

## **TUGAS 2**

### **3D WEBGL HOLLOW OBJECT**

Diajukan untuk memenuhi nilai tugas 2  
Mata Kuliah IF3260 Grafika Komputer

Dosen Pengampu: Dr. Judhi Santoso, M.Sc.



Dibuat Oleh:

Ilham Prasetyo Wibowo	13520013
Maharani Ayu Putri Irawan	13520019
Yohana Golkaria Nainggolan	13520053

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**2023**

# **BAB I**

## **PENDAHULUAN**

Dalam tugas ini, dibuat sebuah aplikasi berbasis web untuk melakukan manipulasi pada *hollow object* yang juga dibuat. Kode dibuat dalam bahasa Javascript dan menggunakan API WebGL. WebGL merupakan API Javascript untuk *me-render* grafika 2 dan 3 dimensi menggunakan browser. Kami menggunakan browser Google Chrome dengan mengaktifkan WebGL developer extension yang telah tersedia.

Kode sumber yang telah dibuat, dapat diakses melalui [https://github.com/ilhamwibowo/IF3260\\_Tugas2\\_K01\\_G08](https://github.com/ilhamwibowo/IF3260_Tugas2_K01_G08) . Untuk menjalankan program, diperlukan perangkat dengan browser yang mendukung WebGL developer extension. *Clone repository* pada direktori lokal, lalu buka jalankan *file* index.html dengan LiveServer, yang dapat diunduh sebagai *extension* VS Code.

## BAB II

### DESKRIPSI DAN FITUR

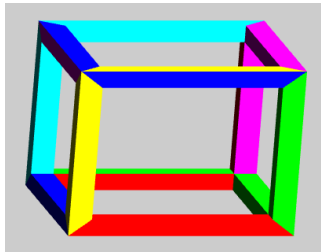
Kode sumber dapat diakses pada [https://github.com/ilhamwibowo/IF3260\\_Tugas2\\_K01\\_G08](https://github.com/ilhamwibowo/IF3260_Tugas2_K01_G08). Untuk menjalankan program dapat dilakukan dengan menggunakan ekstensi Live Server pada VSCode. Klik F1 lalu website akan terbuka di browser Anda. Jika tidak menggunakan VSCode, program dapat dijalankan menggunakan http-server pada folder src.

#### A. Hollow Objects

Terdapat 3 *hollow object* yang dibuat, yakni:

1. Hollow Cube

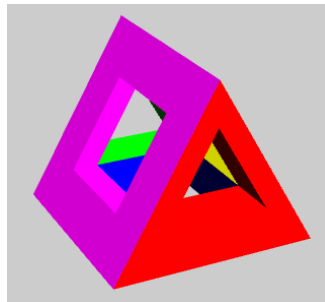
Dibuat oleh: Ilham Prasetyo Wibowo (13520013)



*Gambar 1. Hollow cube*

2. Hollow Triangular Prism

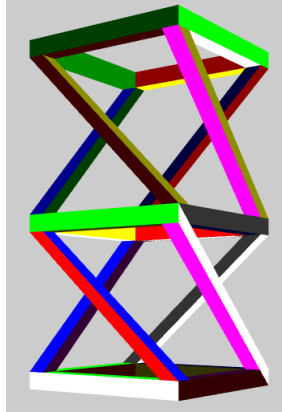
Dibuat oleh: Maharani Ayu Putri Irawan (13520019)



*Gambar 2. Hollow triangular prism*

3. Hollow Cuboid

Dibuat oleh: Yohana Golkaria Nainggolan (13520053)



Gambar 3. Hollow cuboid

## B. Input & Output

Ada 2 fungsi input dan output yang dibuat, yakni *load* dan *save*.

