

Pembuatan *Game Tower Defense* Menggunakan *Augmented Reality* Dengan *Unity Engine* dan *Vuforia* pada *Android*

Reynaldo Yoda Wijaya¹, Kristo Radion Purba²
Program Studi Teknik Informatika,
Fakultas Teknologi Industri, UK Petra
Jln. Swialankerto 121 – 131 Surabaya 60236
Telp. (031)-2983455, Fax. (031)-8417658
Reynaldo.yoda.wijaya@gmail.com¹, kristo@petra.ac.id²

ABSTRAK

Seiring kemajuan teknologi banyak aplikasi sudah memanfaatkan teknologi yang disebut *Augmented Reality*, *Augmented Reality (AR)* adalah teknologi yang menggabungkan benda maya dua dimensi ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata. *Augmented Reality* sendiri mengizinkan penggunaannya untuk berinteraksi secara *real-time* dengan sistem. *AR* sendiri banyak digunakan dalam berbagai hal contohnya pada bidang *entertainment*, bidang iklan dan juga digunakan dalam dunia permainan (*game*). Namun dari banyaknya *game* yang beredar, masih sedikit *game* yang menggunakan *Augmented Reality* sebagai kunci utama dalam permainan.

Game Tower defense memiliki tujuan utama untuk melindungi suatu basis agar tidak dicapai oleh musuh yang berjalan pada jalur yang sudah ditentukan dengan cara membangun penempatan *tower*. Dengan menggunakan *augmented reality*, user dapat memainkan *game* melalui berbagai sudut pandang sesuai dengan pergerakan dari kamera pada perangkat android yang mengarah pada *marker* sehingga akan menambah interaksi antara pemain dengan *game*.

Melalui pengujian dapat dibuktikan bahwa penggunaan *Augmented Reality Vuforia* dapat digunakan untuk pembangunan *game* karena dapat memberikan hasil yang cukup baik. Penerapan *Augmented Reality* pada *game* ini membuat permainan menjadi lebih menarik sehingga pemain tidak akan merasa bosan walau sudah memainkan *game* berkali-kali.

Kata kunci: *Augmented Reality*, *mobile game*, *tower defense*, *Android*

ABSTRACT

Along with technological advances, many applications are already using technology called *Augmented Reality*, a technology that combines two-dimensional or three-dimensional virtual objects into a real three-dimensional environment and projects those virtual objects in real-time. *Augmented Reality* itself allows users to interact in real-time with the system. *Augmented Reality* is widely used in various things, for example in the field of entertainment, advertising and also used in the games. But of the many games in circulation, there are still few games that use *Augmented Reality* as the main key in the game.

Tower defense games have the primary purpose of protecting a base from being reached by enemies running on predetermined

paths by constructing tower placements. By using augmented reality, user can play the game through various viewpoints in accordance with the movement of the camera on the android device that leads to the marker so that it will increase the interaction between players with the game.

Through testing can be proven that the use of Vuforia's Augmented Reality can be used for game development because it can provide good results. Application of Augmented Reality in this game makes the game more interesting so players will not feel bored even though it has played the game many times.

Keywords: *Augmented Reality*, *mobile game*, *tower defense*, *Android*

1. PENDAHULUAN

Game tower defense yang ada saat ini kebanyakan masih menggunakan sistem *static-line*, yang artinya jalur yang di tempuh oleh karakter *Non Playable Character (NPC)* masih monoton dan tidak dinamis.

Dalam *gameplay*-nya karakter *NPC* tidak bisa menyerang *tower*, hanya bisa berjalan pada jalurnya saja, dan tidak dapat berpindah jalur pada saat berhadapan dengan *tower*. Dalam *gameplay*-nya juga pemain hanya terpaku pada strategi penempatan *tower* saja dan tidak ada bentuk *gameplay* lainnya yang membuat *game* lebih variatif.

Teknologi *augmented reality* memungkinkan pemain untuk dapat memainkan *game* melalui berbagai sudut pandang sesuai dengan pergerakan dari kamera pada perangkat android yang mengarah pada *marker* sehingga akan menambah interaksi antara pemain dengan *game*.

2. LANDASAN TEORI

2.1 Definisi *Game*

Game sudah banyak dikenal oleh manusia sejak dahulu, hampir semua orang pernah memainkan *game* dikarenakan saat ini *game* merupakan kebutuhan bagi manusia, berikut adalah beberapa definisi *game* dari beberapa ahli:

- a. Menurut Chris Crawford, seorang designer *game* komputer mengemukakan bahwa *game*, pada intinya adalah sebuah interaktif, aktivitas yang berpusat pada sebuah pencapaian, mempunyai pelaku aktif, mempunyai lawan.

