

Tugas 2 IF3260 Grafika Komputer
3D WebGL Hollow Object



Disusun oleh:

Alifia Rahmah	(13520122)
Azmi Alfatih Shalahuddin	(13520158)
Frederik Imanuel Louis	(13520163)

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2022

Daftar Isi

Daftar Isi	2
Bab I Deskripsi Program	3
Bab II Spesifikasi Fitur	4
2.1. Spesifikasi Wajib	4
2.1.1. Hollow Model	4
2.1.2. Proyeksi	5
2.1.3. Transformasi	5
2.1.4. Kamera	5
2.1.5. Shading	5
2.1.6. Reset	5
2.1.7. Load Model	5
2.2. Spesifikasi Bonus	6
2.2.1. Implementasi Shader Manual	6
2.2.2. Modifikasi Sumber Cahaya	6
Bab III Hasil	7
3.1. Proyeksi Orthographic	7
3.2. Proyeksi Perspective	7
3.3. Proyeksi Oblique	8
3.4. Rotasi	8
3.5. Translasi	9
3.6. Scaling	9
3.7. Radius Kamera	10
3.8. Rotasi Kamera	10
3.9. Reset	11
3.10. Load Model	11
3.11. Help	12
Bab IV Manual	13
4.1. Menjalankan Program	13
4.2. Membuka Model	13
4.3. Lighting & Shading	14
4.4. Projection	15
4.5. Camera	15
4.6. Transformation	16
4.7. Reset to Default	16
4.8. Help	17

Bab I

Deskripsi Program

Program ini merupakan aplikasi berbasis website *3D hollow object viewer* yang dapat menampilkan beberapa model objek 3 dimensi. Program ini dibuat untuk memenuhi Tugas Besar 2 mata kuliah IF3260 Grafika Komputer.

Dalam menampilkan model, program juga dapat menambahkan *lighting*, *shading*, menampilkan model dengan beberapa jenis proyeksi (*orthographic*, *perspective*, *oblique*), pengubahan radius dan rotasi kamera, serta transformasi objek yang terdiri dari translasi, rotasi, dan *scaling*.

Selain model yang tersedia di dalam website, model juga dapat di-*load* oleh pengguna dari *file* eksternal bertipe JSON dengan format:

```
{
  "points" : [
    ...
  ],
  "colors" : [
    ...
  ]
}
```

Format tersebut akan langsung digambar menggunakan mode `gl.Triangles`, dimana tiap tiga titik yang berurutan mulai dari titik pertama akan digambar sebagai satu bidang. Perhitungan normal dari bidang juga akan dilakukan berdasarkan urutan masukan titik, dimana tiap kelompok tiga titik dianggap sudah diorientasikan ke luar (menggunakan *right hand rule*).

Bab II

Spesifikasi Fitur

2.1. Spesifikasi Wajib

2.1.1. Hollow Model

Terdapat tiga hollow model yang dibuat, yaitu:

- Ring
- Cube
- Diamond

Ketiga model dapat dimanipulasi secara langsung dalam folder model yang akan otomatis di-load ke program, atau melalui json yang tersedia, yang kemudian dapat di-load secara manual ke program. Khusus untuk model Ring, terdapat file Python pada **src/models/ring.py** yang dapat digunakan untuk men-generate model dengan mudah. Terdapat beberapa parameter yang mudah di-edit, seperti banyak sisi dari ring, ketinggian ring, dan sebagainya. Script Python tersebut akan secara otomatis men-generate prisma hollow dengan banyak sisi yang didefinisikan, dengan ketinggian yang diberikan, berpusat pada titik (0, 0, 0). Warna yang digunakan akan di-randomize untuk tiap generation. Model ring yang digunakan sebenarnya adalah prisma hollow dengan 360 sisi.

Model didefinisikan oleh array titik dan warnanya, dimana array titik merupakan array yang bisa langsung dimasukkan ke dalam buffer WebGL dan digambar menggunakan mode Triangles. Array colors memiliki ukuran yang sama dengan array titik, dimana tiap elemennya menandakan warna dari masing-masing titik yang pada array titik sebelumnya.

Ketiga model yang sudah didefinisikan di atas dapat langsung dipilih dalam input model dalam aplikasi. Model juga dapat dibuka melalui file JSON yang telah didefinisikan pada folder , dimana file JSON untuk ketiga model tersebut juga sudah didefinisikan, dengan format:

```
{
  "points" : [
    ...
  ],
  "colors" : [
    ...
  ]
}
```

Format tersebut akan langsung digambar menggunakan mode `gl.Triangles`, dimana tiap tiga titik yang berurutan mulai dari titik pertama akan digambar sebagai satu bidang. Perhitungan normal dari bidang juga akan dilakukan berdasarkan urutan masukan titik, dimana tiap kelompok tiga titik dianggap sudah diorientasikan ke luar (menggunakan right hand rule).

2.1.2. Proyeksi

Terdapat tiga proyeksi yang diimplementasikan dalam tugas ini, yaitu orthographic, perspective, dan oblique. Proyeksi diimplementasikan dengan mengalikan dengan matriks yang bersesuaian baik untuk proyeksi orthographic, perspective, maupun oblique. Proyeksi dapat dilakukan dengan memilih salah satu dari Projection orthographic, perspective, maupun oblique pada toolbar.

2.1.3. Transformasi

Pada setiap model, dapat dilakukan transformasi seperti translasi, rotasi, dan scaling. Translasi dan rotasi dapat dilakukan berdasarkan sumbu X, Y, dan Z. Transformasi dilakukan dengan mengalikan matriks yang bersesuaian baik untuk operasi translasi, rotasi, maupun scaling.

2.1.4. Kamera

Kamera dapat diubah posisinya berdasarkan tiga parameter, yaitu radius terhadap titik origin, rotasi sumbu y, dan rotasi sumbu x. Ketiga parameter tersebut dapat dimanipulasi menggunakan slider yang tersedia pada toolbar, dimana perubahan parameter akan menghasilkan matriks view yang berbeda hingga menghasilkan kesan kamera menjauh, mendekat, dan berputar terhadap objek.

