



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

**APLIKASI GAME DENGAN LOCATION BASED AUGMENTED
REALITY BERBASIS PERANGKAT BERGERAK DENGAN
MENGUNAKAN KUDAN SDK**

NAUFAL BAYU FAUZAN
NRP 5113 100 175

Dosen Pembimbing I
Ridho Rahman Hariadi, S.Kom.,M.Sc.

Dosen Pembimbing II
Dr.Eng. Darlis Heru Murti, S.Kom., M.Kom

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2017

(Halaman ini sengaja dikosongkan)



TUGAS AKHIR - KI141502

**APLIKASI GAME DENGAN LOCATION BASED AUGMENTED
REALITY BERBASIS PERANGKAT BERGERAK DENGAN
MENGGUNAKAN KUDAN SDK**

NAUFAL BAYU FAUZAN
NRP 5113 100 175

Dosen Pembimbing I
Ridho Rahman Hariadi, S.Kom.,M.Sc.

Dosen Pembimbing II
Dr.Eng. Darlis Heru Murti, S.Kom., M.Kom

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2017

(Halaman ini sengaja dikosongkan)



UNDERGRADUATE THESIS - KI141502

**MOBILE-BASED GAME APPLICATION WITH LOCATION
BASED AUGMENTED REALITY WITH KUDAN SDK**

NAUFAL BAYU FAUZAN
NRP 5113 100 175

Supervisor I
Ridho Rahman Hariadi, S.Kom.,M.Sc.

Supervisor II
Dr.Eng. Darlis Heru Murti, S.Kom., M.Kom

Department of INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya, 2017

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

APLIKASI GAME DENGAN LOCATION BASED AUGMENTED REALITY BERBASIS PERANGKAT BERGERAK DENGAN MENGGUNAKAN KUDAN SDK

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Interaksi, Grafika dan Seni
Program Studi S1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

NAUFAL BAYU FAUZAN
NRP. 5113 100 175

Disetujui oleh Dosen Pembimbing Tugas Akhir

Ridho Rahman Hariadi, S.Kom., M.Kom.
NIP: 198702132014041001

Dr.Eng. Darlis Heru Murti, S.Kom., M.Kom.
NIP: 197712172003121001



SURABAYA
JUNI 2017

(Halaman ini sengaja dikosongkan)

APLIKASI GAME DENGAN LOCATION BASED AUGMENTED REALITY BERBASIS PERANGKAT BERGERAK DENGAN MENGGUNAKAN KUDAN SDK

Nama : NAUFAL BAYU FAUZAN
NRP : 5113 100 175
Jurusan : Teknik Informatika FTIf
Pembimbing I : Ridho Rahman Hariadi, S.Kom.,M.Sc.
Pembimbing II : Dr.Eng. Darlis Heru Murti, S.Kom.,
M.Kom

Abstrak

Perkembangan teknologi yang begitu pesat bukanlah suatu hal yang buruk namun sebaliknya adalah suatu hal yang baik, namun ada efek samping yang terjadi karena perkembangan teknologi yaitu terlalu fokusnya seorang individu akan teknologi yang ada. Sehingga berkurangnya interaksi secara langsung antar individu serta waktu yang diluangkan oleh individu tersebut untuk berjelajah ke lingkungan sekitar.

Salah satu teknologi yang sedang berkembang pesat belakangan ini adalah teknologi augmented reality. Augmented reality bisa berperan sebagai jembatan antara dunia maya dan dunia nyata menggunakan sebuah alat gawai yang berperan sebagai jembatan itu sendiri karena dengan adanya augmented reality ini bisa memungkinkan untuk objek-objek yang berada di dunia maya dipresentasikan pada layar alat gawai.

Dengan adanya augmented reality khususnya Location-Based Augmented Reality ini dibangun sebuah aplikasi permainan yang menggunakan teknologi Location-Based Augmented Reality ini. Yang akan menampilkan monster dalam tampilan augmented reality jika pengguna aplikasi ini berada pada beberapa tempat yang memiliki koordinat monster. Peta

yang akan ditampilkan menggunakan data dari Google Maps dan untuk pengembangan tampilan augmented reality, akan menggunakan Kudan AR SDK, serta untuk bagian back-end akan menggunakan Unity3D Game Engine.

Hasil pengujian aplikasi menunjukkan bahwa aplikasi permainan ini dapat menampilkan objek-objek dalam tampilan augmented reality serta tidak ada kebutuhan fungsional yang tidak berkerja untuk fitur-fitur dasar yang diperlukan pada aplikasi permainan ini. Dan dapat ditarik kesimpulan bahwa aplikasi permainan ini cukup nyaman untuk digunakan.

Kata-Kunci: *Augmented Reality, Kudan, Location-Based Service, ITS, Perangkat Bergerak, Unity3D, Permainan.*

MOBILE-BASED GAME APPLICATION WITH LOCATION BASED AUGMENTED REALITY WITH KUDAN SDK

Name : NAUFAL BAYU FAUZAN
NRP : 5113 100 175
Major : Informatics FTIf
Supervisor I : Ridho Rahman Hariadi, S.Kom.,M.Sc.
Supervisor II : Dr.Eng. Darlis Heru Murti, S.Kom.,
M.Kom

Abstract

Technology development which is really fast is not a bad thing but actually a good thing. But there are some drawbacks that shown up from that development such as over-focused user of a technology that make direct interaction between a person become less important and the time that they use to explore the outside becomes fewer.

As written before actually a fast technology development is a good thing, one of them is augmented reality. Augmented reality can becomes a bridge between a cyber-world and real-world with a device that take the role as a bridge itself because with this augmented reality, it is possible to presentate an cyber-world's object on device's screen which can be seen as if that object is exist on the real-world.

With this augmented reality, or more accuratedly location-based augmented reality, an game application will be developed which use location-based augmented reality technology. This game will shown a monster object if the user is located at a certain position which has a monster's coordinates. The map that will used in this game is using a data from google maps and for developing the augmented reality view, Kudan AR

SDK will be used, and lastly for the back-end will use Unity3D Game Engine. From the application testing, it is shown that this application can show the objects in augmented reality view and there is no functional requirements that is not working as the base features in this game. And it can be concluded that this application is quite good and comfortable to use.

Keyword: *Augmented Reality, Kudan, Location-Based Service, ITS, Mobile, Unity3D, Games.*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji bagi Allah Tuhan Yang Maha Esa, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **Aplikasi Game dengan Location Based Augmented Reality Berbasis Perangkat Bergerak dengan menggunakan Kudan SDK**. Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari. Selesaiannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah Tuhan Yang Maha Esa atas anugerahnya yang tidak terkira kepada penulis.
2. Kedua orang tua penulis, Bapak Imron Rosadi dan Ibu Widiastuti Listionrini S. yang selalu mencurahkan doa, dukungan semangat, perhatian, serta kasih sayang pada penulis.
3. Kakak-kakak dan adik penulis, Taufiqurrachman, Ratih Nurfitriana, dan Akbar Ramadhan yang selalu membimbing, menghibur dan juga menyemangati penulis.
4. Bapak Ridho Rahman Hariadi, S.Kom, M.sc. Selaku pembimbing Tugas Akhir pertama yang telah memberikan arahan dalam mengerjakan Tugas Akhir ini.
5. Bapak Dr. Eng. Darlis Heru Murti, S.Kom, M.Kom selaku pembimbing Tugas Akhir kedua yang dengan sabar membimbing penulis dalam mengerjakan Tugas Akhir ini.

6. Seluruh rekan-rekan sesama admin laboratorium IGS. dan juga seluruh rekan-rekan #IGSahabat terutama TC13 yang mengambil Tugas Akhir juga (Faiz, Mooy, Apep, Guruh, Purbo, Ardhan, Yahya, Zikrul, Saddam, Cayza).
7. Indra, Bayu, Asyraf, Taufik, Iky, dan Mathias selaku teman-teman Legapard ITS yang terkadang menghibur penulis.
8. Adit, Bi, Age, Dimas, Bono, Wiwit, Hanun, Shogun dan Padang serta seluruh anggota "Brigade" selaku teman-teman baik penulis yang tidak berada di Surabaya yang selalu menghibur dan menyemangati penulis.
9. Seluruh teman-teman TC13 yang telah membantu penulis dalam mengerjakan Tugas Akhir ini, terutama Akram, Bintar dan Juan.
10. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2017

Naufal B. Fauzan

DAFTAR ISI

ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR TABEL	xvii
DAFTAR GAMBAR	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Metodologi	3
1.7 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	7
2.1 Augmented Reality	7
2.2 Location Based Service	7
2.3 Unity 3D (Game Engine)	7
2.4 Global Positioning System	7
2.5 Kudan SDK	8
2.6 JSON	8
2.7 LitJSON	9
2.8 Permainan Serupa	9
BAB III DESAIN DAN PERANCANGAN	11
3.1 Analisis Sistem	11
3.1.1 Definisi Umum Perangkat Lunak	12
3.1.2 Spesifikasi Kebutuhan Perangkat Lunak	13
3.1.3 Identifikasi Pengguna	14
3.2 Perancangan Sistem	15
3.2.1 Perancangan Proses	15
3.2.2 Perancangan alur permainan	16
3.2.3 Perancangan Penyimpanan <i>game</i>	18
3.2.4 Perancangan Diagram Kasus Penggunaan	18

3.2.5 Definisi Kasus Penggunaan	19
3.2.6 Arsitektur Umum Aplikasi	22
3.2.7 Perancangan Antarmuka Aplikasi	24
3.2.8 Perancangan Aset Monster Buster	26
BAB IV IMPLEMENTASI	29
4.1 Lingkungan Implementasi	29
4.1.1 Perangkat Keras	29
4.1.2 Perangkat Lunak	29
4.2 Implementasi Antarmuka	30
4.2.1 Implementasi Antarmuka Halaman Menu Peta	30
4.2.2 Implementasi Antarmuka Halaman Pertarungan	31
4.2.3 Implementasi Antarmuka Halaman Menu Utama	32
4.3 Implementasi Alur Penggunaan Aplikasi	33
4.3.1 Implementasi Pengambilan Data Koordinat dari JSON Lokal	33
4.3.2 Implementasi Handler Data GPS	35
4.3.3 Implementasi Monster Spawner	36
4.3.4 Implementasi Pengatur Lokasi Monster	40
4.3.5 Implementasi Pengatur Peluru Pengguna	41
4.3.6 Implementasi Pencari Monster Terdekat	42
4.3.7 Implementasi Tombol Pertarungan	44
4.3.8 Implementasi Pendeteksi Lokasi	45
4.3.9 Implementasi Pertarungan	47
BAB V PENGUJIAN DAN EVALUASI	55
5.1 Lingkungan Uji Coba	55
5.2 Pengujian Fungsionalitas	55
5.2.1 Pengujian Bertarung dengan Monster dalam Tampilan Augmented Reality	55
5.2.2 Pengujian mengisi peluru amunisi	57
5.2.3 Hasil Pengujian Fungsional	59
5.3 Pengujian Pengguna	60
5.3.1 Skenario Uji Coba Pengguna	60
5.3.2 Daftar Penguji Aplikasi	61

5.3.3 Hasil Uji Coba Pengguna	62
BAB VI KESIMPULAN DAN SARAN	63
6.1 Kesimpulan	63
6.2 Saran	63
DAFTAR PUSTAKA	65

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 3.1 Karakteristik Pengguna	15
Tabel 3.2 Skenario Kasus Penggunaan	19
Tabel 3.3 Spesifikasi Kasus Penggunaan Bertarung dengan monster dalam tampilan augmented reality	20
Tabel 3.4 Spesifikasi Kasus Penggunaan Mengisi Peluru Amunisi	21
Tabel 5.1 Pengujian Bertarung dengan Monster dalam Tampilan Augmented Reality Skenario 1	56
Tabel 5.2 Pengujian Bertarung dengan Monster dalam Tampilan Augmented Reality Skenario 2	57
Tabel 5.3 Pengujian Mengisi Peluru Amunisi Skenario 1	58
Tabel 5.4 Pengujian Mengisi Peluru Amunisi Skenario 2	59
Tabel 5.5 Rekapitulasi hasil pengujian fungsional	60
Tabel 5.6 Daftar Nama Penguji Coba Aplikasi	61
Tabel 5.7 Rangkuman Hasil Formulir Pengujian Aplikasi	62

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2.1 <i>Screenshot</i> Pokemon GO	9
Gambar 3.1 Diagram Alur Proses Permainan	16
Gambar 3.2 Diagram Kasus Penggunaan	19
Gambar 3.3 Rancangan Sederhana Arsitektur Aplikasi	23
Gambar 3.4 Rancangan Arsitektur MVC	24
Gambar 3.5 Rancangan Antarmuka Menu Peta	25
Gambar 3.6 Rancangan Antarmuka Pertarungan	26
Gambar 3.7 Model Objek 3D dari <i>Rabbit</i> (kiri), <i>Ghost</i> (Kanan) dan <i>Bat</i> (Atas)	28
Gambar 4.1 Antarmuka Halaman Menu Peta	31
Gambar 4.2 Antarmuka Halaman Pertarungan	32
Gambar 4.3 Antarmuka Halaman Menu Utama	33

(Halaman ini sengaja dikosongkan)