- b. Menurut Roger Caillos seorang sosiolog Perancis, dalam bukunya yang berjudul *Les jeux et les hommes* menyatakan *game* adalah aktivitas yang mencakup karakteristik berikut: *fun* (bebas bermain adalah pilihan bukan kewajiban), *separate* (terpisah), *uncertain* (tidak menentu), *non-productive* (tidak produktif), *governed by rules* (ada aturan), *fictitious* (pura-pura).

2.2 Klasifikasi Game

Berdasarkan Jenis *Platform* yang Digunakan

Berdasarkan *platform* yang digunakan *game* dibagi menjadi beberapa jenis, yaitu:

1. *Arcade games*, *game* yang dijalankan pada mesin dengan input dan output audio visual yang telah terintegrasi dan tersedia ditempat - tempat umum dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa “masuk” dan “menikmati” permainan, seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan setir mobil.
2. *PC games*, yaitu *video game* yang dimainkan menggunakan *personal computer*, memiliki kelebihan yaitu tampilan antarmuka yang baik.
3. *Console games*, yaitu *video games* yang dimainkan menggunakan *console* tertentu seperti, *Playstation 3*, *XBOX 360*, dan *Nintendo Wii*.
4. *Handheld games*, yaitu yang dimainkan di *console* khusus *video game* yang dapat dibawa kemana-mana, contohnya *Nintendo DS* dan *Sony PSP*.
5. *Mobile games*, yaitu *game* yang dapat dimainkan atau khusus untuk *mobile phone*. [2]

Berdasarkan *genre game* dibagi menjadi beberapa jenis, yaitu:

1. *Shooting, video game* jenis ini memerlukan kecepatan *reflex*, koordinasi mata-tangan, juga *timing*, inti dari *game* jenis ini adalah tembak-menembak.
2. *Fighting*, jenis ini memerlukan kecepatan refleksi dan koordinasi mata-tangan, tetapi inti dari *game* ini adalah penguasaan jurus (hafal caranya dan lancar mengeksekusinya), pengenalan karakter dan *timing* sangatlah penting, *combo*-pun menjadi esensial untuk mengalahkan lawan secepat mungkin.
3. *Adventure, video game* jenis petualangan menekankan pada jalan cerita dan kemampuan berpikir pemain dalam menganalisa tempat secara visual, memecahkan teka-teki maupun menyimpulkan rangkaian peristiwa dan percakapan karakter hingga penggunaan benda-benda tepat pada tempat yang tepat.
4. Simulasi, Kontruksi dan manajemen, *video game* jenis ini seringkali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dan memperhatikan dengan detail berbagai faktor. Contohnya seperti *Sim City*, *The Sims*, *Tamagotchi*.
5. *Role Playing Game (RPG), video game* jenis ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh atau peran perwakilan pemain di dalam permainan, yang biasanya adalah tokoh utamanya, dimana seiring kita memainkannya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan

pemain dalam berbagai parameter yang biasanya ditentukan dengan naiknya *level*.

6. *Puzzle, video game* jenis ini sesuai namanya berintikan mengenai pemecahan teka-teki, baik itu menyusun balok, menyamakan warna bola, memecahkan perhitungan matematika, melewati labirin, sampai mendorong kotak masuk ke tempat yang seharusnya.
7. *Sport*, permainan *sport* di PC atau konsol anda. Biasanya permainannya diusahakan serealistik mungkin walau kadang ada yang menambah unsur fiksi. Contoh *game*: Seri *Winning Eleven*, seri *NBA*, seri *FIFA*, *John Madden NFL*, *Lakers vs Celtics*, *Tony hawk pro skater*. [2]

2.3 Tower Defense

Game Tower Defense adalah *game* bertipe *Real Time Strategy* yang berfokus pada pengaturan sumber daya yang dimiliki pemain dan penempatan *tower*. Tujuan dari *game* adalah untuk mencegah musuh memasuki suatu tempat yang dilindungi oleh pemain dengan cara menyerang dan menahan musuh untuk memasuki tempat yang dilindungi tersebut. Cara menahan musuh bervariasi mulai dari menembak, memasang jebakan dan memperlambat pergerakan musuh. Dalam bentuk sederhana, sebuah *game Tower Defense* terdiri dari pemain yang bertugas membeli dan mengorganisir *tower* yang akan menembak musuh yang menyerang pertahanan pemain.

