

**Analisis Penerapan *Markerless Augmented Reality* pada
Video Game Memancing dengan Pendekatan *Simultaneous
Localization and Mapping (SLAM)***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Dany Muhammad
NIM: 115090600111015



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

ANALISIS PENERAPAN *MARKERLESS AUGMENTED REALITY* PADA *VIDEO GAME*
MEMANCING DENGAN PENDEKATAN *SIMULTANEOUS LOCALIZATION AND*
MAPPING (SLAM)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Dany Muhammad
NIM: 115090600111015

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Agustus 2018

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II

Wibisono Sukmo Wardhono, S.T, M.T

NIK: 201008 820404 1 001

Tri Afirianto, S.T, M.T

NIK: 201309 851213 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan , S.T, M.T, Ph.D

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

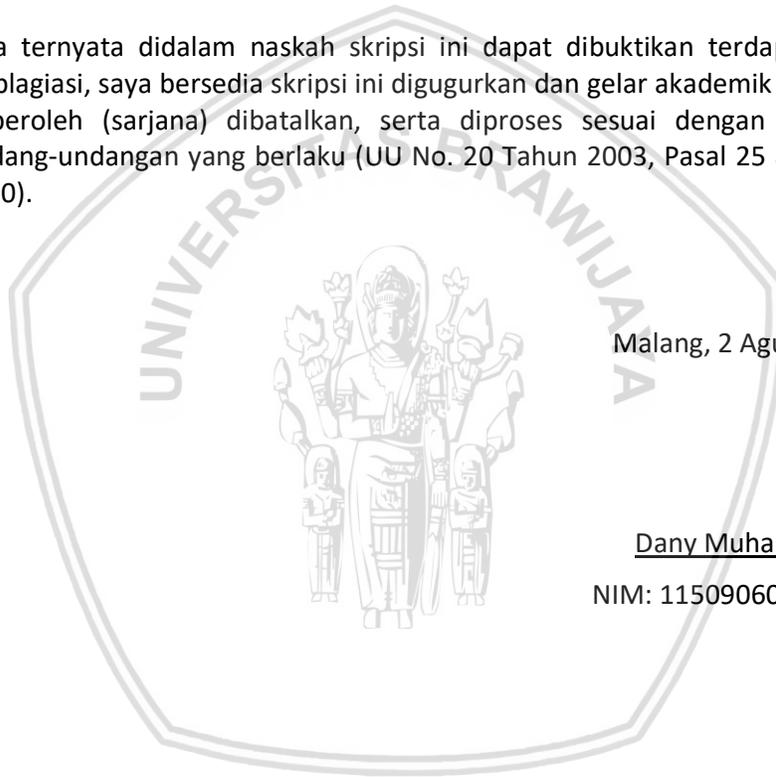
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 2 Agustus 2018

Dany Muhammad

NIM: 115090600111015



KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Analisis Penerapan *Markerless Augmented Reality* pada *Video Game* Memancing dengan Pendekatan *Simultaneous Localization and Mapping* (SLAM)” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Wibisono Sukmo Wardhono, S.T, M.T. dan Tri Afirianto, S.T, M.T selaku Pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini,
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika,
3. Ibu Candra Dewi, S.Kom, M.Sc. dan Ratih Kartika Dewi, S.T., M.Kom. selaku dosen Penasihat Akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi,
4. Ayahanda dan Ibunda dan seluruh keluarga besar atas segala nasihat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini,
5. Teman-teman Program Studi Ilmu Komputer yang telah menemani penulis mencari ilmu sejak masih di Fakultas MIPA, terutama Fadel, Eureka, Yanitra, serta teman-teman keminatan game lain.
6. Teman-teman 99 Komputer yang telah mengajarkan penulis untuk mencari penghasilan sendiri dan banyak membantu proses percetakan skripsi ini.
7. Teman-teman Laboratorium Game yang sudah membantu selama proses penulisan skripsi ini.
8. Para penghuni subforum Anime Manga Haven (AMH) Kaskus yang sudah menemani penulis berdiskusi tentang hobi.
9. Para sahabat kontributor KAORI Nusantara yang sudah menemani penulis menyalurkan hobi menulisnya.
10. Terakhir, teman bermain dan belajar: Hilky, Dwiki, Helmi, Nizar, Danet, Sandi, dan Nanda

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 2 Agustus 2018

Dany Muhammad

Email: dany.muhammad@rocketmail.com



ABSTRAK

Dany Muhammad, Analisis Penerapan *Markerless Augmented Reality* pada *Video Game* Memancing dengan Pendekatan *Simultaneous Localization and Mapping* (SLAM).

Pembimbing: Wibisono Sukmo Wardhono, S.T, M.T dan Tri Afirianto, S.T, M.T

Dalam mengembangkan sebuah aplikasi yang memanfaatkan teknologi *Augmented Reality* (AR), salah satu metode yang paling umum digunakan adalah dengan cara memanfaatkan *marker* khusus. Meskipun begitu, kebutuhan akan adanya *marker* justru dapat membatasi kebebasan pemain saat memainkan permainan. Penelitian ini akan mengembangkan sebuah *video game markerless augmented reality* dengan metode *Simultaneous Localization and Mapping* (SLAM). Metode SLAM merupakan pendekatan di mana *mapping* dilakukan pada sebuah lingkungan yang tidak dikenali dan di saat bersamaan posisi relatif agen yang melakukan aktifitas tersebut ditentukan. Metode SLAM diimplementasikan pada sebuah *video game* memancing yang masih menggunakan *marker* agar dapat dimainkan tanpa menggunakan *marker*. Proses implementasi dilakukan dengan menggunakan *game engine* Unity menggunakan *package* VOID AR yang tersedia secara gratis. Proses pengujian dilakukan dengan *black box testing* yang berfokus pada keberhasilan *video game markerless augmented reality* untuk memunculkan objek virtual yang stabil di lingkungan. Dari hasil pengujian dapat dilihat fungsi-fungsi *video game* memancing dapat berjalan dengan baik sehingga dapat diketahui bahwa *video game markerless augmented reality* memancing dapat dikembangkan dengan metode SLAM menggunakan *package* VOID AR pada *game engine* unity. Objek virtual terlihat paling stabil ketika lingkungan yang dikenali tidak banyak berubah ketika dibandingkan dengan lingkungan awal pada proses inialisasi. Ketika kamera perangkat diarahkan ke lingkungan baru, objek virtual mulai terlihat tidak stabil karena *video game* kesulitan mengenali lingkungan baru.

Kata Kunci: *Simultaneous Localization and Mapping* (SLAM), *Markerless Augmented Reality*, *video game*.

ABSTRACT

Dany Muhammad, *Analyzing the Utilization of Markerless Augmented Reality Fishing Game with Simultaneous Localization and Mapping (SLAM) Approach.*

Adviser: Wibisono Sukmo Wardhono, S.T, M.T dan Tri Afirianto, S.T, M.T

In the development of application that utilize augmented reality, one of the most common approach is using a special marker. However, a marker can limit the player's freedom when playing a video game that utilize augmented reality. This research will develop an augmented reality video game without any marker (markerless) using Simultaneous Localization and Mapping (SLAM) approach. SLAM is an approach of constructing or updating a map of a certain environment while keeping track of an agent's location at the same time. SLAM is implemented in an augmented reality fishing game which use marker so that the game can be played without any marker. The markerless function will be implemented using VOID AR package in Unity game engine. After markerless function implemented, black box testing is used to observe if virtual object can appear correctly in the environment. The testing result shows that markerless augmented reality fishing game can be developed with SLAM method using VOID AR package in Unity. The virtual object looks most stable when the environment that recorded in frame is not changed much. This is happened because it is easier for video game application to use SLAM approach when the features are not changed much. When the camera is directed to the new environment, new features are recorded, make it harder for video game application to adapt.

Keywords: *Simultaneous Localization and Mapping (SLAM), markerless augmented reality, video game.*

DAFTAR ISI

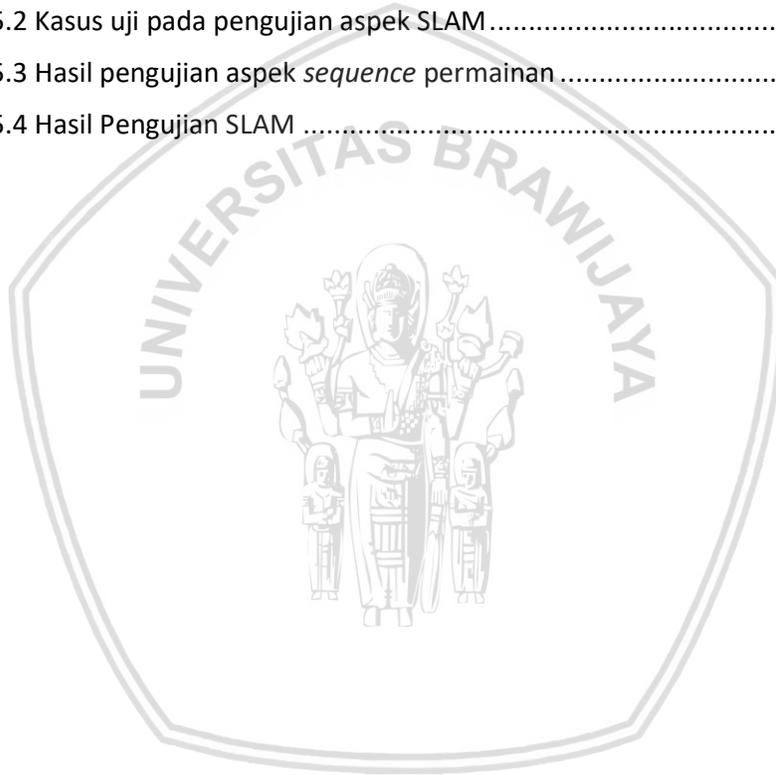
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Manfaat	2
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 <i>Game</i>	5
2.1.1 <i>Formal Element Game</i>	5
2.1.2 <i>Genre Game</i>	7
2.1.3 <i>Sport Game</i>	10
2.2 <i>Augmented Reality</i>	11
2.3 <i>Markerless Augmented Reality</i>	12
2.4 Deteksi dan Pelacakan Pola	13
2.5 <i>Simultaneous Localization and Mapping (SLAM)</i>	14
2.6 Unity	15
2.7 VOID AR	15
2.8 Pengujian <i>Black Box</i>	16
BAB 3 METODOLOGI	17
3.1 Studi Literatur	17
3.2 Desain	18

3.3 Pengembangan	19
3.4 Analisis	19
BAB 4 PERANCANGAN DAN IMPLEMENTASI	20
4.1 Perancangan	20
4.1.1 Analisis Kebutuhan	20
4.1.2 Alur Permainan	21
4.2 Implementasi	24
4.2.1 Penentuan Spesifikasi Hardware dan Software	24
4.2.2 Proses Implementasi SLAM	25
4.2.3 Hasil Implementasi SLAM	26
BAB 5 PENGUJIAN DAN ANALISIS	29
5.1 Rancangan Pengujian	29
5.1.1 Pengujian Aspek <i>Sequence</i> Permainan	30
5.1.2 Pengujian Aspek SLAM	30
5.2 Hasil Pengujian	33
5.2.1 Pengujian Aspek <i>Sequence Game</i>	33
5.2.2 Pengujian Sspek SLAM	34
5.3 Analisis	35
5.3.1 Pengujian Aspek <i>Sequence Game</i>	35
5.3.2 Pengujian Aspek SLAM	35
BAB 6 KESIMPULAN DAN SARAN	37
6.1 Kesimpulan	37
6.2 Saran	37
DAFTAR PUSTAKA	38



DAFTAR TABEL

Tabel 2.1 <i>Formal Elements</i> pada permainan memancing	6
Tabel 4.1 Kebutuhan minimal SDK VOID AR	21
Tabel 4.2 Fungsi dari setiap antarmuka di layar.....	23
Tabel 4.3 Spesifikasi Sistem Perangkat Mobile	25
Tabel 4.4 Spesifikasi Sistem Perangkat Pengembangan.....	25
Tabel 5.1 Kasus uji pada pengujian aspek <i>sequence</i> permainan	30
Tabel 5.2 Kasus uji pada pengujian aspek SLAM.....	32
Tabel 5.3 Hasil pengujian aspek <i>sequence</i> permainan	33
Tabel 5.4 Hasil Pengujian SLAM	34



DAFTAR GAMBAR

Gambar 2.1 <i>RV Continuum</i>	11
Gambar 2.2 Pelacakan pola pada VOID AR.....	13
Gambar 2.3 Proses kerja SLAM	14
Gambar 3.1 Tahapan penelitian	17
Gambar 4.1 Proses Perancangan dan Implementasi.....	20
Gambar 4.2 Alur permainan.....	21
Gambar 4.3 Rancangan antarmuka permainan baru	22
Gambar 4.4 Alur fase permainan	24
Gambar 4.5 Screen permainan hasil implementasi	26
Gambar 4.6 Objek virtual melayang ketika kamera diarahkan ke tempat acak...	27
Gambar 4.7 Tombol berhenti ditekan	28
Gambar 4.8 Objek virtual kembali ditampilkan	28
Gambar 5.1 Kondisi lingkungan pengujian SLAM.....	29
Gambar 5.2 Ilustrasi skenario Kamera bergeser	31
Gambar 5.3 Ilustrasi skenario kamera diputar	31
Gambar 5.4 Ilustrasi skenario kamera memutar objek	32

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Dalam mengembangkan sebuah aplikasi yang memanfaatkan teknologi *Augmented Reality* (AR), salah satu metode yang paling umum digunakan adalah dengan cara memanfaatkan *marker* khusus. *Marker* tersebut berfungsi sebagai titik acuan untuk menentukan posisi dari objek digital di dunia nyata. Dalam prakteknya, kebutuhan akan adanya *marker* menimbulkan masalah baru. Objek digital yang dihasilkan hanya akan stabil bila *marker* terlihat seluruhnya (Raharja, Jalinus, Avileti, 2016). Hal tersebut membuat aplikasi *augmented reality* berbasis *marker* harus memenuhi kondisi-kondisi tertentu untuk dapat menghasilkan objek digital. Karena itu, Diperlukan solusi lain agar objek digital mampu dihasilkan dan stabil tanpa adanya *marker* atau *markerless*.

Manfaat teknologi *markerless augmented reality* akan sangat terasa terutama di bidang hiburan seperti *video game*. *Video game* merupakan media yang dikonsumsi untuk memberikan perasaan senang pada pelakunya (Prensky, 2002). Karena itu, *video game* harus mampu dilakukan dalam kondisi apapun dan di manapun. Memaksa *player* untuk terikat pada *marker* tertentu justru akan membuat bermain *game* tidak lagi menyenangkan. Menurut penelitian sebelumnya yang dilakukan Luthfansyah (2018), 30 persen dari penguji merasa ragu-ragu dengan aspek *augmented reality* dari video game memancing dengan *marker*. Beberapa penguji mengemukakan alasan menjawab ragu-ragu adalah karena tidak terbiasa dengan adanya *marker*.

Salah satu solusi untuk menghilangkan ketergantungan aplikasi *augmented reality* pada kebutuhan akan *marker* khusus adalah dengan mengembangkan aplikasi *augmented reality* yang *markerless*. Pada penelitian yang dilakukan Brito dan Stoyanova (2017), *markerless augmented reality* memiliki beberapa keunggulan dibanding *augmented reality* berbasis *marker*. Penelitian tersebut memperlihatkan aplikasi *markerless augmented reality* lebih memicu gestur tubuh yang lebih positif serta terasa lebih interaktif dibandingkan *augmented reality* berbasis *marker*.

