

**PENGEMBANGAN PETA INTERAKTIF TIGA DIMENSI
GEDUNG UTAMA PT. SEMEN INDONESIA (PERSERO)
Tbk. WILAYAH TUBAN MENGGUNAKAN UNREAL
ENGINE**

Nama Mahasiswa : Randy Anandhita Ilham Firmansyah
NRP : 5209 100 041
Jurusan : Ssitem Informasi FTif-ITS
**Dosen Pembimbing : Dr. Eng. Febriliyan Samopa, S.Kom,
M.Kom**

Abstrak

Peningkatan teknologi yang berkembang pesat saat ini berdampak pada penyajian informasi mengenai bangunan, yang dahulu menggunakan dua dimensi (2D), tetapi sekarang menggunakan tiga dimensi (3D). Tampilan gambar 3D ini membuat bangunan terlihat lebih detail dan menarik, tak terkecuali pada perguruan tinggi Institut Teknologi Sepuluh Nopember (ITS) yang menjadi objek visualisasi 3D sebagai sarana promosi pada konsumen.

Pada tugas akhir ini penulis mengembangkan aplikasi peta tiga dimensi dengan menggunakan Unreal Development Kit (UDK) Engine. Aplikasi peta 3D ini menyajikan tampilan Gedung Utama PT. Semen Indonesia (Persero) Tbk Wilayah Tuban, sehingga pengguna mendapatkan pengalaman yang berbeda dengan mengetahui Gedung Utama PT. Semen Indonesia (Persero) Tbk Wilayah Tuban, secara virtual tanpa harus pergi ke tempat tersebut.

Kata kunci: Peta tiga dimensi, 3D Game Engine, Unreal Engine, Gedung Utama PT. Semen Indonesia (Persero) Tbk Wilayah Tuban.

DEVELOPMENT OF THREE DIMENSIONAL INTERACTIVE MAP ON PT. SEMEN INDONESIA (PERSERO) Tbk. BUILDING USING UNREAL ENGINE

Name : Randy Anandhita Ilham Firmansyah
NRP : 5209 100 041
Department : Information System FTIf-ITS
Supervisor : Dr. Eng. Febriliyan Samopa, S.Kom,
M.Kom

Abstract

Nowadays rapid technology developments have an impact on the presentation of information about the building, which previously used two-dimensional (2D) display, but now using a three-dimensional (3D) display. This 3D display makes the building look more detailed and interesting, including Institut Teknologi Sepuluh Nopember (ITS), which became the object of a 3D visualization as a promotion tool to consumers.

In this thesis the author made a three-dimensional map applications using the Unreal Development Kit (UDK) Engine. This application present the 3D map display of main building PT.Semen Indonesia (Persero) Tbk. Tuban Area interactively and accurately. The users of this application will get a different experience while virtually exploring the building without having to go to the place.

Keywords: *Three-dimensional map, 3D game engine, Unreal Engine, Main Building of PT.Semen Indonesia (Persero) Tbk. Tuban Area.*

BAB II

TINJAUAN PUSTAKA

Untuk memudahkan pemahaman tentang apa yang dilakukan pada tugas akhir ini, berikut ini dijelaskan tentang konsep dan teknologi apa saja yang digunakan atau di terapkan. Adapun penerapan teknologi yang dilakukan pada tugas akhir ini adalah sebagai berikut.

2.1. Peta Interaktif Tiga Dimensi ITS

Peta interaktif tiga dimensi ITS ini biasanya disebut INI3D yang kepanjangannya adalah ITS Now In 3D. INI3D pertama kali dikembangkan pada tahun 2010 oleh Bagit Airlangga (Sistem Informasi angkatan 2006) dengan membuat gedung jurusan Sistem Informasi, dan berlanjut ke pembuatan gedung jurusan-jurusan lainnya oleh pengembang penerusnya. Adapun standarisasi [4] yang menjadi rujukan untuk pengembang selanjutnya yaitu :

- Skala ukuran sebenarnya dengan peta yang akan dibuat adalah 1 meter pada ukuran sebenarnya sama dengan 64 Unreal Unit dalam UDK
- Pembulatan ukuran dibulatkan ke bawah tanpa koma
- UDK yang digunakan adalah UDK versi Februari 2012
- Tampilan langit dengan menggunakan pergantian siang dan malam yang sama dengan waktu 1 jam = 1 menit
- Aktor yang digunakan
- Interaksi standar yang harus ada, yaitu:
 - Membuka dan menutup pintu
 - Menyalakan dan mematikan lampu

Standarisasi ini tidak sepenuhnya juga menjadi rujukan dalam pengembangan INI3D ini, yang harus diperhatikan juga adalah proposional dalam dunia nyata atau sebenarnya, dikarenakan ada kalanya bangunan terlihat lebih kecil padahal sudah memakai standarisasi yang ada.

2.2. Game Engine

Game Engine adalah sistem software yang didesain untuk pembuatan dan pengembangan video games. Fungsi utama dari game engine adalah melakukan graphic processing dalam hal ini biasa disebut dengan rendering (cara grafik komputer membuat gambaran dari informasi seperti tekstur, pencahayaan bayangan), collision detection (metode perhitungan fisika ketika terjadi benturan antara 2 obyek), dan pengaturan suara. Dengan menggunakan game engine, programmer tidak harus menulis kode pemrograman dari awal.

Dahulu hampir semua perusahaan game mengembangkan game engine sendiri, menggunakannya untuk membuat game mereka. Semakin lama kebutuhan untuk game engine yang makin moderen meningkat dan perusahaan mulai berpikir bahwa mengembangkan engine sendiri terasa mahal. Hal tersebut memberikan ide ke beberapa developer untuk memulai mengembangkan game engine yang nantinya akan dikomersilkan ke perusahaan lain. Developer tersebut memberikan harga yang relatif murah dan juga perusahaan pembuat game tersebut tidak perlu membangun sebuah game dari awal sehingga pembangunan menjadi lebih cepat.

Dalam game engine terdapat bahan dasar yang dibutuhkan sebuah game untuk menjalankan tugasnya seperti rendering pixel demi pixel seperti tampilan grafis, menghitung physics seperti deteksi tubrukan, memperkirakan input tombol seperti pengaturan tombol dalam game, dan lainnya. Ibaratnya, sebuah mobil yang sudah setengah jadi berupa kerangka dan mesinnya, tinggal memberikan sentuhan desain dari model mobil tersebut.

Sebuah game engine dibagi lagi menjadi dua bagian besar yaitu API (Application Programming Interfaces) dan SDK (Software Development Kit). API adalah bagian operating system, services

dan libraries yang diperlukan untuk memanfaatkan beberapa fitur yang diperlukan contohnya DirectX. Sedangkan SDK adalah kumpulan dari libraries dan API yang sudah siap digunakan untuk memodifikasi program yang menggunakan operating system dan service yang sama.

Contohnya dalam Unreal Engine, dimana Unreal Engine mempunyai antarmuka yang sederhana untuk programmer untuk mengembangkan sebuah game dengan mudah, melalui scripting engine yang disebut UnrealScript dan juga libraries yang berisi model standar dan tekstur standar dan juga world editor yang disebut sebagai UnrealED.

2.3. Unreal Engine

Pada pengembangan tugas akhir ini, penulis menggunakan Unreal Engine 3 sebagai game engine. Unreal Engine 3 dikeluarkan oleh Epic Games, adalah salah satu game engine yang sangat populer dalam 3D First Person Shooter (FPS) atau game dengan perspektif orang pertama. Pemilihan Unreal Engine sebagai tools untuk membuat aplikasi peta 3D adalah karena game engine buatan Epic Games ini mempunyai beberapa fitur yang memungkinkan pengembang untuk membuat gambaran virtual yang sesuai dengan lingkungan nyata.

Unreal Engine memiliki beberapa komponen yang dapat berdiri sendiri-sendiri, namun tetap berada dalam kesatuan yang terpusat pada "core engine" [30]. Berikut adalah komponen-komponen dari Unreal Engine:

- Graphics Engine. Modul ini mengatur apa yang akan ditampilkan ke layar pengguna. Seperti benda apa yang harus didepan. Menampilkan material sesuai yang diatur sebelumnya. Hingga mengatur pencahayaan dari lingkungan virtual yang dibuat.

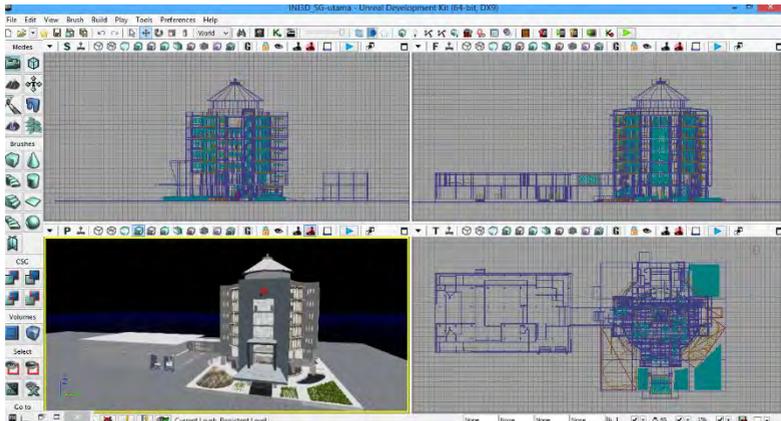
- Sound Engine. Modul ini mengatur efek suara dari lingkungan virtual.
- Physics Engine. Modul ini digunakan untuk mengatur benturan antar dua objek yang terjadi.
- Input Manager. Modul ini digunakan untuk mengatur input, seperti tombol ditekan, tombol lepas.
- Network infrastructure. Epic Games dengan gamenya Unreal Tournament telah berhasil mengembangkan network gaming yang efisien. Fitur network gaming yang efisien tersebut juga tersedia di Unreal Engine.
- Unreal Script Interpreter. Salah satu bahasa scripting yang dapat digunakan oleh programmer untuk mengatur apa yang dilakukan oleh engine, tanpa menyentuh source code asli. Script ini mirip dengan bahasa pemrograman terkenal lain seperti Java dan C++. Bahkan bahasa ini lebih mudah dari 2 bahasa pemrograman yang telah disebutkan sebelumnya. Unreal Script Interpreter adalah yang mengubah script yang dibuat oleh pengembang menjadi sesuatu yang bisa diproses oleh engine.

Setiap aspek dari Unreal Engine 3 telah dirancang dengan kemudahan-kemudahan yang diberikan untuk pembuatan konten dan pemrograman, dengan tujuan memberikan desainer sebuah kesempatan dan kebebasan untuk mengembangkan aset visual yang ada didalamnya.

2.4. Unreal Editor

Unreal Development Kit menyediakan alat untuk membuat dunia virtual yaitu Unreal Editor. Editor ini juga bisa melakukan *import* dari perangkat lunak pembuat objek tiga dimensi yang sudah banyak digunakan seperti 3D Studio Max dan Maya. Selain melakukan hal tersebut, Unreal Editor menyediakan cara untuk membuat tekstur, *material*, suara dan objek yang ada seperti di dunia nyata. Antarmuka pengguna menyerupai perangkat lunak

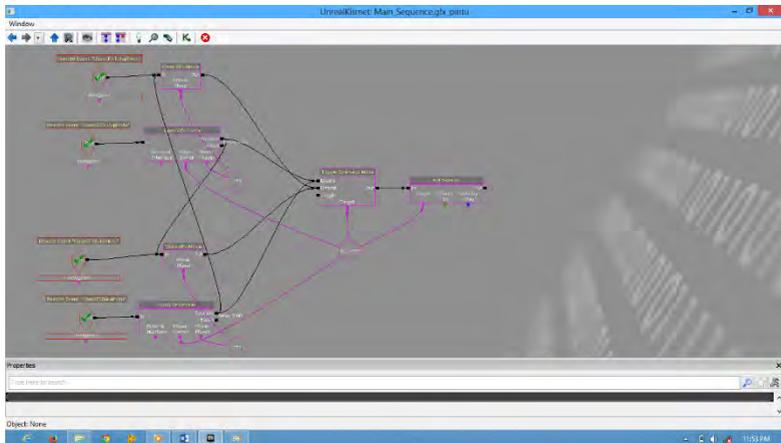
pembuat objek tiga dimensi seperti 3D Studio Max yang dapat dilihat pada Gambar 2.1.



Gambar 2.1. Unreal Editor

2.4.1. UnrealKismet

Dalam Unreal Editor terdapat banyak fungsi editor lainnya seperti UnrealKismet yang digunakan untuk mengolah logika *game*. UnrealKismet atau lebih sering disebut Kismet merupakan bentuk visual dari *script* yang sebenarnya kompleks, hingga Kismet menjadi tulang punggung interaksi didalam *game*. Dengan menggunakan *object* yang tersedia di dalamnya dan saling dihubungkan menjadi suatu modul di dalam Kismet, maka suatu interaksi atau semua yang akan dilakukan di dalam *game* menjadi lebih mudah dan cepat dibuat serta dipahami. Contoh modul di dalam Kismet dapat dilihat pada Gambar 2.2.

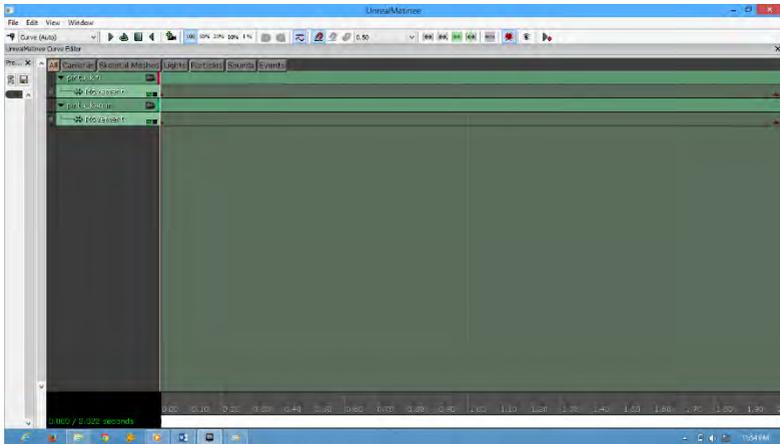


Gambar 2.2. Contoh sequence sederhana di dalam Kismet

Mulai dari sequence yang sederhana yaitu untuk membuat interaksi membuka pintu hingga sequence yang kompleks seperti untuk membuat sequence teleportasi yang lengkap.

2.4.2.UnrealMatinee

UnrealMatinee merupakan suatu editor bagian dari Unreal Editor yang digunakan untuk merubah properties Actor seperti lokasi, rotasi, ukuran, dan lainnya (termasuk dalam satu objek yang bernama Matinee). UnrealMatinee juga dapat digunakan untuk mengaktifkan suatu event didalam Kismet, seperti memainkan suara, membuat animasi objek, dan lainnya. Gambar 2.3 menggambarkan bagaimana sebuah Matinee dijalankan ketika sebuah objek trigger tersentuh oleh pengguna.

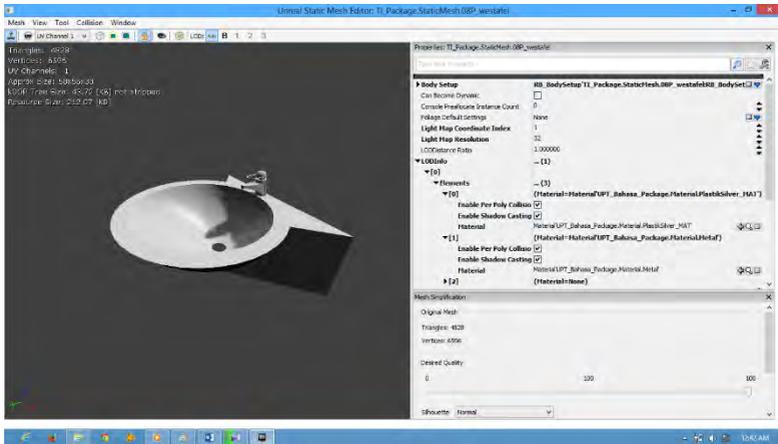


Gambar 2.3. UnrealMatinee merupakan editor Matinee di dalam UnrealKismet

UnrealMatinee ini memang terintegrasi dengan Kismet. Oleh karena itu, untuk menjalankan Matinee perlu membuat sequence objek di dalam Kismet.

2.4.3. Unreal Static Mesh Editor

Ada juga Unreal StaticMesh Editor yang digunakan untuk mengolah objek tiga dimensi. Dengan Unreal Static Mesh Editor ini, objek yang telah dibuat atau di-*import* ke dalam Content Browser, dapat diatur. Saat melakukan *import* ke dalam Content Browser, harus dipastikan bertipe StaticMesh, sehingga nantinya bisa melakukan pengaturan properties pada objek seperti mengubah material-nya dan mengatur collision-nya. Tampilan Unreal Static Mesh Editor dapat dilihat pada gambar Gambar 2.4.



Gambar 2.4. Contoh penggunaan Unreal Static Mesh Editor

Khusus untuk aplikasi pengolah objek tiga dimensi Autodesk 3ds Max, terdapat beberapa tipe file hasil pemodelan objek tiga dimensi yang dapat diterima oleh Unreal Editor. Tipe file tersebut adalah sebagai berikut:

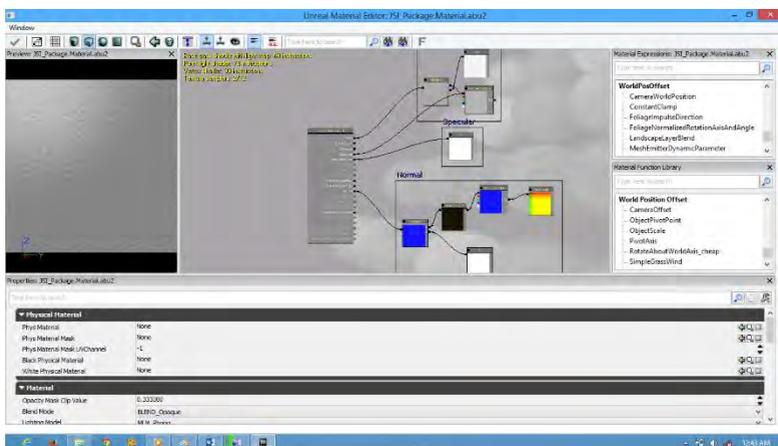
- **.ASE**
Karakter objek tiga dimensi yang dihasilkan dengan tipe file ini hanya dapat menerima satu jenis tekstur yang bisa ditempelkan pada objek hasil *import* dari tipe file ini.
- **.FBX**
Karakter objek tiga dimensi yang dihasilkan dengan tipe file ini dapat mengatur beberapa jenis tekstur yang bisa ditempelkan pada objek hasil *import* dari tipe file ini.
- **.DAE**
Karakter objek tiga dimensi yang dihasilkan dengan tipe file ini dapat mengatur beberapa jenis tekstur yang bisa ditempelkan pada objek hasil *import* dari tipe file ini.

2.4.4. Unreal Material Editor

Ketika membuat sebuah `Level Map` dengan Unreal Editor perlu untuk memberikan tekstur pada objek 3D di dalam `Level Map` agar terlihat seperti bentuk nyatanya. Sebuah tekstur disebut `material` dalam Unreal Editor yang dapat dibuat dengan menggunakan bantuan dari Unreal Material Editor. `Material` dapat ditempelkan pada `Level Map` seperti pada permukaan geometri tembok, lantai, atap dan lainnya agar terlihat lebih hidup atau nyata.

Konsep Unreal Material Editor secara umum adalah terdapat dua jenis gambar dengan tipe file tertentu (misalnya `.PNG` atau `.TGA`) yang dapat diolah menjadi sebuah `material`. Gambar pertama adalah gambar tekstur yang biasa kita lihat di kehidupan nyata (misalnya tekstur paving) tanpa mengetahui tekstur naik turunnya. Gambar tersebut biasa disebut *height image*. Gambar kedua adalah gambar yang menyimpan data tekstur atau lebih sering disebut *normal image*.

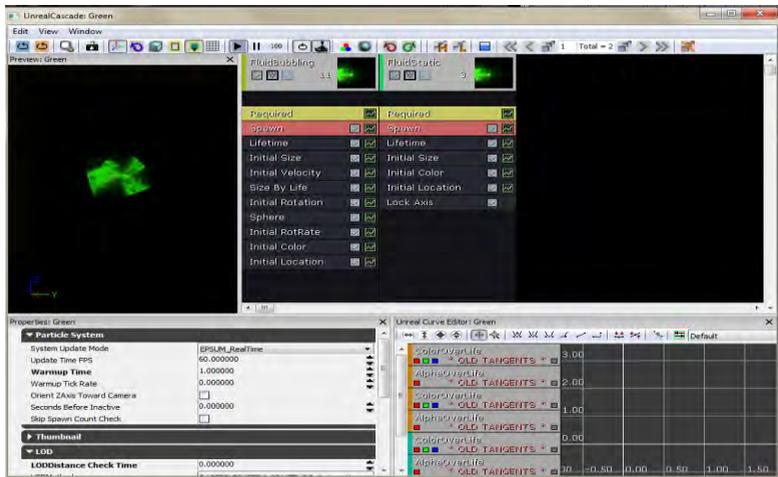
`Material` yang dibuat melalui Unreal Material Editor bisa hanya sekedar warna saja, namun dapat ditambahkan efek-efek semisal pantulan cahaya seperti pada logam, efek memancarkan cahaya atau bahkan `material` yang bisa bergerak seperti arus air kolam. `Material` yang bisa bergerak dapat dibuat dengan menggunakan `Material Expression` yang tersedia kemudian membuat suatu `sequence` dari kumpulan `expression` yang diatur sedemikian rupa dan digerakkan dengan kecepatan tertentu.



