

SKRIPSI

VIRTUAL JOGGING APP UNTUK GOOGLE CARDBOARD



RICHARD WIJAYA

NPM: 2016730014

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2020

UNDERGRADUATE THESIS

VIRTUAL JOGGING APP FOR GOOGLE CARDBOARD



RICHARD WIJAYA

NPM: 2016730014

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2020**

LEMBAR PENGESAHAN

VIRTUAL JOGGING APP UNTUK GOOGLE CARDBOARD

RICHARD WIJAYA

NPM: 2016730014

Bandung, «**tanggal**» «**bulan**» 2020

Menyetujui,

Pembimbing

Pascal Alfadian, M.Comp.

Ketua Tim Penguji

Anggota Tim Penguji

«**penguji 1**»

«**penguji 2**»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

VIRTUAL JOGGING APP UNTUK GOOGLE CARDBOARD

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal «tanggal» «bulan» 2020

Meterai Rp. 6000

RICHARD WIJAYA
NPM: 2016730014

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Kata-kata kunci: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Keywords: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»

«kepada siapa anda mempersembahkan skripsi ini. . . ?»

KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini . . . »

Bandung, «bulan» 2020

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	1
1.4 Batasan Masalah	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	3
2.1 Google VR	3
2.1.1 Google VR SDK	3
2.1.2 Aplikasi HelloVR	3
2.2 Google <i>StreetView API</i>	6
2.2.1 <i>API Key</i>	6
2.2.2 Penggunaan <i>StreetView API</i>	6
2.2.3 Atribut Parameter <i>StreetView API</i>	7
2.3 Google <i>Directions API</i>	8
2.3.1 Penggunaan <i>Directions API</i>	9
2.3.2 Hasil Pemanggilan <i>Directions API</i>	9
2.4 <i>Motion Sensor</i>	12
3 ANALISIS	13
3.1 Dunia VR	13
3.2 Menampilkan <i>StreetView API</i> pada Dunia VR	13
3.3 Integrasi dengan <i>Directions API</i>	14
3.3.1 Menentukan Atribut yang Bermanfaat	14
3.3.2 Cara Memanfaatkan Atribut	15
3.4 Cara Memanfaatkan <i>Step Detector Sensor</i>	15

DAFTAR GAMBAR

2.1	Tampilan <i>UI</i> permainan <i>treasure hunt</i> pada aplikasi HelloVR	5
2.2	Gambar <i>asset</i> dari tekstur ruangan	5
2.3	Gambar <i>asset</i> dari <i>texture</i> bentuk yang sedang tidak di tengah-tengah titik pengelihatan pengguna	6
2.4	Gambar <i>asset</i> dari <i>texture</i> bentuk yang sedang ada di tengah-tengah titik pengelihatan pengguna	7
2.5	Tampilan <i>UI Google Cloud</i> saat mengakses <i>API Key (API Key disamaraskan)</i>	7
2.6	Pemanggilan <i>StreetView API</i> yang berhasil	8
3.1	Tampilan <i>UI Blender</i> Blender versi 2.81	13
3.2	Contoh hasil penggabungan empat gambar <i>StreetView</i>	14

DAFTAR TABEL

2.1 Beberapa <i>class</i> dan <i>interface</i> pada <i>package</i> com.google.vr.sdk.base	4
2.2 Atribut-Atribut Opsional <i>StreetView API</i>	9
2.3 Atribut-Atribut Opsional <i>Directions API</i>	10

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Zaman modern adalah zaman saat profesi sudah dan sedang berkembang sehingga ada banyak sekali jumlah bidangnya serta perkembangannya. Mayoritas orang menekuni bidang-bidang profesi yang tak terhitung banyaknya untuk mengembangkan setiap bidang profesi. Hal ini menyebabkan kesulitan pengaturan waktu untuk berolahraga, yang adalah salah satu kebutuhan manusia untuk menjaga kesehatan. Salah satu aktivitas olahraga yang paling mudah dan tidak memerlukan gerakan yang sulit adalah berlari, namun kegiatan ini memerlukan lahan yang cukup besar agar dapat dilakukan dengan leluasa. Selain kebutuhan lahan, aktivitas berlari lebih menyenangkan jika dilakukan di luar rumah. Agar dapat dilakukan di dalam rumah, berlari dapat dilakukan di rumah adalah *treadmill*, akan tetapi masalah lingkungan yang monoton dan membosankan di dalam rumah membuat orang enggan untuk melakukan aktivitas berlari. Bila suasana dunia luar dapat dibawa ke dalam rumah, aktivitas ini dapat dilakukan di dalam rumah, tetapi suasana yang dirasakan adalah seperti di luar rumah.

Pada skripsi ini, akan dibuat sebuah perangkat lunak yang dapat menampilkan simulasi aktivitas berlari pada lingkungan yang diinginkan saat berlari di *treadmill*. Dengan menggunakan perangkat lunak tersebut, orang yang berlari dapat menikmati pemandangan yang dipilih saat berlari di dalam rumah sehingga merasa seperti berlari di lingkungan yang dipilih tersebut.

Teknologi yang dapat dimanfaatkan untuk membuat aplikasi VR untuk berlari adalah *Google Cardboard* dan sensor perangkat bergerak, dan untuk *Application Programming Interface (API)* yang digunakan adalah *Google Streetview API* dan *Google Directions API*.

