

# APLIKASI PRATINJAU 3 DIMENSI BERBASIS WEB

NANCY VALENTINA—2014730049

## 1 Data Skripsi

Pembimbing utama/tunggal: **Pascal Alfadian Nugroho**

Pembimbing pendamping: -

Kode Topik : **PAN4304**

Topik ini sudah dikerjakan selama : **1 semester**

Pengambilan pertama kali topik ini pada : **Semester 43 - Ganjil 17/18**

Pengambilan pertama kali topik ini di kuliah : **Skripsi 1**

Tipe Laporan : **B** - Dokumen untuk reviewer pada presentasi dan **review Skripsi 1**

## 2 Detail Perkembangan Pengerjaan Skripsi

Detail bagian pekerjaan skripsi sesuai dengan rencan kerja/laporan perkembangan terakhir :

1. Mempelajari standar WebGL sebagai *Application Programming Interface* untuk menampilkan grafis 3 dimensi pada *web browser*).

**Status** : Ada sejak rencana kerja skripsi.

**Hasil** : Berikut ini merupakan hasil dari pembelajaran standar WebGL sebagai *Application Programming Interface* untuk menampilkan grafis 3 dimensi pada *web browser*:

WebGL adalah sebuah Application Programming Interface (API) yang membangun objek 3 dimensi dengan mode langsung yang dirancang untuk *web*. WebGL diturunkan dari OpenGL ES 2.0, WebGL menyediakan fungsi pembangunan sejenis tetapi di dalam konteks HTML. WebGL dirancang sebagai konteks pembangunan objek pada elemen *canvas* HTML. *Canvas* pada HTML menyediakan suatu destinasi untuk pembangunan objek secara programatik pada halaman *web* dan memungkinkan menampilkan objek yang sedang dibangun menggunakan API pembangun objek yang berbeda. Berikut ini merupakan *interfaces* dan fungsionalitas yang ada pada WebGL:

### (a) *WebGLObject*

*Interface WebGLObject* merupakan *interface* awal untuk diturunkan kepada semua objek GL.

```
interface WebGLObject {
};
```

Listing 1: *Interface* awal pada WebGL.

### (b) *WebGLFrameBuffer*

*Interface WebGLFrameBuffer* merepresentasikan sebuah OpenGL *Frame Buffer Object*.

```
interface WebGLFramebuffer : WebGLObject {
};
```

Listing 2: *Frame Buffer Object* pada OpenGL.

### (c) *WebGLProgram*

*Interface WebGLProgram* merepresentasikan sebuah OpenGL *Program Object*.

```
interface WebGLProgram : WebGLObject {
};
```

Listing 3: *Program Object* pada OpenGL.

*WebGLShader*

*Interface WebGLShader* merepresentasikan sebuah OpenGL *Shader Object*.

```
interface WebGLShader : WebGLObject {
};
```

Listing 4: *Shader Object* pada OpenGL.

(d) *WebGLTexture*

*Interface WebGLTexture* merepresentasikan sebuah OpenGL *Texture Object*.

```
interface WebGLTexture : WebGLObject {
};
```

Listing 5: *Texture Object* pada OpenGL.

(e) *ArrayBuffer* dan *Typed Arrays*

*Vertex*, *index*, *texture*, dan data lainnya ditransfer ke implementasi WebGL menggunakan *ArrayBuffer*, *Typed Arrays*, dan *DataViews* seperti yang telah didefinisikan pada spesifikasi ECMA-Script.

```
var numVertices = 100; // for example

// Hitung ukuran buffer yang dibutuhkan dalam bytes dan floats
var vertexSize = 3 * Float32Array.BYTES_PER_ELEMENT +
4 * Uint8Array.BYTES_PER_ELEMENT;
var vertexSizeInFloats = vertexSize / Float32Array.BYTES_PER_ELEMENT;

// Alokasikan buffer
var buf = new ArrayBuffer(numVertices * vertexSize);

// Map buffer ke Float32Array untuk mengakses posisi
var positionArray = new Float32Array(buf);

// Map buffer yang sama ke Uint8Array untuk mengakses warna
var colorArray = new Uint8Array(buf);

// Inisialisasi offset dari vertices dan warna pada buffer
var positionIdx = 0;
var colorIdx = 3 * Float32Array.BYTES_PER_ELEMENT;

// Inisialisasi buffer
for (var i = 0; i < numVertices; i++) {
    positionArray[positionIdx] = ...;
    positionArray[positionIdx + 1] = ...;
    positionArray[positionIdx + 2] = ...;
    colorArray[colorIdx] = ...;
```

```

        colorArray[ colorIdx + 1] = ...;
        colorArray[ colorIdx + 2] = ...;
        colorArray[ colorIdx + 3] = ...;
        positionIdx += vertexSizeInFloats;
        colorIdx += vertexSize;
    }

```

Listing 6: Transfer data ke implementasi WebGL.

