

# PERANCANGAN PC GAME *FIRST PERSON SHOOTER* MENGGUNAKAN UNREAL DEVELOPMENT KIT

Rizka Lukmana Afif<sup>1)</sup>, Kodrat Iman Satoto<sup>2)</sup>, Kurniawan Teguh Martono<sup>2)</sup>  
 Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro  
 Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia  
 lukmanaafif@gmail.com

**Abstract**— The development of hardware increasing rapidly has made game developers take advantage of a variety of new resources that can improve their games. Epic games is a mature game developer who managed to make thousands game and delivered to the hands of gamers . A game engine called Unreal Engine is a big secret behind the success of Epic Games . The game engine is free if you just want to learn or just want to create a personal project and the game is not to be in commercial purposes . It is unfortunate that many students don't even know of the existence of unreal engine , most of them make use of simpler game engine like game maker , rpg maker , fps creator , and so on. Though unreal engine is superior in any aspect other than the game engine , be it graphics , tools , mechanisms of development , flexible in export-import assets , etc . Based on this information , the author had the idea to make a first person shooter game using the unreal engine as the engine game .

Before doing the development process, the next step is studying the literature of unreal engine and other supporting software such as 3d studio max to create 3D assets , adobe flash to create the menus , adobe photoshop to create a 2D texture and speedtree assets to create the foliage elements . The next thing is to go into the design phase of scenarios , maps, missions , characters and items that will be placed in the game .

The next stage is the development and testing phase to test the game that has finished .The results of the design of this game is the realization of a first person shooter game application using unreal engine with features that can support the player 's interest in playing the game . It's also introducing unreal engine to students who are interested in designing games .

**Index Terms** - game engine , UDK , directx , game , first person shooter .

## I. PENDAHULUAN

**G**ame di awal tahun 80-an dirilis *game* tersebut masih tergolong primitif dikarenakan resolusi aplikasi yang amat kecil dan *hardware* pada saat itu kurang mencukupi untuk membuat *game* yang bergrafis baik. *Game* yang dirancang oleh para *developer* pada umumnya menggunakan teknik *full script* yang mana *programmer* menulis *code* untuk tiap sudut layar sehingga tercipta *game* yang memiliki logika dan grafis yang baik, akan tetapi hal itu memerlukan waktu yang tidak sebentar.

Dalam industri *game* dikenal *game engine*, yang merupakan *software* yang membantu *developer* untuk dapat merancang *game* dalam waktu yang singkat. Di

dalam *game engine* ini sendiri terdapat semua *tools* yang diperlukan untuk merancang sebuah *game*, baik itu *tool* animasi, merancang *game environment*, membuat *video*, serta *tool* untuk *coding/scripting*..

Unreal Development Kit (UDK) adalah *software* yang merupakan *game engine* dari Epic Games disertai dengan berbagai *tools* yang dapat membantu developer dalam membuat *game* yang diinginkan. Selain itu, developer dapat menggunakan *software* modelling 3D seperti 3Ds Max, Maya, Blender, dan lain-lain kemudian model tersebut di-*import* ke UDK, sehingga model tersebut dapat digunakan dalam pembuatan *game*.

“Pembuatan *Game Alien Hunter* Menggunakan FPS Creator X10” adalah judul penelitian tugas akhir tahun 2011 oleh Ferdhika Balfa dari sekolah tinggi manajemen informatika dan komputer (AMIKOM) Yogyakarta, yang membahas pembuatan *game* bergenre *First Person Shooter* menggunakan FPS Creator sebagai *game engine*-nya. FPS Creator merupakan *game engine* yang dapat dengan mudah membantu *developernya* membuat *game* yang spesifik bergenre FPS, tetapi UDK lebih unggul hampir disegala *tools* yang disediakan sehingga menungkingkan akan menghasilkan *game* yang lebih baik walaupun memang memerlukan spesifikasi komputer yang lebih tinggi.

Dengan berbasis latar belakang ini, terbesit ide untuk membuat *game* FPS (*first person shooter*) menggunakan UDK unreal engine yang lebih menarik perhatian *player* dengan segi grafis dan *gameplay* sehingga dapat lebih membuat *player* terhibur. Seperti *game* FPS pada umumnya, terdapat alur cerita dan misi-misi serta *scoring* di tiap *level* sehingga *player* dapat saling berkompetisi. *Game* ini akan ditujukan kepada *player* dengan kategori umur remaja hingga dewasa.