1.2 Rumusan Masalah

Rumusan masalah yang ada pada skripsi ini adalah:

- Bagaimana memanfaatkan Google VR SDK for Android untuk menampilkan gambar dengan perangkat VR?
- Bagaimana menampilkan hasil dari *Google StreetView API* dalam bentuk VR?
- Bagaimana mengintegrasikan *Google Directions API*, gambar *Google StreetView* dan Google VR dalam perangkat lunak *virtual jogging*?

1.3 Tujuan

Pada skripsi ini, hal-hal yang coba untuk dicapai adalah :

- Menggunakan Google VR SDK for Android untuk menampilkan gambar dengan *Google Cardboard*.
- Menampilkan hasil gambar dari *Google StreetView API* pada *Google Cardboard*.

- Mengintegrasikan *Google Directions API*, gambar dari *Google StreetView* dan Google VR (*Cardboard*) dalam perangkat lunak *virtual jogging*.

1.4 Batasan Masalah

Karena ada banyak tempat atau pemandangan di dunia ini, hanya beberapa lokasi saja yang dapat dipilih dari yang telah disediakan.

1.5 Metodologi Penelitian

Metodologi penelitian yang akan digunakan adalah sebagai berikut:

- Melakukan studi literatur dari situs-situs web tentang Google VR SDK, *StreetView API*, *Directions*, sensor tentang langkah, baik melalui media tulisan maupun video.
- Menampilkan pemandangan *StreetView* pada *Google Cardboards*.
- Mengintegrasikan *Google Directions API* dengan pemandangan *StreetView* yang telah ditampilkan pada *Google Cardboard*.
- Menganalisis sensor langkah dan menyinkronisasikannya dengan perubahan pemandangan *StreetViews*.

Untuk membuat skripsi ini, peneliti . Setelah mempelajari semua komponen dari aplikasi yang akan dibuat, peneliti akan melakukan implementasi.

1.6 Sistematika Pembahasan

Dokumen dibagi ke dalam beberapa bab dengan sistematika pembahasan sebagai berikut:

1. Bab 1: Pendahuluan, yang menjelaskan gambaran umum penelitian. Mengandung latar belakang, rumusan masalah, tujuan,, batasan masalah, metodologi penelitian, serta sistematika pembahasan.
2. Bab 2: Dasar Teori, berisi landasan dari teori-teori yang berhubungan serta mendukung penelitian. Mengandung Google VR, Google *StreetView API*, Google *Directions API*, dan sensor.
3. Bab 3: Analisis, menjelaskan mengenai proses analisis masalah untuk menemukan solusi.

BAB 2

LANDASAN TEORI

Pada bab ini akan dijelaskan mengenai Google VR, Google StreetView API, Google Directions API, dan *motion sensor*.

2.1 Google VR

Google VR adalah teknologi yang diciptakan perusahaan Google untuk membuat pemandangan dunia maya yang terlihat nyata dengan menempatkan *smartphone* pada alat yang bernama *viewer*. *Viewer* adalah alat untuk melihat dunia VR pada aplikasi VR di *smartphone*. Dua *viewer* yang diciptakan Google adalah *Google Daydream* dan *Cardboard*. Pada penelitian ini, yang digunakan *Google Cardboard*. Google menyediakan sebuah alat bantu bagi pengembang perangkat lunak untuk memudahkan pengembangan aplikasi VR yang disebut Google VR SDK.

2.1.1 Google VR SDK

Google VR SDK adalah alat bantu berlisensi Apache 2.0 yang disediakan Google pada repository Github yang berisi kode program dari aplikasi *virtual reality* (VR) yang tersedia untuk Android (Java), Android NDK, Unity, dan iOS. . Secara umum, SDK ini dibuat agar pengembang perangkat lunak dapat mempelajari serta memanfaatkan teknologi VR yang disediakan Google. Ada beberapa aplikasi yang tersedia pada SDK tersebut seperti aplikasi demo bernama HelloVR dan pemutar video dalam VR, tetapi penulis akan memanfaatkan bagian aplikasi demo HelloVR untuk Android Java pada Google VR SDK. Untuk menggunakan SDK ini, dibutuhkan perangkat lunak Android Studio 2.3.3 dan lebih tinggi, dengan Android SDK versi 7.1.1 (API Level 25) atau lebih tinggi.

Kelas-kelas Google VR Java

Google menyediakan *library* untuk bahasa pemrograman Java pada *package* com.google.vr.sdk.base agar dapat digunakan pengembang perangkat lunak untuk membuat aplikasi VR. Tabel 2.1 menjelaskan mengenai beberapa kelas (atau *interface* yang ada pada package com.google.vr.sdk.base).

Bagian dari Google VR SDK yang akan digunakan pada penelitian ini adalah aplikasi HelloVR yang akan dijelaskan pada Bagian 2.1.2.