Setiap kali pemain berhasil mengalahkan musuh, pemain akan mendapatkan poin atau uang. Poin tersebut dapat dipakai untuk menambah objek tertentu untuk melawan musuh dengan level yang lebih tinggi. Objek *game* tersebut dapat berupa senjata baru atau kekuatan yang lebih tinggi dan dapat digunakan seiring dengan berlangsungnya permainan. Contoh dari *game tower defense* yaitu: *Kingdom Rush*, *Gem Craft*, *Plant VS Zombies*, *Tower Madness*, *Sanctum*. [5]

2.4 Augmented Reality (AR)

Augmented reality merupakan teknologi yang menggabungkan benda maya dua dimensi ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut secara *real-time*. *Augmented reality* menampilkan informasi berupa label maupun objek virtual yang hanya dapat dilihat dengan kamera *handphone* maupun dengan komputer. Sistem dalam *augmented reality* bekerja dengan menganalisa secara *real-time* objek yang ditangkap dalam kamera. [4]

Augmented Reality memperbolehkan pengguna melihat objek maya tiga dimensi yang diproyeksikan terhadap dunia nyata. (*Emerging Technologies of Augmented Reality: Interface and Design*)

Ronald T. Azuma mendefinisikan *augmented reality* sebagai penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata, dan terdapat integrasi antar benda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata. [4]

2.5 Vuforia SDK

Vuforia adalah *Augmented Reality Software Development Kit (SDK)* untuk perangkat *mobile* yang memungkinkan pembuatan aplikasi *augmented reality*. Menggunakan teknologi *Computer Vision* untuk mengenali dan melacak gambar (*Target Image*) dan objek 3D sederhana, seperti kotak dan kaleng secara *real-time*.

Vuforia SDK memerlukan beberapa komponen penting agar dapat bekerja dengan baik. Komponen tersebut antara lain:

1. Kamera
Kamera dibutuhkan untuk memastikan bahwa setiap frame ditangkap dan diteruskan secara efisien ke tracker. Para developer hanya tinggal memberi tahu kamera kapan mereka mulai menangkap dan berhenti.
2. Image Converter
Mengkonversi format kamera (misalnya YUV12) kedalam format yang dapat dideteksi oleh OpenGL (misalnya RGB565) dan untuk *tracking* (misalnya *luminance*).
3. Tracker
Mengandung algoritma computer vision yang dapat mendeteksi dan melacak objek dunia nyata yang ada pada video kamera.
4. Video Background Renderer
Me-render gambar dari kamera yang tersimpan di dalam *state object*. Performa dari *video background renderer* sangat bergantung pada *device* yang digunakan.
5. Target Resources
Dibuat menggunakan *on-line Target Management System*. *Assets* yang diunduh berisi sebuah konfigurasi xml yang memungkinkan *developer* untuk mengkonfigurasi beberapa fitur dalam *trackable* dan *binary file* yang berisi *database trackable*. [4]

2.6 Unity Engine

Unity *game engine* merupakan sebuah *game engine* dan *editor*, yang memungkinkan pengembang membuat *object*, meng-import *external asset* dan menggabungkan semuanya dengan menggunakan kode secara efisien dan cepat.

Editor pada Unity dibuat dengan *user interface* yang sederhana. Unity dapat beroperasi pada Mac OS dan Windows dan dapat menghasilkan *game* untuk Mac, Windows, Wii, iPhone, iPad dan Android. Scripting di dalam Unity dapat menggunakan tiga bahasa pemrograman, yaitu JavaScript, C#, dan Boo. [4]

2.7 Components

Components adalah class dan fungsi yang berfungsi untuk memberikan karakteristik pada *game objects*, *component* dapat berupa *script*, *collider*, *particle system*, *shader*, dan lain-lain.

2.7.1 NavMeshAgent

Komponen NavMeshAgent membantu karakter untuk saling menghindari saat bergerak menuju tujuan mereka. NavMesh Agent menerima data pada dunia *digame* dan menggunakannya untuk menghindari satu sama lain serta menghindari rintangan yang bergerak lainnya. *Pathfinding* dan penalaran spesial ditangani dengan menggunakan API *scripting* dari NavMesh Agent.

2.7.2 Navmesh Obstacle

Komponen NavMesh Obstacle digunakan untuk menggambarkan hambatan yang harus dihindari oleh NavMesh Agents saat menavigasi dunia. Bila objek rintangannya tidak bergerak, NavMesh akan menandai posisi objek ini dan membuat lubang, kemudian NavMesh Agent mengubah jalur mereka untuk mengarahkannya, atau menemukan rute yang berbeda jika rintangan tersebut menyebabkan jalur tersebut benar-benar diblokir. [3]

2.7.3 Collision Detection

Collision detection adalah proses pendeteksian tabrakan antara dua *object*. Dalam *game* tabrakan tidak hanya terjadi antara dua objek, tetapi dapat terjadi juga antara satu objek dengan banyak atau banyak objek dengan banyak objek.