## II. DASAR TEORI

### A. Definisi Permainan

Definisi *game* menurut Chris Crawford yaitu sebuah aktifitas interaktif yang berpusat pada pencapaian, ada pelaku aktif, dan ada lawan. Chris Crawford mengklasifikasikan *game* (permainan) menjadi lima kelompok yakni permainan papan, permainan kartu, permainan atletik, permainan anak-anak, dan permainan komputer. Permainan papan adalah permainan tradisional yang terdiri dari sebuah bidang yang terbagi menjadi beberapa bagian yang dikuasai oleh pion bergerak. Pion yang bergerak merepresentasikan pemain (*player*),

1) Mahasiswa Sistem Komputer UNDIP

2) Dosen Sistem Komputer UNDIP

sedangkan bidang merepresentasikan lingkungan bermain. (Crawford,C.,1982)

### B. Pengertian Game Engine

Game engine merupakan software yang didesain untuk membuat dan mengembangkan video game. Fungsi utama yang diberikan oleh game engine meliputi rendering untuk 2D atau 3D graphic, collision detection, sound, scripting, animasi, artificial intelligence, networking, memory management dan masih banyak lagi.

Game engine juga sering disebut juga 'Game Middleware'. Game Middleware adalah software yang didalamnya sudah terdapat elemen-elemen yang dibutuhkan oleh seorang game developer, kebanyakan dari game middleware memberikan fasilitas guna menunjang pengembangan game dengan mudah seperti graphics, sound dan Artificial intelligence.

### C. Unreal Development Kit

Unreal development kit adalah salah satu game engine yang dirilis oleh Epic Games, sebuah developer besar di Utara Carolina, Amerika yang telah menciptakan banyak game terkenal yang hingga sekarang masih aktif dalam industri game. Sebutlah god of war, unreal tournament, batman, x-men, bulletstorm, infinity blade, spec ops dan masih banyak lagi merupakan hasil dari game engine yang dimiliki oleh epic games ini. Di dalam UDK terdapat tools yang dapat membantu developer dalam merancang game yang diinginkan. Berikut adalah tools dari UDK serta fungsi-fungsinya:

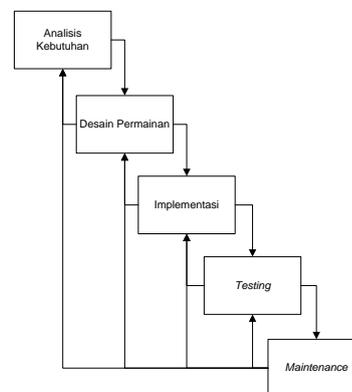
- *Unreal Editor* adalah editor dimana tingkat game yang diinginkan didesain. Editor ini developer dapat membuat environment, menempatkan static mesh, mengatur pencahayaan, dan optimisasi level.
- *Unreal Terrain* adalah tool untuk membuat dan memodifikasi permukaan lahan yang akan menjadi pijakan karakter dan objek-objek lainnya.
- *Unreal Matinee* adalah tools untuk membuat cinematic scene di dalam game. Developer dapat membuat adegan cinematic dengan menggunakan kamera aktor yang berada di editor dan di asosiasikan dengan unreal kismet.
- *Unreal Cascade* merupakan tool yang berfungsi untuk membuat dan memodifikasi efek-efek partikel fisik. Efek partikel ini dibuat dari beberapa texture kemudian dianimasikan menjadi satu sequence, misalnya partikel api, es, asap, dan masih banyak lagi.
- *Unreal Kismet* adalah tool yang digunakan untuk mempermudah developer dalam scripting dengan menampilkan script dalam bentuk visual. Unreal kismet memudahkan artist untuk merancang game walaupun tidak memiliki keahlian dalam programming, Function dari program dibentuk menjadi satu komponen yang didalamnya memiliki berbagai properties sesuai dengan fungsinya.

- *Unreal PhAT(Physics Asset Tool)* merupakan tool untuk membuat efek fisika pada objek, seperti collision(benturan) terhadap objek lain seperti hancurnya dinding ketika ditembak, Bergeraknya objek ketika disentuh player, dan lain-lain.
- *Unreal Materials* adalah tool yang digunakan untuk membuat dan memodifikasi material.
- *Unreal FrontEnd* adalah tool yang digunakan untuk build script, cook maps, console starter dan membuat paket dari game yang telah dibuat menjadi satu executable.
- *UnrealScript* adalah bahasa pemrograman yang digunakan dalam unreal engine. Unrealscript adalah bahasa pemrograman berbasis objek yang mirip dengan java, tetapi tanpa multiple inheritance( semua kelas di turunkan dari satu kelas umum yaitu kelas object) dan tidak seperti java, unrealscript memiliki sifat case insensitive. Unrealscript memiliki ekstensi \*.uc (contoh : MegapolisGame.uc).(EpicGames.2012)