Terdapat beberapa solusi untuk mengembangkan aplikasi *markerless augmented reality*. Salah satu metode yang dapat digunakan adalah metode *Simultaneous Localization and Mapping* (SLAM). SLAM merupakan pendekatan mengenali struktur dunia nyata dengan melakukan *mapping* di tiap citra dengan membandingkan posisi pola garis, pola titik, pola warna, dan dan pola-pola lain yang terbaca. SLAM membandingkan citra yang dikenali oleh perangkat dengan citra sebelumnya, kemudian melakukan perhitungan untuk menentukan tempat relatif objek. Untuk setiap citra yang masuk, aplikasi akan melakukan perhitungan untuk menentukan koordinat baru dari objek digital. Perlu diperhatikan bahwa SLAM bukanlah sebuah algoritme khusus melainkan suatu pendekatan. Setiap peneliti mungkin memiliki algoritme berbeda, tapi algoritme tersebut memiliki tujuan yang sama yaitu untuk melakukan *mapping* secara *real-time* (Riisgaard dan Blas, 2005: 6).

Metode SLAM sendiri lebih banyak digunakan di bidang robotika. Di bidang robotika, SLAM dimanfaatkan untuk menciptakan sebuah robot yang mampu mengenali lingkungannya dan bergerak tanpa adanya masukan khusus dari luar (Claraco, 2009). Di bidang *video game*, *augmented reality* yang menggunakan pendekatan SLAM dikembangkan oleh beberapa perusahaan teknologi seperti Wikitude, Kudan, dan VOID. Perusahaan-perusahaan tersebut menyediakan *package* untuk membantu para peneliti luar dalam mengembang *video game* dengan *markerless augmented reality* untuk berbagai *platform*.

Dari latar belakang di atas, maka disusunlah penelitian Analisis penerapan *video game markerless augmented reality* memancing dengan pendekatan *Simultaneous Localization and Mapping* (SLAM). Penelitian ini akan mengembangkan sebuah *video game* memancing *augmented reality* yang sudah disusun oleh Luthfansyah pada 2018. Hasil penelitian tersebut akan dikembangkan sehingga *video game* memancing *augmented reality* tersebut dapat dimainkan secara *markerless*.

1.2 Rumusan Masalah

Berdasarkan latar belakang berikut, maka rumusan masalah dapat dituliskan sebagai berikut:

1. Bagaimana merancang sebuah *video game augmented reality* memancing yang memungkinkan pengguna untuk tidak terikat dengan *marker*?
2. Bagaimana tingkat keberhasilan aplikasi *video game augmented reality* memancing yang tidak menggunakan *marker* (*markerless*)?

1.3 Tujuan

Tujuan dari penyusunan skripsi ini adalah:

1. Merancang sebuah *video game augmented reality* yang memungkinkan pengguna untuk tidak terikat dengan kebutuhan *marker*.
2. Mengukur tingkat keberhasilan metode SLAM saat menjalankan *video game augmented reality* memancing.

1.4 Manfaat

Manfaat dari penelitian ini adalah:

1. Bagi Penulis: Untuk menganalisis sebuah *video game augmented reality* yang tidak bergantung pada *marker* tertentu.
2. Bagi pengguna: Untuk menghilangkan kebutuhan akan adanya *marker* khusus saat bermain *video game* yang menggunakan teknologi *augmented reality*.
3. Bagi institusi: Untuk meneliti pendekatan SLAM dalam mengembangkan sebuah *video game* dengan teknologi *markerless augmented reality*.

1.5 Batasan Masalah

Batasan-batasan masalah dari penelitian skripsi ini adalah sebagai berikut:

1. *Video game* dibuat untuk *platform* Android.
2. *Augmented Reality* dibuat dengan *package* VOID AR yang tersedia dalam *game engine* Unity.

1.6 Sistematika Pembahasan

Sistematika dari penelitian analisis pengembangan *video game markerless augmented reality* memancing adalah sebagai berikut:

Bab 1: Pendahuluan

Bagian ini merupakan pembahasan tentang tema, masalah, atau pertanyaan yang hendak dijawab oleh penelitian ini. Dari bagian ini dirumuskan apa saja masalah-masalah yang dapat dipecahkan oleh penelitian ini.

Bab 2: Landasan Kepustakaan

Bagian ini berisi pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian. Pembahasan didasarkan oleh studi literatur dari berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian.

Bab 3: Metodologi

Metodologi penelitian adalah proses atau cara ilmiah untuk mendapatkan data yang akan digunakan untuk keperluan penelitian (Sugiyono, 2012). Bagian ini berisi pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian. Pembahasan didasarkan oleh studi literatur dari berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian. Bagian ini berisi tata cara dan alur penelitian.

Bab 4: Perancangan dan Implementasi

Bagian ini menjelaskan dengan detail bagaimana *video game* dirancang sebelum dikembangkan. Perancangan *video game* didasarkan pada latar belakang, rumusan masalah, dan metodologi yang telah dideskripsikan pada bagian sebelumnya. Setelah perancangan ditentukan, pengembangan *video game* dilakukan sesuai dengan rancangan yang telah dibuat. Rancangan tersebut akan diimplementasikan agar dapat menjadi sebuah *video game* yang utuh.

Bab 5: Pengujian dan Analisis

Bagian ini menjelaskan bagaimana pengujian dari *video game* dilakukan. Bagian ini mencakup skenario pengujian yang akan dilakukan, proses pengujian, dan hasil dari pengujian. Setelah dilakukan pengujian, analisis dari pengujian

akan dilakukan berdasarkan informasi-informasi yang didapat dari hasil pengujian.

Bab 6: Kesimpulan dan Saran

Bagian ini menyimpulkan hasil penelitian yang telah dilakukan. Kesimpulan yang ditarik pada penelitian ini diharapkan mampu menjawab rumusan masalah yang telah dirumuskan di bab sebelumnya. Pada bagian ini, peneliti juga dapat memberikan saran-saran untuk membantu proses penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Game

Game adalah kompetisi antara para pemain yang berinteraksi satu sama lain dengan menggunakan aturan tertentu untuk mencapai tujuan tertentu (Sadiman, 2010). Pemain dituntut untuk mendapatkan hasil yang telah ditentukan dengan batasan-batasan yang telah ditentukan oleh *game* tersebut. Pemain dapat dinyatakan menang jika berhasil memperoleh hasil yang sesuai dengan apa yang dikehendaki oleh aturan. *Game* berkembang di banyak media, salah satunya dalam media video. *Game* yang dimainkan dalam bentuk video dalam layar khusus disebut *video game* (Rogers, 2010).

2.1.1 Formal Element Game

Menurut Nacke (2014) *game* terdiri dari sekumpulan elemen dasar yang disebut *formal elements*. Elemen-elemen tersebut saling berpengaruh satu sama lain. Perubahan salah satu elemen akan membuat bentuk keseluruhan *game* tersebut berubah. Beberapa elemen yang dimaksud adalah:

1. **Pemain (*Player*)**. Setiap *game* selalu memiliki pemain yang berinteraksi dengan sistem game. Aktifitas bermain game biasanya dilakukan pemain untuk bersenang-senang. Bagi pemain, aktifitas tersebut dilakukan secara sukarela. Permainan dapat dilakukan secara berkelompok ataupun sendiri-sendiri.
2. **Tujuan (*Objective*)**. Sebuah *game* memiliki tujuan tertentu yang harus dilakukan oleh pemain. Tujuan setiap game berbeda-beda tergantung sistemnya masing-masing.
3. **Aturan (*Rules*)**. Aturan dibuat untuk membatasi aktifitas pemain. Pemain diharapkan untuk bisa mencapai tujuannya dengan segala batasan aturan yang telah ditetapkan sistem game
4. **Prosedur (*Procedures*)**. Prosedur merujuk pada cara pemain untuk mencapai tujuannya. Prosedur dapat berupa instruksi spesifik yang harus dilakukan atau kontrol yang dimiliki pemain.
5. **Sumber Daya (*Resources*)**. Sumber daya adalah objek yang membantu pemain mencapai tujuannya.
6. **Konflik (*Conflict*)**. Konflik hadir dikarenakan prosedur dan aturan dalam permainan mencegah pemain mencapai tujuan mereka. Pemain harus menghadapi konflik ini untuk bisa mencapai tujuannya.
7. **Batas (*Boundaries*)**. Batas disini dapat mengacu pada hal-hal yang ada di dunia nyata yang membatasi aktifitas permainan. Hal-hal yang termasuk dalam elemen ini di antaranya luas ruang permainan, spesifikasi tertentu yang dibutuhkan untuk menjalankan game, dan Kebutuhan akan *marker* tertentu.

8. **Hasil (Outcome).** Hasil berbeda dengan tujuan. Hasil merujuk pada ukuran yang ada dalam game untuk menentukan *winning condition* (contoh: skor).

Permainan memancing yang dikembangkan pada penelitian sebelumnya yang dilakukan Luthfansyah (2018) juga tersusun dari elemen-elemen tersebut. Penjelasan tentang elemen-elemen tersebut dijelaskan pada Tabel 2.1.

Tabel 2.1 Formal Elements pada permainan memancing

<i>Formal Elements</i>	Penjelasan
Pemain	Pengguna yang memainkan permainan memancing tersebut
Tujuan	Menangkap ikan
Aturan	<ol style="list-style-type: none"> 1. Ikan hanya bisa ditangkap ketika kail mengenai ikan 2. Ketika kail mengenai ikan, nyawa ikan harus dihabiskan terlebih dahulu dengan tombol “reel” 3. Setiap ikan yang tertangkap akan menambah skor pemain sebanyak seratus 4. Pemain akan lolos ke tahap selanjutnya apabila telah mendapat skor tertentu 5. Skor tersebut harus dipenuhi dalam batas waktu tertentu
Prosedur	<ol style="list-style-type: none"> 1. Menekan tombol “throw” untuk melempar kail 2. Ketika ikan tertangkap, tombol “reel” harus ditekan secepat mungkin 3. Setelah nyawa ikan habis, ikan tertangkap
Sumber daya	Kemampuan pemain untuk menekan tombol “reel” secepat mungkin
Konflik	Adanya batas waktu dan skor minimal membuat sejumlah ikan harus ditangkap secepat mungkin
Batas	Permainan hanya akan dapat dijalankan dengan adanya <i>Marker</i>
Hasil	Skor

Salah satu elemen pada permainan tersebut adalah adanya batas yaitu kebutuhan akan *marker*. Permainan tersebut akan dikembangkan sehingga batas *marker* dapat diatasi. Meskipun begitu bukan berarti penelitian lanjutan akan menghilangkan batas permainan karena perubahan dari *marker* menjadi *markerless* akan membuat batasan baru berupa kemampuan aplikasi *markerless* mengenali lingkungan.

2.1.2 Genre *Game*

Rodgers (2010) mengklasifikasikan *game* menjadi beberapa genre dan subgenre tertentu, di antaranya:

1. *Action*

Action merupakan genre yang mana untuk memainkannya pemain membutuhkan koordinasi antara tangan dan mata. Beberapa subgenre yang masih termasuk dalam genre *action* di antaranya:

- a. **Action-adventure.** Genre ini mengkombinasikan beberapa *gameplay* seperti eksplorasi, pemecahan teka-teki, pencarian dan manajemen *item*, serta fokus pada penceritaan jangka panjang. Beberapa contoh antara lain *Prince of Persia* dan *Tomb Raider*.
- b. **Platformer.** *Game platformer* biasanya memiliki karakter maskot (contoh: Mario, Crash, Donkey Kong) yang melompati (atau berayun atau melewati) rintangan berupa *platform* (biasanya bertingkat seperti sekumpulan panggung atau podium). Element lain seperti tembakan-menembak atau pertarungan juga dapat disertakan meski tidak harus. Beberapa contoh di antaranya seri *Mario* produksi Nintendo (*Super Mario World*, *Mario 64*) dan seri *Metroid* (*Metroid Prime*).
- c. **Stealth.** Salah satu variasi dari genre *action* yang berfokus pada bagaimana pemain menghindari musuh, contohnya Seri *Metal Gear* (*Metal Gear Solid*, *Metal Gear Rising*) dan *Thief: The Dark Project*.
- d. **Fighting.** Sebuah *game* yang mana pemain harus menghadapi dua atau lebih lawan dalam sebuah *setting* tempat tertentu. Genre ini dapat dibedakan dari subgenre *action* lain dari bagaimana kompleksnya kontrol pemain saat pertarungan terjadi. Contohnya adalah seri *Street Fighter* dan seri *Mortal Kombat*.
- e. **Beat 'em up** atau **hack 'n' slash.** Genre ini menempatkan pemain dalam situasi yang mana mereka harus melawan banyak gelombang musuh dengan tingkat kesulitan yang makin meningkat seiring berjalannya waktu. Beberapa contoh di antaranya *Double Dragon* dan *Castle Crashers*.

2. *Shooter*

Genre *Shooter* berfokus pada bagaimana pemain menembakkan proyektil untuk mengalahkan musuh. Dalam perkembangannya genre ini dapat dibagi menjadi beberapa subgenre yang dibedakan melalui penempatan kameranya.

- a. **First Person Shooter (FPS).** *Game shooter* yang mana kamera ditempatkan pada perspektif pemain. Penggunaan kamera ini menyebabkan jarak pandang dan layar menjadi terbatas tapi juga lebih personal. Beberapa contoh di antaranya *Quake*, dan *Team Fortress 2*.

- b. **Shoot 'em up.** *Shoot 'em ups* (atau terkadang disingkat *shmups*) merupakan *shooter* dengan gaya *arcade* yang mana kamera pemain terletak di salah satu sisi pemain (biasanya dari atas atau samping). Beberapa contoh di antaranya *Space Invaders* dan *Contra*.
- c. **Third Person Shooter (TPS).** Game *shooter* yang mana kamera ditempatkan lebih jauh di belakang pemain, memungkinkan pemain untuk melihat sebagian atau penuh lingkungan yang dihuni karakternya. Beberapa contoh di antaranya *Star Wars Battlefront* dan *Grand Theft Auto*.

3. **Adventure**

Game *adventure* berfokus pada eksplorasi, pemecahan puzzle, serta pencarian dan manajemen *item*. Contoh awal game *adventure* seperti *Colossal Cave*, *The King's Quest*, dan *Leisure Suit Larry* hanya terdiri dari teks tanpa grafis penunjang.

- a. **Graphical Adventure.** Terkadang disebut juga sebagai *Point 'n' Click Adventure*, pemain menavigasikan karakternya dan memecahkan puzzle hanya dengan menggunakan *mouse* atau penunjuk *cursor*. Beberapa contoh di antaranya *Myst*, *Monkey Island*, dan *Sam and Max*.
- b. **Role-Playing Game (RPG).** Subgenre ini dikembangkan berdasarkan permainan *Role-Playing* seperti *Dungeons and Dragons*. Pemain memilih sebuah karakter atau kelas tertentu yang mempengaruhi *skill*, status dan kemampuan karakter. Hal tersebut mempengaruhi bagaimana pemain dalam menyelesaikan permainannya. Beberapa contoh di antaranya *Star Wars: Knights of the Old Republic* and *Mass Effect*.
- c. **Massively Multiplayer Online Role-Playing Game (MMORPG).** Permainan RPG yang terdiri dari banyak pemain sekaligus yang saling terhubung secara online. Beberapa contoh di antaranya *World of Warcraft*, dan *DC Universe Online*.
- d. **Survival/Horror.** Pemain ditempatkan dalam sebuah situasi yang mana mereka harus bertahan hidup dalam sebuah skenario yang menakutkan dengan sumber daya yang terbatas, seperti jumlah peluru yang sedikit. Contohnya adalah seri *Resident Evil* dan seri *Silent Hill*.