DAFTAR KODE SUMBER

Kode Sumber 4.1	Kode Untuk Membaca JSON	34
Kode Sumber 4.2	Contoh Format JSON	34
Kode Sumber 4.3	Kode Untuk Handler Data GPS	35
Kode Sumber 4.4	Kode Untuk Monster Spawner	36
Kode Sumber 4.5	Kode Untuk Bullet Spawner	38
Kode Sumber 4.6	Kode Untuk Pengatur Lokasi Monster	40
Kode Sumber 4.7	Kode Untuk Pengatur Lokasi Peluru	40
Kode Sumber 4.8	Kode Untuk Pengatur Peluru Pengguna	42
Kode Sumber 4.9	Kode Untuk Pengatur Peluru Pada Peta	42
Kode Sumber 4.10	Kode Untuk Pencari Monster Terdekat	43
Kode Sumber 4.11	Kode Untuk Tombol Pertarungan	44
Kode Sumber 4.12	Kode Untuk Pendeteksi Lokasi	46
Kode Sumber 4.13	Kode Untuk Pertarungan Bagian Monster	47
Kode Sumber 4.14	Kode Untuk Tombol <i>Shoot</i> dan <i>Flee</i>	48
Kode Sumber 4.15	Kode Untuk Pergerakan Peluru	49
Kode Sumber 4.16	Kode Untuk <i>Controller</i> Pertarungan	49
Kode Sumber 4.17	Kode Untuk <i>View</i> Pertarungan	51

(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

Perkembangan aplikasi *Augmented Reality* sudah dimulai sejak tahun 1968, dan selama tiga dekade terakhir ini perkembangannya masih terlihat menjanjikan dan juga menantang, serta ditambah dengan adanya perkembangan komputer serta *mobile platforms* yang pesat seperti *smart phone*, *Augmented Reality* ini menjadi lebih terkenal [1], yaitu *Pose Computation* dan *Object Recognition*. Namun kebanyakan dari *Augmented Reality* ini sendiri menggunakan metode *Object Recognition*. Namun, pendekatan ini hanya akan berkerja jika pengguna mengarahkan kamera ke objek yang telah dikenal oleh *Augmented Reality* ini sendiri [2]. Pada metode kedua yang dikenal dengan *Pose Computation* adalah metode yang lebih fleksibel yang mana metode ini berfokus pada posisi sensor kamera yang nantinya akan memproyeksikan *3D Object*. Sensor-sensor yang dapat mendukung metode ini ada beberapa seperti GPS, GSM, *Wireless-LAN*, dan masih banyak lagi. Dan untuk jarak yang sangat jauh, GPS adalah sensor terbaik yang digunakan dengan memanfaatkan *Location-Based Service* atau yang biasa dikenal dengan LBS [3]. Dengan adanya LBS ini, jika sensor GPS berada pada tempat yang telah ditentukan sebelumnya maka *3D Object* akan diproyeksikan pada kamera, metode ini biasa dikenal sebagai *Location-Based Augmented Reality*. *Augmented Reality* ini sendiri juga memiliki banyak kegunaan seperti permainan, pembelajaran, *tourist guide*, navigasi, dan banyak lagi [4].

‘Monster Buster’ merupakan sebuah *game* yang menitik beratkan pada penggunaan *Location-Based Augmented Reality*. *Game* ini sendiri juga akan dirancang berbasis Android. Sebagai *game* yang menggunakan LBS maka GPS adalah hal yang sangat diperlukan pada *game* ini, karena *game* ini akan menggunakan informasi *latitude* dan *longitude* yang akan digunakan sebagai titik yang menjadi posisi monster. Sehingga ketika pengguna telah berada di titik koordinat yang sesuai nilai *latitude* dan *longitude*-nya, pengguna bisa berinteraksi dengan *3D Object* yang telah diproyeksikan.

Tujuan dari pengerjaan Tugas Akhir ini adalah mampu menghasilkan sebuah *game* dengan berbasis Android dengan mengimplementasikan teknologi *Location-Based Augmented Reality*. Selain itu, juga diharapkan dengan adanya aplikasi Tugas Akhir ini mahasiswa-mahasiswa ITS tidak akan hanya berdiam diri pada jurusan masing-masing namun juga menjelajahi tempat-tempat lain yang berada di ITS dan juga berkenalan dengan sesama pemain.

1.2 Rumusan Masalah

Berikut adalah beberapa hal yang menjadi rumusan masalah dalam tugas akhir ini:

1. Bagaimana cara mengimplementasikan teknologi *Location-Based Augmented Reality* dalam *game* berbasis Android?
2. Bagaimana mengintegrasikan koordinat dari keadaan nyata kepada aplikasi yang dibangun?
3. Bagaimana menggunakan informasi *latitude* dan *longitude* dari pengguna dan data informasi yang ditentukan untuk memproyeksikan objek jika kedua data tidak terlalu jauh dari satu dan yang lainnya?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan masalah, sebagai berikut:

1. Aplikasi yang dibuat merupakan aplikasi yang berjalan di Android.
2. Lingkungan pengembangan yang digunakan menggunakan aplikasi Unity 3D 5.6.1 *Free License* dan bahasa pemrograman C#.
3. Studi kasus yang digunakan hanyalah lingkungan ITS.
4. Aplikasi hanya dapat memperhitungkan posisi tempat secara horizontal (*longitude* dan *latitude*), tidak dapat memperhitungkan ketinggian tempat (*altitude*).

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah membangun *game* berbasis perangkat bergerak Android dengan menggunakan teknologi *Location-Based Augmented Reality*.

1.5 Manfaat

Manfaat dari penelitian tugas akhir ini adalah terciptanya *game* yang akan dapat digunakan oleh orang-orang yang sedang berada di ITS untuk lebih menikmati lingkungan ITS.

1.6 Metodologi

1. Penyusunan proposal tugas akhir
Tahap pertama dalam proses pengerjaan tugas akhir ini adalah menyusun proposal Tugas Akhir. Pada proposal tugas akhir ini telah diajukan *game* berbasis perangkat bergerak dengan menggunakan teknologi *Location Based*

Augemented Reality pada sistem operasi Android.

2. Studi literatur

Pada tahap ini, akan dicari studi literatur yang relevan untuk dijadikan refrensi dalam pengerjaan tugas akhir. Studi literatur ini didapatkan dari buku, internet, dan materi-materi perkuliahan yang berhubungan dengan metode yang akan digunakan.

3. Analisis dan desain perangkat lunak

Perangkat lunak yang akan dibangun merupakan aplikasi untuk perangkat bergerak. Akan ada tiga macam tampilan dari aplikasi ini. Tampilan pertama adalah menu utama yang mana ini adalah tampilan yang akan pertama kali dilihat oleh pengguna, tampilan kedua ialah menu peta, yang mana ialah menu yang akan menampilkan peta dimana posisi pengguna serta posisi monster-monster dan peluru yang akan dilawan dan digunakan di permainan, dan terakhir ialah tampilan pertarungan yang menggunakan kamera dan menampilkan objek *Augmented Reality*.

4. Pengembangan perangkat lunak

Pembangunan aplikasi akan dilakukan menggunakan bahasa pemrograman C#, *Unity3D Game Engine*, dan *Kudan SDK*.

5. Pengujian dan evaluasi

Pengujian akan dilakukan oleh sepuluh orang pengguna. Pengguna akan diminta mengisi lembar formulir pengujian aplikasi untuk memberikan umpan balik. Kemudian akan dilakukan evaluasi untuk memeriksa ketepatan pembacaan dan pemrosesan koordinat *latitude* dan *longitude* oleh

aplikasi dan apakah sudah nyaman digunakan oleh pengguna.

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 metode dan algoritma 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
 - g. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Perancangan
4. Implementasi
5. Pengujian dan Evaluasi
6. Kesimpulan dan Saran
7. Daftar Pustaka

1.7 Sistematika Penulisan

Buku tugas akhir ini disusun dengan sistematika penulisan sebagai berikut:

BAB I. PENDAHULUAN

Bab ini berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu, permasalahan, batasan

masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

BAB II. TINJAUAN PUSTAKA

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tuas akhir ini.

BAB III. DESAIN DAN PERANCANGAN

Bab ini membahas tahap analisis permasalahan dan perancangan dari sistem yang akan dibangun. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan tugas akhir.

BAB IV. IMPLEMENTASI

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya.

BAB V. PENGUJIAN DAN EVALUASI

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

BAB VI. KESIMPULAN DAN SARAN

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II

TINJAUAN PUSTAKA

2.1 Augmented Reality

Augmented Reality atau AR adalah sebuah teknologi yang menampilkan atau memproyeksikan informasi-informasi dengan menggunakan sensor-sensor seperti suara, video, gambar atau data GPS. Informasi yang ditampilkan bisa berupa gambar, objek, atau lokasi [5].

2.2 Location Based Service

LBS atau *Location-Based Service* ialah *software-level service* yang menggunakan *location data*. LBS mempunyai beberapa kegunaan seperti mengidentifikasi lokasi dari seseorang atau sebuah objek, seperti mesin ATM atau keberadaan teman atau orang lain. Layanan LBS ini menggunakan informasi-informasi geografis yang ada pada sebuah perangkat [6].

2.3 Unity 3D (Game Engine)

Unity merupakan sebuah *Game Engine* yang dikembangkan oleh Unity Technologies dan digunakan untuk mengembangkan permainan untuk PC, namun tak hanya pada PC unity juga bisa digunakan untuk mengembangkan permainan berbasis konsol, *mobile devices* dan juga website. Untuk *mobile devices* tidak hanya tertutup pada Android namun game pada iOS juga bisa dikembangkan menggunakan Unity ini. Hingga saat ini sudah terdaftar 21 Platform yang dapat dikembangkan menggunakan Unity [7].

2.4 Global Positioning System

GPS atau *Global Positioning System* adalah *Global Navigation Satellite System* yang menyediakan *geolocation* dan

informasi waktu kepada GPS *receiver*. Pada awalnya GPS ini adalah untuk keperluan militer namun seiring berjalannya waktu GPS sudah bisa dipakai oleh semua orang [8].

2.5 Kudan SDK

Kudan SDK merupakan sebuah *library* yang digunakan untuk membangun aplikasi yang menggunakan teknologi *Augmented Reality* yang dirilis oleh Kudan Computer Vision. Kudan SDK ini berisi *library* untuk *location based augmented reality* atau *markerless augmented reality*, dan juga *image recognition augmented reality* atau *marker augmented reality* [9].

Kudan SDK ini adalah salah satu *Augmented Reality Software Development Kit* yang digunakan untuk membangun aplikasi dengan teknologi *Augmented Reality* pada *mobile devices*. Kudan SDK ini sendiri juga memiliki banyak fitur, seperti digunakan untuk mengembangkan *marker* atau *markerless tracking and location requirements*. Untuk *Augmented Reality* dengan *marker* Kudan SDK ini juga mendukung sampai 15000 *local marker*.

Selain Kudan SDK sendiri, Kudan Computer Vision juga merilis Kudan SLAM, yang menghubungkan antara *Artificial Intelligence* dan *plethora of devices and sensors* yang kita kenali sebagai *Internet of Things* atau IoT. Ini memberikan komputer kemampuan untuk mendapatkan, memproses, menganalisa, mengerti mengenai *digital images* dan juga kemampuan untuk memetakan lingkungan serta memahami lokasi didalamnya.

2.6 JSON

JSON (*JavaScript Object Notation*) adalah sebuah *lightweight data-interchange format*. JSON ini mudah untuk disimpan dan dibaca oleh komputer. JSON berbasis pada *subset* dari *JavaScript*

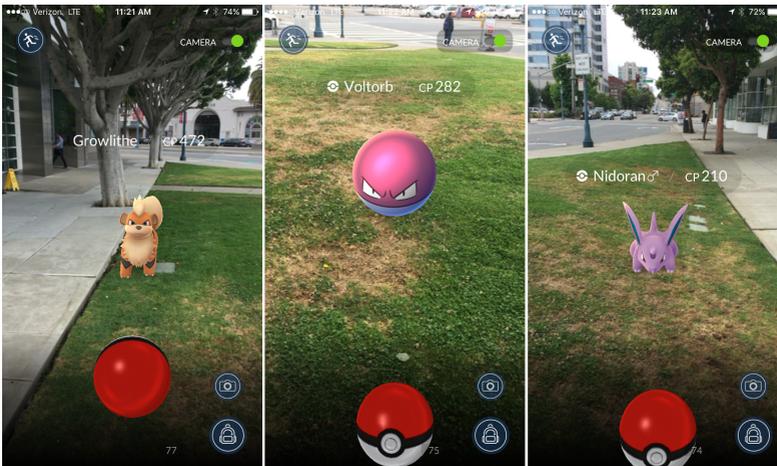
Programming Language. JSON adalah sebuah teks format yang merupakan *language independent* namun dapat digunakan oleh *C-family of languages*.

2.7 LitJSON

LitJSON adalah sebuah *library* yang kecil dan cepat yang mana digunakan untuk mengatur data yang tersimpan dalam format JSON. LitJSON ini terbuat dalam bahasa C# dan juga bisa digunakan oleh semua bahasa .NET.

2.8 Permainan Serupa

Dalam pembuatan game "Monster Buster" ini terinspirasi dari Pokemon GO. Pokemon GO adalah permainan *multiplayer* yang bertema *location based augmented reality* yang dibuat untuk iOS dan Android [10]. Tampilan bermain pada Pokemon GO bisa dilihat pada Gambar 2.1.



Gambar 2.1 Screenshot Pokemon GO

Pokemon GO ini adalah hasil kerja sama dari The Pokemon Company, Nintendo dan Niantic, Inc. Game ini bisa diunduh secara gratis dengan tambahan pembelian di dalam aplikasi. Game ini diumumkan pada *press conference* di Jepang pada 10 September 2015 dan telah dirilis pada tanggal 6 Juli 2016.