## III. PERANCANGAN SISTEM

### A. Tahap-tahap dalam Pengembangan Sistem

Sebuah karya aplikasi multimedia, tidak akan terlihat bagus apabila cerita atau ide yang disajikan dalam animasi tersebut juga tidak bagus. Pembuatan sebuah cerita atau ide yang bagus, sangat diperlukan struktur cerita yang jelas. Untuk itu, perlu dilakukan tahap pengembangan seperti dalam diagram blok Gambar 1.

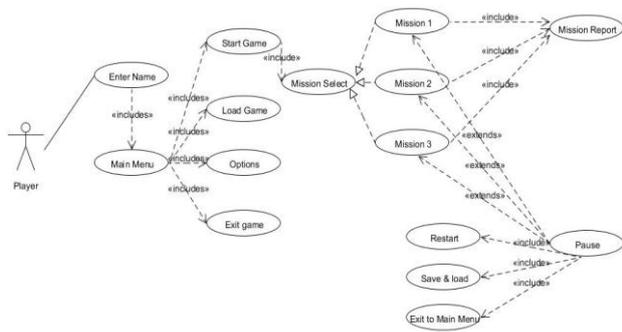


Gambar 1. Diagram waterfall dari tahap-tahap pengembangan Game

### B. Perancangan Pemrograman

#### 1). Diagram Use Case

Diagram ini menggambarkan interaksi antara pengguna dengan sistem, dalam hal ini yaitu pemain dan permainan yang dimainkan. Diagram use case dalam permainan ini dapat dilihat pada Gambar 2.



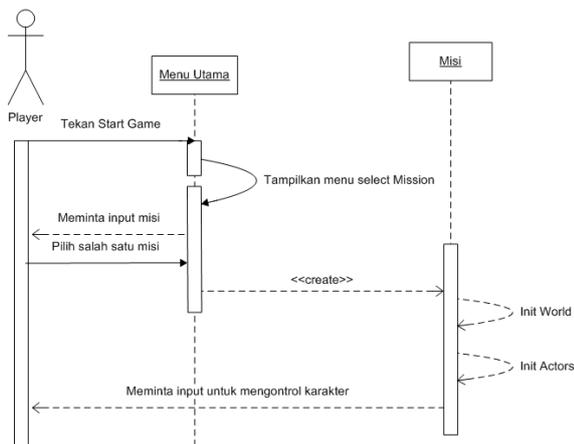
Gambar 2 Diagram Use Case pada permainan MegapolisGame

## 2). Diagram Sequence

Diagram ini menggambarkan fungsionalitas dari masing-masing use case dan interaksi yang terjadi antar objek dalam sistem.

### Diagram Sequence untuk Use Case Input nama

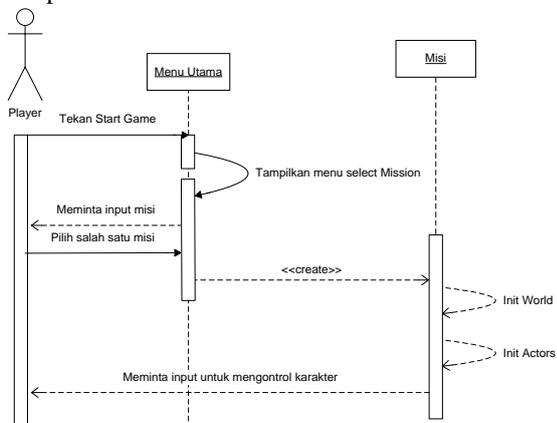
Diagram Sequence untuk Use Case input nama dapat dilihat pada Gambar 3.



Gambar 3. Diagram Sequence untuk Use Case input nama

### Diagram Sequence untuk Use Case Mission

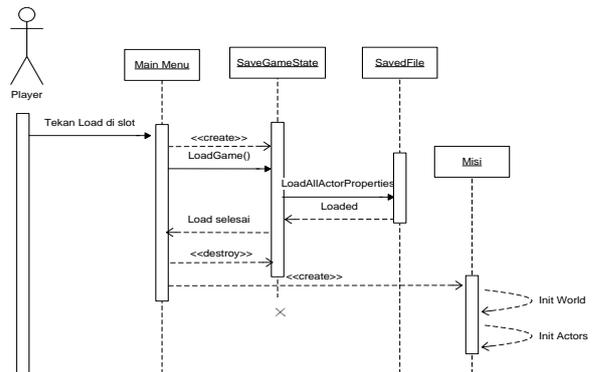
Diagram Sequence untuk Use Case Mission dapat dilihat pada Gambar 4.