Collision detection juga berguna untuk menentukan posisi dari satu objek dengan objek yang lain sehingga tidak ada objek yang saling menembus.

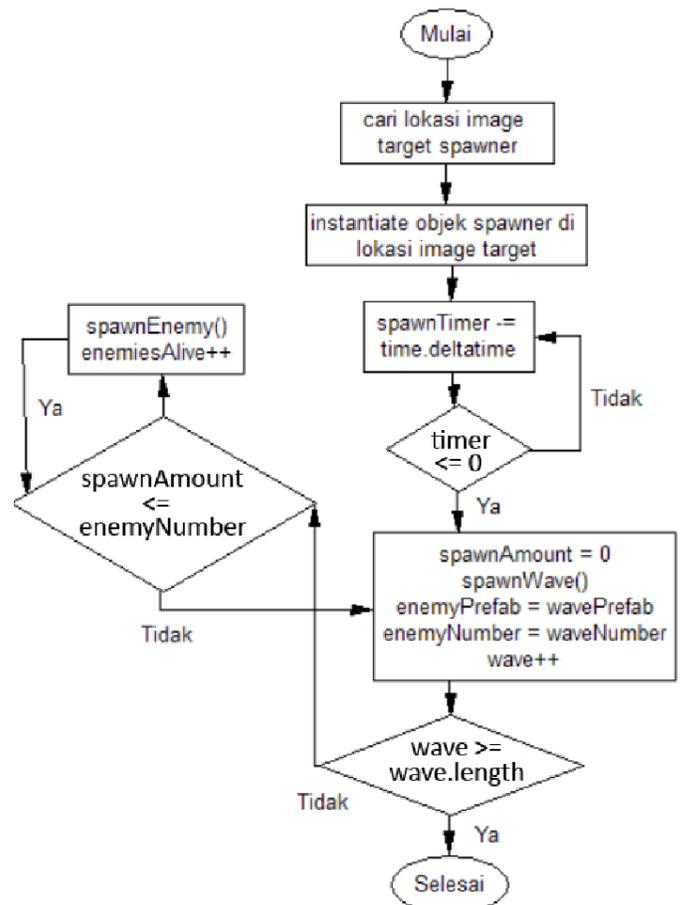
2.7.4 Rigidbody

Rigidbody adalah sebuah fungsi di Unity yang berfungsi agar menambahkan efek gravitasi pada sebuah *object*, menentukan berat suatu *object*, dan memberi efek kinematik pada objek. [1]

3. RANCANGAN SISTEM

3.1 Sistem Spawner

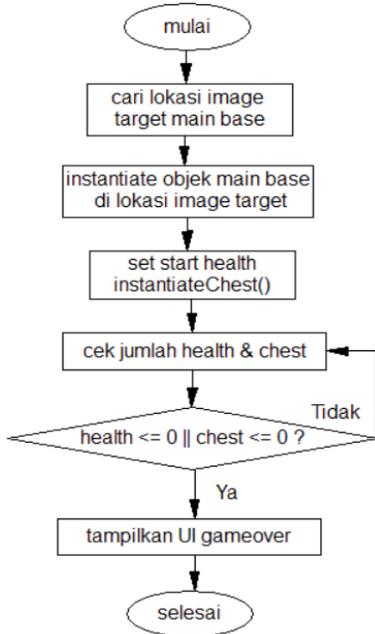
Setelah *marker spawner* di-*scan*, sistem mencari lokasi dan memunculkan objek *spawner*, men-*set array* musuh dan menghitung mundur. Jika *timer* ≤ 0 *spawner* akan *instantiate* karakter musuh pada lokasi *spawner* dan me-*reset* nilai *timer*. Algoritma sistem dapat dilihat pada gambar 1.



Gambar 1. Flowchart Sistem Spawner

3.2 Sistem Main Base

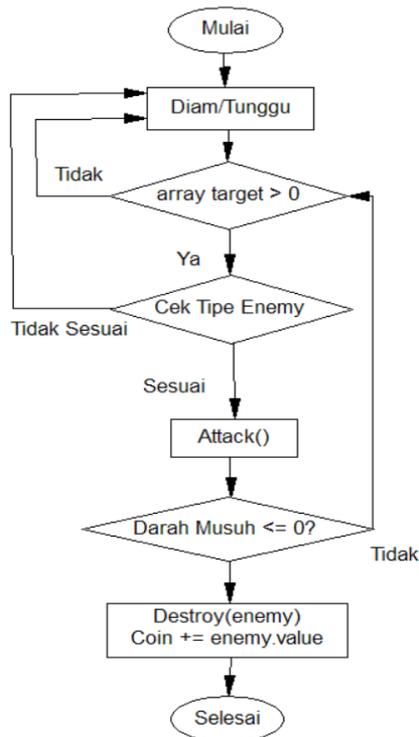
Main base adalah tujuan akhir enemy dan kunci dari jalannya permainan, main base akan memunculkan sejumlah harta disekitarnya dan harus dijaga dari jangkauan tipe musuh tertentu. Algoritma sistem main base dapat dilihat pada gambar 2.