BAB III

DESAIN DAN PERANCANGAN

Pada bab ini dibahas mengenai analisis dan perancangan *game* dengan *location based augmented reality* berbasis perangkat bergerak dengan menggunakan Kudan SDK. Pembahasan yang dilakukan meliputi analisis fitur yang dibutuhkan dan perancangan perangkat.

3.1 Analisis Sistem

Dari latar belakang yang ditulis pada Subbab 1.1, dapat diketahui bahwa *game* yang akan dibangun merupakan perpaduan dari *2D games* dan *3D games*, 2D pada dibagian peta dan 3D pada bagian interaksi atau pertarungan dengan monsternya karena menggunakan teknologi *Augmented Reality*. *Game* ini akan dibangun pada perangkat bergerak dengan sistem operasi Android.

Game dan *Augmented Reality* adalah sebuah teknologi yang dapat memberikan suasana baru yang tidak biasa karena perpaduan antara dunia digital dengan dunia nyata yang mana menjadi suasana yang menarik. *Game* yang akan dibangun ini akan menggunakan *Augmented Reality Software Development Kit* dari Kudan, yang mana akan menawarkan teknologi *marker* maupun *markerless Augmented Reality*. Untuk membangun *markerless augmented reality* ini diperlukan sebuah sensor yang akan menjadi *trigger* dan pada *game* ini sensor yang digunakan sebagai *trigger* ialah sebuah sensor GPS.

Game engine yang digunakan untuk membangun *game* ini ialah *Unity3D Game Engine*. *Game engine* ini mudah digunakan untuk mengembangkan permainan 3D dan juga satu satunya *game engine* yang didukung oleh Kudan SDK yang digunakan sebagai penyedia fitur *markerless Augmented Reality*. Selain itu *Unity3D* ini juga menyediakan *cross-platform development* yang mana mempermudah ketika melakukan *debugging game*

Android ketika masih pada fase perancangan pada Windows.

3.1.1 Definisi Umum Perangkat Lunak

Pada Tugas Akhir ini akan dibangun *game* Monster Buster dengan menggunakan teknologi *Location Based Augmented Reality* yang menggunakan Kudan SDK. *Game* ini akan dirancang untuk digunakan pada perangkat bergerak dengan sistem operasi Android, karena penggunaan perangkat bergerak ini akan memudahkan dalam memainkan permainan ini seperti pengambilan data GPS dan menggunakan kamera untuk fungsi *Augmented Reality*.

Game ini direncanakan untuk dititik beratkan pada interaksi pengguna dengan objek yang akan dimunculkan pada *Augmented Reality*. *Augmented Reality* disini bersifat *markerless* jadi pengguna harus menggunakan sensor GPS dan lokasi koordinat monster-monster yang sudah dideklarasikan sebelumnya untuk men-*trigger Augmented Reality* ini. Jadi permainan ini akan menggunakan *Maps* dari Google untuk membantu dalam mencari atau mendekati lokasi koordinat monster-monster yang ada, namun ketika sudah mendekati monster, pengguna akan diberi kesempatan untuk berinteraksi dengan monster-monster ini dengan cara bertarung walaupun sudah dekat dengan lokasi koordinat monster yang dituju masih akan dilakukan pengecekan jarak antara pengguna dengan koordinat monster yang mana jika terlalu jauh tidak akan men-*trigger Augmented Reality* ini. Pada peta akan tersebar peluru yang bisa digunakan dan peluru ini digunakan untuk menyerang monster pada pertarungan.

Pada saat *game* ini dibuka pengguna akan ditunjukkan menu utama yang mana akan memberikan pilihan untuk bermain. Ketika pengguna memilih untuk bermain maka pengguna akan masuk kepada halaman peta permainan yang mana adalah peta dunia nyata, pada peta tersebut akan ditunjukkan lokasi-lokasi

koordinat monster dan peluru yang dekat dengan pengguna. Selain itu halaman peta ini juga menampilkan berbagai macam informasi seperti jumlah peluru yang dimiliki pengguna saat ini, nilai koordinat *latitude* dan *longitude* pengguna saat ini, dan monster terdekat dengan pengguna. Jika pengguna mendekati peluru dan jumlah peluru yang dimiliki oleh pengguna kurang dari maksimal maka jumlah peluru akan bertambah dan jika mendekati monster akan menampilkan tombol untuk masuk ke menu pertarungan atau menu *Augmented Reality*.

Pada menu pertarungan ini akan dilakukan pengecekan lokasi jika pengguna berada didekat monster dan dalam jarak yang diizinkan oleh aplikasi maka akan men-*trigger Augmented Reality* untuk menampilkan objek yang mana adalah monster. Setelah berhasil ditampilkan maka pengguna diminta untuk mengalahkan monster dengan cara menyerangnya dengan senapan secara *First Person Shooter*. Namun, target dimana peluru akan diluncurkan akan berpindah secara terus-menerus sehingga pengguna diminta untuk fokus pada permainan untuk dapat memenangkan pertarungan. Pertarungan akan berakhir ketika monster kehabisan *Health Point* atau poin nyawa, atau ketika pengguna kehabisan peluru atau memilih menu untuk kabur dari pertarungan.

3.1.2 Spesifikasi Kebutuhan Perangkat Lunak

Kebutuhan sistem yang dibuat ini melibatkan dua hal, yaitu kebutuhan fungsional dan non-fungsional. Dua kebutuhan tersebut akan dijelaskan lebih lanjut pada Subbab 3.1.2.1 dan Subbab 3.1.2.2.

3.1.2.1 Kebutuhan Fungsional Perangkat Lunak

Berdasarkan deskripsi umum aplikasi, maka disimpulkan bahwa kebutuhan fungsional dari aplikasi ini adalah sebagai

berikut:

1. Bertarung dengan monster dalam tampilan *augmented reality*.
2. Mengisi peluru amunisi.

3.1.2.2 Kebutuhan Non-Fungsional Perangkat Lunak

Selain memiliki kebutuhan fungsional, sebuah aplikasi juga memerlukan kebutuhan non-fungsional yang mana tidak wajib terpenuhi namun ketika terpenuhi akan meningkatkan kualitas dari aplikasi yang akan dibangun.

1. Frame Rate

Untuk *game* perlulah untuk bisa mencegah tidak terjadinya *low frame rate*. Dan kejadian ini rawan terjadi pada *game* 3D, yang mana akan mengganggu jalannya aplikasi. Hal ini terjadi karena banyaknya objek 3D yang di-*render* dengan pola-pola yang rumit, sehingga diperlukan adanya pertimbangan untuk memilih objek 3D yang akan digunakan.

2. Koneksi Internet

Game ini menggunakan koneksi internet untuk mendapatkan data dari sensor GPS yang sangat penting. Sehingga aplikasi yang dibangun perlu mempertimbangkan koneksi internet yang tidak stabil.

3.1.3 Identifikasi Pengguna

Berdasarkan definisi umum perangkat lunak pada Subbab 3.1.1, maka dapat diketahui bahwa pengguna yang akan menggunakan aplikasi ini hanya satu orang, yaitu pengguna yang

menjalankan aplikasi. Karakteristik pengguna tercantum dalam Tabel 3.1.

Tabel 3.1 Karakteristik Pengguna

Nama Aktor	Tugas	Hak akses aplikasi	Kemampuan yang harus dimiliki
Pengguna	Pihak luar yang mencoba aplikasi	Menjalankan aplikasi	Tidak ada

3.2 Perancangan Sistem

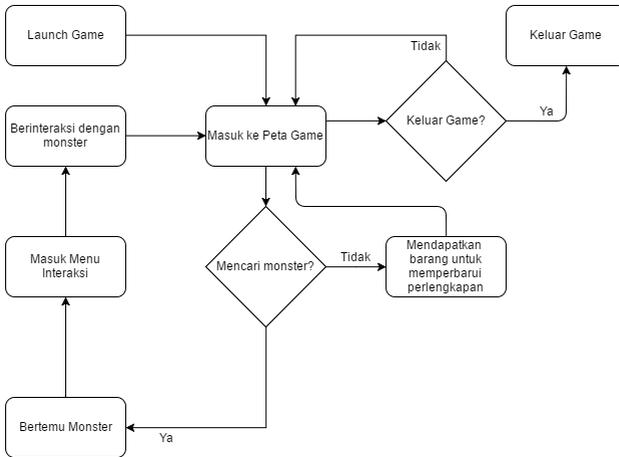
Tahap perancangan pada bab ini dibagi menjadi beberapa bagian yaitu perancangan proses, perancangan alur permainan, perancangan diagram kasus penggunaan, definisi kasus penggunaan, arsitektur umum aplikasi, perancangan antarmuka aplikasi dan perancangan aset aplikasi.

3.2.1 Perancangan Proses

Pada subbab ini akan dijelaskan mengenai rancangan proses yang dilakukan untuk mendukung fungsionalitas yang sudah dirancang pada *game*. Proses ini penting agar perangkat lunak dapat berjalan secara baik dan benar.

Secara umum alur proses pada *game* ini hanya satu. Dimana pengguna akan membuka permainan dan memasuki menu peta, lalu akan terjadi percabangan apakah pengguna akan mencari monster untuk melakukan pertarungan, jika tidak maka akan dianggap bahwa pengguna akan mencari barang yang digunakan untuk bertarung dengan monster. Jika pengguna ingin mencari monster untuk diajak bertarung, maka pengguna akan bertemu

dengan monster lalu bertarung dengan monster hingga pertarungan selesai dan kembali lagi ke menu peta. Alur proses ini bisa dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alur Proses Permainan

3.2.2 Perancangan alur permainan

Pada subbab ini skenario permainan dan alur permainan pada *game* yang ingin dibangun.

3.2.2.1 Skenario *game*

”Monster Buster” adalah *game* dengan tema aksi. *Game* ini dimainkan oleh satu orang. Target dari *game* ini adalah mengalahkan semua monster yang berada pada peta. Pada peta terdapat tiga macam monster yaitu *Rabbit*, *Ghost* dan *Bat*. Monster yang ada akan hilang pada peta jika pengguna telah mencoba bertarung dengan monster tersebut. Pada setiap pertarungan monster darah berupa *Health Point*. Pengguna dapat mengurangi jumlah *Health Point* dari musuh dengan cara

menembakan peluru milik pengguna yang mempunyai nilai maksimal lima. Jumlah peluru yang pengguna bisa gunakan pada setiap pertarungan sesuai dengan jumlah peluru yang dimiliki oleh pengguna pada saat berada di halaman peta sebelum pindah ke halaman pertarungan dengan monster. Setiap peluru yang mengenai monster akan mengurangi jumlah *Health Point* sebanyak satu poin dari maksimal tiga poin. Pengguna bisa mengisi peluru dengan cara mengoleksi peluru yang berada pada peta.

3.2.2.2 Aturan Main

Game ini memiliki beberapa aturan main, yaitu:

- Setiap kali pengguna memasuki halaman peta pasti akan ada monster dan peluru yang terpanggil.
- Semua monster akan terpanggil ulang ketika semua monster telah hilang dari peta.
- Semua peluru akan terpanggil ulang ketika semua peluru telah hilang dari peta.
- Pengguna dapat menggerakkan dirinya pada game dengan cara berjalan di dunia nyata.
- Pengguna dapat menantang musuh untuk bertarung ketika jarak dari pengguna dan monster kurang dari lima belas meter.
- Pengguna dapat pergi dari area pertarungan dan membuat monster kabur dengan cara berpindah sehingga jarak antara pengguna dan monster mencapai lima belas meter.
- Pengguna dapat mengalahkan monster dengan cara menembakan peluru dan mengenai monster sebanyak tiga buah peluru.
- Pengguna dapat kalah dari pertarungan ketika kehabisan peluru dalam satu pertarungan.

3.2.2.3 Perancangan Penghitung Jarak

Sebagai pendukung fitur permainan akan dirancang penghitung jarak antara monster terdekat dan pengguna. Karena keduanya memiliki nilai koordinat latitude dan longitude maka akan digunakan Harvesine Formula yang bisa menghitung jarak antar dua titik menggunakan nilai koordinat latitude dan longitude.

$$a = \sin^2(\Delta\phi/2) + \cos \phi_1 \cos \phi_2 \sin^2(\Delta\lambda/2)$$

$$c = 2 \operatorname{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R c$$

ϕ = latitude

λ = longitude

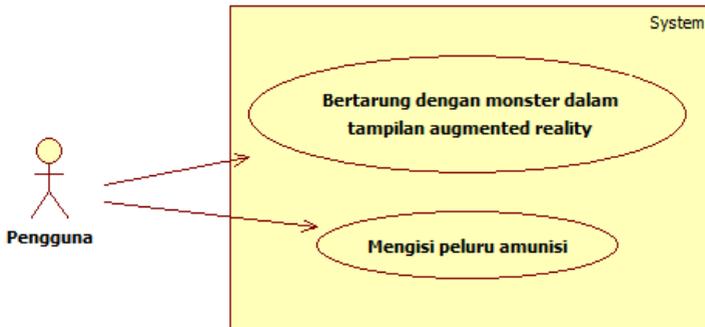
R = Jari-jari bumi

3.2.3 Perancangan Penyimpanan *game*

Dalam *game* ini akan digunakan media penyimpanan PlayerPrefs yang telah disediakan oleh Unity3D. Dengan adanya PlayerPrefs ini bisa dilakukan penyimpanan serta pembacaan secara lokal.

3.2.4 Perancangan Diagram Kasus Penggunaan

Dalam *game* Tugas Akhir ini, terdapat dua kasus penggunaan yang ada yaitu bertarung dengan monster dalam tampilan *augmented reality* dan mengisi peluru amunisi. Rancangan kasus penggunaan dapat dilihat pada Gambar [3.2](#) dan Tabel [3.2](#).



