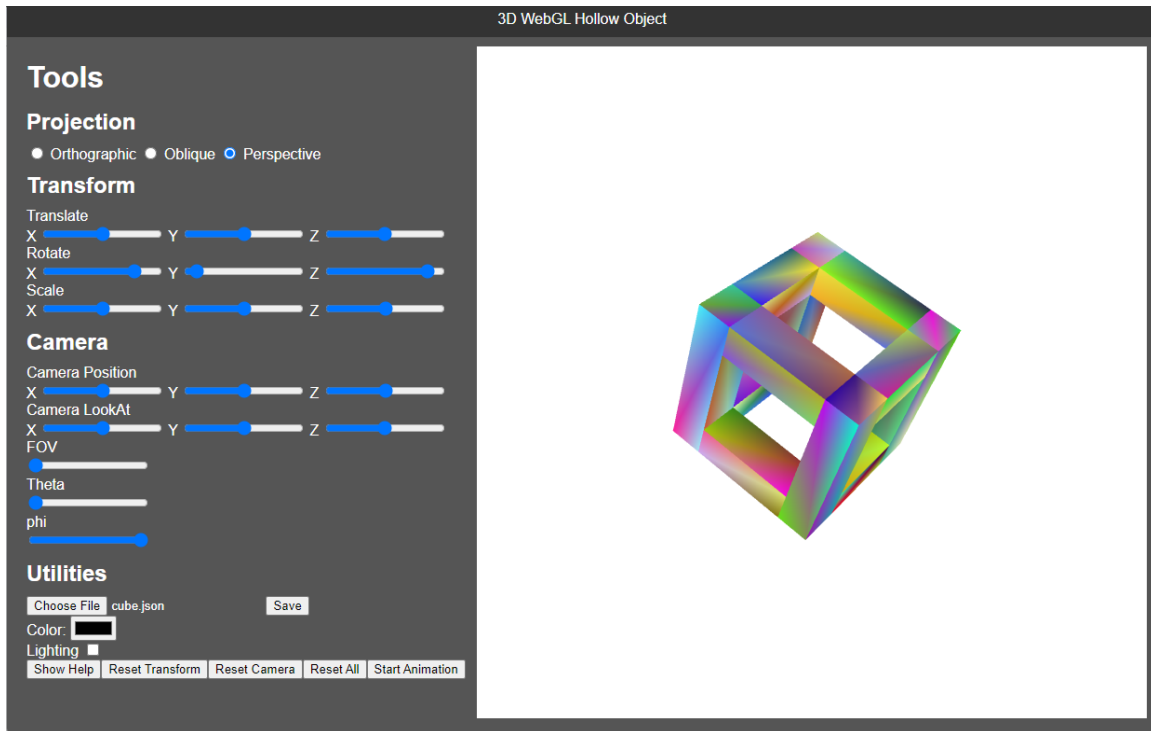




2. Transformasi Objek (Translasi, Rotasi, Dilatasi)

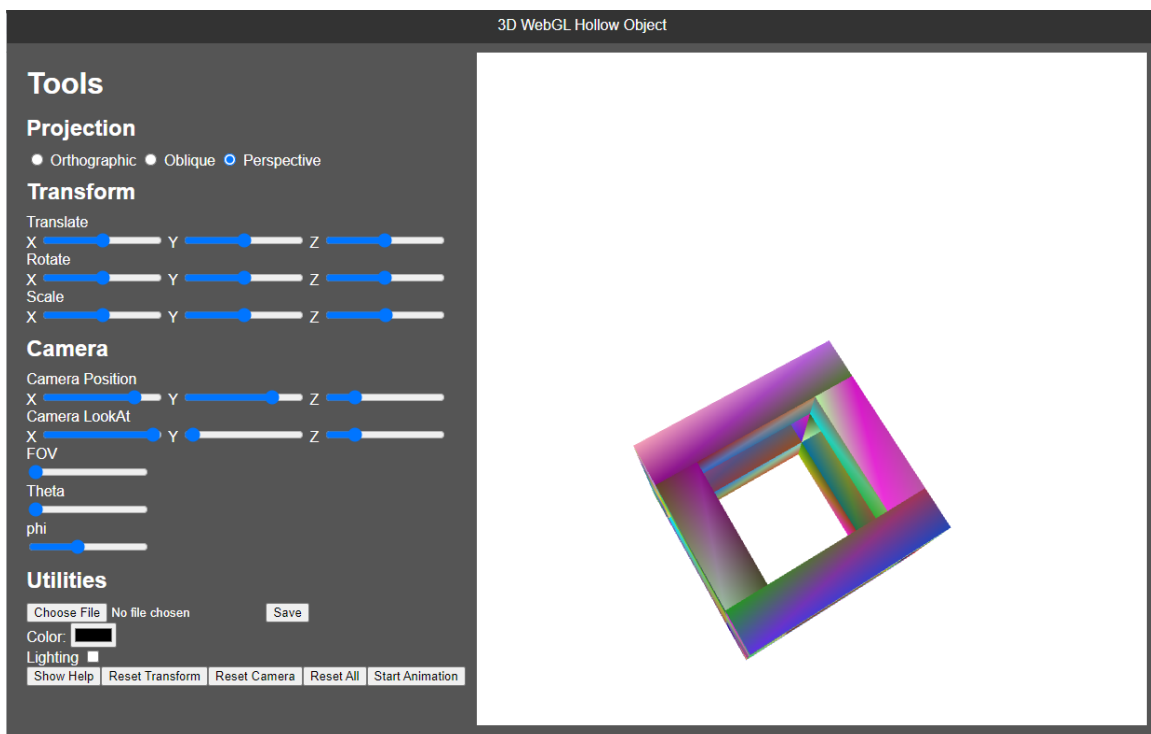
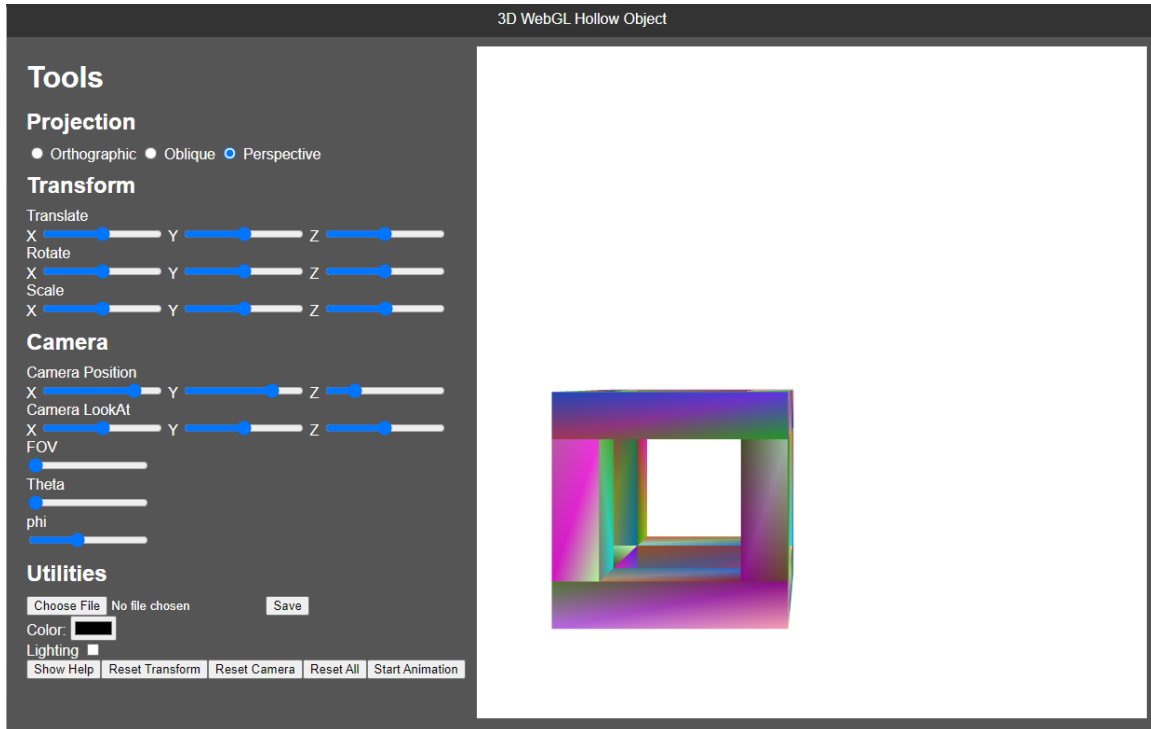
Program dapat melakukan transformasi berupa translasi, rotasi, dan dilatasi pada objek. Hal ini dilakukan dengan membentuk suatu matriks transformasi yang dimasukkan pada program GL yang telah dibuat.