Gambar 2.5. Unreal Material Editor dengan contoh tekstur yang kompleks

2.4.5. Unreal Cascade

Dengan menggunakan Unreal Cascade, suatu particle system dapat dibuat. Particle system yang masuk dalam level map disebut dengan emitter. Dengan memanfaatkan *tools* yang ada, komponen-komponen particle system dapat dibuat dengan dasar material atau StaticMesh. Salah satu contoh particle system yang dapat dibuat yaitu ledakan seperti pada Gambar 2.6



Gambar 2.6. Particle system sebuah cahaya interaksi pintu

2.4.6. Unreal AnimSet Editor

Unreal AnimSet merupakan salah satu bagian dari Unreal Engine yang digunakan untuk mengatur aktor yang akan digunakan di dalam peta 3D kita. Beberapa pengaturan yang umum dilakukan, meliputi material aktor, AnimSet aktor serta lokasi dan rotasi aktor. Dengan memanfaatkan fungsi pengaturan material, bisa mempercantik tampilan aktor, seperti memberikan warna baju, kulit, dan lainnya.

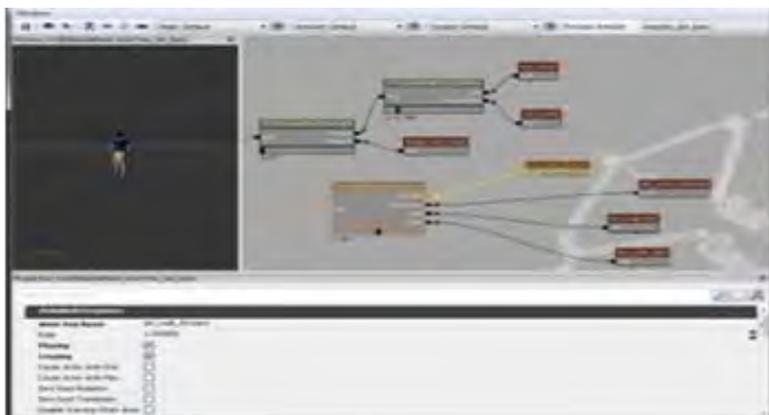
Sedangkan melalui pengaturan AnimaSet aktor, dapat diberikan animasi untuknya, seperti bergerak maju, mundur, loncat dan lainnya. selain itu, juga bisa mengatur posisi aktor, melalui pengaturan koordinat x, y dan z.



Gambar 2.7. AnimSet Aktor yang telah diberi material

2.4.7. Unreal AnimTreeEditor

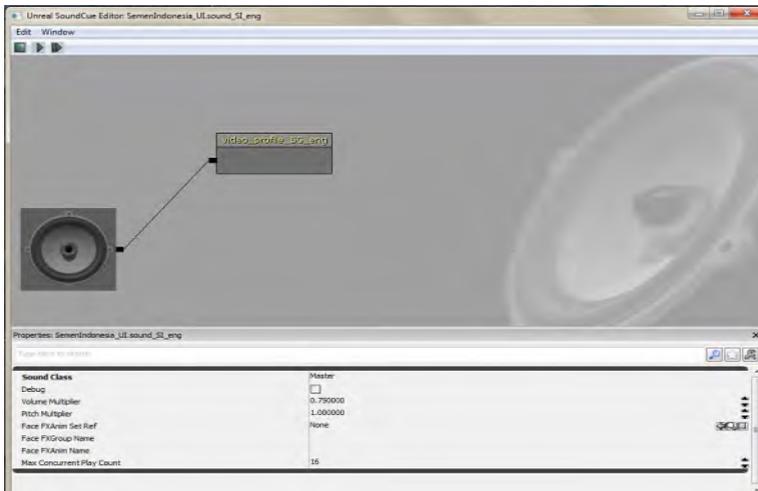
Di dalam AnimTree Editor ini, dapat dilakukan pengaturan kapan suatu AnimSet aktor akan dijalankan saat peta 3D dimainkan. misalnya, AnimSet gerakan ke depan akan terjadi ketika ditekan tombol panah ke atas. dalam pembuatannya, digunakan *sequence object* yang sudah disediakan didalam editor ini.



Gambar 2.8. Pembuatan animasi aktor dalam Unreal Animset Editor

2.4.8. Unreal SoundCue Editor

Unreal SoundCue Editor digunakan untuk memainkan suara di dalam peta3D. Suara yang dibuat di dalam editor ini berasal suara dengan tipe .WAV yang di-*import* kedalam Content Browser menjadi SoundWave. SoundCue ini dapat berasal dari lebih dari satu SoundWave dan memberikan efek-efek yang tersedia di dalam editor ini seperti efek *attenuation*, *random*, *looping* dan efek lainnya sehingga terbentuk bunyi baru yang siap digunakan di dalam peta3D. Gambar 2.9 adalah contoh penggunaan Unreal SoundCue Editor.



Gambar 2.9. penggunaan Unreal SoundCue Editor

2.4.9. SpeedTree Modeler dan SpeedTree Compiler

Ketika mengunduh Unreal Development Kit dari situs web resmi Unreal Engine (www.udk.com), disediakan pula perangkat lunak tambahan yaitu SpeedTree Modeler yang berfungsi untuk membuat objek tiga dimensi tumbuh-tumbuhan. Sedangkan SpeedTree Compiler digunakan untuk melakukan *compile* objek

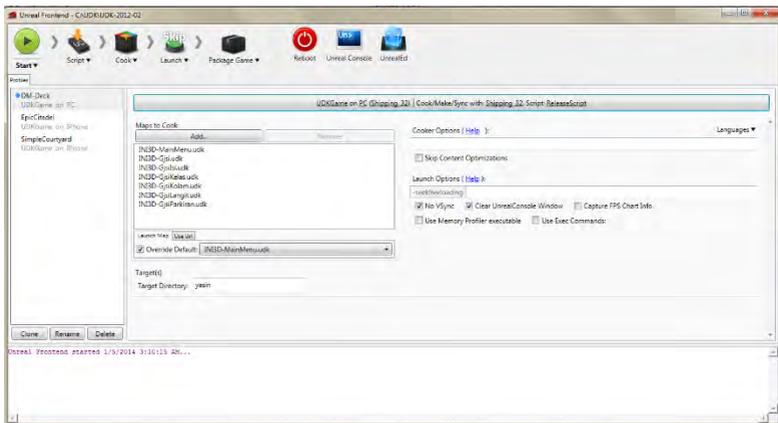
tiga dimensi tumbuh-tumbuhan hasil dari pemodelan yang dilakukan di SpeedTree Modeler.

2.4.10. UnrealFrontend

Unreal Development Kit juga menyediakan Unreal Frontend (UFE). UFE adalah sebuah alat yang menyediakan cara seragam untuk melakukan banyak tugas-tugas umum dalam ekosistem Unreal. Ini biasanya mencakup:

- Meluncurkan permainan
- Memulai server
- Menambahkan klien ke server untuk server lokal
- Menjalankan editor
- Kompilasi kode *script*
- *Cooking data*

Tanpa UFE banyak dari tugas-tugas ini akan membutuhkan file *batch* terpisah yang tidak perlu dan akan meningkatkan kompleksitas alur kerja sekaligus menciptakan sebuah mimpi buruk. Gambar 2.10 merupakan antarmuka pengguna dari UFE.



Gambar 2.10. Antarmuka pengguna Unreal Frontend

2.5. Perangkat Lunak Pembuat Peta 2D

Peta 2D dibuat sebelum membuat peta 3D. Bagian ini sedikit mengulas mengenai perangkat lunak yang digunakan untuk membuat peta 2D dalam pengerjaan tugas akhir ini.

- **AutoCAD Map 3D.** Sebuah perangkat lunak yang digunakan untuk membuat peta 2D yang dapat memuat informasi bangunan secara detail seperti *blueprint*.
- **Microsoft Office Visio.** Sebuah tools expansion dari Microsoft yang dapat digunakan untuk membangun sebuah diagram, chart ataupun denah ruang/lingkungan dalam konsep 2D. Perangkat lunak ini mampu menghasilkan peta 2D yang kemudian di-save dalam ke bentuk .PNG untuk dimasukkan pada Adobe Flash yang berfungsi sebagai penunjuk arah 2D.

2.6. Perangkat Lunak Modelling 3D

Perangkat lunak *modelling* 3D yang dimaksud disini adalah perangkat lunak untuk membuat objek tiga dimensi untuk nantinya dimasukkan dalam peta tiga dimensi yang telah dibuat. Perangkat lunak *modelling* 3D telah banyak tersedia dalam bentuk berbayar ataupun gratis. Berikut adalah sedikit ulasan mengenai perangkat lunak *modelling* tiga dimensi yang digunakan dalam pengerjaan tugas akhir ini.

- **Autodesk 3ds Max.** Sebuah perangkat lunak keluaran autodesk yang digunakan untuk melakukan *modelling* tiga dimensi, animasi, hingga *rendering*. 3D Studio Max dapat melakukan *export* hasil *modelling* yang dapat diterima oleh Unreal Development Kit seperti .ASE, .FBX dan .DAE.

2.7. Perangkat Lunak Pengolah Gambar

Pada pembuatan tugas akhir ini, dibutuhkan perangkat lunak pengolah gambar untuk membuat *material* dan *texture* 2D dari benda yang ada dalam peta. Untuk itu penulis menggunakan

perangkat lunak digital imaging yang banyak tersedia mulai dari yang berbayar hingga yang tidak berbayar. Berikut adalah sedikit ulasan mengenai program pengolah gambar yang digunakan dalam pengerjaan tugas akhir ini.

- **Adobe Photoshop.** Sebuah perangkat lunak keluaran dari Adobe System. Sering digunakan untuk melakukan editing gambar. Perangkat lunak yang berguna untuk memanipulasi gambar ini juga menyediakan fitur *image selection* yang cukup berguna saat memanipulasi gambar.

2.8. Perangkat Lunak Pengolah Suara

Untuk mengisi suara dari peta dibutuhkan perangkat lunak untuk merekam dan melakukan editing suara. Berikut sedikit ulasan mengenai perangkat lunak pengolah suara yang digunakan dalam pengerjaan tugas akhir ini.

- **Adobe Audition.** Aplikasi yang diproduksi oleh Adobe. Adobe Audition menyediakan fitur perekam, penyatuan, dan mengubah suara yang memungkinkan untuk memperbaiki kualitas suara, menambahkan berbagai efek, dan menggabungkan berbagai track atau file menjadi satu dan menyimpannya dalam berbagai format.
- **Audacity.** Aplikasi tidak berbayar yang digunakan untuk merekam dan mengubah suara.
- **Wavosaur.** Perangkat lunak ini tidak berbayar dan dapat digunakan untuk merekam dan melakukan editing suara.

2.9. Perangkat Lunak Pendukung

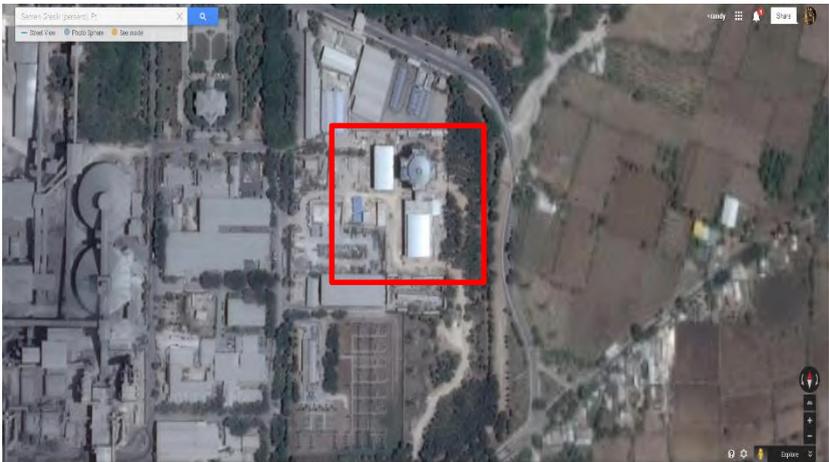
Beberapa perangkat lunak juga digunakan untuk membuat tampilan animasi flash, *video*. Berikut sedikit ulasan mengenai perangkat lunak pendukung yang digunakan dalam pengerjaan tugas akhir ini.

- **Adobe Flash CS4.** Unreal Engine mendukung penggunaan animasi flash di dalam membuat peta 3D. Dengan menggunakan Adobe Flash kita dapat membuat animasi flash dan memasang animasi flash tersebut ke dalam Unreal Engine. Penggunaan *script* di dalam animasi flash juga dapat menambah sebuah peta 3D Unreal Engine menjadi lebih interaktif, karena dapat menerima suatu input dari Unreal Engine dan sebaliknya menampilkan suatu output ke dalamnya.
- **Bink.** Bink merupakan *video codec* untuk game dan sudah mendapatkan lisensi lebih dari 5800 *game*, termasuk Unreal Engine. Dengan membuat *video* bertipe .BINK, maka *video* dapat dijalankan dalam Unreal Engine sebagai *video* pembuka yang menarik.

BAB III METODOLOGI

Pada bab ini dibahas mengenai objek penelitian serta metode penelitian bagaimana langkah-langkah penelitian dilakukan. Objek penelitian dijelaskan secara lebih jelas pada bab ini.

Objek yang digunakan sebagai penelitian tugas akhir ini adalah wilayah Gedung Utama PT.Semen Indonesia (Persero) Tbk. Wilayah Tuban seperti yang terlihat pada gambar 3.1. Dimana terdapat batasan-batasan wilayah yang digunakan pada objek penelitian tersebut. Hal penting yang menjadi perhatian utama dalam pengerjaan tugas akhir ini adalah interaksi yang terjadi di dalam wilayah tersebut oleh pengguna dengan benda-benda yang terdapat di dalamnya.



Gambar 3.1. Gedung Utama PT. Semen Indonesia (Persero) Tbk Wilayah Tuban (Google Map)

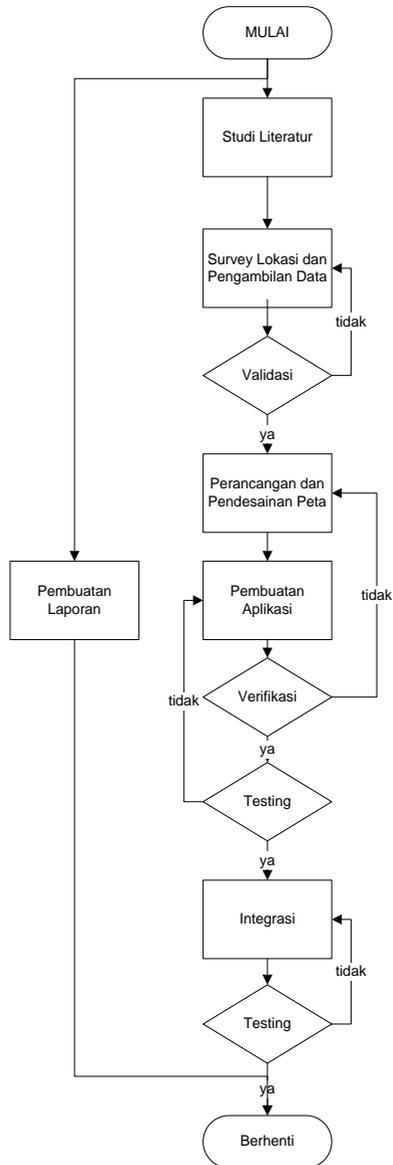
Bangunan Gedung Utama PT.Semen Indonesia (persero) Tbk Wilayah Tuban berada di Desa Sumberarum, Kerek Tuban.

Interaksi tambahan yang dimasukkan untuk tiap-tiap bangunan antara lain :

- Interaksi di Gedung Utama PT. Semen Indonesia (Persero) Tbk Wilayah Tuban :
 - Prosedur memasuki Gedung Utama
 - Video Profile Perusahaan dalam bahasa indonesia dan bahasa inggris di lobby perusahaan
 - Video Produk OPC, PPC, dan SBC
 - Video kumpulan rewards perusahaan
 - Prosedur peminjaman kendaraan perusahaan.

Metode atau tahapan penelitian merupakan hal yang sangat diperlukan dalam melakukan suatu penelitian, hal ini berlaku juga dalam pengerjaan tugas akhir. Metode diperlukan sebagai kerangka dan panduan proses pengerjaan tugas akhir, sehingga rangkaian pengerjaan tugas akhir dapat dilakukan secara terarah, teratur, dan sistematis.

Tahapan metodologi penelitian dalam penyusunan tugas akhir ini secara umum terdiri dari beberapa tahap yang dapat diilustrasikan dalam diagram alir Gambar 3.2. Penelitian diawali dengan melakukan studi literatur, kemudian dilanjutkan dengan standarisasi, survey lokasi dan pengambilan data. Adapun penjelasan langkah-langkah penelitian yang dilakukan yaitu sebagai berikut:



Gambar 3.2. Diagram Alir Metodologi

3.1. Studi Pendahuluan dan Literatur

Studi literatur yang dilakukan adalah pembelajaran dan pemahaman literatur yang berkaitan dengan permasalahan yang ada. Beberapa yang akan dipelajari seperti cara mengoperasikan Unreal Engine dan memanfaatkannya untuk membuat peta. Serta cara penggunaan perangkat lunak lainnya yang mendukung pengembangan aplikasi ini.

3.2. Survey Lokasi dan Pengambilan Data

Pada tahap ini akan dilakukan pengambilan data dan survey pada lokasi gedung, ini dilakukan dengan cara mengambil foto atau membuat video gedung serta obyek-obyek yang akan divisualisasikan ke dalam Unreal Engine. Selain obyek dan gedung, data lain yang diambil adalah data interaksi yang bisa dilakukan dengan obyek-obyek tersebut. Tujuan dilakukannya tahap ini adalah agar gedung dan obyek yang dihasilkan dapat sesuai dan akurat seperti aslinya.

3.3. Validasi Data Survey

Kegiatan ini berupa pengecekan hasil survey yang telah didapat yang berguna untuk meyakinkan bahwa data yang telah kita peroleh sudah sesuai dengan kondisi nyata dari area, gedung-gedung dan juga objek yang ada di Gedung Utama PT.Semen Indonesia (Persero) Tbk Wilayah Tuban.

3.4. Perancangan Desain Peta

Tahap ini berupa pembuatan desain dari peta untuk nantinya digunakan sebagai dasar dalam pembuatan aplikasi peta digital 3D ini. Pembuatan desain berupa rancangan-rancangan peta secara 2D.

3.5. Pembuatan Aplikasi

Tahapan ini merupakan tahapan inti dalam pengerjaan tugas akhir ini, yaitu pembuatan aplikasi. Pada tahap ini aplikasi mulai dibangun dengan menggunakan Unreal Development Kit (UDK) sesuai dengan dua rancangan desain yang telah dibuat sebelumnya. Hampir semua pekerjaan dilakukan menggunakan perangkat lunak ini. Mulai dari membuat geometri bangunan, hingga membuat animasi interaksi. Untuk membuat suatu objek, penulis menggunakan perangkat lunak 3D Studio Max. Pembuatan aplikasi terdiri dari beberapa tahapan di dalamnya, yaitu:

a) Pembuatan Peta 3D

Tahapan ini terdiri dari proses pembuatan Level Map yang mencakup pembuatan geometri bangunan, pemberian texture2D, pemberian material, pemberian terrain, pemberian FluidSurface, pemberian tanaman, pemberian tanda interaksi serta penggunaan aktor. Level Map merupakan peta tiga dimensi yang dibangun dengan UDK. Beberapa package disiapkan untuk memuat gambar tekstur, gambar normal, objek 3D, suara, animasi flash dan sumber daya lainnya. Membangun geometri bangunan dilakukan dengan memanfaatkan *tools CSG* pada UDK. Konsep geometri mencakup menambah dan mengurangi bagian geometri. Tekstur diberikan pada setiap permukaan geometri bangunan yang telah dibuat disesuaikan dengan kondisi nyata yang ada di Gedung Utama PT.Semen Indonesia (Persero) Tbk. Wilayah Tuban. Dalam UDK, tekstur dapat diatur skala, pola peletakan dan posisi peletakannya. Tekstur pertama kali dibuat dari gambar foto yang sesuai dengan kenyataan, kemudian dibuat *normal image*-nya dengan aplikasi xNormal. Kedua gambar tersebut kemudian dimasukkan (di-*import*) kedalam package yang tersedia. Kemudian kedua gambar tersebut dapat di-*edit* menjadi material yang siap ditempelkan pada permukaan geometri.

b) Pembuatan dan Peletakan Objek

Tahapan kedua dari pembuatan aplikasi ini adalah pembuatan objek-objek yang ada di dalam atau di luar gedung objek penelitian. Membuat objek tiga dimensi dilakukan dengan cara *modelling* menggunakan aplikasi 3D Studio Max dan disimpan kedalam format file yang dapat diterima oleh UDK seperti .FBX, .ASE atau .DAE yang diimpor ke dalam UDK. Objek tiga dimensi yang sudah siap, dapat di-*import* kedalam UDK dan dapat diatur material-nya, atau dalam istilah di kehidupan nyata lebih lazim disebut dengan tekstur, menggunakan Unreal Static Mesh Editor. Dalam UDK, objek-objek tiga dimensi seperti ini dikenal dengan sebutan *StaticMesh*. Khusus untuk membuat objek tumbuh-tumbuhan, aplikasi SpeedTree Modeler digunakan kemudian hasilnya dapat di-*import* kedalam UDK dan diolah menggunakan Unreal SpeedTree Editor. Objek aktor dapat dikembangkan dengan aplikasi *modelling* 3D untuk kemudian dibuat lengkap dengan animasi gerakan-gerakannya.

c) Penambahan Interaksi

Tahapan ketiga dari pembuatan aplikasi ini adalah menambahkan interaksi dan informasi terhadap objek-objek yang dapat diinteraksikan. Penambahan interaksi dapat dilakukan dengan langkah awal sebagai berikut:

- menentukan objek-objek yang dapat diinteraksikan,
- menambahkan informasi dari objek tersebut, dan
- menambahkan efek gerakan terhadap objek tersebut

Mengembangkan interaksi aktor dengan objek yang ada dalam peta dimungkinkan dengan editor yaitu *UnrelKismet*. Beberapa interaksi dimungkinkan dengan pemberian kontrol pada pengguna untuk berinteraksi. Untuk dapat melakukan hal tersebut, perlu dilakukan pengembangan kode program. Mengembangkan kode program dilakukan dengan bantuan aplikasi *UnCodeX* sehingga *default* aplikasi sesuai dengan *gameplay* yang direncanakan. *Packaging* aplikasi mencakup pembuatan menu-menu, *video* pembukaan.

d) Pengaturan Pencahayaan

Pengaturan pencahayaan dilakukan terhadap keseluruhan peta dan objek-objek peta tertentu. Pengaturan pencahayaan mencakup simulasi cahaya matahari dan cahaya lampu. pemilihan pencahayaan yang tepat dapat membuat peta 3D seperti nyata.

e) Penambahan Suara

Pengaturan lain yang perlu dilakukan adalah penambahan efek suara terhadap objek. Penambahan suara tidak jauh berbeda dengan penambahan objek, suara dalam bentuk .WAV dapat diimpor kedalam UDK dan kemudian dapat digunakan.