4. **Construction/Management**

Genre ini berfokus pada bagaimana pemain membangun sebuah objek tertentu dengan sumber daya tertentu. Contoh yang termasuk dalam genre ini adalah *SimCity* dan *Zoo Tycoon*.

5. **Life Simulation**

Meskipun memiliki beberapa kemiripan dengan genre *management*, genre *Life Simulation* berfokus pada bagaimana pemain mengembangkan makhluk hidup virtual. Contoh yang dapat dikategorikan dalam genre ini adalah *The Sims* dan *Princess Maker*.

6. **Music/Rhythm**

Genre yang mana pemain mencapai tujuannya dengan cara menyesuaikan ritme atau *beat* tertentu. Tujuan permainan biasanya untuk mendapatkan poin atau skor tertinggi. Contoh dari genre ini dapat berupa *game* sederhana seperti *Simon* atau *game* rumit seperti *Rock Band*.

7. **Party**

Genre ini berfokus pada permainan yang dilakukan secara berkelompok. Dalam banyak kasus, permainan yang dimainkan biasanya berformat *minigame*. Contohnya adalah *Mario Party*, *Bishi Bashi*, dan *Buzz!*.

8. **Puzzle**

Puzzle merupakan genre yang mana pemain menggunakan logika untuk mencapai tujuannya. Beberapa contoh di antaranya *The Incredible Machine* dan *Tetris*.

9. **Sport**

Sport merupakan genre yang didasarkan pada kompetisi olahraga kompetitif, baik olahraga tradisional atau olahraga ekstrim. Beberapa contoh di antaranya seri *Madden* yang berdasarkan olahraga *Rugby* atau *Tony Hawk Pro Skater* yang berdasarkan olahraga *Skateboard*. Terdapat juga variasi genre *sport* yang mana pemain berperan sebagai pelatih atau manajer dari sebuah tim, bukan pemainnya. Contohnya adalah seri *FIFA Manager* dan seri *NFL Head Coach*.

10. **Strategy**

Ciri khas dari *game strategy* adalah bagaimana pemain harus menggunakan perencanaan dan pemikiran untuk mencapai tujuannya. Contoh dari genre ini dapat berupa *game* sederhana seperti catur atau *game* kompleks seperti *Sid Meir's Civilization*.

- a. **Real-Time Strategy (RTS).** *Game strategy* yang dimainkan secara *real-time*. Subgenre ini berfokus pada empat "x": *expansion, exploration, exploitation, and extermination*. Contohnya adalah seri *Command and Conquer* dan seri *Dawn of War*.
- b. **Turn-Based.** *Game strategy* yang dimainkan secara bergantian antar pemainnya. Tempo permainannya yang cenderung lambat memberikan ruang pada pemainnya untuk berpikir tentang strategi yang dapat digunakan. Contohnya adalah seri *X-Com* dan *Valkyria Chronicle*.
- c. **Tower Defense.** Merupakan salah satu subgenre strategi yang sedang berkembang yang mana pemain harus bertahan menghadapi serangan dengan menggunakan menara yang menembakkan proyektil. Beberapa contoh di antaranya *Plant vs Zombie* dan *Defense Grid: Awakening*.

11. Vehicle Simulation

Genre yang mana pemain mengendarai sebuah kendaraan, baik kendaraan biasa seperti mobil atau kendaraan rumit pesawat ruang angkasa. Fokus utama dari genre ini adalah bagaimana pemain mendapatkan pengalaman mengendarai kendaraan virtual yang nyata seperti pengalaman mengendarai kendaraan aslinya. Contohnya adalah *Lunar Lander* dan *Densha de Go! 64*.

- a. **Driving.** Dalam subgenre ini pemain meningkatkan kemampuan berkendara dan kekuatan kendaraannya untuk menjadi yang terbaik di bidang tersebut. Dalam perkembangannya, genre ini dapat lebih terasa cenderung *action* seperti dalam seri *Need for Speed* atau lebih realistis seperti seri *Grand Turismo*.
- b. **Flying.** Pemain berperan sebagai pilot suatu pesawat terbang. Seperti subgenre *driving*, pengalaman yang dihadirkan dapat terasa lebih aksi seperti seri *Ace Combat* atau lebih realistis seperti *game* Microsoft *Flying Simulator*. Pemain juga dapat mengendarai pesawat hingga ke luar angkasa seperti dalam *Starfox*.

2.1.3 Sport Game

Jenis game yang dikembangkan pada penelitian dikembangkan oleh Luthfansyah (2018) dapat dikategorikan sebagai game *sport*. Menurut Adams (2014), *video game sport* adalah *game* yang melakukan simulasi beberapa aspek nyata atau imajiner dari olahraga atletik, apakah itu aspek permainan dalam suatu pertandingan, aspek mengelola tim atau pemain atau keduanya. Dalam *game sport*, tantangan dapat berupa tantangan fisik, tantangan strategi, atau tantangan manajemen olahraga yang disederhanakan.

Genre *sport* sudah muncul sejak awal dikembangkannya *video game* itu sendiri. Salah satu generasi awal dari *video game* yang dibuat adalah *game* dengan judul *Tennis for Two* yang dikembangkan oleh William Higinbotham. *Game* ini dikembangkan berdasarkan olahraga tenis pada tahun 1958 di Laboratorium Nasional Brookhaven. Pada penelitian Luthfansyah (2018), olahraga yang disimulasikan adalah olahraga memancing, dimana skor akan dihitung berdasarkan banyaknya ikan yang ditangkap dalam waktu tertentu. Untuk olahraga memancing sendiri, berbagai judul sudah dikembangkan mulai dari seri *The Black Bass*, *Fishing Master*, dan *Fishing Planet* yang merupakan sebuah *game* online. Beberapa *game* juga memasukkan unsur memancing sebagai *minigame*, seperti seri *Harvest Moon*, seri *Pokemon*, dan *Final Fantasy XV*.

Satu hal yang perlu diperhatikan adalah meski *sport* merupakan genre yang berusaha untuk membuat simulasi dari permainan olahraga dunia nyata, pengembang perlu menyadari adanya batasan-batasan yang dapat menghalangi olahraga dapat sepenuhnya disimulasikan di dunia virtual. Adams (2006) menekankan bahwa dengan teknologi saat ini, belum ada teknologi *input-output* yang secara nyata dapat membuat pemain merasakan layaknya menjadi seorang

atlet sungguhan dalam lapangan. Namun dengan berkembangnya teknologi, hal ini dapat semakin diatasi di masa yang akan datang. Karena itu, permainan yang dikembangkan pada penelitian sebelumnya hanya akan mensimulasikan aspek memancing seperti melempar kail, menarik ikan dan menahan pancing dengan tumbol-tumbol khusus.

Selain itu, Crawford (2015) berpendapat bahwa tidak semua *video game* bergenre *sport* berusaha secara realistis mensimulasikan olahraga dalam *game* mereka. Contohnya adalah seri *NBA Street* dan *Super Shot Soccer* yang cenderung lebih *cartoonish* dalam merepresentasikan olahraganya. Oleh karena itu, tidak semua olahraga harus disimulasikan secara akurat, dan pengembang *game* hanya meniru aspek tertentu dari olahraga tersebut.

2.2 Augmented Reality

Augmented reality merupakan salah satu bagian dari konsep yang disebut *Reality-Virtuality Continuum (RV Continuum)*. Secara garis besar *augmented reality* dapat didefinisikan sebagai penggabungan objek virtual dan nyata di dalam lingkungan nyata. *Augmented Reality* hadir untuk melengkapi objek dunia nyata dengan objek virtual dengan tujuan tertentu. Penggunaanya merasakan seolah-olah objek virtual dan objek nyata hadir di lingkungan yang sama (Azuma, 1997).



Gambar 2.1 RV Continuum

Sumber: Milgram (1995)

Gambar 2.1 menunjukkan bahwa *augmented reality* menempati dunia yang lebih mendekati dunia nyata. Bagian lain dari *RV Continuum* adalah *augmented virtuality* yang lebih mendekati dunia virtual. *Augmented virtuality* adalah kebalikan dari *augmented reality* yaitu penggabungan objek nyata dan virtual di dalam lingkungan virtual.

Menurut Milgram (1995), objek virtual dapat terlihat di dunia nyata melalui dua jenis layar:

1. Layar monitor.

Augmented reality yang berbasis monitor atau *window-on-the-world* bekerja pada video yang terekam baik secara live ataupun tidak. Pada pendekatan ini dunia nyata direkam terlebih sebelum objek virtual ditampilkan

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Dalam mengembangkan sebuah aplikasi yang memanfaatkan teknologi *Augmented Reality* (AR), salah satu metode yang paling umum digunakan adalah dengan cara memanfaatkan *marker* khusus. *Marker* tersebut berfungsi sebagai titik acuan untuk menentukan posisi dari objek digital di dunia nyata. Dalam prakteknya, kebutuhan akan adanya *marker* menimbulkan masalah baru. Objek digital yang dihasilkan hanya akan stabil bila *marker* terlihat seluruhnya (Raharja, Jalinus, Avileti, 2016). Hal tersebut membuat aplikasi *augmented reality* berbasis *marker* harus memenuhi kondisi-kondisi tertentu untuk dapat menghasilkan objek digital. Karena itu, Diperlukan solusi lain agar objek digital mampu dihasilkan dan stabil tanpa adanya *marker* atau *markerless*.

Manfaat teknologi *markerless augmented reality* akan sangat terasa terutama di bidang hiburan seperti *video game*. *Video game* merupakan media yang dikonsumsi untuk memberikan perasaan senang pada pelakunya (Prensky, 2002). Karena itu, *video game* harus mampu dilakukan dalam kondisi apapun dan di manapun. Memaksa *player* untuk terikat pada *marker* tertentu justru akan membuat bermain *game* tidak lagi menyenangkan. Menurut penelitian sebelumnya yang dilakukan Luthfansyah (2018), 30 persen dari penguji merasa ragu-ragu dengan aspek *augmented reality* dari video game memancing dengan *marker*. Beberapa penguji mengemukakan alasan menjawab ragu-ragu adalah karena tidak terbiasa dengan adanya *marker*.

Salah satu solusi untuk menghilangkan ketergantungan aplikasi *augmented reality* pada kebutuhan akan *marker* khusus adalah dengan mengembangkan aplikasi *augmented reality* yang *markerless*. Pada penelitian yang dilakukan Brito dan Stoyanova (2017), *markerless augmented reality* memiliki beberapa keunggulan dibanding *augmented reality* berbasis *marker*. Penelitian tersebut memperlihatkan aplikasi *markerless augmented reality* lebih memicu gestur tubuh yang lebih positif serta terasa lebih interaktif dibandingkan *augmented reality* berbasis *marker*.

Terdapat beberapa solusi untuk mengembangkan aplikasi *markerless augmented reality*. Salah satu metode yang dapat digunakan adalah metode *Simultaneous Localization and Mapping* (SLAM). SLAM merupakan pendekatan mengenali struktur dunia nyata dengan melakukan *mapping* di tiap citra dengan membandingkan posisi pola garis, pola titik, pola warna, dan dan pola-pola lain yang terbaca. SLAM membandingkan citra yang dikenali oleh perangkat dengan citra sebelumnya, kemudian melakukan perhitungan untuk menentukan tempat relatif objek. Untuk setiap citra yang masuk, aplikasi akan melakukan perhitungan untuk menentukan koordinat baru dari objek digital. Perlu diperhatikan bahwa SLAM bukanlah sebuah algoritme khusus melainkan suatu pendekatan. Setiap peneliti mungkin memiliki algoritme berbeda, tapi algoritme tersebut memiliki tujuan yang sama yaitu untuk melakukan *mapping* secara *real-time* (Riisgaard dan Blas, 2005: 6).

Metode SLAM sendiri lebih banyak digunakan di bidang robotika. Di bidang robotika, SLAM dimanfaatkan untuk menciptakan sebuah robot yang mampu mengenali lingkungannya dan bergerak tanpa adanya masukan khusus dari luar (Claraco, 2009). Di bidang *video game*, *augmented reality* yang menggunakan pendekatan SLAM dikembangkan oleh beberapa perusahaan teknologi seperti Wikitude, Kudan, dan VOID. Perusahaan-perusahaan tersebut menyediakan *package* untuk membantu para peneliti luar dalam mengembang *video game* dengan *markerless augmented reality* untuk berbagai *platform*.

Dari latar belakang di atas, maka disusunlah penelitian Analisis penerapan *video game markerless augmented reality* memancing dengan pendekatan *Simultaneous Localization and Mapping* (SLAM). Penelitian ini akan mengembangkan sebuah *video game* memancing *augmented reality* yang sudah disusun oleh Luthfansyah pada 2018. Hasil penelitian tersebut akan dikembangkan sehingga *video game* memancing *augmented reality* tersebut dapat dimainkan secara *markerless*.

1.2 Rumusan Masalah

Berdasarkan latar belakang berikut, maka rumusan masalah dapat dituliskan sebagai berikut:

1. Bagaimana merancang sebuah *video game augmented reality* memancing yang memungkinkan pengguna untuk tidak terikat dengan *marker*?
2. Bagaimana tingkat keberhasilan aplikasi *video game augmented reality* memancing yang tidak menggunakan *marker* (*markerless*)?

1.3 Tujuan

Tujuan dari penyusunan skripsi ini adalah:

1. Merancang sebuah *video game augmented reality* yang memungkinkan pengguna untuk tidak terikat dengan kebutuhan *marker*.
2. Mengukur tingkat keberhasilan metode SLAM saat menjalankan *video game augmented reality* memancing.

1.4 Manfaat

Manfaat dari penelitian ini adalah:

1. Bagi Penulis: Untuk menganalisis sebuah *video game augmented reality* yang tidak bergantung pada *marker* tertentu.
2. Bagi pengguna: Untuk menghilangkan kebutuhan akan adanya *marker* khusus saat bermain *video game* yang menggunakan teknologi *augmented reality*.
3. Bagi institusi: Untuk meneliti pendekatan SLAM dalam mengembangkan sebuah *video game* dengan teknologi *markerless augmented reality*.

1.5 Batasan Masalah

Batasan-batasan masalah dari penelitian skripsi ini adalah sebagai berikut:

1. *Video game* dibuat untuk *platform* Android.
2. *Augmented Reality* dibuat dengan *package* VOID AR yang tersedia dalam *game engine* Unity.

1.6 Sistematika Pembahasan

Sistematika dari penelitian analisis pengembangan *video game markerless augmented reality* memancing adalah sebagai berikut:

Bab 1: Pendahuluan

Bagian ini merupakan pembahasan tentang tema, masalah, atau pertanyaan yang hendak dijawab oleh penelitian ini. Dari bagian ini dirumuskan apa saja masalah-masalah yang dapat dipecahkan oleh penelitian ini.

Bab 2: Landasan Kepustakaan

Bagian ini berisi pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian. Pembahasan didasarkan oleh studi literatur dari berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian.