### 1. Load

Fungsi *load* digunakan untuk menggambar objek 3 dimensi pada *canvas* dengan *vertices*, *colors*, *normals*, dan *indices* yang didefinisikan pada file berformat JSON dengan struktur:

```
{
  "vertices": [
    ...
  ],
  "colors": [
    ...
  ],
  "indices": [
    ...
  ],
  "normals": [
    ...
  ]
}
```

Gambar 4. Struktur file masukan berformat JSON

Terdapat 4 file objek yang sudah tersedia pada subfolder `src/modules/models`, yakni *simple cube* serta ketiga *hollow object* pada poin sebelumnya. Tampilan awal ketika membuka website menunjukkan model *hollow cube*. Model-model lain dapat di-*load* dengan:

- Mengklik Choose File lalu memilih file model yang tersedia
- Mengklik Load



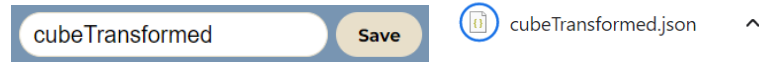
Gambar 5. Fungsionalitas Load

Ketika event klik dideteksi pada tombol load, dilakukan pembacaan file yang dibaca dari input file, lalu dibuat *Object3D* dengan *vertices*, *colors*, *normals*, dan *indices* hasil bacaan file.

## 2. Save

Untuk menyimpan Object3D pada *canvas* beserta transformasi yang telah dilakukan, dapat digunakan fungsionalitas *save*. Untuk melakukan penyimpanan, langkah yang dapat dilakukan sebagai berikut:

- Untuk menyimpan dengan nama file tertentu, masukkan nama file yang diinginkan (tanpa ekstensi *.json*) pada input teks di sebelah kiri tombol *Save*. Bila tidak dimasukkan, nama file default adalah *hollowobject.json*.
- Klik tombol *Save*. Object3D akan tersimpan di *Downloads/*



Gambar 6. Fungsionalitas *Save*

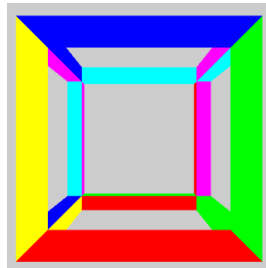
Fungsi *Save* diimplementasikan dengan terlebih dahulu mengaplikasikan matriks transformasi pada seluruh *vertices* dan *normals*. Lalu, dibuat file *JSON* dan diisi *vertices*, *colors*, *normals*, dan *indices* Object3D tersebut. Terakhir, file disimpan dengan *Blob* dan *create URL* untuk di *download*.

## C. Proyeksi

Terdapat 3 jenis proyeksi yang diimplementasikan, berikut dicobakan pada *hollow cube*.

### 1. Perspective

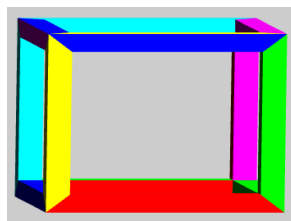
Merupakan jenis proyeksi default. Titik yang terletak jauh (menuju sumbu *z*-) terlihat mengecil. *Canvas* juga dibatasi pada  $x = \{-1, +1\}$ ,  $y = \{-1, +1\}$ , dan  $z = \{-1, +1\}$ . Objek akan di-clip saat melebihi 0.1 di depan dan 100 di belakang. Berikut merupakan hasil *perspective projection*.



Gambar 7. *Perspective projection* pada *hollow cube* dengan *scale sumbu z*

### 2. Orthographic

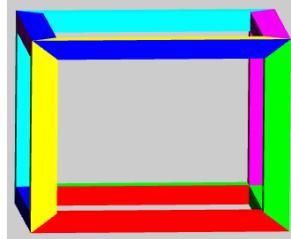
Pada proyeksi ini, sisi-sisi 3D ditampakkan sebagai sejumlah sisi-sisi 2D. Berikut merupakan hasil *orthographic projection*.



*Gambar 8. Orthographic projection hollow cube dengan shading dan rotation x-y*

### 3. Oblique

Proyeksi oblique diimplementasikan dengan melakukan proyeksi orthographic dan melakukan shear lalu sedikit ditranslasikan ke sumbu x, y, dan z. Berikut merupakan hollow cube yang sama dengan poin sebelumnya direpresentasikan dengan proyeksi oblique.



