

PENGGUNAAN UNITY ENGINE PADA PERANCANGAN GAME THE CIENT DENGAN NAVIGATION MESH

Riyadi J. Iskandar¹, Antonius², Edwinyo³

^{1,2,3}Informatika, Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak
e-mail: ¹riyadijiskandar@gmail.com, ²antoniusok@yahoo.com, ³yoedwin98@gmail.com

Abstract

At present, the development of the gaming world is very rapid. This is reinforced by the source of newzoo.com which provides game development statistics from 2012 to 2019. These developments can not be denied anymore, because the game is indeed one of the hobbies that are most interested throughout the world. The development of existing games is not comparable with the mobile market that exists at the moment. This is evidenced in data from newzoo.com which contains that the sequence of platforms with the largest market currently is as much as 51% mobile, 25% console, and 24% PC. Data from wepc.com states that 53% of games are made for PC platforms, 37% of games are made for mobile platforms. The author uses a descriptive research design and research design in writing this study. The method of data collection used is by tracing the literature and articles that can be used as material to compile research. UML is used as a system analysis technique used to describe and document the design of game applications. C # programming language to do program code and apply navigation mesh. The resulting application is an offline survival horror game that uses a navigation mesh method with the Android platform. The conclusion of this study is that the application of the navigation mesh method is very supportive of the search path and for chasing and location of players. Suggestions from this research are that the addition of the FOV method for several monsters is needed so that the game challenges are more varied.

Keywords: Games, Offline, Survival Horror, Navigation Mesh, Android

Abstrak

Pada masa kini, perkembangan dunia *game* sangatlah pesat. Hal ini diperkuat dengan adanya sumber dari newzoo.com yang memberikan nilai statistik perkembangan *game* dari tahun 2012 hingga 2019. Perkembangan tersebut tidak dapat dipungkiri lagi, karena *game* memang merupakan salah satu hobi yang paling banyak peminatnya diseluruh dunia. Perkembangan *game* yang ada tidak sebanding dengan *market mobile* yang ada pada saat ini. Hal ini dibuktikan pada data dari newzoo.com yang berisi bahwa urutan *platform* dengan *market* terbesar saat ini adalah *mobile* sebanyak 51%, *console* sebanyak 25%, dan PC sebanyak 24%. Data dari wepc.com menyatakan bahwa sejumlah 53% *game* dibuat untuk *platform* PC, 37% *game* dibuat untuk *platform mobile*. Penulis menggunakan rancangan penelitian deskriptif dan perancangan penelitian dalam penulisan penelitian ini. Metode pengumpulan data yang digunakan adalah dengan menelusuri literatur dan artikel yang dapat digunakan sebagai bahan untuk menyusun penelitian. UML digunakan sebagai teknik analisis sistem yang digunakan untuk menggambarkan, dan mendokumentasikan perancangan aplikasi *game*. Bahasa pemrograman C# untuk melakukan *code program* dan menerapkan *navigation mesh*. Aplikasi yang dihasilkan adalah *game survival horror offline* yang menggunakan metode *navigation mesh* dengan *platform* Android. Kesimpulan dari penelitian ini adalah dengan penerapan metode *navigation mesh* sangatlah mendukung pencarian jalur serta untuk mengejar serta lokasi dari pemain. Saran dari penelitian ini adalah dengan penambahan metode FOV untuk beberapa *monster* sangatlah dibutuhkan agar tantangan permainan lebih bervariasi.

Kata Kunci: Game, Offline, Survival Horror, Navigation Mesh, Android

1. PENDAHULUAN

Pada masa kini perkembangan teknologi informatika sudah berkembang dengan sangat pesat. Berdasarkan sumber dari ourworldindata.com mengatakan bahwa perkembangan teknologi informatika selalu meningkat tiap tahun baik dari peningkatan kualitas *chips*, efisiensi sumber daya listrik untuk menjalankan komputer, peningkatan kualitas kamera tanpa meningkatkan harga jualnya, peningkatan kecepatan *microprocessor* dan lain sebagainya. *Game* pada masa kini sudah menjadi salah satu kegiatan atau hobi banyak orang, terutama pada kalangan anak muda. Berdasarkan sumber dari newzoo.com yang memberikan statistik bahwa perkembangan *game* dari tahun 2012 hingga 2019 terus meningkat sehingga pendapatan dari *market*

dalam dunia *game* meningkat. Akan tetapi perkembangan *game* pada *platform* Android tidak sebanding dengan market yang ada pada masa kini.

Game survival horror adalah *sub-genre* dari *game* yang ada yakni pemain *game* tersebut akan ditakuti dengan suasana mencekam, efek suara dan grafis *horror* yang ada di dalamnya. *Game survival horror* juga mengharuskan pemain yang memainkan *game* dengan *sub-genre* ini untuk mencari kunci-kunci yang ada dan menggunakan *item* yang disediakan untuk menyelesaikan teka-teki yang ada untuk menyelesaikan permainan. *Game survival-horror* yang beredar di pasaran Indonesia kebanyakan adalah *game* dari luar negeri, namun ada beberapa *game* buatan Indonesia sendiri dapat sampai terkenal ke luar negeri seperti “Dreadout”, “Pamali” dan lain sebagainya. Namun *game* buatan dari Indonesia sendiri kebanyakan *platform* PC bukan Android.

Platform yang digunakan untuk memainkan *game* sangatlah banyak jenis, akan tetapi *platform* yang paling banyak digunakan saat ini adalah *mobile* seperti Android dan IOS. Berdasarkan statistik yang bersumber dari newzoo.com mengatakan dengan jelas bahwa urutan *platform* dengan *market* terbesar saat ini adalah *mobile* (51%), *console* (25%), dan PC (24%). *Engine* Unity adalah salah satu *game engine* yang banyak dipakai oleh banyak orang. Berdasarkan ulasan yang dipaparkan dari pcgamesn.com mengatakan bahwa *engine* Unity adalah salah satu dari 9 *engine* terpopuler di dunia ini. *Navigation mesh* adalah sebuah fitur dari Unity yang merupakan pengembangan dari algoritma A* untuk melakukan pencarian jalur terpendek dengan *cost* terendah. Selain itu salah satu fitur yang ada adalah untuk menentukan area mana saja yang boleh atau tidak boleh dilalui oleh *artificial intelligence*.

2. METODE PENELITIAN

2.1 Metode Penelitian

Metode-metode penelitian yang digunakan penulis untuk melakukan penelitian sekaligus menyusun penelitian ini antara lain: Rancangan Penelitian, Metode Pengumpulan Data, Teknik Analisis Sistem, dan Teknik Perancangan Aplikasi.

2.1.1 Rancangan Penelitian

Rancangan penelitian yang digunakan penulis dalam penulisan penelitian ini adalah penelitian deskriptif dan perancangan eksperimen. Eksperimen dilakukan dengan perancangan dan implementasi sistem sebagai gambaran yang jelas dari masalah. Desain penelitian ini akan memberikan gambaran dari sistem aplikasi *game* tiga dimensi menggunakan *navigation mesh* pada Android.