Gambar 4. Diagram Sequence untuk Use Case Mission

### Diagram Sequence untuk Use Case Load

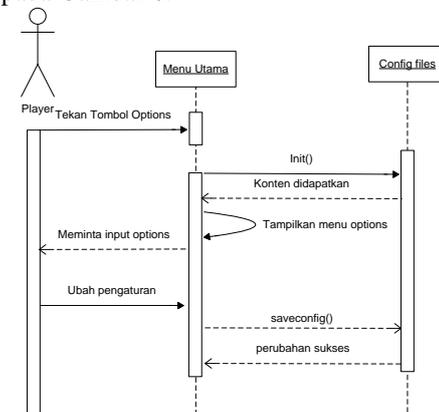
Diagram Sequence untuk Use Case Load dapat dilihat pada Gambar 5.



Gambar 5. Diagram Sequence untuk Use Case Load

### Diagram Sequence untuk Use Case Options

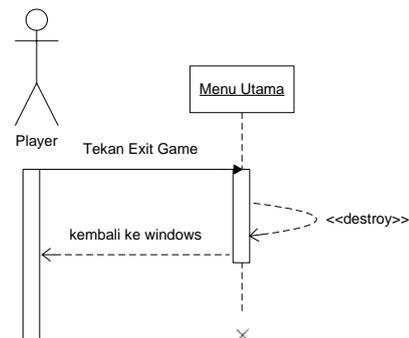
Diagram Sequence untuk Use Case Options dapat dilihat pada Gambar 6.



Gambar 6. Diagram Sequence untuk Use Case Options

### Diagram Sequence untuk Use Case Exit

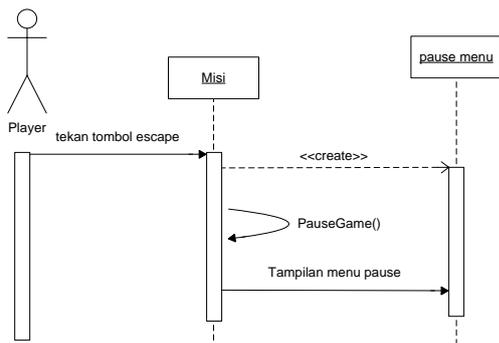
Diagram Sequence untuk Use Case Exit dapat dilihat pada Gambar 7.



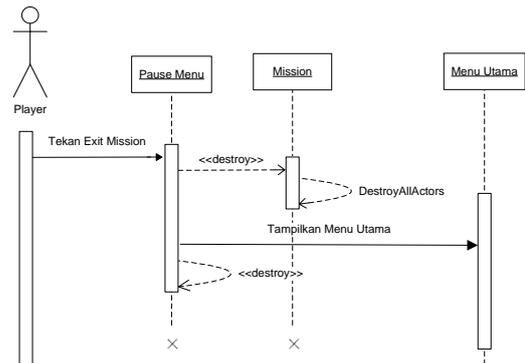
Gambar 7. Diagram Sequence untuk Use Case exit

### Diagram Sequence untuk Use Case Pause

Diagram Sequence untuk Use Case Pause dapat dilihat pada Gambar 8.



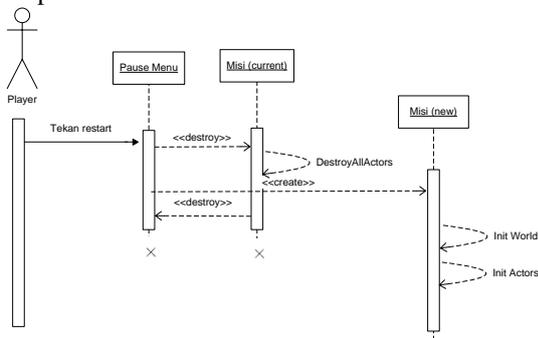
Gambar 8. Diagram Sequence untuk Use Case pause



Gambar 11. Diagram Sequence untuk Use Case to main menu

### Diagram Sequence untuk Use Case Restart

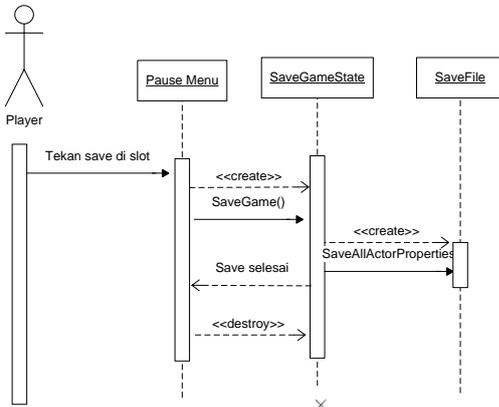
Diagram Sequence untuk Use Case Restart dapat dilihat pada Gambar 9.