2.1.2 Aplikasi HelloVR

HelloVR, aplikasi yang diperoleh dari Google VR SDK pada direktori ./samples/sdk-hellovr adalah sebuah aplikasi demo permainan *treasure hunt*, yaitu sejenis permainan mencari bentuk yang mengapung di dunia VR dengan melihat tepat pada bentuk tersebut dan menyalakan pemicu pada Google Cardboard. Setelah kondisi untuk menangkap bentuk yang ada, bentuk tersebut akan menghilang, lalu bentuk yang lain akan muncul di tempat lain. Gambar 2.1 menunjukkan tampilan UI permainan *treasure hunt* HelloVR.

Tabel 2.1: Beberapa *class* dan *interface* pada *package com.google.vr.sdk.base*

Nama	Tipe	Deskripsi
GvrView.Renderer	<i>interface</i>	merender gambar atau pemandangan <i>stereo</i> oleh dirinya sendiri
GvrView.StereoRenderer	<i>interface</i>	interface <i>renderer</i> yang menyerahkan penyajian gambar atau pemandangan <i>stereo</i> pada <i>view</i>
Distortion	<i>class</i>	Kelas yang mendefinisikan atribut-atribut untuk memperbaiki <i>distorsi</i> dari lensa
Eye	<i>class</i>	Menjelaskan penyajian gambar atau pemandangan <i>stereo</i> pada satu mata
Eye.Type	<i>class</i>	Menjelaskan atribut atau konstan dari sebuah objek kelas <i>Eye</i> yang diacu
GvrActivity	<i>class</i>	Kelas <i>activity</i> yang memudahkan diintegrasikan dengan <i>VR headset</i> dari Google
GvrView	<i>class</i>	<i>view</i> yang mendukung penyajian pemandangan VR
GvrViewerParams	<i>class</i>	kelas yang mendefinisikan parameter fisik dari <i>viewer</i> Google VR

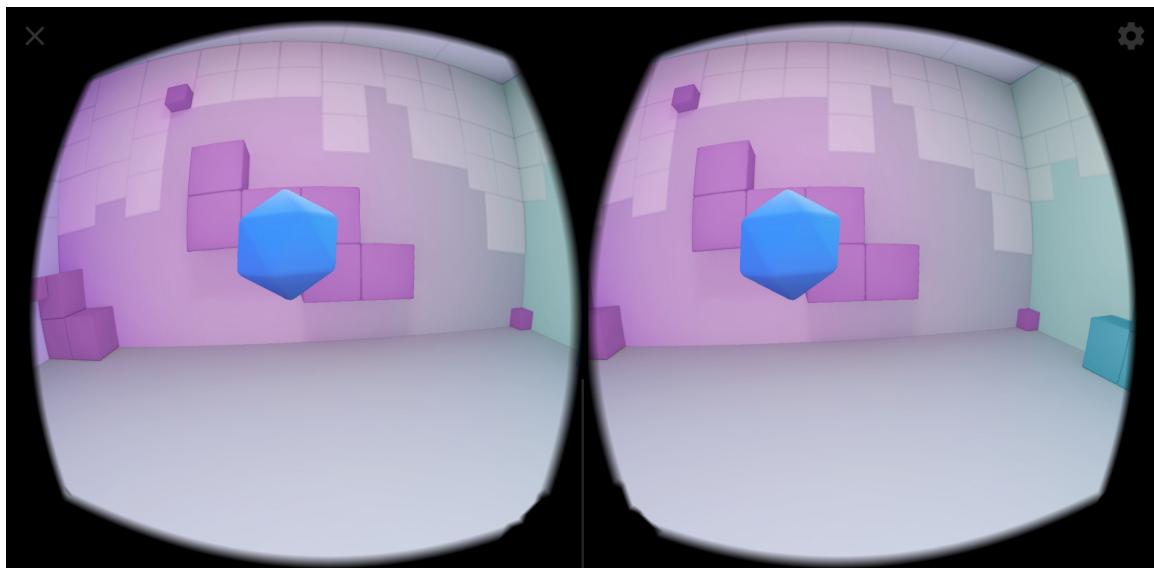
Komponen Aplikasi HelloVR

Dunia VR pada aplikasi ini dibuat dari *file Wavefront Object* (OBJ) dengan tekstur *file Portable Network Graphics* (PNG) (Gambar 2.2) yang telah dengan sangat tepat dipetakan pada *file OBJ* yang ada sehingga dunia VR terlihat sangat nyata. Bentuk-bentuk yang akan dicari pengguna dibuat dari tiga file OBJ yang merepresentasikan tiga macam bentuk yang akan muncul. Masing-masing file OBJ memiliki dua tekstur yang telah dipetakan pada masing-masing file OBJ dalam file PNG. Satu tekstur (Gambar 2.3) digunakan ketika bentuk sedang tidak ada di tengah-tengah titik pengelihatan pengguna, sedangkan satu tekstur yang lain (Gambar 2.4) digunakan ketika pengguna sedang melihat bentuk tepat di titik tengah pengelihatan pengguna.

