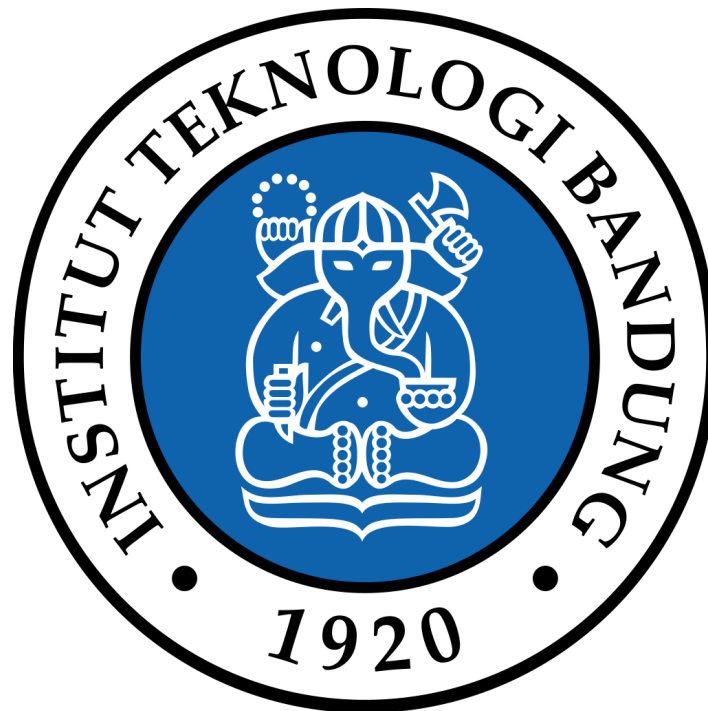


**Laporan Tugas Besar II**  
**IF3260 Grafika Komputer**  
**Hollow Objects**



**Anggota:**

Mario Gunawan	13518114
Naufal Dean Anugrah	13518123
Stephen Thajeb	13518150

**Institut Teknologi Bandung**  
**2020**

## Daftar Isi

<b>Daftar Isi</b>	<b>2</b>
<b>Bab I</b>	
<b>Deskripsi</b>	<b>3</b>
<b>Bab II</b>	
<b>Hasil Pengerjaan</b>	<b>4</b>
Halaman utama	4
Menu Help	6
Model Prism (Prisma Segitiga)	7
Model Cube (Kubus)	8
Model Block (Balok)	9
Shading	9
File Penyimpanan Model	10
<b>Bab III</b>	
<b>Manual</b>	<b>11</b>
<b>Bab IV</b>	
<b>Kontribusi Individual</b>	<b>12</b>

## Bab I

### Deskripsi

Dalam tugas ini kami membuat 3 hollow object. Hollow object adalah objek 3 dimensi yang tengah tengahnya berongga sehingga kita bisa melihat latar dari canvas. Adapun spesifikasi tambahan untuk hollow object ini adalah kami harus membuat beberapa transformasi yang saat diaplikasikan akan mengubah hollow object tersebut. Selain itu, kami harus mengimplementasikan shading untuk masing masing objek, sehingga dapat terlihat warna dan bayangan dari setiap objek.

Untuk menuliskan kode yang lebih efisien, kami menuliskan kode program dengan paradigma object oriented programming di mana kami membuat class utama:

- **abstract class** Shape
- **class** SlideManager

Karena dalam tugas ini dibutuhkan membuat 3 hollow objects, dan masing-masing hollow objects akan memiliki persamaan method yang dimiliki, maka kami mengeneralisasikan method-method tersebut di **abstract class** Shape dan dari class tersebut kami inherit ke 3 buah class yang masing-masing merepresentasikan 1 buah hollow object antara lain:

1. **class** TriangularPrism yang merepresentasikan bangun ruang Prisma Segitiga hollow
2. **class** Cube yang merepresentasikan bangun ruang Kubus hollow
3. **class** Block yang merepresentasikan bangun ruang Balok hollow (penjelasan lebih lanjut ketiga buah class tersebut akan dijelaskan di Bab B)