- (f) *WebGL Context WebGLRenderingContext* merepresentasikan API yang memungkinkan gaya pembangunan OpenGL ES 2.0 ke elemen *canvas*.
- (g) *WebGLContextEvent* WebGL menghasilkan sebuah *WebGLContextEvent* sebagai respon dari perubahan penting pada status konteks pembangunan WebGL. *Event* tersebut dikirim melalui *DOM Event System* dan dilanjutkan ke *HTMLCanvasEvent* yang diasosiasikan dengan konteks pembangunan WebGL.

## 2. Mempelajari penggunaan pustaka Three.js sebagai pustaka perantara untuk penggunaan WebGL.

**Status :** Ada sejak rencana kerja skripsi.

**Hasil :** Berikut ini merupakan hasil dari pembelajaran Three.js sebagai pustaka perantara untuk penggunaan WebGL:

Pustaka Three.js ini bertujuan untuk membuat pustaka 3 dimensi yang mudah dan ringan untuk digunakan. Pustaka ini menyediakan `<canvas>`, `<svg>`, CSS3D, dan pembangun WebGL. Terdapat beberapa fungsi penting yang disediakan oleh pustaka Three.js dalam pembuatan grafis 3 dimensi, di antaranya adalah:

- *Cameras*

- *Camera*, kelas abstrak untuk *cameras*. Kelas ini harus selalu diimplementasikan saat membangun suatu kamera. Konstruktorku pada kelas ini digunakan untuk membuat kamera baru, namun kelas ini tidak dipergunakan secara langsung melainkan menggunakan *PerspectiveCamera* atau *OrthographicCamera*.
- *PerspectiveCamera*, kamera yang menggunakan proyeksi perspektif. Konstruktorku pada kelas ini menerima parameter berupa *frustum* pandangan vertikal, *frustum* pandangan horizontal, dan *frustum* jarak dekat, dan *frustum* jarak jauh. Contoh untuk kelas *PerspectiveCamera* dapat dilihat pada *listing 7*.

```

var camera = new THREE.PerspectiveCamera( 45, width / height ,
1, 1000 );
scene.add( camera );

```

Listing 7: Contoh instansiasi kelas *PerspectiveCamera*

- *Geometries*

- *BoxGeometry*, merupakan kelas primitif geometri berbentuk segi empat. Contoh penggunaannya sama seperti kelas *BoxBufferGeometry*. Konstruktorku pada kelas ini menerima parameter berupa lebar sisi pada sumbu X dengan nilai awal adalah 1, tinggi sisi pada sumbu Y dengan nilai awal adalah 1, kedalaman sisi pada sumbu Z dengan nilai awal adalah 1, jumlah permukaan yang berpotongan dengan lebar sisi dengan nilai awal adalah 1 dan bersifat fakultatif, jumlah permukaan yang berpotongan dengan tinggi sisi dengan nilai awal adalah 1 dan bersifat fakultatif, dan jumlah permukaan yang berpotongan dengan kedalaman sisi dengan nilai awal adalah 1 dan bersifat fakultatif

- *Lights*

- *AmbientLight*, sebuah cahaya yang menyinari objek secara global dan merata. Konstruktor pada kelas ini menerima parameter berupa warna dalam RGB dan intensitas. Contoh untuk kelas *AmbientLight* dapat dilihat pada *listing 2.38*.