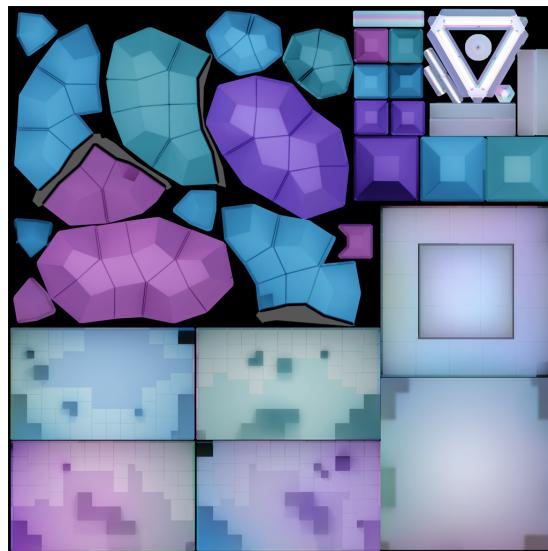
Rancangan kelas Aplikasi HelloVR

Aplikasi HelloVR memiliki empat kelas pada programnya, di antaranya:

- Texture
- Kelas Texture adalah kelas yang memuat tekstur yang akan digunakan.
- TexturedMesh



Gambar 2.1: Tampilan *UI* permainan *treasure hunt* pada aplikasi HelloVR



Gambar 2.2: Gambar *asset* dari tekstur ruangan

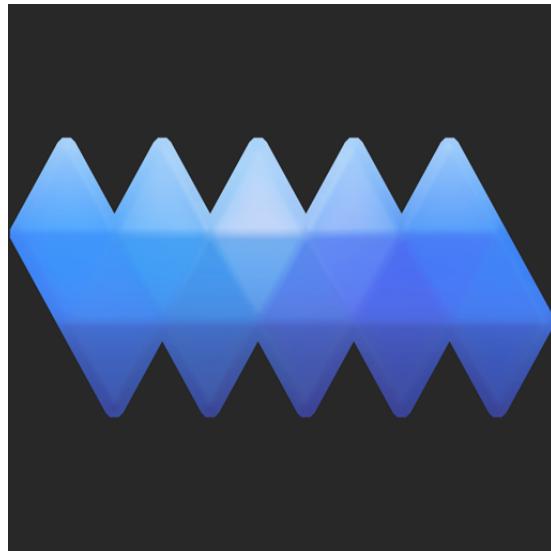
Kelas TexturedMesh adalah sebuah bentuk tiga dimensi yang sudah diberi tekstur sehingga terlihat indah dan berwarna.

- Util

Kelas Util adalah kelas yang digunakan untuk menghitung vektor dan sudut yang dibentuk antara mata pengguna dan bentuk yang akan dicari, serta mengatur pengaturan yang tepat untuk OpenGL, yang adalah *renderer* yang digunakan untuk menggambar bentuk dan ruangan.

- HelloVrActivity

Kelas HelloVrActivity adalah kelas yang adalah kelas *activity* Google VR. Berikut adalah diagram kelas untuk memperjelas hubungan antara semua kelas aplikasi HelloVR. Kelas ini akan menggunakan tiga kelas lainnya untuk mendapat ruangan dan bentuk yang akan digambar, serta keadaan (*state*) dari permainan, seperti sedang menatap pada bentuk atau tidak dan bagian ruangan yang sedang dilihat.



Gambar 2.3: Gambar *asset* dari *texture* bentuk yang sedang tidak di tengah-tengah titik pengelihatan pengguna

2.2 Google StreetView API

Google StreetView API adalah API yang disediakan Google untuk mendapatkan pemandangan sesuai masukan pengguna melalui *HTTP request*. Ada dua jenis *StreetView API* yang disediakan Google, yaitu *static* dan *dynamic*. *StreetView API* yang statis akan menampilkan pemandangan yang tetap tanpa pergerakan pada pemandangannya, sedangkan yang dinamis menampilkan pemandangan yang berubah-ubah seperti *video*. *StreetView API* yang digunakan pada penelitian ini adalah *Static StreetView API*.

2.2.1 API Key

Agar dapat menggunakan *StreetView API* (dan *Directions API* pada Bagian 2.3), ada *API key* yang harus diperoleh pada Google Cloud Platform Console dengan memasukkan nomor kartu kredit. Gambar 2.6 menunjukkan tampilan *Google Cloud* setelah mendapatkan API key. *API key* yang diberikan terdiri atas dua puluh dan delapan belas karakter alfanumerik (bisa huruf kapital dan huruf kecil) yang dihubungkan dengan tanda "-". *API key* yang telah diperoleh akan digunakan sebagai salah satu parameter masukan agar Google API dapat diakses.

2.2.2 Penggunaan StreetView API

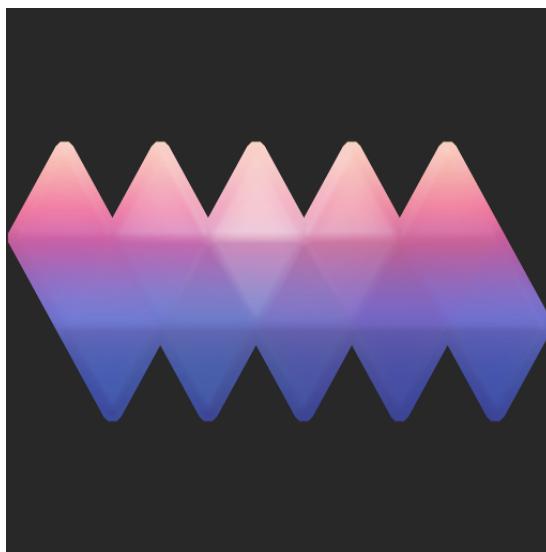
Secara umum, API diakses menggunakan URL Web sebagai berikut:

`https://maps.googleapis.com/maps/api/streetview?parameters`

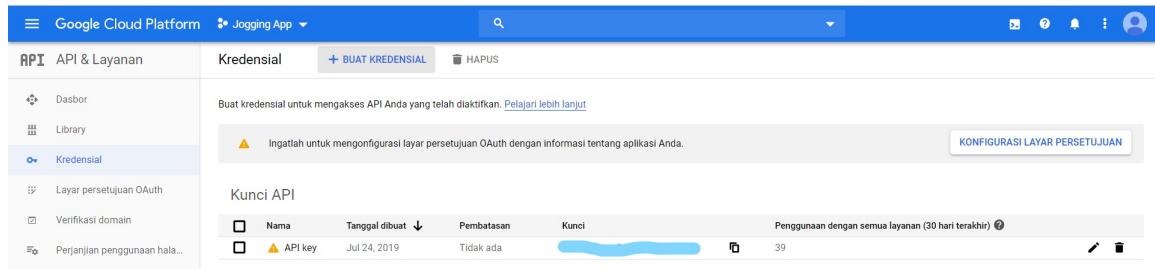
"Parameters" pada URL Web adalah atribut-atribut dengan parameter yang diterima StreetView. Sintaks parameter tersebut adalah:

$$X = Y$$

X adalah atribut dari StreetView, sedangkan Y adalah nilainya, dan nilai tersebut harus sesuai dengan tipe dan rentang nilai masing-masing atribut. Untuk atribut kedua dan seterusnya yang akan dimasukkan dalam parameter (jika ada), dapat diteruskan dengan tanda "&", lalu diikuti dengan pola seperti rumus di atas. Saat mengakses *StreetView API*, ada dua kemungkinan hasil yang diperoleh, yaitu berhasil dan gagal. Pemanggilan *API* yang berhasil akan menghasilkan gambar pemandangan dari lokasi sesuai masukan pengguna, sementara pemanggilan yang gagal menghasilkan sebuah



Gambar 2.4: Gambar *asset* dari *texture* bentuk yang sedang ada di tengah-tengah titik pengelihatan pengguna



Gambar 2.5: Tampilan UI Google Cloud saat mengakses API Key (API Key disamarkan)

gambar dengan penjelasan bahwa gambar tidak tersedia. Gambar 2.6 memperlihatkan pemanggilan *StreetView API* yang berhasil.

2.2.3 Atribut Parameter *StreetView API*

Untuk menampilkan pemandangan yang sesuai keinginan pengguna, beberapa parameter masukan harus ditentukan. Ada dua jenis parameter masukan, di antaranya parameter wajib dan parameter opsional. Pengaksesan atau pemanggilan *StreetView API* yang berhasil akan mengembalikan sebuah gambar pemandangan dari lokasi sesuai parameter masukan.

Parameter Wajib

Parameter wajib adalah parameter yang harus dimasukkan oleh pengguna dan jika tidak dimasukkan akan mengakibatkan pemanggilan yang gagal. Beberapa parameter wajib pada *StreetView API* adalah

- *key*

API key yang dijelaskan pada Bagian 2.2.1.

- *location* atau *pano* (salah satu) Lokasi dari pemandangan yang ingin ditampilkan. *locaction* menerima dua jenis parameter garis lintang dan garis bujur (*longitude* dan *latitude*) atau *String* nama lokasi, sementara *pano* menerima *panorama id* dari lokasi atau panorama.
-



Gambar 2.6: Pemanggilan *StreetView API* yang berhasil

location atau *pano* untuk menentukan lokasi pemandangan yang ingin ditampilkan dan *size* untuk menentukan ukuran gambar pemandangan.

Parameter Opsional

Selain parameter wajib, ada parameter opsional, yaitu parameter yang tidak perlu diisi agar pengaksesan *API* berhasil dan biasanya atribut parameter tersebut sudah memiliki nilai bawaan (*default*). Ada beberapa parameter opsional yang dapat digunakan sebagai parameter untuk mengubah pengaturan dari pemandangan yang diambil:

- *signature*
- *heading*
- *fov (field of view)*
- *pitch*
- *radius*
- *source*.

Tabel 2.2 menjelaskan semua parameter opsional dari *StreetView API*.

2.3 Google Directions API

Google *Directions API* adalah layanan berbasis *HTTP/HTTPS* dari Google yang membantu mencari dan menghitung arah dari satu tempat ke tempat yang lain (sumber). Ada beberapa *mode* dari arah yang dapat dicari seperti *driving*, *transit*, *walking*, dan *cycling*. Pengaksesan *Directions API* sangat mirip dengan *StreetView API*, yaitu membutuhkan *API Key*, seperti yang dijelaskan pada Bagian 2.2.1, sebagai salah satu atribut wajib, juga memiliki atribut wajib dan opsional yang dapat diatur lewat parameter.