Gambar 2. Flowchart Sistem Main Base

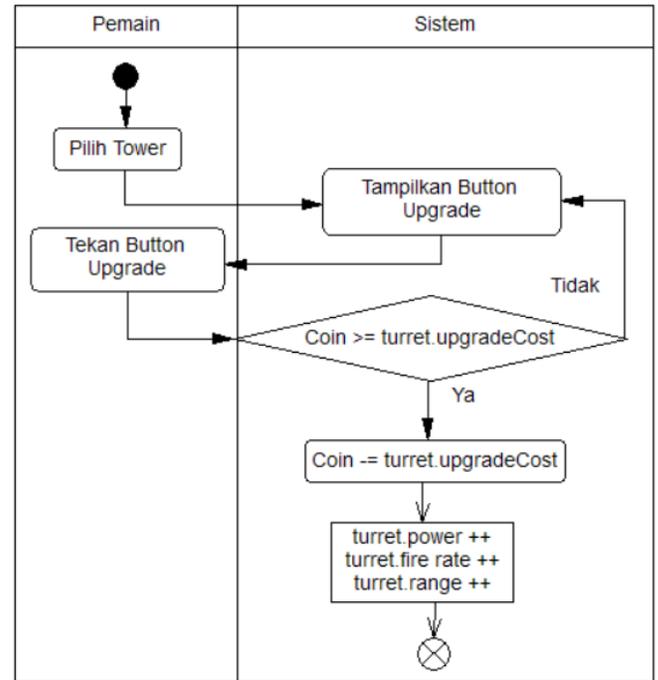
3.3 Sistem Turret

1. Sistem Serang Turret pada game AR Tower Defense dapat dilihat pada gambar 3.



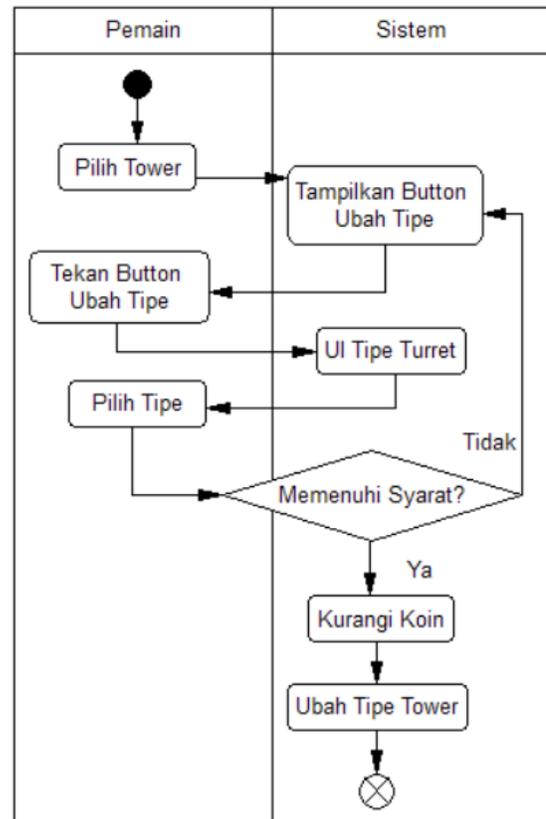
Gambar 3. Flowchart Sistem Serang Tower

2. Alur Upgrade Tower pada game AR Tower Defense dapat dilihat pada gambar 4.



Gambar 4. Activity Diagram Upgrade Tower

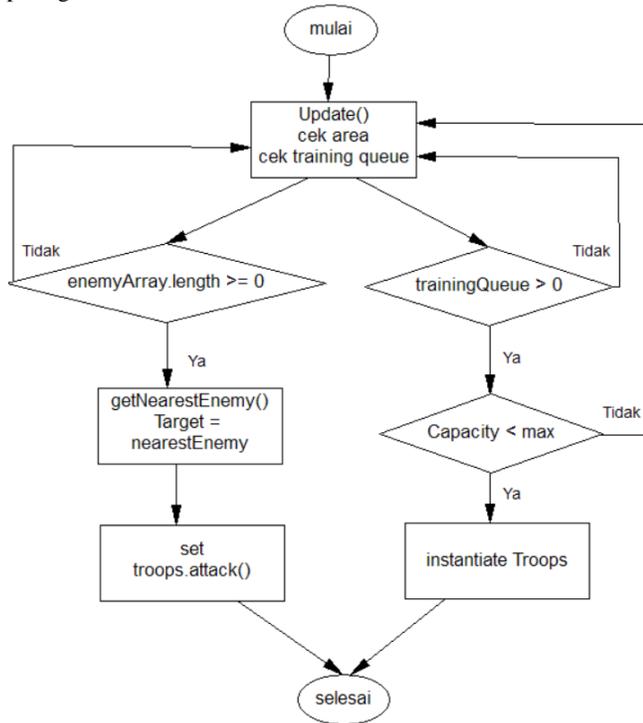
3. Alur Ubah Tipe Tower pada game AR Tower Defense dapat dilihat pada gambar 5.



Gambar 5. Activity Diagram Ubah Tipe Tower

3.4 Sistem Barrack

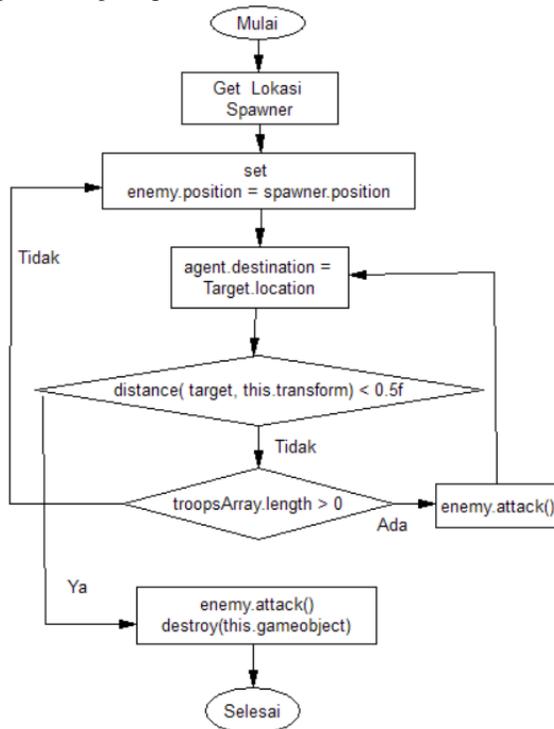
Alur sistem *barrack* pada *game AR Tower Defense* dapat dilihat pada gambar 6.



Gambar 6. Flowchart alur sistem *barrack*

3.5 Sistem Enemy

Alur sistem pergerakan musuh pada *game AR Tower Defense* dapat dilihat pada gambar 7.



Gambar 7. Flowchart Diagram Pergerakan *Enemy*

4. PENGUJIAN SISTEM

4.1 Pengujian Marker

4.1.1 Pengujian Ukuran dan Jarak Marker

Pengujian dilakukan dengan kamera 12 *mega pixel* dengan sudut 90° diatas *marker*. Hasil pengujian dapat dilihat pada tabel 1.