- *Loaders*

- *JSONLoader*, sebuah pemuat untuk memuat objek dalam format JSON. Konstruktor pada kelas ini menerima parameter berupa *loadingManager*. Contoh untuk kelas *JSONLoader* dapat dilihat pada *listing 8*.

```
// inisiasi pemuat
var loader = new THREE.JSONLoader();

// memuat sumber daya
loader.load(

    // sumber daya URL
    'models/animated/monster/monster.js',

    // fungsi yang dijalankan saat sumber daya telah dimuat
    function ( geometry, materials ) {

        var material = materials[ 0 ];
        var object = new THREE.Mesh( geometry, material );

        scene.add( object );

    }

);
```

Listing 8: Contoh penggunaan kelas *JSONLoader*.

- *Loader*, kelas dasar untuk implementasi pemuat.
- *TextureLoader*, kelas untuk memuat tekstur. Konstruktor pada kelas ini menerima parameter berupa *loadingManager*. Contoh untuk kelas *TextureLoader* dapat dilihat pada *listing 9*.

```
// inisiasi pemuat
var loader = new THREE.TextureLoader();

// memuat sumber daya
loader.load(

    // sumber daya URL
    'textures/land_ocean_ice_cloud_2048.jpg',

    // fungsi yang dijalankan saat sumber daya telah dimuat
    function ( texture ) {

        // melakukan sesuatu dengan tekstur
        var material = new THREE.MeshBasicMaterial( {
            map: texture
        } );

    },
```

```

// fungsi yang dipanggil saat unduh dalam proses
function ( xhr ) {
    console.log( (xhr.loaded / xhr.total * 100)
        + '% loaded' );
},
// fungsi yang dipanggil saat unduh gagal
function ( xhr ) {
    console.log( 'An error happened' );
}
);

```

Listing 9: Contoh penggunaan kelas *TextureLoader*.

- *Materials*

- *Material*, kelas dasar abstrak untuk bahan.
- *MeshBasicMaterial*, sebuah bahan untuk menggambar geometri dengan cara sederhana yang datar. Konstruktor pada kelas ini menerima parameter berupa objek dan bersifat fakultatif.

- *Objects*

- *Mesh*, sebuah kelas yang merepresentasikan object dengan dasar segitiga. Konstruktor pada kelas ini menerima parameter berupa geometri dan material. Contoh untuk kelas *Mesh* dapat dilihat pada *listing 10*.

```

var geometry = new THREE.BoxBufferGeometry( 1, 1, 1 );
var material = new THREE.MeshBasicMaterial( { color: 0xffff00 } );
var mesh = new THREE.Mesh( geometry, material );
scene.add( mesh );

```

Listing 10: Contoh penggunaan kelas *Mesh*.

- *Renderers*

- *WebGLRenderer*, pembangun WebGL menampilkan layar indah yang dibuat oleh Anda menggunakan WebGL. Konstruktor pada kelas ini menerima parameter berupa *canvas*, konteks, presisi, dan parameter relevan lainnya.

- *Scenes*

- *Scene*, sebuah layar yang memungkinkan untuk membuat dan menempatkan sesuatu pada pustaka Three.js.

### 3. Memodelkan ruangan belajar mengajar secara 3 dimensi.

**Status :** Ada sejak rencana kerja skripsi.

**Hasil :** Ruang belajar mengajar yang dipilih untuk dijadikan acuan pemodelan adalah ruangan kelas pada gedung sembilan lantai satu di Universitas Katolik Parahyangan. Ruang tersebut kurang lebih dapat menampung 60 orang. Pada ruangan tersebut terdapat objek-objek yang dapat mendukung kegiatan belajar mengajar seperti kursi mahasiswa, kursi dosen, meja dosen, proyektor, layar, papan tulis, pendingin ruangan, dan lain-lain. Pemodelan ruangan belajar mengajar secara tiga dimensi ini dilakukan langsung pada web dengan memanfaatkan pustaka Three.js. Pustaka tersebut berperan untuk membangun objek kubus yang merepresentasikan ruangan belajar mengajar. Kemudian dibangun juga objek-objek pendukung yang merepresentasikan ruangan tersebut dengan menggunakan sebuah perangkat lunak bernama Blender. Blender berfungsi untuk memodelkan objek tiga dimensi seperti meja dan kursi pada ruangan, objek tersebut kemudian dapat diekspor dan digunakan sesuai kebutuhan. Pada skripsi ini hasil representasi objek dari Blender diekspor menjadi format JSON untuk

mendukung bahasa pemrograman yang digunakan pada pembuatan aplikasi *web* untuk skripsi ini. Format JSON hanya tersedia apabila naskah tambahan dari pustaka Three.js telah dimasukkan ke dalam perangkat lunak Blender.