Gambar 9. Diagram Sequence untuk Use Case restart

### Diagram Sequence untuk Use Case Save

Diagram Sequence untuk Use Case Save dapat dilihat pada Gambar 10.



Gambar 10. Diagram Sequence untuk Use Case save

### Diagram Sequence untuk Use Case to main menu

Diagram Sequence untuk Use Case to main menu dapat dilihat pada Gambar 11.

## IV. IMPLEMENTASI DAN PENGUJIAN

### A. Implementasi Program

#### 1. Core Gameplay Class

##### GameInfo

Inti alur permainan bersarang di kelas *GameInfo* yang memegang peran penting dalam membuat *rules* selama permainan berjalan. *GameInfo* memegang informasi kelas *playercontroller* yang merupakan alat kendali player untuk mengendalikan kelas *pawn* sebagai pion(karakter) di *game world*. Bukan hanya *playercontroller* saja, tapi juga *HUD(Heads up Display)* sebagai penyedia informasi yang player butuhkan untuk mengetahui poin *health*, jumlah *ammo*, senjata, dan lain-lain.

##### PlayerController

*Playercontroller* adalah *child* dari kelas *controller* yang memiliki fungsi untuk alat kendali kelas *pawn* yang menjadi karakter yang dikendalikan player di *game world*.

##### AIController

Seperti halnya *playercontroller*, *Aicontroller* adalah kelas *controller* yang berfungsi untuk mengontrol *pawn* karakter di *game world*, hanya saja yang di kontrol adalah *NPC(non playable character)* baik itu *ally* atau *enemy*.

##### Pawn

Kelas *Pawn* adalah pion yang dikendalikan oleh *controller*, baik itu *Playercontroller* maupun *Aicontroller*. Tiap *pawn* hanya dapat memiliki 1 *controller*, tetapi 1 *controller* dapat memiliki lebih dari 1 *pawn*. Tidak seperti *GameInfo* dan *controller* yang bersifat abstrak (tidak punya wujud di *game world*), kelas *pawn* memiliki wujud yang merupakan *Skeletal Mesh* (objek tiga dimensi yang memiliki 'tulang' agar objek dapat digerakkan atau dianimasikan).

##### Weapon

Senjata juga adalah salah satu *properties* dari *pawn* yang memiliki peranan penting. Dengan senjata ini maka *pawn* dapat saling membunuh satu sama lain. Tiap *pawn* dapat memiliki 1 senjata atau lebih tergantung dengan peraturan(*rules*) di kelas *gameinfo*. Kelas *weapon* adalah

*child* dari kelas *inventory* yang di atur keberadaannya oleh *inventory manager*.

#### Camera

Kelas *camera* adalah kelas yang membuat *player* dapat melihat *game world*, kelas ini diatur dari kelas *playercontroller*. Di *unreal engine*, sudut pandang dari kamera *player* dapat di ubah-ubah sesuai dengan kondisi yang diinginkan dari *gameinfo* maupun dari *kismet*.

#### HUD

HUD (*Heads up display*) adalah kelas yang mengatur tampilan informasi saat permainan berjalan. Informasi penting seperti *health*, jumlah *ammo* senjata, jenis senjata, peringatan, hint diatur di kelas HUD. Kelas HUD dapat menginisialisasi, menyembunyikan, menutup elemen-elemen *display* informasi ini agar informasi dapat tersampaikan dengan baik.

#### Gfx MoviePlayer

Kelas yang satu ini berhubungan erat dengan flash. Elemen flash \*.swf dapat di eksekusi menggunakan kelas ini. Kelas *MainMenu*, *PauseMenu*, *MainHUD*, *RestartMenu*, *MissionObjectiveWarning* dan *MissionReport* adalah *child* dari kelas *GfxMoviePlayer*. Dengan kelas ini *user interface* dibangun dengan kelas HUD sebagai menejernya.