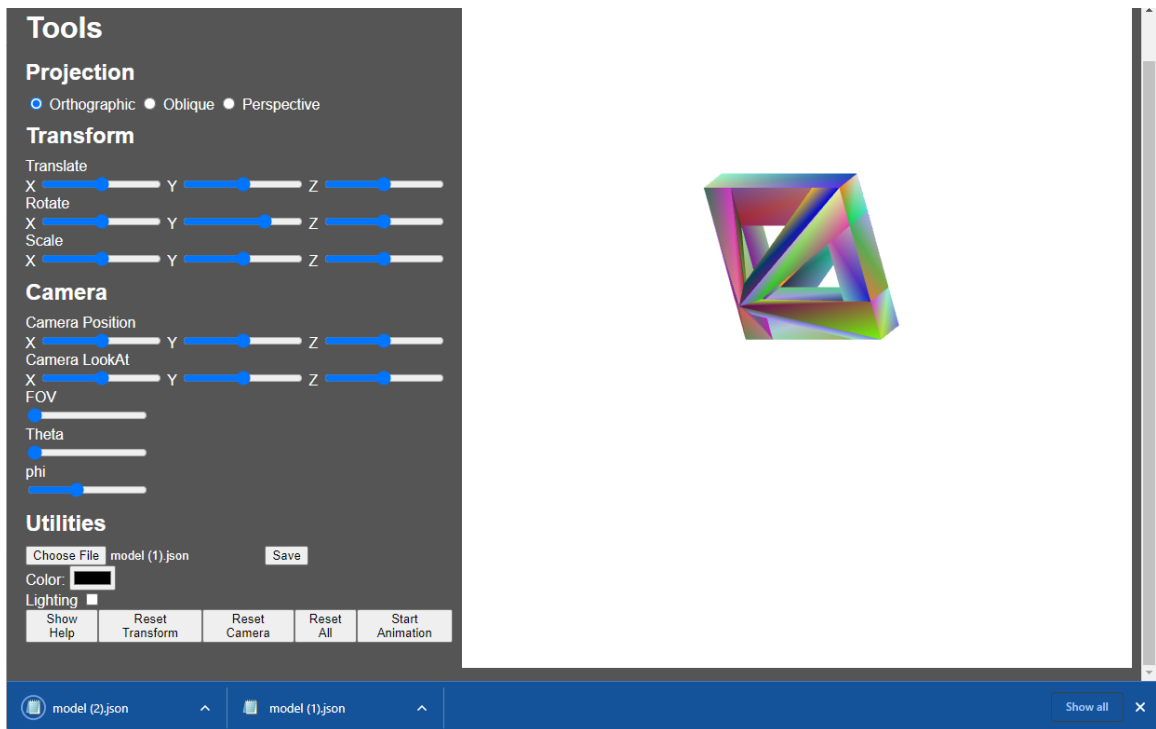
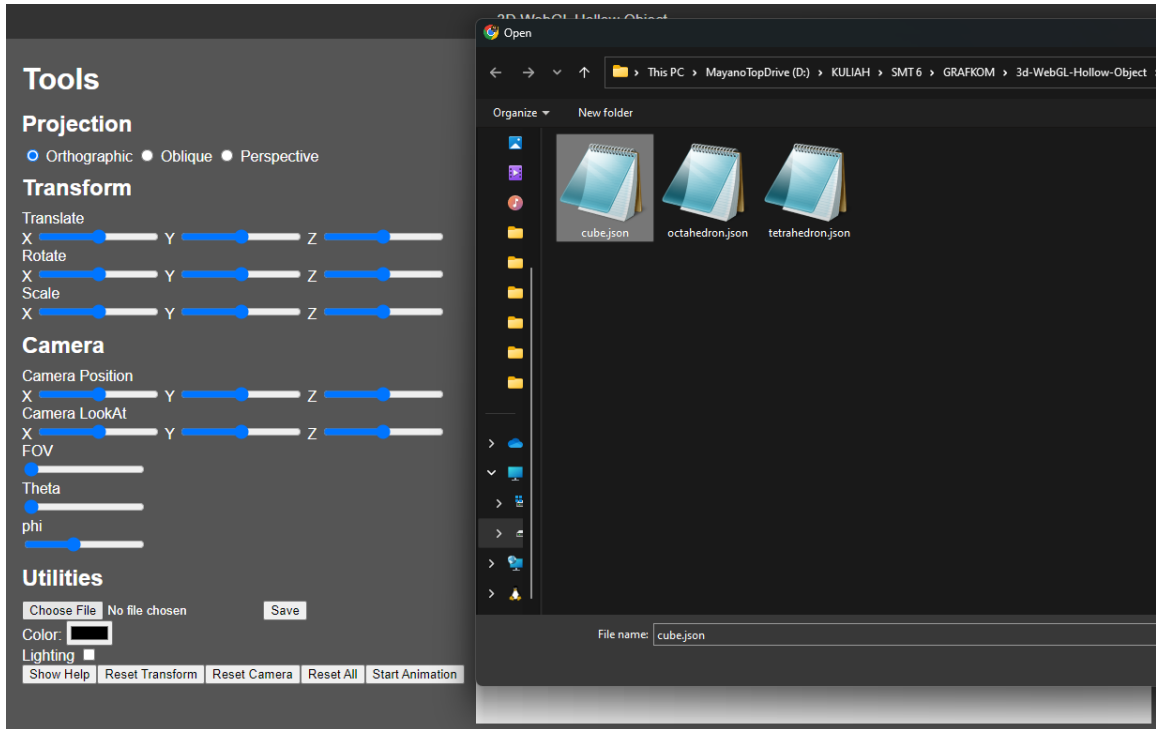
3. Transformasi Kamera (Translasi, Rotasi, FOV, theta, phi)