2.1.2 Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan adalah dengan cara menelusuri pustaka-pustaka maupun literatur-literatur yang tersedia dalam objek penelitian. Data ini dapat berupa bahan-bahan pendukung seperti teori-teori, konsep-konsep yang berasal dari literatur-literatur.

2.1.3 Teknik Analisis Aplikasi

Teknik analisis sistem yang digunakan oleh penulis dalam penelitian ini adalah dengan menggunakan *Unified Modeling Language* (UML) untuk, memvisualisasikan, dan mendokumentasikan perancangan dari aplikasi *game* The Cient.

2.1.4 Teknik Perancangan Aplikasi

Perancangan sistem menggunakan program Unity 2018.2.7f1 untuk membuat aplikasi *game* Android, penerapan *navigation mesh* sebagai metode penelitian.

2.2 Landasan Teori

2.2.1 Aplikasi

Program aplikasi merupakan perangkat lunak (*software*) yang digunakan untuk tujuan tertentu^[1]. Aplikasi adalah koleksi *window* dan objek-objek yang menyediakan fungsi untuk aktivitas *user*, seperti pemasukan data, proses, dan pelaporan^[2].

2.2.2 Desain Sistem

Desain sistem adalah mendesain solusi sistem yang telah dikonsept pada saat analisis sistem berdasarkan kebutuhan dan permasalahan sistem sehingga sistem dapat diterapkan dengan menggunakan teknologi komputerisasi^[3]. Desain sistem adalah sebuah teknik pemecahan masalah yang saling melengkapi (dengan analisis sistem) yang mengangkat kembali bagian-bagian komponen menjadi sistem yang lengkap, harapannya sebuah sistem yang diperbaiki^[4]. Desain *input* adalah desain yang digunakan untuk menerima masukan data yang akan disimpan ke dalam *database* dan akan digunakan ke dalam proses transaksi maupun sebagai *output* yang dilaporkan^[5]. Desain *input* ditentukan dari diagram arus data (DAD) sistem baru yang telah dibuat. *Input* di DAD ditunjukkan oleh arus data dari kesatuan luar ke kesatuan proses dan bentuk tampilan *input* dan alat *input* yang ditunjukkan oleh proses pemasukkan data^[6]. Desain *output* adalah produk dari sistem informasi yang dapat dilihat^[6]. Desain *output* adalah *output* yang berupa tampilan di media kertas dan di layar *monitor*^[7]. Perancangan antarmuka adalah sesuatu yang harus dirancang sesuai dengan keahlian, pengalaman dan harapan pengguna^[8]. *Interface design is a complex process involving the UI design team, the target users and the client.* (Desain antarmuka adalah proses kompleks yang melibatkan tim desain UI, target pengguna dan klien)^[9].

2.2.3 Game

A game is a problem-solving activity, approached with a playful attitude. (Sebuah game adalah kegiatan pemecahan masalah, dilakukan dengan tindakan bermain)^[10]. Games are a type of play activity, conducted in the context of a pretended reality, in which the participant(s) try to achieve at least one arbitrary, nontrivial goal by acting in accordance with rules. (Game adalah jenis kegiatan bermain, dilakukan dalam konteks pura-pura nyata, dimana peserta mencoba untuk mencapai setidaknya satu wewenang, tujuan yang tidak sepele dengan bertindak sesuai dengan aturan)^[11] Genre game adalah gaya atau format dari sebuah game^[12]. Genre game adalah klasifikasi game yang didasari interaksi pemainnya^[13].

2.2.4 Navigation Mesh

A navigation mesh is a data structure that uses a set of two-dimensional (2D) regions to represent the walkable space in an environment. These regions are commonly used to plan visually convincing paths through complicated environments. (Navigation mesh adalah struktur data yang menggunakan satu set wilayah dua dimensi (2D) untuk mewakili ruang walkable di lingkungan. Daerah-daerah ini biasanya digunakan secara konvensional untuk meyakinkan jalur melalui lingkungan yang rumit^[14].) A Navigation Mesh is a term mostly used by the computer game community to refer to a data structure that is specifically designed for supporting path planning and navigation computations. While the term is well accepted and widely used, no formal definition or expected properties are attached to it. Path finding and navigation have become an important part of video games and the term navigation mesh has become the most used one when designing and implementing a given solution. (Navigation mesh adalah istilah yang sebagian besar digunakan oleh komunitas game komputer untuk merujuk pada struktur data yang dirancang khusus untuk mendukung perencanaan jalur dan perhitungan navigasi. Sementara istilah itu diterima dengan baik dan digunakan secara luas, tidak ada definisi formal atau sifat yang diharapkan yang melekat padanya. Pencarian jalur dan navigasi telah menjadi bagian penting dari video game dan istilah navigasi mesh menjadi yang paling banyak digunakan ketika merancang dan mengimplementasikan solusi yang diberikan^[15].)

2.2.5 Unity

Unity 3D is a Game Engine primarily, that also provides you with some flexibility on modifying and or creating simple 3D content! (Unity 3D adalah dasarnya sebuah mesin game, yang juga menyediakan beberapa fleksibilitas pada modifikasi dan atau membuat konten 3D yang simpel^[16].) Unity merupakan game engine yang dikembangkan oleh Unity Technologies. Software ini pertama kali diluncurkan pada tahun 2005 dan menjadi salah satu dari sekian banyak game engine yang dipakai oleh banyak pengembang profesional di dunia. Unity merupakan alat bantu pengembangan game dengan kemampuan rendering yang terintegrasi didalamnya^[17].

2.2.6 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet^[18]. Dalam membuat program biasanya yang banyak dipakai untuk menentukan target versi Android adalah nomor API Level^[19]. Android adalah salah satu sistem operasi yang pada awalnya, kemudian berkembang menjadi bahasa pemrograman yang banyak dicari dan digunakan oleh programmer^[20].

2.2.7 Unified Modelling Language (UML)

UML (Unified Modelling Language) adalah sebuah “bahasa” yang telah menjadi standar dalam industri yang visualisasi, merancang dan mendokumentasikan sistem piranti lunak^[21]. UML (Unified Modeling Language) adalah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berbasis objek^[22].