3.6. Verifikasi

Setelah melakukan perancangan maka pada tahap ini kita melakukan pengecekan/pembuktian apakah semua area, gedung dan objek sudah terpenuhi dan sesuai dengan rancangan peta 2D yang telah dibuat sebagai acuan pembangunan aplikasi ini.

3.7.Integrasi

Pada tahap ini dilakukan integrasi terhadap dua atau lebih peta 3D dengan standar yang sama. Yang dimaksud dengan standar yang sama adalah peta 3D yang sama-sama dibuat sebagai tugas akhir oleh tim dengan judul tugas akhir yang sama namun berbeda studi kasus. Integrasi bersifat tidak wajib dilaksanakan saat kondisi peta 3D yang lain tersebut belum dapat dikatakan sama standarnya atau belum terselesaikan. Integrasi antar peta tidak ada saling ketergantungan. Integrasi mencakup beberapa aspek sebagai berikut:

1. Lokasi peta 3D yang saling menyesuaikan
2. Aktor dan kontrol aktor yang sama
3. Menu utama dan video pembukaan yang sama

3.8. Pembuatan laporan

Pada tahap ini dilakukan penulisan setiap langkah-langkah pengerjaan dan pendokumentasian pengerjaan tugas akhir ini dari awal hingga akhir serta ditulis dalam format tugas akhir sehingga menghasilkan sebuah buku tugas akhir.

BAB IV

PERANCANGAN APLIKASI

Pada bab ini dijelaskan mengenai rancangan aplikasi yang dibangun pada tugas akhir ini. Desain sistem dibuat dengan mengacu pada dua jenis kebutuhan aplikasi, yaitu fungsionalitas dan non fungsionalitas. Kebutuhan fungsionalitas aplikasi didefinisikan sebagai berikut:

- Melihat Peta Tiga Dimensi (3D)
- Melihat Peta Dua Dimensi (2D)
- Navigasi
- Interaksi dengan Objek
- Mengubah Resolusi
- Interaksi mengenai aktivitas atau informasi khusus dari lokasi

Kebutuhan non-fungsional aplikasi didefinisikan sebagai berikut:

- Hardware
- Unreal Development Kit versi Februari 2012
- Aplikasi pendukung lain yang dibutuhkan

4.1. Interaksi

Dalam pengembangannya, direncanakan apa saja interaksi yang dapat dilakukan pengguna dengan objek-objek yang ada dalam peta tiga dimensi nantinya. Adapun interaksi yang dimasukkan sebagai berikut :

Tabel 4.1. Daftar Interaksi

No.	Interaksi	Deskripsi
1.	Membuka pintu	Pintu dapat terbuka
2.	Menutup pintu	Pintu dapat tertutup
3.	Menyalakan lampu	Lampu dapat menyala
4.	Mematikan lampu	Lampu dapat mati

5.	Informasi ruangan	Pada setiap ruang yang memiliki interaksi terdapat penjelasan singkat dari ruangan tersebut
6.	Prosedur memasuki Gedung Utama PT.Semen Indonesia (Persero) Tbk Wilayah Tuban	Simulasi ini dilakukan pada Di Gedung Utama, dengan urutan pengujian sebagai berikut : 1. Pengunjung selain karyawan dari PT.Semen Indonesia melapor kepada petugas resepsionis. 2. pihak resepsionis akan menanyakan keperluan kunjungan. 3. dan pengunjung diminta meninggalkan kartu identitas (KTP,SIM atau yang lainnya). 4. Kemudian pihak resepsinis akan memberikan kartu pengenalan atau kartu visitor.
7.	Video profile di lobby Gedung Utama PT.Semen Indonesia (Persero) Tbk Wilayah Tuban	Video ini memperlihatkan video profile dari PT.Semen Indonesia (Persero) tbk dalam 2 bahasa (bahasa Indonesia dan Bahasa Inggris).
8.	Video Produk OPC	Video ini memperlihatkan spesifikasi dari produk semen jenis OPC PT. Semen Indonesia (Persero) Tbk.

9.	Prosedur peminjaman kendaraan kantor	<p>Simulasi di lakukan di dalam Gedung Utama, dengan urutan pengujian sebagai berikut :</p> <ol style="list-style-type: none"> 1. Pegawai melakukan pengisian data melalui website jaringan intranet perusahaan. 2. Pegawai akan menuju kebagian divisi sarana umum untuk melakukan validasi bukti permintaan peminjaman. 3. Sebelum mengambil kendaraan, pegawai akan di minta menunjukkan bukti permintaan peminjamana yang sudah di validasi. 4. Pegawai di persilahkan mengambil kendaraan di halaman parkir Gedung utama.
10.	Video Produk PPC	Video ini memperlihatkan spesifikasi dari produk semen jenis PPC PT. Semen Indonesia (Persero) Tbk.
11.	Video Produk SBC	Video ini memperlihatkan spesifikasi dari produk semen jenis SBC PT. Semen Indonesia (Persero) Tbk.
12.	Video Rewards Perusahaan	Video ini memperlihatkan kumpulan – kumpulan raihan penghargaan yang pernah di raih oleh perusahaan.

13.	Penggunaan Lift	Fitur ini merupakan fitur untuk menggunakan lift
-----	-----------------	--

4.2.GUI Story Board

GUI Story Board memuat tampilan dan alur bagaimana aplikasi dijalankan. *GUI Story Board* dalam aplikasi ini memuat beberapa tampilan *static* dan tampilan peta tiga dimensi yang dinamis. Tampilan *static* berupa tampilan menu-menu yang disediakan untuk aplikasi.

Menu utama diawali oleh Menu Awal yang berisi dua menu yaitu menu Jelajahi Peta dan menu Keluar. Menu Jelajahi Peta yang dapat dilihat pada Gambar 4.2 berisi menu Pilihan Peta, menu Pilihan Resolusi, menu Bantuan dan menu Mulai. Menu Pilihan Peta yang dapat dilihat pada Gambar 4.3 berisi daftar peta 3D yang dapat dijelajahi oleh pengguna aplikasi.



Gambar 4.1. Tampilan Menu Awal



Gambar 4.2. Tampilan Menu Jelajahi Peta

Menu Pilihan Resolusi berisi pilihan resolusi yang telah disediakan yang terlihat pada Gambar 4.4. Menu Bantuan yang dapat dilihat pada Gambar 4.5 adalah menu bantuan pertama yang memuat informasi kontrol yang dipakai oleh pengguna aplikasi. Sedangkan menu bantuan selanjutnya memuat informasi penanda interaksi yang dapat dilihat pada Gambar 4.6.



Gambar 4.3. Tampilan Menu Pilihan Peta

Menu Keluar akan muncul jika tombol Keluar yang ada pada menu Awal dipilih. Gambar 4.7 menunjukkan tampilan menu Keluar.



Gambar 4.4. Tampilan Menu Resolusi



Gambar 4.5. Tampilan Menu Bantuan bagian kontrol



Gambar 4.6. Tampilan Menu Bantuan bagian penanda interaksi

Ketika dalam peta 3D, terdapat menu In-Game yang akan muncul jika pengguna menekan tombol Esc pada keyboard. menu In-Game berisi dua pilihan menu yaitu menu Kembali ke Peta dan Keluar. Tampilan menu In-Game dapat dilihat pada Gambar 4.8.

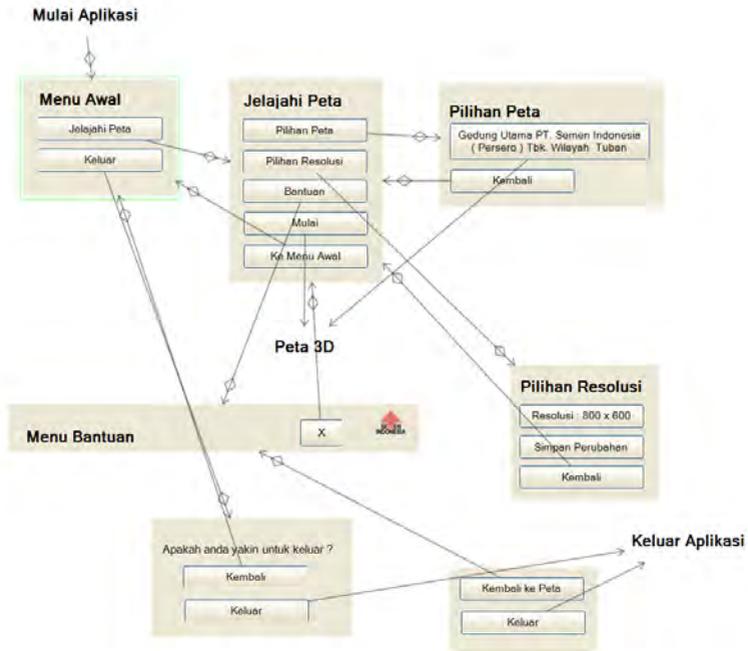


Gambar 4.7. Tampilan Menu Keluar



Gambar 4.8. Tampilan Menu In-Game

Dari tampilan-tampilan menu diatas kemudian ditambah dengan tampilan dinamis peta tiga dimensi, maka alur aplikasi dapat dibuat dengan menambahkan hubungan antar tampilan yang dapat diilustrasikan pada Gambar 4.9

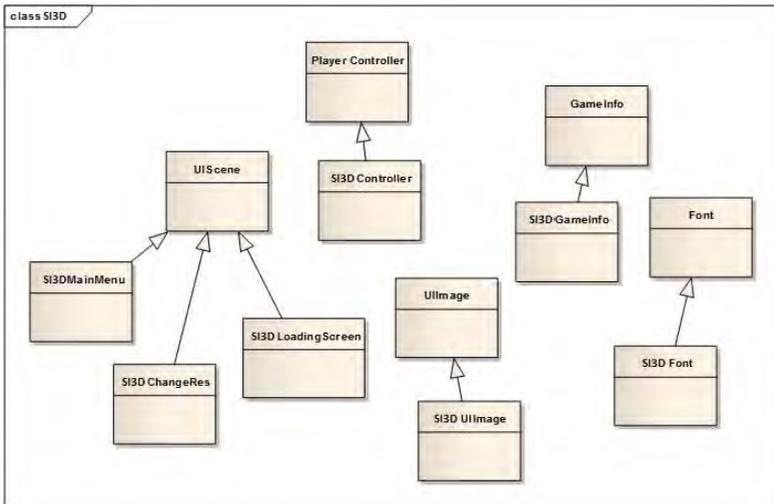


Gambar 4.9. GUI Story Board

4.3. Domain Model

Pada pengerjaan tugas akhir ini, pendefinisian *domain model* sangat penting karena *domain model* menggambarkan objek-objek utama yang akan digunakan. *Domain model* dapat berubah seiring dengan pengembangan desain dan aplikasi, sehingga objek-objek yang digambarkan pada *domain model* akan semakin lengkap dan akurat sesuai dengan alur sistem.

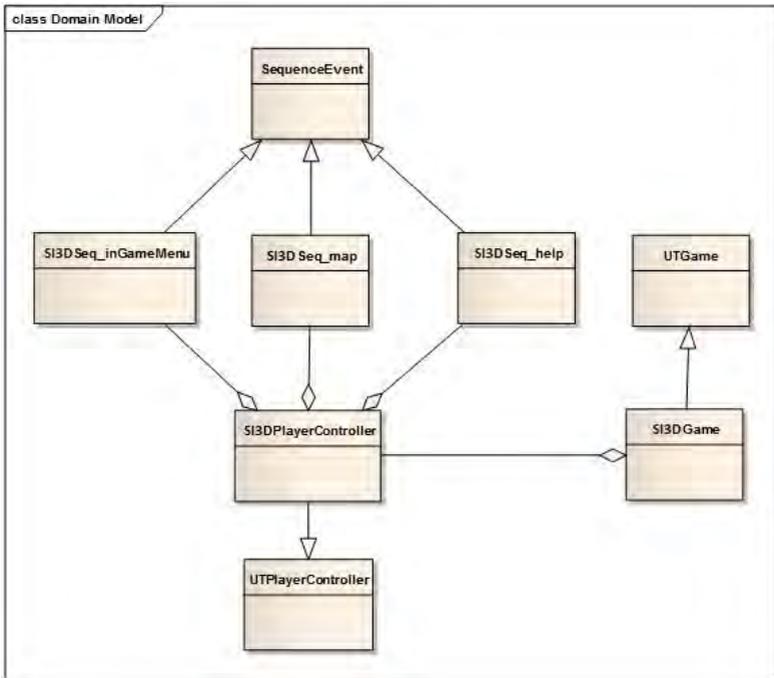
Model awal pada pengerjaan tugas akhir ini dapat dilihat pada Gambar 4.10. Terdapat beberapa objek hasil turunan domain model bawaan dari Unreal Engine.



Gambar 4.10. Domain model awal

Seiring dengan pengerjaan desain lainnya, didapati bahwa *domain model* berubah. Hal ini menghasilkan *domain model* baru yang dapat dilihat pada Gambar 4.11. Pada *domain model* baru, *domain model* bawaan dari unreal engine yang digunakan adalah *UTGame*, *SequenceEvent*, dan *UTPlayerController*.

Unreal Engine memiliki beberapa permainan bawaan yang bisa digunakan. Berdasarkan pertimbangan kemiripan navigasi, dan tipe permainan, maka diputuskan untuk menggunakan *UTGame* sebagai tipe permainan dari *project* ini. Pengaturan utama navigasi berada di *class* ini. Karena terdapat perbedaan navigasi, maka dibuat *class* baru bernama *Ini3DGame* dengan *parent class* *UTGame*.



Gambar 4.11. Domain model akhir

UTPlayerController adalah *class* bawaan unreal yang mengatur navigasi untuk tipe permainan UTGame. Karena terdapat beberapa navigasi tambahan, maka terdapat *class* baru dengan *parent class* UTPlayerController dengan nama Ini3DplayerController. Nantinya, *class* baru ini akan memanfaatkan class turunan dari sequence event di Kismet untuk menghubungkan unreal *script* dengan unreal Kismet. Sequence event digunakan untuk membuat *object* event di Kismet. Karena terdapat 3 tombol tambahan (Level Map, help, dan in game menu.) maka terdapat 3 *class* dengan *parent class* sequence event. Tiga *Class* tersebut adalah Ini3DSeq_Level Map, Ini3DSeq_help, dan Ini3DSeq_inGameMenu.

4.4. Use Case Diagram

Use case yang dirancang seharusnya dan sepatutnya memenuhi kebutuhan fungsionalitas yang telah disebutkan sebelumnya. *Use case* menggambarkan fungsionalitas yang dibangun pada tugas akhir ini sehingga berguna sebagai acuan dalam pemenuhan fungsionalitas pada pengembangan aplikasi.

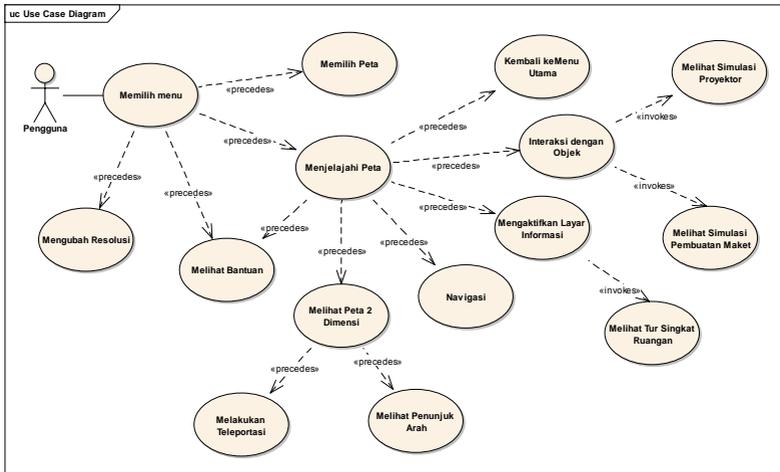
Pada *use case package* ini, terdapat rancangan pengembangan aplikasi dengan satu *actor* yaitu pengguna. Sebuah diagram dirancang dari komponen-komponen pada *use case package* yang memuat hubungan antar *use case* seperti yang terlihat pada Gambar 4.12. Rancangan deskripsi setiap *use case* dapat dilihat pada lampiran A.

4.5. Sequence Diagram

Sequence diagram memuat alur dalam *use case* dengan pendeskripsian yang mengarah pada pemrograman aplikasi. Sehingga sebelum merancang *sequence diagram* terlebih dahulu harus mengerti tentang teknologi yang akan diterapkan. Rancangan *sequence diagram* dapat dilihat pada lampiran B.

4.6. Test Case

Test case dirancang untuk menjaga performa aplikasi agar sesuai dengan desain yang dibuat. Dalam hal ini, *test case* akan dijalankan dengan beberapa skenario yang sesuai dengan rancangan pada diagram *use case*. Untuk rancangan skenario dan *test* dapat dilihat pada lampiran C. *Test case* nantinya akan diuji coba berupa *unit test*.

Gambar 4.12. Diagram *use case*

4.7. Analisa Pemilihan Tombol Navigasi dan Kontrol

Pemilihan tombol navigasi sangatlah penting dalam jalannya aplikasi peta tiga dimensi. Untuk itu penulis membuat analisa pemilihan tombol navigasi yang disesuaikan dengan kondisi umum mirip seperti permainan game tiga dimensi yang umum digunakan. Bagian ini mencakup berbagai kombinasi keyboard atau mouse yang dapat digunakan untuk mengontrol navigasi, fokus jendela, dan cara kerja aplikasi. Tabel 4.2 adalah tabel analisa pemilihan tombol navigasi.