Gambar 3.2 Diagram Kasus Penggunaan

Tabel 3.2 Skenario Kasus Penggunaan

Kode Kasus Penggunaan	Nama Kasus Penggunaan
UC-001	Bertarung dengan monster dalam tampilan <i>augmented reality</i>
UC-002	Mengisi peluru amunisi

3.2.5 Definisi Kasus Penggunaan

Setiap kasus penggunaan memiliki detail untuk setiap tahapnya. Detail mengenai kasus penggunaan tersebut dapat dilihat pada Subbab 3.2.4.1 dan Subbab 3.2.4.2.

3.2.5.1 Bertarung Dengan Monster Dalam Tampilan *Augmented Reality*

Pada kasus penggunaan bertarung dengan monster dalam tampilan *augmented reality* ini, pengguna bisa memilih monster

mana yang akan diajak bertarung. Ada tiga tipe monster yang bisa diajak bertarung, yaitu: *Rabbit*, *Ghost* dan *Bat*. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.3.

Tabel 3.3 Spesifikasi Kasus Penggunaan Bertarung dengan monster dalam tampilan augmented reality

Nama Kasus Penggunaan	Bertarung Dengan Monster Dalam Tampilan <i>Augmented Reality</i>
Nomor	UC-001
Deskripsi	Kasus penggunaan aktor untuk bertarung dengan monster dalam tampilan <i>augmented reality</i>
Aktor	Pengguna
Kondisi Awal	Pengguna sedang berada pada menu peta di permainan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna berjalan mendekati monster pada peta 2. Pengguna sudah berada didekat koordinat monster 3. Aplikasi menunjukkan tombol bertarung 4. Pengguna menekan tombol bertarung ketika jarak antara pengguna dan monster tidak lebih dari lima belas meter 5. Aplikasi akan menampilkan <i>scene</i> pertarungan sesuai dengan monster yang ditemui

Alur Alternatif	A1. Jarak antara pengguna dan monster lebih atau setara dengan lima belas meter 1. Aplikasi akan menampilkan <i>scene</i> pertarungan sesuai dengan monster. Namun, akan memberitahu pengguna bahwa monster telah kabur dari pertarungan
Kondisi Akhir	<i>Game</i> menampilkan halaman pertarungan dengan monster dalam <i>augmented reality</i>

3.2.5.2 Mengisi Peluru Amunisi

Spesifikasi kasus penggunaan mengisi peluru amunisi dapat dilihat pada Tabel 3.4.

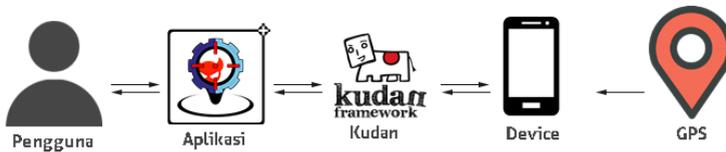
Tabel 3.4 Spesifikasi Kasus Penggunaan Mengisi Peluru Amunisi

Nama Kasus Penggunaan	Mengisi Peluru Amunisi
Nomor	UC-002
Deskripsi	Kasus penggunaan aktor untuk mengisi amunisi peluru
Aktor	Pengguna
Kondisi Awal	Pengguna sedang berada pada menu peta di permainan dan jumlah peluru yang dimiliki pengguna kurang dari lima

Alur Normal	<ol style="list-style-type: none"> 1. Pengguna berjalan mendekati peluru pada peta 2. Pengguna menyentuh objek peluru pada peta 3. Petunjuk pada layar menunjukkan jumlah peluru bertambah
Alur Alternatif	<p>A1. Jumlah peluru pengguna setara dengan lima</p> <ol style="list-style-type: none"> 1. Peluru pada peta akan menghilang seperti digunakan namun jumlah peluru pengguna tidak akan bertambah karena sudah mencapai nilai maksimal
Kondisi Akhir	<p><i>Game</i> menampilkan halaman pertarungan dengan monster dalam <i>augmented reality</i></p>

3.2.6 Arsitektur Umum Aplikasi

Arsitektur umum pada *game* ini menggunakan teknologi *Location Based Augmented Reality* dan didukung dengan beberapa perangkat seperti kamera *smartphone* dan koordinat GPS. Implementasi aplikasi menggunakan *Unity3D Game Engine* dan untuk konten *Augmented Reality* menggunakan *Kudan*. Arsitektur umum aplikasi yang akan dibuat dapat dilihat pada Gambar [3.3](#).

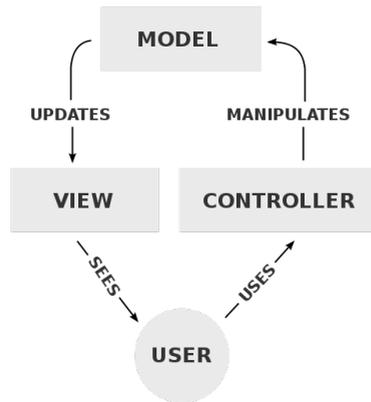


Gambar 3.3 Rancangan Sederhana Arsitektur Aplikasi

Dari Gambar 3.3 bisa dilihat ada lima entitas pada arsitektur. Singkatnya aplikasi ini akan dimulai dari pengambilan data GPS oleh perangkat yang mana ini akan berfungsi sebagai alternatif dari *marker* untuk *augmented reality* yang akan bekerja pada aplikasi melewati Kudan.

Secara umum aplikasi akan dibangun seperti yang terlihat pada Gambar 3.3, namun tipe arsitektur yang akan dipakai saat membangun tiap *scene* pada aplikasi akan menggunakan metode arsitektur MVC atau *Model-View-Controller*. Gambaran mengenai MVC bisa dilihat pada Gambar 3.4.

Pada arsitektur MVC ini akan terlibat tiga entitas yaitu *model*, *controller* dan *view*. Dimana setiap kali pengguna melakukan sesuatu pada aplikasi akan selalu menggunakan *controller* yang mana akan memanipulasikan *model* dan *model* ini bisa merubah apa yang akan ditampilkan oleh *view*.

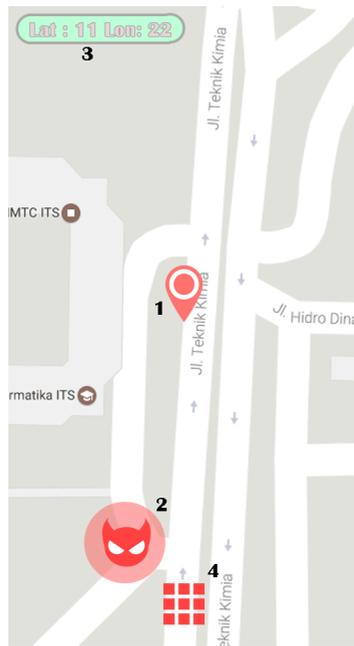


Gambar 3.4 Rancangan Arsitektur MVC

3.2.7 Perancangan Antarmuka Aplikasi

Rancangan antarmuka aplikasi diperlukan untuk memberikan gambaran umum kepada pengguna bagaimana sistem yang ada dalam aplikasi ini berinteraksi dengan pengguna. Selain itu, rancangan ini juga memberikan gambaran bagi pengguna apakah tampilan yang dirancang mudah dipahami dan digunakan. Secara garis besar antarmuka yang didesain ada dua dan cukup sederhana. Dari dua antarmuka yang didesain ada antarmuka menu peta dan antarmuka pertarungan.

Menu peta ialah tampilan utama dari *game* ini, di mana antarmuka yang berfungsi sebagai penunjuk lokasi pengguna berada pada peta, lokasi monster yang ada dan juga informasi terkait pengguna. Perancangan menu peta dapat dilihat di Gambar [3.5](#).



Gambar 3.5 Rancangan Antarmuka Menu Peta

Berikut penjelasan komponen antarmuka menu peta berdasarkan Gambar **3.5**:

1. Posisi pengguna.
2. Posisi mahluk digital.
3. Keterangan status pengguna.
4. Tombol menu (jika diperlukan).

Antarmuka pertarungan ialah tampilan antarmuka yang akan dibuka pengguna ketika akan bertarung dengan mahluk digital. Pada antarmuka ini akan terlihat pilihan-pilihan yang berhubungan dengan pertarungan dengan monster digital, dan juga akan menampilkan objek monster dalam bentuk 3D secara *augmented reality*. Perancangan antarmuka pertarungan ini dapat

dilihat di Gambar 3.6.



Gambar 3.6 Rancangan Antarmuka Pertarungan

Berikut penjelasan komponen antarmuka pertarungan berdasarkan Gambar 3.6:

1. Status monster yang dilihat.
2. Tombol interaksi.
3. Tombol menu (jika diperlukan).

3.2.8 Perancangan Aset Monster Buster

Pada subbab ini akan dibahas rancangan-rancangan aset yang digunakan untuk *game* ini. Beberapa aset yang digunakan pada *game* ini adalah *free assets* dan penulis ambil dari Unity Asset Store.

3.2.8.1 Aset Monster *Rabbit*

Untuk aset monster *Rabbit* penulis menggunakan aset milik PI Entertainment Limited pada Unity Asset Store. Aset yang ada ini sudah menyediakan objek 3D serta dengan animasinya yaitu animasi terkena serangan serta animasi ketika monster telah kalah. Aset monster *Rabbit* ini bisa dilihat pada Gambar [3.7](#).

3.2.8.2 Aset Monster *Ghost*

Untuk aset monster *Ghost* penulis menggunakan aset milik PI Entertainment Limited pada Unity Asset Store. Aset yang ada ini sudah menyediakan objek 3D serta dengan animasinya yaitu animasi terkena serangan serta animasi ketika monster telah kalah. Aset monster *Ghost* ini bisa dilihat pada Gambar [3.7](#).

3.2.8.3 Aset Monster *Bat*

Untuk aset monster *Bat* penulis menggunakan aset milik PI Entertainment Limited pada Unity Asset Store. Aset yang ada ini sudah menyediakan objek 3D serta dengan animasinya yaitu animasi terkena serangan serta animasi ketika monster telah kalah. Aset monster *Bat* ini bisa dilihat pada Gambar [3.7](#).



Gambar 3.7 Model Objek 3D dari *Rabbit* (kiri), *Ghost* (Kanan) dan *Bat* (Atas)

BAB IV

IMPLEMENTASI

Pada bab ini dibahas mengenai implementasi dari perancangan *game*. Di dalamnya mencakup proses penerapan dan pengimplementasian dalam bentuk kode sumber dan antarmuka yang mengacu pada rancangan yang telah dibahas pada Subbab 3.2.

4.1 Lingkungan Implementasi

Lingkungan implementasi merupakan lingkungan dimana aplikasi akan dibangun dan lingkungan implementasi dibagi menjadi dua, yaitu lingkungan implementasi berupa perangkat keras dan lingkungan implementasi berupa perangkat lunak.

4.1.1 Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan aplikasi ini adalah komputer dengan spesifikasi sebagai berikut:

- Sony Vaio Fit E SVF15221P2E
- Prosesor Intel Core i5 3rd Gen. (2.4GHz)
- Memori 8182 MB DDR3L

Sedangkan untuk melakukan proses *debugging*, digunakan perangkat bergerak dengan spesifikasi sebagai berikut:

- Asus Zenfone 2 ZE500CL
- Prosesor Intel Atom Z2560
- Memori 2048 MB

4.1.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan aplikasi ini adalah sebagai berikut:

- Sistem Operasi Windows 10 Home 64-bit
- IDE Unity3D Game Engine
- Text Editor MonoDevelop

- AR SDK Kudan SDK

Sedangkan untuk melakukan proses *debugging*, digunakan perangkat bergerak dengan spesifikasi sebagai berikut:

- Sistem Operasi Android 5.0

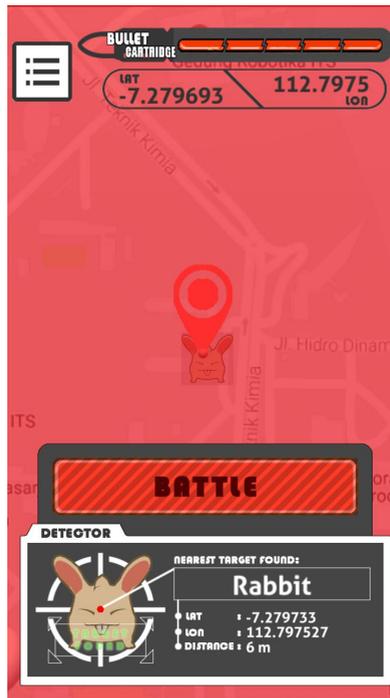
4.2 Implementasi Antarmuka

Pada subbab ini akan dibahas mengenai implementasi antarmuka seperti yang sudah dijelaskan pada Subbab 3.2.6. Implementasi antarmuka aplikasi akan meliputi antarmuka halaman menu peta dan antarmuka halaman pertarungan. Serta ditambah antarmuka halaman menu utama sebagai pelengkap.

4.2.1 Implementasi Antarmuka Halaman Menu Peta

Pada halaman ini akan ditampilkan peta di mana posisi pengguna berada, posisi monster dan peluru yang berada didekat pengguna. Selain itu halaman ini juga akan menampilkan lokasi monster terdekat dengan pengguna serta data informasi mengenai monster tersebut seperti posisi *longitude* dan *latitude*-nya serta jarak antara pengguna dengan monster tersebut dalam ukuran meter. Tampilan antarmuka halaman menu peta bisa dilihat pada Gambar 4.1.