Tabel 2.2: Atribut-Atribut Opsional *StreetView API*

Nama Atribut	Tipe	Penjelasan	Rentang Nilai Valid (yang berdampak)
signature	String (alfabetik)	Atribut untuk memastikan bahwa <i>request</i> dikirim dengan <i>API key</i> sesuai jenis <i>signature</i> yang diatur pemilik <i>API key</i> .	-
heading	integer	menyatakan arah pandangan secara horisontal, nilai atribut menyatakan sudut yang dibentuk dari arah utara dengan arah pandang yang diinginkan (sudut yang dibentuk dari arah berlawanan jarum jam)	$0 \leq x \leq 360$
fov (<i>field of view</i>)	integer	menyatakan seberapa perbesaran pemandangan (nilai dalam satuan derajat)	$10 \leq x \leq 120$ (yang berdampak)
pitch	integer	menyatakan pandangan pengguna secara vertikal, satuan nilai dalam derajat.	$-90 \leq x \leq 90$
radius	integer	menyatakan jarak dalam meter, yang adalah titik pengambilan pemandangan	$x > 0$
source	String (alfabetik)	menyatakan pemandangan, <i>default</i> atau <i>outdoor</i>	"default" atau "outdoor"

2.3.1 Penggunaan *Directions API*

Sintaks untuk mengaksesnya pun mirip dengan *StreetView API*, hanya saja ada perbedaan pada bagian "streetview" pada URL Web pada Bagian 2.2.2 diganti dengan "directions". Selain sintaks, *Directions API* juga memiliki dua jenis parameter, wajib dan opsional. Tabel 2.3 menyebutkan dan menjelaskan mengenai atribut-atribut parameter dari *Directions API*.

2.3.2 Hasil Pemanggilan *Directions API*

Hal yang dihasilkan oleh pemanggilan *Directions API* adalah *script Javascript Notation Object (JSON)* yang menyatakan arah sesuai lokasi asal dan tujuan, serta mode perjalanan yang adalah masukan pengguna. Selain arah, *script JSON* yang dikembalikan juga mengandung informasi mengenai jarak jalan yang ditempuh serta waktu tempuh perjalanan. Listing 2.1 menunjukkan contoh *script JSON* dengan masukan lokasi asal "UNPAR" dan lokasi tujuan "Rumah Sakit Santo Borromeus". Pemanggilan yang gagal akan mengembalikan *script JSON* yang menyatakan jalan di antara dua lokasi masukan tidak dapat ditemukan.

Listing 2.1: Hasil Pemanggilan *Directions API* yang Berhasil

```
{
  "geocoded_waypoints" : [
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJbYmcEu7maC4RRijB2oKhHLA",
      "types" : [ "establishment", "point_of_interest", "university" ]
    },
  ]}
```

Tabel 2.3: Atribut-Atribut Opsional *Directions API*

Nama Atribut	Wajib/Opsional	Tipe	Penjelasan	Rentang Nilai <i>Valid</i> (yang berdampak)
origin	Wajib	String (alfabetik)	Atribut yang menyatakan lokasi asal yang dimasukkan pengguna.	String lokasi yang <i>sah</i>
destination	Wajib	String	Atribut yang menyatakan lokasi tujuan yang dimasukkan pengguna	String lokasi yang <i>valid</i>
key	Wajib	String	<i>API Key</i> pengguna yang membuat pengguna dapat mengakses <i>API</i>	String <i>API key</i> yang <i>valid</i> (Bagian 2.2.1)
mode	Opsional	String	menyatakan <i>mode</i> perjalanan yang akan ditempuh.	"driving", "walking", "bicycling", atau "transit"
waypoints	Opsional	integer	menyatakan lokasi yang ingin ditempuh dalam perjalanan	String lokasi yang <i>valid</i>
alternatives	Opsional	<i>bit (boolean)</i>	menyatakan apakah rute yang disediakan <i>Directions API</i> menyediakan beberapa pilihan rute	<i>true</i> atau <i>false</i>
avoid	Opsional	String (alfabetik)	menyatakan jenis-jenis jalan yang harus dihindari, seperti jalan tol, jembatan, dan lain-lain	"tolls", "highways", "ferries, dan/atau "indoor (gunakan " " jika ada beberapa)
language	Opsional	String (alfabetik)	bahasa yang digunakan untuk menyajikan rute perjalanan	Bahasa yang didukung Google
units	Opsional	String (alfabetik)	jenis satuan yang akan digunakan	"metrics" atau "imperial"

```
{
    "geocoder_status" : "OK",
    "place_id" : "ChIJU8k7DlHmaC4RQ2mUo1ERm1k",
    "types" : [ "establishment", "hospital", "point_of_interest" ]
},
"routes" : [
{
    "bounds" : {
        "northeast" : {
            "lat" : -6.8746719,
            "lng" : 107.6137497
        },
        "southwest" : {
            "lat" : -6.89377709999999,
            "lng" : 107.6034922
        }
    }
}
```

```
},
"copyrights" : "Map\u002fdata\u20222020",
"legs" : [
  {
    "distance" : {
      "text" : "3.0\u00a0km",
      "value" : 3042
    },
    "duration" : {
      "text" : "10\u00a0mins",
      "value" : 614
    },
    "end_address" : "Jl.\u002fIr.\u002fH.\u002fJuanda\u2022No.100,\u002fLebakgede . . .",
    "end_location" : {
      "lat" : -6.89377099999999,
      "lng" : 107.613021
    },
    "start_address" : "Jl.\u002fCiumbuleuit\u2022No.94,\u002fHegarmanah . . .",
    "start_location" : {
      "lat" : -6.8746719,
      "lng" : 107.6046127
    },
    "steps" : [
      {
        "distance" : {
          "text" : "1.0\u00a0km",
          "value" : 1008
        },
        "duration" : {
          "text" : "4\u00a0mins",
          "value" : 251
        },
        "end_location" : {
          "lat" : -6.8833328,
          "lng" : 107.6049108
        },
        "html_instructions" : "Head\u002f\u003cb\u003esouth\u003c\u003e . . .",
        "polyline" : {
          "points" : "tuh@yowoSdAP|AL'Cb@dAHHBvC'@l@ . . ."
        },
        "start_location" : {
          "lat" : -6.8746719,
          "lng" : 107.6046127
        },
        "travel_mode" : "DRIVING"
      },
      ...
    ],
    "status" : "OK"
  }
]
```

