



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - IF184802

**PROTOTIPE APLIKASI *MIXED REALITY*  
UNTUK MENGEKSPLORASI DUNIA NYATA  
MELALUI SUDUT Pandang ORANG KETIGA  
SECARA *REAL-TIME***

PUTRI NURUL APRILLIANDINI  
NRP 0511164000090

Dosen Pembimbing  
Ridho Rahman H., S.Kom., M.Sc.  
Hadziq Fabroyir, S.Kom., Ph.D.

DEPARTEMEN INFORMATIKA  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020









**TUGAS AKHIR - IF184802**

**PROTOTIPE APLIKASI *MIXED REALITY*  
UNTUK MENGEKSPLORASI DUNIA NYATA  
MELALUI SUDUT Pandang ORANG KETIGA  
SECARA *REAL-TIME***

**PUTRI NURUL APRILLIANDINI  
NRP 0511164000090**

**Dosen Pembimbing  
Ridho Rahman H., S.Kom., M.Sc.  
Hadziq Fabroyir, S.Kom., Ph.D.**

**DEPARTEMEN INFORMATIKA  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020**

*[Halaman ini sengaja dikosongkan]*



**UNDERGRADUATE THESIS - IF184802**

**MIXED REALITY APPLICATION PROTOTYPE TO  
EXPLORE THE REAL WORLD IN REAL-TIME  
THROUGH A THIRD-PERSON POINT OF VIEW**

**PUTRI NURUL APRILLIANDINI  
NRP 0511164000090**

**Supervisors**

**Ridho Rahman H., S.Kom., M.Sc.**

**Hadziq Fabroyir, S.Kom., Ph.D.**

**DEPARTMENT OF INFORMATICS**

**Faculty of Information and Communication Technology**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2020**

*[Halaman ini sengaja dikosongkan]*



## LEMBAR PENGESAHAN

# PROTOTYPE APLIKASI *MIXED REALITY* UNTUK MENGEKSPLORASI DUNIA NYATA MELALUI SUDUT PANDANG ORANG KETIGA SECARA *REAL-TIME*

## TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Interaksi, Grafika, dan Seni  
Program Studi S-1 Departemen Informatika  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**PUTRI NURUL APRILLIANDINI**  
**NRP: 0511164000090**

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Ridho Rahman H., S.Kom., M.Sc.  
NIP. 198702132014041001   
(Pembimbing 1)
2. Hadziq Fabroyir, S.Kom., Ph.D.  
NIP. 198602272019031006   
(Pembimbing 2)

**SURABAYA**  
**JUNI 2020**

*[Halaman ini sengaja dikosongkan]*

# **PROTOTYPE APLIKASI *MIXED REALITY* UNTUK MENGEKSPLORASI DUNIA NYATA MELALUI SUDUT PANDANG ORANG KETIGA SECARA *REAL-TIME***

Nama Mahasiswa : Putri Nurul Aprilliandini  
NRP : 0511164000090  
Jurusan : Departemen Informatika FTIK-ITS  
Dosen Pembimbing 1 : Ridho Rahman H., S.Kom., M.Sc.  
Dosen Pembimbing 2 : Hadziq Fabroyir, S.Kom., Ph.D.

## **ABSTRAK**

*Mixed reality adalah teknologi yang menggabungkan benda maya ke dalam sebuah lingkungan nyata dan citra dari benda nyata ke dalam lingkungan maya secara bersamaan dengan cara memproyeksikan benda-benda tersebut dalam waktu nyata. Secara sederhana, mixed reality menggabungkan teknik virtual reality dan augmented reality secara bersamaan. Pada dasarnya, mixed reality merupakan bagian dari sebuah kontinum yang terbentang dari lingkungan yang benar-benar nyata sampai lingkungan yang sepenuhnya maya. Mixed reality berada di tengah-tengah kontinum tersebut.*

*Seiring dengan berkembangnya teknologi, semakin berkembang pula cara untuk mengamati perilaku manusia, contohnya seperti interaksi manusia dengan lingkungan spasial di sekitarnya. Salah satu interaksi yang dapat diamati adalah dengan mengaugmentasi realitas yang dialami serta dengan memproyeksikan diri kita sehingga kita dapat melihat lingkungan sekitar melalui sudut pandang orang ketiga dengan proyeksi little planet. Namun, saat ini masih belum ada aplikasi yang memungkinkan hal tersebut.*

*Untuk mengatasi masalah tersebut, maka sebuah prototipe aplikasi dibuat pada Tugas Akhir ini untuk menyimulasikan pengalaman realitas lingkungan sekitar melalui sudut pandang*

*orang pertama dan melalui sudut pandang orang ketiga dengan proyeksi little planet. Prototipe aplikasi ini dikembangkan melalui Unity karena Unity merupakan engine yang sangat fleksibel serta sangat mendukung pengembangan AR/VR.*

*Dengan adanya prototipe aplikasi ini, pengguna dapat mengeksplorasi lingkungan di sekitarnya dengan fitur mixed reality dan juga dengan proyeksi little planet.*

***Kata kunci: augmented reality, mixed reality, sudut pandang orang pertama, little planet, interaksi manusia***

# **MIXED REALITY APPLICATION PROTOTYPE TO EXPLORE THE REAL WORLD IN REAL-TIME THROUGH A THIRD-PERSON POINT OF VIEW**

Name : Putri Nurul Aprilliandini  
NRP : 05111640000090  
Major : Informatics Department FTIK-ITS  
Supervisor I : Ridho Rahman H., S.Kom., M.Sc.  
Supervisor II : Hadziq Fabroyir, S.Kom., Ph.D.

## **ABSTRACT**

*Mixed reality is a technology that combines virtual objects into a real environment and images of real objects into a virtual environment simultaneously by projecting these objects in real-time. Simply put, mixed reality combines virtual reality and augmented reality techniques together. Basically, mixed reality is part of a continuum that extends from a truly real environment to a fully virtual environment. Mixed reality is in the middle of the continuum.*

*As technology develops, ways to observe human behavior also develop, for example human interactions with their surrounding spatial environment. One interaction that can be observed is to augment the reality experienced and by projecting ourselves so that we can see the surrounding environment through a third person point of view with the little planet projection. However, there are currently no applications that allow this.*

*To overcome this problem, an application prototype was made in this Final Project to simulate the experience of exploring an environment through a first-person perspective and through a third-person perspective with a little planet projection. This application prototype was developed through*

*Unity because Unity is a very flexible engine and strongly supports the development of AR/VR.*

*With this prototype, users can explore the surrounding environment with mixed reality features and also through the little planet projection.*

***Keywords: augmented reality, mixed reality, first person point of view, little planet, human interaction***

## **KATA PENGANTAR**

Puji syukur kepada Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

### **PROTOTYPE APLIKASI *MIXED REALITY* UNTUK MENGEKSPLORASI DUNIA NYATA MELALUI SUDUT PANDANG ORANG KETIGA SECARA *REAL-TIME***

Melalui lembar ini, penulis ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Allah SWT atas segala nikmat dan rahmat yang telah diberikan selama ini serta senantiasa menemani dan memberi kemudahan kepada penulis dalam menyelesaikan Tugas Akhir.
2. Kedua orang tua Bapak Moch. Rizal Chaidir dan Ibu Nurmasita tersayang yang telah memberikan doa dan dukungan selama ini.
3. Bapak Ridho Rahman H., S.Kom., M.Sc. selaku dosen pembimbing I yang membimbing penulis selama pengerjaan Tugas Akhir.
4. Bapak Hadziq Fabroyir, S.Kom., Ph.D. selaku dosen pembimbing II yang senantiasa memberikan arahan dan bantuan sehingga penulis dapat menyelesaikan Tugas Akhir ini.
5. Bapak dan Ibu dosen Teknik Informatika ITS yang telah membina dan memberikan ilmu kepada penulis selama menempuh studi di Teknik Informatika ITS.
6. Izzan yang selama tiga setengah tahun terakhir, dengan segala naik-turun yang terjadi, selalu ada untuk memberikan dukungan kepada penulis.
7. Mas Angga sebagai satu-satunya teman yang penulis dapat

secara lepas ajak bicara tentang *gacha*, *anime*, dan hal-hal *wibu* lainnya sehingga kehidupan penulis selama di Surabaya jauh lebih menyenangkan.

8. Teman-teman dekat di Bandung; Irin, Adrie, Reyza, Girin, Shaumil, Rama, Fadli, Tila, dan Zhafran yang senantiasa memberikan dukungan kepada penulis selama pengerjaan Tugas Akhir.
9. Teman-teman angkatan 2016 yang telah memberikan motivasi selama penulis berkuliah di Informatika ITS.
10. Serta pihak lain yang namanya tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari sepenuhnya bahwa Tugas Akhir ini masih memiliki kekurangan. Oleh karena itu, dengan tangan terbuka, penulis menerima segala saran dan kritik dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2020

Putri Nurul Aprilliandini



## DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL.....	xxi
DAFTAR KODE SUMBER.....	xxiii
1 BAB I PENDAHULUAN.....	33
1.1 Latar Belakang.....	33
1.2 Rumusan Masalah.....	34
1.3 Batasan Masalah.....	34
1.4 Tujuan.....	35
1.5 Manfaat.....	35
1.6 Metodologi Pembuatan Tugas Akhir.....	36
1.7 Sistematika Penulisan.....	37
2 BAB II TINJAUAN PUSTAKA.....	39
2.1 <i>Mixed Reality</i> .....	39
2.2 Kamera 360°.....	39
2.3 Unity.....	40
2.3.1 Hierarki <i>File</i> .....	40
2.3.2 <i>Runtime C#</i> .....	42
2.3.3 <i>Renderer</i> .....	43
2.4 Proyeksi Stereografi.....	44
2.4.1 <i>Little Planet</i> .....	46
2.5 Oculus Rift.....	46
2.6 Vuuforia.....	46
2.7 <i>Spatial Cognition</i> .....	46
3 BAB III ANALISIS DAN PERANCANGAN SISTEM....	47
3.1 Analisis Metode Secara Umum.....	47
3.1.1 Analisis Permasalahan.....	47
3.1.2 Deskripsi Umum Sistem.....	48

3.1.3	Spesifikasi Kebutuhan Perangkat Lunak.....	48
3.2	Perancangan .....	54
3.2.1	Perancangan Perangkat Lunak.....	54
3.2.2	Perancangan Antarmuka Pengguna .....	63
4	BAB IV IMPLEMENTASI.....	65
4.1	Lingkungan Implementasi Perangkat Lunak .....	65
4.2	Implementasi Antarmuka Pengguna.....	65
4.2.1	Implementasi Antarmuka Menu Utama.....	66
4.2.2	Implementasi Antarmuka Mode Sudut Pandang Orang Pertama .....	66
4.2.3	Implementasi Antarmuka Mode Sudut Pandang Orang Ketiga.....	67
4.3	Implementasi Fitur Aplikasi .....	68
4.3.1	Implementasi Menu Utama .....	69
4.3.2	Implementasi Mode Sudut Pandang Orang Pertama .....	69
4.3.3	Implementasi Mode Sudut Pandang Orang Ketiga.....	70
5	BAB V PENGUJIAN DAN EVALUASI .....	75
5.1	Lingkungan Pengujian.....	75
5.2	Uji Performa .....	75
5.3	Preferensi terhadap Mode Sudut Pandang Orang Pertama dan ketiga.....	77
6	BAB VI KESIMPULAN DAN SARAN.....	79
6.1	Kesimpulan.....	79
6.2	Saran.....	81
7	DAFTAR PUSTAKA .....	83
8	LAMPIRAN A.....	89
9	LAMPIRAN B.....	90
10	LAMPIRAN C.....	93
11	LAMPIRAN D.....	98
12	LAMPIRAN E.....	102
13	LAMPIRAN F .....	105
14	LAMPIRAN G.....	108
15	LAMPIRAN H.....	110
16	LAMPIRAN I.....	112
17	LAMPIRAN J.....	114

18	LAMPIRAN K.....	116
19	LAMPIRAN L.....	120
20	LAMPIRAN M.....	122
21	LAMPIRAN N.....	124
22	LAMPIRAN O.....	126
23	BIODATA PENULIS .....	127

*[Halaman ini sengaja dikosongkan]*

## DAFTAR GAMBAR

Gambar 2.1 Bidang Pandang Kamera 360° Dua Lensa.....	40
Gambar 2.2 Hierarki <i>File</i> dalam Sebuah Proyek Unity .....	41
Gambar 2.3 <i>Default Template</i> untuk Sebuah <i>Controller Script</i> ..	42
Gambar 2.4 Perbandingan Sebuah Bola dengan Hasil Proyeksinya .....	45
Gambar 3.1 Arsitektur Sistem .....	48
Gambar 3.2 Diagram Kasus Penggunaan.....	50
Gambar 3.3 Diagram Aktivitas UUC-001.....	52
Gambar 3.4 Diagram Aktivitas UUC-002.....	53
Gambar 3.5 Diagram Aktivitas UUC-003.....	54
Gambar 3.6 Skenario Mode Sudut Pandang Orang Pertama.....	56
Gambar 3.7 Skenario Mode Sudut Pandang Orang Ketiga .....	58
Gambar 3.8 Gambar yang telah Diaplikasikan <i>Filter Polar Coordinates</i> .....	60
Gambar 3.9 Rancangan Antarmuka Menu Utama.....	62
Gambar 3.10 Rancangan Antarmuka Mode Sudut Pandang Orang Pertama .....	63
Gambar 3.11 Rancangan Antarmuka Mode Sudut Pandang Orang Ketiga.....	64
Gambar 4.1 Implementasi Antarmuka Menu Utama .....	66
Gambar 4.2 Implementasi Antarmuka Mode Sudut Pandang Orang Pertama .....	67
Gambar 4.3 Implementasi Antarmuka Mode Sudut Pandang Orang Ketiga.....	68
Gambar 4.4 <i>Node Polar Coordinates, Rotate, dan Sample Texture</i> 2D pada <i>Shader</i> .....	71
Gambar 4.5 <i>Node</i> yang Digunakan untuk Mengaplikasikan <i>Mask</i> .....	72
Gambar 4.6 Implementasi <i>Shader Little Planet</i> .....	72

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 2.1 Tipe <i>File</i> pada Unity.....	41
Tabel 2.2 Tipe <i>Shader</i> pada Unity .....	44
Tabel 3.1 Kebutuhan Fungsional.....	49
Tabel 3.2 Kualitas Perangkat Lunak .....	49
Tabel 3.3 Kasus Penggunaan.....	50
Tabel 3.4 Spesifikasi Kasus Penggunaan UC-001.....	51
Tabel 3.5 Spesifikasi Kasus Penggunaan UC-002.....	52
Tabel 3.6 Spesifikasi Kasus Penggunaan UC-003.....	53
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak .....	65
Tabel 5.1 Lingkungan Pengujian Latensi Perangkat Lunak .....	75
Tabel 5.2 Hasil Uji Performa Mode Sudut Pandang Orang Pertama .....	76
Tabel 5.3 Hasil Uji Performa Mode <i>Little Planet</i> .....	76
Tabel 8.1 Penjelasan Kode Sumber 8-1 .....	89
Tabel 9.1 Penjelasan Kode Sumber 9-1 .....	92
Tabel 10.1 Penjelasan Kode Sumber 10-1 .....	97
Tabel 11.1 Penjelasan Kode Sumber 11-1 .....	100
Tabel 12.1 Penjelasan Kode Sumber 12-1 .....	104
Tabel 13.1 Penjelasan Kode Sumber 13-1 .....	107
Tabel 14.1 Penjelasan Kode Sumber 14-1 .....	109
Tabel 15.1 Penjelasan Kode Sumber 15-1 .....	111
Tabel 16.1 Penjelasan Kode Sumber 16-1 .....	113
Tabel 17.1 Penjelasan Kode Sumber 17-1 .....	115
Tabel 18.1 Penjelasan Kode Sumber 18-1 .....	119
Tabel 19.1 Penjelasan Kode Sumber 19-1 .....	121
Tabel 20.1 Penjelasan Kode Sumber 20-1 .....	123
Tabel 21.1 Penjelasan Kode Sumber 21-1 .....	125
Tabel 22.1 Kuisisioner Terhadap Pengguna .....	126