Bab 3: Metodologi

Metodologi penelitian adalah proses atau cara ilmiah untuk mendapatkan data yang akan digunakan untuk keperluan penelitian (Sugiyono, 2012). Bagian ini berisi pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian. Pembahasan didasarkan oleh studi literatur dari berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian. Bagian ini berisi tata cara dan alur penelitian.

Bab 4: Perancangan dan Implementasi

Bagian ini menjelaskan dengan detail bagaimana *video game* dirancang sebelum dikembangkan. Perancangan *video game* didasarkan pada latar belakang, rumusan masalah, dan metodologi yang telah dideskripsikan pada bagian sebelumnya. Setelah perancangan ditentukan, pengembangan *video game* dilakukan sesuai dengan rancangan yang telah dibuat. Rancangan tersebut akan diimplementasikan agar dapat menjadi sebuah *video game* yang utuh.

Bab 5: Pengujian dan Analisis

Bagian ini menjelaskan bagaimana pengujian dari *video game* dilakukan. Bagian ini mencakup skenario pengujian yang akan dilakukan, proses pengujian, dan hasil dari pengujian. Setelah dilakukan pengujian, analisis dari pengujian

akan dilakukan berdasarkan informasi-informasi yang didapat dari hasil pengujian.

Bab 6: Kesimpulan dan Saran

Bagian ini menyimpulkan hasil penelitian yang telah dilakukan. Kesimpulan yang ditarik pada penelitian ini diharapkan mampu menjawab rumusan masalah yang telah dirumuskan di bab sebelumnya. Pada bagian ini, peneliti juga dapat memberikan saran-saran untuk membantu proses penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Game

Game adalah kompetisi antara para pemain yang berinteraksi satu sama lain dengan menggunakan aturan tertentu untuk mencapai tujuan tertentu (Sadiman, 2010). Pemain dituntut untuk mendapatkan hasil yang telah ditentukan dengan batasan-batasan yang telah ditentukan oleh *game* tersebut. Pemain dapat dinyatakan menang jika berhasil memperoleh hasil yang sesuai dengan apa yang dikehendaki oleh aturan. *Game* berkembang di banyak media, salah satunya dalam media video. *Game* yang dimainkan dalam bentuk video dalam layar khusus disebut *video game* (Rogers, 2010).

2.1.1 Formal Element Game

Menurut Nacke (2014) *game* terdiri dari sekumpulan elemen dasar yang disebut *formal elements*. Elemen-elemen tersebut saling berpengaruh satu sama lain. Perubahan salah satu elemen akan membuat bentuk keseluruhan *game* tersebut berubah. Beberapa elemen yang dimaksud adalah:

1. **Pemain (*Player*)**. Setiap *game* selalu memiliki pemain yang berinteraksi dengan sistem game. Aktifitas bermain game biasanya dilakukan pemain untuk bersenang-senang. Bagi pemain, aktifitas tersebut dilakukan secara sukarela. Permainan dapat dilakukan secara berkelompok ataupun sendiri-sendiri.
2. **Tujuan (*Objective*)**. Sebuah *game* memiliki tujuan tertentu yang harus dilakukan oleh pemain. Tujuan setiap game berbeda-beda tergantung sistemnya masing-masing.
3. **Aturan (*Rules*)**. Aturan dibuat untuk membatasi aktifitas pemain. Pemain diharapkan untuk bisa mencapai tujuannya dengan segala batasan aturan yang telah ditetapkan sistem game
4. **Prosedur (*Procedures*)**. Prosedur merujuk pada cara pemain untuk mencapai tujuannya. Prosedur dapat berupa instruksi spesifik yang harus dilakukan atau kontrol yang dimiliki pemain.
5. **Sumber Daya (*Resources*)**. Sumber daya adalah objek yang membantu pemain mencapai tujuannya.
6. **Konflik (*Conflict*)**. Konflik hadir dikarenakan prosedur dan aturan dalam permainan mencegah pemain mencapai tujuan mereka. Pemain harus menghadapi konflik ini untuk bisa mencapai tujuannya.
7. **Batas (*Boundaries*)**. Batas disini dapat mengacu pada hal-hal yang ada di dunia nyata yang membatasi aktifitas permainan. Hal-hal yang termasuk dalam elemen ini di antaranya luas ruang permainan, spesifikasi tertentu yang dibutuhkan untuk menjalankan game, dan Kebutuhan akan *marker* tertentu.

8. **Hasil (Outcome).** Hasil berbeda dengan tujuan. Hasil merujuk pada ukuran yang ada dalam game untuk menentukan *winning condition* (contoh: skor).

Permainan memancing yang dikembangkan pada penelitian sebelumnya yang dilakukan Luthfansyah (2018) juga tersusun dari elemen-elemen tersebut. Penjelasan tentang elemen-elemen tersebut dijelaskan pada Tabel 2.1.

Tabel 2.1 Formal Elements pada permainan memancing

<i>Formal Elements</i>	Penjelasan
Pemain	Pengguna yang memainkan permainan memancing tersebut
Tujuan	Menangkap ikan
Aturan	<ol style="list-style-type: none"> 1. Ikan hanya bisa ditangkap ketika kail mengenai ikan 2. Ketika kail mengenai ikan, nyawa ikan harus dihabiskan terlebih dahulu dengan tombol “reel” 3. Setiap ikan yang tertangkap akan menambah skor pemain sebanyak seratus 4. Pemain akan lolos ke tahap selanjutnya apabila telah mendapat skor tertentu 5. Skor tersebut harus dipenuhi dalam batas waktu tertentu
Prosedur	<ol style="list-style-type: none"> 1. Menekan tombol “throw” untuk melempar kail 2. Ketika ikan tertangkap, tombol “reel” harus ditekan secepat mungkin 3. Setelah nyawa ikan habis, ikan tertangkap
Sumber daya	Kemampuan pemain untuk menekan tombol “reel” secepat mungkin
Konflik	Adanya batas waktu dan skor minimal membuat sejumlah ikan harus ditangkap secepat mungkin
Batas	Permainan hanya akan dapat dijalankan dengan adanya <i>Marker</i>
Hasil	Skor

Salah satu elemen pada permainan tersebut adalah adanya batas yaitu kebutuhan akan *marker*. Permainan tersebut akan dikembangkan sehingga batas *marker* dapat diatasi. Meskipun begitu bukan berarti penelitian lanjutan akan menghilangkan batas permainan karena perubahan dari *marker* menjadi *markerless* akan membuat batasan baru berupa kemampuan aplikasi *markerless* mengenali lingkungan.

2.1.2 Genre *Game*

Rodgers (2010) mengklasifikasikan *game* menjadi beberapa genre dan subgenre tertentu, di antaranya:

1. *Action*

Action merupakan genre yang mana untuk memainkannya pemain membutuhkan koordinasi antara tangan dan mata. Beberapa subgenre yang masih termasuk dalam genre *action* di antaranya:

- a. **Action-adventure.** Genre ini mengkombinasikan beberapa *gameplay* seperti eksplorasi, pemecahan teka-teki, pencarian dan manajemen *item*, serta fokus pada penceritaan jangka panjang. Beberapa contoh antara lain *Prince of Persia* dan *Tomb Raider*.
- b. **Platformer.** *Game platformer* biasanya memiliki karakter maskot (contoh: Mario, Crash, Donkey Kong) yang melompati (atau berayun atau melewati) rintangan berupa *platform* (biasanya bertingkat seperti sekumpulan panggung atau podium). Element lain seperti tembakan-menembak atau pertarungan juga dapat disertakan meski tidak harus. Beberapa contoh di antaranya seri *Mario* produksi Nintendo (*Super Mario World*, *Mario 64*) dan seri *Metroid* (*Metroid Prime*).
- c. **Stealth.** Salah satu variasi dari genre *action* yang berfokus pada bagaimana pemain menghindari musuh, contohnya Seri *Metal Gear* (*Metal Gear Solid*, *Metal Gear Rising*) dan *Thief: The Dark Project*.
- d. **Fighting.** Sebuah *game* yang mana pemain harus menghadapi dua atau lebih lawan dalam sebuah *setting* tempat tertentu. Genre ini dapat dibedakan dari subgenre *action* lain dari bagaimana kompleksnya kontrol pemain saat pertarungan terjadi. Contohnya adalah seri *Street Fighter* dan seri *Mortal Kombat*.
- e. **Beat 'em up** atau **hack 'n' slash.** Genre ini menempatkan pemain dalam situasi yang mana mereka harus melawan banyak gelombang musuh dengan tingkat kesulitan yang makin meningkat seiring berjalannya waktu. Beberapa contoh di antaranya *Double Dragon* dan *Castle Crashers*.

2. *Shooter*

Genre *Shooter* berfokus pada bagaimana pemain menembakkan proyektil untuk mengalahkan musuh. Dalam perkembangannya genre ini dapat dibagi menjadi beberapa subgenre yang dibedakan melalui penempatan kameranya.

- a. **First Person Shooter (FPS).** *Game shooter* yang mana kamera ditempatkan pada perspektif pemain. Penggunaan kamera ini menyebabkan jarak pandang dan layar menjadi terbatas tapi juga lebih personal. Beberapa contoh di antaranya *Quake*, dan *Team Fortress 2*.

- b. **Shoot 'em up.** *Shoot 'em ups* (atau terkadang disingkat *shmups*) merupakan *shooter* dengan gaya *arcade* yang mana kamera pemain terletak di salah satu sisi pemain (biasanya dari atas atau samping). Beberapa contoh di antaranya *Space Invaders* dan *Contra*.
- c. **Third Person Shooter (TPS).** Game *shooter* yang mana kamera ditempatkan lebih jauh di belakang pemain, memungkinkan pemain untuk melihat sebagian atau penuh lingkungan yang dihuni karakternya. Beberapa contoh di antaranya *Star Wars Battlefront* dan *Grand Theft Auto*.

3. **Adventure**

Game *adventure* berfokus pada eksplorasi, pemecahan puzzle, serta pencarian dan manajemen *item*. Contoh awal game *adventure* seperti *Colossal Cave*, *The King's Quest*, dan *Leisure Suit Larry* hanya terdiri dari teks tanpa grafis penunjang.

- a. **Graphical Adventure.** Terkadang disebut juga sebagai *Point 'n' Click Adventure*, pemain menavigasikan karakternya dan memecahkan puzzle hanya dengan menggunakan *mouse* atau penunjuk *cursor*. Beberapa contoh di antaranya *Myst*, *Monkey Island*, dan *Sam and Max*.
- b. **Role-Playing Game (RPG).** Subgenre ini dikembangkan berdasarkan permainan *Role-Playing* seperti *Dungeons and Dragons*. Pemain memilih sebuah karakter atau kelas tertentu yang mempengaruhi *skill*, status dan kemampuan karakter. Hal tersebut mempengaruhi bagaimana pemain dalam menyelesaikan permainannya. Beberapa contoh di antaranya *Star Wars: Knights of the Old Republic* and *Mass Effect*.
- c. **Massively Multiplayer Online Role-Playing Game (MMORPG).** Permainan RPG yang terdiri dari banyak pemain sekaligus yang saling terhubung secara online. Beberapa contoh di antaranya *World of Warcraft*, dan *DC Universe Online*.
- d. **Survival/Horror.** Pemain ditempatkan dalam sebuah situasi yang mana mereka harus bertahan hidup dalam sebuah skenario yang menakutkan dengan sumber daya yang terbatas, seperti jumlah peluru yang sedikit. Contohnya adalah seri *Resident Evil* dan seri *Silent Hill*.

4. **Construction/Management**

Genre ini berfokus pada bagaimana pemain membangun sebuah objek tertentu dengan sumber daya tertentu. Contoh yang termasuk dalam genre ini adalah *SimCity* dan *Zoo Tycoon*.

5. **Life Simulation**

Meskipun memiliki beberapa kemiripan dengan genre *management*, genre *Life Simulation* berfokus pada bagaimana pemain mengembangkan makhluk hidup virtual. Contoh yang dapat dikategorikan dalam genre ini adalah *The Sims* dan *Princess Maker*.

6. **Music/Rhythm**

Genre yang mana pemain mencapai tujuannya dengan cara menyesuaikan ritme atau *beat* tertentu. Tujuan permainan biasanya untuk mendapatkan poin atau skor tertinggi. Contoh dari genre ini dapat berupa *game* sederhana seperti *Simon* atau *game* rumit seperti *Rock Band*.

7. **Party**

Genre ini berfokus pada permainan yang dilakukan secara berkelompok. Dalam banyak kasus, permainan yang dimainkan biasanya berformat *minigame*. Contohnya adalah *Mario Party*, *Bishi Bashi*, dan *Buzz!*.

8. **Puzzle**

Puzzle merupakan genre yang mana pemain menggunakan logika untuk mencapai tujuannya. Beberapa contoh di antaranya *The Incredible Machine* dan *Tetris*.

9. **Sport**

Sport merupakan genre yang didasarkan pada kompetisi olahraga kompetitif, baik olahraga tradisional atau olahraga ekstrim. Beberapa contoh di antaranya seri *Madden* yang berdasarkan olahraga *Rugby* atau *Tony Hawk Pro Skater* yang berdasarkan olahraga *Skateboard*. Terdapat juga variasi genre *sport* yang mana pemain berperan sebagai pelatih atau manajer dari sebuah tim, bukan pemainnya. Contohnya adalah seri *FIFA Manager* dan seri *NFL Head Coach*.

10. **Strategy**

Ciri khas dari *game strategy* adalah bagaimana pemain harus menggunakan perencanaan dan pemikiran untuk mencapai tujuannya. Contoh dari genre ini dapat berupa *game* sederhana seperti catur atau *game* kompleks seperti *Sid Meir's Civilization*.

- a. **Real-Time Strategy (RTS).** *Game strategy* yang dimainkan secara *real-time*. Subgenre ini berfokus pada empat "x": *expansion, exploration, exploitation, and extermination*. Contohnya adalah seri *Command and Conquer* dan seri *Dawn of War*.
- b. **Turn-Based.** *Game strategy* yang dimainkan secara bergantian antar pemainnya. Tempo permainannya yang cenderung lambat memberikan ruang pada pemainnya untuk berpikir tentang strategi yang dapat digunakan. Contohnya adalah seri *X-Com* dan *Valkyria Chronicle*.
- c. **Tower Defense.** Merupakan salah satu subgenre strategi yang sedang berkembang yang mana pemain harus bertahan menghadapi serangan dengan menggunakan menara yang menembakkan proyektil. Beberapa contoh di antaranya *Plant vs Zombie* dan *Defense Grid: Awakening*.

11. *Vehicle Simulation*

Genre yang mana pemain mengendarai sebuah kendaraan, baik kendaraan biasa seperti mobil atau kendaraan rumit pesawat ruang angkasa. Fokus utama dari genre ini adalah bagaimana pemain mendapatkan pengalaman mengendarai kendaraan virtual yang nyata seperti pengalaman mengendarai kendaraan aslinya. Contohnya adalah *Lunar Lander* dan *Densha de Go! 64*.