2.1.5. Shading

Shading diimplementasikan pada shader program dan dapat dimatikan atau dinyalakan dari toolbar pengguna. Jika shader dimatikan, maka warna yang akan ditampilkan langsung diambil dari warna objek. Implementasi shading dilakukan secara manual sesuai dengan spesifikasi 2.2.1, dimana warna dan posisi cahaya juga dapat diubah sesuai dengan spesifikasi 2.2.2.

2.1.6. Reset

Tombol reset mengubah konfigurasi view seperti lighting, translasi, camera, rotasi, serta scaling yang telah diubah-ubah oleh user kembali ke konfigurasi awal.

2.1.7. Load Model

Selain model yang terdapat pada pilihan daftar model, pengguna juga dapat melakukan load model dari file JSON dengan format:

```
{
  "points" : [
    ...
  ],
  "colors" : [
    ...
  ]
}
```

2.2. Spesifikasi Bonus

2.2.1. Implementasi Shader Manual

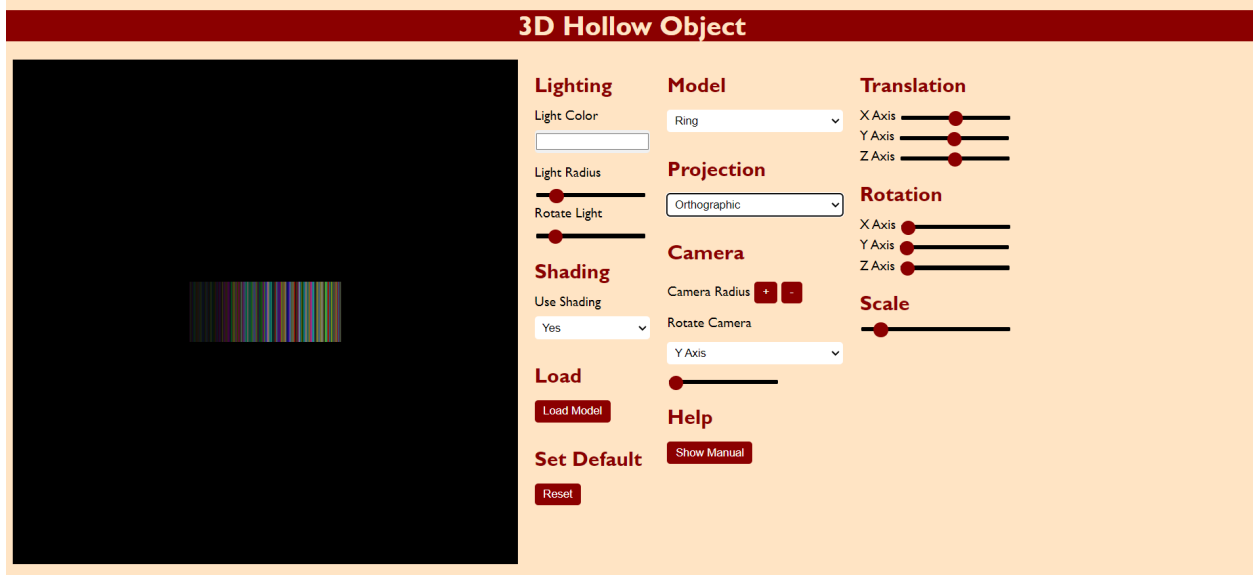
Shader menggunakan ambient lighting dan diffuse lighting, dimana intensitas cahayanya juga diskalakan berdasarkan jarak cahaya ke objek. Shading ini diimplementasikan pada fragment shader dari program. Shading dilakukan pada world view, dimana koordinat objek yang di-pass ke fragment shader dari vertex shader adalah koordinat dunia, sebelum ditransformasi oleh view matrix. Oleh karena itu, posisi kamera juga diperlukan sebagai masukan pada fragment shader. Terakhir, fragment shader juga membutuhkan masukan normal objek, posisi sumber cahaya, dan warna cahaya. Normal sendiri dihitung pada program javascript, dimana masing-masing triangle dari model dihitung normalnya menggunakan cross product. Posisi dan warna sumber cahaya dapat diubah-ubah, seperti pada spesifikasi bagian 2.2.2.

2.2.2. Modifikasi Sumber Cahaya

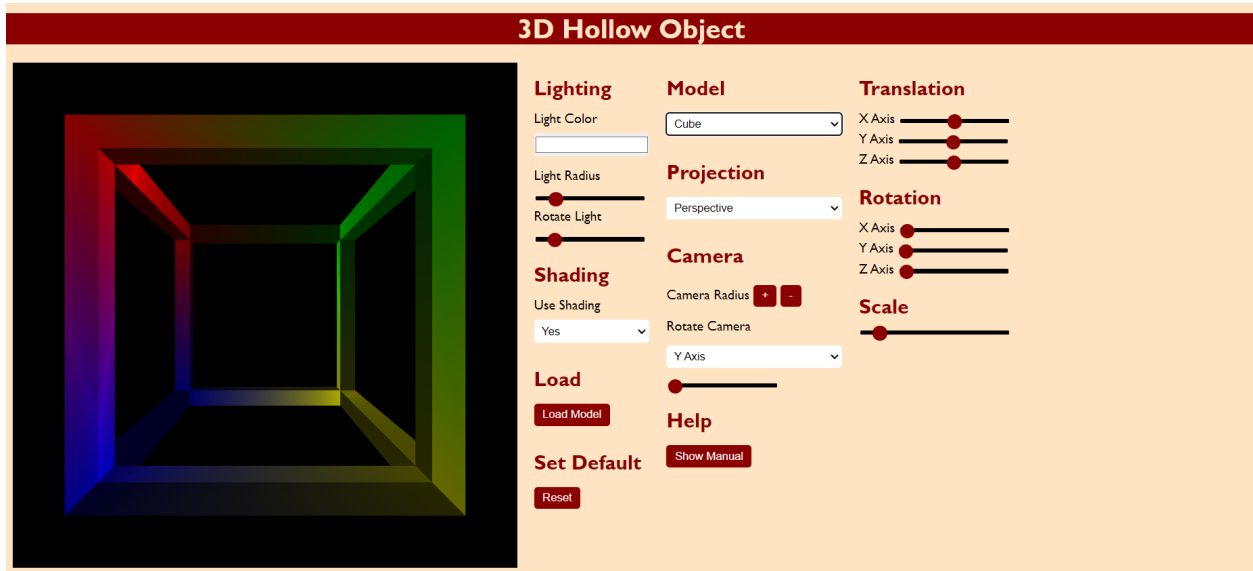
Sumber cahaya dapat diubah posisinya menggunakan dua parameter, yaitu perubahan radius cahaya (terhadap titik origin), dan perubahan rotasi cahaya terhadap sumbu y. Sumber cahaya juga dapat diubah warnanya menggunakan color picker yang tersedia pada toolbar. Pemrosesan efek perubahan sumber cahaya tersebut akan dilakukan pada fragment shader, sesuai dengan spesifikasi pada bagian 2.2.1.