Tabel 4.2. Analisa pemilihan tombol navigasi

N o	Perintah	Tombol	Hasil	Analisa
A	Navigasi			
1	Bergerak ke kiri	A	Menggerakkan tampilan sesuai dengan arah kiri	Umum dipakai pada permainan tiga dimensi

2	Bergerak ke kanan	D	Menggerakkan tampilan sesuai dengan arah kanan	Umum dipakai pada permainan tiga dimensi
3	Bergerak maju	W	Menggerakkan tampilan sesuai dengan arah depan	Umum dipakai pada permainan tiga dimensi
4	Bergerak mundur	S	Menggerakkan tampilan sesuai dengan arah belakang	Umum dipakai pada permainan tiga dimensi
5	Memutar searah jarum jam	panah kiri	Memutar tampilan searah jarum jam Peta akan berputar berlawanan dengan arah jarum jam	Umum dipakai pada permainan tiga dimensi
6	Memutar berlawanan arah jarum jam	panah kanan	Memutar tampilan berlawanan arah jarum jam	Umum dipakai pada permainan tiga dimensi
7	Bergerak maju	panah atas	Menggerakkan tampilan sesuai dengan arah tanda panah	Umum dipakai pada permainan tiga dimensi
8	Bergerak mundur	panah bawah	Menggerakkan tampilan sesuai dengan arah tanda panah	Umum dipakai pada permainan tiga dimensi
9	Membungkuk	C	Menggerakkan tampilan seakan aktor pengguna sedang membungkuk	Pada permainan tiga dimensi terdapat 2 pilihan umum, yaitu Ctrl atau C pada keyboard. Namun penulis

				memutuskan tombol C yang dipakai karena Umum dipakai pada permainan tiga dimensi
10	Melompat	Spasi	Menggerakkan tampilan seakan aktor pengguna sedang melompat	Umum dipakai pada permainan tiga dimensi
11	Berinteraksi dengan objek peta	Enter/Klik kiri	Menggerakkan tampilan sesuai dengan interaksi objek peta	Dalam permainan tiga dimensi dengan genre First Person Shooter (FPS) biasanya dipakai untuk menembak karena hal yang paling sering dilakukan pengguna adalah menembak. Begitu pula dalam aplikasi peta tiga dimensi ini, berinteraksi dengan objek adalah hal yang paling sering dilakukan oleh pengguna.
B	Kontrol tingkat peta			
1	Menu Bantuan	F1	Membuka menu Bantuan.	Umum dipakai pada permainan

			Tekan Tombol sekali lagi untuk keluar dari menu	
2	Menu In-Game	Esc	Membuka menu In-Game. Tekan Tombol sekali lagi untuk keluar dari menu	Umum dipakai pada permainan
3	Menu Peta 2D	M	Membuka menu Peta 2D. Tekan Tombol sekali lagi untuk keluar dari menu	Huruf M merepresentasikan kata Map/Peta yang juga umum dipakai pada permainan tiga dimensi untuk merepresentasikan masuk pada halaman peta dua dimensi

BAB V

IMPLEMENTASI DAN UJI COBA SISTEM

5.1. Lingkungan Implementasi

Aplikasi ini diimplementasikan pada komputer *client*. Spesifikasi lingkungan perangkat keras yang digunakan dalam pengembangan dan implementasi dapat dilihat pada Tabel 5.1.

Tabel 5.1. Spesifikasi perangkat keras dan sistem operasi untuk implementasi sistem

Spesifikasi
Prosesor: Intel® Core™ i5-3570 CPU @3.40GHz 3.80 GHz
Memori: 8 GB RAM
VGA: NVIDIA GeForce GTX 660 Ti 2805 MB
Sistem Operasi: Windows 7 Ultimate 64-bit (6.1, Build 7601)

Perangkat lunak utama yang digunakan adalah UDK. Perangkat lunak pendukungnya antara lain Autodesk 3ds Max, Adobe Flash, Audacity dan Adobe Photoshop. Tabel 5.2 berikut ini merangkum perangkat lunak yang digunakan dalam implementasi ini.

Tabel 5.2. Perangkat lunak yang digunakan

Teknologi	Versi
Editor	Unreal Development Kit 32bit (Februari'12)
3D Editor	Autodesk 3ds Max 2013
Animation Editor	Adobe Flash CS5
Sound Editor	Audacity
Texture Editor	Adobe Photoshop CS5

5.2. Peta Dua Dimensi

Area yang meliputi peta terdiri dari 7 lantai yang mana pada lantai 7 dari Gedung Utama PT.Semen Indonesia (Persero) Tbk. Wilayah Tuban pihak perusahaan melarang untuk menampilkan peta 2D untuk menjaga privasi, jadi yang hanya di tampilkan pada peta dua dimensi adalah sampai lantai 6. Bisa dilihat pada Gambar 5.1 sampai dengan Gambar 5.6.

5.3. Pembuatan Aplikasi

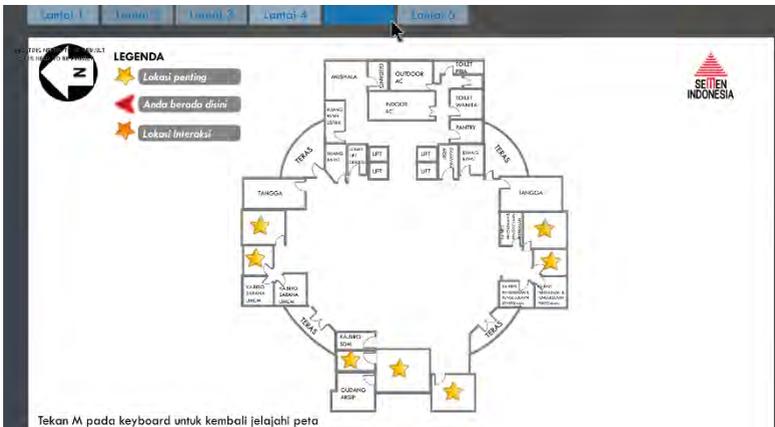
Dalam sub bab ini berisi penjelasan mengenai pembuatan aplikasi mulai dari pembuatan Level Map, pembuatan dan peletakan objek, penambahan interaksi, pengaturan pencahayaan, sampai dengan penambahan suara.

5.3.1. Pembuatan Level Map

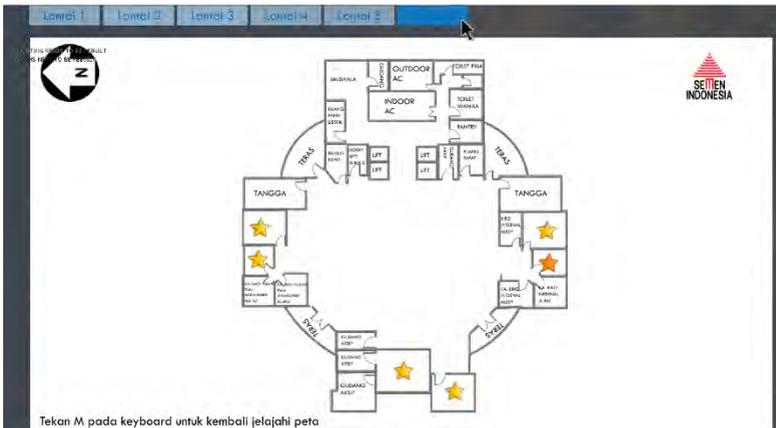
Langkah pertama dalam pembuatan aplikasi ini dimulai dengan pembuatan Level Map. Level Map ini dibangun dari data hasil survey primer dan sekunder, serta Level Map 2D yang telah dibuat sebelumnya pada tahap desain. Pembuatan Level Map ini mencakup pembuatan geometri dan pemberian material.

■ Pembangunan Geometri

Pembuatan Level Map dimulai dengan membuat geometri. Geometri pada UDK berfungsi untuk membentuk model tiga dimensi dari peta, yang mencakup bangunan dan semua permukaan bangunan peta, seperti tangga, tembok dan permukaan tanah/dasar gedung.



Gambar 5. 5 Peta dua dimensi Gedung Utama PT. Semen Indonesia (Persero) Tbk. Wilayah Tuban lantai 5



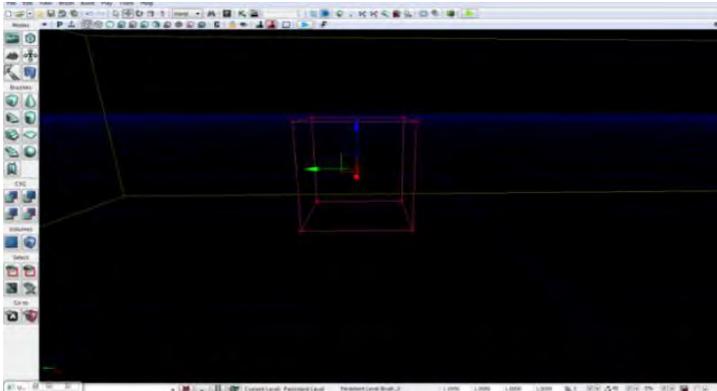
Gambar 5. 6 Peta dua dimensi Gedung Utama PT. Semen Indonesia (Persero) Tbk. Wilayah Tuban lantai 6

Geometri dalam UDK terdapat 2 mode utama ketika membuat sebuah `Level Map` baru, yaitu *subtract*, dan *additive*. Dalam membangun geometri `Level Map`, dalam UDK dikenal dengan istilah `brush`. `Brush` ini memiliki banyak bentuk, dalam *mode primitive* (bentuk `brush` umum yang disediakan oleh UDK).

Geometri ini berperan sebagai rangka atau *wireframe* untuk membangun model tiga dimensi gedung. Model geometri yang ada dalam UDK antara lain adalah bentuk *cube*, *cone*, *cylinder*, *curved staircase*, *linear staircase*, *sheet*, *tetrahedron*, dan *cards*. Bentuk yang digunakan dalam pembuatan `Level Map` ini adalah *cube* untuk membentuk permukaan dataran lantai dan tembok, *linear staircase* untuk membentuk tangga, dan *cylinder*. Selain terdapat model untuk membentuk geometri tersebut, dalam `properties brush` tersebut juga terdapat banyak modul `CSG`, yaitu `CSG add` dan `CSG subtract`. Untuk membentuk permukaan pejal, yang dipakai adalah `CSG add`, sedangkan untuk membuat rongga pada permukaan yang telah dibuat.

Geometri dalam UDK memiliki satuan, yaitu dalam bentuk satu satuan `unreal`. Dalam standarisasi, satu satuan meter dalam kondisi nyata disetarakan dalam 64 satuan unit `unreal`. Untuk membuat suatu gedung dalam geometri mode ini, proses yang dilakukan antara lain:

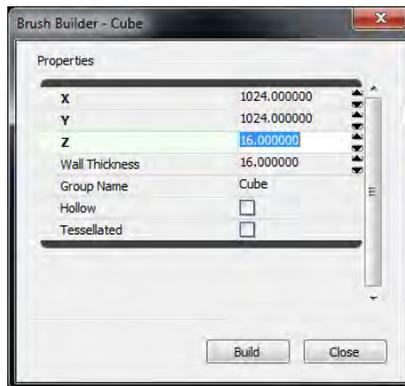
1. Menentukan `brush` yang akan digunakan
 Dalam menentukan `brush`, terlebih dahulu dipertimbangkan bentuk permukaan `Level Map` yang akan dibuat, yang didefinisikan dalam `red builder brush` dalam UDK. Apakah dalam bentuk lempengan yang direpresentasikan dalam bentuk *cube*, tangga dalam *staircase*, ataukah *cylinder*. `brush` untuk membangun geometri dalam `Unreal Editor` disebut dengan `Red Builder Brush` karena warna merah khasnya seperti yang terlihat pada Gambar 5.7 .



Gambar 5. 7 Red Builder Brush

2. Menentukan ukuran brush

Ukuran brush yang dibuat berdasarkan pada ukuran Level Map dari hasil survey, dan desain Level Map 2D yang kemudian dihitung berdasarkan skala standarisasi (1:64) dalam UDK. Untuk mengganti ukuran brush, bagian propertilah yang harus di-*edit* yang dapat dilihat pada 5.8



Gambar 5. 8 Properti Red Builder Brush

Ukuran ini akan disesuaikan dengan ukuran Level Map nyata, yang telah dikonversi ke skala unreal.

3. Menentukan jenis brush

Setelah ditentukan bentuk dan ukuran brush, selanjutnya adalah menentukan jenis brush. Dalam UDK terdapat tiga macam brush (red builder brush). Brush yang digunakan penulis dalam membangun geometri Level Map ini antara lain:

- CSG_add

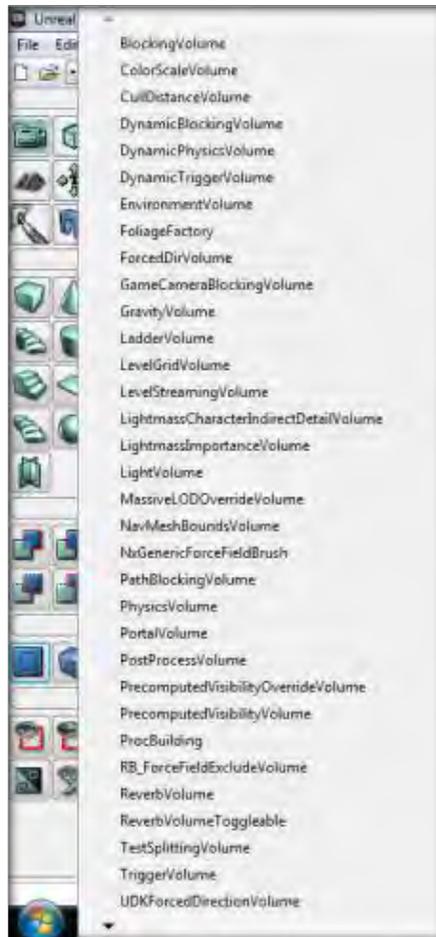
Merupakan brush pejal, dengan permukaan yang terisi penuh di dalamnya, bukan merupakan brush berongga. CSG add ini dipakai untuk membentuk *surface* atau permukaan dan tembok.

- CSG_substract

Merupakan brush yang digunakan untuk membuat rongga pada brush *add*, berfungsi untuk membuat rongga pintu dan jendela sebelum dipasang dengan objek.

- Special brush

Untuk membuat geometri yang dibutuhkan adalah CSG_add dan CSG *substract*, tetapi untuk menambahkan geometri selain kedua brush tersebut adalah special brush. Special brush yang paling sering dipakai antara lain adalah *LightmassVolume* dan *trigger volume*. Volume lain dapat juga dipilih melalui tombol *Add Volume* seperti yang terlihat pada Gambar 5.9.

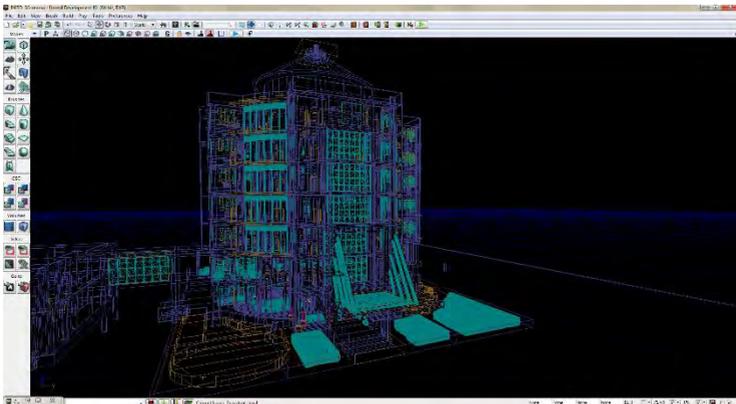


Gambar 5. 9 Pilihan volume yang ada pada Unreal Editor

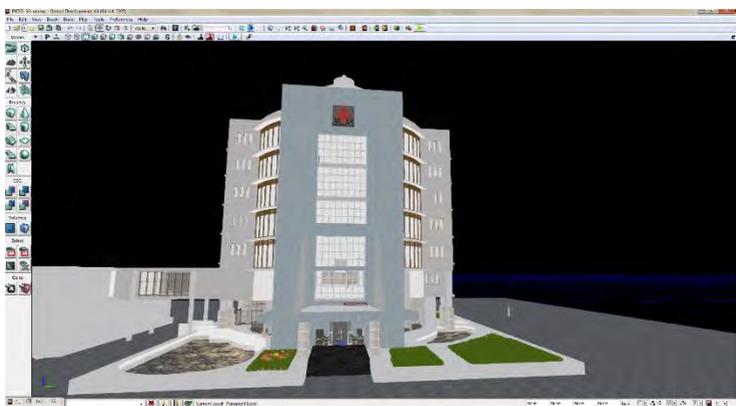
4. Geometry Build

Untuk melihat hasil brush yang di-add dalam geometri Level Map, harus dilakukan build geometry, sehingga akan terlihat geometry Level Map yang telah dibuat. Mode view

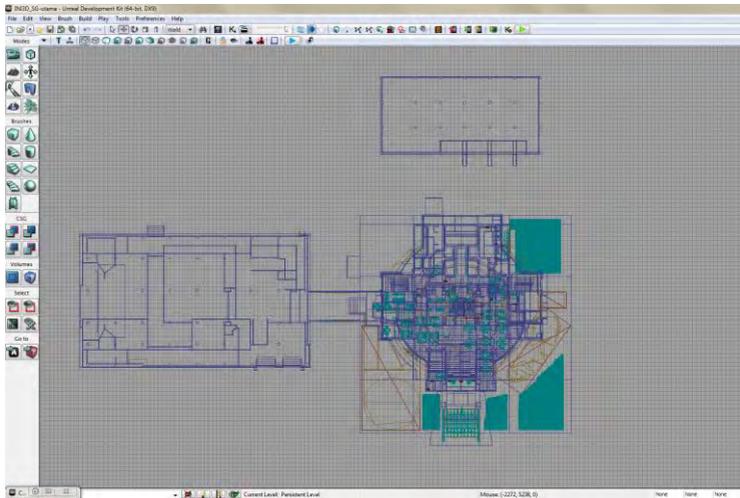
dalam UDK dapat diset ke dalam mode perspektif dan *wireframe*. Seperti yang terlihat pada Gambar 5.10, Gambar 5.11 dan Gambar 5.12.



Gambar 5. 10 Mode Wireframe Viewport Perspective Unreal Editor



Gambar 5. 11 Mode Unlit Viewport Perspective Unreal Editor



Gambar 5. 12 Mode Wireframe Viewport Top Unreal Editor

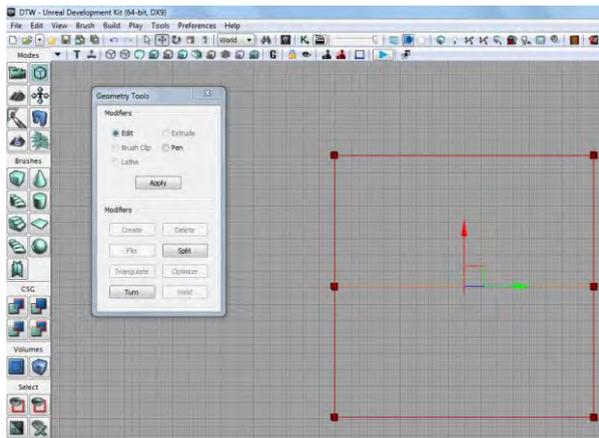
Dalam *geometry mode*, terdapat beberapa *modifier* yang digunakan penulis untuk, merubah brush yang telah dibentuk, yang ada dalam *geometry tools*. *Modifier* yang sering dipakai dalam pembangunan geometri Level Map ini adalah *modifier* split dan extrude. Penggunaan *geometry tools* dapat dilihat pada Gambar 5.13.

Untuk melakukan split sebuah brush, baik *red builder* brush maupun brush *CSG_add* atau *CSG_substract* dapat dilakukan dengan masuk ke *Geometry Mode* (*shift+2*), memilih sisi tegak lurus yang akan di-*split* dan memilih *split*. *Split* ini digunakan untuk memberikan material berbeda pada satu permukaan brush.



Gambar 5.13 Geometry tools

Gambar 5.14 merupakan pengaplikasian split pada red builder brush sebelum brush diatur sebagai CSG *add* atau CSG *subtract*.



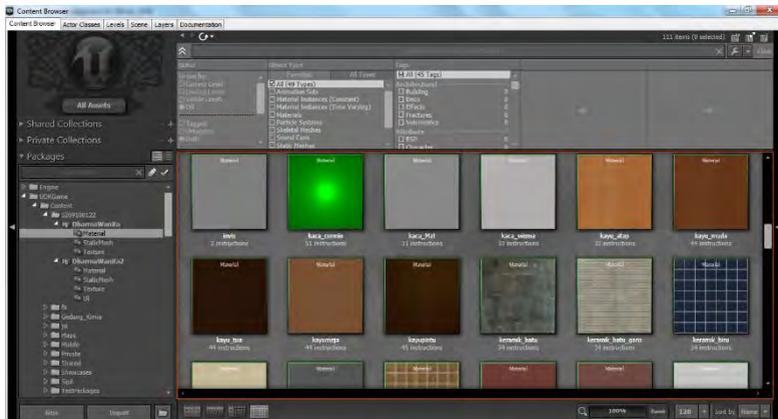
Gambar 5.14 Penggunaan split geometry tools

■ Pemberian Material

Pembuatan geometri merupakan bagian pembuatan aplikasi untuk membuat *wireframe* atau kerangka gedung. Kerangka gedung ini belum dilakukan penambahan atau pemasangan tekstur. Untuk dapat digunakan dan dipasang di dalam *brush* yang sudah dibuat dalam geometri sebelumnya, tekstur tersebut harus dibentuk menjadi *material*. *Material* ini dibuat dengan menggunakan Unreal Material Editor.

Material ini dapat disusun langsung dari Unreal Material Editor atau menggunakan tekstur yang sudah ada. Tekstur yang akan dipakai sebagai tekstur dalam *material* tersebut harus disimpan dalam *package* UDK. File *package* tersebut semua akan disimpan dalam format *.upk. *Package* tersebut disimpan dalam direktori UDK\UDKGame\Content. Setelah *package* dibuat, *package* tersebut dapat diisi dengan tekstur dan *material*.