- a. **Driving.** Dalam subgenre ini pemain meningkatkan kemampuan berkendara dan kekuatan kendaraannya untuk menjadi yang terbaik di bidang tersebut. Dalam perkembangannya, genre ini dapat lebih terasa cenderung *action* seperti dalam seri *Need for Speed* atau lebih realistis seperti seri *Grand Turismo*.
- b. **Flying.** Pemain berperan sebagai pilot suatu pesawat terbang. Seperti subgenre *driving*, pengalaman yang dihadirkan dapat terasa lebih aksi seperti seri *Ace Combat* atau lebih realistis seperti *game* Microsoft *Flying Simulator*. Pemain juga dapat mengendarai pesawat hingga ke luar angkasa seperti dalam *Starfox*.

2.1.3 *Sport Game*

Jenis game yang dikembangkan pada penelitian dikembangkan oleh Luthfansyah (2018) dapat dikategorikan sebagai game *sport*. Menurut Adams (2014), *video game sport* adalah *game* yang melakukan simulasi beberapa aspek nyata atau imajiner dari olahraga atletik, apakah itu aspek permainan dalam suatu pertandingan, aspek mengelola tim atau pemain atau keduanya. Dalam *game sport*, tantangan dapat berupa tantangan fisik, tantangan strategi, atau tantangan manajemen olahraga yang disederhanakan.

Genre *sport* sudah muncul sejak awal dikembangkannya *video game* itu sendiri. Salah satu generasi awal dari *video game* yang dibuat adalah *game* dengan judul *Tennis for Two* yang dikembangkan oleh William Higinbotham. *Game* ini dikembangkan berdasarkan olahraga tenis pada tahun 1958 di Laboratorium Nasional Brookhaven. Pada penelitian Luthfansyah (2018), olahraga yang disimulasikan adalah olahraga memancing, dimana skor akan dihitung berdasarkan banyaknya ikan yang ditangkap dalam waktu tertentu. Untuk olahraga memancing sendiri, berbagai judul sudah dikembangkan mulai dari seri *The Black Bass*, *Fishing Master*, dan *Fishing Planet* yang merupakan sebuah *game* online. Beberapa *game* juga memasukkan unsur memancing sebagai *minigame*, seperti seri *Harvest Moon*, seri *Pokemon*, dan *Final Fantasy XV*.

Satu hal yang perlu diperhatikan adalah meski *sport* merupakan genre yang berusaha untuk membuat simulasi dari permainan olahraga dunia nyata, pengembang perlu menyadari adanya batasan-batasan yang dapat menghalangi olahraga dapat sepenuhnya disimulasikan di dunia virtual. Adams (2006) menekankan bahwa dengan teknologi saat ini, belum ada teknologi *input-output* yang secara nyata dapat membuat pemain merasakan layaknya menjadi seorang

atlet sungguhan dalam lapangan. Namun dengan berkembangnya teknologi, hal ini dapat semakin diatasi di masa yang akan datang. Karena itu, permainan yang dikembangkan pada penelitian sebelumnya hanya akan mensimulasikan aspek memancing seperti melempar kail, menarik ikan dan menahan pancing dengan tumbol-tumbol khusus.

Selain itu, Crawford (2015) berpendapat bahwa tidak semua *video game* bergenre *sport* berusaha secara realistis mensimulasikan olahraga dalam *game* mereka. Contohnya adalah seri *NBA Street* dan *Super Shot Soccer* yang cenderung lebih *cartoonish* dalam merepresentasikan olahraganya. Oleh karena itu, tidak semua olahraga harus disimulasikan secara akurat, dan pengembang *game* hanya meniru aspek tertentu dari olahraga tersebut.

2.2 Augmented Reality

Augmented reality merupakan salah satu bagian dari konsep yang disebut *Reality-Virtuality Continuum (RV Continuum)*. Secara garis besar *augmented reality* dapat didefinisikan sebagai penggabungan objek virtual dan nyata di dalam lingkungan nyata. *Augmented Reality* hadir untuk melengkapi objek dunia nyata dengan objek virtual dengan tujuan tertentu. Penggunaanya merasakan seolah-olah objek virtual dan objek nyata hadir di lingkungan yang sama (Azuma, 1997).



Gambar 2.1 RV Continuum

Sumber: Milgram (1995)

Gambar 2.1 menunjukkan bahwa *augmented reality* menempati dunia yang lebih mendekati dunia nyata. Bagian lain dari *RV Continuum* adalah *augmented virtuality* yang lebih mendekati dunia virtual. *Augmented virtuality* adalah kebalikan dari *augmented reality* yaitu penggabungan objek nyata dan virtual di dalam lingkungan virtual.

Menurut Milgram (1995), objek virtual dapat terlihat di dunia nyata melalui dua jenis layar:

1. Layar monitor.

Augmented reality yang berbasis monitor atau *window-on-the-world* bekerja pada video yang terekam baik secara live ataupun tidak. Pada pendekatan ini dunia nyata direkam terlebih sebelum objek virtual ditampilkan

2. Layar tembus (*See through screen*)

Layar tembus merupakan cara melihat objek virtual tanpa adanya perekaman. Objek virtual dapat ditampilkan secara langsung secara real-time dan komputasi dilakukan langsung oleh perangkat yang memiliki layar.

Jenis layar yang akan digunakan pada penelitian ini adalah layar tembus menggunakan perangkat *smartphone* berbasis *android*. Objek virtual akan ditampilkan langsung pada layar *smartphone* melalui kamera. *Smartphone* juga digunakan untuk berinteraksi dengan objek virtual tersebut.

Menurut Laksono dan Rohman (2014), terdapat dua metode *augmented reality* yang berkembang secara signifikan dalam beberapa tahun terakhir:

1. *Marker*

Teknologi *augmented reality* yang menggunakan objek sejenis *barcode* tertentu untuk menampilkan objek virtual.

2. *Markerless*

Teknologi *augmented reality* yang tidak membutuhkan objek tertentu untuk menampilkan objek virtualnya.

Pada penelitian Luthfansyah (2018), *augmented reality* dikembangkan dengan basis *marker*. Hasil penelitian tersebut akan dikembangkan lebih lanjut agar permainan memancing yang dikembangkan dapat dimainkan secara *markerless*.

2.3 *Markerless Augmented Reality*

Markerless augmented reality merupakan istilah yang ditunjukkan pada teknologi *Augmented Reality* yang tidak membutuhkan suatu pengetahuan khusus tentang lingkungan pengguna untuk menampilkan objek virtualnya pada suatu titik tertentu. Dalam *markerless augmented reality*, sistem harus mengidentifikasi objek dan tempat di dunia nyata tanpa *marker* khusus. Identifikasi dilakukan dengan menggunakan informasi-informasi seperti koordinat lokasi, orientasi, dan pergerakan agen (Janoso, 2010).

Perkembangan *markerless augmented reality* banyak dipengaruhi oleh berkembangnya perangkat penunjang berbasis *mobile* seperti *smartphone* di masyarakat. Sebelum berkembangnya perangkat penunjang, pengembangan aplikasi *augmented reality* lebih banyak menggunakan *marker*. Namun kekurangan teknologi *markerless Augmented Reality* mulai dapat diatasi dengan dengan berkembangnya teknologi sensor dan kamera. (Fetters, 2014).

Ziegler (2010) merumuskan prosedur umum dari sistem *tracking* pada *Markerless Augmented Reality* sebagai berikut:

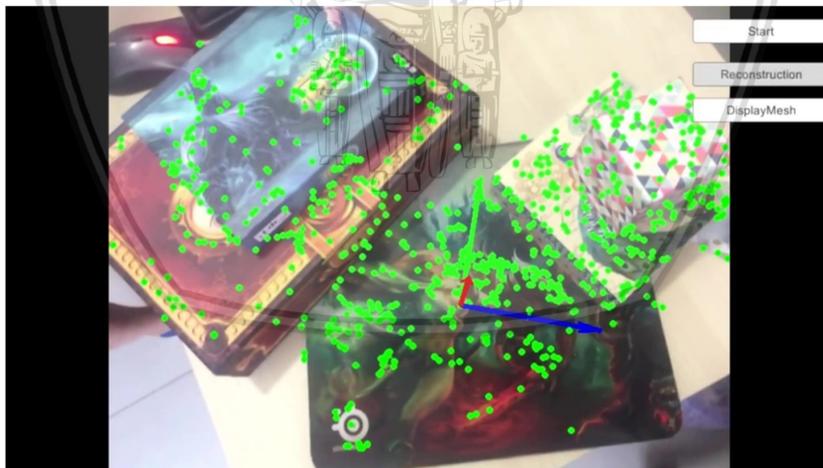
1. Sistem mendeteksi pola-pola umum yang terlihat dalam layar.
2. Sistem membandingkan pola-pola umum tersebut dan menyesuaikannya.

3. Memperkirakan bagaimana pengaruh kamera dengan berpatokan pada objek 3D yang telah diketahui posisinya.
4. Menghitung posisi 3D baru yang belum diketahui.

Menurut Jang, et al (2018), teknologi *markerless augmented reality* memiliki keuntungan karena kemampuannya untuk mengenali sudut, arah, cahaya, dan pola-pola lain yang berubah-ubah. Salah satu kekurangan terbesar dari implementasi *markerless augmented reality* yang berbasis mobile adalah kemampuan sumber daya *smartphone* yang tidak sebaik PC. Komputasi dari *markerless augmented reality* lebih rumit daripada *augmented reality* yang menggunakan *marker*. Hal inilah yang menyebabkan diperlukannya algoritme khusus yang mampu melakukan *tracking* lingkungan pada dunia nyata secara efisien.

2.4 Deteksi dan Pelacakan Pola

Salah satu pendekatan untuk menampilkan objek virtual dalam *markerless augmented reality* adalah dengan melakukan deteksi dan pelacakan pola-pola natural yang terlihat pada citra. Pola-pola tersebut akan dideteksi agar sistem dapat mengenali lingkungan. Menurut Comport dan Pressigout (2006), pola di dalam citra dapat berupa titik, segmen, garis lurus, kontur atau suatu titik pada kontur, objek kerucut, objek silinder, atau kombinasi antara elemen-elemen tersebut.



Gambar 2.2 Pelacakan pola pada VOID AR

(Sumber: Kanal Youtube VOID AR)

Contoh pelacakan dan deteksi pola dapat dilihat pada Gambar 2.2. Pola-pola yang terdeteksi akan ditandai dengan warna titik-titik berwarna hijau. Pola-pola yang terbaca akan dijadikan patokan untuk melakukan *mapping* lingkungan.

2.6 Unity

Unity merupakan salah satu perangkat lunak yang berfungsi sebagai *game engine*. Unity dikembangkan pada tahun 2004 oleh David Helgason, Nicholas Francis, dan Joachim Ante sebagai bentuk kepedulian pada pengembang *video game* Indie yang tidak bisa mengembangkan *video game* sendiri karena masalah biaya. Unity tersedia secara gratis dan dapat digunakan untuk membuat permainan berbasis 2D maupun 3D. Unity dapat digunakan untuk melakukan pengembangan *video game* dalam berbagai platform seperti, web, Mac, flash, Windows, android, iOS, Xbox, Wii, dan Playstation 3. Terdapat berbagai macam bahasa yang disediakan dalam Unity termasuk Javascript, C#, dan Boo. Bahasa yang umumnya dipakai adalah Javascript dan C# (Roedavan, 2014).

2.7 VOID AR

VOID AR adalah sebuah SDK (*Software Development Kit*) yang dikembangkan oleh Chengdu Miyo Network Technology Co., Ltd, sebuah perusahaan berbasis Tiongkok yang bergerak di bidang teknologi. VOID AR dikembangkan untuk membantu para peneliti untuk merancang perangkat lunak yang menggunakan teknologi *Augmented Reality*.

Menurut website VOID AR, Beberapa fitur yang disediakan VOID AR di antaranya adalah:

1. *Image Tracker*

Image Tracker digunakan untuk mengembangkan aplikasi *augmented reality* yang menggunakan *marker*. VOID AR dapat mengenali gambar yang sangat detail (kartu nama, halaman majalah, dan lain-lain) dan menggunakannya sebagai *marker*.

2. VOID CLOUD

Sebuah *library* untuk menyimpan target *image* pada *augmented reality* berbasis *marker*. *Image* yang digunakan sebagai *marker* akan tersimpan di dalam *cloud* dan dapat diakses sewaktu-waktu.

3. *Multi-target Recognition*

Multi-target Recognition merupakan fitur untuk mengenali banyak *marker* sekaligus di dalam satu layar kamera.

4. *Video Playback*

Fitur *Video Playback* merupakan fitur untuk menampilkan video langsung ke gambar target dalam *augmented reality*. Dengan begitu, objek virtual yang dapat ditampilkan tidak hanya terbatas pada objek 3D melainkan juga video.

5. VOID SLAM

VOID SLAM merupakan fitur untuk menampilkan objek virtual di dunia nyata tanpa adanya *marker* khusus dengan pendekatan SLAM.

Untuk mengembangkan perangkat lunak *markerless augmented reality*, peneliti dapat menggunakan fitur VOID SLAM. SDK untuk VOID SLAM tersedia secara gratis di halaman resmi VOID AR (<https://www.voidar.net>) dan dapat dikembangkan dengan *game engine* Unity. Salah satu keuntungan mengembangkan perangkat lunak *markerless augmented reality* dengan VOID AR adalah SDK VOID AR tidak membutuhkan koneksi internet untuk menampilkan objek virtualnya, sedangkan kelemahannya adalah untuk versi gratis terdapat *watermark* yang muncul di kiri bawah layar ketika aplikasi berjalan.

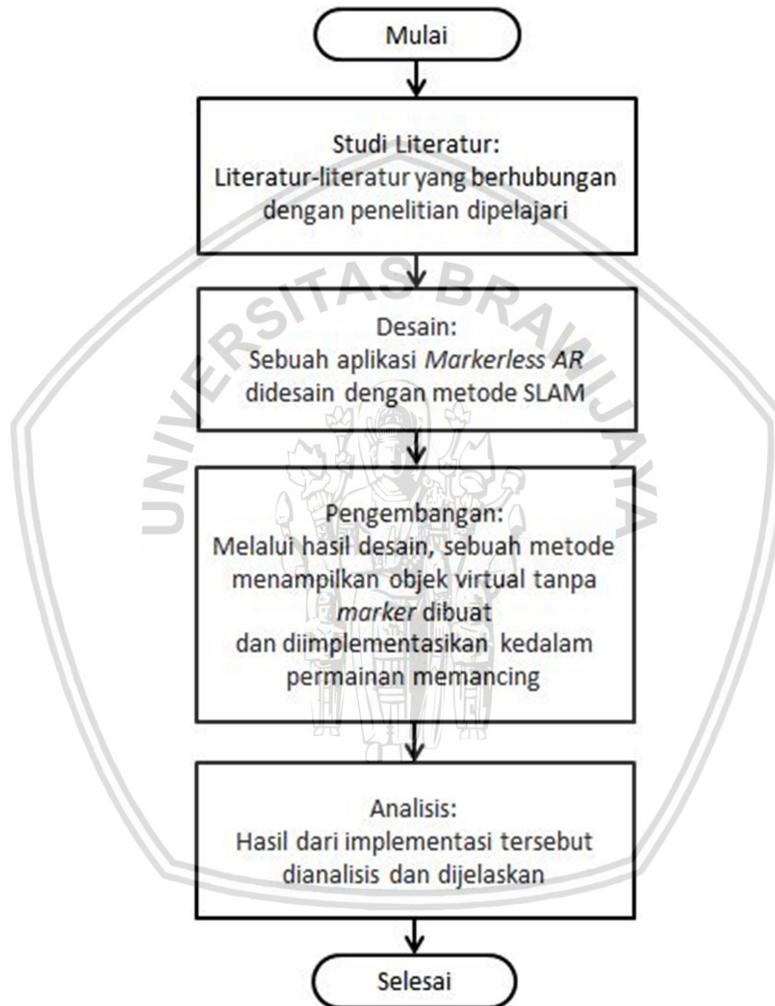
2.8 Pengujian *Black Box*

Pengujian *Black Box* (*Black box testing*) merupakan pengujian yang berfokus pada sisi fungsional pada perangkat lunak. Menurut Mustaqbal, Firdaus, dan Rahmadi (2015), pengujian *black box* dilakukan pada spesifikasi fungsional perangkat lunak dengan terlebih dahulu mendefinisikan kumpulan kondisi *input*. Pengujian *black box* dilakukan untuk menemukan jawaban dari pertanyaan-pertanyaan berikut:

1. Pengujian seperti apa yang dilakukan pada fungsi-fungsi untuk dapat dinyatakan valid?
2. Bagaimana menemukan *input* yang merupakan bahan kasus uji yang baik?
3. Apakah suatu *input* tertentu direspon lebih cepat oleh sistem?
4. Bagaimana cara mengisolasi kumpulan data?
5. Bagaimana kemampuan sistem dalam menangani jumlah data dan rata-rata data?
6. Apakah terdapat suatu efek yang memungkinkan operasi sistem untuk menangani suatu kombinasi data secara spesifik?