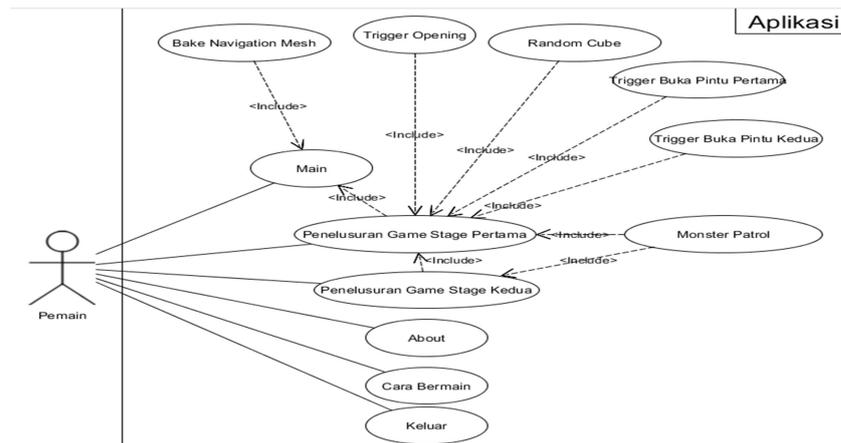
2.2.8 Kecerdasan Buatan

Artificial Intelligence (AI) is usually defined as the science and engineering of imitating, extending and augmenting human intelligence through artificial means and techniques to make intelligent machines. (Kecerdasan buatan (AI) adalah biasanya didefinisikan sebagai ilmu pengetahuan dan teknik meniru, memperluas dan menambah kecerdasan manusia melalui cara dan teknik buatan untuk membuat mesin cerdas^[23].) Artificial intelligence (AI) is a field of computer science that looks at the logic behind human intelligence. The field seeks ways to understand how we think and to re-create this intelligence in machines. Because of its nature, AI extends across human activities, making it relevant in different ways to every industry. (Kecerdasan buatan (AI) adalah bidang ilmu komputer yang melihat logika di balik kecerdasan manusia. Bidang ini mencari cara untuk memahami bagaimana kami berpikir dan menciptakan kembali kecerdasan ini dalam mesin. Karena sifatnya, AI meluas di seluruh aktivitas manusia, membuatnya relevan dengan cara yang berbeda untuk setiap industri^[24].)

3. HASIL DAN PEMBAHASAN

3.1 Gambaran Umum Rancangan Aplikasi Menggunakan Unified Modelling Language (UML)

3.1.1 Diagram Use Case



Gambar 1. Diagram Use Case

3.1.1.1 Main

Pada bagian ini pemain akan langsung dihadapkan pada tampilan permainan *game* The Cient dan pemain dapat langsung memainkan permainan ini. Sistem akan membawakan pemain langsung pada *stage* pertama dari *game* The Cient.

3.1.1.2 Bake Navigation Mesh

Bagian ini dilakukan pada saat pemain menekan tombol Main. Sistem akan melakukan *bake* pada area yang sudah ditentukan oleh pembuat *game* sebagai jalur area yang dapat atau tidak boleh dilalui oleh *monster*, sehingga *monster* memiliki navigasi jalur atau area pencarian yang boleh dilalui oleh *monster* untuk mengejar pemain yang memasuki radius pencarian maupun serangan.

3.1.1.3 Trigger Opening

Pada saat pemain memasuki permainan maka bagian ini akan menjalankan teks *story game*. Pada saat proses menjalankan teks tersebut sistem akan mem-*pause game* sehingga pemain tidak dapat bergerak. *Game* akan di-*resume* kembali pada waktu tertentu sehingga pemain dapat menggerakkan karakternya dan memainkan *game* The Cient seperti semula.

3.1.1.4 Monster Patrol

Pada bagian ini memberikan perintah untuk *monster* melakukan patroli pada titik yang sudah ditentukan oleh pembuat *game*. Selain itu pada bagian ini juga *monster* akan mengejar pemain apabila memasuki radius pencarian hingga menyerang pemain apabila memasuki radius serangan dari *monster* tersebut.

3.1.1.5 Random Cube

Pada bagian ini berisikan perintah yang digunakan untuk melakukan pengacakan lokasi *spawn cube*, sehingga lokasi *cube* pada saat permainan dijalankan akan berbeda-beda. Kemungkinan *spawn cube* berlokasi sama dari sebelumnya cukup kecil sehingga alur permainan pada *stage* pertama ini tidak terlalu monoton.

3.1.1.6 Trigger Buka Pintu Pertama

Pada bagian ini berisikan perintah *trigger* pada saat pemain mengambil lentera sehingga melewati *trigger* tersebut maka sistem akan membuka pintu dengan mengaktifkan animasi beserta *sound effect* buka pintu.

3.1.1.7 Trigger Buka Pintu Kedua

Pada saat pemain berjalan dan melewati *trigger* buka pintu kedua maka sistem akan menjalankan animasi dan *sound effect* ketuk pintu dan beberapa saat kemudian sistem akan menjalankan animasi dan *sound effect* pintu terbuka.

3.1.1.8 Penelusuran Game Stage Pertama

Pada bagian ini pemain akan menelusuri *game stage* pertama yang memiliki latar ruang bawah tanah. Pada saat proses penelusuran, pemain akan menemukan objek misi berupa *cube*, *cube hole*. Objek lain seperti *medical kit* yang berfungsi untuk memulihkan darah pemain.

3.1.1.9 Penelusuran Game Stage Kedua

Pada bagian ini pemain akan dilanjutkan untuk menelusuri permainan pada *stage* kedua dengan latar belakang hutan. Pada saat proses penelusuran, pemain diharuskan untuk mencari objek misi berupa pemantik, *signal rocket*, dan *tower*. Selain itu pemain juga dapat menemukan objek tambahan seperti *medical kit*. *Monster* pada *stage* ini memiliki kecepatan yang lebih dibandingkan dengan *stage* sebelumnya..

3.1.1.10 About

Pada bagian ini pemain akan memperoleh informasi mengenai pembuat *game* The Cient, *asset* yang digunakan, musik yang digunakan, dan lain sebagainya.

3.1.1.11 Cara Bermain

Bagian ini pemain akan memperoleh informasi mengenai fungsi-fungsi tombol yang sudah disediakan dan objek-objek yang harus diambil untuk menyelesaikan *quest* yang sudah disediakan.

3.1.1.12 Keluar

Merupakan tombol yang dapat digunakan oleh pemain apabila ingin keluar dari aplikasi *game* The Cient. Setelah menekan tombol Keluar, sistem akan langsung mengarahkan pemain pada *home screen smartphone* mereka masing-masing.

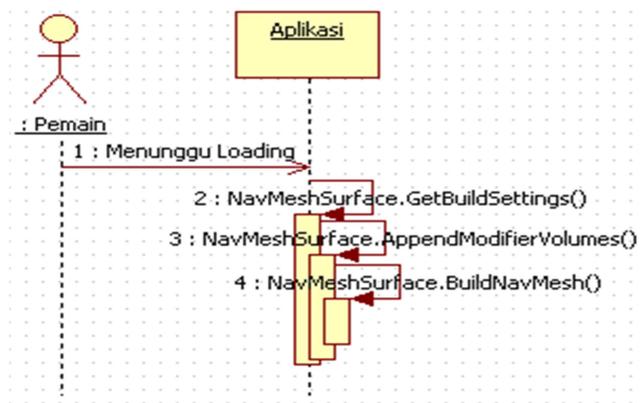
3.1.2 Sequence Diagram

3.1.2.1 Sequence Diagram Main

Sequence diagram main menggambarkan proses yang terjadi pada saat pemain memilih tombol Main pada saat berada pada menu utama. Sistem akan menjalankan perintah *MainMenu.Playgame()* yang berfungsi untuk memanggil *scene stage* pertama dan perintah *MainMenu.LoadAsynchronously()* berfungsi untuk memanggil *loading bar* sebagai penunjuk proses persiapan *game*. Sistem akan langsung memunculkan *loading bar* pada *interface* pemain. Setelah proses selesai ditunjukkan *loading bar* penuh dan sistem akan memulai permainan.