#### Inventory Manager

Seperti yang telah dibahas di bagian kelas *weapon*, kelas *Inventorymanager* bertugas untuk mengatur *inventory* (termasuk kelas *weapon*). *Inventory manager* mengatur *weapon* memiliki konsep yang sama seperti kelas HUD mengatur kelas *GfxMoviePlayer*. Dapat menghapus, memnambahkan, dan masih banyak fungsi lainnya yang ditujukan untuk mengatur *inventory(weapon)*. Kelas ini dideklarasikan di *PlayerController*.

#### PlayerInput

*PlayerInput* dideklarasikan di *PlayerController* yang berfungsi untuk mengatur masukan dari *player* yang berasal dari *mouse & keyboard* atau *XBOX controller*.

#### SaveGameState

Kelas khusus ini bertugas untuk menyimpan (*save*) dan memuat (*load*) status permainan saat permainan sedang berjalan. Kelas ini dibuat dan dipanggil oleh kelas-kelas yang nilai dari variabelnya perlu disimpan. Kelas-kelas tersebut adalah *GameInfo*, *PlayerController* dan *Pawn*. Variabel-variabel tersebut di *encode* saat melakukan fungsi *savegame()* dan di *decode* saat melakukan fungsi *loadgame()*.

#### Other Actors

Selain yang disebutkan diatas, masih ada lagi aktor-aktor yang sengaja di rancang untuk mempermudah *gameplay* yaitu :

- *RoamNode* : Menjadi point AI musuh untuk berpatroli ketika belum melihat *player*.

- *CoverNode* : Menjadi point perlindungan AI ketika *fire fight* dengan *player*.
- *WeaponAmmo* : Sebagai amunisi dari senjata.
- *ObjectiveTarget* : Target dari misi.
- *Navigator* : Menjadi alat bantu navigasi yang menunjukkan jalan ke arah target dari misi.
- *Spawner* : Digunakan untuk poin lokasi dimana musuh akan di *spawn*.

### B. Implementasi Grafis

#### 1) Map Misi

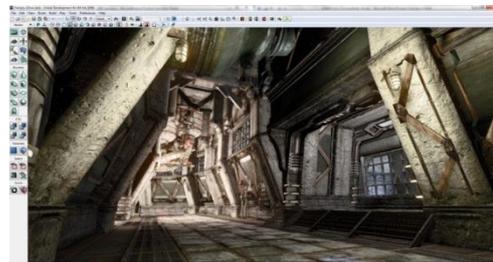
##### Map Misi Pertama



Gambar 12. Lorong di map misi pertama

Bertempat di lantai teratas gedung ICT, terdapat ruangan *server*, ruang tanaman, *dock* serta ruang kontrol yg menjadi tujuan utama.

##### Map Misi Kedua



Gambar 13. Lorong pabrik di map misi kedua

Lokasi berada di pabrik tempat dimana robot-robot diproduksi. Di sini *player* akan dikepung oleh puluhan robot yang siap untuk dibasmi.

##### Map Misi Ketiga



Gambar 14. Pemandangan luar gedung misi pertama di map misi ketiga

Map terakhir merupakan map terbuka di megapolis. Map akan dibatasi oleh dinding tak terlihat untuk mencegah *player* keluar dari area misi. Pada dasarnya

map ketiga adalah map besar yang menyatukan bangunan map pertama dengan map kedua.

## 2) Menu Masukan Nama



Gambar 15. Menu utama

Menu ini adalah menu yang akan pertama kali ditampilkan kepada *player* saat sudah memasukkan nama setelah itu akan ditampilkan menu utama.

## Menu Utama



Gambar 16. Menu utama



Gambar 17 Menu utama Level Select



Gambar 18. Menu utama options

Menu ini adalah menu yang akan pertama kali dilihat *player* saat sudah memasukkan nama di awal dan dimana *player* akan memilih untuk memulai permainan, load permainan sebelumnya, mengatur permainan, atau keluar dari permainan.

## Pause Menu



Gambar 19. Pause menu

Di *pause* menu akan disediakan berbagai menu yang akan membantu *player* untuk bernavigasi, *save* permainan, *load* permainan, keterangan misi yang sedang berjalan, *restart* misi, dan keluar dari misi dan menuju menu utama.