2.4 Motion Sensor

Motion sensor adalah sensor pada *smartphone* yang mendeteksi pergerakan gawai *smartphone*. Pergerakan yang dapat dideteksi termasuk saat gawai dimiringkan, digoyangkan, diayunkan, atau diputar (sumber). Beberapa contoh *motion sensor* pada *smartphone* adalah:

- *accelerometer*

Sensor yang mendeteksi gerakan *smartphone* terhadap sumbu *x*, *y*, dan *z*, termasuk gaya gravitasi terhadap masing-masing sumbu.

- *gravity sensor*

Sensor yang mendeteksi gaya gravitasi terhadap sumbu *x*, *y*, dan *z*.

- *gyroscope*

Sensor yang mendeteksi putaran gawai terhadap sumbu *x*, *y*, dan *z*.

- *linear acceleration sensor*

Sensor yang mendeteksi pergerakan linier, terhadap sumbu *x*, *y*, dan *z* tanpa gaya gravitasi pada masing-masing sumbu.

- *rotation vector sensor*

Sensor yang mendeteksi vektor putaran pada gawai terhadap sumbu *x*, *y*, dan *z*.

- *significant motion sensor*

- *step counter*

Sensor yang menghitung jumlah langkah saat berjalan yang pengguna gawai ambil.

- *step detector*

Sensor yang men-trigger sebuah *event* saat pengguna mengambil langkah saat berjalan.

Pada penelitian ini, *motion sensor* yang akan digunakan adalah *step detector*. Sintaks untuk menggunakan sensor ini tertera pada Listing 2.2.

Listing 2.2: Sintaks menggunakan sensor *step detector*

```
private SensorManager sensorManager;  
private Sensor sensor;  
...  
sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
sensor = sensorManager.getDefaultSensor(Sensor.TYPE_STEP_DETECTOR);
```

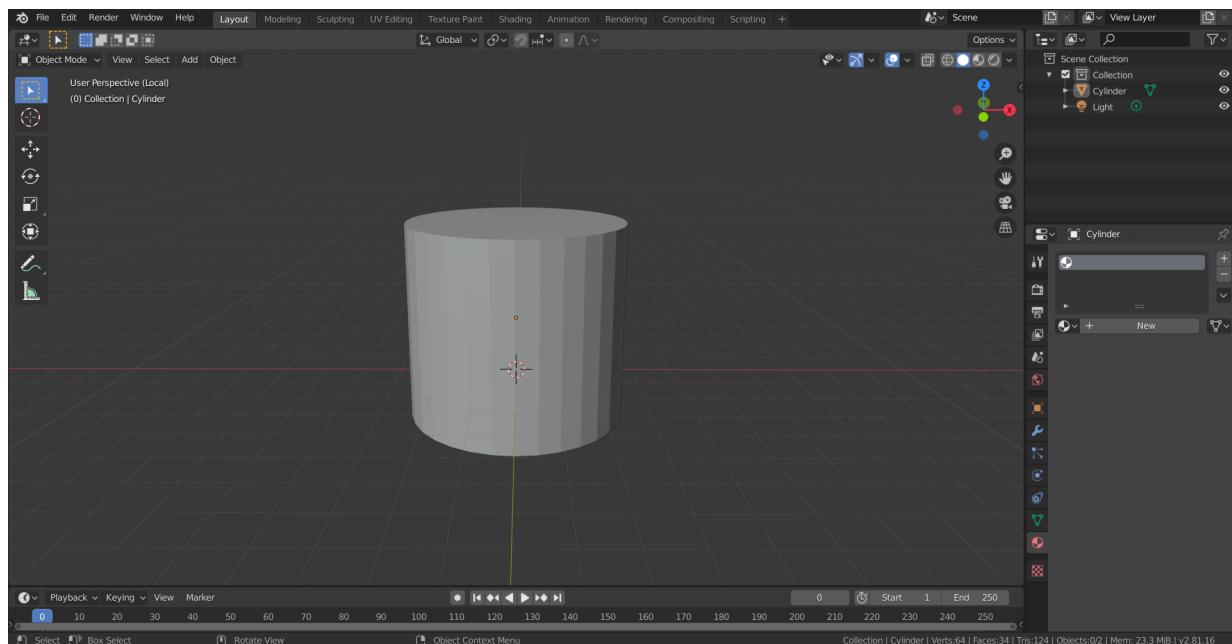
BAB 3

ANALISIS

Pada bab ini dijelaskan mengenai cara memanfaatkan Google VR SDK, *Google StreetView API*, *Google Directions API*, dan cara memanfaatkan sensor *step detector*, termasuk cara mengintegrasikan semua komponen tersebut secara bertahap.