BAB 3 METODOLOGI

Bab ini menjelaskan tentang tahapan penelitian yang akan dilakukan pada skripsi berjudul Analisis Pengembangan *Markerless Augmented Reality* Pada *Video Game* Memancing Dengan Pendekatan *Simultaneous Localization and Mapping* (SLAM). Tahap-tahap penelitian pada *video game markerless augmented reality* memancing dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tahapan penelitian

3.1 Studi Literatur

Studi literatur yang akan dilakukan pada penelitian ini dilakukan untuk mempelajari informasi-informasi yang dibutuhkan dalam pengembangan *video game markerless augmented reality* memancing. Literatur diperoleh dari berbagai sumber referensi yaitu buku, jurnal, *e-book*, dan berbagai sumber lain. Berikut beberapa literatur yang dipelajari untuk digunakan pada penelitian ini:

1. Teori mengenai *augmented reality*.

Penelitian ini merupakan analisis pengembangan *video game markerless augmented reality* memancing. Dasar-dasar ilmu yang berhubungan dengan *augmented reality* dipelajari agar peneliti mampu mengembangkan sebuah *video game* yang menggunakan teknologi *markerless augmented reality*.

2. Penggunaan Unity.

Pada penelitian ini, *game engine* Unity digunakan untuk mengembangkan sebuah *video game markerless augmented reality* memancing. Pemahaman tentang penggunaan alat, fitur, serta *scripting* pada *game engine* Unity diperlukan agar metode SLAM pada *video game* mampu di implementasikan dengan baik dan sesuai dengan perancangannya.

3. Teori *Simultaneous Localization and Mapping*.

Simultaneous Localization and Mapping (SLAM) merupakan metode yang digunakan sebagai solusi pada penelitian ini. Dasar-dasar ilmu yang berhubungan SLAM perlu dipelajari agar sebuah *video game markerless augmented reality* memancing dapat dirancang dengan baik.

4. Implementasi VOID AR.

VOID AR adalah sebuah SDK yang digunakan untuk mengimplementasikan metode SLAM pada penelitian ini. Untuk mengimplementasikan VOID AR pada *video game augmented reality* memancing, studi tentang implementasi SDK VOID AR pada Unity perlu dilakukan agar *markerless augmented reality* dapat dikembangkan.

5. Dasar-dasar pengujian *game*.

Video game yang telah dirancang kemudian akan diuji untuk menentukan apakah permasalahan yang ada mampu dipecahkan dengan metode yang telah ditentukan. Untuk itu, teori-teori tentang pengujian *video game* perlu dipelajari agar cara dan metode pengujian *video game markerless augmented reality* memancing dapat ditentukan. Setelah studi literatur dilakukan, diharapkan peneliti dapat menentukan metode pengujian yang tepat untuk menilai apakah permasalahan yang ada mampu dipecahkan.

3.2 Desain

Setelah studi literatur dilakukan, sebuah perangkat lunak akan didesain sesuai dengan kebutuhan yang ada. Menurut Whitten dan Bentley (2007), Desain perangkat lunak adalah upaya, tahapan, atau aktivitas yang berfokus pada penjelasan detail dari suatu solusi yang akan dikerjakan perangkat lunak. Sementara itu dalam *game*, Rouse (2005) mendeskripsikan desain sebagai proses menentukan bagaimana bentuk dari permainan. Pada tahap desain, detail-detail dari *video game* mulai dari bagaimana *video game* akan dimainkan, bagaimana antarmuka akan didesain, dan bagaimana skenario-skenario yang akan terjadi selama masa permainan.

Pada penelitian ini, perangkat lunak yang akan dikembangkan adalah *video game markerless augmented reality* memancing. Sebuah *video game* memancing berbasis *marker* yang telah dikembangkan pada penelitian sebelumnya akan dikembangkan agar dapat dimainkan tanpa *marker*. Desain *video game* akan disesuaikan dengan fungsi-fungsi yang telah ada sebelumnya. Sebagian besar fungsi *video game* tidak akan berubah namun disesuaikan agar dapat menjalankan sebuah permainan *augmented reality* tanpa menggunakan *marker*.

3.3 Pengembangan

Setelah proses desain dilakukan, sebuah *video game markerless augmented reality* akan dikembangkan. Metode pemecahan yang telah dikemukakan akan diimplementasikan pada *video game augmented reality* untuk kemudian diuji. Pada penelitian ini, metode SLAM akan dikembangkan dari *library* VOID SLAM yang tersedia secara gratis di *game engine* Unity. Hasil implementasi tersebut akan menghasilkan sebuah *video game markerless augmented reality*.

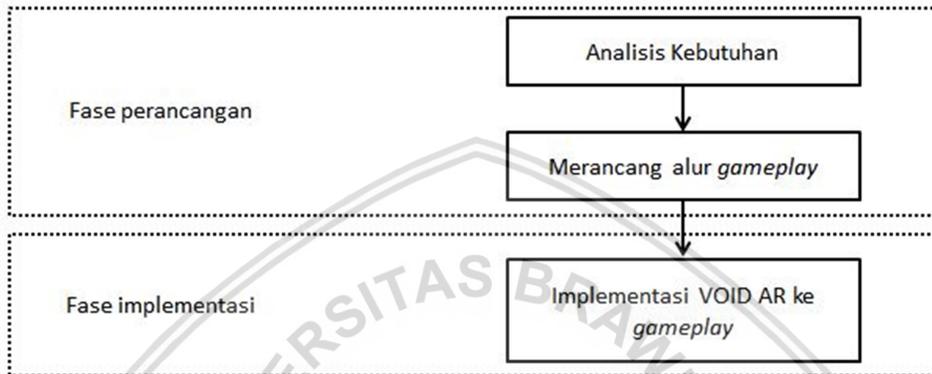
3.4 Analisis

Video game yang telah dirancang kemudian akan diuji untuk menentukan apakah permasalahan yang ada mampu dipecahkan dengan metode tersebut. Hasil pengujian kemudian akan diamati untuk kemudian dianalisis. Pada penelitian ini, analisis dilakukan untuk mengetahui apakah *video game augmented reality* dapat dikembangkan melalui metode SLAM dan bagaimana hasil implementasi SLAM terhadap *video game augmented reality*.

Dari hasil analisis, ditarik kesimpulan apakah metode tersebut mampu memecahkan permasalahan yang ada. Pada penelitian ini, metode SLAM dianggap sukses apabila mampu menghasilkan sebuah *video game augmented reality* yang stabil dan dapat dimainkan dengan nyaman oleh pengguna.

BAB 4 PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan tentang bagaimana proses perancangan pada penelitian berjudul Analisis Pengembangan *Markerless Augmented Reality* Pada *Video Game* Memancing Dengan Pendekatan *Simultaneous Localization and Mapping* (SLAM). Proses perancangan dan implementasi SLAM pada *video game markerless augmented reality* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Proses Perancangan dan Implementasi

4.1 Perancangan

Pada penelitian ini, pendekatan SLAM akan digunakan pada penelitian lain yang berjudul Pengembangan Permainan *Mobile AR Fishing* Berbasis *Marker* menggunakan Metode *Iterative and Rapid Prototyping* yang dikembangkan oleh Luthfansyah (2018). Pendekatan SLAM akan digunakan untuk mengubah *video game fishing AR* berbasis *marker* tersebut menjadi *markerless*.

Sebagian besar perancangan dari penelitian sebelumnya akan dipertahankan. Salah satu perubahan yang perlu dilakukan adalah bagaimana penyesuaian dari *augmented reality* berbasis *marker* ke *markerless* dilakukan. Dalam penelitian sebelumnya, objek virtual akan langsung ditampilkan ketika kamera mendeteksi adanya *marker*. Dalam penelitian ini, perlu ada perintah khusus yang mengatur kapan sebuah objek virtual tampil atau tidak di dunia nyata.

Untuk mengimplementasikan metode SLAM pada *video game markerless augmented reality*, penelitian ini menggunakan SDK yang disediakan oleh VOID AR. *Video game* akan dikembangkan melalui *game engine* Unity.

4.1.1 Analisis Kebutuhan

Untuk merancang *video game markerless augmented reality* dengan SDK yang disediakan VOID AR, ada beberapa kebutuhan yang harus dipenuhi. Terdapat beberapa kebutuhan yang harus dipenuhi untuk menggunakan SDK VOID SLAM. Kebutuhan-kebutuhan yang harus dipenuhi dapat dilihat di Tabel 4.1.

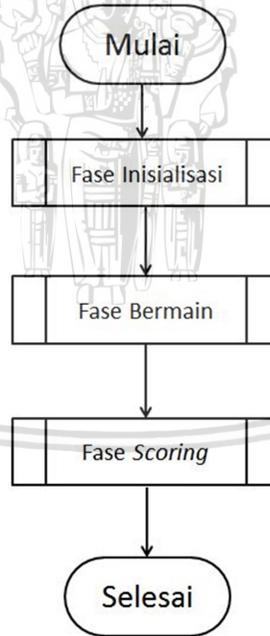
Tabel 4.1 Kebutuhan minimal SDK VOID AR

Komponen	Spesifikasi minimal
Versi Unity	- Unity 5: 5.2 sampai 5.6 (32bit atau 64bit) - Unity 2017: 2017.1 sampai versi terbaru
Sistem Operasi untuk mengembangkan perangkat lunak	- Windows 7 sampai versi terbaru - Mac OS X 10.10 sampai versi terbaru
Sistem Operasi perangkat target	- Windows - Mac OS - iOS - Android

Sumber: Website Void AR

4.1.2 Alur Permainan

Karena permainan yang dikembangkan pada penelitian ini tidak menggunakan *marker*, alur permainan memancing akan sedikit berbeda dengan permainan sebelumnya. Alur permainan setelah sistem SLAM diimplementasikan dapat dilihat di Gambar 4.2.



Gambar 4.2 Alur permainan

1. Fase Inisialisasi

Fase ini tidak ditemukan pada penelitian sebelumnya yang menggunakan *marker*. Pada *markerless augmented reality*, proses inisialisasi awal perlu



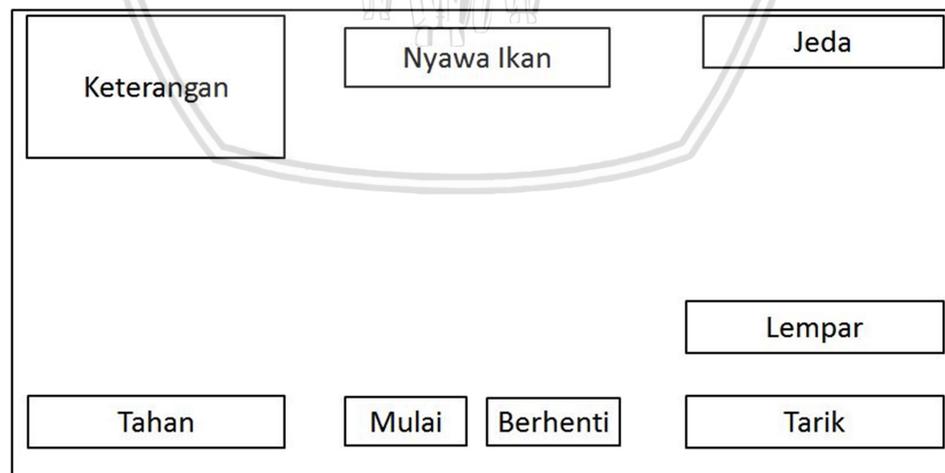
dilakukan untuk menentukan pola-pola awal. Melalui masukan pola awal, objek virtual dapat dimunculkan dengan metode SLAM.

Karena *video game augmented reality* penelitian ini tidak menggunakan *marker* untuk menentukan tempat bermain, pemain perlu terlebih dahulu mencari tempat objek virtual akan ditampilkan. Idealnya, objek virtual akan ditampilkan di atas bidang horizontal (contoh: meja atau lantai). Meskipun demikian, pemain dapat menampilkan objek virtual di tempat yang tidak seharusnya. Resikonya, objek virtual tidak dapat ditampilkan dengan baik dan permainan tidak dapat dilakukan.

Setelah tempat untuk menampilkan objek ditentukan, pemain dapat memasukkan perintah untuk memulai proses ditampilkannya objek virtual. Permainan dapat dimulai ketika objek virtual tampil di layar. Gambar yang terekam kamera akan dikenali polanya dan SLAM dapat dilakukan. Sistem akan menampilkan objek virtual untuk kemudian dimainkan oleh pemain.

Pada penelitian ini, perintah untuk menampilkan objek virtual akan diimplementasikan melalui tombol yang terletak di tengah layar. Perintah tersebut akan berbentuk sebuah tombol yang berlabel "Mulai". Selain tombol "Mulai". Selain tombol mulai, salah satu perintah yang perlu ditambahkan adalah fungsi untuk menghilangkan objek virtual. Perintah ini ditambahkan untuk menghilangkan objek virtual ketika terjadi sesuatu yang tidak diinginkan, seperti objek virtual yang terlihat tidak stabil. Untuk mengimplementasikan perintah tersebut, sebuah tombol "Berhenti" juga akan ditambahkan. Tombol berhenti juga akan diletakkan dibagian tengah bawah layar disebelah tombol "Mulai".

Rancangan antarmuka permainan pada penelitian dapat dilihat pada Gambar 4.3.