## Misi Selesai



Gambar 20. Rekap misi

Saat misi berakhir dengan sukses, maka *player* akan mendapatkan total nilai yang diperoleh selama berjalan misi tersebut dan *record* nilai tersebut akan disimpan untuk diletakkan di *scoreboard* untuk perbandingan nilai. Selain itu dari menu ini *player* dapat memilih untuk lanjut ke misi selanjutnya atau kembali ke menu utama.

## Restart Menu



Gambar 21. Restart Menu

Menu *restart* akan ditampilkan saat karakter yang dikendalikan *player* kehabisan poin *health*. Dari sini *player* dapat menentukan apakah akan mengulang dari awal misi, dari *check point* terakhir atau kembali ke menu utama.

### 3) Karakter

#### Cycus

Cycus adalah karakter utama yang dikendalikan oleh *player*. Memiliki kemampuan bertempur yang sangat baik, dapat regenerasi sel-sel yang terluka dan memiliki *defense system* yang mutakhir sehingga dapat bertahan menghadapi berbagai serangan musuh. Model 3D ini dibuat oleh crytek yang dikenalkan di *game* *Crysis Series*.

#### Shockwave



Gambar 22. Karakter bos musuh dengan animasi 'melee attack'

Shockwave adalah bos yang menjadi musuh terakhir dan yang terkuat dalam permainan ini. Dalam cerita robot yang sangat besar ini hanya akan menggunakan serangan *melee* dikarenakan belum selesainya modifikasi yang sedang berlangsung. Walaupun hanya dengan serangan jarak dekat, dapat menghancurkan apapun yang menghadangnya. Model 3D ini di desain oleh Activision yang dikenalkan di *Transformers 3 Dark of The Moon*.

#### Nemesis



Gambar 23. Karakter robot *nemesis*

*Nemesis* adalah *codename* untuk bot yang menggunakan senjata *sniper rifle*. *Damage* bertubi-tubi yang diberikan oleh musuh yang satu ini dapat melumpuhkan *player*. Model terdapat pada *UDK Assets*.

#### Trooper

Tidak terlalu berbeda dengan *nemesis*, hanya saja *trooper* adalah bot yang menggunakan senjata jenis *assault rifle* dan lebih tangkas dalam menghindari serangan *player*.

#### Hunter

Seperti *codename* yang diberikan, *hunter* merupakan bot yang mengejar *player* saat bot ini melihat dan mengetahui keberadaan *player*. Setelah didekat *player* bot ini akan menembak *player* dengan *assault rifle* dengan *damage* jarak dekat.

#### 4) Senjata dan *items* lain *Feline* (Senjata)



Gambar 24. Senjata *feline*

*Feline* adalah *sub-machine gun* yang berasal dari eropa, merupakan senjata yang memiliki tingkat akurasi yang cukup tinggi untuk membantu *player* mengalahkan musuh. Model 3D di adopsi dari *game* *Crysis*.

#### *Revolver* (Senjata)



Gambar 25. Senjata *Revolver*

*Revolver* yang satu ini bisa dibilang spesial karena berasal dari *game* action RPG *Darksiders 2*. Dengan *damage* jarak dekat yang besar, senjata ini menjadi pilihan untuk menghentikan serangan musuh yang berlari mendekati ke *player* seperti *Hunter*.

#### *Ammo*

Hanya dengan *ammo* ini, stok *ammo* untuk *feline* dan *revolver* akan terisi kembali.

#### C. Pengujian Permainan

Pengujian permainan ini dilakukan dengan menggunakan metode *black-box*. Tahap ini berisi serangkaian pengujian fungsi dan tombol pada permainan yang dijalankan pada sistem operasi *Windows 7* dan *Windows 8*. Cara pengujiannya hanya dilakukan dengan mengeksekusi kode sumber, kemudian diamati apakah hasilnya sesuai dengan spesifikasi kebutuhan. Fungsi-fungsi yang dapat dilakukan permainan yaitu

menampilkan menu **masukkan nama, menu utama**, memilih menu **Start game, load game, options** dan **exit**, memainkan permainan misi 1 sampai 3, tampilkan menu **pause, save game, Misi gagal**, dan **misi selesai**. Sedangkan tombol-tombol yang ada pada permainan yaitu tombol **startgame, restart, mission start, load game, save game, cancel, ok, apply, resume, exit mission** dan **next mission**. Sedangkan tombol pada controller yang diuji adalah **spasi, W, S, A, D, E, tombol mouse kiri dan kanan, R, C, shift kiri, CTRL kiri, scroll mouse** dan **Escape.a**

Hasil pengujian terhadap fungsi-fungsi dan tombol-tombol permainan menunjukkan hasil uji "Berhasil". Artinya secara fungsional, permainan sudah berjalan sesuai dengan spesifikasi kebutuhan dan skenario permainan.