Bab III Hasil

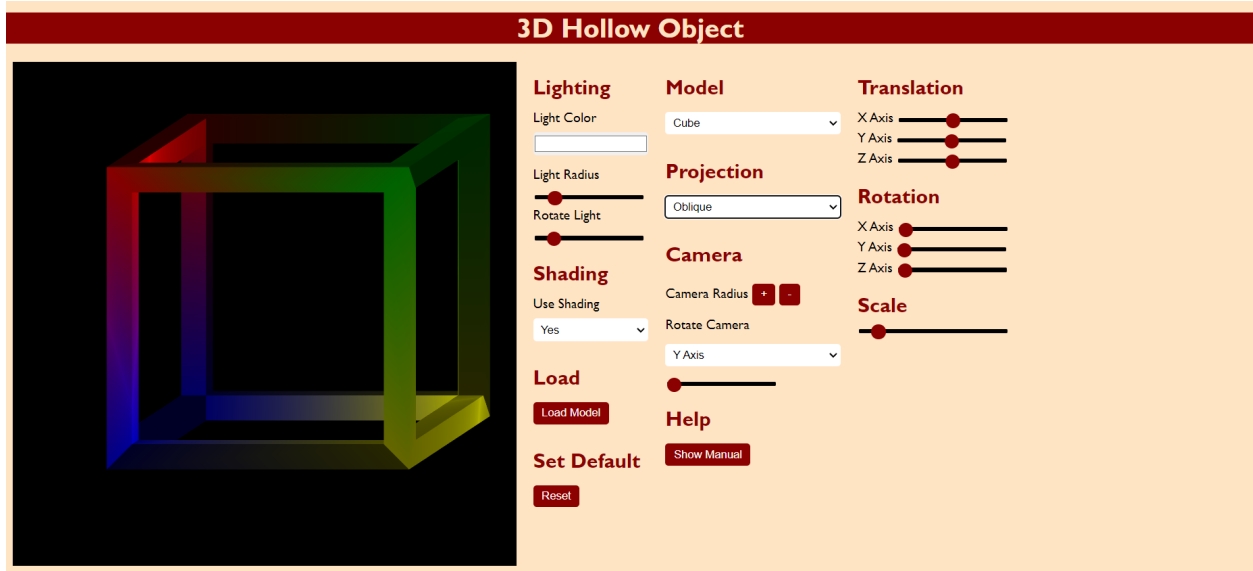
3.1. Proyeksi Orthographic



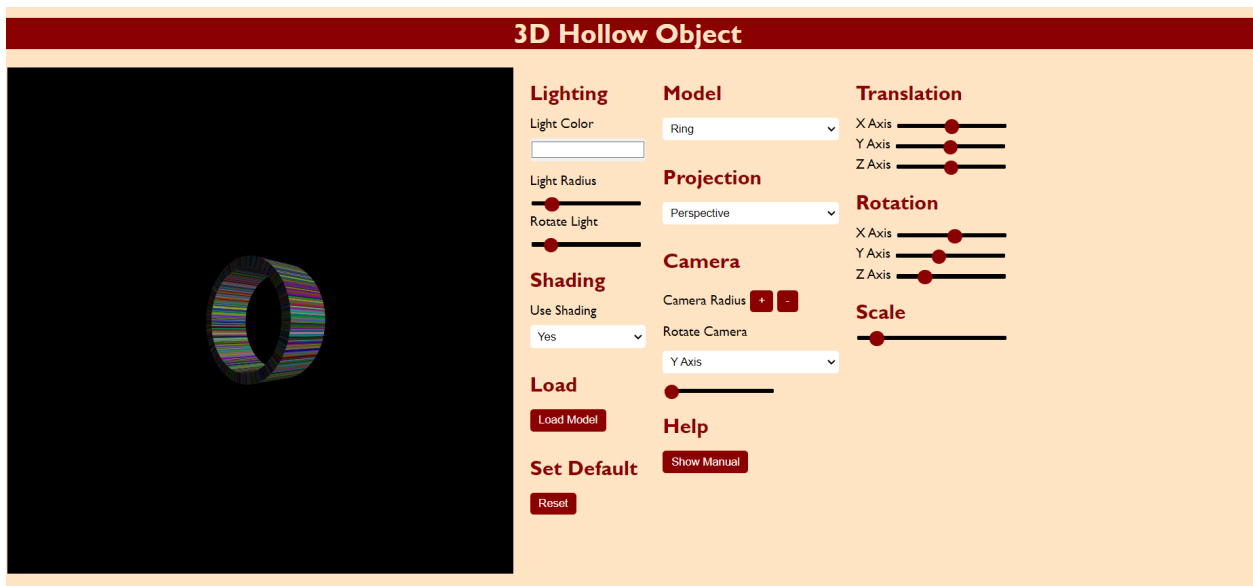
3.2. Proyeksi Perspective



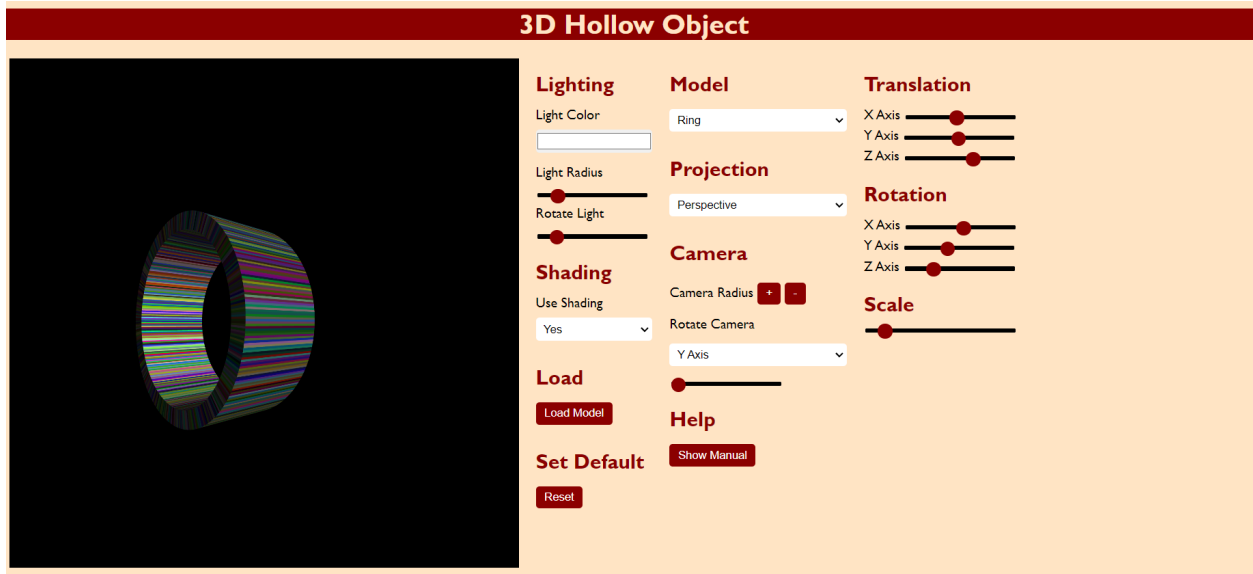
3.3. Proyeksi Oblique



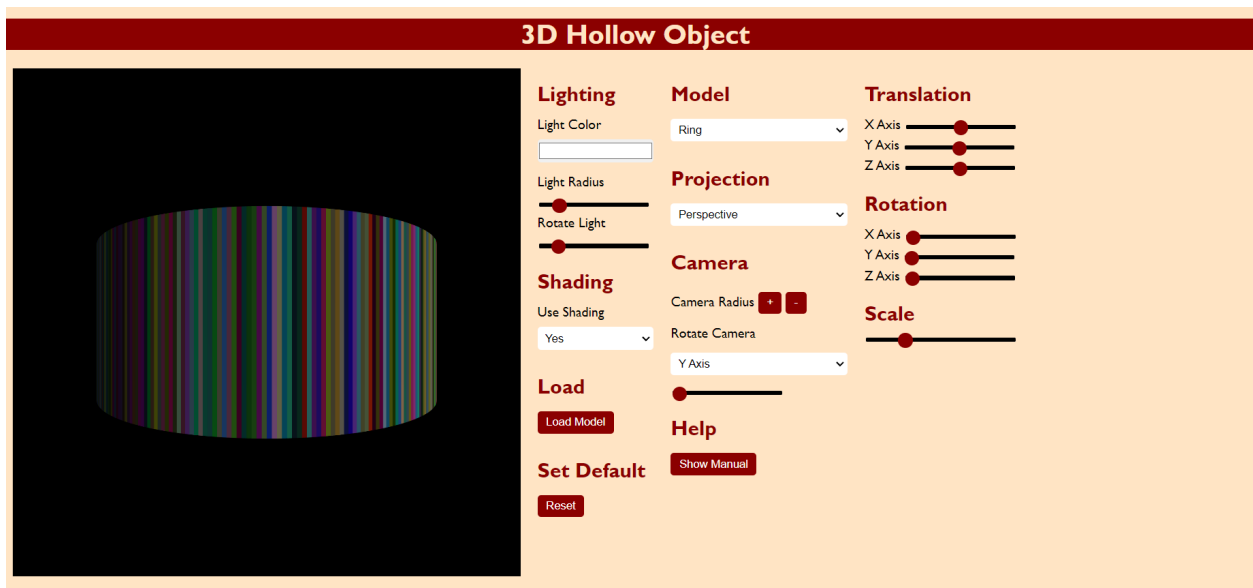
3.4. Rotasi



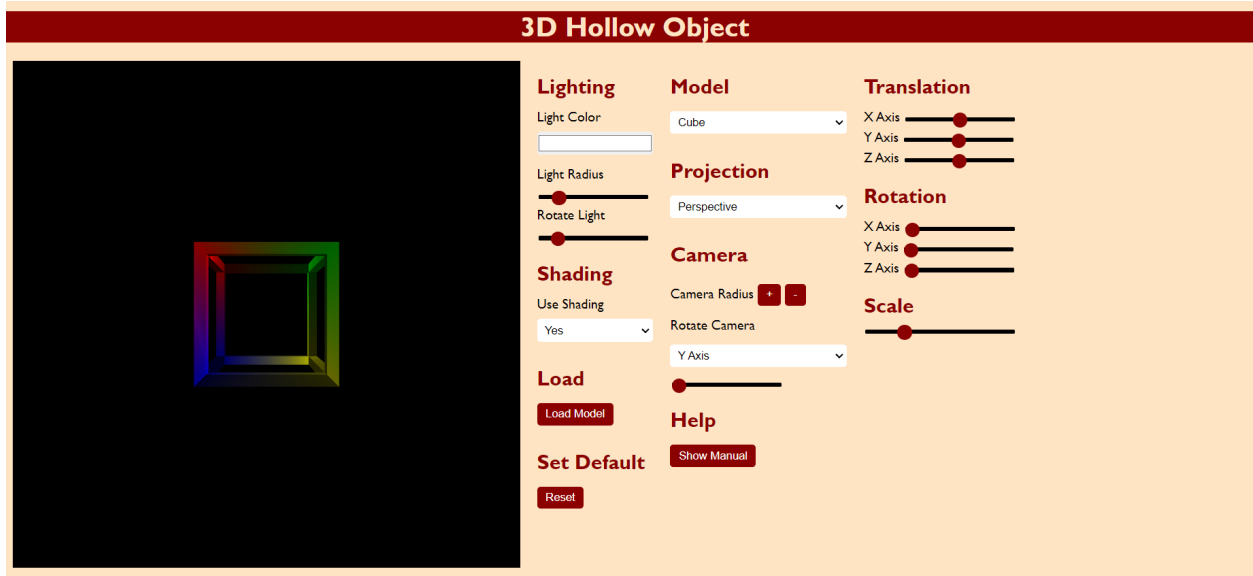
3.5. Translasi



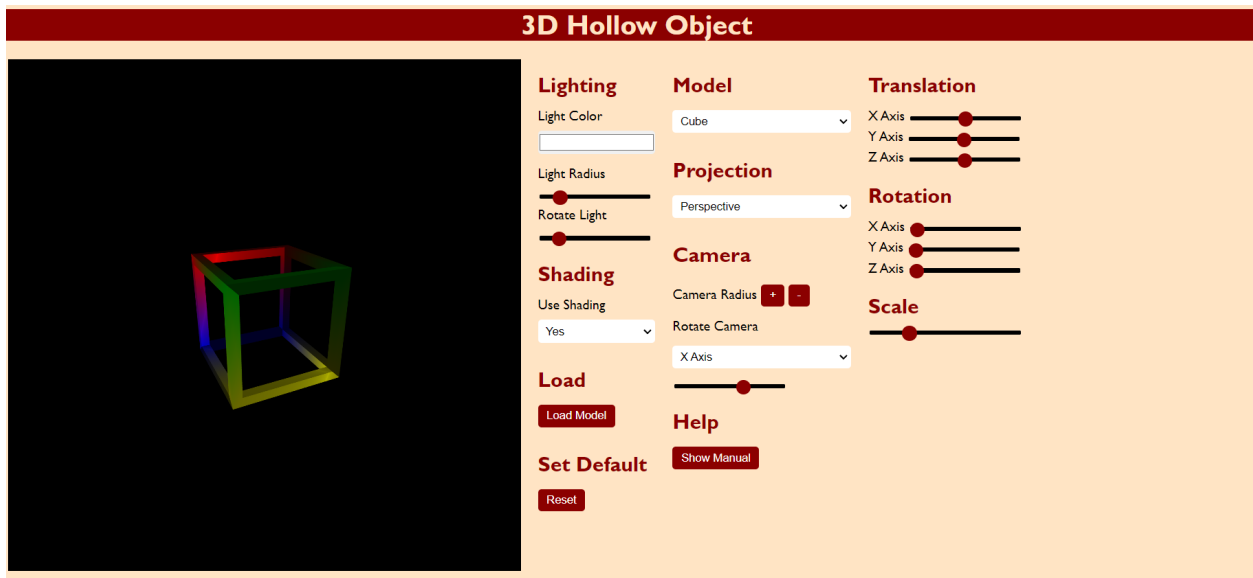
3.6. Scaling



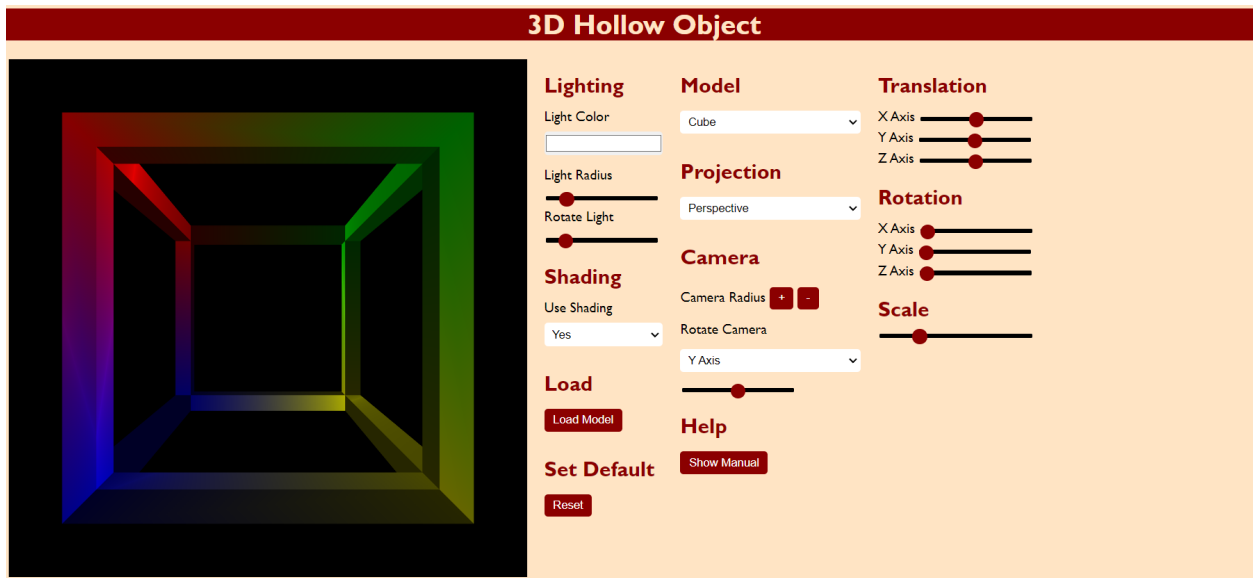