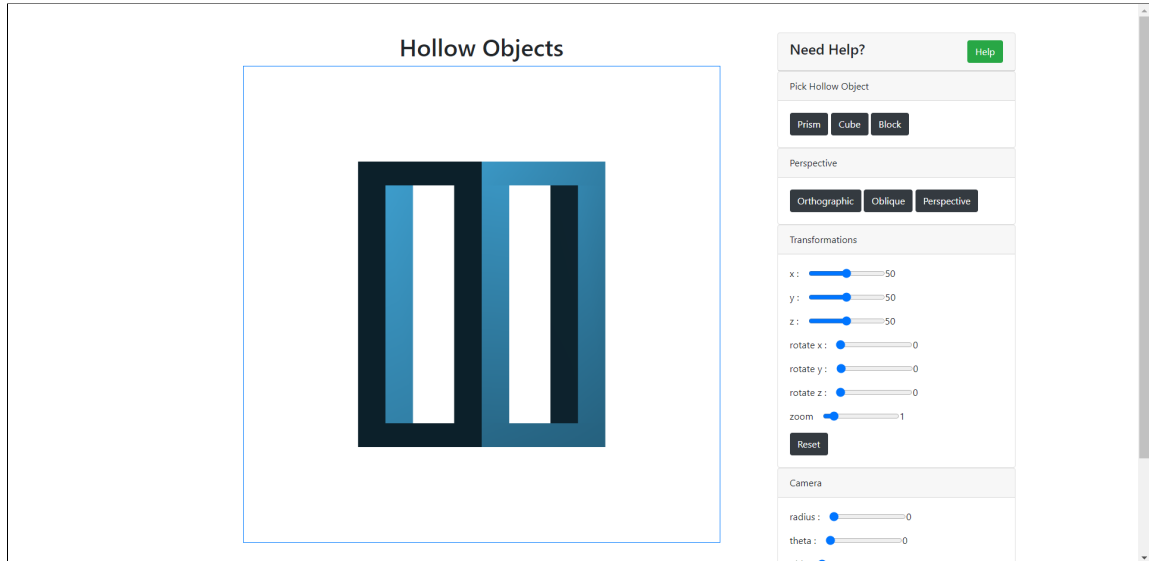
Terkait **class** SlideManager, class tersebut memiliki responsibility untuk sebagai controller untuk mengatur transformasi (translasi, rotasi, scaling), pengaturan perspektif (orthographic, oblique, perspective) dengan mengubah komponen matriks M, V, atau P dalam membentuk bangun ruang.

## Bab II

### Hasil Pengerjaan

#### A. Halaman utama

Berikut adalah screenshot halaman website kami saat pertama diloat:

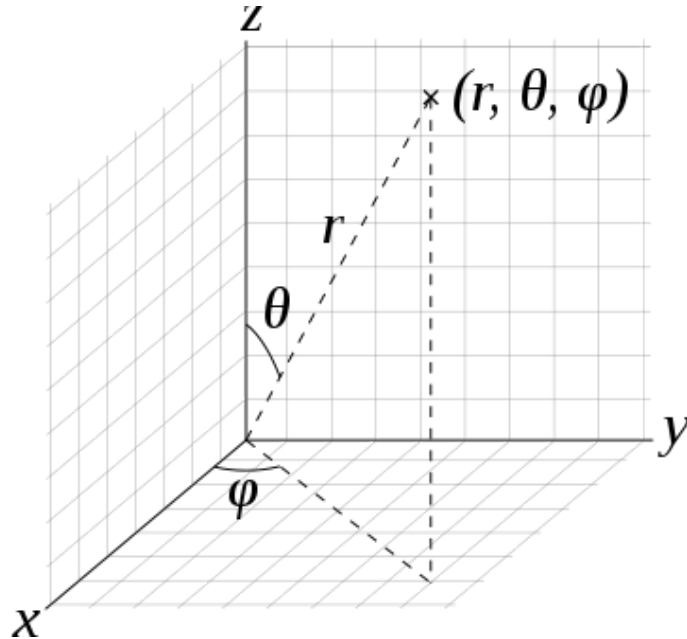


Gambar 2.1 Halaman utama website

Terdapat dua bagian utama pada halaman tersebut. Pertama, pada sisi kiri terdapat canvas yang digunakan untuk menampilkan model yang sedang aktif. Kedua, pada sisi kanan terdapat menu yang dapat digunakan untuk berinteraksi dengan model, yaitu:

- Pada section paling atas, yaitu "Need Help?". Terdapat tombol Help yang dapat digunakan untuk menampilkan popup yang berisi manual untuk menggunakan website.
- Pada section "Pick Hollow Object", terdapat tiga tombol yang dapat digunakan untuk mengganti model yang sedang aktif, masing-masing untuk model prism (prisma segitiga), cube (kubus), dan block (balok).
- Pada section "Perspective", terdapat tiga tombol yang dapat digunakan untuk mengganti proyeksi (orthographic, oblique, dan perspektif) dari model.
- Pada bagian "Transformation", terdapat 6 slider yang dapat digunakan untuk mentransformasi model. Translasi menggunakan slider 1-3, rotasi menggunakan slider 4-6, dan scaling menggunakan slider 7. Selain itu, terdapat tombol Reset untuk mereset ke transformasi default.