3.1.2.2 Sequence Diagram Bake Navigation Mesh

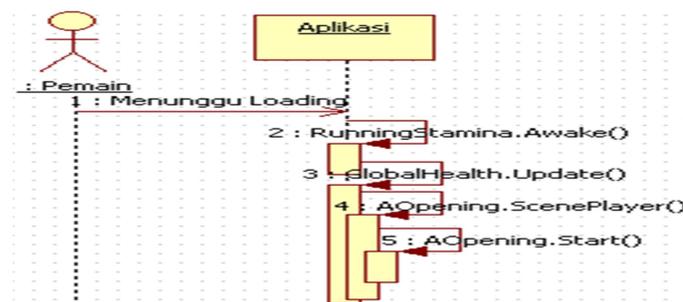
Pada saat pemain menyentuh tombol main maka sistem akan langsung menjalankan perintah untuk melakukan *bake navigation mesh*. Perintah *NavMeshSurface.GetBuildSettings()* berfungsi untuk megambil hasil pengaturan yang telah dibuat untuk di-bake, *NavMeshSurface.AppendModifierVolumes()* berfungsi untuk mengambil area dan volume *agent* yang akan di-bake. Perintah *NavMeshSurface.BuildNavMesh()* berfungsi untuk *bake terrain* sesuai dengan yang diinginkan oleh pembuat.



Gambar 2. *Sequence Diagram Bake Navigation Mesh*

3.1.2.3 Sequence Diagram Trigger Opening

Pada bagian ini sistem menjalankan perintah ketika pemain yang menunggu *loading game* hingga permainan dimulai. Sistem akan menjalankan perintah dari *method RunningStamina.Awake()* berfungsi untuk memberikan nominal stamina pemain, *method* dari *GlobalHealth.Update()* berfungsi untuk memberikan nominal darah pemain, *AOpening.ScenePlayer()* berfungsi untuk menjalankan *story game* dan menonaktifkan gerakan pemain hingga beberapa saat dan *AOpening.Start()* berfungsi untuk memberikan nominal darah pemain.

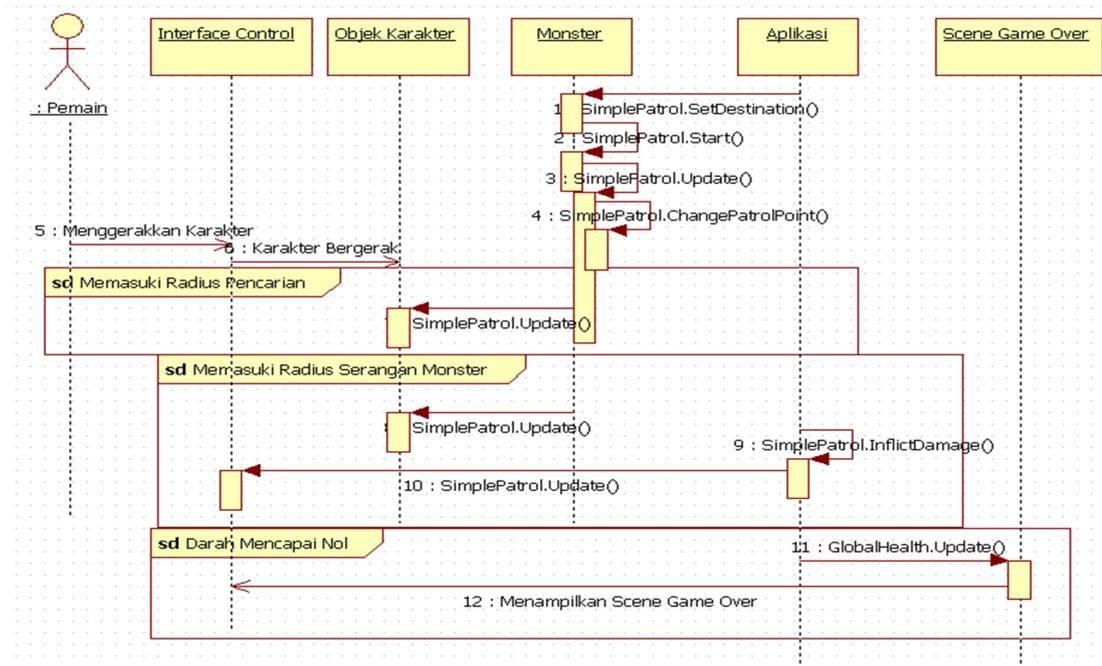


Gambar 3. *Sequence Diagram Trigger Opening*

3.1.2.4 Sequence Diagram Monster Patrol

Pada saat *game* siap dimainkan maka sistem akan menjalankan perintah *SimplePatrol.SetDestination()* berfungsi untuk memberikan tujuan yang akan dituju oleh *monster*, *SimplePatrol.Start()* berfungsi untuk memberikan target yang akan dicari seperti pemain dan memberikan kondisi untuk penggantian tujuan yang akan dituju *monster*. *SimplePatrol.Update()* berfungsi untuk memberikan kondisi apabila pemain belum menemukan pemain maka *monster* akan menjalankan animasi berjalan, menuju pada tujuan yang akan dituju

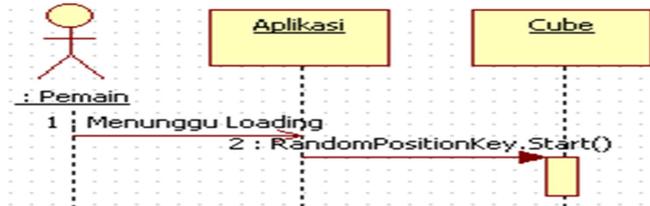
monster, dan kondisi monster mengejar pemain dan kemudian menyerang pemain apabila memasuki radius pencarian dan serangan dari monster tersebut. *SimplePatrol.ChangePatrolPoint()* berfungsi untuk mengganti tujuan patroli monster.



Gambar 4. Sequence Diagram Monster Patrol

3.1.2.5 Sequence Diagram Random Cube

Pada saat pemain menunggu *loading game* The Cient, sistem juga akan menjalankan perintah *RandomPositionKey.Start()* berfungsi untuk melakukan *random spawn cube* (gambar 5).