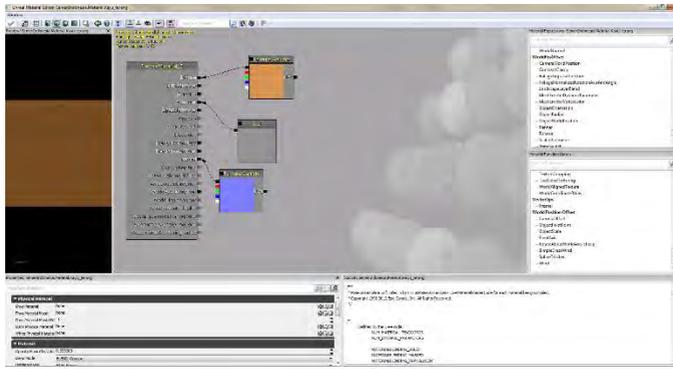
Untuk dapat disimpan atau di-*import* ke dalam *package* UDK, tekstur harus berukuran kelipatan kuadrat 2, seperti 64x64 pixel, 128x128 pixel, 256x256 pixel, 512x512 pixel, dan seterusnya. Tekstur yang dapat dipakai dalam *material*, dalam Unreal Material Editor terdapat di dalam *channel texture sample*. Tekstur ini digabungkan dalam *channel diffuse* dan *normal* untuk texture sampe normal yang di-*generate* dari gambar normal. Gambar 5.15 menunjukkan contoh tekstur yang telah di-*import* ke dalam *package*.



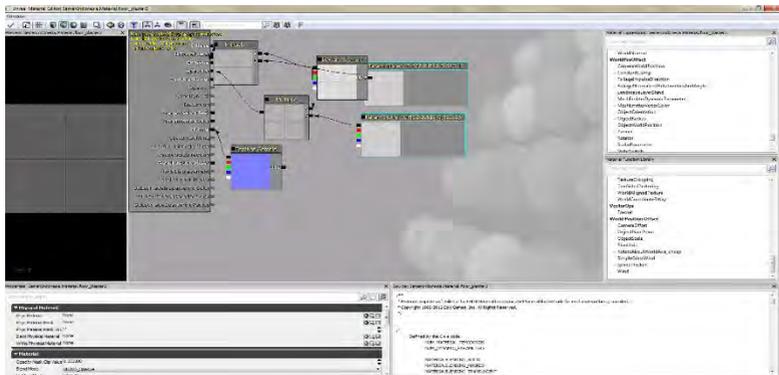
Gambar 5. 15 Contoh Material pada Content Browser

Tekstur yang di-*import* ini dapat ditambahkan pengaturan untuk warna dan koordinat dengan menambahkan channel multiply dan texture koordinat. Untuk membuat Material dasar atau material sederhana, material yang melibatkan hanya satu channel yaitu texture sample yang dihubungkan ke channel diffuse material. Gambar 5.16 menunjukkan contoh penggunaan Unreal Material Editor.

Untuk membuat material yang lebih kompleks, yaitu material yang memiliki permukaan dengan lebih tekstur dan pantulan tertentu dapat dibuat dengan memanfaatkan multiple channel yang ada dalam Unreal Material Editor. Untuk pantulan cahaya, channel yang dipakai adalah specular dengan nilai diatur sesuai dengan kebutuhan. Gambar 5.17 merupakan contoh untuk material yang mengaplikasikan multiple channel.

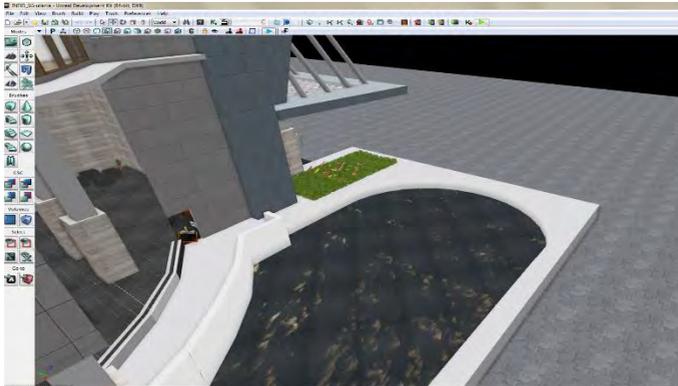


Gambar 5. 16 Penggunaan Unreal Material Editor



Gambar 5. 17 Penggunaan multiple channel pada Unreal Material Editor untuk material tembok

Setelah material dibuat dan disimpan dalam *package*, material tersebut yang akan diaplikasikan dalam brush dan objek. Untuk menambahkan material pada permukaan brush, dilakukan dengan *apply* material yang terpilih pada permukaan brush. Pemasangan material pada masing-masing brush ditunjukkan pada Gambar 5.18.

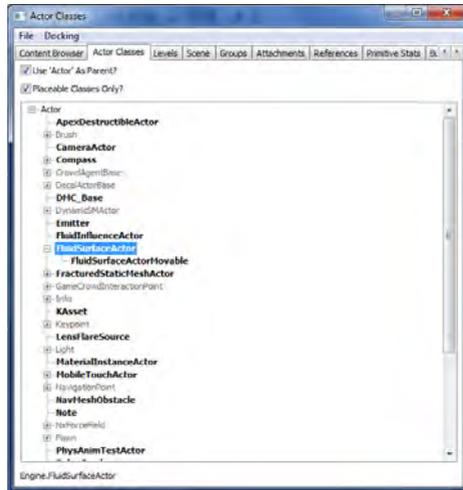


Gambar 5. 18 Hasil pemberian material

■ Pemberian FluidSurface

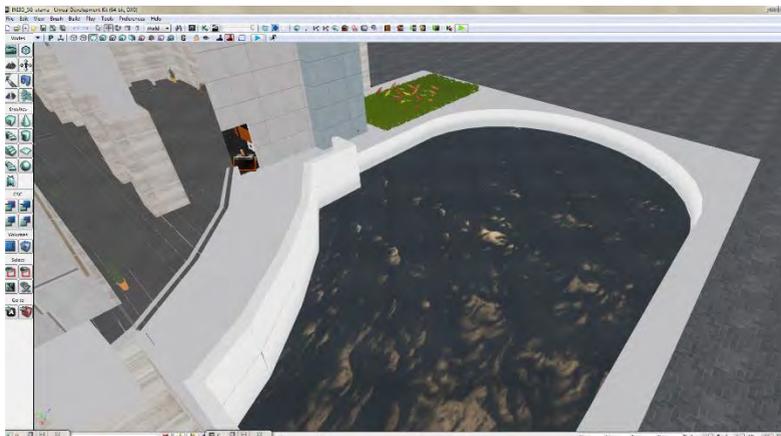
Di dalam Unreal Engine, tersedia FluidSurface yang berada di Actor Classes seperti pada Gambar 5.19. FluidSurface ini digunakan untuk menggambarkan kolam dan selokan air.

Dengan FluidSurface, dapat dibuat suatu riakan cairan di dalam Level Map yang dibuat untuk merepresentasikan kolam air. Bentuk dari FluidSurface ini berupa segi empat dengan ketinggian nol, sehingga untuk menggunakan Actor ini, sebelumnya harus dibuat suatu lubang pada Level Map tersebut, kemudian meletakkan Actor ini di lubang tersebut, dengan ketinggian yang diinginkan.



Gambar 5. 19 FluidSurfaceActor

Setelah menambahkan FluidSurfaceActor ke dalam Level Map, selanjutnya melakukan pemberian material yang sudah disediakan pada Actor tersebut.. Actor ini digunakan untuk memberikan efek air pada saluran air dan kolam yang dapat dilihat contohnya untuk kolam pada Gambar 5.20.



Gambar 5. 20 Kolam air menggunakan FluidSurface

■ Pemberian Tanda Interaksi

Dengan menggunakan Unreal Cascade, suatu particle system dapat dibuat. Pembuatan particle system ini tidak lepas dari pemanfaatan actor yang ada, Emitter yang diinginkan, pemberian material, dan juga memanfaatkan StaticMesh. Particle system yang dibuat ada dua buah yaitu untuk penanda interaksi informasi ruangan dan penanda interaksi objek. Gambar 5.21 menunjukkan penanda interaksi informasi objek, sedangkan Gambar 5.22 menunjukkan penanda interaksi ruangan.



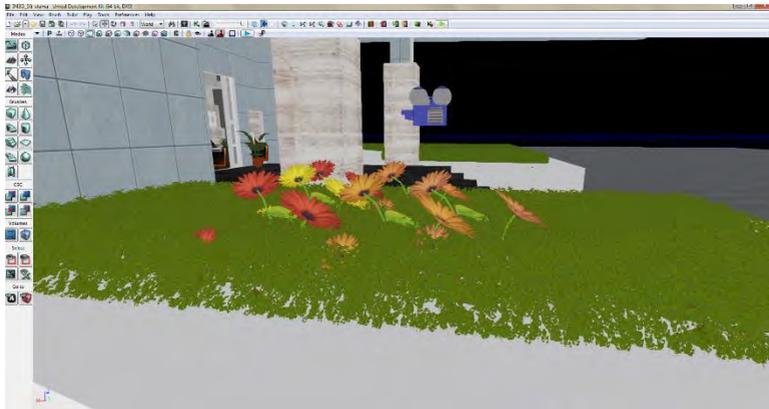
Gambar 5. 21 Particle System untuk informasi interaksi



Gambar 5. 22 Particle System untuk interaksi informasi ruangan

■ Pemberian Tanaman dan Pepohonan

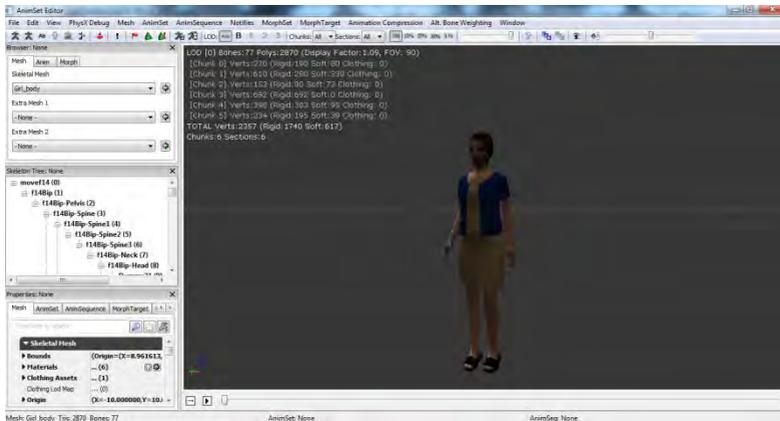
Supaya peta 3D terlihat lebih hidup, maka perlu diberikan tanaman dan pepohonan sebagai makhluk hidup yang terdapat pada kondisi nyata. Hal tersebut dapat dilakukan melalui pemberian *StaticMesh* atau dapat menggunakan objek tanaman yang dibuat melalui *SpeedTree Modeler* dan *SpeedTree Compiler*. Pemberian dengan hasil dari *SpeedTree* lebih terlihat hidup karena hasil tersebut mirip dengan pohon asli dari kondisi nyata dan terdapat efek bergerak terkena hembusan angin, berbeda dengan *StaticMesh*. Gambar 5.23 menunjukkan tanaman dan pepohonan hasil *SpeedTree* dan *StaticMesh*.



Gambar 5. 23 Tanaman hasil SpeedTree dan Staticmesh

Penentuan Karakter Aktor

Unreal AnimSet merupakan salah satu bagian dari Unreal Engine yang digunakan untuk mengatur aktor yang akan digunakan di dalam Level Map kita. Beberapa pengaturan yang umum dilakukan, meliputi material aktor, AnimSet aktor serta lokasi dan rotasi aktor. dengan memanfaatkan fungsi pengaturan material, kita bisa mempercantik tampilan aktor, seperti memberikan warna baju, kulit, dan lainnya. Karakter yang digunakan pada aplikasi INI3D adalah seorang karakter wanita seperti yang dapat dilihat pada Gambar 5.24.



Gambar 5. 24 Aktor wanita

5.3.2. Pembuatan dan Peletakan Objek

Pembuatan dan peletakan objek dilakukan dengan beberapa langkah yang runut. Dimulai dari pembuatan *object* 3D terlebih dahulu kemudian dilakukan beberapa hal yang harus diatur agar objek tersebut dapat diletakkan pada peta 3D.

5.3.2.1 Pembuatan Objek 3D

Dalam membuat suatu *object* 3D menggunakan aplikasi Autodesk 3ds Max. Proses pembuatan *object* tersebut meliputi:

a) Pembuatan *Object* 3D

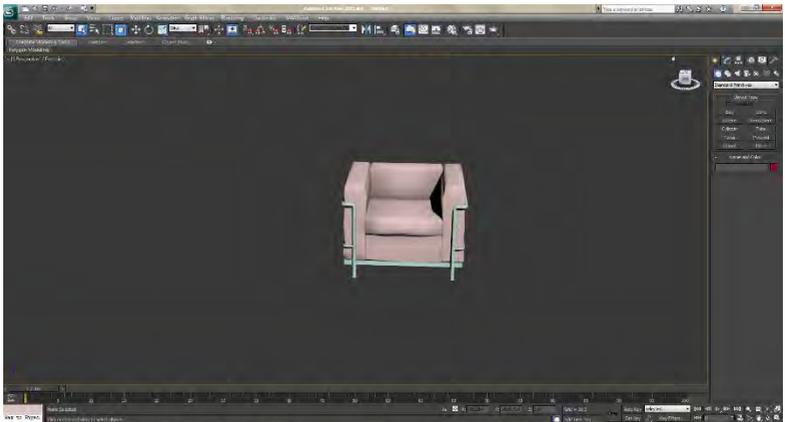
Di dalam Autodesk 3ds Max, suatu *object* memiliki Ukuran tinggi, lebar, dan panjang dengan masing-masing jumlah segmennya. Selain itu suatu *object* juga memiliki posisi x, y, dan z. *Object* di dalam Autodesk 3ds Max juga memiliki satu material *default*.

Untuk mempermudah pengaturan kita terhadap suatu *object*, kita bisa melihat *object* dalam banyak persepective yaitu, dari

atas, bawah, samping kanan, samping kiri, depan, belakang, perspective, orthogonal, dll.

b) Pemberian Material pada *Object* 3D

Pada Autodesk 3ds Max, kita memanfaatkan *Modifier Tools*, dimana hanya beberapa *Modifier List* yang bisa digunakan untuk mengatur material *object*. Penulis menggunakan Edit Poly atau Edit Mesh untuk mengimputkan id material *object* tersebut. Kemudian menggunakan Material Editor untuk memberikan material pada *Object* Tersebut. Material Editor berfungsi sebagai index untuk tiap satuan warna yang nantinya akan diberikan material di UDK. Gambar 5.25 menunjukkan pemberian material pada object.



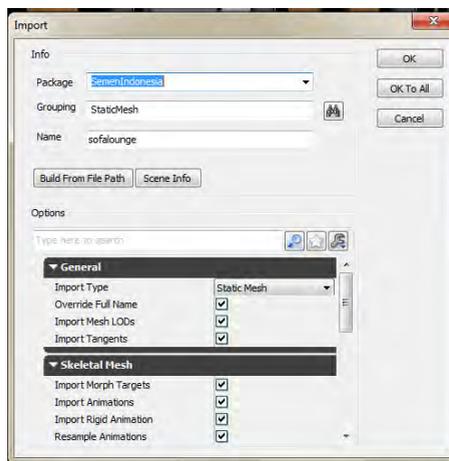
Gambar 5. 25 Pemberian material Editor pada Object

c) Export *Object 3D*

Setelah *object* selesai dibuat, maka *object* tersebut di Export. Banyak sekali tipe yang tersedia di sini. Namun yang umum digunakan. Salah satunya .ASE jika *object* tersebut hanya memiliki satu jenis *material*. Selain itu juga ada .FBX dan .DAE jika memiliki lebih dari satu jenis *material*, dengan kelebihan dan kekurangan masing-masing. Ketiga tipe tersebut yang dapat digunakan di Unreal Engine.

5.3.2.2 Import Objek 3D

Setelah *object 3D* dibuat dengan menggunakan Autodesk 3Ds Max, export objek dengan format .FBX. Harus dipastikan saat meng*import object*, tipe yang dipilih *StaticMesh*. Didalam content browser, agar *object* yang sudah di-*import* tertata rapi, maka dibuatkan package, dan atau *group* untuk *object* tersebut. Gambar 5.26 menunjukkan *options import* ke dalam Unreal Editor.

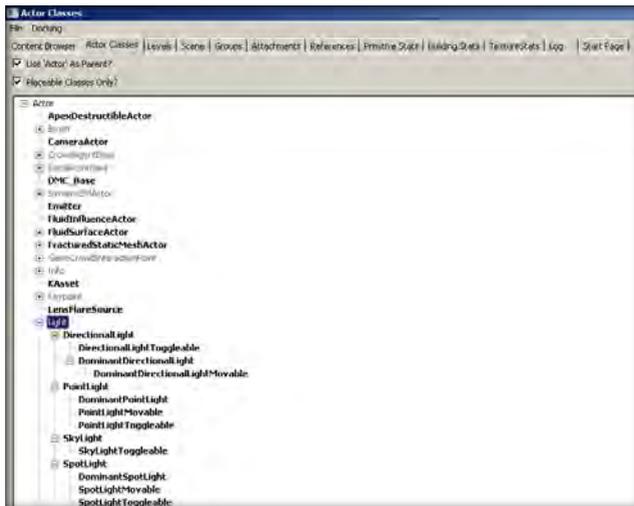


Gambar 5. 26 Pilihan pengaturan import StaticMesh

Di dalam content browser, *object* tersebut bisa diberikan material yang sudah tersedia. Banyak material yang bisa dimasukkan, sesuai dengan *id* material yang sudah dibuat di sebelumnya.

5.3.3. Pengaturan Pencahayaan

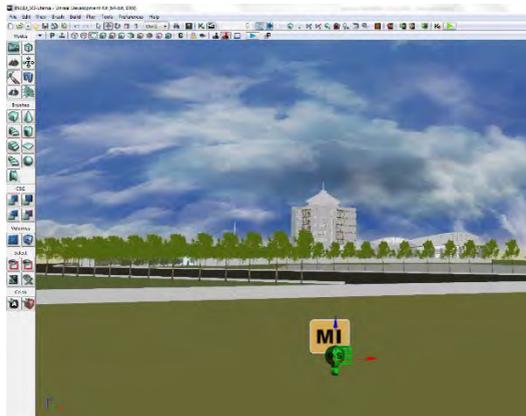
Pengaturan pencahayaan dalam aplikasi ini dimaksudkan untuk membuat keadaan peta mirip dengan keadaan nyata. Pengaturan cahaya di UDK dilakukan melalui beberapa kelas aktor light (cahaya). Seperti pada Gambar 5.27 terdapat beberapa jenis kelas aktor light yang ada, yaitu `DirectionalLight`, `PointLight`, `SkyLight`, dan `SpotLight`.



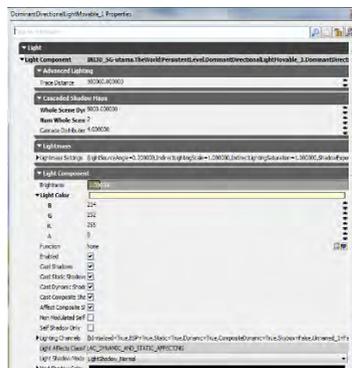
Gambar 5. 27 Macam-macam Aktor Light pada Unreal Editor

Setiap elemen aktor Light memiliki *properties* yang dapat dikonfigurasi sesuai kebutuhan. Komponen *properties* yang akan dikonfigurasi sesuai kebutuhan adalah *brightness*, *light color*, dan *radius*. Berdasar pada keadaan yang sebenarnya, dalam aplikasi ini hanya menggunakan `DominantDirectionalLight` dan `Point`

Light. `DominantDirectionalLight` memiliki sifat cahaya yang lurus untuk menerangi bagian peta tertentu dan memiliki efek pencahayaan yang mirip dengan matahari dapat dilihat pada Gambar 5.28. Sehingga tentunya aktor `DominantDirectionalLight` digunakan untuk merepresentasikan matahari. Dalam merepresentasikan matahari perlu ada konfigurasi pada *properties* pada aktor `DominantDirectionalLight` tersebut seperti pada Gambar 5.29.



Gambar 5. 28 Penggunaan `DominantDirectionalLight` pada peta 3D



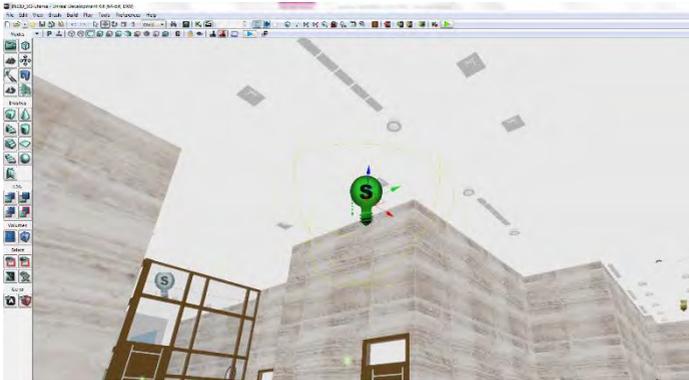
Gambar 5. 29 Konfigurasi `DominantDirectionalLight`

Seperti pada Gambar 5.30, peta sudah terlihat terang oleh matahari namun terdapat bagian dari peta yang tidak tersinari cahaya. Oleh karena itu, `PointLight` digunakan memberikan cahaya untuk bagian-bagian yang tidak tersinari cahaya `DominantDirectionalLight`.