- e. Pada section “Camera”, terdapat 3 slider yang dapat digunakan untuk mengubah posisi kamera. Posisi kamera dinyatakan menggunakan koordinat spherical sehingga pergerakan kamera mendekat-menjauh dan orbit untuk mengitari model menjadi lebih mudah. Selain itu, terdapat tombol Reset untuk mereset ke posisi kamera default (di posisi (0,0,0) mengarah ke sumbu z negatif).



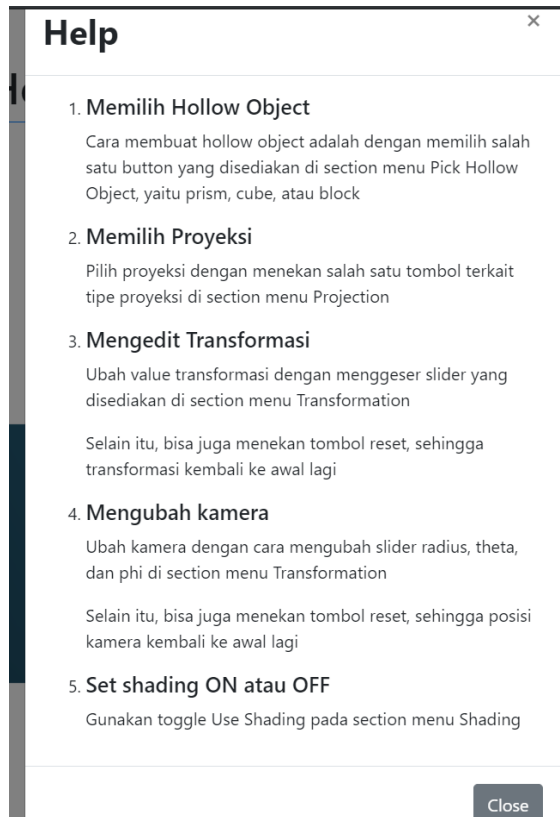
Gambar 2.2 Koordinat spherical

Sumber: [https://en.wikipedia.org/wiki/Spherical\\_coordinate\\_system](https://en.wikipedia.org/wiki/Spherical_coordinate_system)

- f. Pada section “Shading”, terdapat switch Use Shading yang dapat digunakan untuk menghidupkan dan mematikan shading pada model.

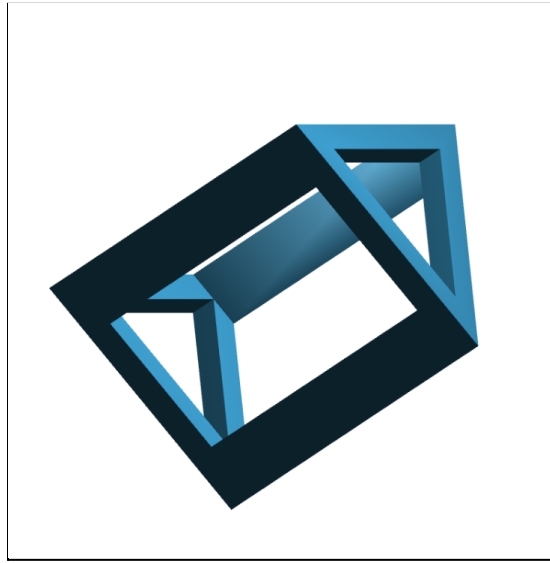
## B. Menu Help

Menu help dapat diakses dengan menekan tombol help yang berada pada pojok kanan yang akan menampilkan pop up instruksi tersebut.