Selain model 3D, program juga dapat melakukan transformasi pada kamera. Ini memungkinkan kamera untuk dapat bergerak mengitari model-model.



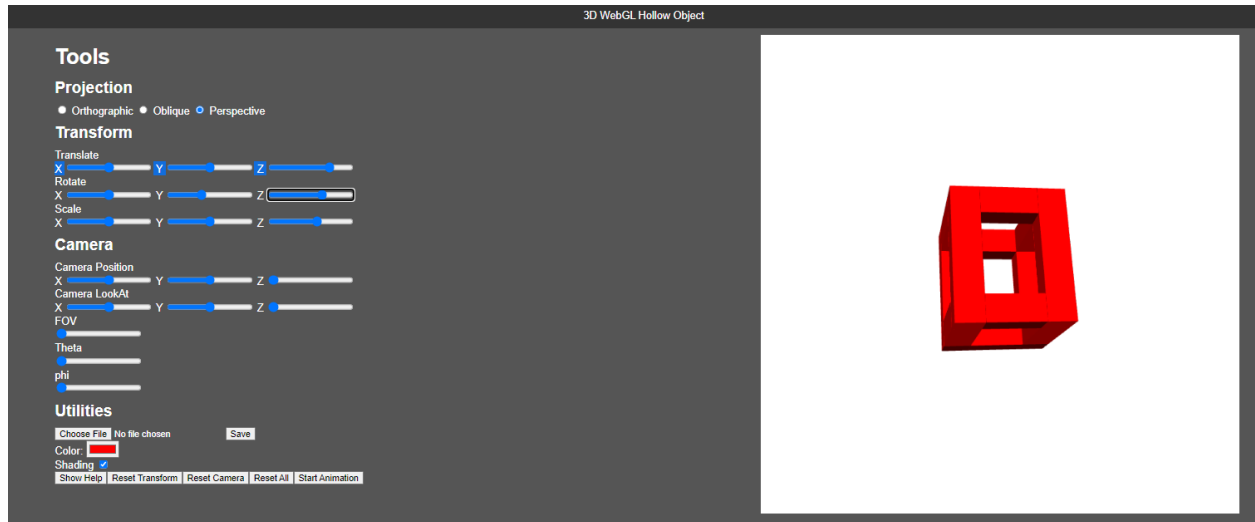
4. Menyimpan dan Memuat Objek (Save and Load)

Seperti yang dijelaskan sebelumnya, model yang dimuat dan disimpan berupa file .json berisi javascript object yang menyimpan nilai *vertices*, *faces*, *colors*, dan *normals*.