Gambar 5. 30 Penggunaan PointLight di luar ruangan

Untuk di dalam ruangan, `PointLightToggleable` diatur untuk merepresentasikan lampu ruangan. Sehingga `PointLightToggleable` akan bercahaya menyerupai lampu pada ruangan seperti pada Gambar 5.311. Aktor `PointLightToggleable` digunakan dikarenakan memiliki sifat yang dapat diatur kapan pencahayaan muncul dan kapan tidak muncul melalui `Kismet` sehingga dapat diatur seperti lampu ruangan.



Gambar 5. 31 Penggunaan PointLightToggleable di dalam ruangan

5.3.4. Penambahan Suara

Untuk menambahkan suara pada Unreal Editor, dibutuhkan file suara yang mempunyai format .WAV. File yang mempunyai format .WAV tersebut kemudian di-*import* ke dalam content browser. File hasil *import* tersebut berubah menjadi SoundNodeWave. Contoh file SoundNodeWave yang sudah berhasil di-*import*.

Supaya dapat digunakan untuk dimasukkan ke dalam Unreal Kismet atau Unreal Matinee, maka dibutuhkan SoundCue. SoundCue merupakan gabungan dari SoundNodeWave. Contoh file SoundCue dapat dilihat pada Gambar 5.32.

5.3.5. Pembuatan Interaksi

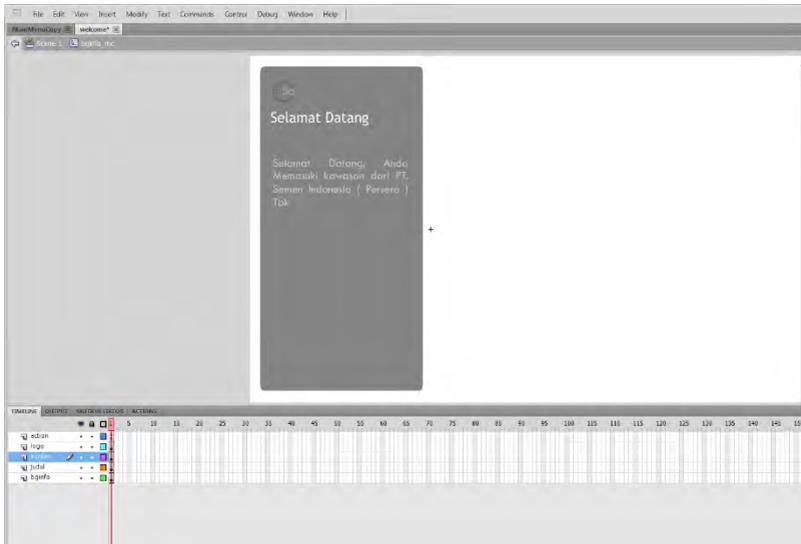
Secara garis besar, pembuatan interaksi menggunakan tampilan animasi flash dan tampilan *Matinee* dan diatur di *UnrealKismet*. Interaksi dapat berupa tampilan animasi flash, tampilan *Matinee* atau gabungan antara keduanya. Namun, interaksi banyak didominasi oleh tampilan animasi flash. Interaksi dibedakan menjadi tiga ranah utama yaitu interaksi layar informasi, interaksi informasi objek dan interaksi peta 2D.

Interaksi dengan tampilan animasi flash perlu terlebih dahulu membuat file dengan tipe SWF, tipe file yang dapat digunakan oleh *Unreal Editor*. File SWF dibuat dengan aplikasi pengolah animasi dan dalam tugas akhir ini digunakan aplikasi pengolah animasi *Adobe Flash CS5*. Aplikasi tersebut dapat membuat file FLA yang merupakan file proyek animasi flash dan file SWF yang merupakan file animasi flash.

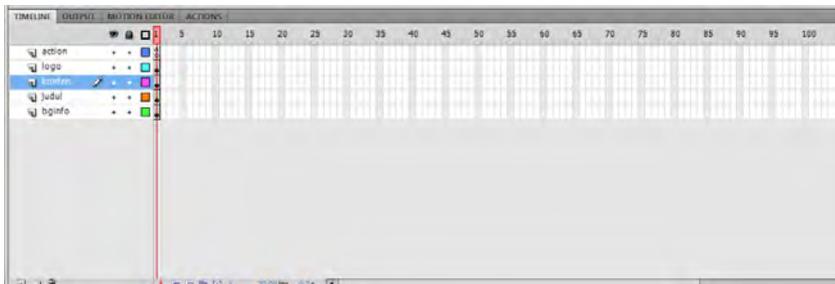
5.3.5.1 Layar Informasi

Layar informasi adalah interaksi tampilan animasi flash yang memuat informasi suatu tempat. Layar informasi muncul setiap aktor melewati tempat-tempat penting yang perlu diketahui oleh pengguna, misalnya ruang seminar, ruang ketua jurusan atau ruang dekan fakultas. Tempat-tempat penting ini dapat dilihat oleh pengguna melalui Menu Peta Dua Dimensi. Layar informasi akan menghilang dengan sendiri setiap aktor memasuki tempat tersebut atau menjau dari tempat tersebut.

Informasi mengenai tempat tersebut akan muncul di samping kiri layar Peta 3D yang sedang aktif. Animasi yang diusung adalah transisi transparansi atau beningnya gambar grafik. Gambar 5.34 menunjukkan bagaimana layar informasi diletakkan berdasarkan layar yang sedang aktif, sedangkan animasi transisi yang diusung ditunjukkan pada Gambar 5.35.

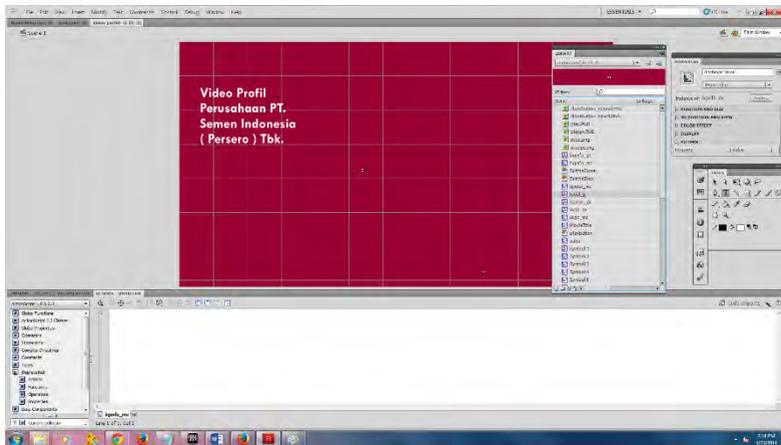


Gambar 5. 34 Animasi Flash Layar Informasi

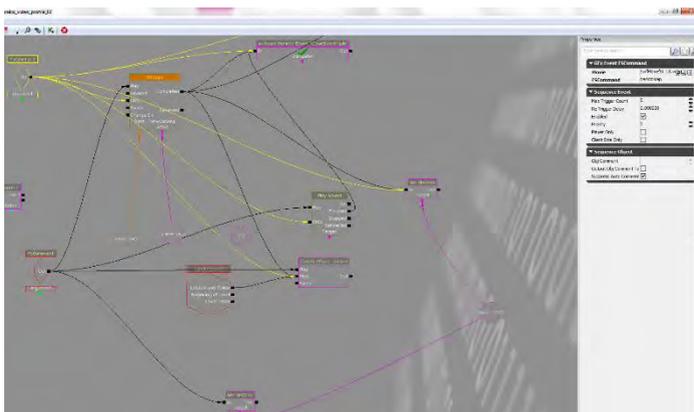


Gambar 5. 35 Alur Animasi Flash Layar Informasi

Terdapat ActionScript yang dijalankan pada layer action frame. ActionScript tersebut menghentikan jalannya animasi flash ketika animasi transisi transparansi telah dilakukan. Potongan kode ActionScript tersebut dapat dilihat pada Gambar 5.36. Sedangkan kismet-nya pada Unreal Editor dapat dilihat pada Gambar 5.37.



Gambar 5. 36 ActionScript pada Animasi Flash Layar Informasi



Gambar 5. 37 Kismet Interaksi Video Profil Perusahaan

5.3.5.2 Informasi Objek

Interaksi informasi objek merupakan interaksi dengan gabungan antara tampilan animasi flash dan tampilan *Matinee*. Terdapat berbagai interaksi informasi objek seperti menyalakan lampu, memadamkan lampu, membuka pintu, menutup pintu, simulasi maket, peminjaman buku, dan lainnya.

- **Menyalakan dan memadamkan lampu**

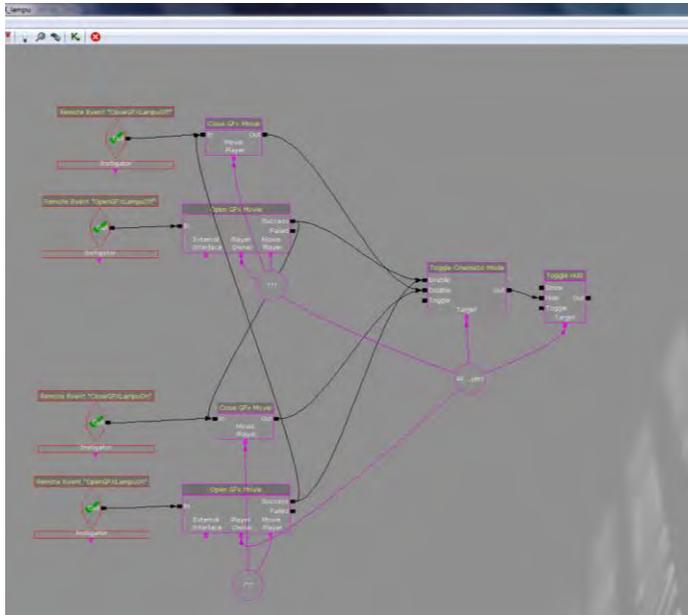
Beberapa interaksi merupakan suatu kesatuan interaksi seperti misalnya menyalakan dan memadamkan lampu. Kedua interaksi tersebut memerlukan dua tampilan animasi flash yang berbeda, namun menggunakan objek lampu *StaticMesh* yang sama dengan tipe objek *InterpActor* di *Level Map*. Pencahayaannya pun menggunakan *ToggleableLight* yang sama sehingga erat sekali hubungan keduanya pada *UnrealKismet*.

Untuk membuat *Kismet* yang dapat mengatur lampu bisa menyala atau padam, konsepnya adalah sebagai berikut. Kondisi awal adalah lampu menyala, kemudian ketika aktor berada di daerah jangkauan *Trigger* saklar lampu, maka akan muncul tampilan animasi flash. Animasi flash tersebut memuat informasi bahwa pengguna dapat berinteraksi dengan saklar tersebut dan akan memberikan efek lampu padam.

Dua animasi flash yang dibutuhkan untuk interaksi informasi objek menyalakan lampu dan memadamkan lampu sebagian besar adalah sama, yang berbeda hanya pada tulisan informasi yang ditampilkan pada pengguna.

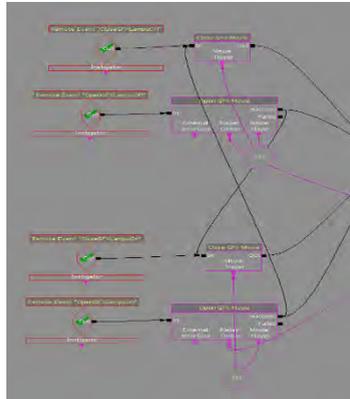
Hal ini dapat terjadi dan didukung oleh *UnrealKismet* dengan objek *Kismet RemoteEvent*. Cara kerja *RemoteEvent* adalah menghubungkan dua objek *Kismet* yang berada saling berjauhan. *RemoteEvent* dipakai untuk

menampilkan atau menutup animasi flash yang diinginkan ketika dibutuhkan di banyak tempat. Gambar 5.38 adalah RemoteEvent pada Kismet yang dapat dipanggil sesuai kebutuhan untuk menampilkan atau menutup animasi flash tertentu.



Gambar 5. 38 RemoteEvent pada Interaksi Lampu

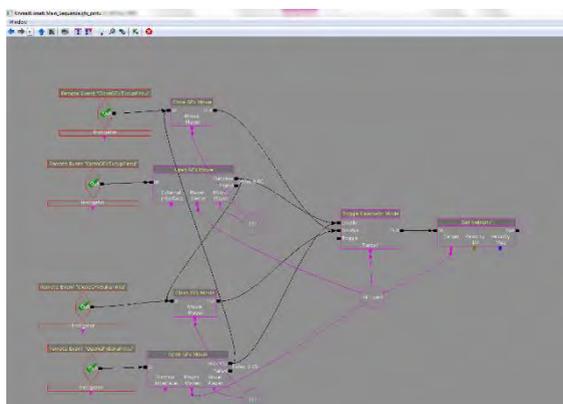
Setelah terdapat RemoteEvent yang menyediakan Kismet untuk menampilkan atau menutup animasi *flash* tertentu, maka pada setiap Trigger saklar yang ada, Kismet dapat dibuat dan Remote Event dapat dipanggil sesuai kebutuhan dengan menggunakan Activate Remote Event. Gambar 5.39 memperlihatkan penggunaan ActivateRemoteEvent.



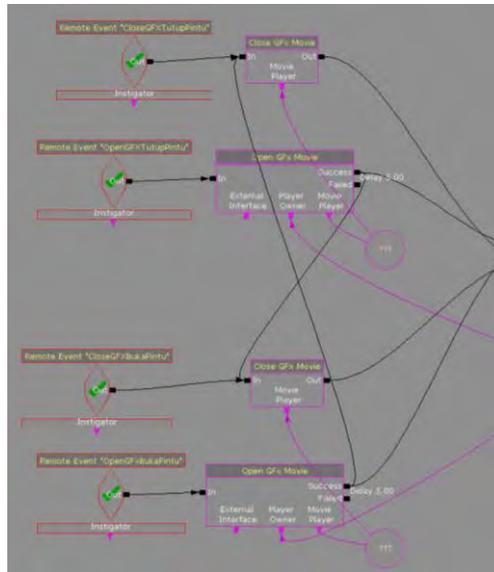
Gambar 5. 39 RemoteEvent pada Kismet interaksi menyalaikan dan memadamkan lampu

- **Membuka dan menutup pintu**

Konsep interaksi pintu sama dengan interaksi lampu yaitu dengan penggunaan flash sebagai penanda area jangkauan interaksi, RemoteEvent dan ActivateRemoteEvent. Gambar 5.40 memperlihatkan penggunaan ActivateRemoteEvent pada interaksi pintu. Sedangkan untuk penggunaan ActivateRemoteEvent pada interaksi pintu dapat dilihat pada Gambar 5.41.



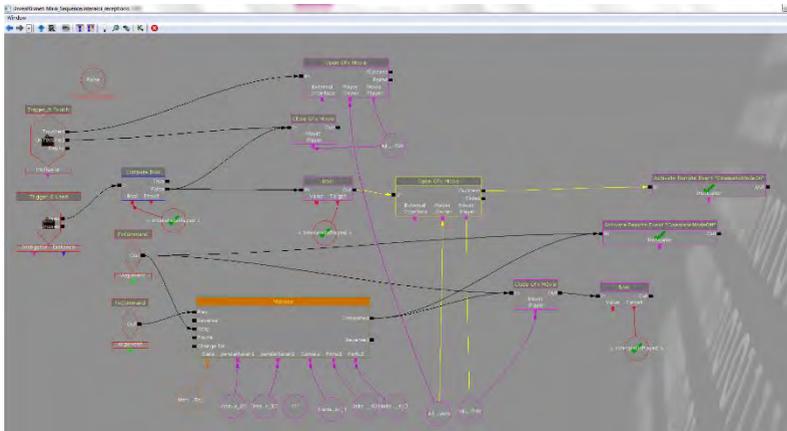
Gambar 5. 40 Kismet interaksi membuka dan menutup pintu



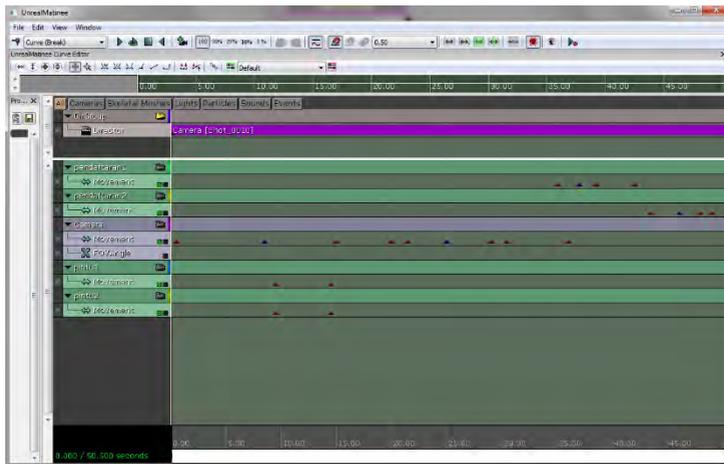
Gambar 5. 41 RemoteEvent pada Kismet interaksi membuka dan menutup pintu

- **Interaksi Prosedur Memasuki Gedung Utama**

Pada interaksi simulasi prosedur memasuki gedung utama berbeda dengan interaksi lampu dan pintu, dikarenakan dibutuhkan matinee yang lebih rumit dan tampilan *flash* yang sedikit berbeda. Tampilan *flash* tersebut berisi informasi simulasi penjelasan prosedur - prosedur dan terdapat tombol untuk menjalankan simulasi serta tombol untuk menghentikan simulasi yang sedang berjalan. Contoh tampilan kismet interaksi prosedur masuk Gedung Utama dapat dilihat pada gambar 5.42. Sedangkan Matinee untuk interaksi tersebut dapat dilihat pada Gambar 5.43.



Gambar 5. 42 Kismet Interaksi Prosedur memasuki Gedung Utama



Gambar 5. 43 Matinee Interaksi Prosedur memasuki Gedung Utama

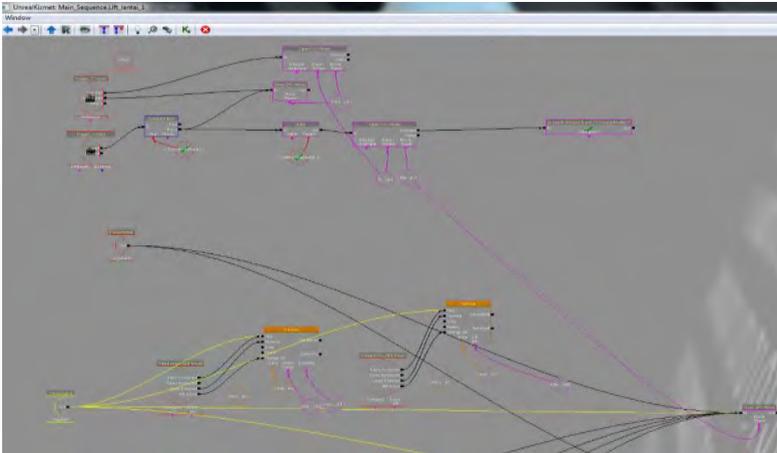
Proyek animasi flashnya dapat dilihat pada Gambar 5.44. Tampilan flash tersebut berisi informasi simulasi pembuatan maket dan terdapat tombol untuk menjalankan simulasi serta tombol untuk menghentikan simulasi yang sedang berjalan.



Gambar 5. 44 Proyek animasi flash interaksi pendaftaran siswa baru

- **Interaksi Menggunakan Lift**

Dalam membuat interaksi lift, beberapa komponen terkait dengan pembuatan objek lift perlu disiapkan terlebih dahulu. Komponen penyusun interaksi lift ini terdiri dari objek lift yang bergerak, ruangan untuk naik dan turunnya lift dan pintu lift. Setelah semua komponen terpenuhi, selanjutnya merubah komponen objek lift yang bergerak dan pintu lift kedalam bentuk interactor. Hal ini dilakukan untuk dapat menggerakkan objek tersebut sesuai dengan kondisi nyata. Selanjutnya, Kismet juga perlu dibuat. Kismet ini berfungsi sebagai pengatur gerakan lift, mulai dari pintu yang terbuka dan menutup secara otomatis, juga pada objek lift yang bergerak secara otomatis ketika aktor menggunakan lift. Gambar 5.45 merupakan tampilan kismet untuk interaksi lift.



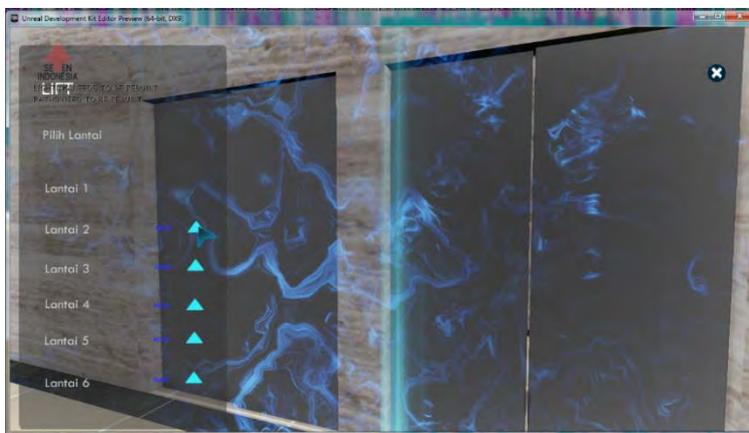
Gambar 5. 45 Kismet penggunaan interaksi lift

Berikut ini adalah detail dari proses yang dilakukan dalam pembuatan interaksi lift:

1. Objek bangunan memiliki 6 lantai, maka untuk setiap lantai perlu ada 1 trigger.
2. Selanjutnya menyiapkan pintu dan liftbox yang sebelumnya telah diubah menjadi Interaptor.
3. Kondisi awal Interaptor pintu adalah tertutup, dan Interaptor liftbox diletakkan dibawah area lift.
4. Selanjutnya pada daerah sekitar pintu tersebut, diberi trigger dimana ketika aktor berada di daerah jangkauan trigger tersebut, akan muncul tampilan animasi *flash*. Animasi *flash* tersebut memuat informasi bahwa pengguna dapat berinteraksi dengan lift yaitu untuk naik maupun turun lantai.
5. Untuk setiap lantai, ada 1 trigger yang telah disiapkan sebelumnya disambungkan dengan interaptor pintu dan Interaptor liftbox..