3.7. Radius Kamera



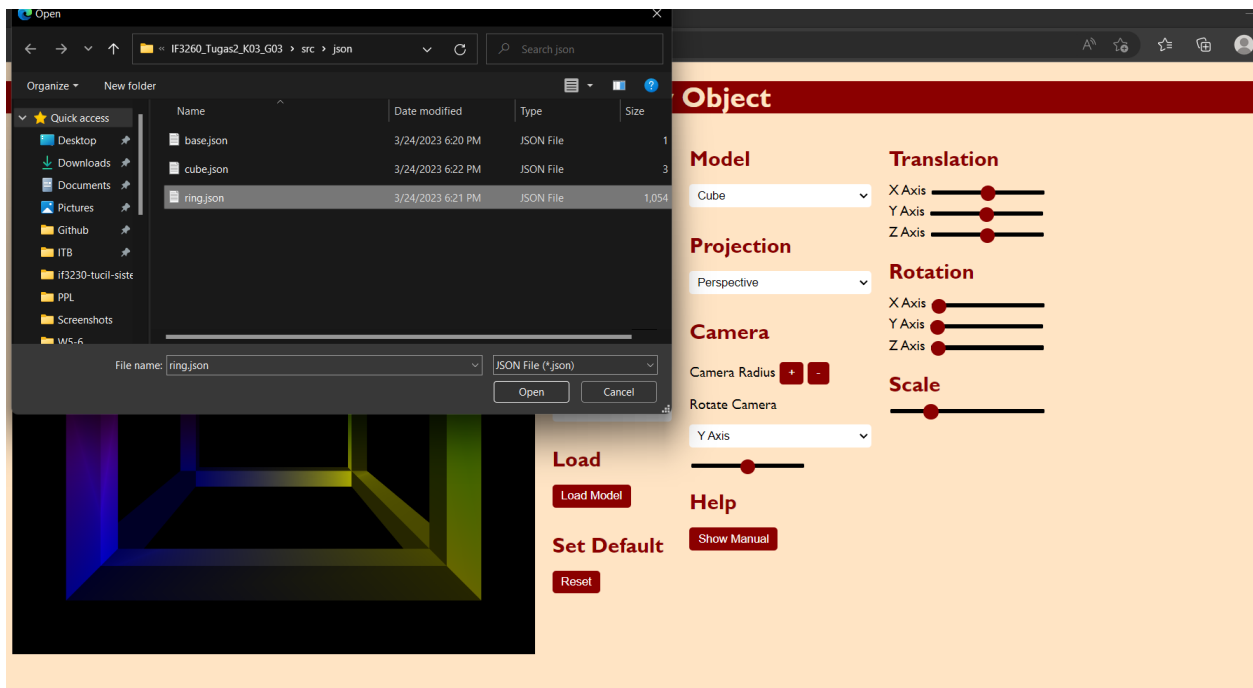
3.8. Rotasi Kamera

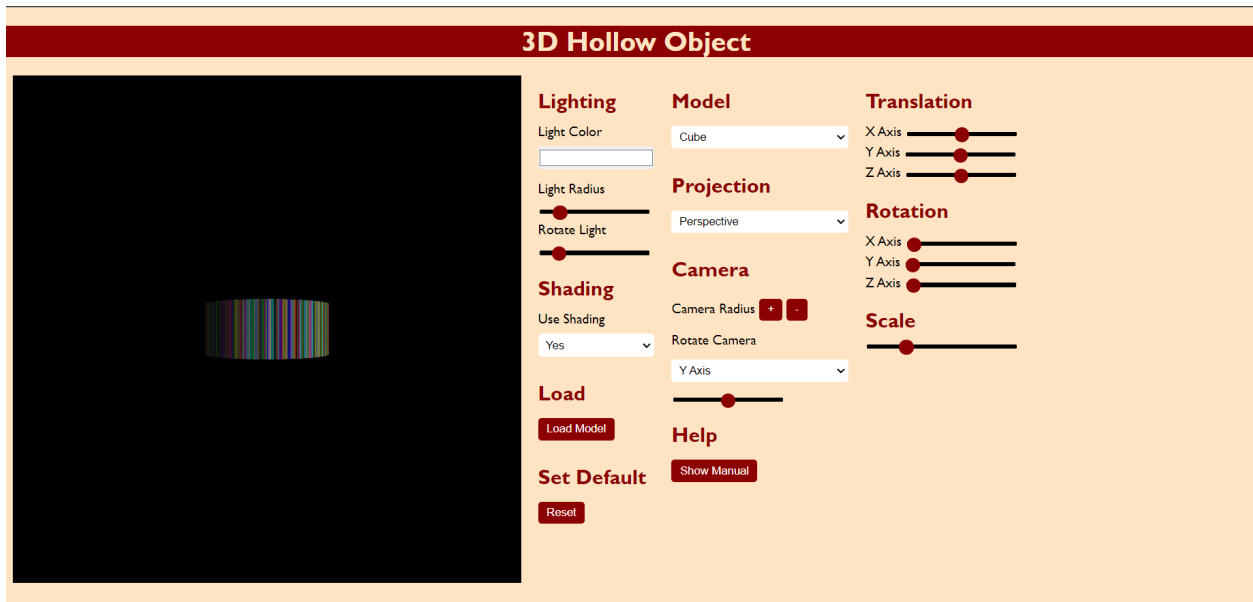


3.9. Reset

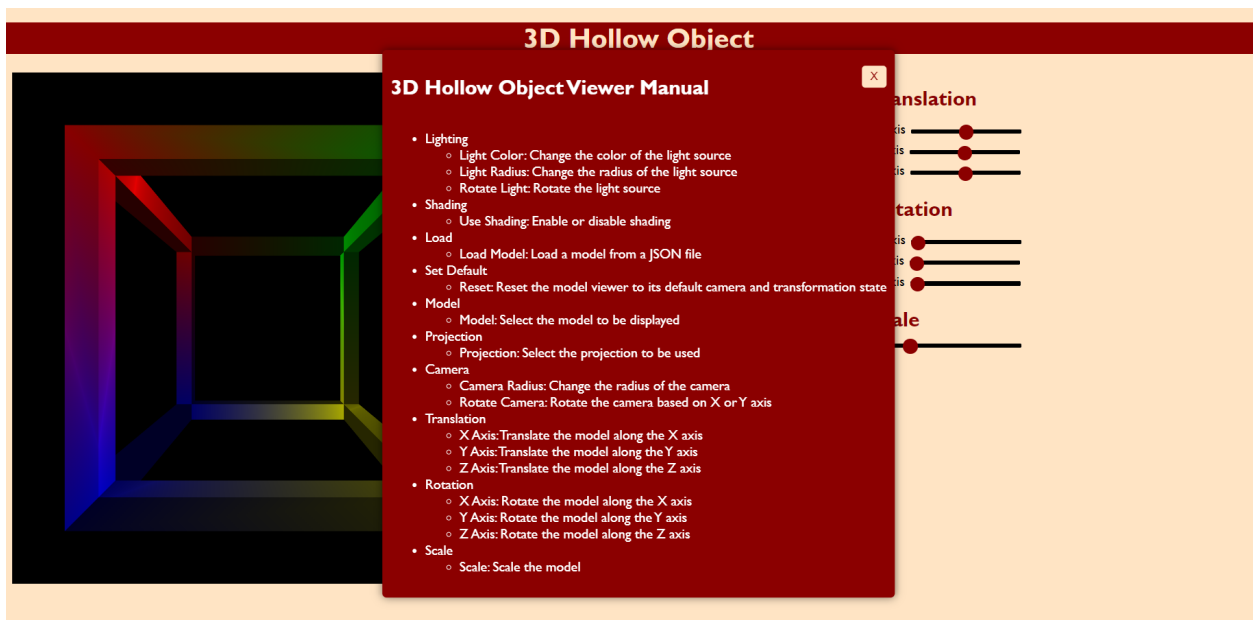


3.10. Load Model





3.11. Help



Bab IV

Manual

4.1. Menjalankan Program

Program dapat dijalankan langsung dengan membuka file `src/index.html` pada source code. Jika ingin menjalankan di server, dapat dilakukan dengan