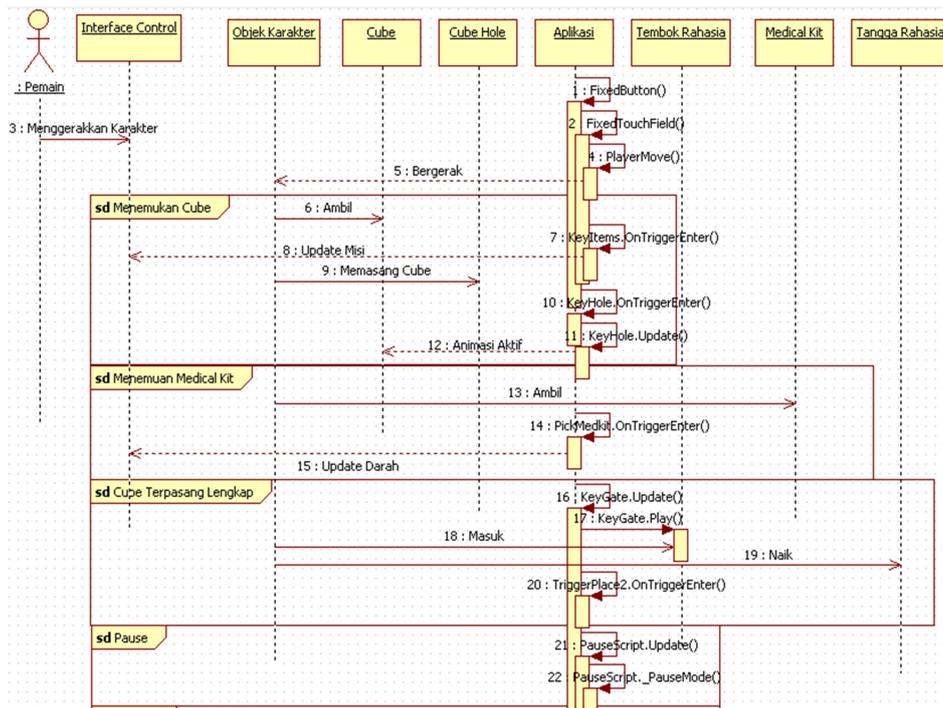
Gambar 5. Sequence Diagram Random Cube

3.1.2.6 Sequence Diagram Penelusuran Game Stage Pertama

Pada saat pemain memasuki permainan *stage* pertama, sistem akan menjalankan perintah *FixedTouchField()* berfungsi untuk menggerakkan kamera pada saat panel kamera digeser, *FixedButton()* berfungsi untuk menentukan *controller* mana yang akan dipakai, *PlayerMove()* berfungsi apabila analog digerakkan dan tombol disentuh maka karakter juga akan ikut bergerak sesuai perintah, setelah semua perintah dijalankan dengan baik maka karakter dapat dikendalikan oleh pemain dengan baik. Apabila pemain menemukan *cube* maka pemain mengambilnya dan sistem akan menjalankan perintah *KeyItems.OnTriggerEnter()* berfungsi untuk menonaktifkan objek *cube* tersebut dan kemudian mengubah misi. Ketika pemain memasang *cube* tersebut pada lokasi yang sudah ditentukan maka Sistem akan menjalankan perintah *KeyHole.OnTriggerEnter()* berfungsi untuk mengecek apakah *cube* yang akan dipasang cocok atau tidak, apabila cocok maka *cube* tersebut akan diaktifkan dan perintah *KeyHole.Update()* berfungsi untuk menjalankan animasi *cube* bergerak menuju tembok rahasia.

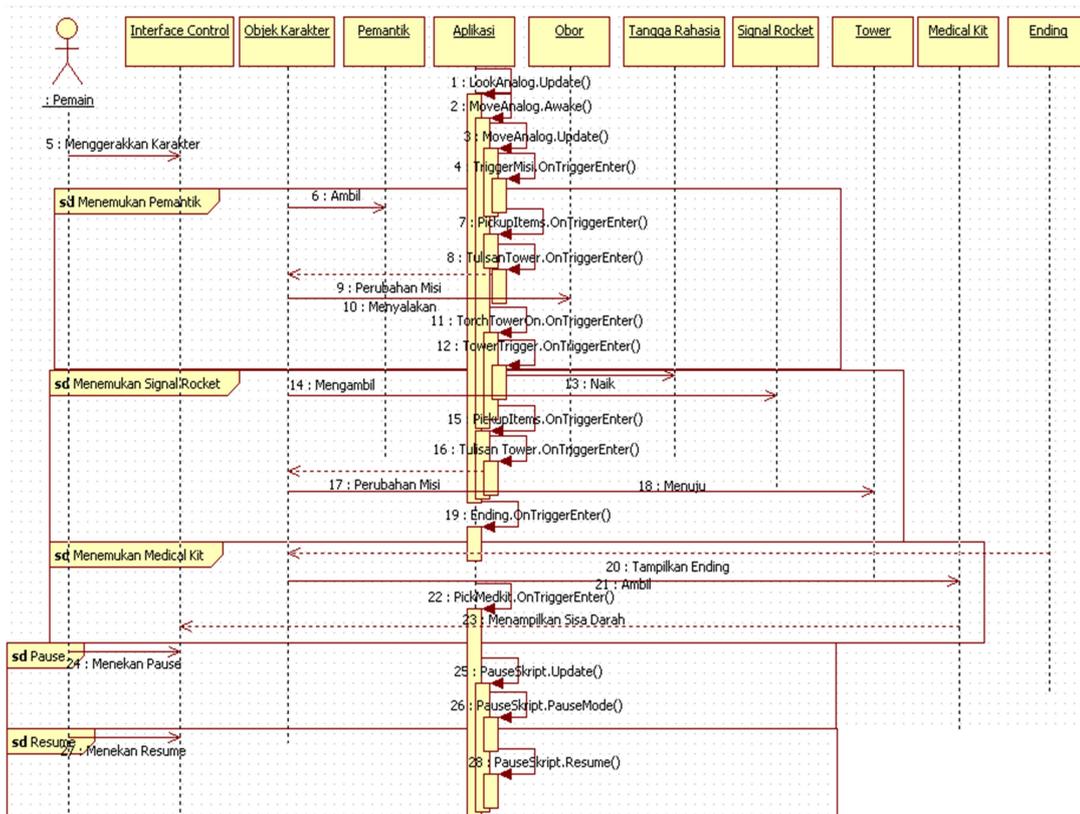
Pada saat pemain menemukan *Medical Kit* maka pemain mengambilnya sehingga sistem menjalankan perintah *MedicalKit.OnTriggerEnter* berfungsi untuk mengecek kondisi darah pemain, apabila darah pemain kurang dari 100 maka sistem akan memulihkan darah pemain, menghilangkan objek *Medical Kit* dan menampilkan sisa darah pemain. Apabila darah pemain masih 100 maka sistem tidak akan melakukan apa-apa. Apabila semua *cube* sudah terpasang lengkap, maka sistem akan menjalankan perintah *KeyGateUpdate()* berfungsi untuk mengecek apakah *cube* yang sudah terpasang lengkap atau belum. Pemain yang menekan tombol Pause maka sistem akan menjalankan perintah *PauseSkript.Update()* yang berfungsi untuk mengecek

apakah tombol Pause disentuh atau tidak, dan sistem menjalankan perintah *PauseSkript._PauseMode()* yang berfungsi untuk *pause* permainan, dan pada saat pemain menekan tombol Resume maka sistem akan menjalankan perintah *PauseSkript.Resume()* yang berfungsi untuk *resume* kembali permainan (gambar 6).