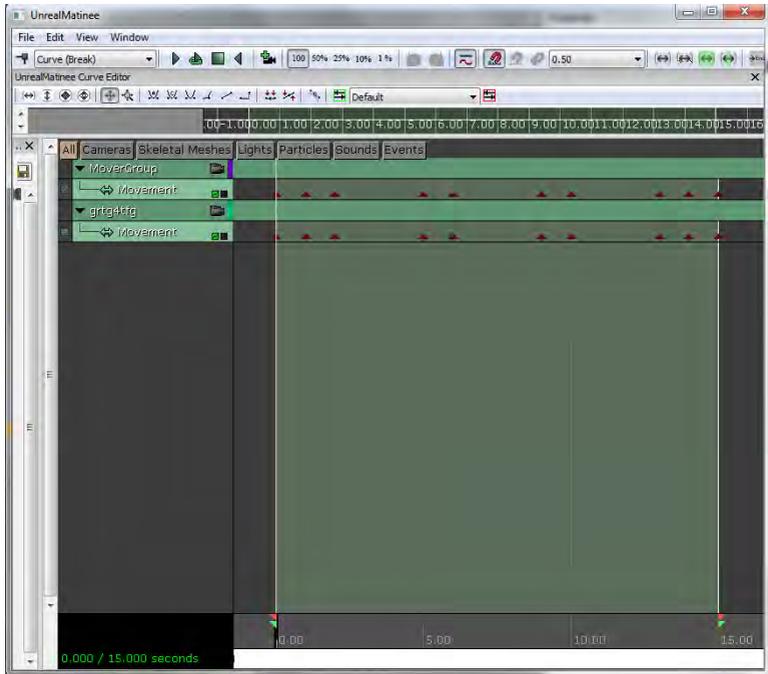
- Setelah proses pengaturan selesai dilakukan, aktor dapat melakukan interaksi yaitu dengan melakukan klik pada *mouse* atau menekan tombol *Enter* pada *keyboard*, untuk menggunakan lift.

Untuk membuka pintu klik kiri pada *mouse* sekali atau tekan *Enter* pada keyboard setelah muncul animasi flash. Selanjutnya objek lift akan bergerak sesuai dengan lantai yang dituju oleh pengguna. Gambar 5.46 adalah contoh penggunaan animasi *flash* pada interaksi lift.



Gambar 5. 46 Animasi flash pada interaksi lift

Pada interaksi lift diperlukan juga pengaturan pada *Matinee* untuk mendapatkan pergerakan pintu dan liftbox sesuai dengan yang diinginkan. Gambar 5.47 adalah tampilan *Matinee* untuk interaksi lift.



Gambar 5. 47 Matinee untuk interaksi penggunaan lift

5.3.5.3 Peta Dua Dimensi

Menu Peta Dua Dimensi yang ada pada aplikasi tugas akhir ini merupakan presentasi dari interaksi peta dua dimensi (2D). Menu Peta 2D merupakan tampilan menu animasi flash untuk tiga fungsi yang dapat digunakan oleh aktor. Tiga fungsi tersebut yaitu mengetahui posisi aktor, teleportasi ke suatu tempat dan menunjukkan arah menuju suatu tempat.

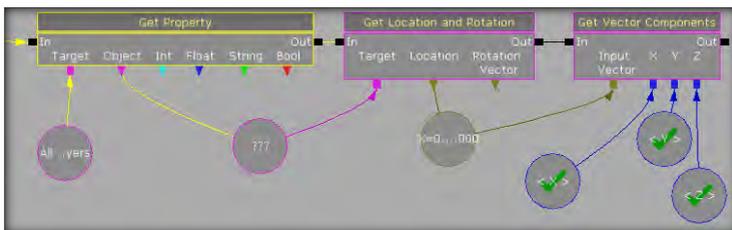
- **Posisi aktor**

Menu Peta Dua Dimensi menampilkan peta 2D dari peta 3D per lantai bangunan. Peta 2D yang digunakan adalah peta 2D yang memuat informasi peta ruangan atau tempat. Tempat-tempat penting diberi simbol khusus agar pengguna langsung

dapat mengakses tempat tersebut. Akses yang diberikan adalah aktor dapat langsung memasuki atau menempati tempat tersebut dengan fungsi teleportasi, aktor dapat menuju tempat tersebut dengan bantuan penunjuk arah, selain itu aktor juga dapat memilih peta 2D bangunan lantai kesekian. Dalam menu tersebut juga terdapat legenda yang menunjukkan keterangan simbol-simbol dalam peta 2D sehingga pengguna dapat mengetahui maksud simbol yang ada pada Menu Peta Dua Dimensi.

Aktor yang digunakan pada aplikasi pada dasarnya juga adalah sebuah objek, seperti yang telah diketahui bahwa objek memiliki properti yang dapat diketahui nilainya. Untuk mengetahui posisi aktor, properti aktor mengenai posisi dapat diambil nilainya. Properti aktor yang perlu diketahui untuk mengetahui posisi aktor adalah `Location` dan `Rotation`.

Properti `Location` yang dimiliki aktor memiliki tiga nilai yaitu `X`, `Y`, `Z` yang dapat diambil sekaligus berupa vektor. Kismet untuk mengambil nilai properti tersebut dapat dilihat pada Gambar 5.48. Pada Gambar tersebut, terlihat bahwa untuk properti Actor `Get Property`, Pawn adalah `Property Name` yang dipakai untuk dapat mengambil nilai vektor `Location` milik aktor.

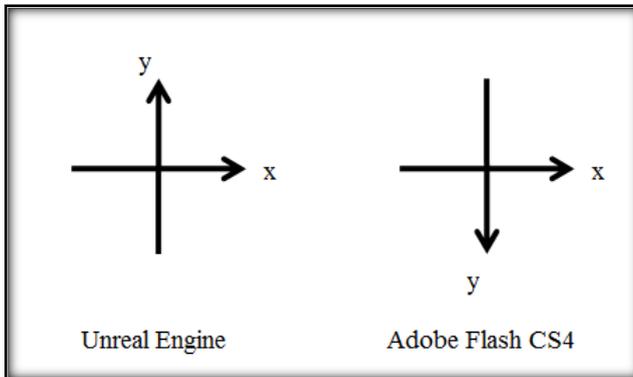


Gambar 5. 48 Kismet pengambilan nilai properti `Location` aktor

Sistem koordinat peta 3D pada Unreal Engine menggunakan sistem koordinat kartesian tiga dimensi yang umum dipakai

dalam matematika. Sistem koordinat tiga dimensi memiliki tiga axis yaitu x-axis, y-axis dan z-axis. Peta 2D dibuat menggunakan tampilan animasi flash yang menggunakan aplikasi animasi flash. Untuk itu, perlu dilakukan penyesuaian antara peta 3D Unreal Engine dengan peta 2D pada aplikasi animasi flash.

Gambar 5.49 menunjukkan perbedaan yang cukup signifikan mengenai x-axis dan y-axis dimana y-axis pada aplikasi animasi flash (Tugas akhir ini menggunakan Adobe Flash CS4) terbalik dibandingkan dengan y-axis peta 3D Unreal Engine. Hal ini menunjukkan bahwa sistem koordinat peta 3D Unreal Engine dan animasi flash Adobe Flash CS4 berbeda sehingga perlu disesuaikan. Penyesuaian dilakukan dengan cara mengaturnya melalui *script* yang ada pada aplikasi animasi flash tersebut, artinya peta 2D yang menyesuaikan dengan peta 3D.



Gambar 5. 49 Perbandingan x-axis dan y-axis peta 3D Unreal Engine dan aplikasi animasi flash Adobe Flash CS5

Penyesuaian dimulai dengan mencari skala antara peta 3D Unreal Engine dengan peta 2D aplikasi animasi flash. Hal ini dapat dilakukan dengan menentukan dua titik tempat tertentu terlebih dahulu pada keduanya. Informasi titik yang diambil

adalah koordinat x-axis dan koordinat y-axis, sehingga akan mendapatkan delapan angka yaitu koordinat x-axis titik tempat pertama aplikasi animasi flash, koordinat y-axis titik tempat pertama aplikasi animasi flash, koordinat x-axis titik tempat kedua aplikasi animasi flash, koordinat y-axis titik tempat kedua aplikasi animasi flash, koordinat x-axis titik tempat pertama peta 3D Unreal Engine, koordinat y-axis titik tempat pertama peta 3D Unreal Engine, koordinat x-axis titik tempat kedua peta 3D Unreal Engine dan koordinat y-axis titik tempat kedua peta 3D Unreal Engine. Persamaan dalam rumus (1) digunakan untuk mendapatkan perbandingan skala peta.

$$\text{perbandingan skala} = \frac{A-B}{C-D} \quad (1) [4]$$

Keterangan rumus (1):

A = koordinat x-axis titik tempat pertama aplikasi animasi flash

B = koordinat x-axis titik tempat kedua aplikasi animasi flash

C = koordinat x-axis titik tempat pertama peta 3D Unreal Engine

D = koordinat x-axis titik tempat kedua peta 3D Unreal Engine

Rumus (1) memperlihatkan cara perhitungan selisih antara koordinat x-axis titik tempat pertama pada aplikasi animasi flash (A) dengan koordinat x-axis titik tempat kedua pada aplikasi animasi flash (B) dibagi dengan selisih antara koordinat x-axis titik tempat pertama pada peta 3D Unreal Engine (C) dengan koordinat x-axis titik tempat kedua pada peta 3D Unreal Engine (D). Dengan mengetahui perbandingan skala antara peta 2D aplikasi animasi flash dengan peta 3D Unreal Engine, maka letak suatu titik di peta 3D Unreal Engine yang diketahui, dapat diposisikan pada peta 2D aplikasi animasi flash.

Tidak cukup hanya dengan mengetahui perbandingan skala antara peta 2D aplikasi animasi flash dengan peta 3D Unreal Engine. Karena sistem koordinat antara kedua peta tersebut berbeda, maka koordinat aktor pada aplikasi animasi flash masih harus diposisikan sesuai dengan objek gambar peta 2D yang dipakai.

Properti `Rotation` yang dimiliki aktor juga memiliki tiga nilai yaitu `Yaw`, `Roll` dan `Pitch` yang juga dapat diambil sekaligus berupa vektor. Kismet untuk mengambil nilai properti tersebut dapat dilihat pada Gambar 5.50.



Gambar 5. 50 Kismet pengambilan nilai properti `Rotation` aktor

`Yaw` adalah rotasi aktor ke samping kanan (searah jarum jam) dan ke samping kiri (berlawanan arah jarum jam) layar menurut pandangannya pada layar dan membentuk lingkaran, maksudnya adalah ketika aktor memutar 360 derajat maka pandangan aktor akan kembali seperti semula. `Roll` adalah rotasi aktor menurut axis layar. `Pitch` adalah rotasi aktor ke atas dan ke bawah menurut pandangannya pada layar. Keterangan mengenai nilai untuk `Pitch`, `Yaw` dan `Roll` pada UDK dapat dilihat pada Tabel 5.3.

Kedua properti tersebut dapat diambil nilainya dalam UnrealKismet dengan menggunakan objek Kismet `Get Property` dan objek Kismet `Get Location and Rotation`. Khusus pada properti `Rotation`, hasil vektor yang diambil tidak terurut seperti pada properti aktor yang telah dijelaskan sebelumnya. `Paw`, `Roll`, kemudian `Yaw`

adalah urutan nilai vektor yang terambil melalui `Get Location and Rotation`. Kedua vektor akhirnya menghasilkan enam nilai yang menunjukkan posisi dan rotasi aktor pada peta 3D.

Tabel 5.3. Nilai vektor Rotation pada properti aktor

No.	Vektor	0	+	-
1	Yaw	Pandangan ke arah timur peta 3D	Pandangan ke arah utara peta 3D	Pandangan ke arah selatan peta 3D
2	Roll	Pandangan lurus ke depan	Pandangan ke samping kanan memutar searah jarum jam	Pandangan ke samping kiri memutar berlawanan arah jarum jam
3	Pitch	Pandangan lurus ke depan	Pandangan ke atas	Pandangan ke bawah

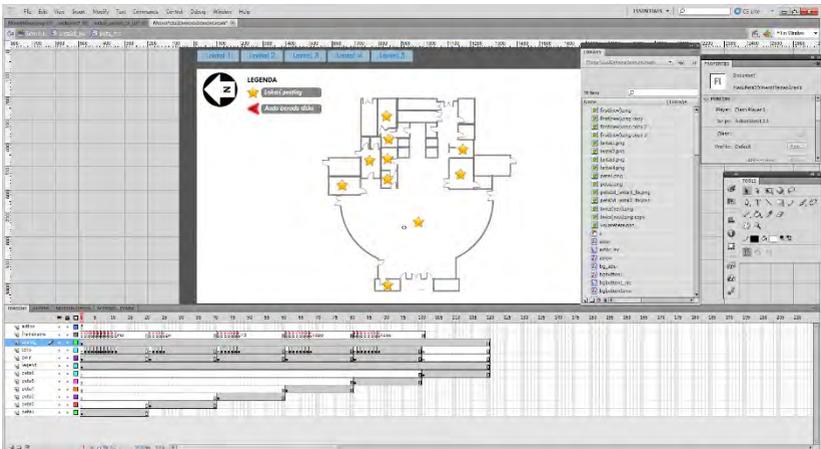
Peta 2D dibuat menggunakan tampilan animasi flash yang menggunakan aplikasi Adobe Flash CS4. Untuk itu, rotasi yang dikenal oleh aplikasi tersebut, sama dengan yang biasa dipakai sehari-hari mengenai rotasi yaitu rotasi berdasarkan derajat. Padahal, sistem rotasi yang ada pada peta 3D Unreal Engine tidak menggunakan sistem rotasi derajat, berbeda dengan sistem rotasi pada aplikasi animasi flash. Unreal Engine memiliki ukuran sendiri mengenai rotasi. Selain itu, perlu mengetahui sistem koordinat pada aplikasi tersebut.

Sistem rotasi pada peta 3D Unreal Engine dapat dilihat dari nilai `Yaw` pada properti aktor `Rotation`. `Yaw` memiliki jangkauan nilai -1 sampai 1. Sedangkan sistem rotasi menurut derajat memiliki jangkauan dari derajat 0 sampai derajat 360. Perlu dilakukan konversi dari sistem rotasi peta 3D Unreal Engine ke sistem rotasi derajat agar dapat mengetahui sudut rotasi pandangan aktor. Penelitian tugas akhir ini menemukan

persamaan yang dapat melakukan konversi tersebut. Dalam persamaan rumus konversi Y_{aw} ke derajat, yaitu rumus (2), terlihat bahwa nilai Y_{aw} pada Unreal Engine (y_{aw}) terlebih dahulu diproses dengan fungsi trigonometrik terbalik arccosinus ($arccos$). Hasil tersebut menghasilkan sudut dalam radian. Sudut dalam radian kemudian diproses menjadi sudut dalam derajat dengan cara mengalikan dengan hasil pembagian 180 dengan konstanta matematika pi (π), maka hasil akhirnya adalah sudut dalam derajat (deg).

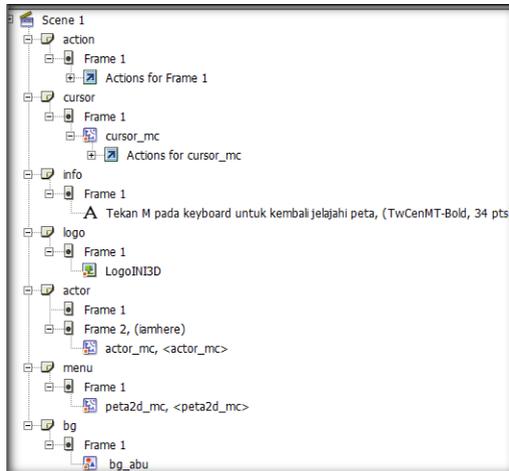
$$deg = arccos(yaw) * \left(\frac{180}{\pi}\right) \quad (2) [4]$$

Pada tugas akhir ini, peta 2D dibuat dengan arah utara yang menuju pada x-axis negatif sistem koordinat aplikasi animasi flash Adobe Flash CS4 atau sama dengan sistem koordinat kartesian menuju pada x-axis negatif. Sedangkan, arah utara peta 3D adalah menuju pada y-axis positif sistem koordinat kartesian. Gambar 5.51 memperlihatkan proyek animasi flash Menu Peta Dua Dimensi dapat dilihat dengan arah utara peta 2D yang telah menuju pada x-axis negatif sistem koordinat aplikasi animasi flash Adobe Flash CS4.



Gambar 5. 51 Animasi flash Menu Peta Dua Dimensi Scene 1

Struktur utama proyek animasi *flash* Menu Peta dibagi menjadi tujuh layer yang berisi Symbol dan/atau ActionScript yang dapat dilihat pada gambar 5.52.



Gambar 5. 52 Struktur utama proyek animasi flash Menu Peta Dua Dimensi

Sebuah `Symbol Movie Clip cursor_mc` digunakan pada animasi flash Menu Peta Dua Dimensi untuk merepresentasikan kursor *mouse*. Gambar 5.53 menunjukkan ActionScript agar `Symbol cursor_mc` dapat beranimasi sesuai dengan *pointer mouse* yang sedang aktif.

```

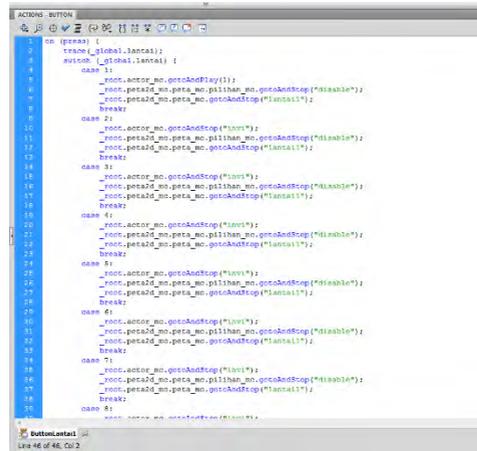
1  onClipEvent (enterFrame) {
2      _x=_root._xmouse;
3      _y=_root._ymouse;
4  }

```

Gambar 5. 53 Potongan kode ActionScript animasi flash Menu Peta Dua Dimensi `Symbol cursor_mc`

- **Teleportasi**

Fungsi selanjutnya dalam Menu Peta Dua Dimensi adalah teleportasi. Dengan fungsi ini, aktor bisa melakukan perpindahan ke tempat yang ingin dituju pada peta 3D Unreal Engine. Pilihan menu tersebut berisi tiga pilihan yaitu “Masuk ke dalam ruangan” merepresentasikan fungsi teleportasi, “Tunjukkan arah menuju ruangan” merepresentasikan fungsi penunjuk arah dan “Tutup dialog pilihan” untuk menutup dialog pilihan menu. Gambar 5.54 memperlihatkan pilihan menu ketika pengguna menekan `Symbol Button` penanda lokasi, yang kemudian muncul pilihan menu untuk melakukan teleportasi yaitu menu Masuk ke dalam ruangan. Pengaturan fungsi teleportasi terdiri dari beberapa konfigurasi melalui ActionScript.



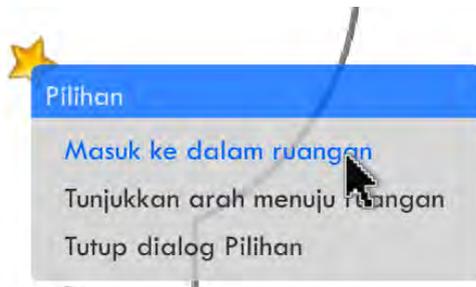
```

1 on (press) {
2     trace(_lib1.lantasi);
3     switch (_lib1.lantasi) {
4         case 1:
5             _root._xcor_mc.gotoAndPlay(1);
6             _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
7             _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
8             break;
9         case 2:
10            _root._xcor_mc.gotoAndStop("Luv");
11            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
12            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
13            break;
14         case 3:
15            _root._xcor_mc.gotoAndStop("Luv");
16            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
17            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
18            break;
19         case 4:
20            _root._xcor_mc.gotoAndStop("Luv");
21            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
22            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
23            break;
24         case 5:
25            _root._xcor_mc.gotoAndStop("Luv");
26            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
27            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
28            break;
29         case 6:
30            _root._xcor_mc.gotoAndStop("Luv");
31            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
32            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
33            break;
34         case 7:
35            _root._xcor_mc.gotoAndStop("Luv");
36            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
37            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
38            break;
39         case 8:
40            _root._xcor_mc.gotoAndStop("Luv");
41            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
42            _root.peta2d_mc.peta_mc.pilihan_mc.gotoAndStop("Masuk");
43            break;
44     }
45 }

```

Gambar 5. 54 Potongan kode ActionScript animasi flash Menu Peta Dua Dimensi Scene 1 layer action frame 1

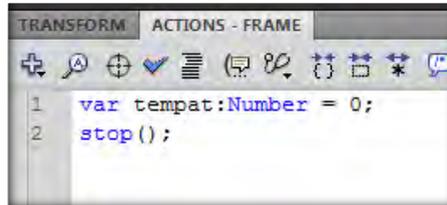
Gambar 5.55 memperlihatkan pilihan menu ketika pengguna menekan *Symbol Button* penanda lokasi, yang kemudian muncul pilihan menu untuk melakukan teleportasi yaitu menu Masuk ke dalam ruangan. Pengaturan fungsi teleportasi terdiri dari beberapa konfigurasi melalui *ActionScript*.