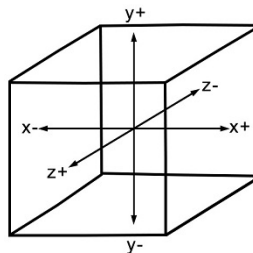
*Gambar 9. Oblique projection pada hollow cube Gambar 8*

## D. Transformasi

Ada 3 jenis transformasi yang diimplementasikan pada ketiga sumbu, x, y, dan z.

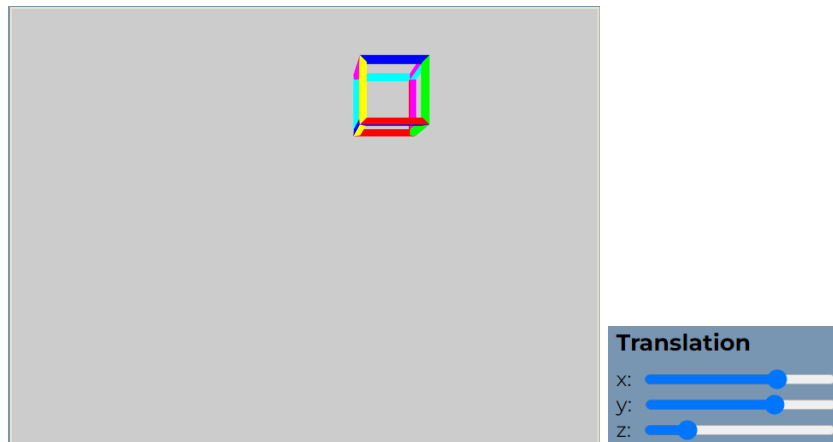
### 1. Translasi

Menggeser posisi Object3D, baik ke arah sumbu x, y, maupun z dengan slider masing-masing sumbu. Pergeseran ke kanan menandakan sumbu positif merujuk pada gambar berikut.



*Gambar 10. Sumbu acuan (Sumber: TutorialsPoint)*

Berikut contoh posisi hasil translasi ke arah sumbu x+, y+, dan z- dengan hollow cube yang di-reset default.

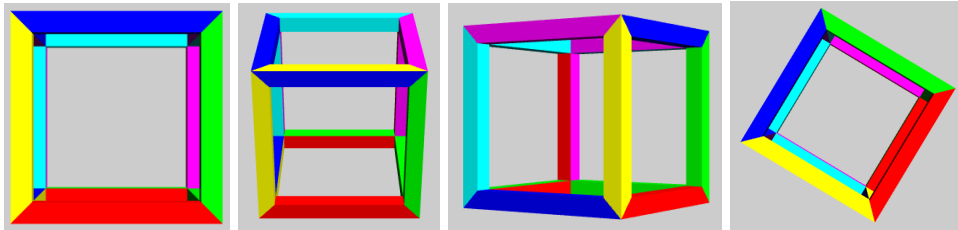


*Gambar 11. Hasil translasi sesuai slider*

Translasi diimplementasikan dengan mengalikan matriks proyeksi dengan matriks translasi dengan nilai selisih slider dengan nilai slider sebelumnya.

2. Rotasi

Rotasi digunakan untuk merotasi Object3D pada sumbu x, y, dan z. Slider ke kanan merepresentasikan nilai positif. Berikut merupakan contoh rotasi hollow object masing-masing ke sumbu x, y, dan z.

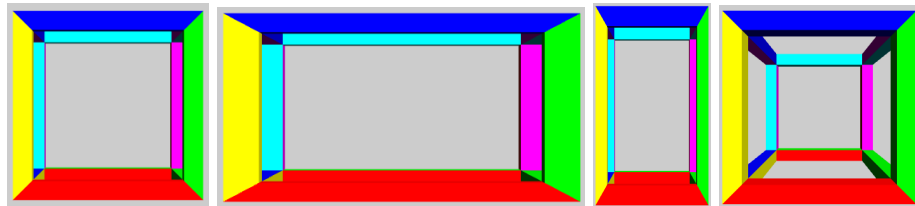


*Gambar 12. Perbandingan hollow object awal, setelah rotasi sumbu x, y, dan z.*

Rotasi diimplementasikan dengan mengalikan matriks proyeksi dengan matriks rotasi masing-masing sumbu dengan nilai selisih slider dengan nilai slider sebelumnya.