Permainan ini sudah dijadikan berkas installer \*.exe, sehingga harus di-install terlebih dahulu di komputer untuk dapat memainkannya. Ukuran berkas permainan ini 680MB untuk ukuran 3 misi permainan ini cukup besar dan memang akan banyak menyita ruang pada *harddisk*.

#### KESIMPULAN

1. Penelitian ini telah menghasilkan permainan PC *desktop* yang berjudul "Megapolis" dengan menggunakan unreal engine sebagai *game engine*-nya.
2. Permainan dapat dijalankan pada sistem operasi Windows 7 dan Windows 8 dengan cara melakukan instalasi aplikasi terlebih dahulu sebelum memainkannya.
3. Setiap fungsi dan tombol-tombol yang ada dalam *game* ini telah sukses berjalan sesuai dengan perancangan.
4. *Game* ini memiliki banyak petunjuk yang ditampilkan untuk mempermudah permainan bagi *player* yang baru dalam permainan jenis FPS.
5. Dalam segi grafis tekstur objek 3D, animasi, efek cahaya, efek *bloom*, efek partikel dapat dikatakan memuaskan walaupun masih menggunakan DirectX9 sebagai API *render*nya (bukan DirectX11).
6. Semua *tools* yang disediakan oleh UDK sangat membantu dan mempersingkat waktu yang dibutuhkan dalam pembuatan *game* ini.
7. UDK adalah *game engine* yang *powerfull*, terbukti dalam penelitian ini walaupun hanya dengan satu orang *developer* dapat membuat sebuah *game modern*.

#### SARAN

Permainan yang sudah dibuat sudah berjalan dengan baik sesuai dengan tujuan awal dari penelitian ini. Adapun saran untuk pengembangan pada permainan "Memory Training" ini yaitu :

1. *Game* sebaiknya ditambahkan fitur *minimap* yang dimiliki oleh *game FPS modern* lain.
2. Penambahan fitur *controller binding* di menu *options*(pengaturan) akan menambah kenyamanan *player* dalam bermain.
3. Spesifikasi komputer minimum untuk memainkan "MegapolisGame" dapat dikategorikan tinggi karena LOD (*level of detail*) dari tiap objek 3D yang terbatas sehingga diperlukan objek tambahan untuk tiap objek 3D yang digunakan.
4. Lebih banyak misi akan lebih baik.

#### DAFTAR PUSTAKA

- Adam, E. (2010). *Fundamentals of Game Design*. USA : Pearson Education.
- Bigelow, D. (2012). *Construct Game Development*. Birmingham : PACKT.
- Crawford, C., (1982). *The Art of Computer Game Design*, Washington : Departement of History Washington State University.
- Cordone, R. (2011). *Unreal Development Kit Game Programming with UnrealScript*. Birmingham : PACKT.
- EpicGames (2012). *Development Topics* retrieved at 2AM, 12 December 2012 from <http://udn.epicgames.com/Three/WebHome.html>
- GameDev (2013). *Development & Programming Topics* retrieved at 10PM, 28 November 2013 from <http://www.gamedev.net>
- John, Doran. (2013). *Mastering UDK Game Development HOTSHOT*. Birmingham : PACKT.
- Martono, K. Teguh, Perancangan Game Edukasi "Fish Identity" Dengan Menggunakan Java TM, *Jurnal Sistem Komputer – Vol.1 No.1*, ISSN: 2087-4685, hal 49-53, 2011.
- Mooney, T. (2012). *Unreal Development Kit Game Design Cookbook*. Birmingham : PACKT.
- Moore, R. (2011). *Unreal Development Kit 3*. Birmingham : PACKT.
- Sherrod, A. (2012). *Beginning DirectX11 Game Programming*. USA : Course Technology.

#### BIODATA PENULIS

**Rizka Lukmana Afif**, lahir di Bengkulu, 22 Juli 1991. Telah menempuh pendidikan dasar di SD 81 Kota Bengkulu. Melanjutkan ke SMP Negeri 4 Kota Bengkulu, dan meneruskan pendidikan tingkat atas di SMA Negeri 5 Kota Bengkulu, lulus tahun 2009. Dari tahun 2009 sampai saat ini tengah menyelesaikan pendidikan Strata Satu di Program Studi Teknik Sistem Komputer, Universitas Diponegoro, Semarang, Indonesia Angkatan Tahun 2009.