Tabel 1. Pengujian Ukuran dan Jarak Kamera Dengan *Marker*

		Jarak (cm)							
		< 15	< 20	< 40	< 45	< 50	< 85	< 90	<1 00
Ukuran (cm)	3x3	V	O	X	X	X	X	X	X
	5X5	V	V	V	O	X	X	X	X
	7x7	V	V	V	V	V	V	O	X

Keterangan:

V: Dapat terdeteksi dengan baik

O: Dapat terdeteksi, tetapi hasil tidak stabil (bergetar/ berpindah/ berputar)

X: Tidak dapat terdeteksi

4.1.2 Pengujian Marker Secara Bersamaan

Pengujian ini dilakukan untuk menguji kecepatan *Vuforia* dalam mendeteksi *marker* secara bersamaan. Pengujian dilakukan dengan *marker* berukuran 3cm x 3cm, kamera 12 *mega pixel* dengan sudut 90° diatas *marker* dengan jarak 15cm. Hasil pengujian dapat dilihat pada tabel 2.

Tabel 2. Pengujian Penggunaan *Marker* Secara Bersamaan

Jumlah	Hasil
2-5	Terdeteksi tanpa masalah
6-10	Terdeteksi tanpa masalah
> 10	Beberapa <i>marker</i> tidak dapat terdeteksi, karena terlalu banyak <i>marker</i> yang harus difokuskan. Terdapat beberapa <i>game object</i> yang berpindah tempat karena kesalahan deteksi <i>marker</i> .