3. Scaling

Scaling digunakan untuk melebar/rampingkan objek, meninggikan/memendekkan objek, serta menebalkan/menipiskan objek (secara berurutan pada sumbu x, y, dan z). Untuk melakukan scaling, dapat menggeser slider masing-masing sumbu. Untuk membesarkan, geser ke arah luar. Semakin tengah, objek akan semakin kecil. Berikut merupakan contoh hasil scaling.



*Gambar 13. Perbandingan hollow object awal, setelah scaling sumbu x, y, dan z*

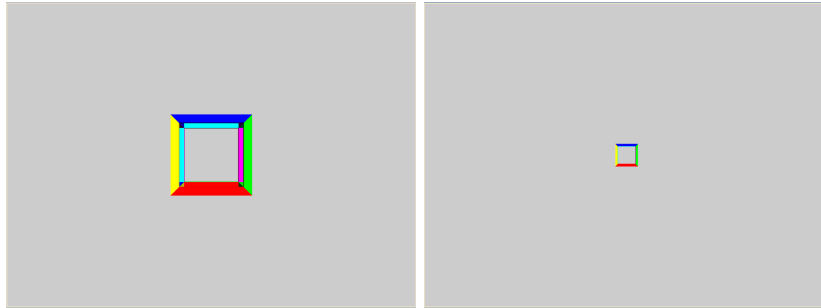
Scaling diimplementasikan dengan mengalikan matriks proyeksi dengan matriks scaling masing-masing sumbu dengan nilai proporsi slider saat ini dengan pembacaan slider sebelumnya.

## **E. Kamera**

Kamera digunakan untuk “melihat” objek pada jarak dan sudut tertentu. Oleh karena itu, terdapat 2 transformasi kamera, yakni perubahan jarak dan sudut.

1. Camera View Radius

Untuk mengubah jarak pandang kamera, digunakan slider Camera View Radius. Pergeseran ke kanan memperjauh jarak kamera dan sebaliknya. Berikut merupakan perbandingan view awal dengan menjauhkan jarak pandang kamera.

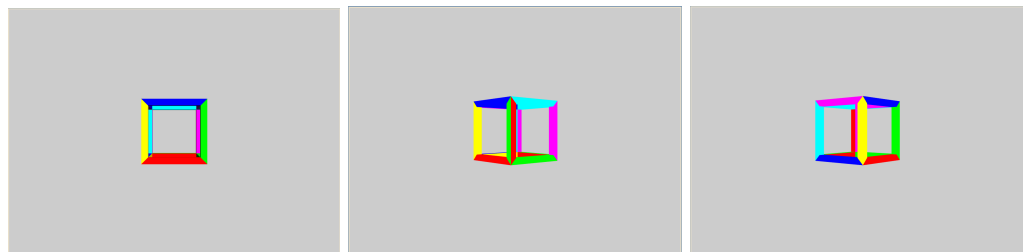


*Gambar 14. Jarak kamera default dan setelah diperjauh*

Perubahan jarak kamera diimplementasikan dengan membuat transformasi matriks tersendiri untuk menentukan posisi kamera dengan rotasi pada sumbu y, lalu mentranslasikan matriks itu pada sumbu z sejauh radius yang bernilai selisih slider saat itu dengan nilai slider sebelumnya.