5. Pewarnaan Dasar Objek (Teknik Shading)

Shading dibuat dengan menggunakan directional lighting (Lambertian surface). Setiap vertex akan memiliki nilai normalnya masing-masing. Vektor cahaya yang berasal dari (0.5, 0.7, 1) akan mengenai objek dan memberikan kesan warna terang gelap berdasarkan vektor normal dari vertex yang menghadap cahaya tersebut.



6. Reset Default View

Program memiliki 3 fitur reset, Reset Transform, Reset Camera, dan Reset All. Ketiga fitur digunakan untuk mengembalikan default configuration dari object.

Tools

Projection

☐ Orthographic ☒ Oblique ☐ Perspective

Transform

Translate

X Y Z

Rotate

X Y Z

Scale

X Y Z

Camera

Camera Position

X Y Z

Camera LookAt

X Y Z

FOV

Theta

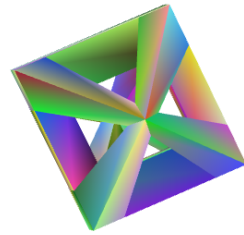
phi

Utilities

model (1).json

Color:

Lighting ☐



Tools

Projection

☐ Orthographic ☒ Oblique ☐ Perspective

Transform

Translate

X Y Z

Rotate

X Y Z

Scale

X Y Z

Camera

Camera Position

X Y Z

Camera LookAt

X Y Z

FOV

Theta

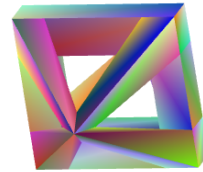
phi

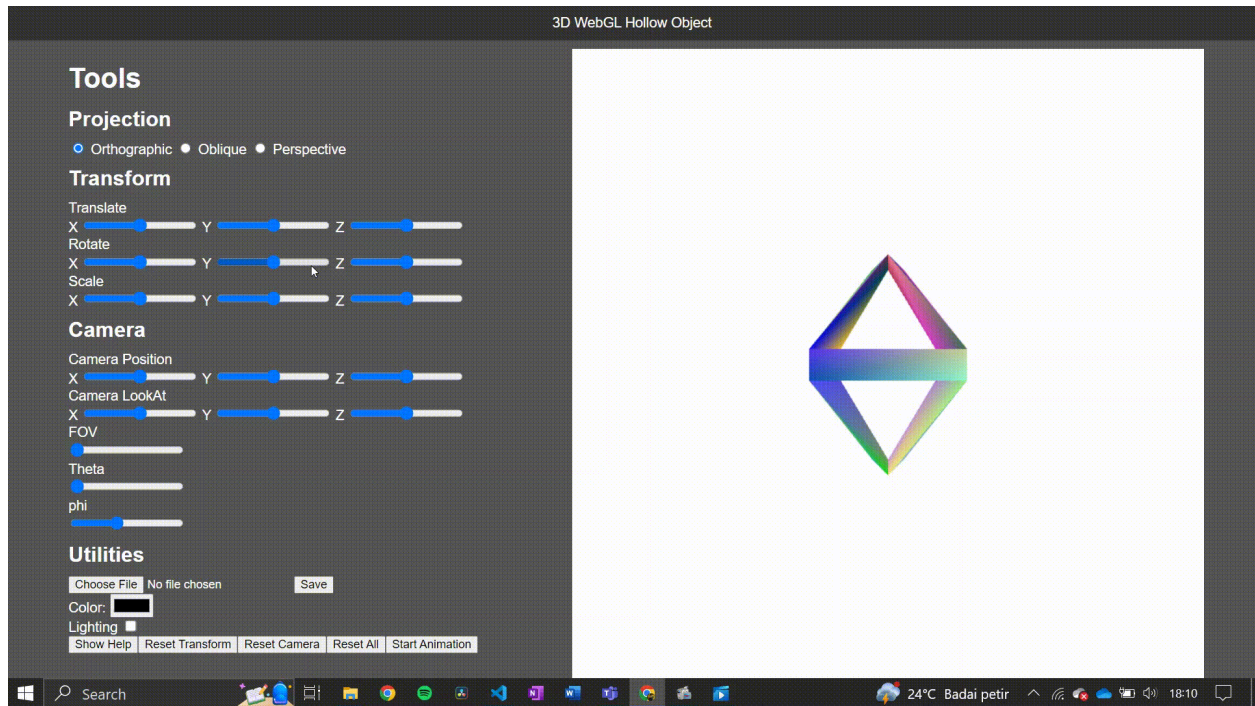
Utilities

model (1).json

Color:

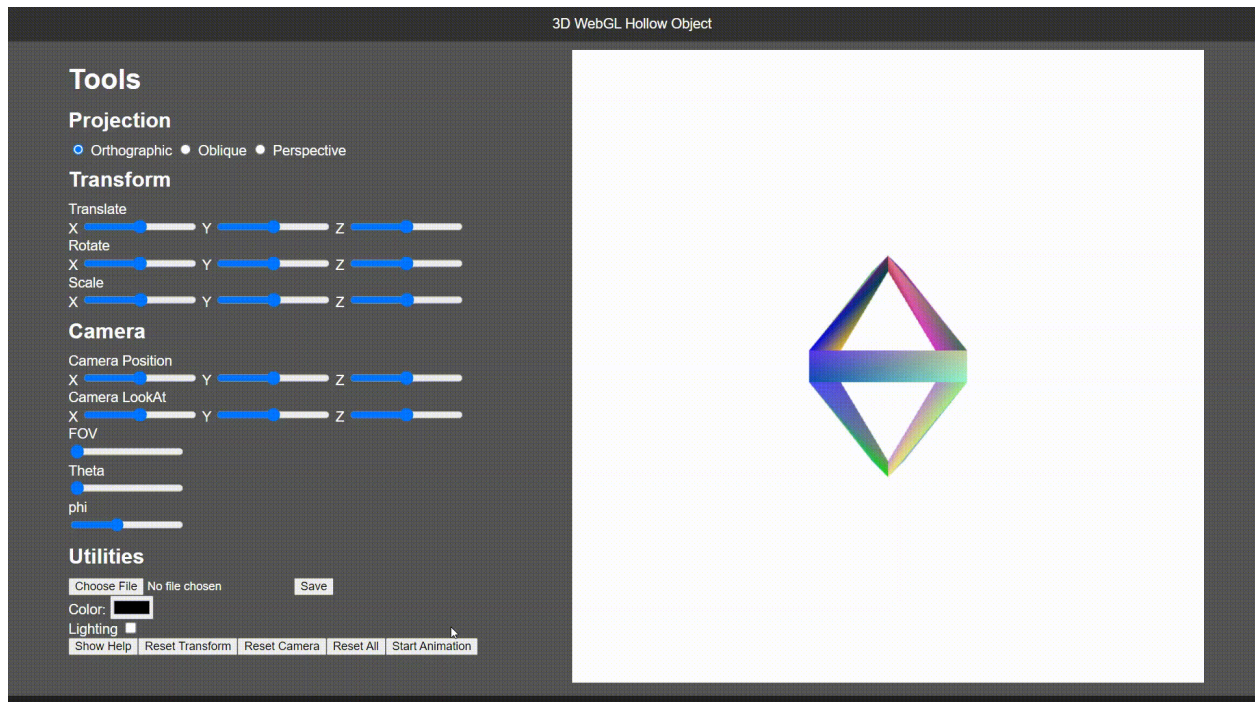
Lighting ☐





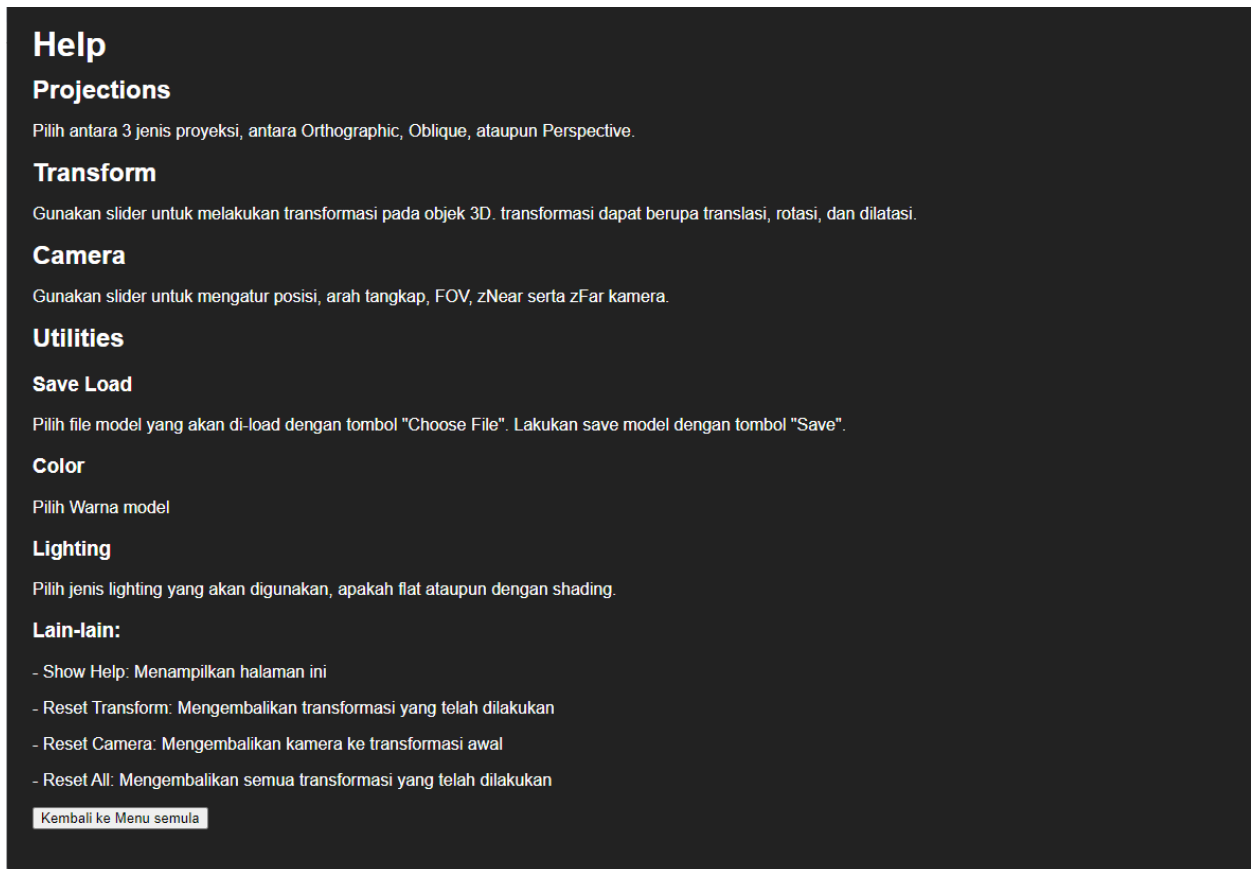
7. Animasi Model

Spesifikasi lanjutan yang diimplementasikan adalah animasi untuk setiap model menggunakan suatu tombol. Ketika tombol tersebut ditekan, model akan melakukan animasi rotasi sumbu-X. Animasi diimplementasikan pada program saat rendering object.



8. Halaman Help

Halaman Help menampilkan cara penggunaan program.



4. Pembagian Tugas

NIM	Nama	Pembagian Kerja
13520003	Dzaky Fattan Rizqullah	Model Tetrahedron , load model, save model dengan transformasi, proyeksi perspective, translasi model, translasi dan rotasi kamera
13520029	Muhammad Garebaldhie Er Rahman	Model Cube , shading, proyeksi orthographic, rotasi model, load model, save model, translasi dan rotasi kamera, proyeksi oblique
13520031	Taufan Fajarama Putrawansyah R	Model Octahedron , setting parameter default, proyeksi oblique, scale model, animasi model