Gambar 2.3 Popup menu help

### C. Model Prism (Prisma Segitiga)

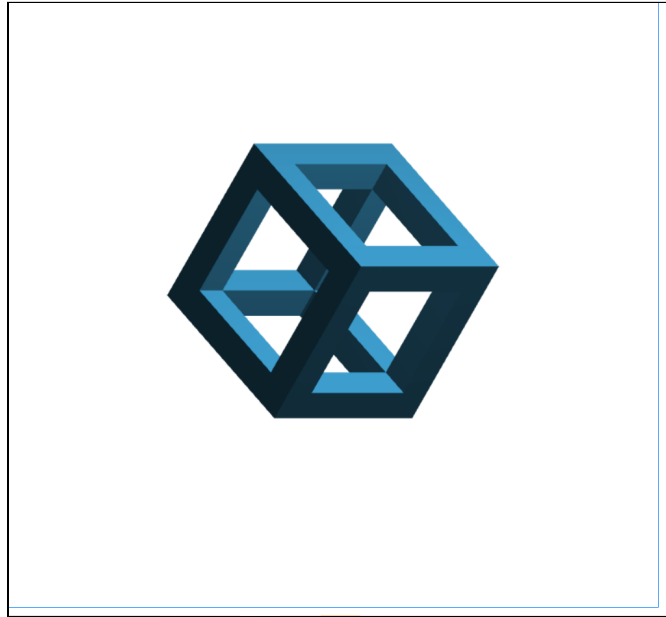


Gambar 2.4 Model prism (prisma segitiga)

Penggambaran bangun ruang prisma segitiga digambarkan dengan menggambarkan rusuk-rusuk bagian badan (vertikal) dengan sisi-sisinya dibuat sebagai bangun datar persegi panjang. Sementara untuk bagian alas dan atas prisma digambarkan dengan penggambaran area di antara 2 buah segitiga, dengan salah 1 segitiga memiliki ukuran lebih besar (outer triangle) dari yang lain (inner triangle). Terdapat beberapa fungsi helper utama untuk membangkitkan titik-titik yang dibutuhkan, yaitu **generateTrianglePoints**, **buildHollowTriangle**, dan **buildTriangleConnector**.

Untuk menggambarkan segi empat yang menjadi penyusun semua rusuk pada model tersebut, digunakan fungsi **buildQuad**. Fungsi di atas akan menerima 4 buah titik dan menerima sebuah array Normal yang merepresentasikan kontainer di mana vektor normal dari vertex pada quad tersebut akan disimpan. Vektor normal tersebut dihitung dengan cross product vektor sisi pada quad tersebut.

#### D. Model Cube (Kubus)

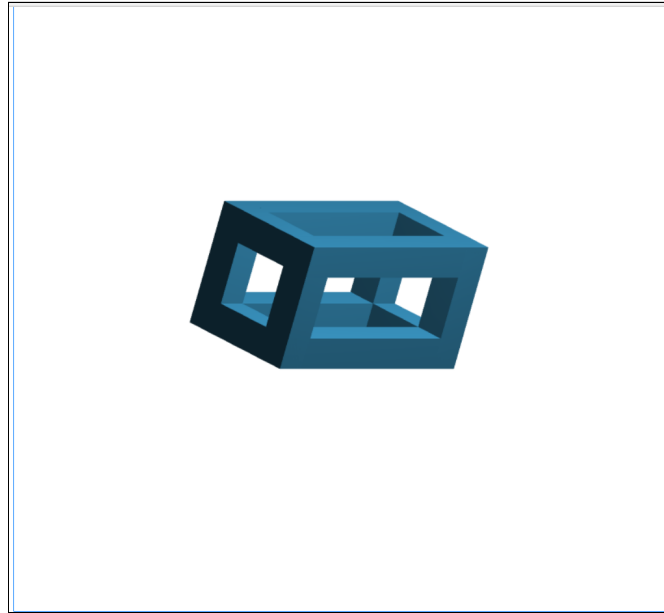


Gambar 2.5 Model cube (kubus)

Konsep penggambaran bangun ruang kubus yang hollow hampir sama dengan penggambaran prisma yaitu dengan menggambarkan masing-masing rusuk kubus yang direpresentasikan sebagai bangun datar (2D) persegi panjang. Namun lebih mudah dari penggambaran prisma, untuk kasus kubus, semua rusuk bangun adalah homogen. Untuk menggambarkan kubus dibuatkan fungsi bantu **createSquare2D(po,p1,..p3,w,orientation)** di mana  $p_0,..p_3$  merepresentasikan 4 buah titik sudut sebuah persegi dan  $w$  merepresentasikan tebal hollow object tersebut secara 2D, dan  $orientation$  merepresentasikan sisi yang hendak digambarkan (front,ground,or side).