Gambar 6. Sequence Diagram Penelusuran Game Stage Pertama

3.1.2.7 Sequence Diagram Penelusuran Game Stage Kedua

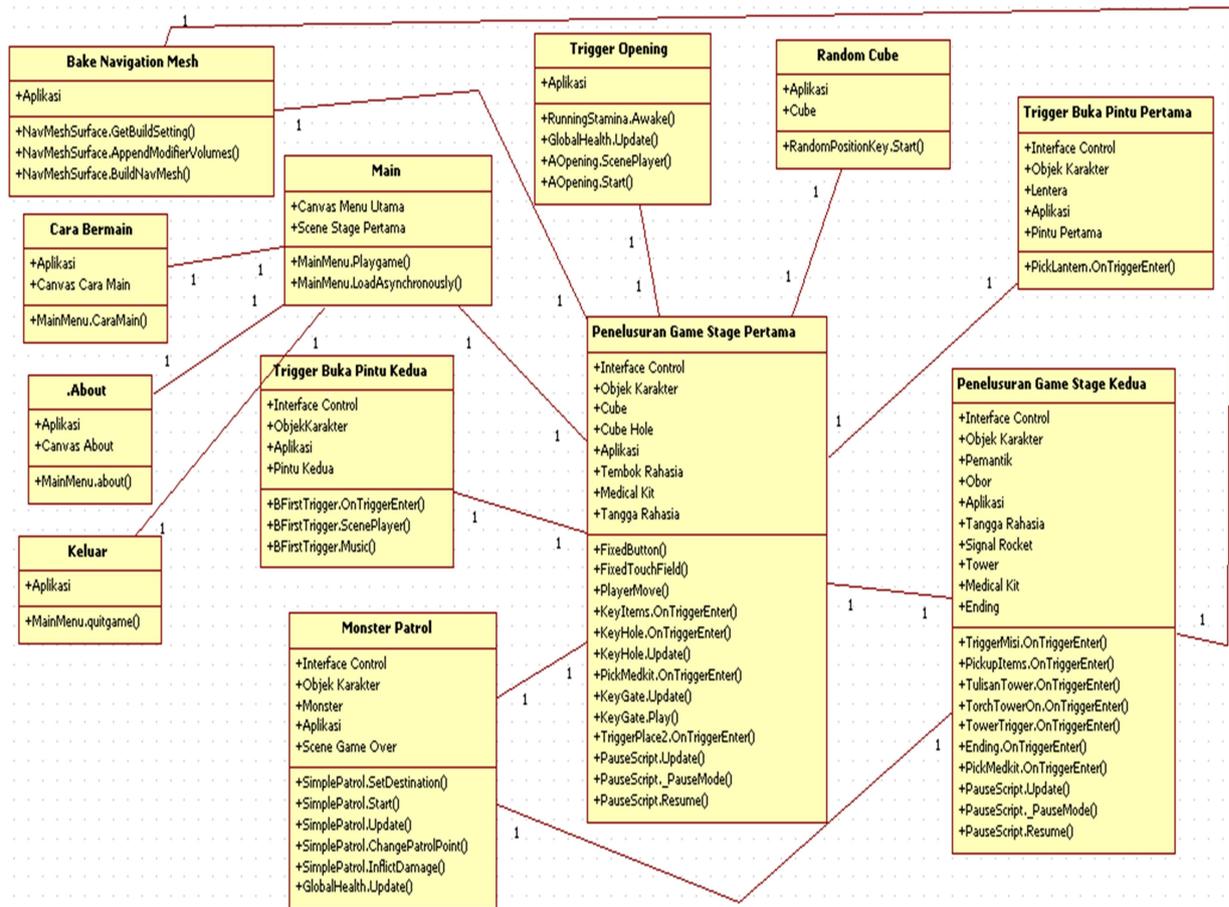


Gambar 7. Sequence Diagram Penelusuran Game Stage Kedua

Setelah pemain melewati *stage* pertama, pemain akan dilanjutkan untuk menelusuri *stage* kedua. Sistem akan menjalankan perintah *TriggerMisi.OnTriggerEnter()* berfungsi untuk memunculkan misi pada *stage* kedua yang harus diselesaikan oleh pemain. Pada saat pemain menemukan pemantik dan mengambilnya maka sistem akan menjalankan perintah *PickupItems.OnTriggerEnter()* yang berfungsi untuk mengambil pemantik, sistem menjalankan perintah *TulisanTower.OnTriggerEnter()* yang berfungsi untuk menampilkan misi selanjutnya, pada saat pemain menyalakan obor maka sistem akan menjalankan perintah *TorchTowerOn.OnTriggerEnter()* yang berfungsi untuk menyalakan obor dan perintah *TowerTrigger.OnTriggerEnter()* berfungsi untuk memainkan animasi tangga rahasia naik.

Pada saat pemain mengambil *signal rocket* maka sistem akan menjalankan perintah *method* dari *PickupItems.OnTriggerEnter()* yang berfungsi untuk mengambil objek *signal rocket*, dan perintah *TulisanTower.OnTriggerEnter()* yang berfungsi untuk mengganti misi. Ketika pemain menuju pada ujung *tower* maka sistem akan menjalankan perintah *Ending.OnTriggerEnter()* yang berfungsi untuk memunculkan animasi *ending* dan menonaktifkan semua kontrol karakter pemain. Perintah *PickMedkit.OnTriggerEnter()* dijalankan pada saat pemain mengambil *medical kit* dan berfungsi untuk pengecekan darah pemain, pada saat darah pemain kurang dari 100 maka sistem akan memulihkan darah pemain dan bila masih 100 maka tidak akan terjadi apa-apa. Pemain yang menekan tombol Pause maka sistem akan menjalankan perintah *PauseSkript.Update()*, *PauseSkript.PauseMode()*, dan ketika pemain menekan tombol Resume maka sistem akan menjalankan perintah *Pause.Skript.Resume()* (gambar 7).

3.1.3 Class Diagram



Gambar 8. Class Diagram

Pada kelas *bake navigation mesh* terdapat atribut aplikasi dan method yang digunakan adalah *NavMeshSurface.GetBuildSetting()*, *NavMeshSurface.AppendModifierVolumes()*, dan yang terakhir adalah method *NavMeshSurface.BuildNavMesh()*. Pada kelas *cara bermain* terdapat atribut aplikasi dan canvas cara main, sedangkan method yang digunakan adalah *MainMenu.CaraMain()*. Pada kelas *About* terdapat atribut yang digunakan adalah aplikasi dan canvas about, sedangkan method yang digunakan adalah *MainMenu.about()*. Pada kelas *keluar* terdapat atribut aplikasi dan method yang digunakan adalah *MainMenu.quitgame()*. Pada kelas *main*

terdapat atribut canvas menu utama dan scene stage pertama, sedangkan method yang digunakan adalah `MainMenu.Playgame()` dan `MainMenu.LoadAsynchronously()`. Pada kelas trigger opening terdapat atribut aplikasi, sedangkan method yang digunakan adalah `RunningStamina.Awake()`, `GlobalHealth.Update()`, `AOpening.ScenePlayer()` dan `AOpening.Start()`.