beberapa cara, salah satunya dengan menggunakan package `serve` dalam Node.js.

Untuk melakukan instalasi, pastikan sudah terdapat Node.js dan dapat menjalankan perintah `npm` terlebih dahulu. Kemudian lakukan instalasi dengan perintah:

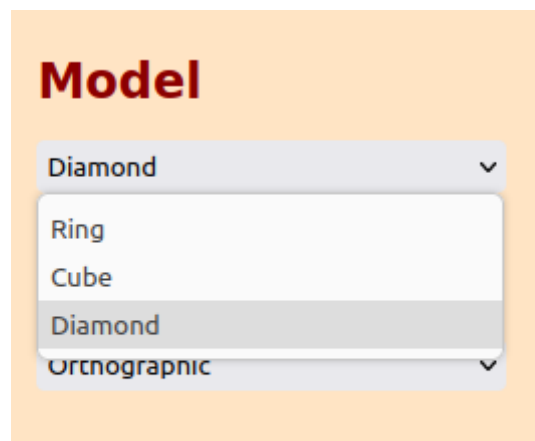
```
npm install -g serve
```

Untuk menggunakan `serve`, dapat dilakukan dari direktori root source code dengan perintah:

```
serve src
```

4.2. Membuka Model

Pengguna dapat menampilkan model *hollow object* yang sudah tersedia dalam aplikasi dengan memilih antara beberapa model yang ada pada panel Model.



Pilihan model

Selain itu, pengguna juga dapat me-load model dari file JSON dengan men-upload file tersebut dengan tombol “Load Model”.



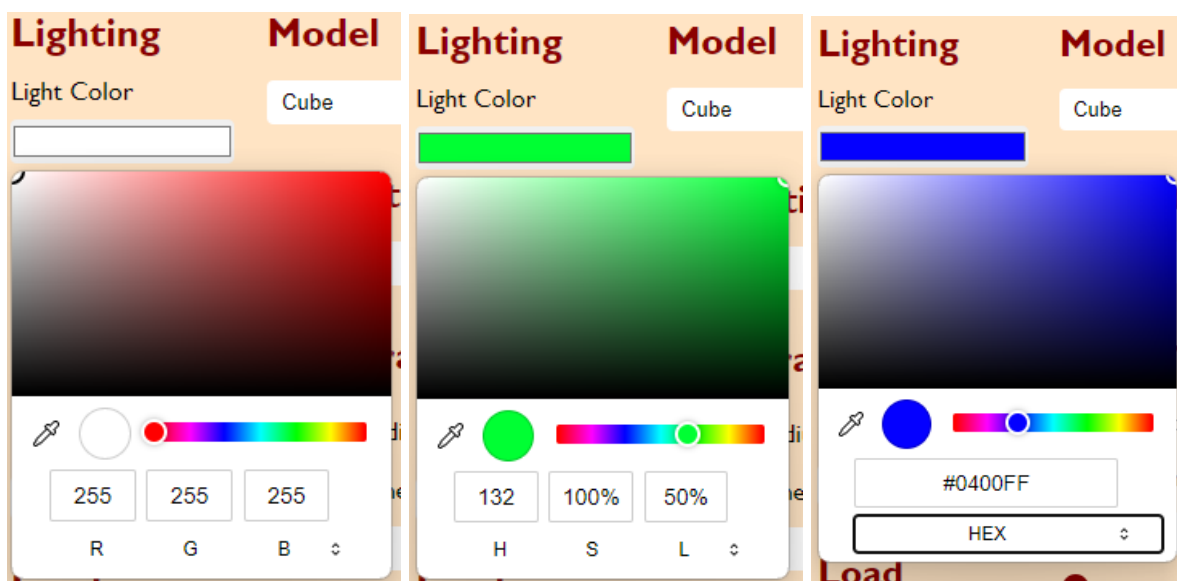
Tombol untuk load model

4.3. Lighting & Shading

Pengguna dapat menentukan warna lighting, radius, serta melakukan rotasi pada sumber cahaya. Warna lighting dapat dipilih dengan menekan tombol Light Color (secara default berwarna putih). Setelah diklik, warna dapat dipilih baik dari klik sebuah warna langsung dari gradasi warna maupun dengan memasukkan kadar RGB pada input angka di bawahnya. Tombol RGB dapat diubah untuk melakukan input warna dari representasi HSL maupun HEX.