## E. Model Block (Balok)



Gambar 2.6 Model block (balok)

Untuk penggambaran bangun ruang balok hollow, menggunakan konsep yang sama seperti penggambaran sebuah kubus hollow, yaitu menggambarkan rusuk bangun sebagai persegi panjang. Perbedaannya hanya terletak pada inputan yang dimasukkan saat menggunakan fungsi `createSquare2D`. Pada kasus balok, inputan point-point dibuat tidak homogen untuk jenis orientasi yang berbeda akibatnya terbentuk ukuran bangun datar persegi panjang yang berbeda-beda tergantung pada sisi bangun.

## F. Shading

Shading dibuat dengan menggunakan model blinn-phong. Kalkulasi komponen diffuse, specular, dan ambient dilakukan pada vertex shader dan kemudian hasilnya akan dipass ke fragment shader untuk diassign ke `gl_FragColor`. Kedua kode shader ini dapat dilihat pada file **`src/shapes/shape.ts`**. Proses build model blinn-phong tersebut memanfaatkan nilai properti cahaya ( $I_d$ ,  $I_s$ , dan  $I_a$ ), properti material ( $K_d$ ,  $K_s$ ,  $K_a$ , dan shininess), serta posisi sumber cahaya yang dibuat fix di posisi (1,1,1).

## G. File Penyimpanan Model

Vertex, normal, fungsi helper untuk proses generasi model, dsb. dapat dilihat pada folder **src/shapes/initialPoints** dengan data model prism (prisma segitiga) berada di file **prismPoints.ts**, model cube (kubus) berada di file **cubePoints.ts**, dan model block (balok) berada di file **blockPoints.ts**.

Kemudian, proses load data model tersebut dilakukan pada kelas model masing-masing di folder **src/shapes**, secara berturut-turut pada file **triangularPrism.ts**, **cube.ts**, dan **block.ts**. Lebih tepatnya di method **setupPoints**.

## **Bab III**

### **Manual**

Berikut ini adalah manual untuk menggunakan website yang telah dibuat. Manual poin b hingga f juga dapat dilihat pada menu help di website tersebut.

- a. Mendapatkan bantuan  
Click tombol help di bagian kanan atas menu
- b. Memilih Hollow Object  
Cara membuat hollow object adalah dengan memilih salah satu button yang disediakan di section menu Pick Hollow Object, yaitu prism, cube, atau block
- c. Memilih Proyeksi  
Pilih proyeksi dengan menekan salah satu tombol terkait tipe proyeksi di section menu Projection
- d. Mengedit Transformasi  
Ubah value transformasi dengan menggeser slider yang disediakan di section menu Transformation  
Selain itu, bisa juga menekan tombol reset, sehingga transformasi kembali ke awal lagi
- e. Mengubah kamera  
Ubah kamera dengan cara mengubah slider radius, theta, dan phi di section menu Transformation  
Selain itu, bisa juga menekan tombol reset, sehingga posisi kamera kembali ke awal lagi
- f. Set shading ON atau OFF  
Gunakan toggle Use Shading pada section menu Shading

## Bab IV

### Kontribusi Individual

Tabel 4.1 Kontribusi individual

NIM	Nama	Kontribusi
13518114	Mario Gunawan	<ul style="list-style-type: none"><li>- Membuat shape balok</li><li>- Membuat struktur kode awal dan framework</li><li>- Membuat helper dan halaman HTML utama termasuk slider</li><li>- Membuat transformasi dasar (yang masih buggy), rotasi, translasi, dan zoom (scale)</li><li>- Membantu membuat shading untuk kubus</li></ul>
13518123	Naufal Dean Anugrah	<ul style="list-style-type: none"><li>- Membuat shape prisma segitiga</li><li>- Membuat sistem proyeksi model</li><li>- Membuat sistem pergerakan kamera</li><li>- Membuat shading menggunakan model blinn-phong</li><li>- Membuat switch shading on/off</li><li>- Membuat fungsionalitas reset transformasi dan kamera</li></ul>
13518150	Stephen Thajeb	<ul style="list-style-type: none"><li>- Membuat shape kubus</li><li>- Style layout tampilan</li><li>- Ikut menuliskan laporan</li></ul>