3.1 Dunia VR

Dari komponen-komponen Google VR SDK, bagian *assets*-lah yang dapat dimodifikasi untuk menampilkan pemandangan VR yang ingin ditampilkan. Dunia VR terdiri atas dua *file*, yaitu *file* OBJ sebagai bentuk ruangan dan *file* PNG sebagai tekstur ruangan *file* OBJ. Dunia yang ingin ditampilkan pada dunia VR adalah dunia dengan bentuk silinder agar terlihat seperti dunia nyata, terutama bagian permukaan samping (yang melengkung). Untuk membuat dunia berbentuk silinder, kertas yang digunakan adalah *Blender* versi 2.81. Gambar menunjukkan tampilan *UI Blender* versi 2.81.



Gambar 3.1: Tampilan *UI Blender* Blender versi 2.81

3.2 Menampilkan *StreetView API* pada Dunia VR

Android menyediakan kelas untuk menampilkan langsung pada *fragment* yang disebut *StreetViewPanoramaFragment* (sumber). Namun, hal ini tidak dapat dimanfaatkan karena tidak dapat

dimanfaatkan karena *fragment* tidak dapat ditampilkan dalam pemandangan dunia VR. Alasan inilah yang membuat *Street View API* harus diakses melalui HTTP/HTTPS menggunakan URL.

Dengan menggunakan HTTP/HTTPS URL, gambar yang dihasilkan merupakan gambar persegi panjang saat menghadap satu arah sehingga tidak dapat secara langsung ditampilkan pada dunia VR. Untuk melakukannya, gambar dari empat arah harus digabungkan untuk menghasilkan pemandangan seperti dunia nyata. Hal yang harus dilakukan untuk mencapai hal tersebut adalah menggabungkan gambar-gambar dari atribut *heading* dari empat arah, dengan satu arah yang berlawanan dengan arah yang lain, lalu dua arah lain yang tegak lurus dengan arah pertama. Dengan kata lain, selisih setiap dua nilai *heading* yang berurutan bernilai 90 (misalnya *heading* dengan nilai 0, 90, 180 dan 270). Gambar 3.2 menunjukkan hasil penggabungan empat gambar dengan deskripsi di atas.



Gambar 3.2: Contoh hasil penggabungan empat gambar *Street View*

Setelah mendapatkan gambar *StreetView* yang sudah digabungkan tersebut, gambar tersebut dapat dimanfaatkan sebagai tekstur untuk ruang pada file OBJ yang berbentuk silinder sehingga pemandangan *StreetView* dapat ditampilkan pada dunia VR.

3.3 Integrasi dengan *Directions API*

Untuk mendapatkan rute perjalanan, *Directions API* dapat dimanfaatkan. File yang diperoleh lewat *Directions API* adalah file JSON yang memiliki *key* dan *value*. Ada beberapa atribut (*key*) dengan nilai (*value*) yang ada pada file JSON dari hasil pemanggilan *Directions API* yang dapat dimanfaatkan.

3.3.1 Menentukan Atribut yang Bermanfaat

File JSON yang dihasilkan memiliki beberapa tingkat *key* dan *value*. Pada tingkat pertama, ada tiga *key*: *geocoded_waypoints*, *status* dan *routes*. *Key* yang dapat digunakan adalah *routes* yang menunjukkan jalan yang akan ditempuh. Pada tingkat berikutnya, *key routes* memiliki *value*

seperti *bounds*, *copyrights*, dan *legs*. Jika melihat bagian *legs* yang menampung atribut-atribut jalan yang ditempuh, ada *distance*, *duration*, *end_location*, *html_instructions*, *maneuver*, *polyline*, *start_location*, dan *travel_mode*. Dari beberapa atribut dari *legs*, yang dapat digunakan adalah *end_location* dan *start_location*, yang menunjuk kepada posisi garis lintang dan garis bujur titik ujung dari jalan yang sedang ditempuh.

3.3.2 Cara Memanfaatkan Atribut

Setelah memperoleh nilai *end_location* dan *start_location*, jalur tempuh pada jalan yang sedang ditempuh dapat ditentukan lewat posisi garis lintang dan garis bujur kedua titik ujung jalan. Cara untuk membentuk jalan secara matematis adalah menggunakan selisih antara garis lintang dan garis bujur satu titik ujung jalan ke titik ujung jalan lain.

3.4 Cara Memanfaatkan *Step Detector* Sensor

Untuk membuat animasi atau perubahan pemandangan sesuai rute tempuh yang sudah diperoleh, harus ada perubahan gambar sesuai langkah kaki yang diambil pengguna. Jadi, setiap kali sensor mendapat rangsang, *event* yang dipicu agar terjadi adalah gambar berubah seperti yang dijelaskan pada Bagian 3.3.2, tetapi perubahan gambar *StreetView* yang sesuai *Directions API* harus berubah dengan tahap yang benar agar animasi pemandangan terlihat mulus.