*[Halaman ini sengaja dikosongkan]*



## DAFTAR KODE SUMBER

Kode Sumber 8-1 Implementasi Menu Utama .....	89
Kode Sumber 9-1 Implementasi VR <i>Input</i> .....	90
Kode Sumber 10-1 Implementasi VR <i>Eye Raycaster</i> .....	93
Kode Sumber 11-1 Implementasi <i>Reticle</i> .....	98
Kode Sumber 12-1 Implementasi VR <i>Interactive Item</i> .....	102
Kode Sumber 13-1 Implementasi <i>Example Interactive Item</i> .....	105
Kode Sumber 14-1 Implementasi Menunjukkan Jam (Text) ....	108
Kode Sumber 15-1 Implementasi Menunjukkan Hari (Text) .....	110
Kode Sumber 16-1 Implementasi Menunjukkan Tanggal (Text) .....	112
Kode Sumber 17-1 Implementasi <i>Counter</i> “Objects Collected” .....	114
Kode Sumber 18-1 Implementasi <i>Web Camera Manager</i> .....	116
Kode Sumber 19-1 Implementasi Menunjukkan Jam (TextMesh) .....	120
Kode Sumber 20-1 Implementasi Menunjukkan Hari (TextMesh) .....	122
Kode Sumber 21-1 Implementasi Menunjukkan Tanggal (TextMesh) .....	124

*[Halaman ini sengaja dikosongkan]*

## **BAB I**

### **PENDAHULUAN**

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

#### **1.1 Latar Belakang**

Manusia pada dasarnya adalah makhluk sosial yang selalu berinteraksi dengan lingkungan sekitarnya. Hampir semua aktivitas manusia membutuhkan interaksi dengan lingkungan spasial, yang membuat kemampuan spasial sesuatu yang sangat penting untuk menjalankan kehidupan sehari-hari. Dalam berinteraksi, tiap manusia mempunyai reaksi dan kesadaran yang berbeda terhadap lingkungannya. Reaksi dan kesadaran tersebut pada dasarnya merupakan bagian dari *spatial cognition*. *Spatial cognition* adalah bagaimana manusia memproses, memanipulasi, mengintegrasikan, mendapatkan, dan memanfaatkan informasi spasial, yaitu informasi mengenai unsur-unsur yang merupakan bagian dari ruang dua atau tiga dimensi. [1]

Seiring berjalannya waktu, perkembangan teknologi semakin berkembang dengan pesat. Dengan perkembangan teknologi, berkembanglah juga teknologi untuk mengamati manusia yang memungkinkan kita untuk mengamati interaksi manusia dengan lingkungan spasial di sekitarnya. Salah satu interaksi yang dapat dilakukan untuk mengamati hal tersebut adalah ketika kita mengaugmentasi realitas yang kita alami serta dengan memproyeksikan diri kita sehingga kita dapat melihat lingkungan kita melalui sudut pandang orang ketiga dalam *little planet view*. *Little planet view* digunakan agar kita dapat melihat keseluruhan dari lingkungan di sekitar kita pada saat yang bersamaan. Hal tersebut dapat dicapai dengan menggunakan teknologi *mixed reality* dan *omnidirectional camera* atau yang biasa disebut sebagai kamera 360°.

*Mixed reality* adalah teknologi yang menggabungkan benda maya ke dalam sebuah lingkungan nyata dan citra dari benda nyata ke dalam lingkungan maya secara bersamaan dengan cara memproyeksikan benda-benda tersebut dalam waktu nyata. Secara sederhana, *mixed reality* menggabungkan teknik *virtual reality* dan *augmented reality* secara bersamaan. Pada dasarnya, *mixed reality* merupakan bagian dari sebuah kontinum yang terbentang dari lingkungan yang benar-benar nyata sampai lingkungan yang sepenuhnya maya. *Mixed reality* berada di tengah-tengah kontinum tersebut.

Kamera 360° adalah kamera yang dapat memiliki bidang pandang yang mencakup seluruh *sphere* atau setidaknya selingkaran penuh pada bidang horizontal. Umumnya, kamera 360° mempunyai dua atau lebih lensa *fisheye* dengan jangkauan 180° untuk masing-masing lensa. Setiap lensa tersinkronisasi untuk mengambil dan menggabungkan gambar secara bersamaan.

Dengan memanfaatkan teknologi *mixed reality* dan kamera 360°, diharapkan kita dapat mengalami realitas dengan cara yang baru dan mengamati reaksi seseorang melihat dirinya sendiri dan lingkungannya melalui sudut pandang orang ketiga.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana cara menyalurkan *live footage* dari kamera secara langsung ke VR *headset*?
2. Bagaimana cara menampilkan objek virtual tiga dimensi di dunia nyata melalui *augmented reality*?
3. Bagaimana implementasi komponen antarmuka pada aplikasi?
4. Bagaimana cara melakukan *post-processing video equirectangular* menjadi *little planet*?

## 1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki batasan, yaitu:

1. Perangkat keras yang akan digunakan pada Tugas Akhir ini adalah Oculus Rift, kamera 360°, dan *action cam*.
2. Unity 3D akan digunakan sebagai *graphic renderer* serta untuk membuat pemodelan objek virtual tiga dimensi.
3. Dalam Tugas Akhir ini, pengguna dapat menjelajahi lingkungan di sekitarnya dengan dilengkapi *augmented reality* atau melalui sudut pandang *little planet* selama mengenakan sistem ini.
4. Pada mode *first person*, terdapat fitur di mana pengguna dapat mengambil objek virtual. Fitur tersebut tidak diimplementasikan pada mode *little planet* karena Vuforia tidak mendukung penggunaan kamera 360°.

## 1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah untuk menyimulasikan pengalaman realita lingkungan di sekitar kita melalui sudut pandang orang pertama yang dilengkapi dengan *augmented reality* serta melalui sudut pandang orang ketiga dengan proyeksi *little planet*.

## 1.5 Manfaat

Manfaat yang dapat diperoleh dari implementasi Tugas Akhir ini antara lain:

1. Untuk mengalami realitas dengan informasi tambahan yang jika diimplementasikan lebih lanjut diharapkan dapat bermanfaat dalam kehidupan sehari-hari. Pada dunia nyata, terdapat keterbatasan dalam mendapatkan informasi yang dibutuhkan manusia untuk menjelajahi lingkungannya. Secara fisik, lebar pandangan manusia adalah 120°. [2] Akan tetapi, dengan fitur ini, lebar pandangan manusia dapat menjadi 360°. Hal tersebut memungkinkan manusia untuk melihat apa yang secara normal tidak dapat dilihat dalam kondisi kehidupan nyata. Contohnya adalah dengan

mengimplementasi sudut pandang orang ketiga untuk kasus kemudi kendaraan di mana seseorang dapat melihat kondisi jalan di sekitarnya dengan pandangan yang lebih luas, sehingga dapat menghindari macet.

2. Diharapkan dapat menjadi dasar untuk riset tentang interaksi manusia dengan lingkungan spasialnya, terutama tentang reaksi manusia ketika mengeksplorasi lingkungan dengan sudut pandang orang ketiga.

## 1.6 Metodologi Pembuatan Tugas Akhir

Adapun beberapa tahap dalam proses pengerjaan Tugas Akhir ini, yaitu:

1. Penyusunan proposal Tugas Akhir

Tahap pertama dalam proses pengerjaan Tugas Akhir ini adalah menyusun proposal Tugas Akhir. Pada proposal Tugas Akhir ini diajukan penggunaan *software development kit* Vuforia serta penggunaan *game engine* Unity untuk mengembangkan aplikasi.

2. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi yaitu mengenai penggunaan kamera 360° dalam Unity, *little planet*, *post-processing*, serta *mixed reality*.

3. Analisis dan desain perangkat lunak

Tahap ini meliputi perumusan kebutuhan fungsional, kebutuhan nonfungsional, kasus penggunaan, serta rancangan antarmuka pengguna.

4. Implementasi

Implementasi perangkat lunak yang dipakai menggunakan Unity yang ditulis menggunakan bahasa pemrograman C#.

5. Uji coba dan evaluasi

Pengujian akan dilakukan dengan menguji fungsionalitas dari aplikasi yang ada.

## 6. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini. Pada tahap ini juga disertakan hasil dari implementasi perangkat lunak yang telah dibuat. Sistematika penulisan buku Tugas Akhir ini secara garis besar antara lain:

1. Pendahuluan
  - a. Latar Belakang
  - b. Rumusan Masalah
  - c. Batasan Masalah
  - d. Tujuan
  - e. Manfaat
  - f. Metodologi Pembuatan Tugas Akhir
  - g. Sistematika Penulisan
2. Tinjauan Pustaka
3. Analisis dan Perancangan Sistem
4. Implementasi
5. Pengujian dan Evaluasi
6. Kesimpulan dan Saran
7. Daftar Pustaka

### **1.7 Sistematika Penulisan**

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

#### **Bab I Pendahuluan**

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

#### **Bab II Tinjauan Pustaka**

Memaparkan dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

### **Bab III Analisis dan Perancangan Sistem**

Membahas tentang analisis permasalahan, deskripsi umum sistem, spesifikasi kebutuhan perangkat lunak, lingkungan perancangan, perancangan sistem, dan struktur data.

### **Bab IV Implementasi**

Bab ini berisi implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode sumber yang digunakan untuk proses implementasi.

### **Bab V Pengujian dan Evaluasi**

Bab ini membahas kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

### **Kesimpulan dan Saran**

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan pada Tugas Akhir ini. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

### **Daftar Pustaka**

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.



## **BAB II**

### **TINJAUAN PUSTAKA**

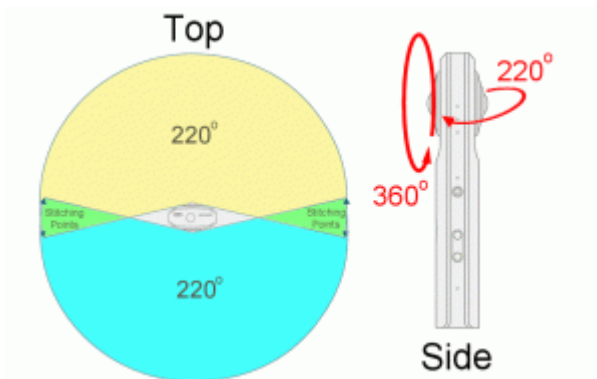
Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir ini. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap perangkat lunak yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

#### **2.1 *Mixed Reality***

*Mixed reality* (MR) adalah penggabungan dunia nyata dan virtual untuk menghasilkan lingkungan yang baru, di mana objek nyata dan maya berinteraksi dalam waktu nyata. Pada dasarnya, *mixed reality* menggabungkan teknik *virtual reality* dan *augmented reality* secara bersamaan. *Mixed reality* merupakan bagian dari sebuah kontinum yang terbentang dari lingkungan yang benar-benar nyata sampai lingkungan yang sepenuhnya virtual. *Mixed reality* berada di tengah-tengah kontinum tersebut. [2]

#### **2.2 Kamera 360°**

Kamera 360° adalah kamera dengan kemampuan untuk menangkap bidang pandang sebesar 360° di bidang horizontal dan memiliki kemampuan untuk menangkap cahaya yang jatuh dari semua arah pada titik fokus. [3] Pada dasarnya, kamera 360° menangkap dua *file* gambar atau video dari lensa ganda dengan bidang pandang 180° dan secara otomatis “dijahit” menjadi satu di dalam kamera. [4] Bidang pandang dari sebuah kamera 360° dua lensa akan diilustrasikan pada Gambar 2.1.



**Gambar 2.1 Bidang Pandang Kamera 360° Dua Lensa [6]**

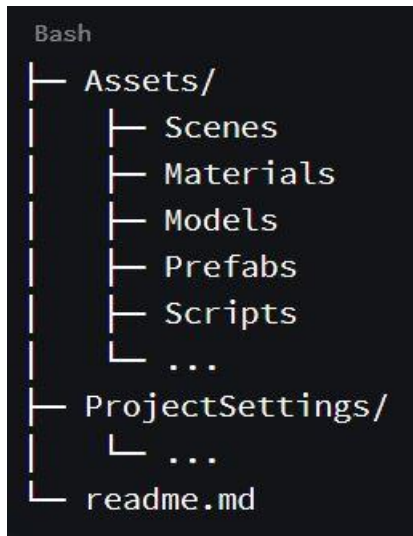
Kamera 360° dua lensa bekerja dengan menggunakan dua kamera dengan sudut 220° yang dipasang berlawanan dengan satu sama lain. Pada bagian yang tumpang tindih, gambar yang ditangkap oleh kedua kamera “dijahit” agar menampilkan gambar 360° yang utuh. [5]

## 2.3 Unity

Unity adalah *game engine* lintas-*platform* yang dikembangkan oleh Unity Technologies, yang pertama kali dirilis pada Juni 2005. *Engine* ini dapat digunakan untuk membuat game tiga dimensi, dua dimensi, *virtual reality*, *augmented reality*, serta simulasi lainnya. [6] Berikut akan dibahas arsitektur Unity dan bagaimana *engine* tersebut bekerja.

### 2.3.1 Hierarki *File*

Dalam Unity, *file-file* dari sebuah proyek biasanya diorganisasi berdasarkan tipe. Gambar 2.2 akan menggambarkan hierarki *file* yang digunakan dalam sebuah proyek Unity.



**Gambar 2.2** Hierarki *File* dalam Sebuah Proyek Unity [8]

Tabel 2.1 mendeskripsikan tipe-tipe *file* dalam sebuah proyek Unity.