## 2. Camera View Angle

Untuk mengubah sudut pandang kamera, digunakan slider Camera View Angle. Pergeseran ke kanan menggeser kamera memutar titik 0,0,0 ke arah kanan, dan sebaliknya. Selain itu, pergeseran angle dapat juga dilakukan dengan menggunakan tombol *arrow right* dan *arrow left* pada keyboard. *Arrow right* digunakan untuk menggeser kamera ke kiri, dan sebaliknya untuk *arrow left*. Berikut merupakan perbandingan awal, setelah digeser ke kanan, dan setelah digeser ke kiri.



*Gambar 15. Posisi kamera awal, setelah digeser ke kanan, dan setelah digeser ke kiri*

Perubahan angle kamera diimplementasikan dengan merotasikan matriks transformasi kamera pada sumbu y dengan nilai sudut dalam radius sesuai nilai slider saat itu. Nilai slider dibatasi antara -360 derajat hingga 360 derajat.

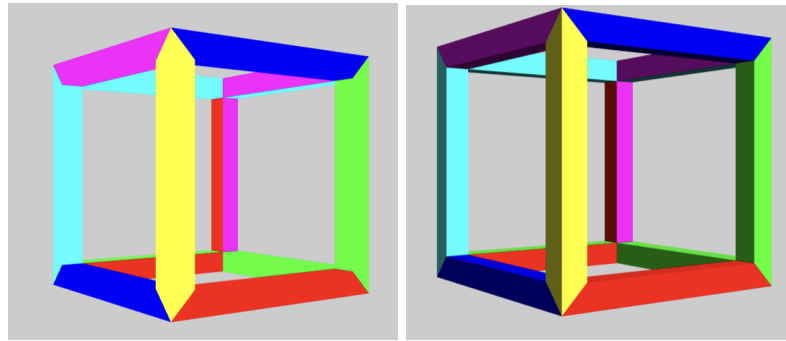
## F. Reset

Reset digunakan untuk mengembalikan *state* objek ke awal sebelum dilakukannya transformasi. Untuk melakukan reset, pengguna dapat mengklik tombol Set Default. Fungsionalitas reset diimplementasikan dengan membuat ulang Object3D sesuai *vertices*, *colors*, *normals*, dan *indices* yang telah disimpan, dengan demikian me-reset matriks transformasi, serta mengembalikan nilai slider ke nilai default dan menginisialisasi nilai variabel kamera dengan nilai default.



## G. Shading

Shading diimplementasikan menggunakan model Phong, dengan mempertimbangkan tiga refleksi yaitu ambient, diffuse, dan specular. Perhitungan warna shading dilakukan di *fragment shader*. Warna *ambient*, *diffuse*, dan *specular* didefinisikan di dalam kode *fragment shader*. Berikut merupakan tampilan Hollow Cube sebelum dan sesudah shading diaktifkan.



Gambar 16. Objek Hollow Cube sebelum dan sesudah ditambahkan shading. Pewarnaan *shading* dipengaruhi oleh normal dari tiap sisi. Untuk *diffuse*, dihitung intensitas cahaya dengan dot product antara normal dan arah datang cahaya yang kemudian dikalikan dengan warna bidang dan warna *diffuse*. Kemudian *ambient light* juga ditambahkan untuk menambahkan sedikit cahaya pada setiap sisi, warna *ambient light* diaplikasikan pada seluruh warna rgb sesuai masukan. Kemudian specular dihitung dengan memangkatkan dot antara normal dengan *halfway vector*, dengan koefisien *shininess*. Shading bisa diaktifkan atau dinonaktifkan dengan mengeklik *checkbox* “Enable shading” pada *website*.

## H. Animasi

Animasi diimplementasikan dengan program melakukan rotasi sebesar 0.01 radian terhadap sumbu *x*, dan *y*, diteruskan dengan meminta frame animasi untuk fungsi *drawScene()*. Animasi bisa diaktifkan dengan mengklik tombol “Auto Rotate” pada *website*.

## I. Bantuan

Bantuan dapat diakses dengan mengklik tombol Help. Bantuan ini menampilkan deskripsi fungsionalitas yang tersedia pada *website*. Bantuan ditampilkan dalam bentuk modal pop-up.