Gambar 4.3 Rancangan antarmuka permainan baru

Fungsi dari setiap antarmuka di layar dapat dilihat pada Tabel 4.2

Tabel 4.2 Fungsi dari setiap antarmuka di layar

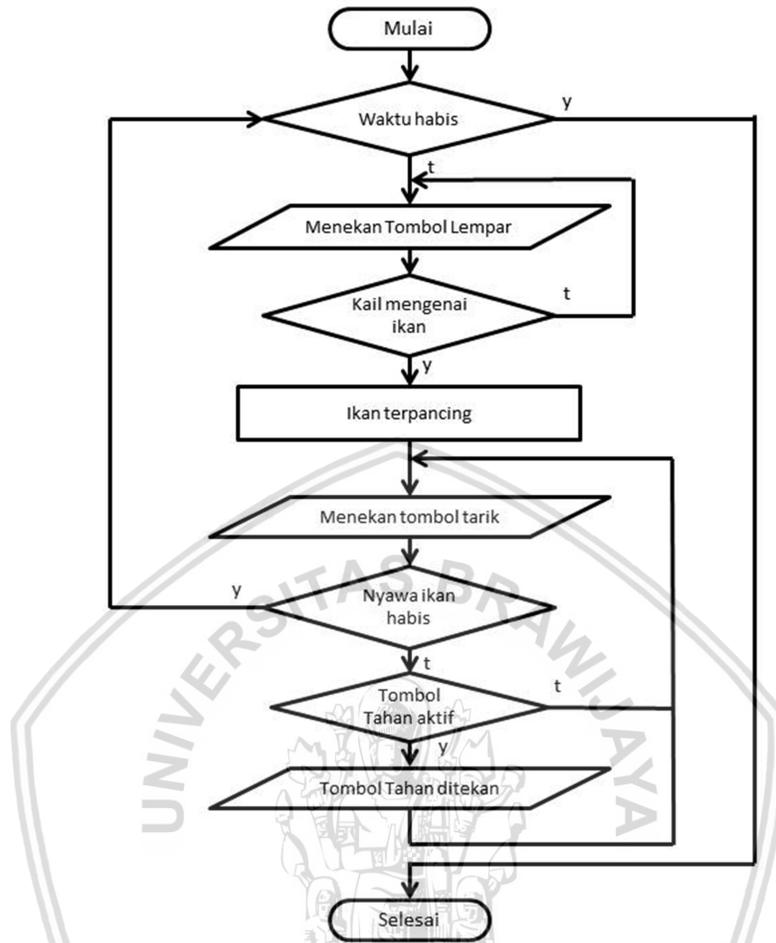
Nama Komponen	Jenis	Keterangan
Keterangan	Layar informasi	Menampilkan informasi tentang skor dan waktu permainan
Nyawa Ikan	Layar Informasi	Menampilkan informasi tentang nyawa ikan. Akan tampil ketika ikan tertangkap
Jeda	Tombol	Menghentikan sementara permainan
Mulai	Tombol	Menampilkan objek virtual agar permainan memancing dapat dimulai
Berhenti	Tombol	Menghilangkan objek virtual agar permainan dapat diulangi
Lempar	Tombol	Melempar kail untuk menangkap ikan
Tahan	Tombol	Menahan pancing. Hanya akan aktif di saat-saat tertentu
Tarik	Tombol	Menarik kail saat ikan tertangkap

2. Fase Permainan

Cara bermain pada permainan memancing yang dimainkan oleh pemain tidak akan berubah dari penelitian sebelumnya. Pemain akan mencoba memancing ikan yang ada di dalam kolam dengan menekan tombol "Lempar". Saat tombol ditekan, *icon* kail akan terlempar ke dalam kolam. Pemain akan terus menekan tombol "Lempar" hingga kail mengenai ikan. Ketika kail mengenai ikan, pemain akan menekan tombol "Tarik" secara berulang hingga poin nyawa ikan habis. Saat menarik ikan, pemain juga diharuskan untuk menekan tombol "Tahan" jika tombol tersebut aktif.

Salah satu perbedaan alur fase permainan pada penelitian ini dibandingkan penelitian sebelumnya adalah adanya implementasi SLAM saat permainan berlangsung. Saat permainan memancing berlangsung, SLAM akan menentukan bagaimana objek virtual akan ditampilkan di dalam layar kamera. Selama permainan dijalankan, kamera akan merekam perubahan citra tiap detiknya. Melalui perbedaan pola tiap citra, pengenalan lingkungan dilakukan agar objek virtual dapat terus muncul di layar.

Alur fase permainan secara jelas dapat dilihat pada Gambar 4.4.



Gambar 4.4 Alur fase permainan

3. Fase Perhitungan Skor

Fase perhitungan skor pada penelitian ini tidak akan berubah dari penelitian sebelumnya. Skor akan dihitung setiap ikan yang ditangkap. Satu ikan akan bernilai 100. Tiap level memiliki jumlah skor minimal yang berbeda-beda dan harus dipenuhi oleh pemain.

4.2 Implementasi

4.2.1 Penentuan Spesifikasi Hardware dan Software

1. Spesifikasi Sistem Perangkat Mobile

Tabel sistem *mobile* untuk uji coba *video game augmented reality markerless* yang dikembangkan pada penelitian ini dapat dilihat pada Tabel 4.3.

Tabel 4.3 Spesifikasi Sistem Perangkat Mobile

Perangkat	Deskripsi
Nama Perangkat	Asus ZenPhone 4 Selfie
Nomor Model	ZD553KL
Processor	Qualcomm® Snapdragon™ 430 Mobile Platform, Octa-core 64-bit
Display	5.5-inci layar IPS
Versi Android	Android™ N dengan ASUS ZenUI
Memori	4,00 GB
Kamera	Kamera depan 10 dan 4 Megapixel, Kamera belakang 16 Megapixel

2. Spesifikasi Sistem Perangkat Pengembangan

Tabel spesifikasi PC yang digunakan untuk mengembangkan *video game augmented reality markerless* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Spesifikasi Sistem Perangkat Pengembangan

Perangkat	Deskripsi
Nama Perangkat	Asus X550I
Sistem Model	X550IU
Sistem operasi	Windows 10 Pro 64-bit
VGA	AMD RADEON RX 460
Processor	AMD FX-9830P RADEON R7
Direct X	12
Memori	8,00 GB
Versi Unity	Unity 5.6.5f1 Personal

4.2.2 Proses Implementasi SLAM

Implementasi SLAM pada *video game* memancing dilakukan untuk mengubah jenis marker pada *video game augmented reality* memancing menjadi *markerless*. ARCamera dan *gameobject* yang telah dibuat sebelumnya akan diimplementasikan ke dalam *scene* game memancing. Beberapa fungsi dari penelitian lama juga akan dihilangkan.

Penyesuaian yang perlu dilakukan adalah mengubah koordinat *pathfinding* ikan. Dalam penelitian sebelumnya, ikan bergerak pada garis lurus antara dua titik yaitu *waypoint1* dan *waypoint2*. Posisi *waypoint* terluar selalu tetap dan ditentukan sejak awal untuk menjaga pergerakan ikan agar tidak keluar dari kolam. Hal ini dapat dilakukan karena penelitian sebelumnya menggunakan

marker. Pada penelitian yang *markerless*, posisi objek virtual tidak terikat pada suatu acuan tertentu yang sehingga posisi *waypoint* terluar ikan dapat berubah-ubah. Logika untuk menghitung posisi *waypoint* terluar dapat dilihat pada *Pseudocode* berikut:

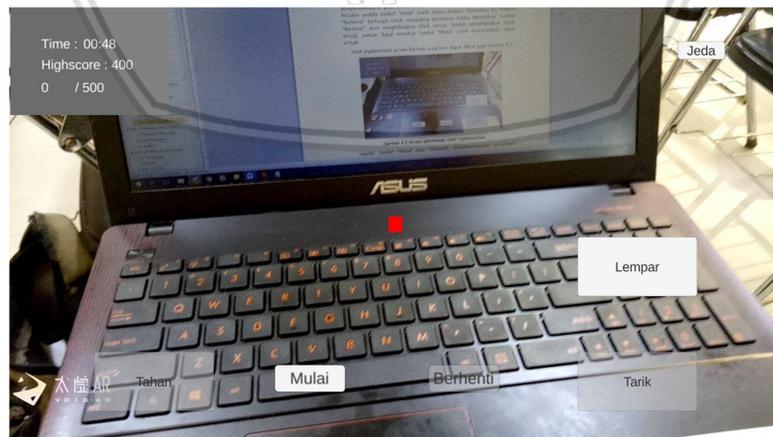
Pseudocode 1: Menghitung koordinat waypoint	
1	IF game start = TRUE
2	THEN
3	x1 = Cube x position * 0.6
4	y1 = Cube y position * 0.6
5	z1 = Cube (2) z position * 0.6
6	
7	x2 = Cube (1) x position * 0.6
8	y2 = Cube y position * 0.6
9	z2 = Cube (3) position * 0.6
10	ENDIF
11	Waypoint1 coordinate = x1, y1, z1
12	Waypoint2 coordinate = x2, y2, z
13	

Pada *pathfinding* yang dimodifikasi, *waypoint* terluar dihitung melalui posisi dinding kolam. Ikan tidak akan berenang menyusuri tempat yang tidak diinginkan karena *waypoint* terluar tidak akan lebih dari koordinat kolam.

Pada *video game* ini, tembok tiap sisi kolam direpresentasikan dengan empat *Gameobject* “Cube”, “Cube (1)”, “Cube (2)”, dan “Cube (3)”. Koordinat tiap sisi x dan z tembok akan dikalikan dengan 0,6, sehingga ikan tidak akan keluar dari tembok samping. Untuk koordinat y, *waypoint* disamakan dengan koordinat y sehingga ikan akan berenang tepat di tengah kolam. Setelah nilai x1, y1, z1, x2, y2, dan z2 ditentukan, nilai tersebut akan digunakan sebagai *waypoint*. x1, y1, dan z1 *waypoint* pertama (*waypoint1*) dan x2, y2, dan z2 *waypoint* kedua (*waypoint2*).

4.2.3 Hasil Implementasi SLAM

Hasil implementasi screen bermain yang baru dapat dilihat pada Gambar 4.5

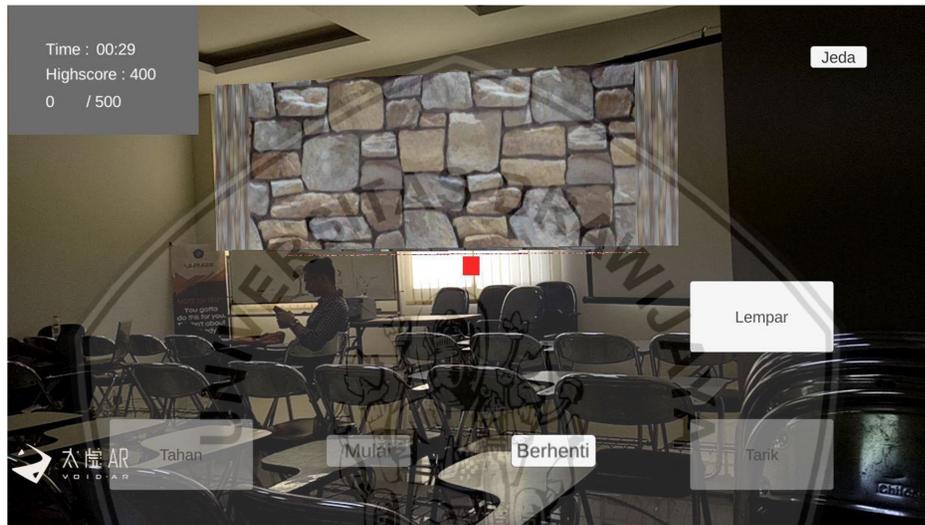


Gambar 4.5 Screen permainan hasil implementasi

Tombol “Mulai” dan “Berhenti” yang telah dirancang sebelumnya akan diimplementasikan pada *screen* awal saat game dimulai. Permainan tidak dapat

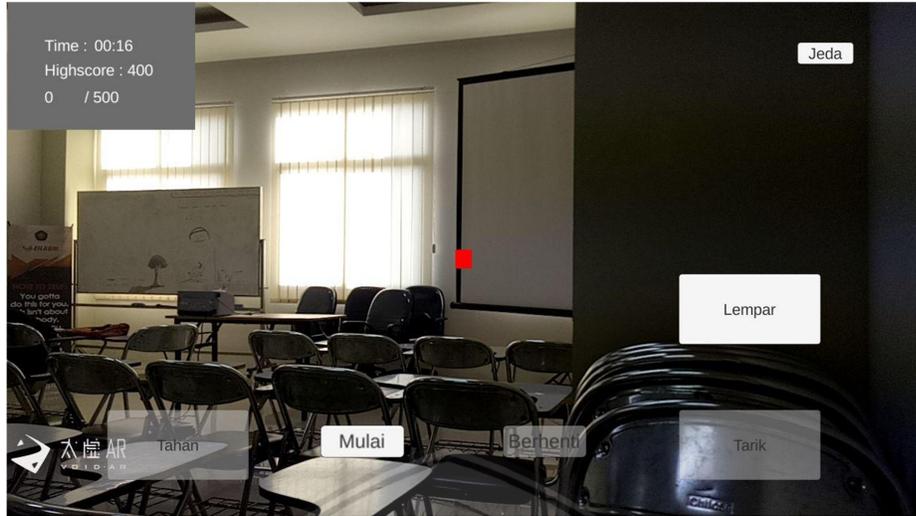
berjalan apabila tombol “Mulai” masih belum ditekan. Sementara itu, tombol “Berhenti” berfungsi untuk mengulang permainan ketika dibutuhkan. Tombol “Berhenti” akan menghilangkan objek virtual. Setelah menghilangkan objek virtual, pemain dapat menekan tombol “Mulai” untuk memunculkan objek virtual.

Setelah tombol “Mulai” dan “Berhenti” diimplementasikan, permainan memancing *markerless augmented reality* dapat dijalankan. Objek virtual telah dapat ditampilkan di atas bidang horizontal dan permainan dapat dimainkan. Pada Gambar 4.6, terlihat bagaimana objek virtual muncul setelah tombol “Mulai” ditekan.



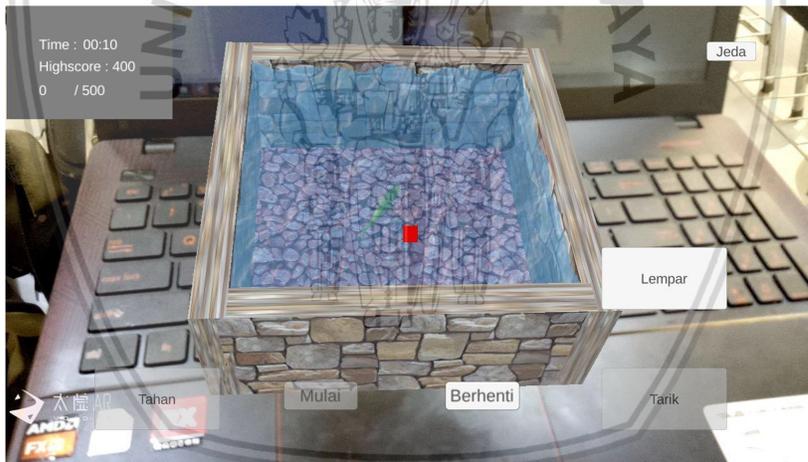
Gambar 4.6 Objek virtual melayang ketika kamera diarahkan ke tempat acak

Dalam Gambar 4.6, Objek virtual melayang di tempat yang tidak seharusnya. Hal tersebut terjadi karena perangkat tidak diarahkan ke bidang horizontal. Untuk memperbaikinya, pemain dapat terlebih dahulu menekan tombol “Berhenti” untuk menghilangkan objek virtual.



Gambar 4.7 Tombol berhenti ditekan

Dari Gambar 4.7 terlihat bahwa ketika tombol “Berhenti” ditekan, objek virtual menghilang dan tombol “Mulai” juga kembali aktif. Kemudian, pemain dapat mengarahkan perangkat *smartphone* pada bidang datar yang diinginkan kemudian memunculkannya kembali dengan menekan tombol “Mulai”.