**Tabel 2.1** Tipe *File* pada Unity

No	Tipe <i>File</i>	Deskripsi
1.	<i>Scenes</i>	Berisi lingkungan dan menu yang digunakan dalam sebuah proyek. [9]
2.	<i>Materials</i>	Menentukan bagaimana sebuah permukaan di-render, dengan mereferensikan tekstur, <i>tiling</i> , serta <i>color tint</i> yang digunakan. [10]
3.	<i>Models</i>	Sebuah file model dapat berisi model tiga dimensi dan juga data animasi yang dapat digunakan untuk menganimasikan sebuah model. [11]
4.	<i>Prefabs</i>	Memungkinkan pengguna untuk membuat, mengonfigurasi, dan menyimpan sebuah <i>GameObject</i> lengkap dengan semua komponennya, nilai-nilai

No	Type File	Deskripsi
		propertinya, serta GameObject anak sebagai sebuah aset yang dapat digunakan secara berkali-kali. [12]
5.	<i>Scripts</i>	Kode yang memungkinkan pengguna untuk membuat komponen, memicu <i>game event</i> , memodifikasi properti komponen, dan menanggapi input pengguna. [13]

### 2.3.2 Runtime C#

Script dalam Unity ditulis menggunakan C#. Unity memungkinkan pengguna untuk mengontrol *engine* melalui *controller script* dengan mengembangkan kelas *MonoBehaviour*. [8] Gambar 2.3 menunjukkan *default template* untuk sebuah *controller script*.

```
using UnityEngine;

// Classes/Variables can be decorated
[RequireComponent(typeof(Renderer))]
public class MyController : MonoBehaviour
{
    void Start()
    {
        // Start is called when the scene and all object have been initialized.
        // Grab references to other Components
        // Or references to other GameObjects
    }

    void Update()
    {
        // Update is called every frame
    }

    /// <summary>
    /// You can write documentation in XML.
    /// </summary>
    /// <param name="res">You can even refer to parameters.</param>
    /// <returns>And Return Types!</returns>
    void _PrivateFunction(int i)
    {
        // ...
    }
}
```

**Gambar 2.3** *Default Template* untuk Sebuah *Controller Script* [8]

Terdapat dua fungsi utama dalam sebuah *controller script*, yaitu `void Start()` serta `void Update()`. `void Start()` dipanggil ketika sebuah *scene* serta obyek-obyek di dalamnya telah diinisialisasi dan umumnya juga digunakan untuk mereferensikan komponen atau `GameObject` lain, sedangkan `void Update()` merupakan fungsi yang dipanggil setiap *frame*. Pengguna juga dapat membuat fungsi *custom* lainnya. [8]

### **2.3.3 Renderer**

*Renderer* berfungsi untuk membuat objek muncul di layar. Pengguna dapat mengakses *renderer* dari masing-masing objek, *mesh* atau sistem partikel. *Renderer* dapat dinonaktifkan untuk membuat objek tidak terlihat. Untuk *render* sebuah gambar, Unity menggunakan *mesh*, material, *shader*, dan tekstur. [14]

#### **2.3.3.1 Mesh**

*Mesh* adalah susunan struktural model 3D yang terdiri dari poligon. Sebuah *mesh* memiliki titik-titik referensi dalam sumbu X, Y, dan Z untuk menentukan tinggi, lebar, dan kedalaman dari *mesh* tersebut. Poligon yang digunakan untuk membentuk sebuah *mesh* biasanya berbentuk segi empat atau segitiga. [15]

#### **2.3.3.2 Material**

Material adalah *file* yang berisi informasi tentang bagaimana suatu objek di-*render*. Material memiliki peran penting dalam menentukan bagaimana sebuah objek ditampilkan. Material menampilkan properti-properti yang sudah ditentukan oleh sebuah *shader*. [16]

#### **2.3.3.3 Shader**

Pada dasarnya, *shader* adalah program singkat yang *render graphics data*. Fungsi dari sebuah *shader* adalah mengambil *mesh*, tekstur, dan lain-lain sebagai *input* dan menghasilkan gambar sebagai *output*. [17]

Tabel 2.2 mendeskripsikan tipe-tipe *shader* dalam sebuah proyek Unity.

**Tabel 2.2 Tipe Shader pada Unity**

No	Tipe File	Deskripsi
1.	<i>Surface shader</i>	Biasanya digunakan untuk memanipulasi bagaimana cahaya dan bayangan di-render. [18]
2.	<i>Vertex/fragment shader</i>	Digunakan jika <i>shader</i> tidak berinteraksi dengan pencahayaan. Biasanya digunakan untuk efek-efek kompleks yang tidak bisa ditangani oleh <i>surface shader</i> . [18]
3.	<i>Fixed function shader</i>	Digunakan untuk membuat efek-efek yang sederhana. [18]

### 2.3.3.3.1 Shader Graph

*Shader graph* memungkinkan pengguna untuk membuat *shader* secara visual. Melalui fitur ini, pengguna tidak perlu menulis kode dan sebagai gantinya pengguna dapat membuat dan menghubungkan *node* dalam sebuah *graph network*. Kerangka kerja dari *graph* memberikan umpan balik instan pada perubahan-perubahan yang dilakukan oleh pengguna. [19] Pada Tugas Akhir ini, *shader graph* akan digunakan untuk melakukan *post-processing* untuk membuat *little planet view*.

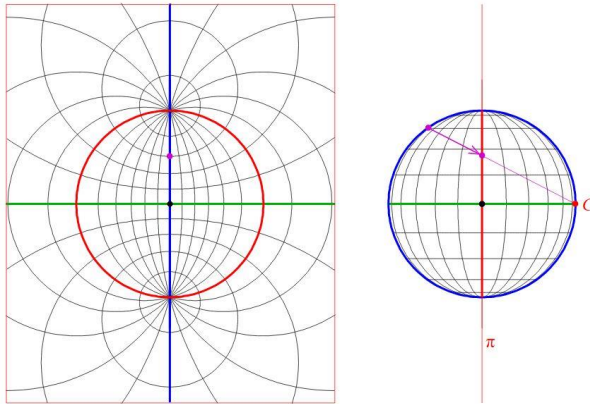
### 2.3.3.4 Tekstur

Tekstur pada Unity merupakan gambar yang digunakan saat me-render elemen *GameObject*, *sprite*, atau *user interface*. Biasanya, tekstur diaplikasikan pada permukaan sebuah *mesh* untuk memberikan detail visual pada *mesh* tersebut. Tekstur dapat diaplikasikan pada sebuah objek melalui material. [20]

## 2.4 Proyeksi Stereografi

Proyeksi stereografi merupakan sebuah metode pemetaan yang memproyeksikan sebuah bola pada sebuah dataran. Proyeksi

ini merupakan proyeksi konformal yang berarti gambar hasil proyeksi mempertahankan lingkaran dan sudut dari gambar asli; sebuah lingkaran pada bola akan diproyeksikan sebagai lingkaran pada dataran, serta sudut di antara dua buah garis pada bola sama dengan sudut di antara hasil proyeksi dari garis-garis tersebut. [21] Gambar 2.4 mengilustrasikan permukaan sebuah bola yang sudah diproyeksikan pada sebuah dataran.



**Gambar 2.4 Perbandingan Sebuah Bola serta Hasil Proyeksinya**  
[22]

Rumus transformasi untuk sebuah bola dengan radius  $R$  adalah [23]

$$x = k \cos \phi \sin(\lambda - \lambda_0)$$

$$y = k[\cos \phi_1 \sin \phi - \sin \phi_1 \cos \phi \cos(\lambda - \lambda_0)]$$

di mana  $(x, y)$  merupakan titik koordinat pada hasil proyeksi,  $\lambda_0$  merupakan garis bujur tengah,  $\phi_1$  merupakan garis lintang tengah, serta  $k$  dijabarkan sebagai [23]

$$k = \frac{2R}{1 + \sin \phi_1 \sin \phi + \cos \phi_1 \cos \phi \cos(\lambda - \lambda_0)}$$

### **2.4.1 *Little Planet***

Proyeksi stereografi telah digunakan untuk memetakan panorama bola, yang menghasilkan sebuah efek yang dikenal sebagai *little planet*, dengan pusat proyeksinya merupakan nadir. [24] Pada Tugas Akhir ini, akan diaplikasikan efek *little planet* terhadap pandangan pengguna terhadap lingkungannya.

## **2.5 Oculus Rift**

Oculus Rift merupakan serangkaian VR *headset* yang dikembangkan dan diproduksi oleh Oculus VR. Oculus Rift versi consumer pertama dirilis secara komersial pada 25 Maret, 2016. [25] Oculus Rift dapat dijalankan pada perangkat Windows, macOS, dan GNU/Linux. Terdapat tiga komponen utama dalam sebuah *set* Oculus Rift, yaitu *headset*, sensor, serta *controller*. [26] Oculus Rift umumnya digunakan sebagai perangkat *game*, tetapi juga dapat digunakan untuk mengonsumsi media seperti film [27], sebagai aplikasi sosial [28], serta digunakan sebagai alat pendidikan. [29] Pada Tugas Akhir ini, Oculus Rift digunakan untuk melihat lingkungan sekitar melalui kamera 360°.

## **2.6 Vuforia**

Vuforia merupakan *software development kit* (SDK) untuk mengembangkan aplikasi *augmented reality*. SDK ini dapat mengenali gambar planar serta objek tiga dimensi virtual dalam waktu nyata. Fitur tersebut memungkinkan kita untuk meletakkan objek virtual pada objek di dunia nyata ketika dilihat melalui sebuah kamera. [30] Pada Tugas Akhir ini, Vuforia digunakan untuk mengembangkan fitur *augmented reality* pada mode sudut pandang orang pertama.

## **2.7 *Spatial Cognition***

*Spatial cognition* merupakan kemampuan terdiri dari pengkodean, manipulasi, integrasi, pengambilan, dan pemanfaatan informasi spasial, yaitu informasi mengenai unsur-unsur yang merupakan bagian dari ruang dua atau tiga dimensi. [1]



## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

Pada bab 3 ini akan dijelaskan mengenai analisis dan perancangan perangkat lunak untuk mencapai tujuan dari Tugas Akhir. Perancangan ini meliputi analisis metode secara umum dan perancangan sistem.

#### **3.1 Analisis Metode Secara Umum**

Pada subbab berikut akan dijelaskan analisis aplikasi untuk menyimulasikan pengalaman realita lingkungan di sekitar kita melalui sudut pandang orang pertama yang dilengkapi *augmented reality* dan sudut pandang orang ketiga dengan proyeksi *little planet*. Analisis yang dilakukan meliputi analisis permasalahan, deskripsi umum sistem, dan kebutuhan fungsional sistem.

##### **3.1.1 Analisis Permasalahan**

Permasalahan utama yang diangkat pada pembuatan Tugas Akhir ini adalah bagaimana membuat aplikasi untuk mengamati reaksi manusia dalam mengalami lingkungan di sekitarnya dengan realitas yang diaugmentasi.

Untuk mengatasi permasalahan tersebut, pada Tugas Akhir ini akan dilakukan pembuatan aplikasi di mana pengguna dapat melihat lingkungan sekitarnya melalui dua sudut pandang, yaitu:

1. Sudut pandang orang pertama dengan menyalurkan rekaman dari sebuah *action cam* ke Oculus Rift secara *real-time*, yang dilengkapi elemen *augmented reality* berupa *heads-up display* serta pengambilan objek virtual.
2. Sudut pandang orang ketiga dengan menyalurkan rekaman secara *real-time* dari kamera 360° ke Oculus Rift, yang dilengkapi dengan elemen *augmented reality* berupa *heads-up display*. Pada sudut pandang ini, akan dilakukan *post-processing* agar tampilan di-render sebagai *little planet*.

Aplikasi ini dikembangkan melalui Unity karena Unity merupakan *engine* yang fleksibel serta mendukung pengembangan aplikasi AR/VR.

### 3.1.2 Deskripsi Umum Sistem

Pada Tugas Akhir ini akan dibuat sebuah aplikasi untuk menyimulasikan pengalaman realita lingkungan di sekitar kita melalui sudut pandang orang pertama yang dilengkapi dengan *augmented reality* dan sudut pandang orang ketiga dengan proyeksi *little planet*. Pengguna akan melihat lingkungan di sekitarnya dengan menggunakan Oculus Rift yang menampilkan rekaman yang disalurkan dari *action cam* serta kamera 360° secara *real-time*.

Pada Gambar 3.1 dijelaskan cara kerja aplikasi secara umum.



**Gambar 3.1** Arsitektur Sistem

### 3.1.3 Spesifikasi Kebutuhan Perangkat Lunak

Subbab ini membahas spesifikasi kebutuhan perangkat lunak dari hasil analisis yang telah dilakukan. Bagian ini berisi kebutuhan perangkat lunak yang direpresentasikan dalam bentuk kebutuhan fungsional, kebutuhan nonfungsional, diagram kasus penggunaan, dan diagram aktivitas.

#### 3.1.3.1 Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan fitur yang harus dimiliki oleh perangkat lunak, reaksi dari perangkat lunak terhadap suatu masukan, hasil yang dilakukan perangkat lunak pada situasi khusus.

Kebutuhan fungsional dari perangkat lunak dijelaskan pada Tabel 3.1.

**Tabel 3.1 Kebutuhan Fungsional**

No	Kebutuhan Fungsional	Deskripsi
1	Melihat lingkungan sekitar dengan sudut pandang orang pertama	Melihat lingkungan sekitar dengan sudut pandang orang pertama yang dilengkapi dengan fitur <i>augmented reality</i> .
2	Melihat lingkungan sekitar dengan sudut pandang orang ketiga	Melihat lingkungan sekitar dengan sudut pandang orang pertama yang dilengkapi dengan yang telah di- <i>post-process</i> menjadi <i>little planet</i> .
3	Mencatat jumlah objek yang sudah diambil pengguna	Mencatat jumlah objek virtual yang sudah diambil oleh pengguna dalam mode sudut pandang orang pertama.

### 3.1.3.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional mendefinisikan batasan layanan yang ditawarkan oleh sistem. Kebutuhan non fungsional dari perangkat lunak dijelaskan pada Tabel 3.2.

**Tabel 3.2 Kualitas Perangkat Lunak**

No	Parameter	Deskripsi
1	Bahasa	Bahasa yang digunakan pada antarmuka merupakan bahasa Inggris.
2	<i>Functional completeness</i>	Aplikasi memenuhi fungsi yang sudah ditentukan.
3	<i>Learnability</i>	Pengguna dapat dengan mudah mempelajari cara menggunakan aplikasi.
4	<i>Operability</i>	Aplikasi mudah untuk dioperasikan pada komputer.

### 3.1.3.3 Aktor

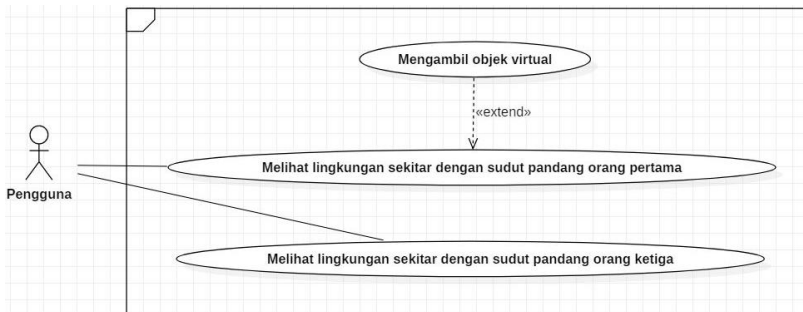
Pengertian pengguna adalah pihak-pihak, baik manusia maupun sistem atau perangkat lain yang terlibat dan berinteraksi secara langsung dengan sistem. Pada aplikasi ini, tidak terdapat pengguna dengan peran spesifik dan hanya terdapat satu tipe pengguna untuk menguji fungsionalitas sistem.

### 3.1.3.4 Kasus Penggunaan

Berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem, dibuat kasus penggunaan sistem. Kasus penggunaan dijelaskan pada Tabel 3.3 dan diagram kasus penggunaan ditunjukkan pada Gambar 3.2 Diagram Kasus Penggunaan Tabel 3.3.

**Tabel 3.3 Kasus Penggunaan**

Kode Kasus Penggunaan	Nama	Aktor
UC-001	Melihat lingkungan sekitar dengan sudut pandang orang pertama	Pengguna
UC-002	Melihat lingkungan sekitar dengan sudut pandang orang ketiga	Pengguna
UC-003	Mengambil objek virtual	Pengguna



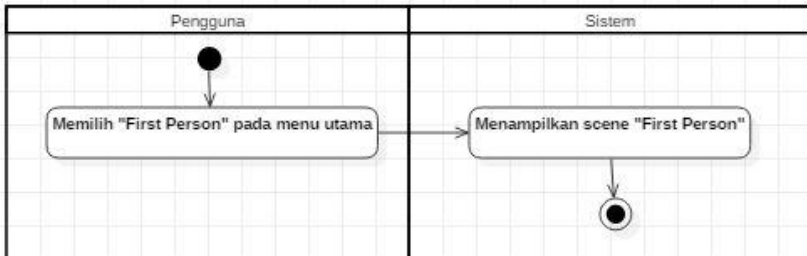
**Gambar 3.2 Diagram Kasus Penggunaan**

### 3.1.3.4.1 Melihat Lingkungan Sekitar dengan Sudut Pandang Orang Pertama (UC-001)

Kasus penggunaan nomor UC-001 diakses ketika pengguna hendak mengakses mode sudut pandang orang pertama. Spesifikasi serta diagram aktivitas ditunjukkan pada Tabel 3.4 dan Gambar 3.3.