#### 4. Melakukan analisis terhadap situs *web* yang akan dibangun.

**Status :** Ada sejak rencana kerja skripsi.

**Hasil :** Berdasarkan hasil analisis, situs *web* akan dibangun dengan menggunakan bahasa pemrograman JavaScript serta bahasa *markup* HyperText Markup Language (HTML). Pemilihan bahasa tersebut didasari oleh digunakannya pustaka Three.js yang juga dibuat dengan bahasa pemrograman JavaScript, sehingga implementasi dari situs *web* akan menjadi lebih mudah. Kemudian untuk tampilan situs, akan dibuat satu halaman saja untuk memudahkan pengguna dalam berinteraksi dengan aplikasi pratinjau tiga dimensi ini. Pada satu halaman tersebut, akan disediakan tampilan pemodelan dari ruang belajar mengajar serta satu kolom di pojok kanan tempat pengguna memilih tekstur dinding dan lantai.

#### 5. Merancang tampilan situs *web* yang akan dibangun.

**Status :** Ada sejak rencana kerja skripsi.

**Hasil :** Keseluruhan layar situs *web* akan menampilkan pemodelan ruangan belajar mengajar pada gedung sembilan lantai satu di Universitas Katolik Parahyangan sampai dengan diimplementasikannya fitur untuk mengganti warna dinding bagian atas, dinding bagian bawah, dan lantai nantinya.

#### 6. Mengimplementasikan situs *web*.

**Status :** Ada sejak rencana kerja skripsi.

**Hasil :** Telah berhasil diimplementasikan pemodelan ruangan belajar mengajar gedung sembilan lantai satu di Universitas Katolik Parahyangan ke dalam situs *web*.



Gambar 1: Pemodelan ruangan belajar mengajar gedung sembilan lantai satu di Universitas Katolik Parahyangan (1).



Gambar 2: Pemodelan ruangan belajar mengajar gedung sembilan lantai satu di Universitas Katolik Parahyangan (2).



Gambar 3: Pemodelan ruangan belajar mengajar gedung sembilan lantai satu di Universitas Katolik Parahyangan (3).

7. Melakukan pengujian terhadap situs *web* yang telah dibangun.

**Status :** Ada sejak rencana kerja skripsi.

**Hasil :**

8. Menulis dokumen skripsi.

**Status :** Ada sejak rencana kerja skripsi.

**Hasil :** Dokumen skripsi telah ditulis dari bab 1 sampai dengan bab 2. Pada bab 1 dibahas mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, serta metodologi dan sistematika pembahasan. Bab 2 berisikan studi literatur tentang WebGL dan Pustaka Three.js.

### 3 Pencapaian Rencana Kerja

Persentase penyelesaian skripsi sampai dengan dokumen ini dibuat dapat dilihat pada tabel berikut :

| 1*    | 2*(%) | 3*(%) | 4*(%) | 5*                | 6*(%) |
|-------|-------|-------|-------|-------------------|-------|
| 1     | 8     | 8     |       |                   | 8     |
| 2     | 8     | 8     |       |                   | 8     |
| 3     | 15    | 15    |       |                   | 10    |
| 4     | 6     | 6     |       |                   | 3     |
| 5     | 8     |       | 8     |                   | 3     |
| 6     | 30    |       | 30    |                   | 10    |
| 7     | 10    |       | 10    |                   |       |
| 8     | 15    | 3     | 12    | bab 1 dan 2 di S1 | 3     |
| Total | 100   | 40    | 60    |                   | 45    |

Keterangan (\*)

1 : Bagian pengerjaan Skripsi (nomor disesuaikan dengan detail pengerjaan di bagian 5)

2 : Persentase total

3 : Persentase yang akan diselesaikan di Skripsi 1

4 : Persentase yang akan diselesaikan di Skripsi 2

5 : Penjelasan singkat apa yang dilakukan di S1 (Skripsi 1) atau S2 (skripsi 2)

6 : Persentase yang sudah diselesaikan sampai saat ini

## 4 Kendala yang dihadapi

Bandung, 23/11/2017

Nancy Valentina

Menyetujui,

Nama: Pascal Alfadian Nugroho  
Pembimbing Tunggal