Pada halaman ini ketika pengguna tidak berada didekat monster maka tampilan akan normal, namun ketika pengguna telah mendekati monster maka halaman ini akan menampilkan berbagai macam perubahan seperti layar yang akan berkedip berwarna merah serta tombol 'BATTLE' yang akan membawa pengguna ke halaman pertarungan.

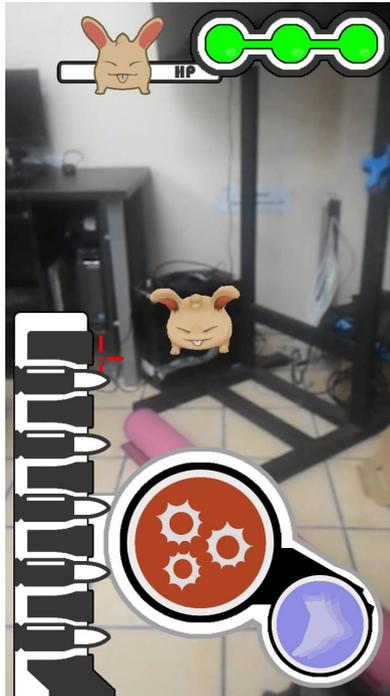


Gambar 4.1 Antarmuka Halaman Menu Peta

4.2.2 Implementasi Antarmuka Halaman Pertarungan

Pada halaman ini akan ditampilkan tampilan pertarungan dengan monster dalam bentuk *Augmented Reality*. Tampilan antarmuka halaman pertarungan bisa dilihat pada Gambar [4.2](#).

Tombol-tombol yang ditampilkan pada halaman ini juga cukup sedikit yaitu tombol menembak dan juga tombol untuk kabur dari pertarungan. Serta akan ditampilkan juga *Health Point* dari monster dan jumlah peluru yang dimiliki pengguna.



Gambar 4.2 Antarmuka Halaman Pertarungan

4.2.3 Implementasi Antarmuka Halaman Menu Utama

Pada halaman ini akan ditampilkan tampilan menu utama seperti mana yang ada pada aplikasi pada umumnya. Tampilan antarmuka halaman menu utama bisa dilihat pada Gambar [4.3](#).

Pada halaman ini akan ditampilkan dua tombol yaitu tombol untuk masuk ke halaman peta serta tombol untuk mengatur volume suara.



Gambar 4.3 Antarmuka Halaman Menu Utama

4.3 Implementasi Alur Penggunaan Aplikasi

Pada subbab ini akan dibahas mengenai implementasi alur aplikasi seperti yang sudah dijelaskan pada Subbab 3.2.1.

4.3.1 Implementasi Pengambilan Data Koordinat dari JSON Lokal

Semua data koordinat dari monster dan juga peluru diambil dari file JSON yang sudah ada di dalam aplikasi. Implementasi kode yang digunakan untuk mengambil atau membaca data dari file JSON tertera seperti pada Kode Sumber [4.1](#).

```

1  using LitJson;
2
3  TextAsset Jsonfile = Resources.Load<TextAsset> ("
    Monster Busters");
4  JSONRead = JsonMapper.ToObject (Jsonfile.text);
5
6  childcount = JSONRead["Data"]["Monster"].Count;
7  for(int i=0; i<childcount; i++) {
8      templon = double.Parse(JSONRead["Data"]["Monster"][i]
          ["lon"].ToString());
9      templat = double.Parse(JSONRead["Data"]["Monster"][i]
          ["lat"].ToString());
10     tempid = JSONRead["Data"]["Monster"][i]["id"].
        ToString().Replace("\"", "");
11     double[] tempxzm = geoDistance.convertXZ(playerlon,
        playerlat, templon, templat);
12     double[] trueLLm = {templon, templat};
13     monsterid.Add(tempid);
14     monsterLL.Add(trueLLm);
15     monsterXZCoordination.Add(tempxzm);
16     if(!PlayerPrefs.HasKey(monsterid[i])) {
17         PlayerPrefs.SetInt(monsterid[i], 1);
18     }
19     PlayerPrefs.SetString("lon"+monsterid[i], templon.
        ToString());
20     PlayerPrefs.SetString("lat"+monsterid[i], templat.
        ToString());
21 }

```

Kode Sumber 4.1 Kode Untuk Membaca JSON

File JSON yang dibaca akan menjadi sebuah *string* seperti pada saat data itu tersimpan pada file. Contoh isi dari file JSON bisa dilihat pada Kode Sumber [4.2](#).

```

1  {
2      "Data" : {
3          "Monster" : [ {
4              "id" : "Monster01",
5              "lat" : -7.279733,
6              "lon" : 112.797527

```

```

7      }, {
8        "id" : "Monster02",
9        "lat" : -7.277179,
10       "lon" : 112.792044
11     }],
12     "Powerup" : [ {
13       "id" : "Power01",
14       "lat" : -7.279364,
15       "lon" : 112.796989
16     }, {
17       "id" : "Power02",
18       "lat" : -7.276735,
19       "lon" : 112.792692
20     } ]
21   }
22 }

```

Kode Sumber 4.2 Contoh Format JSON

4.3.2 Implementasi Handler Data GPS

Implementasi ini dijalankan saat pengguna berada pada menu peta, dimana dengan adanya implementasi ini akan mengatur semua data koordinat GPS yang diterima *device* sehingga aplikasi bisa berjalan sebagai mana semestinya ketika pengguna bergerak dari titik semula ke titik lainnya. Implementasi kode yang digunakan sebagai *handler* data GPS bisa dilihat pada Kode Sumber [4.3](#).

```

1 void Start () {
2   Input.location.Start ();
3   if (Input.location.isEnabledByUser) {
4     lat = Input.location.lastData.latitude;
5     lon = Input.location.lastData.longitude;
6     map.GetComponent<GoogleMap> ().centerLocation.
7       latitude = lat;
8     map.GetComponent<GoogleMap> ().centerLocation.
9       longitude = lon;
10  }
11 }

```

```

9 }
10 void Update () {
11     if (Input.location.isEnabledByUser) {
12         lat = Input.location.lastData.latitude;
13         lon = Input.location.lastData.longitude;
14         if (lastlat != lat || lastlon != lon) {
15             map.GetComponent<GoogleMap> ().centerLocation.
                latitude = lat;
16             map.GetComponent<GoogleMap> ().centerLocation.
                longitude = lon;
17             spawn.GetComponent<monsterSpawn> ().
                updateMonstersPosition (lon, lat);
18             spawn.GetComponent<monsterSpawn> ().
                updatebulletsPosition (lon, lat);
19             map.GetComponent<GoogleMap> ().Refresh ();
20         }
21         lastlat = lat;
22         lastlon = lon;
23     }
24 }

```

Kode Sumber 4.3 Kode Untuk Handler Data GPS

4.3.3 Implementasi Monster Spawner

Implementasi ini dijalankan saat aplikasi memasuki menu peta, dimana implementasi ini akan bertanggung jawab dalam pemanggilan monster-monster serta peluru yang akan dipanggil dari *prefabs* kedalam peta. Implementasi kode yang digunakan sebagai *monster spawner* bisa dilihat pada Kode Sumber [4.4](#). Sedangkan implementasi kode yang digunakan sebagai *spawner* untuk peluru bisa dilihat pada Kode Sumber [4.5](#).

```

1 private bool mspawn = false;
2
3 void Start(){
4     mspawn = true;
5 }
6

```

```

7 void Update(){
8     if (monster.Count==0) {
9         mrespawn = 1;
10        for (int i = 0; i < monsterXZCoordination.Count; i
            ++){
11            if (PlayerPrefs.GetInt(monsterid[i])==1) {
12                mrespawn = 0;
13                break;
14            }
15        }
16        if (mrespawn == 1) {
17            for (int i = 0; i < monsterXZCoordination.Count; i
                ++){
18                PlayerPrefs.SetInt(monsterid[i], 1);
19            }
20            mrespawn = 0;
21        }
22        mspawn = true;
23    }
24    if (mspawn == true) {
25        monsterSpawning ();
26    }
27 }
28
29 void monsterSpawning(){
30     for (int i = 0; i < monsterid.Count; i++) {
31         if(PlayerPrefs.GetInt(monsterid[i]) == 1) {
32             GameObject m;
33
34             if(monsterid[i] == "Monster01" || monsterid[i] ==
                "Monster04" || monsterid[i] == "Monster07" ||
                monsterid[i] == "Monster10") m = monst[0];
35             else if(monsterid[i] == "Monster02" || monsterid[i]
                ] == "Monster05" || monsterid[i] == "Monster08
                " || monsterid[i] == "Monster11") m = monst
                [1];
36             else m = monst[2];
37             GameObject temp = Instantiate (m, new Vector3 ((
                float)monsterXZCoordination [i][0], 0.07f, (
                float)monsterXZCoordination [i][1]), new
                Quaternion (0, -180, 0, 0)) as GameObject;

```

```

38     GameObject player = GameObject.
        FindGameObjectWithTag ("playerController");
39     GameObject controller = GameObject.
        FindGameObjectWithTag ("controller");
40
41     temp.name = monsterid [i];
42     temp.tag = "monster";
43     Mons = temp.GetComponent<monster>();
44     Mons.setLonLat (monsterLL [i] [0], monsterLL [i]
        [1]);
45     Mons.setMonid (monsterid [i]);
46     Mons.setPlayC (player);
47     Mons.setController (controller);
48     monster.Add (temp);
49 }
50 }
51 mspawn = false;
52 }

```

Kode Sumber 4.4 Kode Untuk Monster Spawner

```

1 private bool bspawn = false;
2
3 void Start(){
4     bspawn = true;
5 }
6
7 void Update(){
8     if (bullet.Count==0) {
9         brespawn = 1;
10        for (int i = 0; i < bulletXZCoordination.Count; i
            ++){
11            if (PlayerPrefs.GetInt(bulletid[i])==1) {
12                brespawn = 0;
13                break;
14            }
15        }
16        if (brespawn == 1) {
17            for (int i = 0; i < bulletXZCoordination.Count; i
                ++){
18                PlayerPrefs.SetInt (bulletid[i], 1);

```

```
19     }
20     brespawn = 0;
21 }
22 bspawn = true;
23 }
24 if (bspawn == true) {
25     bulletSpawning ();
26 }
27 }
28
29 void bulletSpawning(){
30     for (int i = 0; i < bulletid.Count; i++) {
31         if(PlayerPrefs.GetInt(bulletid[i]) == 1) {
32             GameObject temp = Instantiate (bull, new Vector3
33                 ((float)bulletXZCoordination [i][0], 0.07f, (
34                 float)bulletXZCoordination [i][1]), new
35                 Quaternion (0, 0, 0, 0)) as GameObject;
36             GameObject player = GameObject.
37                 FindGameObjectWithTag ("playerController");
38             GameObject controllerUI = GameObject.
39                 FindGameObjectWithTag ("controller");
40             GameObject monspwn = this.gameObject;
41
42             temp.name = bulletid [i];
43             Buls = temp.GetComponent<cartridge>();
44             Buls.setBulid (bulletid [i]);
45             Buls.setPlayC (player);
46             Buls.setUIC (controllerUI);
47
48             bullet.Add (temp);
49         }
50     }
51     bspawn = false;
52 }
```

Kode Sumber 4.5 Kode Untuk Bullet Spawner

4.3.4 Implementasi Pengatur Lokasi Monster

Implementasi ini bertanggung jawab dalam melakukan relokalisasi monster saat pengguna berpindah dari posisi awal ke posisi lainnya, dan membuat monster-monster dan peluru berpindah sejauh pengguna pindah. Implementasi kode yang digunakan sebagai pengatur lokasi monster bisa dilihat pada Kode Sumber [4.6](#). Sedangkan implementasi yang digunakan sebagai pengatur lokasi peluru bisa dilihat pada Kode Sumber [4.7](#).

```

1 void Update() {
2     if (lastlon != playerlon || lastlat != playerlat ||
        booting) {
3         nearobject();
4         updateMonstersPosition ();
5         updateBulletsPosition ();
6         if (booting) {
7             booting = false;
8         }
9     }
10 }
11
12 void updateMonstersPosition() {
13     timeOfupdate++;
14     for (int i = 0; i < monster.Count; i++) {
15         int index = monsterid.FindIndex(x => x==monster[i].
            name);
16         double[] tempxz = geoDistance.convertXZ(playerlon,
            playerlat,monsterLL[index][0],monsterLL[index
            ][1]);
17         monster [i].gameObject.transform.position = new
            Vector3 ((float)tempxz[0],0.07f,(float)tempxz
            [1]);
18     }
19 }

```

Kode Sumber 4.6 Kode Untuk Pengatur Lokasi Monster

```

1 void Update(){
2   if (lastlon != playerlon || lastlat != playerlat ||
      booting) {
3     nearobject();
4     updateMonstersPosition ();
5     updateBulletsPosition ();
6     if (booting) {
7       booting = false;
8     }
9   }
10 }
11
12 void updateBulletsPosition(){
13   timeOfupdate++;
14   for (int i = 0; i < bullet.Count; i++) {
15     int index = bulletid.FindIndex(x => x==bullet[i].
      name);
16     double[] tempxz = geoDistance.convertXZ(playerlon,
      playerlat,bulletLL[index][0],bulletLL[index][1])
      ;
17     bullet [i].gameObject.transform.position = new
      Vector3 ((float)tempxz[0],0.07f,(float)tempxz
      [1]);
18   }
19 }