**Tabel 3.4 Spesifikasi Kasus Penggunaan UC-001**

<b>Kode Use Case</b>	UC-001	
<b>Nama Use Case</b>	Melihat lingkungan sekitar dengan sudut pandang orang pertama	
<b>Aktor</b>	Pengguna	
<b>Deskripsi</b>	Pengguna melihat lingkungan sekitar dengan sudut pandang orang pertama yang dilengkapi dengan fitur <i>augmented reality</i> .	
<b>Relasi</b>	-	
<b>Kondisi Awal</b>	Pengguna belum melihat lingkungan sekitar dengan sudut pandang orang pertama	
<b>Kondisi Akhir</b>	Pengguna sudah melihat lingkungan sekitar dengan sudut pandang orang pertama	
<b>Alur kejadian normal</b>	Pengguna	Sistem
	1. Memilih "First Person" pada menu utama	
		2. Menampilkan <i>scene</i> "First Person"
<b>Alur kejadian alternatif</b>	Aktor	Sistem
	-	-



Gambar 3.3 Diagram Aktivitas UC-001

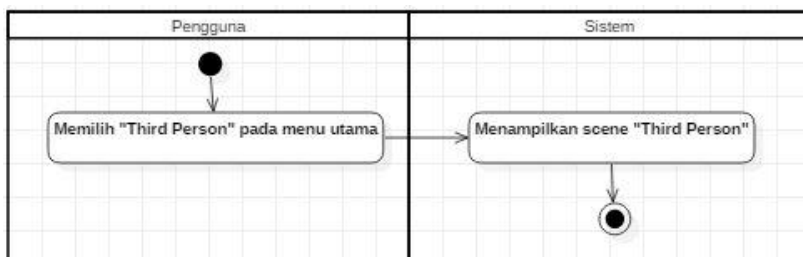
### 3.1.3.4.2 Melihat Lingkungan Sekitar dengan Sudut Pandang Orang Ketiga (UC-002)

Kasus penggunaan nomor UC-002 diakses ketika pengguna hendak mengakses mode sudut pandang orang ketiga. Spesifikasi serta diagram aktivitas ditunjukkan pada Tabel 3.5 dan Gambar 3.3.

Tabel 3.5 Spesifikasi Kasus Penggunaan UC-002

<b>Kode Use Case</b>	UC-002	
<b>Nama Use Case</b>	Melihat lingkungan sekitar dengan sudut pandang orang ketiga	
<b>Aktor</b>	Pengguna	
<b>Deskripsi</b>	Pengguna melihat lingkungan sekitar dengan sudut pandang orang pertama yang dilengkapi dengan yang telah di-post-process menjadi little planet.	
<b>Relasi</b>	-	
<b>Kondisi Awal</b>	Pengguna belum melihat lingkungan sekitar dengan sudut pandang orang ketiga	
<b>Kondisi Akhir</b>	Pengguna sudah melihat lingkungan sekitar dengan sudut pandang orang ketiga	
	Pengguna	Sistem

<b>Alur kejadian normal</b>	1. Memilih “Third Person” pada menu utama	
		2. Menampilkan <i>scene</i> “Third Person”
<b>Alur kejadian alternatif</b>	Aktor	Sistem
	-	-



**Gambar 3.4 Diagram Aktivitas UC-002**

### 3.1.3.4.3 Mengambil Objek Virtual (UC-003)

Kasus penggunaan nomor UC-003 diakses ketika pengguna berada di mode sudut pandang orang pertama dan hendak mengambil sebuah objek virtual. Spesifikasi serta diagram aktivitas ditunjukkan pada Tabel 3.6 dan Gambar 3.5.

**Tabel 3.6 Spesifikasi Kasus Penggunaan UC-003**

<b>Kode Use Case</b>	UC-003
<b>Nama Use Case</b>	Mengambil objek virtual
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna mengambil objek virtual pada mode sudut pandang orang pertama
<b>Relasi</b>	-

<b>Kondisi Awal</b>	Objek virtual belum diambil oleh pengguna	
<b>Kondisi Akhir</b>	Objek virtual sudah diambil oleh pengguna	
<b>Alur kejadian normal</b>	Pengguna	Sistem
	1. Menatap objek virtual dengan <i>reticle</i> selama tiga detik	
		2. Menambah satu pada <i>counter</i> “Objects Collected”
		3. Menampilkan data <i>counter</i> “Objects Collected”
<b>Alur kejadian alternatif</b>	Aktor	Sistem
	-	-



**Gambar 3.5 Diagram Aktivitas UC-003**

## 3.2 Perancangan

Dalam subbab ini membahas perancangan dari aplikasi Tugas Akhir. Subbab ini terdiri dari lingkungan perancangan perangkat lunak dan perancangan antarmuka pengguna.

### 3.2.1 Perancangan Perangkat Lunak



Dalam Tugas Akhir ini, akan dibangun tiga buah *scene*, yaitu menu utama, *scene* untuk menjalankan mode sudut orang pertama, serta *scene* untuk menjalankan mode sudut orang ketiga.

### 3.2.1.1 Menu Utama

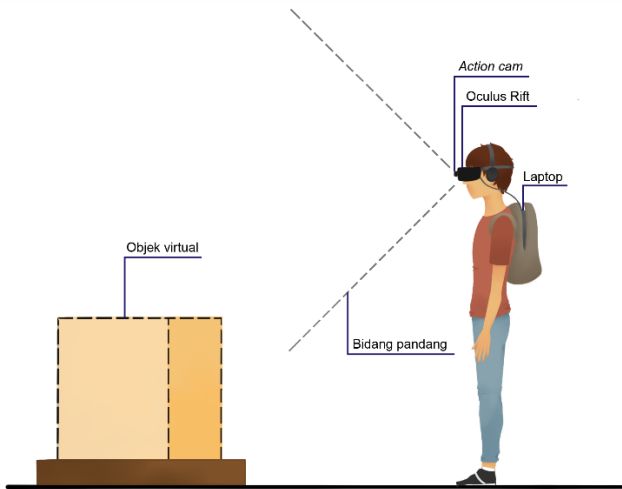
Menu utama diaktifkan ketika pengguna pertama kali menyalakan perangkat lunak. *Scene* ini berfungsi sebagai *main hub* di mana pengguna dapat memilih mode yang diinginkan. Pada *scene* ini, kamera yang digunakan adalah *prefab* yang telah disediakan oleh Oculus, yaitu *OVRCameraRig*. *OVRCameraRig* digunakan agar pengguna dapat melihat sekelilingnya dengan Oculus Rift. Selain itu, digunakan juga *prefab* *UIHelpers* agar pengguna dapat menggunakan *controller* yang berupa Oculus Touch sebagai input.

Pengguna dapat memilih mode dengan mengarahkan *laser pointer* pada “kartu” yang merepresentasikan mode-mode sudut pandang orang pertama dan ketiga. Masing-masing “kartu” merupakan *GameObject Image* yang merupakan anak dari *GameObject UI Canvas*. Masing-masing “kartu” memiliki *GameObject* anak berupa *UI Button “Select”*. Ketika tombol “Select” diklik, perangkat akan menonaktifkan *scene* menu utama dan mengaktifkan *scene* mode yang telah dipilih.

### 3.2.1.2 Mode Sudut Pandang Orang Pertama

Pada mode ini, pengguna dapat mengeksplorasi lingkungan mereka dalam sudut pandang orang pertama seperti di dunia nyata, hanya saja dengan realitas yang diaugmentasi.

Gambar 3.6 mengilustrasikan skenario mode sudut pandang orang pertama.



**Gambar 3.6 Skenario Mode Sudut Pandang Orang Pertama**

Perangkat keras yang digunakan untuk menjalankan mode ini adalah laptop, Oculus Rift, serta kamera berupa *action cam*. Kamera dipasang di bagian depan Oculus Rift agar rekaman yang disalurkan menyimulasikan sudut pandang di dunia nyata. Rekaman dari kamera tersebut kemudian disalurkan secara *real-time* ke Unity dan dapat dilihat melalui Oculus Rift ketika aplikasi dijalankan.

Pada mode ini, akan digunakan ARCamera yang telah disediakan oleh Vuforia. Vuforia digunakan untuk mengembangkan fitur di mana pengguna dapat mengambil objek virtual. Masing-masing objek virtual memiliki *marker* yang sudah

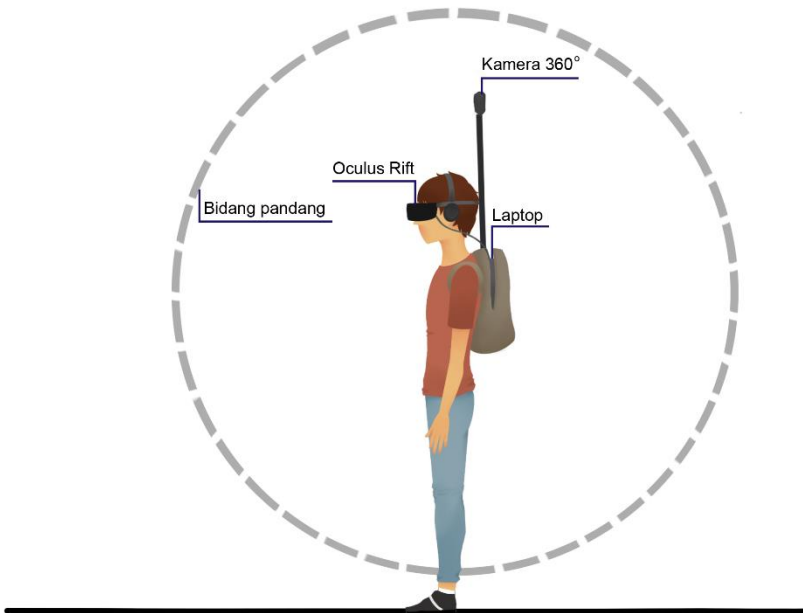
ditentukan sesuai dengan basis data yang digunakan. Pada ARCamera ditambahkan GameObject Reticle sebagai anak yang berfungsi untuk mengambil *user input*. Fitur *user input* yang digunakan berupa *gaze input*, di mana pengguna menatap sebuah objek virtual selama tiga detik untuk mengambil objek virtual.

Selain itu, terdapat juga fitur *heads-up display* yang dibangun pada GameObject Canvas. *Heads-up display* memuat informasi berupa hari, tanggal, dan waktu yang diambil dari sistem pengguna.

### 3.2.1.3 Mode Sudut Pandang Orang Ketiga

Pada mode ini, pengguna dapat mengeksplorasi lingkungan mereka dalam sudut pandang orang ketiga melalui *little planet view*. *Little planet view* digunakan agar pengguna dapat melihat keseluruhan dari lingkungannya pada saat yang bersamaan.

Gambar 3.7 mengilustrasikan skenario mode sudut pandang orang ketiga.



**Gambar 3.7 Skenario Mode Sudut Pandang Orang Ketiga**

Perangkat keras yang digunakan untuk menjalankan mode ini adalah laptop, Oculus Rift, serta kamera 360°. Kamera dipasang di sebuah *monopod* yang diletakkan di belakang pengguna agar rekaman yang disalurkan menyimulasikan sudut pandang orang ketiga (*tethered view*). Rekaman dari kamera tersebut kemudian

disalurkan secara *real-time* ke Unity dan dapat dilihat melalui Oculus Rift ketika aplikasi dijalankan.

### 3.2.1.3.1 Membuat Unity Mengenali Kamera 360°

Sebelum rekaman dari kamera 360° dapat disalurkan ke Unity, terlebih dahulu dilakukan proses *me-register* kamera 360° (pada kasus ini RICOH THETA S) pada Windows agar kamera tersebut dapat dideteksi sebagai *webcam*. Untuk *me-register* RICOH THETA S, langkah-langkah yang harus diambil adalah sebagai berikut:

1. Membuka *Registry Editor*.
2. Membuka *path* HKEY\_LOCAL\_MACHINE/SOFTWARE/Classes.
3. Membuka folder CLSID.
4. Membuka *path* {860BB310-5D01-11d0-BD3B-00A0C911CE86}/Instance/{2219F8F0-38CB-4B19-9DC1-3F1A5C324545}.
5. Menambahkan *string value registry* dengan nama DevicePath dan data foo:bar.

### 3.2.1.3.2 Perancangan Scene

Setelah di-*register*, rekaman dari kamera 360° dapat secara langsung disalurkan ke Unity dengan menggunakan WebCamTexture, yaitu kelas yang memungkinkan untuk *me-render live video* sebagai tekstur.

Pada *scene* ini, digunakan GameObject WebCameraManager. Melalui GameObject tersebut, pengguna dapat menentukan tinggi, lebar, dan *frames per second* dari WebCamTexture serta menerima input berupa material yang akan diaplikasikan *live video* sebagai teksturnya. *Script* WebCameraManager.cs dipasang pada GameObject tersebut. *Script* ini berfungsi untuk mengaktifkan kamera, meng-*assign video* pada tekstur, serta membuat *renderTexture* yang nantinya akan digunakan saat *post-processing*. Pada *script* ini, fungsi void Update() akan memanggil fungsi Graphics.Blit(frontTex, renderTexture) setiap *frame*. Fungsi ini

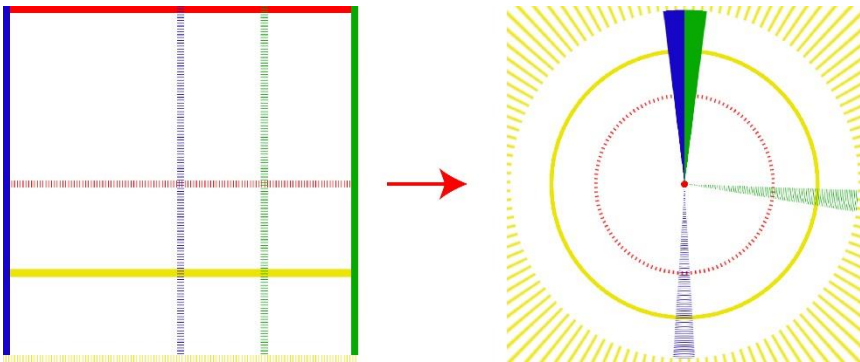
menyalin tekstur sumber ke tekstur *render* tujuan setelah tekstur sumber di-*post-process*. Pada konteks ini, tekstur sumber adalah *frontTex*, yaitu tekstur yang berisi *live video*, sedangkan tekstur *render* tujuannya adalah *renderTexture*. Material yang teksturnya sudah di-*post-process* akan diaplikasikan pada sebuah *plane*.

### 3.2.1.3.3 Perancangan *Shader Graph*

*Post-processing* dilakukan pada rekaman yang disalurkan dengan menggunakan fitur *shader graph* sehingga *final render* berupa *little planet*. *Node* yang akan digunakan adalah *Polar Coordinates* serta *Rotate*.

*Node Polar Coordinates* digunakan untuk mentransformasi gambar *equirectangular* menjadi gambar *circular*. Untuk memahami efek ini lebih lanjut,

Gambar 3.8 mengilustrasikan perbandingan sebuah persegi dengan garis-garis penunjuk dengan gambar yang telah diaplikasikan *filter Polar Coordinates*.



**Gambar 3.8** Gambar yang Telah Diaplikasikan *Filter Polar Coordinates*

Perbedaan-perbedaan yang dapat diobservasi adalah:

1. *Pixel* pada bagian tengah atas menjadi titik pusat.
2. Garis-garis horizontal menjadi lingkaran, dengan radius yang semakin membesar semakin garis berada di bawah.
3. Garis horizontal paling bawah di-*stretch* dan terlihat seperti membingkai lingkaran.
4. Garis-garis vertikal menjadi radius lingkaran.
5. Garis-garis vertikal yang terletak di paling kiri dan kanan menjadi radius atas.
6. Lebar dari garis-garis horizontal dan panjang dari garis-garis vertikal menjadi setengahnya.