Gambar 4.8 Objek virtual kembali ditampilkan

Pada Gambar 4.8, terlihat bagaimana objek virtual kembali ditampilkan setelah tombol “Mulai” Ditekan. Tombol “Berhenti” akan kembali aktif dan permainan dapat dilanjutkan kembali.

BAB 5 PENGUJIAN DAN ANALISIS

Bab ini menunjukkan proses dan hasil pengujian dari penelitian berjudul Analisis Pengembangan *Markerless Augmented Reality* Pada *Video Game* Memancing Dengan Pendekatan *Simultaneous Localization and Mapping* (SLAM). Untuk menguji pendekatan tersebut, *video game* akan diuji dimainkan sesuai dengan skenario yang telah ditentukan. Hasil pengujian kemudian akan dianalisis untuk mengetahui apakah metode SLAM mampu memecahkan permasalahan yang ada. Analisis akan berfokus pada aspek SLAM dari *video game markerless augmented reality* memancing. Jenis pengujian yang akan dilakukan pada penelitian ini adalah pengujian *Black Box*.

5.1 Rancangan Pengujian

Pada pengujian *black box*, *video game markerless augmented reality* memancing akan diuji tanpa melihat *source code* yang ada. Pengujian ini akan dilakukan dengan memainkan dan mengamati bagaimana *video game markerless augmented reality* memancing berjalan. Pengujian akan dinyatakan berhasil apabila objek virtual mampu ditampilkan secara stabil dan semua fungsi dalam *video game augmented reality* memancing dapat dijalankan.

Lingkungan tempat pengujian ini adalah ruang C1.7 Fakultas Ilmu Komputer Universitas Brawijaya. Objek virtual akan ditampilkan ke sebuah *keyboard* komputer yang diletakkan di atas meja. Kondisi ruangan dibuat terang agar kamera mampu menangkap lingkungan dengan jelas. Kamera akan ditempatkan sekitar 30 sentimeter dari target pengujian dengan sudut kemiringan sekitar 30 derajat ke arah bawah. Foto lingkungan pengujian dapat dilihat pada Gambar 5.1.



Gambar 5.1 Kondisi lingkungan pengujian SLAM

Ada dua aspek utama yang diuji yaitu aspek *sequence* permainan dan aspek SLAM.

5.1.1 Pengujian Aspek *Sequence* Permainan

Pada bagian ini, aspek yang diuji pada bagian ini adalah urutan *sequence* menangkap ikan. Pengujian ini dilakukan untuk mengetahui apakah *video game augmented reality* memancing dapat dimainkan setelah fungsi *markerless* diimplementasikan. Kasus uji pada pengujian ini dapat dilihat pada Tabel 5.1.

Tabel 5.1 Kasus uji pada pengujian aspek *sequence* permainan

No	Kasus Uji	Hasil yang diharapkan
1	Tombol “Mulai” ditekan	<ul style="list-style-type: none"> Objek virtual dapat muncul Tombol “Berhenti” aktif Tombol “Mulai” tidak aktif
2	Tombol “Berhenti” ditekan	<ul style="list-style-type: none"> Objek virtual hilang Tombol “Berhenti” tidak aktif Tombol “Mulai” aktif
3	Tombol “Lempar” ditekan	<ul style="list-style-type: none"> Kail terlempar ke arah ikan
4	Ikan tertangkap	<ul style="list-style-type: none"> <i>Smartphone</i> bergetar Tombol “Tahan” dan Tombol “Tarik” aktif secara bergantian Informasi Nyawa ikan muncul
5	Tombol “Tarik” ditekan	<ul style="list-style-type: none"> Nyawa ikan berkurang
6	Tombol “Tahan” ditekan	<ul style="list-style-type: none"> Tombol “Tarik” aktif
7	Nyawa ikan habis	<ul style="list-style-type: none"> Ikan tertarik ke layar Tombol “Tarik” tidak aktif Tombol “Lempar” aktif Informasi nyawa ikan menghilang Skor bertambah

5.1.2 Pengujian Aspek SLAM

Aspek yang diuji pada bagian ini adalah bagaimana SLAM mampu menampilkan objek virtual di dunia nyata di suatu lingkungan dengan stabil. Pada pengujian ini, kamera *smartphone* akan digerakkan sesuai skenario yang telah ditentukan di dalam lingkungan. Terdapat tiga skenario utama pada pengujian ini, yaitu skenario saat kamera digeser, kamera diputar, dan kamera bergerak mengelilingi objek virtual.

1. Kamera bergeser

Pada skenario ini, kamera akan digeser ke atas, bawah, kiri atau kanan dengan jumlah tertentu yang telah ditentukan oleh kasus uji. Pertama-tama, kamera akan ditempatkan di suatu titik menghadap target. Kemudian jarak pergeseran dari titik tersebut akan diukur menggunakan penggaris. Setelah jarak diketahui, kamera digeser ke tempat yang ditentukan. Terakhir kamera akan digeser kembali ke titik awal. Penjelasan lebih jelas dapat dilihat pada Gambar 5.2.



Gambar 5.2 Ilustrasi skenario Kamera bergeser

2. Kamera berputar

Pada skenario ini, kamera akan diputar ke atas, bawah, kiri atau kanan dengan sudut tertentu yang telah ditentukan oleh kasus uji. Sama seperti pengujian sebelumnya, kamera akan ditempatkan di titik tertentu menghadap target. Kemudian jarak perputaran akan diukur dengan busur derajat. Setelah sudut perputaran diketahui, kamera akan diputar ke arah yang ditentukan. Terakhir kamera akan diputar kembali hingga mengarah ke tempat awal. Penjelasan lebih jelas dapat dilihat pada Gambar 5.3.



Gambar 5.3 Ilustrasi skenario kamera diputar

3. Kamera bergeser

Pada skenario ini, kamera akan bergerak memutar objek virtual dengan kamera yang terus mengarah ke target pengujian. Kamera akan terus bergerak hingga membentuk lingkaran penuh hingga kembali ke tempat awal. Penjelasan lebih jelas dapat dilihat pada Gambar 5.4.



Gambar 5.4 Ilustrasi skenario kamera memutari objek

Hasil yang diharapkan dari ketiga skenario uji tersebut adalah objek virtual akan dapat terus stabil meski kamera melakukan gerakan yang berbeda-beda. Kasus uji dari skenario-skenario tersebut dapat dilihat pada Tabel 5.2

Tabel 5.2 Kasus uji pada pengujian aspek SLAM

No	Skenario	Kasus Uji
1	Kamera bergeser	<ul style="list-style-type: none"> • 20 sentimeter ke samping kiri • 20 sentimeter ke samping kanan • 20 sentimeter ke atas • 20 sentimeter ke bawah • 40 sentimeter ke samping kiri • 40 sentimeter ke samping kanan • 40 sentimeter ke atas • 40 sentimeter ke bawah
2	Kamera berputar	<ul style="list-style-type: none"> • 45 derajat ke samping kiri • 45 derajat ke samping kanan • 45 derajat ke atas • 45 derajat ke bawah • 90 derajat ke samping kiri • 90 derajat ke samping kanan • 90 derajat ke atas • 90 derajat ke bawah • 180 derajat
3	Kamera bergerak melingkari objek virtual	<ul style="list-style-type: none"> • Pengguna bergerak memutari objek virtual dengan kamera terus mengarah ke target hingga kembali ke tempatnya semula

5.2 Hasil Pengujian

Pada bagian ini, hasil pengujian dari *video game markerless augmented reality* memancing akan dijelaskan. Pengujian akan dilakukan sesuai dengan kasus uji yang telah dijelaskan di bagian sebelumnya.

5.2.1 Pengujian Aspek *Sequence Game*

Bagian ini akan menjelaskan hasil pengujian pada aspek *sequence game*. Hasil pengujian dapat dilihat pada Tabel 5.3.

Tabel 5.3 Hasil pengujian aspek *sequence* permainan

No	Kasus Uji	Hasil yang diharapkan	Hasil
1	Tombol "Mulai" ditekan	Objek virtual dapat muncul	Valid
		Tombol "Berhenti" aktif	Valid
		Tombol "Mulai" tidak aktif	Valid
2	Tombol "Berhenti" ditekan	Objek virtual hilang	Valid
		Tombol "Berhenti" tidak aktif	Valid
		Tombol "Mulai" aktif	Valid
3	Tombol "Lempar" ditekan	Kail terlempar ke arah ikan	Valid
4	Ikan tertangkap	<i>Smartphone</i> bergetar	Valid
		Tombol "Tahan" dan Tombol "Tarik" aktif secara bergantian	Valid
		Informasi Nyawa ikan muncul	Valid
5	Tombol "Tarik" ditekan	Nyawa ikan berkurang	Valid
6	Tombol "Tahan" ditekan	Tombol "Tarik" aktif	Valid
7	Nyawa ikan habis	Ikan tertarik ke layar	Valid
		Tombol "Tarik" tidak aktif	Valid
		Tombol "Lempar" aktif	Valid
		Informasi nyawa ikan menghilang	Valid
		Skor bertambah	Valid

5.2.2 Pengujian Spek SLAM

Bagian ini akan menjelaskan hasil pengujian pada aspek SLAM. Hasil pengujian dapat dilihat pada Tabel 5.4.

Tabel 5.4 Hasil Pengujian SLAM

No	Kasus Uji	Nilai	Hasil
1	Kamera digeser	20 sentimeter ke kiri	Objek virtual stabil
		20 sentimeter ke kanan	Objek virtual stabil
		20 sentimeter ke atas	Objek virtual stabil
		20 sentimeter ke bawah	Objek virtual berpindah tempat ketika hendak keluar layar namun kembali stabil ketika kamera kembali ke tempat semula
		40 sentimeter ke kiri	Objek virtual sedikit bergetar ketika hendak keluar layar, tapi tetap stabil
		40 sentimeter ke kanan	Objek virtual stabil
		40 sentimeter ke atas	Objek virtual stabil
		40 sentimeter ke bawah	Objek virtual berpindah tempat ketika hendak keluar layar namun kembali stabil ketika kamera kembali ke tempat semula
2	Kamera berputar	45 derajat ke kiri	Objek virtual stabil
		45 derajat ke kanan	Objek virtual stabil
		45 derajat ke atas	Objek virtual stabil
		45 derajat ke bawah	Objek virtual berpindah tempat ketika hendak keluar layar namun kembali stabil ketika kamera kembali ke tempat semula
		90 derajat ke kiri	Objek virtual berpindah tempat ketika hendak memasuki layar namun kembali stabil ketika kamera kembali ke tempat semula
		90 derajat ke kanan	Objek virtual stabil

Tabel 5.4 Hasil Pengujian SLAM (Lanjutan)

		90 derajat ke atas	Objek virtual stabil
		90 derajat ke bawah	Objek virtual berpindah tempat ketika hendak keluar layar namun kembali stabil ketika kamera kembali ke tempat semula
		180 derajat	Objek virtual menghilang pada saat hendak memasuki, namun kembali muncul ketika kamera tepat kembali ke tempat semula
3	Kamera melingkari objek virtual	Berputar mengelilingi objek virtual	Objek virtual stabil

5.3 Analisis

5.3.1 Pengujian Aspek *Sequence Game*

Pada aspek *sequence game*, semua fungsionalitas tombol dari *game* memancing dapat berjalan dengan lancar. *Video game* dapat dimainkan dengan lancar dari awal hingga akhir permainan sesuai dengan aturan yang telah ditetapkan.

Tombol “Mulai” dan “Berhenti” juga dapat berfungsi dengan baik. Selama objek virtual muncul, tombol berhenti akan aktif. Ketika objek virtual menjadi tidak stabil karena suatu hal, tombol “Berhenti” mampu menghilangkan objek virtual. Setelah tombol “Berhenti” ditekan, tombol “Mulai” akan kembali aktif sehingga objek virtual dapat kembali ditampilkan.

5.3.2 Pengujian Aspek SLAM

Melalui pengujian aspek SLAM, secara umum terlihat bagaimana objek virtual mampu muncul dengan baik. Aplikasi *video game* memancing *markerless* dapat menampilkan objek virtual di bidang horizontal yang telah ditentukan.

Masalah muncul dalam beberapa kondisi ketika objek virtual akan memasuki atau keluar dari layar. Objek virtual tampak tidak stabil dalam beberapa skenario. Pada saat kamera digeser ke bawah, objek virtual terlihat ikut terseret oleh layar sebelum akhirnya hilang. Untuk beberapa skenario seperti ketika kamera diputar 180 derajat, objek virtual terlihat menghilang ketika objek seharusnya memasuki layar.

Meskipun begitu, objek virtual selalu kembali terlihat stabil ketika kamera kembali ke tempat semula. Hal ini disebabkan karena aplikasi kembali mengenali pola lingkungan awal. Ketika kamera bergerak menjauhi lingkungan awal, pola-

pola baru akan masuk dan aplikasi mencoba menyesuaikan pola-pola baru tersebut dengan pola-pola lama. Hal inilah yang menyebabkan objek virtual menjadi tidak stabil ketika keluar dari layar kamera.

Terakhir, objek virtual terlihat tetap stabil ketika kamera yang berpindah tempat tetap mengarah kepada objek virtual. Hal ini disebabkan pola-pola yang terbaca tidak berbeda jauh dari pola-pola awal yang terbaca oleh aplikasi. Dari sini dapat diketahui bahwa objek virtual akan cenderung lebih stabil ketika lingkungan yang terekam dalam kamera cenderung tetap atau tidak berubah banyak.



BAB 6 KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, kesimpulan yang dapat ditarik pada penelitian analisis pengembangan *video game markerless augmented reality* memancing dengan pendekatan *Simultaneous Localization and Mapping* (SLAM) ini adalah sebagai berikut:

1. *Video game markerless augmented reality* memancing dapat dikembangkan dengan pendekatan SLAM menggunakan *package* VOID AR pada *game engine* Unity. Objek virtual dapat muncul tanpa adanya marker khusus dengan membaca pola-pola di dalam layar. Pola-pola yang terbaca di layar dapat berupa garis, titik, sudut, atau kombinasi ketiganya.
2. Semua fungsionalitas *video game* memancing dapat berjalan dengan lancar dan dapat dimainkan dengan lancar dari awal hingga akhir permainan sesuai dengan aturan yang telah ditetapkan. Objek virtual terlihat paling stabil ketika lingkungan yang dikenali tidak banyak berubah ketika dibandingkan dengan lingkungan awal pada proses inisialisasi. Hal ini disebabkan karena pola yang baru masuk tidak berbeda jauh dengan pola-pola awal yang didapat saat proses inisialisasi. Ketika arah perangkat diubah ke lingkungan baru, pola-pola baru akan masuk ke layar kamera sehingga aplikasi akan kesulitan mengenali lingkungannya.

6.2 Saran

Saran untuk pengembangan lebih lanjut pada penelitian analisis pengembangan *video game markerless augmented reality* memancing dengan pendekatan *Simultaneous Localization and Mapping* (SLAM) ini perlu dilakukan analisis terhadap pola lingkungan yang terbaca di layar kamera. Hal ini perlu dilakukan agar pengaruh pola lingkungan yang terbaca di layar terhadap objek virtual dapat diamati dan ditentukan dengan lebih jelas.

<http://www.mobilelifecentre.org/sites/default/files/BA_Ziegler-1_final.pdf> [Diakses 21 Maret 2018].