4.2 Pengujian Program

4.2.1 Main Menu

Pada *main menu* terdapat pilihan *start*, *bantuan*, dan *keluar*. Jika *start* di-click maka *scene* akan me-load *level* dan dilanjutkan dengan proses *scanning marker*. *Bantuan* akan mengarahkan pemain ke *scene help* yang berisi panduan cara bermain dan cara mendapatkan *marker*. *Menu keluar* akan menutup aplikasi. Hasil pengujian dapat dilihat pada gambar 8.



Gambar 8. *Scene Main Menu*

4.2.2 Scanning Marker

Proses *scanning marker* harus diawali dengan *scanning marker map* terlebih dahulu, karena *map* merupakan *image target* yang menjadi *parent* dari *marker* lainnya. *Marker* yang wajib discan yaitu *map* sebagai *anchor* dari *camera world* di *Vuforia*, *spawner* sebagai lokasi awal musuh, dan *main base* sebagai lokasi akhir yang dituju. Hasil pengujian dapat dilihat pada gambar 9.



Gambar 9. Proses *scanning markers*

4.2.3 Barrack

Barrack digunakan sebagai tempat memunculkan pasukan untuk membantu pemain melawan musuh, menekan *barrack* akan menampilkan pasukan yang dapat dilatih. *Barrack* mempunyai kapasitas sebesar 20 dan akan berhenti memunculkan pasukan jika kapasitas telah penuh. Hasil pengujian dapat dilihat pada gambar 10.



Gambar 10. *Barrack*

Pada tampilan ini pemain dapat memilih pasukan yang diinginkan, menambah atau mengurangi jumlah yang ingin dilatih, dan melihat atribut pasukan. Hasil pengujian dapat dilihat pada gambar 11.

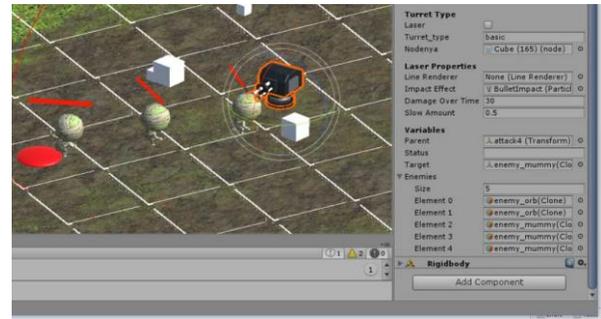


Gambar 11. Tampilan Pasukan pada *Barrack*

Pasukan seperti *warrior* dengan target darat tidak akan menyerang musuh dengan tipe udara dan membiarkannya lewat. Sedangkan pasukan seperti *archer* dan *wizard* akan menyerang musuh terdekat yang ada dalam area *barrack*.

4.2.4 Turret

Turret mempunyai 2 tipe target, yaitu darat dan udara. Untuk *turret* yang hanya bisa menyerang pada target darat, *turret* akan menghiraukan musuh dan membiarkannya lewat. *Turret* akan mencari semua musuh yang ada, membandingkan jarak antara musuh tersebut dengan *turret*, kemudian men-set musuh yang terdekat sebagai targetnya. Hasil pengujian dapat dilihat pada gambar 12.



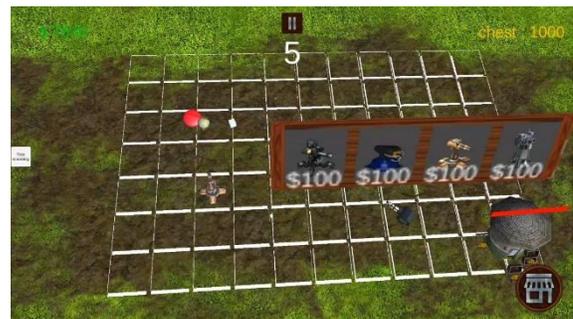
Gambar 12. Pencarian target pada *turret*

Menekan *turret* akan menampilkan tombol upgrade, *upgrade* dilakukan untuk meningkatkan atribut *turret* dan apabila telah mencapai level 5 *turret* dapat berubah menjadi tipe lain. Hasil pengujian dapat dilihat pada gambar 13.



Gambar 13. Tombol *Upgrade Turret*

Memilih salah satu tombol pada gambar dibawah akan mengubah tipe *turret*, atribut dan model *turret* akan berubah menjadi atribut dan model pada tombol tersebut. Hasil pengujian dapat dilihat pada gambar 14.