Berdasarkan observasi tersebut, manipulasi *Polar Coordinate* dapat digunakan untuk membuat sebuah gambar atau *video* memiliki tampilan *little planet*.

***Node Rotate* kemudian digunakan untuk merotasi *video* sebanyak -90°. Secara default, sebuah *video equirectangular* dimuat dengan rotasi -90°, sehingga yang seharusnya titik atas dari sebuah *video* menjadi terletak di sebelah kiri. Sesuai dengan**

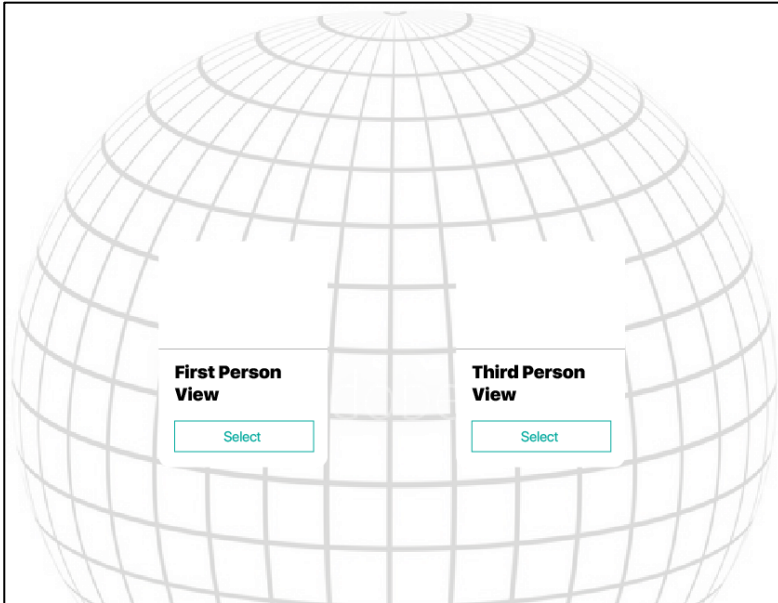
Gambar 3.8, titik tengah atas dari gambar sumber menjadi titik tengah gambar *circular*. Pada Tugas Akhir ini, yang perlu dijadikan titik tengah merupakan permukaan bumi. Oleh karena itu, *video* dirotasi sebanyak -90° agar bagian permukaan bumi terletak di atas.

### 3.2.2 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna merupakan hal yang penting dalam melakukan perancangan perangkat lunak. Antarmuka pengguna yang berhubungan dengan aktor harus memberikan kemudahan serta tampilan yang menarik bagi penggunanya. Pada aplikasi ini, terdapat antarmuka untuk menu utama serta mode sudut pandang orang pertama dan ketiga.

#### 3.2.2.1 Rancangan Antarmuka Menu Utama

Menu utama digunakan oleh pengguna untuk memilih mode yang ingin dijalankan. Pada antarmuka ini terdapat menu untuk memilih mode yang diinginkan. Rancangan antarmuka ditunjukkan pada Gambar 3.9.



**Gambar 3.9 Rancangan Antarmuka Menu Utama**

### **3.2.2.2 Rancangan Antarmuka Mode Sudut Pandang Orang Pertama**

Antarmuka ini digunakan ketika pengguna mengakses mode sudut pandang orang pertama. Pada antarmuka ini terdapat sebuah *reticle* di bagian tengah layer serta *heads-up display* yang menampilkan *counter* “Objects Collected” serta waktu, hari, dan tanggal dari sistem. Rancangan antarmuka ditunjukkan pada Gambar 3.10.





**Gambar 3.10 Rancangan Antarmuka Mode Sudut Pandang Orang Pertama**

### **3.2.2.3 Rancangan Antarmuka Mode Sudut Pandang Orang Ketiga**

Antarmuka ini digunakan ketika pengguna mengakses mode sudut pandang orang ketiga. Pada antarmuka ini terdapat *heads-up display* yang menampilkan waktu, hari, dan tanggal dari sistem. Rancangan antarmuka ditunjukkan pada Gambar 3.11.



**Gambar 3.11 Rancangan Antarmuka Mode Sudut Pandang Orang Ketiga**

## **BAB IV IMPLEMENTASI**

Bab ini membahas implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

Pada bagian implementasi ini juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam program Tugas Akhir ini dan disertai dengan kode sumber masing-masing fungsi utama.

### **4.1 Lingkungan Implementasi Perangkat Lunak**

Lingkungan implementasi sistem yang digunakan untuk mengembangkan Tugas Akhir memiliki spesifikasi perangkat keras dan perangkat lunak seperti ditampilkan pada Tabel 4.1.

**Tabel 4.1 Lingkungan Implementasi Perangkat Lunak**

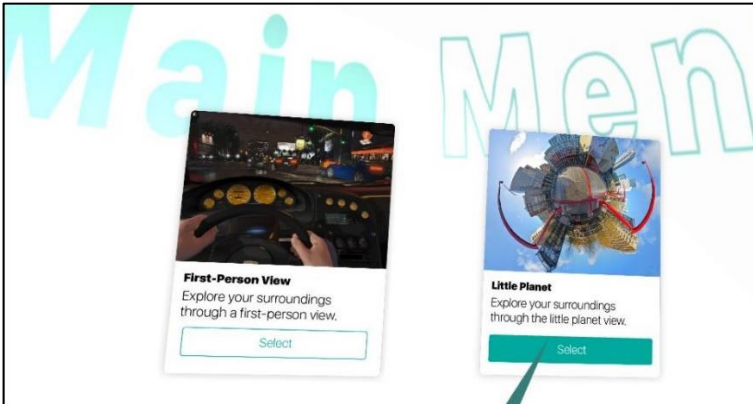
Perangkat Lunak	Sistem Operasi	Windows 10 Home Single Language 64-bit
	Perangkat Lunak	Adobe Illustrator, Microsoft Visual Studio, Unity
Perangkat Keras	Laptop, Oculus Rift, B-PRO5 Alpha Edition ( <i>action cam</i> ), RICOH THETA S (kamera 360°)	

### **4.2 Implementasi Antarmuka Pengguna**

Subbab ini membahas tentang implementasi tampilan antarmuka yang telah dirancang dan dibahas pada Bab III.

### 4.2.1 Implementasi Antarmuka Menu Utama

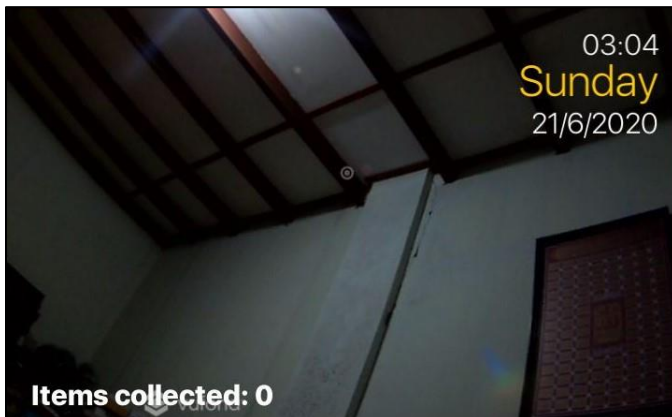
Antarmuka menu utama aplikasi ditunjukkan pada Gambar 4.1. Antarmuka ini mengacu pada kasus penggunaan nomor UC-001 dan UC-002. Terdapat tombol pada setiap “kartu” untuk mengaktifkan mode yang diinginkan.



Gambar 4.1 Implementasi Antarmuka Menu Utama

### 4.2.2 Implementasi Antarmuka Mode Sudut Pandang Orang Pertama

Antarmuka pada Gambar 4.2 merupakan antarmuka yang akan ditampilkan ketika pengguna mengaktifkan mode sudut pandang orang pertama. Antarmuka ini mengacu pada kasus penggunaan nomor UC-001.



**Gambar 4.2 Implementasi Antarmuka Mode Sudut Pandang Orang Pertama**

### **4.2.3 Implementasi Antarmuka Mode Sudut Pandang Orang Ketiga**

Antarmuka pada Gambar 4.3 merupakan antarmuka yang akan ditampilkan ketika pengguna mengaktifkan mode sudut pandang orang ketiga. Antarmuka ini mengacu pada kasus penggunaan nomor UC-002.



**Gambar 4.3 Implementasi Antarmuka Mode Sudut Pandang Orang Ketiga**

### **4.3 Implementasi Fitur Aplikasi**

Implementasi fitur aplikasi menggunakan bahasa C# serta fitur *shader graph* dari Unity. Pada subbab ini akan menjelaskan dan menampilkan kode yang digunakan dalam pembuatan aplikasi.

### **4.3.1 Implementasi Menu Utama**

Kode sumber dari implementasi menu utama ditunjukkan pada Kode Sumber 9-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 9..

### **4.3.2 Implementasi Mode Sudut Pandang Orang Pertama**

Pada mode ini, terdapat implementasi fitur pengambilan objek virtual serta implementasi *heads-up display*. Untuk fitur pengambilan objek, terdapat kode untuk mengembangkan *reticle* serta membuat objek virtual interaktif, sementara untuk fitur *heads-up display* terdapat kode untuk menunjukkan jam, hari, dan tanggal dari sistem serta *counter* “Objects Collected”.

#### **4.3.2.1 Implementasi VR Input**

Kode sumber dari implementasi VR *Input* ditunjukkan pada Kode Sumber 10-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 10.

#### **4.3.2.2 Implementasi VR Eye Raycaster**

Kode sumber dari implementasi VR *Eye Raycaster* ditunjukkan pada Kode Sumber 10-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 10.

#### **4.3.2.3 Implementasi Reticle**

Kode sumber dari implementasi *Reticle* ditunjukkan pada Kode Sumber 10-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 10.

#### **4.3.2.4 Implementasi VR Interactive Item**

Kode sumber dari implementasi VR *Interactive Item* ditunjukkan pada Kode Sumber 10-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 10.

#### **4.3.2.5 Implementasi Example VR Interactive Item**

Kode sumber dari implementasi Example Interactive Item ditunjukkan pada Kode Sumber 10-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 10.

#### **4.3.2.6 Implementasi Menunjukkan Jam**

Kode sumber dari implementasi menunjukkan jam ditunjukkan pada Kode Sumber 10-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 10.

#### **4.3.2.7 Implementasi Menunjukkan Hari**

Kode sumber dari implementasi menunjukkan hari ditunjukkan pada Kode Sumber 10-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 10.

#### **4.3.2.8 Implementasi Menunjukkan Tanggal**

Kode sumber dari implementasi menunjukkan tanggal ditunjukkan pada Kode Sumber 10-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 10.

#### **4.3.2.9 Implementasi Counter “Objects Collected”**

Kode sumber dari implementasi menunjukkan jam ditunjukkan pada Kode Sumber 10-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 10.

### **4.3.3 Implementasi Mode Sudut Pandang Orang Ketiga**

Pada mode ini, terdapat implementasi *post-processing little planet* serta implementasi *heads-up display*. Untuk implementasi *little planet*, digunakan sebuah kode dalam bahasa C# dan juga *shader graph*, sementara untuk fitur *heads-up display* terdapat kode untuk menunjukkan jam, hari, dan tanggal dari sistem.

#### **4.3.3.1 Implementasi Web Camera Manager**



Kode sumber dari implementasi *Web Camera Manager* ditunjukkan pada Kode Sumber 10-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 10.

#### 4.3.3.2 Implementasi *Shader Little Planet*

Untuk mengimplementasi *post-processing little planet*, digunakan sebuah *custom shader*. Pada kasus ini, fitur *shader graph* digunakan untuk membuat *custom shader* tersebut. Gambar 4.4 menunjukkan *node Polar Coordinates*, *Rotate*, dan *Sample Texture 2D* pada *shader graph* yang telah dibuat.



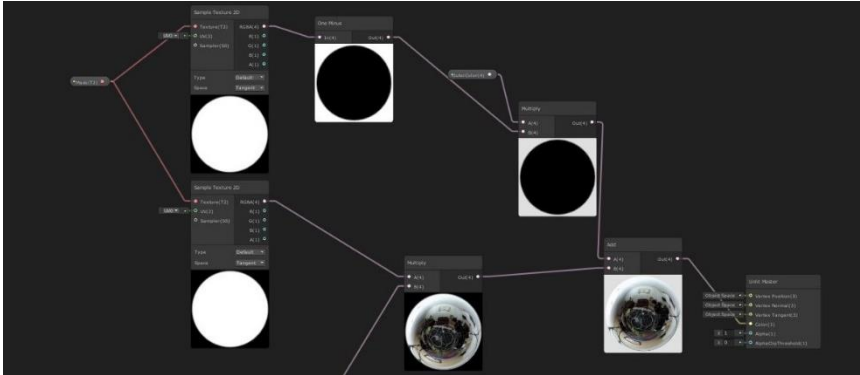
**Gambar 4.4** *Node Polar Coordinates, Rotate, dan Sample Texture 2D pada Shader*

Digunakan *custom property Distance* untuk input *Radial Scale* pada *node Polar Coordinates* serta *custom property Rotation* untuk input *Rotation* pada *node Rotate* agar nilai dari variabel-variabel tersebut dapat dimodifikasi melalui *Inspector*.

Pada *node Sample Texture 2D*, *custom property Texture2D* digunakan untuk menyambungkan *live video* yang telah di-assign oleh *Web Camera Manager* ke *shader*.

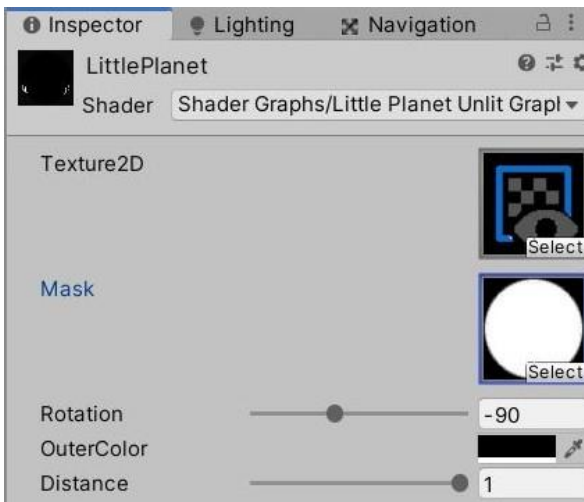
Setelah *video* di-*post-process* menjadi *little planet*, diaplikasikan sebuah *mask* agar hasil *render* akhir terlihat lebih rapi. Gambar 4.5 menunjukkan *node-node* yang

berfungsi untuk mengaplikasikan *mask* pada hasil *render* akhir. Pada *node Multiply*, digunakan *custom property* *OuterColor* agar warna dari *mask* dapat dimodifikasi.



**Gambar 4.5** *Node* yang Digunakan untuk Mengaplikasikan *Mask*

Gambar 4.6 menunjukkan hasil akhir dari *shader little planet*.



**Gambar 4.6** Implementasi *Shader Little Planet*

#### **4.3.3.3 Implementasi Menunjukkan Jam**

Kode sumber dari implementasi menunjukkan jam ditunjukkan pada Kode Sumber 10-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 10.

#### **4.3.3.4 Implementasi Menunjukkan Hari**

Kode sumber dari implementasi menunjukkan hari ditunjukkan pada Kode Sumber 10-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 10.

#### **4.3.3.5 Implementasi Menunjukkan Tanggal**

Kode sumber dari implementasi menunjukkan tanggal ditunjukkan pada Kode Sumber 10-1. Untuk penjelasan kode sumber ditunjukkan pada Tabel 10.