```

Kode Sumber 4.7 Kode Untuk Pengatur Lokasi Peluru

4.3.5 Implementasi Pengatur Peluru Pengguna

Implementasi ini bertanggung jawab dalam mengatur kondisi peluru pengguna, ketika pengguna mengambil peluru, apakah jumlahnya sudah mencapai maksimal atau belum, jika belum bagaimana menambahnya, jika sudah bagaimana pengecekannya. Implementasi kode yang digunakan sebagai pengatur peluru pengguna bisa dilihat pada Kode Sumber [4.8](#). Sedangkan implementasi kode yang digunakan sebagai *trigger* peluru yang berada di peta bisa dilihat pada Kode Sumber [4.9](#).

```

1 public void setBullet (int source) {
2     if ((bullet + source) >= maxbullet) {
3         bullet = maxbullet;
4     }
5     else {
6         bullet = bullet + source;
7     }
8
9     PlayerPrefs.SetInt ("Playerbullet", bullet);
10 }
11
12 void Awake () {
13     if (!PlayerPrefs.HasKey ("Playerbullet")) {
14         PlayerPrefs.SetInt ("Playerbullet", 5);
15     }
16 }
17
18 void Start () {
19     bullet = PlayerPrefs.GetInt ("Playerbullet");
20     uiscript.chargeBar ();
21 }

```

Kode Sumber 4.8 Kode Untuk Pengatur Peluru Pengguna

```

1 void OnTriggerEnter (Collider target) {
2     if (target.tag == "Player") {
3         pscript.setBullet (1);
4         PlayerPrefs.SetInt (cartridgeid, 0);
5         uiscript.chargeBar ();
6         Destroy (this.gameObject);
7     }
8 }

```

Kode Sumber 4.9 Kode Untuk Pengatur Peluru Pada Peta

4.3.6 Implementasi Pencari Monster Terdekat

Implementasi ini dijalankan saat aplikasi memasuki menu peta, dimana implementasi ini akan bertanggung jawab dalam

membantu pengguna dalam mengetahui dimana monster terdekat dengan pengguna, dan berbagai informasi seperti posisi *latitude*, *longitude* dan jarak dari pengguna. Implementasi kode yang digunakan sebagai pencari monster terdekat bisa dilihat pada Kode Sumber [4.10](#).

```

1 void Update(){
2     if (lastlon != playerlon || lastlat != playerlat ||
        booting) {
3         nearobject();
4         updateMonstersPosition ();
5         updateBulletsPosition ();
6         if (booting) {
7             booting = false;
8         }
9     }
10 }
11
12 void nearobject () {
13     GameObject nearest = null;
14     float distance = Mathf.Infinity;
15     Vector3 position = player.transform.position;
16
17     for (int i = 0; i < monster.Count; i++) {
18         Vector3 diff = monster [i].transform.position -
            position;
19         float curDistance = diff.sqrMagnitude;
20         if (curDistance < distance) {
21             nearest = monster[i];
22             distance = curDistance;
23         }
24     }
25     nearmons = nearest.name;
26     near_lat = nearest.GetComponent<monster> ().getLat ();
27     near_lon = nearest.GetComponent<monster> ().getLon ();
28     harvesineFormula ();
29 }
30
31 void harvesineFormula () {
32     float R = 6371;

```

```

33 float lat1 = float.Parse(getnearlat().ToString()) * (
    Mathf.PI / 180);
34 float lat2 = float.Parse(playerlat.ToString()) * (
    Mathf.PI / 180);
35 float disLat = float.Parse((getnearlat() - playerlat).
    ToString()) * (Mathf.PI / 180);
36 float disLon = float.Parse((getnearlon() - playerlon).
    ToString()) * (Mathf.PI / 180);
37 float a = Mathf.Sin (disLat / 2) * Mathf.Sin (disLat /
    2) + Mathf.Cos (lat1) * Mathf.Cos (lat2) * Mathf.
    Sin (disLon / 2) * Mathf.Sin (disLon / 2);
38 float b = 2 * Mathf.Atan2 (Mathf.Sqrt (a), Mathf.Sqrt
    (1 - a));
39 float c = R * b * 1000f;
40 near_distance = double.Parse (Mathf.Round(c).ToString
    ());
41 }

```

Kode Sumber 4.10 Kode Untuk Pencari Monster Terdekat

4.3.7 Implementasi Tombol Pertarungan

Implementasi ini akan bertanggung jawab dalam pengaturan *scene* apa yang akan dibuka ketika pengguna menekan tombol *Battle* untuk memulai pertarungan, bagaimana implementasi ini akan menyimpan lokasi monster yang akan dilawan untuk dicek pada implementasi lain untuk fitur *Location Based Augmented Reality*. Implementasi kode yang digunakan sebagai tombol pertarungan ini bisa dilihat pada Kode Sumber 4.11.

```

1 public void setlastmon (string mons) {
2     lastmonster = mons;
3     if (lastmonster == "Monster01" || lastmonster == "
        Monster04" || lastmonster == "Monster07" ||
        lastmonster == "Monster10") {
4         monstertype = 1;
5     }

```

```

6  else if (lastmonster == "Monster02" || lastmonster ==
      "Monster05" || lastmonster == "Monster08" ||
      lastmonster == "Monster11") {
7      monstertype = 2;
8  }
9  else if (lastmonster == "Monster03" || lastmonster ==
      "Monster06" || lastmonster == "Monster09" ||
      lastmonster == "Monster12") {
10     monstertype = 3;
11 }
12 }
13
14 public void battle() {
15     AudioConScript.PlaySoundEffect (but_press);
16     PlayerPrefs.SetInt (lastmonster, 0);
17     PlayerPrefs.SetString ("lastmonster", lastmonster);
18     if (monstertype == 1) {
19         SceneManager.LoadScene ("Battle Scene 01");
20     }
21     else if (monstertype == 2) {
22         SceneManager.LoadScene ("Battle Scene 02");
23     }
24     else {
25         SceneManager.LoadScene ("Battle Scene 03");
26     }
27 }

```

Kode Sumber 4.11 Kode Untuk Tombol Pertarungan

4.3.8 Implementasi Pendeteksi Lokasi

Implementasi ini akan berfungsi sebagai pengecekan koordinat yang akan menjadi *trigger* untuk *Location Based Augmented Reality*. Dimana akan mengecek secara berkala apakah pengguna sudah berada didekat posisi monster yang ingin dilawan atau belum, jika sudah apakah akan menjauhi, karena jika pengguna menjauhi monster akan membuat *flag* menjadi *false* dan ini akan menyebabkan pembatalan fitur ini. Selain itu

juga akan dilakukan relokalisasi objek secara berkala untuk membuat objek monster selalu berada ditengah tengah layar, guna membuat permainan menjadi lebih mudah. Implementasi kode yang digunakan sebagai Pendeteksi Lokasi ini bisa dilihat pada Kode Sumber [4.12](#).

```

1 void Start() {
2     bcscript = controller.GetComponent<battleController>
3         ();
4     monbatasc = monsters.GetComponent<monsterBattle> ();
5     dLongitude = float.Parse(PlayerPrefs.GetString ("lon"+
6         PlayerPrefs.GetString("lastmonster")));
7     dLatitude = float.Parse(PlayerPrefs.GetString ("lat"+
8         PlayerPrefs.GetString("lastmonster")));
9     targetCoordinates = new Vector2 (dLatitude, dLongitude
10        );
11    StartCoroutine(startGPSService());
12 }
13
14 IEnumerator startGPSService () {
15     Input.location.Start ();
16     sLatitude = Input.location.lastData.latitude;
17     sLongitude = Input.location.lastData.longitude;
18     InvokeRepeating ("getLocation", .01f, 3.5f);
19     yield return new WaitForSeconds (.01f);
20 }
21
22 public void getLocation(){
23     if (flags && monbatasc.getHP() != 0) {
24         sLatitude = Input.location.lastData.latitude;
25         sLongitude = Input.location.lastData.longitude;
26         startCalculate ();
27     }
28 }
29
30 public void startCalculate(){
31     deviceCoordinates = new Vector2 (sLatitude, sLongitude
32        );
33     proximity = Vector2.Distance (targetCoordinates,
34         deviceCoordinates);
35     if (proximity <= distanceFromTarget) {

```

```

30     KC.startLB ();
31 } else {
32     flags = false;
33     bcscript.triggerflee ();
34 }
35 }

```

Kode Sumber 4.12 Kode Untuk Pendeteksi Lokasi

4.3.9 Implementasi Pertarungan

Implementasi ini yang akan mengatur bagaimana pertarungan dengan monster akan berjalan, seperti pengatur *Health Point Bar* dari monster, bagaimana itu akan berkurang, penggunaan peluru untuk menyerang monster, tombol *flee* untuk kabur dari pertarungan dan sebagainya. Implementasi kode yang digunakan sebagai pertarungan bagian monster bisa dilihat pada Kode Sumber [4.13](#). Sedangkan bagian implementasi untuk tombol-tombol yang digunakan pada pertarungan bisa dilihat pada Kode Sumber [4.14](#).

```

1 void Awake () {
2     currentHP = maxHP;
3 }
4
5 void Update () {
6     if (currentHP == 0 && endgame == false) {
7         endgame = true;
8         StartCoroutine (deadEnd ());
9     }
10 }
11
12 public int getHP () {
13     return currentHP;
14 }
15
16 public void reduceHP () {
17     currentHP = currentHP - 1;

```

```

18 }
19
20 void OnTriggerEnter (Collider collision) {
21     if (collision.tag == "bullets") {
22         AudioConScript.PlaySoundEffect (dmged);
23         animations.SetTrigger ("damage");
24         reduceHP ();
25         Destroy (collision.gameObject);
26     }
27 }
28
29 IEnumerator deadEnd(){
30     animations.SetTrigger ("die");
31     AudioConScript.PlaySoundEffect (died);
32     yield return new WaitForSeconds (2f);
33     Destroy (this.gameObject);
34 }

```

Kode Sumber 4.13 Kode Untuk Pertarungan Bagian Monster

```

1 public void bang() {
2     if(PlayerPrefs.GetInt("Playerbullet") > 0 &&
3         buicscript.getCurrentHP() > 0) {
4         AudioConScript.PlaySoundEffect (bangs);
5         PlayerPrefs.SetInt ("Playerbullet", (PlayerPrefs.
6             GetInt ("Playerbullet") - 1));
7         //GameObject bullets = Instantiate(bullet, pos.
8             transform.position, cross.transform.rotation) as
9             GameObject;
10        GameObject bullets = Instantiate(bullet, pos.
11            transform.position, Quaternion.LookRotation (
12                cross.transform.position - pos.transform.
13                position)) as GameObject;
14        bullets.tag = "bullets";
15        bullets.transform.parent = gun.transform;
16    }
17 }
18
19
20 public void flee () {

```

```

13  bcscript.fleebutton ();
14 }

```

Kode Sumber 4.14 Kode Untuk Tombol *Shoot* dan *Flee*

Selain Kode Sumber 4.14 masih ada kode sumber yang berperan dalam mekanisme peluru yang dikeluarkan pada saat pertarungan, yaitu kode sumber yang bertanggung jawab dalam pergerakan peluru dan menghancurkannya ketika peluru tidak mengenai monster. Implementasi untuk pergerakan peluru bisa dilihat pada Kode Sumber 4.15.

```

1 void Start () {
2   rb = GetComponent<Rigidbody>();
3   StartCoroutine (Suicide());
4 }
5
6 void Update () {
7   rb.AddForce(transform.forward * speed);
8 }
9
10 private IEnumerator Suicide(){
11   yield return new WaitForSeconds (4);
12   Destroy (this.gameObject);
13 }

```

Kode Sumber 4.15 Kode Untuk Pergerakan Peluru

Karena *Scene* pertarungan ini dikembangkan dengan metode arsitektur MVC maka ada juga juga kode sumber yang akan bertanggung jawab dalam *Controller* dan *View*. Implementasi untuk *Controller* bisa dilihat pada Kode Sumber 4.16 dan implementasi untuk *View* bisa dilihat pada Kode Sumber 4.17.

```

1 void Awake () {
2   battleuis = GetComponent<battleUIController> ();
3 }
4
5 public void triggerend ( int bul, int hp) {
6   chp = hp;

```

```
7   cbul = bul;
8   if (!endgame) {
9       StartCoroutine (battleended ());
10      endgame = true;
11  }
12 }
13
14 public void triggerflee () {
15     StartCoroutine (monsflee ());
16 }
17
18 public void gotomap(){
19     SceneManager.LoadScene ("Map Scene");
20 }
21
22 public void fleebutton() {
23     battleuis.showendgame (4);
24 }
25
26 public void cancelflee() {
27     battleuis.cancelendgame ();
28 }
29
30 IEnumerator battleended() {
31     yield return new WaitForSeconds (1f);
32     if (chp == 0) {
33         yield return new WaitForSeconds (2f);
34         battleuis.showendgame (1);
35     } else {
36         yield return new WaitForSeconds (1f);
37         battleuis.showendgame (2);
38     }
39 }
40
41 IEnumerator monsflee () {
42     yield return new WaitForSeconds (1f);
43     battleuis.showendgame (3);
44 }
```