Gambar 14. Ubah Tipe *Turret*

4.2.5 Spell

Spell mempunyai 2 tipe target, yaitu musuh dan pasukan/*hero*. *Spell* pada game ini akan memberikan *effect* berupa *heal*, *burn* dan *slow* kepada target dalam areanya. *Spell* dapat diakses dari tombol *shop*, kemudian meletakkannya pada *node*. Implementasi *spell* pada game dapat dilihat pada gambar 15, 16, 17.



Gambar 15. Healing Spell



Gambar 16. Fire Spell



Gambar 17. Slow Spell

Spell akan mencari objek dengan *layer mask* team dalam areanya, memasukkannya kedalam *array*, kemudian untuk semua objek yang ada dalam *array* itu akan dipanggil fungsi yang memberikan *effect* ke objek tersebut.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian *marker*, dapat disimpulkan bahwa penggunaan Vuforia mempunyai masalah sebagai berikut:

- Jarak antara *marker* dengan kamera sangat mempengaruhi berjalannya aplikasi, bila terlalu jauh hasil yang dimunculkan tidak dapat stabil karena kamera kehilangan titik pusat dari peletakan objek tersebut.
- Ukuran pada *marker* juga menjadi pengaruh besar dalam proses mendeteksi, meskipun memiliki *marker* dengan *feature* yang cukup baik dan memiliki *rating* yang tinggi di Vuforia, jika ukuran *marker* yang akan di-*scan* berukuran kecil, kamera tidak dapat mendeteksi *feature* pada *marker* dengan baik.

Berdasarkan hasil pengujian program, dapat disimpulkan:

- Proses berjalannya program dapat berjalan dengan baik, semua fungsi utama dapat bekerja dengan baik pada *Android Device* dan tidak terpengaruh oleh masalah dari *Augmented Reality Vuforia*.

Berdasarkan hasil kuesioner, dapat disimpulkan bahwa:

- *Game* dengan *augmented reality* masih belum begitu dikenal di masyarakat umum.
- *Game AR Tower Defense* cukup mudah untuk digunakan dan ringan saat dijalankan.
- *Game AR Tower Defense* memiliki tingkat kesulitan yang cukup dan dapat memberikan pengalaman baru bagi pemain.

5.2 Saran

Berdasarkan pengujian yang telah dilakukan, penggunaan banyak *marker* secara bersamaan tidak disarankan, karena pendeteksian banyak *marker* secara bersamaan akan membuat posisi objek menjadi tidak stabil dikarenakan perubahan *anchor* pada kamera Vuforia.

Untuk pengembangan selanjutnya, diharapkan menggunakan API *Augmented Reality* yang lain yang lebih sempurna dan lebih mudah untuk digunakan oleh pengguna.

6. REFERENSI

- [1] Ayunita, R. 2013. Pembangunan Aplikasi Game RPG Untuk Pembelajaran Sejarah Awal Mula Berdirinya Kerajaan Singosari. Bandung: Universitas Komputer Indonesia.
- [2] Iman, N. 2013. Pembangunan Aplikasi Game Hybrid Shooter Side-Scrolling Destroyer Garuda Berbasis Desktop. Bandung: Universitas Komputer Indonesia.
- [3] Oktaviyan, D. 2013. Pembangunan game first person shooter (FPS) 3D Palagan Ambarawa berbasis desktop. Bandung: Universitas Komputer Indonesia.
- [4] Santoso, R. 2014. Pembangunan Augmented Reality Denah Museum Geologi Bandung Menggunakan Metode Markerless Berbasis Android. Bandung: Universitas Komputer Indonesia.
- [5] Satriawan, S. 2014. Penerapan Logika Fuzzy Untuk Optimalisasi Perilaku Tower Pada Game Tower Defense. Bandung: Universitas Komputer Indonesia.
- [6] Siswantoko, W. 2015. Implementasi Teknologi Augmented Reality Pada Game Duck Hunt Berbasis Android. Bandung: Universitas Komputer Indonesia
- [7] Somantri, A. 2013. Pembangunan Aplikasi Game Tower Defense Selamatkan Hutan Indonesia Berbasis Desktop. Bandung: Universitas Komputer Indonesia
- [8] Widiyansyah, F. 2014. Implementasi Augmented Reality Pengenalan Arca Menggunakan Metode Pattern Recognition di Museum Sri Baduga Berbasis Android. Bandung: Universitas Komputer Indonesia.
- [9] Yudiantika, A.R., Irma Permata Sari, Eko Suropto Pasinggi, dan Bimo Sunarfri Hantono. 2013. *Implementasi Augmented Reality di Museum: Studi Awal Perancangan Aplikasi Edukasi untuk Pengunjung Museum*. Yogyakarta: Konferensi Nasional Teknologi Informasi dan Komunikasi.