*[Halaman ini sengaja dikosongkan]*

## BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas tentang pengujian dan evaluasi pada perangkat lunak yang dibangun untuk Tugas Akhir ini. Pengujian yang dilakukan berupa uji latensi, sementara evaluasi dilakukan dengan mengadakan sebuah kuisioner untuk mencari tahu preferensi pengguna terhadap mode sudut pandang orang pertama dan ketiga.

### 5.1 Lingkungan Pengujian

Pada proses pengujian perangkat lunak, dibutuhkan suatu lingkungan pengujian yang sesuai dengan standar kebutuhan. Lingkungan pengujian dalam Tugas Akhir ini dilakukan pada uji latensi. Spesifikasi masing-masing lingkungan pengujian dijabarkan pada Tabel 5.1.

**Tabel 5.1 Lingkungan Pengujian Latensi Perangkat Lunak**

Perangkat	Spesifikasi
Perangkat Keras	<b>Prosesor:</b> Intel® Core™ i7-7700HQ CPU @ 2.80GHz <b>Memori:</b> 16384MB RAM
Perangkat Lunak	<b>Sistem Operasi:</b> Windows 10 Home Single Language 64-bit <b>Perangkat Pengembang:</b> Unity 2019.4.0f1, Microsoft Visual Studio Community 2017 version 15.3.3

### 5.2 Uji Performa

Uji performa diadakan dengan mengukur *frame rate* dari masing-masing mode. Setiap mode dijalankan selama 3.44 detik kemudian dilakukan *random point sampling* sebanyak 10 kali. Tabel 5.2 dan Tabel 5.3 menunjukkan hasil uji performa pada mode sudut pandang orang pertama dan mode *little planet*.

**Tabel 5.2 Hasil Uji Performa Mode Sudut Pandang Orang Pertama**

No.	<i>Frame Rate (fps)</i>
1	72.8
2	72.7
3	71.6
4	71.3
5	73.3
6	70.5
7	71.3
8	71.8
9	70.4
10	72.6
<b>Rata-Rata</b>	<b>71.65</b>

**Tabel 5.3 Hasil Uji Performa Mode *Little Planet***

No.	<i>Frame Rate (fps)</i>
1	62.1
2	59.4
3	61.4
4	62.5
5	62.8
6	62.7
7	61.3
8	62.5
9	60.7
10	62
<b>Rata-Rata</b>	<b>61.74</b>

Berdasarkan hasil uji performa di atas, mode sudut pandang orang pertama memiliki rata-rata *frame rate* sebesar 71.65 fps, sementara mode *little planet* memiliki rata-rata *frame rate* sebesar 61.74 fps.

### 5.3 Preferensi terhadap Mode Sudut Pandang Orang Pertama dan Ketiga

Sebagai pengganti pengujian fungsional, diadakan kuisioner untuk mencari tahu preferensi responden terhadap mode sudut pandang orang pertama dan ketiga (*little planet*) serta mencari tahu perspektif responden terhadap mode *little planet*. Pada kuisioner ini, ditunjukkan hasil implementasi dari Tugas Akhir ini. Terdapat 8 orang responden yang terdiri dari 6 orang laki-laki dan 2 orang perempuan. Rentang usia responden adalah 20 sampai dengan 22 tahun. 5 orang responden tidak pernah menggunakan aplikasi VR. Seluruh responden mengisi kuisioner melalui Google Forms.

Berdasarkan hasil kuisioner, 7 dari 8 responden lebih memilih mode sudut pandang orang pertama, dengan mayoritas alasan karena sudah terbiasa dengan sudut pandang orang pertama serta karena sudut pandang *little planet* memusingkan.

Akan tetapi, 4 dari 8 orang berpikir bahwa mode *little planet* masih layak untuk diaplikasikan ke dunia nyata, sementara sisanya berpikir bahwa mode tersebut tidak layak karena tidak *reliable*, banyaknya disorientasi, serta menyulitkan pengguna untuk fokus pada satu hal karena banyak yang ditampilkan pada saat yang bersamaan. Satu responden menuliskan bahwa permasalahan terbesar dari mode ini adalah horizon depan pengguna yang sangat sempit sehingga kemampuan aproksimasi jarak dan kecepatan menjadi berkurang.

Menurut responden, kelebihan dari mode *little planet* adalah sebagai berikut:

1. Dapat menyajikan informasi lebih sehingga memungkinkan pengguna untuk melihat apa yang tidak bisa ditunjukkan di mode sudut pandang orang pertama.
2. Jika ditampilkan tanpa VR *headset* ditampilkan di layar datar, justru akan membantu pengguna karena dapat melihat lingkungan yang lebih luas.
3. Memberikan perspektif yang baru dan unik.

Sementara itu, kekurangan dari mode *little planet* adalah:

1. Penggunaan mode *little planet* dengan VR headset memusingkan.
2. Sulit digunakan untuk melakukan kegiatan yang membutuhkan presisi, contohnya seperti memarkir mobil.
3. Kebutuhan dan instalasi perangkat yang cukup banyak dan tidak praktis tidak sebanding dengan manfaat yang didapatkan.



## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

#### **6.1 Kesimpulan**

Kesimpulan dari Tugas Akhir ini adalah tujuan dari aplikasi untuk mengeksplorasi lingkungan nyata melalui sudut pandang orang pertama dan ketiga telah berhasil dibangun dengan solusi dari beberapa rumusan masalah sebagai berikut:

1. Implementasi penyaluran *live footage* dari kamera ke VR *headset* berhasil dilakukan. Implementasi ini dilakukan dengan memanfaatkan kelas *WebCamTexture* pada Unity.
2. Penampilan objek virtual tiga dimensi di dunia nyata sebagai *augmented reality* berhasil dilakukan. Hal ini diimplementasikan dengan menggunakan SDK Vuforia. Diimplementasikan pula fitur di mana pengguna dapat mengambil objek-objek virtual dengan menggunakan *gaze input*.
3. Implementasi komponen antarmuka pada aplikasi berhasil dilakukan. Komponen antarmuka pada aplikasi diimplementasikan dengan menggunakan *GameObject UI Canvas* pada Unity.
4. *Post-processing video equirectangular* menjadi *little planet* berhasil dilakukan. Hal ini dilakukan dengan membuat *custom shader* menggunakan fitur *shader graph* dari Unity.

Berdasarkan hasil uji performa, mode sudut pandang orang memiliki rata-rata *frame rate* sebesar 71.65 fps, sementara mode *little planet* memiliki rata-rata *frame rate* 61.74 fps.

Kemudian, berdasarkan hasil kuisioner, mayoritas responden masih lebih memilih untuk menggunakan mode sudut pandang orang pertama daripada mode *little planet*, karena memusingkan, sulit digunakan, serta penggunaannya yang terbatas.

## **6.2 Saran**

Saran untuk pengembangan aplikasi ini di masa yang akan datang adalah perlu dilakukan pengujian fungsional yang melibatkan beberapa peserta secara langsung. Saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- [1] M. d. Goede, "Gender differences in spatial cognition," p. 2, 2009.
- [2] R. Hammoud, *Passive Eye Monitoring: Algorithms, Applications and Experiments*, 2008.
- [3] A. F. K. P. Milgram, "Taxonomy of Mixed Reality Visual Displays," in *Taxonomy of Mixed Reality Visual Displays*, IEICE Transactions on Information and Systems, 1994, pp. 1321-1329.
- [4] L. Brown, "What is 360-Degree Camera," Filmora, 24 November 2017. [Online]. Available: <https://filmora.wondershare.com/virtual-reality/what-is-360-degree-camera.html>. [Accessed 12 October 2019].
- [5] J. Meyer, "How a 360 camera works," Camera Jabber, 3 July 2018. [Online]. Available: <https://camerajabber.com/how-a-360-camera-works/>. [Accessed 12 October 2019].
- [6] G. Jameson, "Marketing in New Dimensions: 360 Degree Video," Web Stores Ltd., 31 May 2017. [Online]. Available: <https://webstoresltd.com/2017/05/31/marketing-in-new-dimensions-360-degree-video/>. [Accessed 2 May 2020].
- [7] S. Axon, "Unity at 10: For better—or worse—game development has never been easier," *Ars Technica*, 27 September 2016. [Online]. Available: <https://arstechnica.com/gaming/2016/09/unity-at-10-for-better-or-worse-game-development-has-never-been-easier/>. [Accessed 30 September 2019].
- [8] A. Galvan, "Unity Engine Architecture," 10 February 2017. [Online]. Available: <https://alain.xyz/blog/unity-engine-architecture>. [Accessed 2 May 2020].

- [9] Unity Technologies, "Scenes," Unity Technologies, [Online]. Available: <https://docs.unity3d.com/Manual/CreatingScenes.html>. [Accessed 2 May 2020].
- [10] Unity Technologies, "Meshes, Materials, Shaders and Textures," Unity Technologies, [Online]. Available: <https://docs.unity3d.com/Manual/Shaders.html>. [Accessed 2 May 2020].
- [11] Unity Technologies, "Models," Unity Technologies, [Online]. Available: <https://docs.unity3d.com/530/Documentation/Manual/FBXImporter-Model.html>. [Accessed 2 May 2020].
- [12] Unity Technologies, "Prefabs," Unity Technologies, [Online]. Available: <https://docs.unity3d.com/Manual/Prefabs.html>. [Accessed 2 May 2020].
- [13] Unity Technology, "Scripting," Unity Technology, [Online]. Available: <https://docs.unity3d.com/Manual/ScriptingSection.html>. [Accessed 2 May 2020].
- [14] Unity Technologies, "Meshes, Materials, Shaders and Textures," Unity Technologies, [Online]. Available: <https://docs.unity3d.com/Manual/Shaders.html>. [Accessed 2 May 2020].
- [15] M. Rouse and M. Haughn, "3D mesh," WhatIs.com, [Online]. Available: <https://whatis.techtarget.com/definition/3D-mesh#:~:text=A%203D%20mesh%20is%20the,with%20height%2C%20width%20and%20depth.&text=Some%20modelling%20software%20includes%20a,from%20making%20the%20general%20shape..> [Accessed 14 June 2020].
- [16] Unity Technologies, "Material," Unity Technologies, [Online]. Available:

- <https://docs.unity3d.com/Manual/class-Material.html>.  
[Accessed 14 June 2020].
- [17] J. Hocking, "Introduction to Shaders in Unity," 8 April 2020. [Online]. Available: <https://www.raywenderlich.com/5671826-introduction-to-shaders-in-unity#toc-anchor-009>. [Accessed 14 June 2020].
- [18] Unity Technologies, "Writing Shaders," Unity Technologies, [Online]. Available: <https://docs.unity3d.com/Manual/ShaderOverview.html>. [Accessed 14 June 2020].
- [19] T. Cooper, "Introduction to Shader Graph: Build your shaders with a visual editor," Unity Technologies, 27 February 2018. [Online]. Available: <https://blogs.unity3d.com/2018/02/27/introduction-to-shader-graph-build-your-shaders-with-a-visual-editor/>. [Accessed 14 June 2020].
- [20] Unity Technologies, "Textures," Unity Technologies, [Online]. Available: <https://docs.unity3d.com/Manual/Textures.html>. [Accessed 14 June 2020].
- [21] The Geometry Center, "Stereographic Projection," The Geometry Center, [Online]. Available: <http://www.geom.uiuc.edu/docs/education/institute91/handouts/node33.html>. [Accessed 3 May 2020].
- [22] "Stereographic Projection," Wikipedia, [Online]. Available: [https://en.wikipedia.org/wiki/Stereographic\\_projection](https://en.wikipedia.org/wiki/Stereographic_projection). [Accessed 3 May 2020].
- [23] MathWorld, "Stereographic Projection," MathWorld, [Online]. Available: <https://mathworld.wolfram.com/StereographicProjection.html>. [Accessed 3 May 2020].

- [24] D. German, L. Burchill, A. Duret-Lutz, S. Pérez-Duarte, E. Pérez-Duarte and J. Sommers, in *"Flattening the Viewable Sphere"*. *Proceedings of Computational Aesthetics*, Banff, Eurographics, 2007, pp. 23-28.
- [25] J. A. Falcon, "Oculus Rift VR Headset Starts Shipping to Customers," *Game Watcher*, 25 March 2016. [Online]. Available: <https://www.gamewatcher.com/news/2016-25-03-oculus-rift-vr-headset-starts-shipping-to-customers>. [Accessed 12 October 2019].
- [26] "Oculus User Guide," 2 December 2014. [Online]. Available: [https://web.archive.org/web/20150303181137/http://static.oculus.com/sdk-downloads/documents/Oculus\\_User\\_Guide\\_0.4.4.pdf](https://web.archive.org/web/20150303181137/http://static.oculus.com/sdk-downloads/documents/Oculus_User_Guide_0.4.4.pdf). [Accessed 12 October 2019].
- [27] L. Westaway, "Virtual reality and the silver screen: A match made in heaven," *CNET*, 13 October 2014. [Online]. Available: <https://www.cnet.com/news/virtual-reality-and-the-silver-screen-a-match-made-in-heaven/>. [Accessed 3 May 2020].
- [28] Recode.net, "Oculus Rift Inventor Palmer Luckey: Virtual Reality Will Make Distance Irrelevant (Q&A)," *Recode.net*, 19 June 2015. [Online]. Available: <http://www.recode.net/2015/6/19/11563728/oculus-rift-inventor-palmer-luckey-virtual-reality-will-make-distance>. [Accessed 3 May 2020].
- [29] C. Moro, Z. Štromberga and A. Stirling, "Virtualisation devices for student learning: Comparison between desktop-based (Oculus Rift) and mobile-based (Gear VR) virtual reality in medical and health science education," *Australasian Journal of Educational Technology*, 2017.
- [30] PTC, "Vuforia: Market-Leading Enterprise AR," *PTC*, [Online]. Available:



- <https://www.ptc.com/en/products/augmented-reality/vuforia>. [Accessed 14 June 2020].
- [31] "Allocentric vs. Egocentric Spatial Processing," Mental Imagery and Human-Computer Interaction Lab, [Online]. Available:  
[http://www.nmr.mgh.harvard.edu/mkozhevnlab/?page\\_id=308](http://www.nmr.mgh.harvard.edu/mkozhevnlab/?page_id=308). [Accessed 12 October 2019].
- [32] H. Fabroyir and W.-C. Teng, "Navigation in virtual environments using head-mounted displays;," 2018.
- [33] B. Ricks, C. W. Nielsen and M. A. Goodrich, "Ecological Displays for Robot Interaction: A New Perspective".
- [34] L. B. Rosenberg, *The Use of Virtual Fixtures As Perceptual Overlays to Enhance Operator Performance in Remote Environments*, 1992.
- [35] P. Schueffel, "The Concise Fintech Compendium," in *The Concise Fintech Compendium*, 2017, p. 2.