Kode Sumber 4.16 Kode Untuk *Controller* Pertarungan

```
1 void Awake () {
2   presbullet = PlayerPrefs.GetInt ("Playerbullet");
3   prevbullet = presbullet;
4   awakebullet ();
5   prevhp = preshp;
6   mSc = monsters.GetComponent<monsterBattle> ();
7   battlescript = GetComponent<battleController> ();
8 }
9
10 void Update () {
11   prevbullet = presbullet;
12   presbullet = PlayerPrefs.GetInt ("Playerbullet");
13   if (presbullet != prevbullet && prevbullet != 0) {
14     bullet[presbullet].gameObject.SetActive(false);
15   }
16   prevhp = preshp;
17   preshp = mSc.getHP();
18   if (preshp != prevhp && prevhp != 0) {
19     setHPUI ();
20   }
21   if (preshp == 0 || presbullet == 0) {
22     battlescript.triggerend (presbullet, preshp);
23   }
24 }
25
26 void awakebullet () {
27   for (int i = 0; i < prevbullet; i++) {
28     bullet [i].gameObject.SetActive (true);
29   }
30 }
31
32 void setHPUI () {
33   HP [preshp].GetComponent<Image> ().overrideSprite =
      HP0;
34   if (preshp != 0) {
35     HPAdditional [preshp - 1].GetComponent<Image> ().
      overrideSprite = HPBlok0;
36   }
37   if (preshp == 2) {
38     HPAdditional [0].GetComponent<Image> ().
      overrideSprite = HPBlok2;
```

```
39     }
40     for (int i = 0; i < preshp; i++) {
41         if (prevhp == 3) {
42             HP [i].GetComponent<Image> ().overrideSprite = HP2
43             ;
44         } else {
45             HP [i].GetComponent<Image> ().overrideSprite = HP1
46             ;
47         }
48     }
49     public int getcurrentHP() {
50         return preshp;
51     }
52
53     public void showendgame (int cases) {
54         bg.gameObject.SetActive (true);
55         notif.gameObject.SetActive (true);
56
57         if (cases < 4) {
58             notif_B.gameObject.SetActive (true);
59
60             if (cases == 1) {
61                 notif_T.text = "You killed the monster.\nClick OK
62                 to back to the map.";
63             } else if (cases == 2) {
64                 notif_T.text = "You run out of bullets\nClick OK
65                 to back to the map.";
66             } else {
67                 notif_T.text = "The Monster has been fleed\nClick
68                 OK to back to the map.";
69             }
70         } else {
71             notif_Y.gameObject.SetActive (true);
72             notif_N.gameObject.SetActive (true);
73
74             notif_T.text = "Are you sure want to flee?";
75         }
76     }
77 }
```

```
75 public void cancelendgame () {  
76     bg.gameObject.SetActive (false);  
77     notif.gameObject.SetActive (false);  
78     notif_Y.gameObject.SetActive (false);  
79     notif_N.gameObject.SetActive (false);  
80 }
```

Kode Sumber 4.17 Kode Untuk *View* Pertarungan

(Halaman ini sengaja dikosongkan)

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini membahas pengujian dan evaluasi pada aplikasi yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem dan kegunaan aplikasi. Pengujian fungsionalitas mengacu pada kasus penggunaan yang telah disebutkan pada Subbab 3.1.2.

5.1 Lingkungan Uji Coba

Lingkungan uji coba yang digunakan adalah sebuah *smartphone* dengan spesifikasi sebagai berikut:

1. Perangkat Keras
 - Asus Zenfone 2 ZE500CL
 - Prosesor Intel Atom Z2560
 - Memori 2048 MB
2. Perangkat Lunak
 - Sistem Operasi Android 5.0

5.2 Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan dengan menyiapkan beberapa skenario pengujian sebagai tolok ukur keberhasilan pengujian dan mengacu pada kasus penggunaan yang sudah dijelaskan pada Subbab 3.1.2.

5.2.1 Pengujian Bertarung dengan Monster dalam Tampilan Augmented Reality

Pengujian bertarung dengan monster dalam tampilan *augmented reality* merupakan pengujian terhadap aplikasi untuk menampilkan pertarungan secara *augmented reality* dan pengujian terhadap fitur-fitur pertarungan yang telah diimplementasikan. Skenario pertama ialah pengguna membuka aplikasi lalu diminta mendekati monster terdekat yang ada dan

melakukan pertarungan dalam kondisi *GPS Service* sedang aktif. Skenario kedua yaitu sama dengan skenario pertama, namun kondisi dari *GPS Service* tidaklah aktif. Hasil pengujian skenario pertama bisa dilihat pada Tabel 5.1, sedangkan skenario kedua bisa dilihat pada Tabel 5.2.

Tabel 5.1 Pengujian Bertarung dengan Monster dalam Tampilan Augmented Reality Skenario 1

Nomor	SP-UC-001-01
Referensi Kasus Penggunaan	UC-001
Nama	Bertarung dengan monster dalam tampilan <i>augmented reality</i>
Tujuan	Mengecek apakah aplikasi sudah bisa berjalan sebagaimana mestinya seperti bisa bertarung dengan monster dengan tampilan <i>augmented reality</i> dan apakah aplikasi sudah menampilkan peringatan bahwa GPS harus aktif
Kondisi Awal	Pengguna membuka aplikasi
Skenario	Pengguna menekan tombol <i>PLAY</i> pada menu utama dengan keadaan GPS aktif
Keluaran yang diharapkan	Aplikasi dapat menampilkan halaman menu peta
Hasil Pengujian	Berhasil

Tabel 5.2 Pengujian Bertarung dengan Monster dalam Tampilan Augmented Reality Skenario 2

Nomor	SP-UC-001-02
Referensi Kasus Penggunaan	UC-001
Nama	Bertarung dengan monster dalam tampilan <i>augmented reality</i>
Tujuan	Mengecek apakah aplikasi sudah bisa berjalan sebagaimana mestinya seperti bisa bertarung dengan monster dengan tampilan <i>augmented reality</i> dan apakah aplikasi sudah menampilkan peringatan bahwa GPS harus aktif
Kondisi Awal	Pengguna membuka aplikasi
Skenario	Pengguna menekan tombol <i>PLAY</i> pada menu utama dengan keadaan GPS tidak aktif
Keluaran yang diharapkan	Aplikasi tetap menampilkan halaman menu utama dengan tambahan peringatan bahwa harus menyalakan <i>Location Service</i> atau GPS
Hasil Pengujian	Berhasil

5.2.2 Pengujian mengisi peluru amunisi

Pengujian mengisi peluru amunisi merupakan pengujian terhadap aplikasi untuk mengisi amunisi peluru yang ada dalam *game*. Skenarionya pertama ialah pengguna diminta untuk pergi menuju tempat-tempat yang memiliki koordinat peluru yang ada

dalam permainan dan memulihkan jumlah peluru ketika pelurunya telah berkurang. Skenario kedua sama seperti skenario pertama namun kondisi peluru penuh. Hasil pengujian skenario pertama bisa dilihat pada Tabel 5.3, sedangkan skenario kedua bisa dilihat pada Tabel 5.4.

Tabel 5.3 Pengujian Mengisi Peluru Amunisi Skenario 1

Nomor	SP-UC-002-01
Referensi Kasus Penggunaan	UC-002
Nama	Mengisi peluru amunisi
Tujuan	Mengecek apakah aplikasi sudah bisa berjalan sebagaimana mestinya seperti bisa mengatur kondisi peluru pengguna
Kondisi Awal	Pengguna membuka aplikasi, dan berada dalam halaman menu peta dengan kondisi peluru tidak setara dengan lima
Skenario	Pengguna berjalan ke tempat dimana keberadaan peluru berada
Keluaran yang diharapkan	Aplikasi dapat mengatur perubahan jumlah peluru yang dimiliki pengguna dengan cara menambah jumlah peluru yang dimiliki sebanyak satu buah
Hasil Pengujian	Berhasil

Tabel 5.4 Pengujian Mengisi Peluru Amunisi Skenario 2

Nomor	SP-UC-002-02
Referensi Kasus Penggunaan	UC-002
Nama	Mengisi peluru amunisi
Tujuan	Mengecek apakah aplikasi sudah bisa berjalan sebagaimana mestinya seperti bisa mengatur kondisi peluru pengguna
Kondisi Awal	Pengguna membuka aplikasi, dan berada dalam halaman menu peta dengan kondisi peluru setara dengan lima
Skenario	Pengguna berjalan ke tempat dimana keberadaan peluru berada
Keluaran yang diharapkan	Aplikasi dapat mengatur perubahan jumlah peluru yang dimiliki pengguna. Namun, karena jumlah peluru yang dimiliki pengguna sebelum mendekati peluru sudah menyentuh nilai maksimal maka tidak akan menambahnya sama sekali
Hasil Pengujian	Berhasil

5.2.3 Hasil Pengujian Fungsional

Subbab ini berisi tentang hasil pengujian fungsionalitas yang sudah dilakukan berdasarkan SP-UC-001-01, SP-UC-001-02, SP-UC-002-01 dan SP-UC-002-02. Semua pengujian yang dilakukan menunjukkan bahwa semua fungsionalitas *game* sudah

berjalan dengan baik dan sesuai dengan apa yang diharapkan serta sesuai dengan semua skenario dan alur yang telah dibuat pada perancangan. Rekapitulasi hasil pengujian fungsionalitas ditunjukkan pada Tabel 5.5.

Tabel 5.5 Rekapitulasi hasil pengujian fungsional

No.	Kode Pengujian	Hasil Pengujian
1	SP-UC-001-01	Berhasil
2	SP-UC-001-02	Berhasil
3	SP-UC-002-01	Berhasil
4	SP-UC-002-02	Berhasil

5.3 Pengujian Pengguna

Pengujian pada *game* yang dibangun tidak hanya dilakukan pada fungsionalitas yang dimiliki, tetapi juga pada pengguna untuk melakukan percobaan secara langsung. Pengujian ini berfungsi sebagai pengujian subjektif yang bertujuan untuk mengetahui tingkat keberhasilan *game* yang dibangun dari sisi pengguna. Hal ini dapat dicapai dengan meminta penilaian dan tanggapan dari pengguna terhadap sejumlah aspek perangkat lunak yang ada.

5.3.1 Skenario Uji Coba Pengguna

Dalam melakukan pengujian aplikasi ini, penguji diminta untuk mencoba menggunakan aplikasi yang bersangkutan untuk mencoba semua fungsionalitas dan fitur yang tersedia. Serta meminta saran untuk pengembangan aplikasi.

Pengujian aplikasi oleh pengguna dilakukan dengan sebelumnya memberikan informasi seputar aplikasi, kegunaan, dan fitur-fitur yang dimiliki. Setelah informasi tersampaikan, pengguna kemudian diarahkan untuk langsung mencoba aplikasi

dengan spesifikasi lingkungan yang sama persis dengan yang telah diuraikan pada uji coba fungsionalitas.

Jumlah pengguna yang terlibat dalam pengujian aplikasi ini sebanyak sepuluh orang. Dalam melakukan pengujian, pengguna melakukan percobaan sebanyak satu kali penggunaan.

Dalam memberikan penilaian dan tanggapan terhadap aplikasi, penguji diberikan formulir pengujian aplikasi. Formulir pengujian aplikasi ini memiliki beberapa aspek penilaian dan pada bagian akhir terdapat permintaan saran untuk perbaikan fitur.

5.3.2 Daftar Penguji Aplikasi

Pada subbab ini ditunjukkan daftar pengguna yang bertindak sebagai penguji coba aplikasi yang telah dibangun. Daftar nama penguji perangkat lunak ini ditunjukkan pada Tabel 5.6.

Tabel 5.6 Daftar Nama Penguji Coba Aplikasi

No.	Nama	Pekerjaan
1	Daniel Bintar	Mahasiswa
2	Julio A Leonard	Mahasiswa
3	Faizuddarain Syam	Mahasiswa
4	Guruh Arya Senna	Mahasiswa
5	Afif Ishamsyah H	Mahasiswa
6	Fajar Ade Putra	Mahasiswa
7	Ishardan	Mahasiswa
8	Shafly Naufal A.	Mahasiswa
9	Muhammad Divi Jaya N	Mahasiswa
10	I. G. N. Adi Wicaksana	Mahasiswa

5.3.3 Hasil Uji Coba Pengguna

Jawaban dari pengisian formulir pengujian aplikasi dapat dilihat pada Tabel 5.7, hasil yang terlihat merupakan hasil rata-rata dari semua kuisioner. Namun, hasil formulir pengujian dari masing-masing penguji bisa dilihat pada Lampiran A.

Tabel 5.7 Rangkuman Hasil Formulir Pengujian Aplikasi

No.	Pertanyaan	Nilai
1	Apakah aplikasi mudah digunakan?	4.6
2	Apakah tampilan aplikasi sudah nyaman?	4
3	Apakah aplikasi sudah nyaman digunakan?	4.7
4	Apakah aplikasi mudah dipahami?	4.4
5	Apakah tombol-tombol kontrol pada permainan tidak membingungkan?	4.8
Rata-rata Nilai		4.5

Berdasarkan Tabel 5.7, dapat diketahui nilai rata-rata adalah 4.5. Berdasarkan nilai maksimal 6 dan nilai ini sudah melebihi dari 4 yang memiliki arti "cukup bagus", sehingga dapat disimpulkan bahwa *game* ini secara keseluruhan sudah cukup nyaman dan cukup mudah digunakan serta juga memiliki tampilan yang sudah cukup. Namun, masih diperlukan adanya pengembangan karena ini masih belum mencapai nilai bagus.

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini dijelaskan mengenai kesimpulan yang dapat diambil dari hasil pengujian yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga diberikan saran untuk pengembangan aplikasi kedepannya.

6.1 Kesimpulan

Dari hasil pengamatan selama proses pengerjaan mulai dari proses perancangan, implementasi dan pengujian yang telah dilakukan terhadap aplikasi, diambil kesimpulan sebagai berikut:

1. Dari Pengujian fungsionalitas serta pengujian terhadap pengguna, didapat bahwa GPS memiliki akurasi yang rendah, sehingga hal ini kadang membuat kesan-kesan seperti pengguna yang terlihat "lompat-lompat" serta monster yang sering mengundurkan diri dari pertarungan.
2. Dari hasil pengujian terhadap pengguna, aplikasi ini sudah cukup nyaman dan mudah digunakan oleh pengguna.