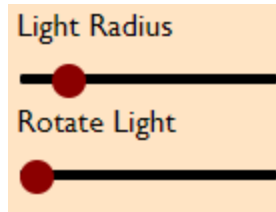


Tombol untuk mengubah warna



Pilihan warna

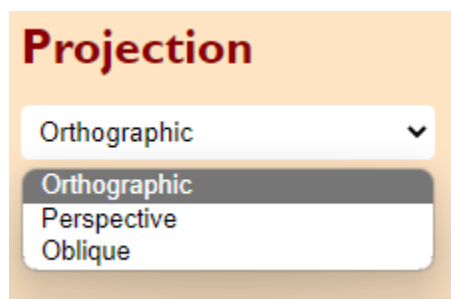
Radius dan rotasi sumber cahaya dapat dilakukan dengan menggeser slider yang bersesuaian.



Slider *Light Radius* dan *Rotate Light*

4.4. Projection

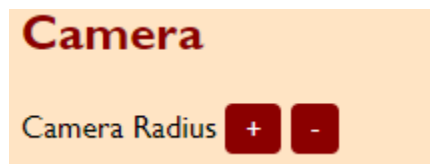
Jenis proyeksi pada viewer dapat diubah menjadi salah satu dari orthographic, perspective, dan oblique dengan memilih salah satu jenis proyeksi pada panel Projection.



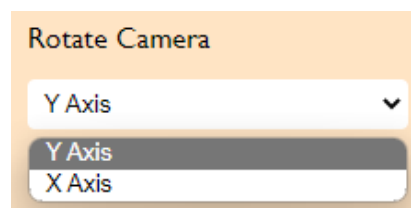
Pilihan jenis proyeksi

4.5. Camera

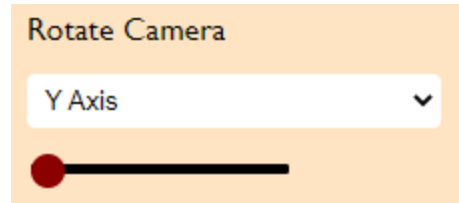
Pengguna dapat mengubah radius dari kamera dari panel Camera dengan menekan tombol tambah (“+”) dan tombol kurang (“-”) untuk mengubah radius sehingga kamera dapat mendekat mapuun menjauh dari model.



Tombol untuk mengubah radius



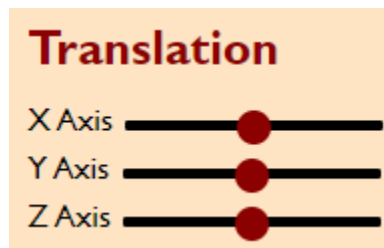
Input untuk mengganti sumbu



Slider untuk melakukan rotasi sesuai sumbu yang dipilih

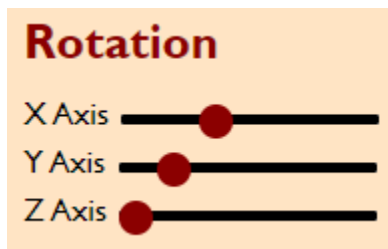
4.6. Transformation

Pada objek, dapat dilakukan transformasi berupa translasi, rotasi, dan scaling. Translasi terhadap sumbu X, sumbu Y, maupun sumbu Z dilakukan dengan menggeser slider dengan label sumbu yang bersesuaian pada panel Translation.



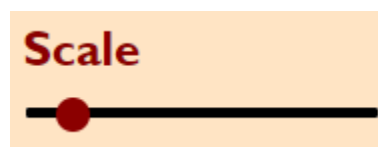
Slider untuk translasi terhadap sumbu X, sumbu Y, dan sumbu Z

Translasi terhadap sumbu X, sumbu Y, maupun sumbu Z dilakukan dengan menggeser slider dengan label sumbu yang bersesuaian pada panel Rotation.



Slider untuk rotasi terhadap sumbu X, sumbu Y, dan sumbu Z

Scaling dilakukan dengan menggeser slider pada panel Scale.



Slider untuk scaling

4.7. Reset to Default

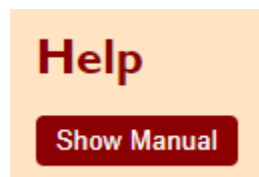
Konfigurasi lighting, shading, kamera, serta transformasi yang telah dilakukan terhadap suatu objek dapat dikembalikan menjadi konfigurasi semula dengan menekan tombol “Reset” pada panel Set Default.



Tombol untuk me-reset konfigurasi

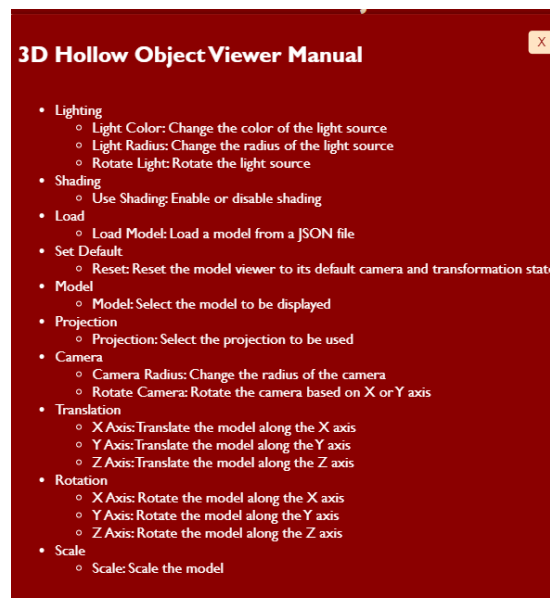
4.8. Help

Untuk melihat panduan singkat dalam aplikasi, dapat dilakukan dengan menekan tombol “Show Manual” pada panel Help.



Tombol “Show Manual”

Panduan akan ditampilkan dalam bentuk pop-up dan dapat ditutup kembali dengan menekan tombol X di pojok kanan atas dari pop-up tersebut.



Pop-up manual