*[Halaman ini sengaja dikosongkan]*

## LAMPIRAN A

### Implementasi Menu Utama

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using OVRTouchSample;
5. using System;
6.
7. public class MainMenu : MonoBehaviour
8. {
9.     public string scene;
10.
11.     public void PlayScene()
12.     {
13.         Application.LoadLevel(scene);
14.     }
15. }
```

Kode Sumber 8-1 Implementasi Menu Utama

Tabel 8.1 Penjelasan Kode Sumber 8-1

No. Baris	Kegunaan
1-5	Deklarasi <i>namespace</i>
9	Deklarasi variabel untuk menyimpan nama <i>scene</i> yang akan dimuat
13	Muat <i>scene</i> yang sudah ditentukan

## LAMPIRAN B

### Implementasi VR *Input*

```

1.  using System;
2.  using UnityEngine;
3.
4.  namespace VRStandardAssets.Utils
5.  {
6.      public class VRInput : MonoBehaviour
7.      {
8.
9.          public event Action OnClick;
10.         public event Action OnDoubleClick;
11.
12.         [SerializeField] private float m_DoubleClickTime =
13.         0.3f;
14.
15.         private Vector2 m_MouseDownPosition;
16.         private Vector2 m_MouseUpPosition;
17.         private float m_LastMouseUpTime;
18.
19.         public float DoubleClickTime { get { return
20.         m_DoubleClickTime; } }
21.
22.         private void Update()
23.         {
24.             CheckInput();
25.         }
26.
27.         private void CheckInput()
28.         {
29.             if (Input.GetButtonDown("LeftClick"))
30.             {
31.                 m_MouseDownPosition = new
32.                 Vector2(Input.mousePosition.x, Input.mousePosition.y);
33.
34.                 if (OnDown != null)
35.                     OnDown();
36.             }
37.         }
38.     }

```

```
34.
35.     if (Input.GetButtonUp("LeftClick "))
36.     {
37.         m_MouseUpPosition = new
Vector2(Input.mousePosition.x, Input.mousePosition.y);
38.     }
39.
40.     if (Input.GetButtonUp("LeftClick "))
41.     {
42.         if (OnUp != null)
43.             OnUp();
44.
45.         if (Time.time - m_LastMouseUpTime <
m_DoubleClickTime)
46.         {
47.             if (OnDoubleClick != null)
48.                 OnDoubleClick();
49.         }
50.         else
51.         {
52.             if (OnClick != null)
53.                 OnClick();
54.         }
55.         m_LastMouseUpTime = Time.time;
56.     }
57. }
58.
59. private void OnDestroy()
60. {
61.     OnClick = null;
62.     OnDoubleClick = null;
63. }
64. }
65. }
```

**Kode Sumber 9-1 Implementasi VR *Input***

**Tabel 9.1 Penjelasan Kode Sumber 9-1**

No. Baris	Kegunaan
1-2	Deklarasi <i>namespace</i>
9-18	Deklarasi variabel
29	Ketika LeftClick ditekan, catat posisi dari <i>mouse</i>
31-32	Panggil fungsi yang memiliki OnDown()
37	Ketika LeftClick dilepas, catat posisi dari <i>mouse</i>
42-43	Panggil fungsi yang memiliki OnUp()
45	Jika waktu di antara terakhir kali LeftClick dilepas dan ditekan lagi kurang dari waktu <i>double click</i> yang telah ditentukan, klik dianggap sebagai <i>double click</i>
47-48	Panggil fungsi yang memiliki OnDoubleClick()
55	Catat waktu terakhir LeftClick dilepas
59-63	Memastikan semua <i>event</i> dimatikan sebelum di- <i>destroy</i>

## LAMPIRAN C

### Implementasi *VR Eye Raycaster*

```
1. using System;
2. using UnityEngine;
3.
4. namespace VRStandardAssets.Utils
5. {
6.     public class VREyeRaycaster : MonoBehaviour
7.     {
8.         public event Action<RaycastHit> OnRaycasthit;
9.
10.         [SerializeField] private Transform m_Camera;
11.         [SerializeField] private LayerMask m_ExclusionLayers;
12.         [SerializeField] private Reticle m_Reticle;
13.         [SerializeField] private VRInput m_VrInput;
14.         VRInteractiveItem.
15.         [SerializeField] private bool m_ShowDebugRay;
16.         [SerializeField] private float m_DebugRayLength = 50f;
17.         [SerializeField] private float m_DebugRayDuration = 1f;
18.         [SerializeField] private float m_RayLength = 5000f;
19.
20.         [SerializeField] private float autoClickTime = 1f;
21.
22.         private VRInteractiveItem m_CurrentInteractable;
23.         private VRInteractiveItem m_LastInteractable;
24.
25.         public VRInteractiveItem CurrentInteractable
26.         {
27.             get { return m_CurrentInteractable; }
28.         }
29.
30.         private void OnEnable()
31.         {
32.             m_VrInput.OnClick += HandleClick;
33.             m_VrInput.OnDoubleClick += HandleDoubleClick;
34.             m_VrInput.OnUp += HandleUp;
35.             m_VrInput.OnDown += HandleDown;
36.         }
37.     }
38. }
```

```
36.
37.     private void OnDisable()
38.     {
39.         m_VrInput.OnClick -= HandleClick;
40.         m_VrInput.OnDoubleClick -= HandleDoubleClick;
41.         m_VrInput.OnUp -= HandleUp;
42.         m_VrInput.OnDown -= HandleDown;
43.     }
44.
45.     private void Update()
46.     {
47.         EyeRaycast();
48.     }
49.
50.     private void EyeRaycast()
51.     {
52.         if (m_ShowDebugRay)
53.         {
54.             Debug.DrawRay(m_Camera.position,
m_Camera.forward * m_DebugRayLength, Color.blue,
m_DebugRayDuration);
55.         }
56.
57.         Ray ray = new Ray(m_Camera.position,
m_Camera.forward);
58.         RaycastHit hit;
59.
60.         if (Physics.Raycast(ray, out hit, m_RayLength,
~m_ExclusionLayers))
61.         {
62.             VRInteractiveItem interactible =
hit.collider.GetComponent<VRInteractiveItem>();
63.             m_CurrentInteractable = interactible;
64.
65.             if (interactable && interactible !=
m_LastInteractable)
66.             {
67.                 interactible.setAutoClickTime(autoClickTime);
68.                 interactible.Over();
```



```
69.         m_Reticle.fillInTime(autoClickTime);
70.     }
71.
72.     if (interactable != m_LastInteractable)
73.         DeactiveLastInteractable();
74.
75.     m_LastInteractable = interactable;
76.
77.     if (m_Reticle)
78.         m_Reticle.SetPosition(hit);
79.
80.     if (OnRaycasthit != null)
81.         OnRaycasthit(hit);
82.     }
83.     else
84.     {
85.         DeactiveLastInteractable();
86.         m_CurrentInteractable = null;
87.
88.         if (m_Reticle)
89.             m_Reticle.SetPosition();
90.     }
91. }
92.
93. private void DeactiveLastInteractable()
94. {
95.     if (m_LastInteractable == null)
96.         return;
97.
98.     m_Reticle.stopFilling();
99.     m_LastInteractable.Out();
100.    m_LastInteractable = null;
101. }
102.
103. private void HandleUp()
104. {
105.     if (m_CurrentInteractable != null)
106.         m_CurrentInteractable.Up();
107. }
```

```
108.
109.     private void HandleDown()
110.     {
111.         if (m_CurrentInteractable != null)
112.             m_CurrentInteractable.Down();
113.     }
114.
115.     private void HandleClick()
116.     {
117.         Debug.Log("HandleClick raycaster");
118.         if (m_CurrentInteractable != null)
119.             m_CurrentInteractable.Click();
120.     }
121.
122.     private void HandleDoubleClick()
123.     {
124.         if (m_CurrentInteractable != null)
125.             m_CurrentInteractable.DoubleClick();
126.     }
127. }
128. }
```

**Kode Sumber 10-1 Implementasi VR Eye Raycaster**

**Tabel 10.1 Penjelasan Kode Sumber 10-1**

No. Baris	Kegunaan
1-2	Deklarasi <i>namespace</i>
8	<i>Event</i> dipanggil setiap tatapan pengguna bertabrakan dengan <i>collider</i>
11	<i>Layer</i> yang tidak dianggap oleh <i>raycast</i>
12	Input untuk GameObject <i>Reticle</i>
13	Memanggil <i>event</i> berdasarkan pada obyek <i>VRInteractiveItem</i>
14	Menunjukkan <i>debug ray</i>
15	Panjang <i>debug ray</i>
16	Lama <i>debug ray</i> akan muncul
17	Panjang <i>ray</i>
19	Waktu yang dibutuhkan untuk meng- <i>autoclick</i> sebuah objek ketika menatap objek tersebut
21	Benda interaksi yang sedang aktif
22	Benda interaksi terakhir
24	<i>Utility</i> untuk digunakan kelas lain untuk mendapatkan benda interaksi yang sedang aktif
52-55	Menunjukkan <i>debug ray</i>
57-58	Membuat <i>ray</i> yang mengarah lurus dari kamera
62-63	Membuat <i>VRInteractiveItem</i> terkena objek
65-70	Jika <i>ray</i> terkena sebuah benda interaksi dan benda berbeda dari benda interaksi sebelumnya, panggil <i>Over()</i>
72-73	Deaktivasi benda interaksi terakhir
77-78	Jika <i>ray</i> terkena sebuah benda interaksi, <i>set</i> posisi <i>reticle</i> pada lokasi pada titik tersebut
85-86	Jika <i>ray</i> tidak terkena apa-apa, deaktivasi benda interaksi terakhir
88-89	Posisikan <i>reticle</i> di posisi <i>default</i>

## LAMPIRAN D

### Implementasi *Reticle*

```
1. using UnityEngine;
2. using UnityEngine.UI;
3.
4. public class Reticle : MonoBehaviour
5. {
6.     [SerializeField] public float m_DefaultDistance = 5f;
7.     [SerializeField] private bool m_UseNormal;
8.     [SerializeField] private Image m_Image;
9.     [SerializeField] private Transform m_ReticleTransform;
10.    [SerializeField] private Transform m_Camera;
11.
12.    private Vector3 m_OriginalScale;
13.    private Quaternion m_OriginalRotation;
14.
15.    private float timeToClick = 1f;
16.    private bool filling = false;
17.
18.    public bool UseNormal
19.    {
20.        get { return m_UseNormal; }
21.        set { m_UseNormal = value; }
22.    }
23.
24.    public Transform ReticleTransform { get { return
m_ReticleTransform; } }
25.
26.    private void Awake()
27.    {
28.        m_OriginalScale = m_ReticleTransform.localScale;
29.        m_OriginalRotation =
m_ReticleTransform.localRotation;
30.    }
31.
32.    public void Hide()
33.    {
34.        Debug.Log("Hiding image");
```

```
35.     m_Image.enabled = false;
36.     }
37.
38.     public void Show()
39.     {
40.         Debug.Log("Hiding image");
41.         m_Image.enabled = true;
42.     }
43.
44.     public void SetPosition()
45.     {
46.         m_ReticleTransform.position = m_Camera.position +
m_Camera.forward * m_DefaultDistance;
47.
48.         m_ReticleTransform.localScale = m_OriginalScale *
m_DefaultDistance;
49.
50.         m_ReticleTransform.localRotation =
m_OriginalRotation;
51.     }
52.
53.     public void SetPosition(RaycastHit hit)
54.     {
55.         m_ReticleTransform.position = hit.point;
56.         m_ReticleTransform.localScale = m_OriginalScale *
hit.distance;
57.
58.         if (m_UseNormal)
59.             m_ReticleTransform.rotation =
Quaternion.FromToRotation(Vector3.forward, hit.normal);
60.         else
61.             m_ReticleTransform.localRotation =
m_OriginalRotation;
62.     }
63.
64.     public void fillInTime(float mainTimeToClick)
65.     {
66.         m_Image.fillAmount = 0f;
67.         filling = true;
```

```

68.     timeToClick = mainTimeToClick;
69.     }
70.
71.     public void stopFilling()
72.     {
73.         filling = false;
74.         m_Image.fillAmount = 0f;
75.     }
76.
77.     private void Update()
78.     {
79.         if (filling)
80.         {
81.             m_Image.fillAmount += Time.deltaTime /
timeToClick;
82.             if (m_Image.fillAmount > 1)
83.             {
84.                 m_Image.fillAmount = 0f;
85.                 filling = false;
86.             }
87.         }
88.     }
89. }

```

**Kode Sumber 11-1 Implementasi *Reticle***

**Tabel 11.1 Penjelasan Kode Sumber 11-1**

No. Baris	Kegunaan
1-2	Deklarasi <i>namespace</i>
6	Jarak dari <i>reticle</i> ke kamera yang digunakan
7	Menentukan apakah <i>reticle</i> diletakkan parallel pada sebuah permukaan
8	Gambar yang akan digunakan sebagai <i>reticle</i>
9	Posisi <i>reticle</i>
10	Posisi kamera
12	Ukuran <i>default</i> dari <i>reticle</i>
13	Rotasi <i>default</i> dari <i>reticle</i>

<b>No. Baris</b>	<b>Kegunaan</b>
44-51	Menjadikan nilai posisi, ukuran, dan rotasi dari <i>reticle</i> menjadi nilai <i>default</i>
53-62	Mengubah nilai posisi, ukuran, dan rotasi dari <i>reticle</i> ketika <i>ray</i> terkena sebuah objek

## LAMPIRAN E

### **Implementasi VR *Interactive Item***

```
1. using System;
2. using UnityEngine;
3.
4. namespace VRStandardAssets.Utils
5. {
6.     public class VRInteractiveItem : MonoBehaviour
7.     {
8.         public event Action OnOver;
9.         public event Action OnOut;
10.        public event Action OnClick;
11.        public event Action OnDoubleClick;
12.
13.        public bool autoClick = false;
14.        private float autoClickTime = 1f;
15.        private float clickTimerState = 0f;
16.        private bool clicked = false;
17.
18.        protected bool m_IsOver;
19.
20.        private void Update()
21.        {
22.            if (autoClick && m_IsOver && !clicked)
23.            {
24.                clickTimerState += Time.deltaTime;
25.                if (clickTimerState >= autoClickTime)
26.                {
27.                    clicked = true;
28.                    Click();
29.                }
30.            }
31.        }
32.
33.        public bool IsOver
34.        {
35.            get { return m_IsOver; }
36.        }
```



```
37.  
38.     public void setAutoClickTime(float time)  
39.     {  
40.         autoClickTime = time;  
41.     }  
42.  
43.     public void Over()  
44.     {  
45.         m_IsOver = true;  
46.  
47.         if (OnOver != null)  
48.             OnOver();  
49.     }  
50.  
51.     public void Out()  
52.     {  
53.         m_IsOver = false;  
54.         clicked = false;  
55.         clickTimerState = 0f;  
56.  
57.         if (OnOut != null)  
58.             OnOut();  
59.     }  
60.  
61.     public void Click()  
62.     {  
63.         if (OnClick != null)  
64.             OnClick();  
65.     }  
66.  
67.     public void DoubleClick()  
68.     {  
69.         if (OnDoubleClick != null)  
70.             OnDoubleClick();  
71.     }  
72. }  
73. }
```

**Kode Sumber 12-1 Implementasi VR *Interactive Item***

**Tabel 12.1 Penjelasan Kode Sumber 12-1**

<b>No. Baris</b>	<b>Kegunaan</b>
1-2	Deklarasi <i>namespace</i>
8	Dipanggil ketika <i>reticle</i> berada di atas objek
9	Diapnggil ketika <i>reticle</i> meninggalkan objek
10	Dipanggil ketika input klik terdeteksi ketika <i>reticle</i> berada di atas objek
11	Dipanggil ketika input <i>double click</i> terdeteksi ketika <i>reticle</i> berada di atas objek
13	Jika <i>true</i> , sistem akan secara otomatis menginputkan sebuah klik setelah waktu yang ditentukan
14	Interval bagi sistem untuk mengaktifkan <i>autoclick</i>
15	Variabel untuk menyimpan waktu klik
16	Variabel untuk menyatakan apakah sebuah objek diklik
43-70	Fungsi-fungsi ini dipanggil oleh VREyeRaycaster ketika input yang sesuai dideteksi

## LAMPIRAN F

### **Implementasi *Example Interactive Item***