6.2 Saran

Adapun saran yang diberikan untuk mengembangkan aplikasi kedepannya antara lain:

1. Aplikasi diharapkan memiliki fitur *multiplayer* yang akan membuat semua pemain ingin bersaing mengalahkan monster yang sama. Selain itu, diharapkan juga adanya perlawanan dari monster serta *reward* ketika berhasil mengalahkannya.
2. Diharapkan untuk menambah fitur pada permainan, seperti *refresh location* karena adanya *miss-location* yang kadang terjadi pada GPS, atau mengakali akurasi GPS yang kadang

tidak akurat. Serta diharapkan adanya aksi pengguna ketika pengguna ingin menambah jumlah peluru.

DAFTAR PUSTAKA

- [1] A. S. Kasbekar, J. S. Dharmani, K. S. Joshi, R. Kumar, and S. S. Pawar, "Location Based Augmented Reality Geo Tag and Geo Locator: An Android Application," in *IRF Internation Conference. IRAJ*, 2015.
- [2] R. Paucher and M. Turk, "Location-based augmented reality on mobile phones," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pp. 9–16, June 2010.
- [3] S. Aurelia, D. Raj, and O. Saleh, "Mobile Augmented Reality and Location Based Service," in *Advances in Information Science and Applications*, 2, 2014.
- [4] J. R. Menzel, M. Königs, and L. Kobbelt, "A Framework for Vision-Based Mobile AR Application," in *In Proceedings of the Workshop on Mobile Vision and HCI (MobiVis). Held in Conjunction with Mobile HCI*, 2012.
- [5] "Augmented reality." [Online]. Tersedia: <https://en.wikipedia.org/wiki/Augmented-reality> [Diakses 5 Desember 2016].
- [6] "Location-based service." [Online]. Tersedia: <https://en.wikipedia.org/wiki/Location-based-service> [Diakses 5 Desember 2016].
- [7] "Unity (game engine)." [Online]. Tersedia: [https://en.wikipedia.org/wiki/Unity-\(game-engine\)](https://en.wikipedia.org/wiki/Unity-(game-engine)) [Diakses 5 Desember 2016].
- [8] "Global Positioning System." [Online]. Tersedia: <https://en.wikipedia.org/wiki/Global-Positioning-System> [Diakses 5 Desember 2016].

- [9] “Products | Kudan.” [Online]. Tersedia: <https://www.kudan.eu/products> [Diakses 5 Desember 2016].
- [10] “Pokémon go - bulbapedia, the community-driven pokémon encyclopedia.” [Online]. Tersedia: <https://bulbapedia.bulbagarden.net/wiki/Pokemon-GO> [Diakses 16 Juni 2017].

LAMPIRAN A

HASIL KUESIONER




Kuisisioner Tugas Akhir "Aplikasi Game dengan Location Based Augmented Reality Berbasis Perangkat Bergerak dengan menggunakan Kudan SDK"

Nama : DANIEL BINTAR
 NRP : 5113100145
 Pekerjaan : Mahasiswa

Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?				✓		
2	Apakah tampilan aplikasi sudah nyaman?				✓		
3	Apakah aplikasi sudah nyaman digunakan?				✓		
4	Apakah aplikasi mudah dipahami?				✓		
5	Apakah tombol-tombol kontrol pada permainan tidak membingungkan?				✓		

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:

- Multiplayer
- Pertolongan tidak hanya 1 orak (monster bisa menyerah)
- Diberikan reward setelah menegalahkan monster

Surabaya, 3 Juni 2017


 (Daniel Bintar)

Hasil Formulir Pengujian Aplikasi: Daniel Bintar



ITS
Institut Teknologi
Sepuluh Nopember



Kuisisioner Tugas Akhir "Aplikasi Game dengan Location Based Augmented Reality Berbasis Perangkat Bergerak dengan menggunakan Kudan SDK"

Nama : Julio A Leonard
 NRP : 5113100240
 Pekerjaan : Mahasiswa

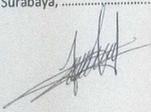
Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?						✓
2	Apakah tampilan aplikasi sudah nyaman?				✓		
3	Apakah aplikasi sudah nyaman digunakan?						✓
4	Apakah aplikasi mudah dipahami?						✓
5	Apakah tombol-tombol kontrol pada permainan tidak membingungkan?						✓

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:
Tampilan lebih dinamis
Dibuatkan cerita atau semacamnya

Surabaya, 5 Juli 2017


 (Julio Anthony Leonard)

Hasil Formulir Pengujian Aplikasi: Julio A Leonard




Kuisiner Tugas Akhir "Aplikasi Game dengan Location Based Augmented Reality Berbasis Perangkat Bergerak dengan menggunakan Kudan SDK"

Nama : *Faizuddarain Syam*
 NRP : *511300111*
 Pekerjaan : *Mahasiswa*

Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?		✗	✓			
2	Apakah tampilan aplikasi sudah nyaman?			✗	✓		
3	Apakah aplikasi sudah nyaman digunakan?		✗	✓			
4	Apakah aplikasi mudah dipahami?				✓		
5	Apakah tombol-tombol kontrol pada permainan tidak membingungkan?					✓	

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:
Gambar icon nya diperkecil, jadi kelihatan kaku berpindah tempat
atau Agak
Ada pilihan refresh lokasi di permainan.

Surabaya, *08* - *2017* 2017

Faizuddarain Syam
 (*Faizuddarain syam*)

Hasil Formulir Pengujian Aplikasi: Faizuddarain Syam




Kuisisioner Tugas Akhir "Aplikasi Game dengan Location Based Augmented Reality Berbasis Perangkat Bergerak dengan menggunakan Kudan SDK"

Nama : GURUH ARYA SENNA
 NRP : 5113100010
 Pekerjaan : MAHASISWA

Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?				✓		
2	Apakah tampilan aplikasi sudah nyaman?				✓		
3	Apakah aplikasi sudah nyaman digunakan?					✓	
4	Apakah aplikasi mudah dipahami?			✓			
5	Apakah tombol-tombol kontrol pada permainan tidak membingungkan?					✓	

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:
 Tampilan dalam aplikasi masih terlihat kurang diperhalus / masih agak kasar.
 Akan terlihat lebih bagus jika diberi tutorial permainan jadi pemain yang
 belum pernah memakai / menggunakan aplikasi tidak kebingungan atau memera fungsi -
 fungsi tombol di dalam aplikasi

Surabaya, 4 Juni 2017


 (Guruh Arya Senna)

Hasil Formulir Pengujian Aplikasi: Guruh Arya Senna




Kuisisioner Tugas Akhir "Aplikasi Game dengan Location Based Augmented Reality Berbasis Perangkat Bergerak dengan menggunakan Kudan SDK"

Nama : Afif Ishamsyah H
 NRP : 5113100172
 Pekerjaan : Mahasiswa

Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?				✓		
2	Apakah tampilan aplikasi sudah nyaman?			✓			
3	Apakah aplikasi sudah nyaman digunakan?				✓		
4	Apakah aplikasi mudah dipahami?				✓		
5	Apakah tombol-tombol kontrol pada permainan tidak membingungkan?			✓			

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:
 • Perbaiki Tampilan UI


 Afif
 Surabaya, 4 April 2017

Hasil Formulir Pengujian Aplikasi: Afif Ishamsyah




Kuisisioner Tugas Akhir "Aplikasi Game dengan Location Based Augmented Reality Berbasis Perangkat Bergerak dengan menggunakan Kudan SDK"

Nama : Fajar Ade Putra
 NRP : 5113100092
 Pekerjaan : Mahasiswa

Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?					✓	
2	Apakah tampilan aplikasi sudah nyaman?				✓		
3	Apakah aplikasi sudah nyaman digunakan?						✓
4	Apakah aplikasi mudah dipahami?					✓	
5	Apakah tombol-tombol kontrol pada permainan tidak membingungkan?					✓	

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:
 simbol untuk posisi selang terlalu besar, saranya
 agak dikurangi supaya bisa dilihat monder lainnya

Surabaya, 4 Juni 2017


 (Fajar Ade Putra)

Hasil Formulir Pengujian Aplikasi: Fajar Ade Putra




Kuisisioner Tugas Akhir "Aplikasi Game dengan Location Based Augmented Reality Berbasis Perangkat Bergerak dengan menggunakan Kudan SDK"

Nama : Ishardan
 NRP : 5113100182
 Pekerjaan : Mahasiswa

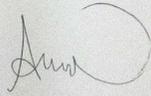
Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?					✓	
2	Apakah tampilan aplikasi sudah nyaman?				✓		
3	Apakah aplikasi sudah nyaman digunakan?				✓		
4	Apakah aplikasi mudah dipahami?				✓		
5	Apakah tombol-tombol kontrol pada permainan tidak membingungkan?					✓	

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:
 Untuk pengisian peluru bisa dipilih dulu baru dapat peluru (tidak otomatis)
 , serta map bisa berwarna.

Surabaya, 4 Juni 2017


 (ISHARDAN)

Hasil Formulir Pengujian Aplikasi: Ishardan




Kuisisioner Tugas Akhir "Aplikasi Game dengan Location Based Augmented Reality Berbasis Perangkat Bergerak dengan menggunakan Kudan SDK"

Nama : Shafly Naufal A.
 NRP : 5114100114
 Pekerjaan : Mahasiswa

Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?					✓	
2	Apakah tampilan aplikasi sudah nyaman?				✓		
3	Apakah aplikasi sudah nyaman digunakan?					✓	
4	Apakah aplikasi mudah dipahami?					✓	
5	Apakah tombol-tombol kontrol pada permainan tidak membingungkan?				✓		

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:
 Variasikan bentuk monster

Surabaya, 04 Juni 2017


 (Shafly N.A.)

Hasil Formulir Pengujian Aplikasi: Shafly Naufal A




Kuisisioner Tugas Akhir "Aplikasi Game dengan Location Based Augmented Reality Berbasis Perangkat Bergerak dengan menggunakan Kudan SDK"

Nama : Muhammad Divi Jaya N
 NRP : 5113100066
 Pekerjaan : Mahasiswa

Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?					✓	
2	Apakah tampilan aplikasi sudah nyaman?				✓		
3	Apakah aplikasi sudah nyaman digunakan?					✓	
4	Apakah aplikasi mudah dipahami?				✓		
5	Apakah tombol-tombol kontrol pada permainan tidak membingungkan?						✓

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:
 Tidak ada masalah serius di fungsionalitas, hanya kendala di GPS yang akurasiya kurang ~~baik~~

Surabaya, 4 Juni 2017

Divi
 (Muhammad Divi Jaya N)

Hasil Formulir Pengujian Aplikasi: Muhammad Divi Jaya N




Kuisisioner Tugas Akhir "Aplikasi Game dengan Location Based Augmented Reality Berbasis Perangkat Bergerak dengan menggunakan Kudan SDK"

Nama : I. G. N. Adi Wicaksana
 NRP : 5113100110
 Pekerjaan : Mahasiswa

Silahkan centang pada kolom yang sesuai berikut ini :

No.	Pertanyaan	Nilai					
		1	2	3	4	5	6
1	Apakah aplikasi mudah digunakan?				✓		
2	Apakah tampilan aplikasi sudah nyaman?			✓			
3	Apakah aplikasi sudah nyaman digunakan?			MM	✓		
4	Apakah aplikasi mudah dipahami?			✓			
5	Apakah tombol-tombol kontrol pada permainan tidak membingungkan?				✓		

Keterangan :
 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Kurang Setuju 4. Cukup Setuju 5. Setuju 6. Sangat Setuju

Kritik dan Saran untuk pengembangan selanjutnya:
 Tampilan gambar/assets masih terlihat pecah. Versi mobile di perbanyak. Gagal. Tadi yg harus mobile dipertbaiki.
 Tambah user guide

Surabaya, 4 Juni 2017


 I. G. N. Adi Wicaksana

Hasil Formulir Pengujian Aplikasi: I. G. N. Adi Wicaksana

BIODATA PENULIS



Naufal Bayu Fauzan, akrab dipanggil Naufal, lahir di Jakarta pada tanggal 8 Januari 1995. Penulis merupakan anak ketiga dari empat bersaudara. Penulis telah menempuh pendidikan formal TK Tunas Asri, Bekasi (2000-2001), SDN Cipinang Melayu 03 Pagi Jakarta Timur (2001-2007), SMPN 109 Jakarta Timur (2007-2010), dan SMAN 12 Jakarta Timur (2010-2013). Setelahnya penulis meneruskan studi di Teknik Informatika ITS. Selama kuliah penulis aktif sebagai administrator Laboratorium Interaksi, Grafika dan Seni (IGS) Teknik Informatika ITS. Selain itu penulis juga mengambil bidang minat Interaksi, Grafika dan memiliki ketertarikan dalam *augmented reality*, *virtual reality* dan *game*. Selain itu dibidang organisasi selama masa kuliah, penulis juga aktif sebagai staf Pengembangan Profesi Himpunan Mahasiswa Teknik Computer-Informatika, Staf Divisi Web3D Schematics 2013, Kepala Sub Divisi Keprofesian Departemen Pengembangan Profesi Himpunan Mahasiswa Teknik Computer-Informatika, dan staf Divisi 3D Schematics 2014. Penulis dapat dihubungi melalui surat elektronik di naufalbfauzan@gmail.com