Pada kelas *random cube* terdapat atribut aplikasi dan *cube*, sedangkan *method* yang digunakan adalah `RandomPositionKey.Start()`. Pada kelas *trigger* buka pintu pertama atribut yang digunakan adalah *interface control*, objek karakter, lentera, aplikasi, dan pintu pertama, sedangkan *method* yang digunakan adalah `PickLantern.OnTriggerEnter()`. Kelas *trigger* buka pintu kedua terdapat atribut *interface control*, objek karakter, aplikasi, dan pintu kedua, sedangkan *method* yang digunakan adalah `BFirstTrigger.OnTriggerEnter()`, `BFirstTrigger.ScenePlayer()` dan `BFirstTrigger.Music()`. Kelas *monster patrol* atribut yang digunakan adalah *interface control*, objek karakter *monster*, aplikasi, dan *scene game over*, sedangkan *method* yang akan digunakan adalah `SimplePatrol.SetDestination()`, `SimplePatrol.Start()`, `SimplePatrol.Update()`, `SimplePatrol.ChangePatrolPoint()`, `SimplePatrol.InFLICTDamage()`, `GlobalHealth.Update()`.

Kelas penelusuran *game stage* pertama atribut yang digunakan adalah *interface control*, objek karakter, *cube*, *cube hole*, aplikasi, tembok rahasia, *medical kit*, dan tangga rahasia, sedangkan *method* yang akan digunakan pada bagian ini adalah sebagai berikut: `method LookAnalog.Update()`, `MoveAnalog.Awake()`, `method MoveAnalog.Update()`, `KeyItems.OnTriggerEnter()`, `method KeyHole.OnTriggerEnter()`, `KeyHole.Update()`, `PickMedkit.OnTriggerEnter()`, `KeyGate.Update()`, `method KeyGate.Play()`, `method TriggerPlace2.OnTriggerEnter()`, `method PauseScript.Update()`, `PauseScript._PauseMode()`, `PauseScript.Resume()`. Pada kelas penelusuran *game stage* kedua terdapat atribut *interface control*, objek karakter, pemantik, obor, aplikasi, tangga rahasia, *signal rocket*, *tower*, *medical kit*, dan *ending*, sedangkan *method-method* yang digunakan adalah sebagai berikut: `TriggerMisi.OnTriggerEnter()`, `PickupItems.OnTriggerEnter()`, `TulisanTower.OnTriggerEnter()`, `method TorchTowerOn.OnTriggerEnter()`, `method TowerTrigger.OnTriggerEnter()`, `Ending.OnTriggerEnter()`, `method PickMedkit.OnTriggerEnter()`, `method PauseScript.Update()`, `method PauseScript._PauseMode()`, dan `method PauseScript.Resume()` (gambar 8).

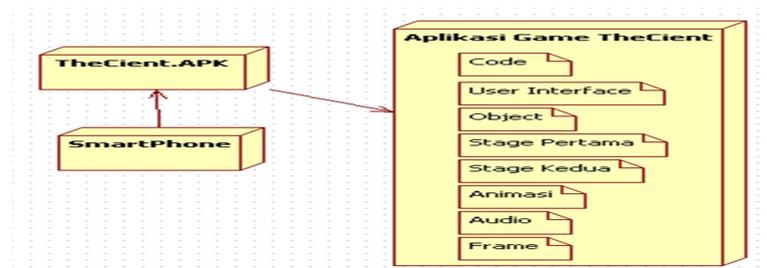
3.1.4 Deployment Diagram

Pada diagram berikut diperlihatkan *hardware* dan *software* yang ada pada aplikasi *game* dengan bagian pertama adalah *node smartphone* yang pada saat membuka aplikasi The Cient, maka pada aplikasi *game* tersebut terdapat delapan bagian yang bekerja yakni: *Code* melibatkan inialisasi dari semua komponen yang ada. Setiap kondisi yang berbeda tergantung dari pembuatannya maka *code* juga harus berubah berdasarkan kondisi yang ada, *User interface* berfungsi sebagai alat kendali bagi pemain untuk mengendalikan karakter, *Object* berfungsi sebagai benda yang menjadi sasaran yang harus dicari dan diambil oleh pemain, *Stage* pertama merupakan bagian dari lokasi *game* yang paling pertama kali muncul pada saat pemain menjalankan permainan, *Stage* kedua merupakan bagian dari lokasi *game* yang akan dimunculkan pada saat pemain telah menyelesaikan semua misi yang ada pada *stage* sebelumnya, Animasi berfungsi sebagai gerakan pada objek dalam *game* ini, Audio berfungsi untuk menambahkan *sound* efek maupun musik dalam *game* ini, *Frame* diolah dan diperbaharui terus menerus secara berkala atas permintaan dari pemain (gambar 9).

3.2 Tampilan Aplikasi

3.2.1 Tampilan Menu Utama

Pada menu utama ini memiliki *background music* yang selalu memutar lagu (*looping*) yang telah disediakan oleh pembuat *game* The Cient hingga program *game* ditutup atau pemain menekan tombol Main. *Music* yang digunakan pada *background music* ini diambil dari *game* Silent Hill dengan judul lagu Silent Hill: Promise Extended (gambar 10).



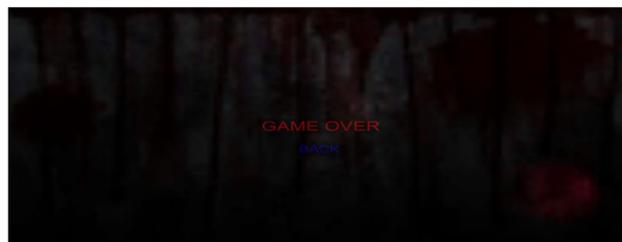
Gambar 9. Deployment Diagram



Gambar 10. Tampilan Menu Utama

3.2.3 Tampilan Game Over

Pada saat darah pemain mencapai nilai nol, maka sistem akan menganggap permainan telah berakhir karena darah pemain sudah habis. Sistem akan langsung memanggil *scene game over* sebagai pemberitahuan kepada pemain bahwa darah karakternya telah habis. Pada tampilan *scene game over* tersebut memiliki tombol Back yang berfungsi untuk menuju pada *scene* menu utama. Pada saat berada pada tampilan *game over* maka sistem juga akan mengaktifkan *background music* khusus untuk *scene game over* (gambar 11).



Gambar 11. Tampilan Game Over

3.2.4 Tampilan Ending

Setelah pemain menyelesaikan semua misi permainan maka sistem akan menjalankan *canvas ending* pada saat pemain men-*trigger* sistem tersebut. Pada bagian *ending* ini sistem akan mengaktifkan animasi yang dapat menggerakkan teks ke atas sehingga hasilnya berupa pergerakan teks *about* yang naik ke atas hingga teks tersebut habis atau ditampilkan semua. Pada saat *ending* ditampilkan maka sistem juga akan mengaktifkan *background music* lain yang digunakan khusus untuk bagian *ending*. Pemain dapat kembali pada menu utama bila diinginkan dengan cara menekan tombol Menu pada tampilan *ending* di pojok kiri atas. Saat pemain menekan tombol tersebut, maka *background music ending* tersebut akan dihentikan dan kembali menjadi *background music* menu utama.