```
1. using UnityEngine;
2. using VRStandardAssets.Utils;
3.
4. public class Global
5. {
6.     public static int count = 0;
7. }
8.
9. namespace VRStandardAssets.Examples
10. {
11.     public class ExampleInteractiveItem : MonoBehaviour
12.     {
13.         [SerializeField] private Material m_NormalMaterial;
14.         [SerializeField] private Material m_OverMaterial;
15.         [SerializeField] private Material m_ClickedMaterial;
16.         [SerializeField] private Material
m_DoubleClickedMaterial;
17.         [SerializeField] private VRInteractiveItem
m_InteractiveItem;
18.         [SerializeField] private Renderer m_Renderer;
19.
20.         private void Awake()
21.         {
22.             m_Renderer.material = m_NormalMaterial;
23.         }
24.
25.         private void OnEnable()
26.         {
27.             m_InteractiveItem.OnOver += HandleOver;
28.             m_InteractiveItem.OnOut += HandleOut;
29.             m_InteractiveItem.OnClick += HandleClick;
30.             m_InteractiveItem.OnDoubleClick +=
HandleDoubleClick;
31.         }
32.
33.
```

```
34.     private void OnDisable()
35.     {
36.         m_InteractiveItem.OnOver -= HandleOver;
37.         m_InteractiveItem.OnOut -= HandleOut;
38.         m_InteractiveItem.OnClick -= HandleClick;
39.         m_InteractiveItem.OnDoubleClick -=
HandleDoubleClick;
40.     }
41.
42.     private void HandleOver()
43.     {
44.         Debug.Log("Show over state");
45.         m_Renderer.material = m_OverMaterial;
46.     }
47.
48.     private void HandleOut()
49.     {
50.         Debug.Log("Show out state");
51.         m_Renderer.material = m_NormalMaterial;
52.     }
53.
54.     private void HandleClick()
55.     {
56.         Debug.Log("Show click state");
57.         //m_Renderer.material = m_ClickedMaterial;
58.         m_Renderer.enabled = false;
59.         Global.count += 1;
60.     }
61.
62.     private void HandleDoubleClick()
63.     {
64.         Debug.Log("Show double click");
65.         m_Renderer.material = m_DoubleClickedMaterial;
66.     }
67. }
68.
69. }
```

**Kode Sumber 13-1 Implementasi *Example Interactive Item***

**Tabel 13.1 Penjelasan Kode Sumber 13-1**

<b>No. Baris</b>	<b>Kegunaan</b>
1-2	Deklarasi <i>namespace</i>
4-7	Deklarasi variabel global untuk digunakan pada <i>counter</i> “Objects Collected”
13-18	Deklarasi variabel
25-40	Fungsi untuk mengelola <i>event</i> berdasarkan input
42-46	Dipanggil ketika <i>reticle</i> berada di atas objek
48-52	Dipanggil ketika <i>reticle</i> tidak berada di atas objek
54-60	Dipanggil ketika terjadi input klik serta menambah 1 pada variabel <i>count</i> yang akan digunakan untuk <i>counter</i> “Objects Collected”
62-66	Dipanggil ketika terjadi input <i>double click</i>

## LAMPIRAN G

### Implementasi Menunjukkan Jam (Text)

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class DisplayTimeOnText : MonoBehaviour
7. {
8.     private Text clockText;
9.
10.    void Start()
11.    {
12.        clockText = GetComponent<Text>();
13.    }
14.
15.    void Update()
16.    {
17.        System.DateTime time = System.DateTime.Now;
18.
19.        clockText.text = time.ToString("HH:mm");
20.    }
21. }
```

**Kode Sumber 14-1 Implementasi Menunjukkan Jam (Text)**

**Tabel 14.1 Penjelasan Kode Sumber 14-1**

<b>No. Baris</b>	<b>Kegunaan</b>
1-4	Deklarasi <i>namespace</i>
8	Deklarasi variabel untuk menyimpan nilai dari komponen Text
12	Mengambil komponen Text dari GameObject di mana <i>script</i> ini di- <i>attach</i>
17	Mengambil waktu dari sistem
19	Mencetak waktu dengan format HH:mm pada variabel <code>clockText</code>

## LAMPIRAN H

### Implementasi Menunjukkan Hari (Text)

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class DisplayDayOnText : MonoBehaviour
7. {
8.     private Text dayText;
9.
10.    void Start()
11.    {
12.        dayText = GetComponent<Text>();
13.    }
14.
15.    void Update()
16.    {
17.        System.DateTime time = System.DateTime.Now;
18.
19.        dayText.text = time.ToString("dddd");
20.    }
21. }
```

**Kode Sumber 15-1 Implementasi Menunjukkan Hari (Text)**



**Tabel 15.1 Penjelasan Kode Sumber 15-1**

<b>No. Baris</b>	<b>Kegunaan</b>
1-4	Deklarasi <i>namespace</i>
8	Deklarasi variabel untuk menyimpan nilai dari komponen Text
12	Mengambil komponen Text dari GameObject di mana <i>script</i> ini di- <i>attach</i>
17	Mengambil waktu dari sistem
19	Mencetak waktu dengan format dddd pada variabel <code>dayText</code>

## LAMPIRAN I

### Implementasi Menunjukkan Tanggal (Text)

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class DisplayDateOnText : MonoBehaviour
7. {
8.     private Text dateText;
9.
10.    void Start()
11.    {
12.        dateText = GetComponent<Text>();
13.    }
14.
15.    void Update()
16.    {
17.        System.DateTime time = System.DateTime.Now;
18.
19.        dateText.text = time.ToString("d/M/yyyy");
20.    }
21. }
```

**Kode Sumber 16-1 Implementasi Menunjukkan Tanggal (Text)**

**Tabel 16.1 Penjelasan Kode Sumber 16-1**

<b>No. Baris</b>	<b>Kegunaan</b>
1-4	Deklarasi <i>namespace</i>
8	Deklarasi variabel untuk menyimpan nilai dari komponen Text
12	Mengambil komponen Text dari GameObject di mana <i>script</i> ini di- <i>attach</i>
17	Mengambil waktu dari sistem
19	Mencetak waktu dengan format d/M/yyyy pada variabel <code>dayText</code>

## LAMPIRAN J

### Implementasi *Counter* “Objects Collected”

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class ItemsCollected : MonoBehaviour
7. {
8.     private Text countText;
9.
10.    void Start()
11.    {
12.        countText = GetComponent<Text>();
13.    }
14.
15.    void Update()
16.    {
17.        countText.text = "Items collected: " + Global.count;
18.    }
19. }
```

**Kode Sumber 17-1 Implementasi *Counter* “Objects Collected”**

**Tabel 17.1 Penjelasan Kode Sumber 17-1**

<b>No. Baris</b>	<b>Kegunaan</b>
1-4	Deklarasi <i>namespace</i>
8	Deklarasi variabel untuk menyimpan nilai dari komponen Text
12	Mengambil komponen Text dari GameObject di mana <i>script</i> ini di- <i>attach</i>
17	Menampilkan nilai dari variabel global int yang berisi jumlah objek virtual yang telah diklik

**LAMPIRAN K****Implementasi *Web Camera Manager***

```
1. using System;
2. using System.Collections;
3. using System.Collections.Generic;
4. using System.IO;
5. using UnityEngine;
6. using UnityEngine.UI;
7.
8. public class WebCameraManager : MonoBehaviour
9. {
10.     public static WebCameraManager Instance;
11.
12.     public WebCamDevice[] devices;
13.
14.     public WebCamTexture frontTex;
15.
16.     public int width = 1920;
17.     public int height = 1080;
18.     public int framesPerSec = 60;
19.
20.     public string frontCamName;
21.     public string backCamName;
22.
23.     public RawImage cameraRawImage;
24.
25.     public Material materialToChange;
26.     public Material materialSecondToChange;
27.
28.     public RenderTexture renderTexture;
29.
30.     private void Awake()
31.     {
32.         if (Instance == null)
33.         {
34.             Instance = this;
35.         }
36.     }
```

```
37.
38.     void Start()
39.     {
40.         InitializeWebcameras();
41.     }
42.
43.     void InitializeWebcameras()
44.     {
45.
46.         devices = WebCamTexture.devices;
47.
48.         foreach (WebCamDevice device in devices)
49.         {
50.             frontCamName = device.name;
51.         }
52.
53.         frontTex = new WebCamTexture(frontCamName, width,
height, framesPerSec);
54.
55.         AssignCamerasToTextures();
56.     }
57.
58.     void AssignCamerasToTextures()
59.     {
60.         if (frontTex != null)
61.         {
62.             frontTex.Play();
63.
64.             cameraRawImage.texture = frontTex;
65.             cameraRawImage.color = new Color32(255, 255, 255,
255);
66.
67.             materialToChange.mainTexture =
cameraRawImage.texture;
68.
69.             materialSecondToChange.SetTexture("_Texture2D",
renderTexture);
70.         }
71.         else
```

```
72.     {
73.         DebugLog("frontText is null...");
74.     }
75. }
76.
77. void Update()
78. {
79.     if (frontTex != null)
80.     {
81.         Graphics.Blit(frontTex, renderTexture);
82.     }
83. }
84. }
```

**Kode Sumber 18-1 Implementasi *Web Camera Manager***



**Tabel 18.1 Penjelasan Kode Sumber 18-1**

<b>No. Baris</b>	<b>Kegunaan</b>
1-6	Deklarasi <i>namespace</i>
12	Menyimpan perangkat-perangkat kamera yang terhubung pada sistem
14	Deklarasi instansi dari kelas WebCamTexture
16-18	Spesifikasi tinggi, lebar, dan <i>frames per second</i> dari WebCamTexture
23	Deklarasi instansi dari kelas RawImage
25	Material untuk menyimpan <i>video equirectangular</i>
26	Material yang menggunakan <i>shader little planet</i>
28	Deklarasi instansi dari kelas RenderTexture yang kan digunakan dalam <i>shader</i>
40	Memanggil fungsi InitializeWebcameras()
41	Mengambil perangkat kamera yang terhubung dengan sistem
48-51	Meng-assign perangkat ke frontCamName
53	Membuat instansi dari WebCamTexture frontTex dengan spesifikasi perangkat, lebar, tinggi, dan <i>frames per second</i> yang sudah ditentukan
55	Panggil fungsi AssignCamerasToTexture()
60	Menyalakan webcam
64	Meng-assign <i>live video</i> sebagai ke cameraRawImage
65	Menentukan warna <i>default</i> dari cameraRawImage yang pada kasus ini warna putih
67	Meng-assign tekstur cameraRawImage sebagai tekstur utama materialToChange
69	Meng-assign tekstur pada materialSecondToChange dengan <i>renderTexture</i> dan menamakannya “ Texture2D)
73	Mengeluarkan <i>debug log</i> jika kamera tidak terdeteksi
81	Mengopi tekstur dari frontTex yang telah di- <i>post-process</i> ke renderTexture

## LAMPIRAN L

### Implementasi Menunjukkan Jam (TextMesh)

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class DisplayTimeOnText : MonoBehaviour
7. {
8.     private TextMesh clockText;
9.
10.    void Start()
11.    {
12.        clockText = GetComponent<TextMesh>();
13.    }
14.
15.    void Update()
16.    {
17.        System.DateTime time = System.DateTime.Now;
18.
19.        clockText.text = time.ToString("HH:mm");
20.    }
21. }
```

**Kode Sumber 19-1 Implementasi Menunjukkan Jam (TextMesh)**

**Tabel 19.1 Penjelasan Kode Sumber 19-1**

<b>No. Baris</b>	<b>Kegunaan</b>
1-4	Deklarasi <i>namespace</i>
8	Deklarasi variabel untuk menyimpan nilai dari komponen TextMesh
12	Mengambil komponen TextMesh dari GameObject di mana <i>script</i> ini di- <i>attach</i>
17	Mengambil waktu dari sistem
19	Mencetak waktu dengan format HH:mm pada variabel clockText

## LAMPIRAN M

### Implementasi Menunjukkan Hari (TextMesh)

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class DisplayDayOnText : MonoBehaviour
7. {
8.     private TextMesh dayText;
9.
10.    void Start()
11.    {
12.        dayText = GetComponent<TextMesh>();
13.    }
14.
15.    void Update()
16.    {
17.        System.DateTime time = System.DateTime.Now;
18.
19.        dayText.text = time.ToString("dddd");
20.    }
21. }
```

**Kode Sumber 20-1 Implementasi Menunjukkan Hari (TextMesh)**

**Tabel 20.1 Penjelasan Kode Sumber 20-1**

<b>No. Baris</b>	<b>Kegunaan</b>
1-4	Deklarasi <i>namespace</i>
8	Deklarasi variabel untuk menyimpan nilai dari komponen TextMesh
12	Mengambil komponen TextMesh dari GameObject di mana <i>script</i> ini di- <i>attach</i>
17	Mengambil waktu dari sistem
19	Mencetak waktu dengan format dddd pada variabel dayText

## LAMPIRAN N

### Implementasi Menunjukkan Tanggal (TextMesh)

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class DisplayDateOnText : MonoBehaviour
7. {
8.     private TextMesh dateText;
9.
10.    void Start()
11.    {
12.        dateText = GetComponent<TextMesh>();
13.    }
14.
15.    void Update()
16.    {
17.        System.DateTime time = System.DateTime.Now;
18.
19.        dateText.text = time.ToString("d/M/yyyy");
20.    }
21. }
```

**Kode Sumber 21-1 Implementasi Menunjukkan Tanggal  
(TextMesh)**

**Tabel 21.1 Penjelasan Kode Sumber 21-1**

<b>No. Baris</b>	<b>Kegunaan</b>
1-4	Deklarasi <i>namespace</i>
8	Deklarasi variabel untuk menyimpan nilai dari komponen TextMesh
12	Mengambil komponen TextMesh dari GameObject di mana <i>script</i> ini di- <i>attach</i>
17	Mengambil waktu dari sistem
19	Mencetak waktu dengan format d/M/yyyy pada variabel dayText

**LAMPIRAN O****Tabel 22.1 Kuisisioner Terhadap Pengguna**

<b>No</b>	<b>Pertanyaan</b>
1.	Preferensi responden terhadap mode sudut pandang orang pertama dan ketiga
2.	Alasan dibalik preferensi responden
3.	Opini responden mengenai <i>viability little planet view</i>
4.	Aplikasi lain dari <i>little planet view</i>
5.	Pro kontra dari penggunaan <i>little planet view</i>
6.	Perbaikan yang dapat dilakukan untuk meningkatkan <i>viability</i> dari <i>little planet view</i>



## BIODATA PENULIS



Putri Nurul Aprilliandini lahir di Bandung pada tanggal 5 April 1998. Penulis telah menempuh pendidikan formal mulai dari TK Taruna Bakti Bandung, SD Taruna Bakti Bandung, SMP Taruna Bakti Bandung, SMA Taruna Bakti Bandung dan terakhir sebagai mahasiswa Departemen Informatika Institut Teknologi Sepuluh Nopember dengan rumpun mata kuliah Interaksi, Grafika, dan Seni (2016-2020). Lulus dari SMA, penulis melanjutkan pendidikan di Departemen Informatika.

Selama perkuliahan, penulis aktif dalam organisasi kemahasiswaan, antara lain sebagai staf Dekorasi & Dokumentasi FTIF Festival 2016, staf Desain ITS Expo 2017, staf Departemen Media dan Informasi Himpunan Mahasiswa Teknik Computer-Informatika ITS 2017-2019, staf Biro 3D Schematics 2017 dan 2018, serta *volunteer* di ITS International Office pada tahun 2018.

Selama kuliah di Departemen Informatika ITS, penulis mengambil bidang minat Interaksi, Grafika, dan Seni (IGS) dengan ketertarikan penulis terdapat pada pengembangan *game*, AR/VR dan UI/UX. Penulis dapat dihubungi melalui surel [putrinapr@gmail.com](mailto:putrinapr@gmail.com).