4. KESIMPULAN

Berdasarkan hasil implementasi pada perancangan *game* The Cient yang menggunakan *platform* Android dan pembahasan yang sudah ada pada bab-bab sebelumnya, maka penulis dapat menyimpulkan bahwa:

- Aplikasi *game* The Cient yang sudah dibuat merupakan sebuah permainan yang ditujukan khusus untuk pengguna *smartphone* dengan OS Android,
- Game* ini menggunakan sisi pandang orang pertama dan bersifat *offline* sehingga tidak dapat bermain dengan orang lain,
- Monster* yang digunakan menggunakan *navigation mesh* berjalan dengan baik sehingga pencarian jalur untuk mengejar pemain berfungsi secara maksimal,
- Pengacakan lokasi *spawn cube* pada *game* berjalan cukup baik,
- User interface* yang dibuat sangat memudahkan pemain untuk mengendalikan karakter,
- Pilihan cara main menyediakan informasi yang cukup lengkap dan mudah dipahami,
- Peraturan permainan *game* The Cient cukup sederhana sehingga pemain awam mengerti apa tujuan dari *game* ini,
- Kondisi kemenangan permainan ditentukan dari keberhasilan menyelesaikan semua misi yang diberikan,
- Kondisi kekalahan dalam permainan ditentukan dari jumlah darah yang tersisa oleh pemain.

5. SARAN

Penulis menyadari bahwa *game* The Cient dengan *platform* Android ini masih belum sempurna. Penulis memiliki harapan agar para pembaca atau *programmer* yang lain dapat mengembangkan dan memperbaiki kekurangan-kekurangan yang ada pada *game* ini. Oleh karena ini, agar penelitian ini dapat memberikan manfaat bagi pembaca atau *programmer*, maka penulis akan memberikan beberapa saran, yaitu:

- a. Memberikan fitur *online* sehingga dapat dimainkan bersama teman untuk mengumpulkan objek misi,
- b. Membuat karakter *monster* yang ada lebih bervariasi seperti hantu yang ada di Indonesia,
- c. Penambahan fitur pemilihan *scene*,
- d. Menambah *story line* cerita agar lebih menarik,
- e. Meningkatkan animasi terutama pada *ending* agar lebih menarik,
- f. Membuat *sound effect* langkah kaki *monster* maupun pemain agar lebih menarik,
- g. Penggunaan metode FOV (*Field Of View*) untuk beberapa *monster* agar tidak monoton.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada civitas akademik Fakultas Teknologi Informasi dan kepada rekan-rekan yang telah membantu untuk menyelesaikan penelitian ini dan tidak lupa juga untuk berterima kasih kepada Tuhan Yang Maha Esa karena telah mengizinkan penulis untuk menyelesaikan penelitian ini dengan tepat dan sesuai dengan rencana penulis.

DAFTAR PUSTAKA

- [1] Hakim S, Rachman. (2010). *Buku Pintar Windows 7*. Penerbit PT Elex Media Komputindo. Jakarta.
- [2] Chan, Syahrial. (2017). *Membuat Aplikasi Database dengan PowerBuilder 12.6 dan MySQL*. Penerbit PT Elex Media Komputindo. Jakarta
- [3] Mulyani, Sri. (2016). *Metode Analisis dan Perancangan Sistem*. Penerbit Abdi Sistematika. Bandung.
- [4] Muslihudin, Muhamad dan Oktafianto. (2016). *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*. Penerbit CV Andi Offset. Yogyakarta.
- [5] Sugiran, Pardamean Daulay, Badrud Zaman, Faried Effendy, Lilis Amalia. (April 2016). "Journal Of Information Systems Engineering and Business Intelligence". Universitas Airlangga. Vol. 2, no. 1:hal 8.
- [6] Arif, M. Firman. (2019). *Analisis Perancangan dan Sistem Informasi*. Penerbit CV. Penerbit Qiara Media. Pasuruan.
- [7] Weli. (2019). *Aplikasi Kasus Siklus Transaksi Bisnis: Suatu Pendekatan Pengembangan Sistem Informasi Akuntansi*. Penerbit Universitas Katolik Indonesia Atma Jaya. Jakarta.
- [8] Maniah dan D. Hamidin. (2017). *Analisis dan Perancangan Sistem Informasi Pembahasan Secara Praktis Dengan Contoh Kasus*. Penerbit Deepublish. Yogyakarta.
- [9] Wood, Dave. (2014). *Interface Design: An Introduction To Visual Communication In UI Design*. Penerbit Bloomsbury Publishing. London.
- [10] Schell, Jesse. (2015). *The Art of Game Design : A Book of Lenses, Second Edition*. Penerbit CRC Press. Newyork.
- [11] Adams, Ernest. (2014). *Fundamentals of Game Design*. New Riders. Iowa.
- [12] Asmiatun, Siti dan A N. Putri. (2017). *Belajar Membuat Game 2D dan 3D Menggunakan Unity*. Penerbit Deepublish. Yogyakarta.
- [13] Sibero, Ivan C. (2009). *Langkah Mudah Membuat Game 3D*. Penerbit Mediakom. Yogyakarta.
- [14] Toll, Wouter G. van, Atlas F. Cook IV, dan Roland Geraerts. (Juni 2012). "Computer Animation and Virtual Worlds." Wiley Online Library. DOI: 10.1002/cav.1468.
- [15] Kallman, Marcelo dan M. Kapadia. (2016). *Geometric and Discrete Path Planning For Interactive Vrtual Worlds*. Penerbit Morgan & Claypool. California (USA).
- [16] Karamian, Vahe. (2016). *Introduction to Game Programming: Using C# and Unity 3D*. Penerbit Nooreon Inc. Los Angeles.
- [17] Pranata, B. Arya, A. K. Pamoedji, dan R. Sanjaya. (2015). *Mudah Membuat Game dan Potensi Finansialnya dengan Unity 3D*. Penerbit PT Elex Media Komputindo. Jakarta.
- [18] Kurniawan, Fachrul dan Gianto Widodo. (2015). *Cepat Menguasai Pemrograman Android*. Penerbit Universitas Brawijaya Press. Malang.
- [19] Jubilee Enterprise. (2015). *Mengenal Dasar-Dasar Pemrograman Android*. Penerbit PT Elex Media Komputindo. Jakarta.
- [20] Supardi, Yuniar. (2017). *Koleksi Program Tugas Akhir dan Skripsi dengan Android*. Penerbit PT Elex Media Komputindo. Jakarta.

- [21] Hanief, Shofwan dan D. Pramana. (2018). *Pengembangan Bisnis Pariwisata dengan Media Sistem Informasi*. Penerbit CV Andi Offset. Yogyakarta.
- [22] Sukanto, R A dan M. Shalahudding (2018). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek Edisi Revisi*. Penerbit Informatika. Bandung.
- [23] Shi, Zhongzhi. (2011). *Advanced Artificial Intelligence*. Penerbit World Scientific. Singapore.
- [24] Luce, Leanne. (2019). *Artificial Intelligence for Fashion: How AI is Revolutionizing the Fashion Industry*. Penerbit Apress. United States of America.