Gambar 5. 55 Pilihan fungsi teleportasi pada Menu Peta Dua Dimensi

Konfigurasi pertama berada pada *Symbol Movie Clip* peta2d_mc layer action frame 1. Gambar 5.56

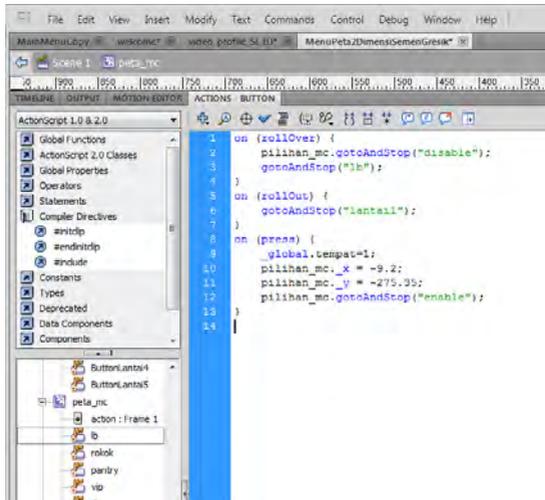
memperlihatkan `ActionScript` dengan melakukan inisialisasi variabel tempat yang nantinya akan berubah ketika suatu `Symbol Button` penanda lokasi diklik dan variabel tempat akan disesuaikan.



Gambar 5. 56 Kode ActionScript animasi flash Menu Peta Dua Dimensi Symbol peta2d_mc layer action frame 1

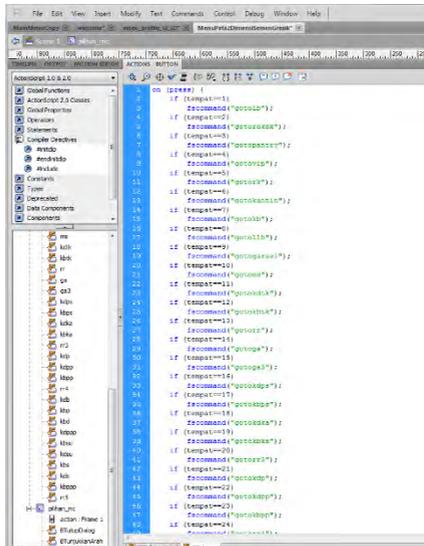
Konfigurasi kedua berada pada pengaturan setiap `Symbol Button` penanda lokasi yang ada pada animasi flash. `Symbol Button` tersebut memiliki `ActionScript` yang berisi inisialisasi variabel tempat dengan mengatur nilai awalnya.

Gambar 5.57 memperlihatkan `ActionScript` melakukan perubahan variabel global tempat ruang lobby gedung utama dengan nilai variabel 1 dan menampilkan pilihan menu yang berupa `Symbol Movie Clip` pilihan_mc. Setiap penanda ruangan berbentuk bintang memuat `ActionScript` dengan konsep yang sama sesuai dengan penomoran tempat atau ruangan ketika `Symbol` tersebut diklik.



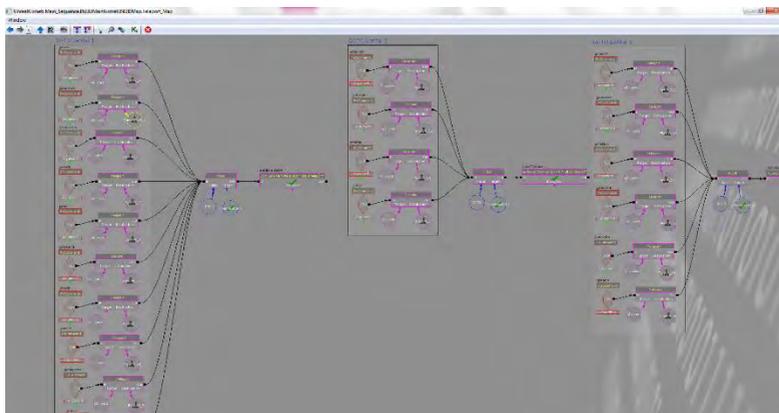
Gambar 5. 57 Potongan kode ActionScript animasi flash Menu Peta Dua Dimensi pada setiap Symbol Button penanda lokasi

Konfigurasi ketiga untuk fungsi teleportasi ada pada Symbol Button Bteleport, yaitu menu “Masuk ke dalam ruangan”. Gambar 5.58 menunjukkan ActionScript pada button tersebut dimana sebuah `fscommand` (fungsi yang dapat memanggil Actor `Fscommand` pada UnrealKismet) akan aktif berdasarkan variabel tempat.



Gambar 5. 58 Potongan kode ActionScript animasi flash Menu Peta Dua Dimensi Symbol Button Bteleport

Sedangkan untuk konfigurasi agar fungsi teleportasi bisa digunakan hingga lebih dari 3 lantai adalah dengan melakukan pengaturan Actionframe pada ActionScript yang mana kita harus menentukan posisi tinggi dari tiap lantai. Konfigurasi harus dilakukan dengan berulang – ulang agar nantinya bisa mendapatkan hasil yang tepat. Karena apabila kita salah dalam menentukan posisi tinggi tiap lantai, maka nantinya fitur teleportasi tidak akan bisa digunakan apabila Gedung mempunyai tinggi lebih dari 3 lantai. Konfigurasi ini bisa dilihat pada Gambar 5.59



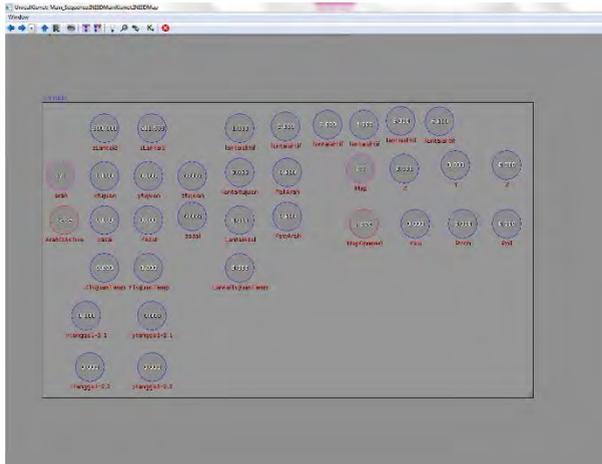
Gambar 5. 60 Kismet teleportasi

Konfigurasi Kismet teleportasi ini sama untuk semua ruangan atau lokasi, yang membedakan hanyalah Actor Fscommandnya karena setiap Fscommand akan merepresentasikan pemanggilan fungsi fscommand pada animasi flash dengan tujuan lokasi ruangan yang berbeda.

- **Penunjuk arah**

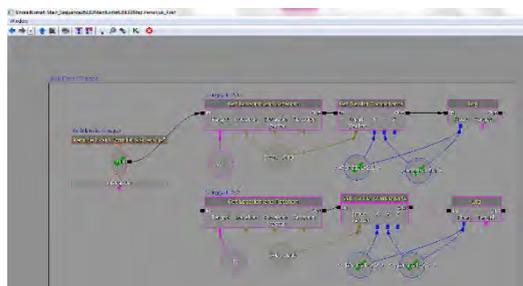
Untuk fungsi penunjuk arah, logika didominasi oleh pengaturan dan perhitungan dalam Kismet. Sebuah tampilan flash yang memuat delapan gambar panah berbeda digunakan sebagai tampilan penunjuk arah dengan mengambil *input* dari peta 3D melalui *function* dalam ActionScript. Jika untuk melakukan sesuatu pada peta 3D, flash bisa menggunakan fungsi fscommand, jika untuk sebaliknya yaitu apa yang terjadi pada peta 3D ingin digunakan oleh flash, maka digunakan *function* dalam flash yang dapat dipanggil oleh Kismet melalui Get Invoke ActionScript.

Pertama-tama, seluruh variabel yang diperlukan harus diinisialisasi seperti yang terlihat pada Gambar 5.61



Gambar 5. 61 Inisialisasi variabel dalam kismet untuk fungsi penunjuk arah

Kemudian perlu dilakukan pengambilan posisi setiap tangga yang ada pada peta 3D sehingga dapat menghubungkan lokasi pada lantai bangunan yang berbeda. Kismet tersebut dapat dilihat pada Gambar 5.62.



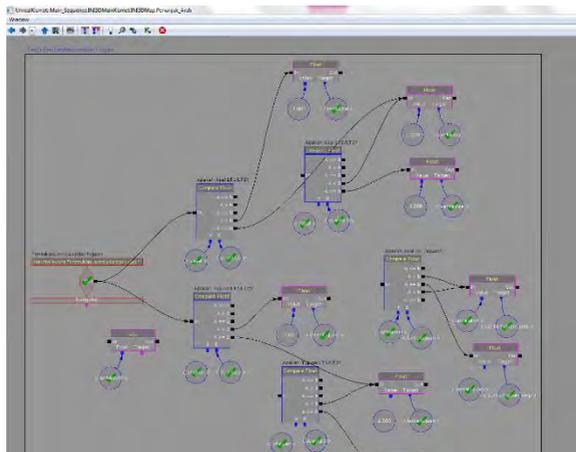
Gambar 5. 62 Kismet mengambil posisi tangga

Setelah posisi tangga diketahui, selanjutnya adalah mengetahui posisi aktor berada. *Kismet* untuk mengambil posisi aktor tersebut dapat dilihat pada Gambar 5.63.



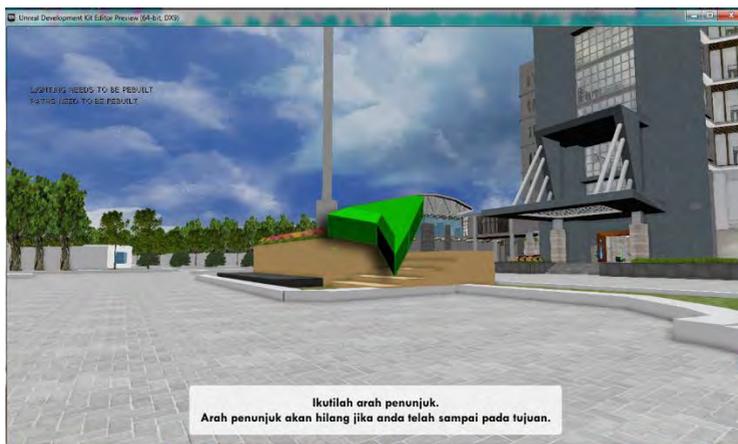
Gambar 5. 63 Kismet penunjuk arah untuk mengambil posisi aktor

Tidak hanya posisi aktor yang perlu diketahui untuk dapat menunjukkan arah menuju suatu lokasi, perlu ditentukan juga lantai asal dan tujuan mengingat peta 3D yang dibangun merupakan sebuah gedung yang memiliki tingkatan lantai. Oleh karena perlu menentukan lantai asal dan tujuan berdasarkan nilai Z vektor *Location* properti milik aktor yang telah didapat sebelumnya. Cara penentuannya hanya melakukan perbandingan dengan batas perpindahan lantai gedung pada Peta 3D dapat dilihat pada Gambar 5.64.



Gambar 5. 64 Kismet penentuan lantai asal dan tujuan lokasi penunjuk arah

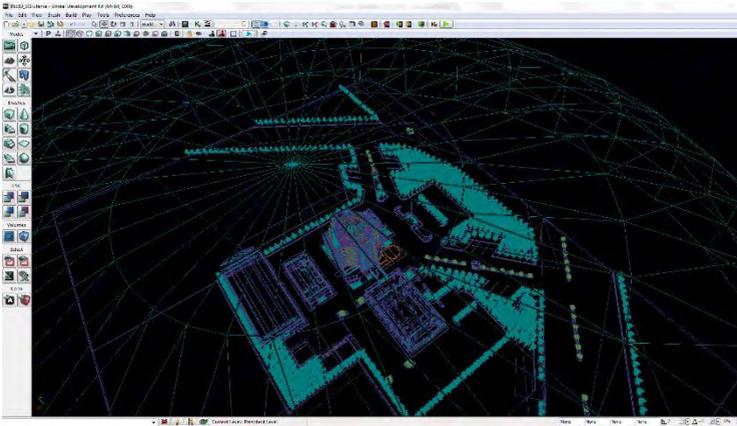
Tampilan penunjuk arah ketika sedang aktif, dapat dilihat pada Gambar 5.65.



Gambar 5. 65 Tampilan ketika penunjuk arah sedang aktif

5.3.5.4 Pergantian Siang dan Malam

Pergantian siang dan malam bertujuan agar pengguna mendapatkan gambaran yang lebih realistis dan sesuai dengan keadaan nyata pada peta 3D. Tahap awal pembuatan simulasi pergantian siang dan malam adalah pemberian *staticmesh* langit. *Staticmesh* ini berbentuk kubah setengah bola yang digunakan sebagai latar belakang langit yang mengelilingi keseluruhan peta dan dapat dilihat pada Gambar 5.66.



Gambar 5. 66 Staticmesh Langit pada Mode Wireframe

Selain menggunakan *staticmesh* langit, digunakan juga 3 aktor berfungsi pada fungsinya masing-masing yaitu pertama aktor *DynamicDirectionalLight* yang berfungsi menampilkan cahaya dominan seperti matahari atau bulan dan bergerak secara dinamis mengikuti pengaturan *Matinee*, kedua aktor *Fog* yang berfungsi menambahkan kabut pada langit agar lebih terlihat realistis, dan terakhir aktor *MaterialInstance* yang berfungsi mengubah warna langit saat pergantian siang ke malam.

Standar yang digunakan dalam simulasi pergantian siang dan malam ini menggunakan standar INI3D [4] dan rumus *playrate* yang digunakan:

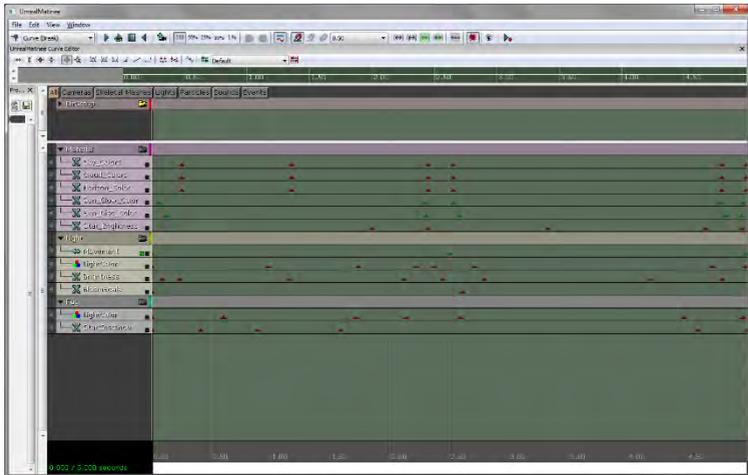
$$Playrate = \frac{1}{\frac{\text{Lama siklus yang diinginkan}}{\text{Durasi matinee}}} \quad (3)$$

Keterangan rumus (3):

Lama siklus yang diinginkan = 24 menit

Durasi matinee = 5 detik

Hasil dari perhitungan *playrate* menampilkan matinee yang mengatur proses diatas yang dapat dilihat pada Gambar 5.67 dan hasil di dalam 3D dapat dilihat pada Gambar 5.68.



Gambar 5. 67 Matinee Simulasi Pergantian Siang dan Malam



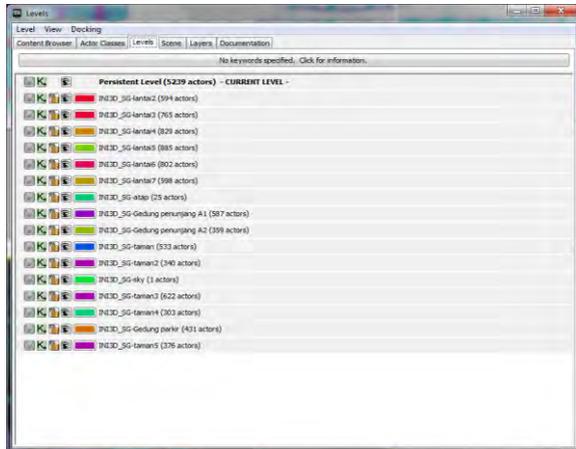
Gambar 5. 68 Simulasi Pergantian Siang dan Malam

5.4.Integrasi

Unreal Engine menyediakan beberapa cara untuk melakukan integrasi antar beberapa peta 3D yang berbeda. Integrasi antar peta di UDK dilakukan dengan dua cara yaitu `Level Streaming` dan `Level Loading`.

5.4.1 Level Streaming

`Level Streaming` artinya integrasi peta dilakukan secara *live* tanpa perantara (*loading*). Untuk melakukan `Level Streaming` diperlukan `level map` baru yang akan menggabungkan peta satu dengan peta yang lainnya. Pada `level` baru tersebut, `Level Streaming` dilakukan dengan cara menghubungkan setiap peta pada jendela `Level` seperti pada Gambar 5.69. Dalam `Level Streaming`, terdapat dua cara. Cara pertama menggabungkan antar peta adalah pilih peta yang akan digabung dengan `Add Existing Level` dan pilih dengan `Kismet`. Setelah peta-peta muncul dalam jendela `Level` seperti yang terlihat pada Gambar 5.61, maka lokasi peta dapat diatur sesuai kebutuhan yaitu menyesuaikan dengan letak peta 3D lain yang akan digabung.



Gambar 5. 69 Contoh integrasi peta 3D Gedung Utama PT. Semen Indonesia (Persero) Tbk Wilayah Tuban yang diintegrasikan melalui Level Streaming

Setelah semua peta yang akan digabungkan tertera pada jendela Levels, diperlukan alur Kismet untuk mengatur ketika pengguna muncul pada level baru tersebut sistem akan menampilkan semua peta yang akan digabungkan. Maka semua peta yang sudah digabungkan dapat dilihat pada Gambar 5.70.

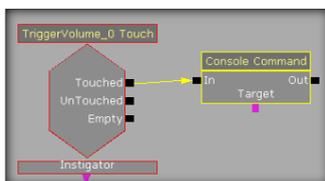


Gambar 5. 70 Beberapa peta 3D Unreal Engine hasil integrasi melalui Level Streaming

Hasil jalannya aplikasi menggunakan metode ini tidak cukup bagus karena aplikasi berjalan sangat lambat. Namun masih dapat dimungkinkan untuk integrasi maksimal tiga buah peta 3D.

5.4.2 Level Loading

Level Loading dilakukan untuk mempermudah pengguna dalam berpindah peta disebabkan dampak lambat dari aplikasi jika memakai *full* Level Streaming. Konsep dari Level Loading yang dilakukan yaitu dengan menggunakan `TriggerVolume` sepanjang perbatasan antar peta yang bertujuan untuk menampilkan peta yang lain. Sehingga pada Kismet, diberikan alur logika jika pengguna menyentuh `TriggerVolume` tersebut, maka sistem akan menampilkan peta baru.



Gambar 5. 71 Kismet integrasi peta 3D dengan Level Loading

5.5. Uji Coba dan Evaluasi

Subbab ini berisi bagian uji coba dan evaluasi implementasi aplikasi. Uji coba dibagi menjadi dua yaitu uji coba fungsional dan uji coba non-fungsional.

5.5.1. Uji Coba Fungsional

Uji coba fungsional dilakukan melalui uji coba dari rancangan *test case* yang telah dirancang pada lampiran C. Setiap skenario pada *test case* dijalankan dan hasil yang ada pada *test case* dibandingkan

dengan hasil aplikasi. Laporan mengenai uji coba *test case* tersebut dapat dilihat pada Tabel 5.4.

Tabel 5.4. Hasil uji coba dari rancangan *test case*

No.	Test Case ID	Hasil
1.	TC1-01	Berhasil
2.	TC1-02	Berhasil
3.	TC2-01	Berhasil
4.	TC3-01	Berhasil
5.	TC3-02	Berhasil
6.	TC3-03	Berhasil
7.	TC3-04	Berhasil
8.	TC4-01	Berhasil
9.	TC5-01	Berhasil
10.	TC5-02	Berhasil

5.5.2. Uji Coba Non-Fungsional

Uji coba non-fungsional dilakukan dengan uji coba performa. Uji coba performa dilakukan dengan membandingkan performa dari beberapa komputer. Performa dinilai berdasarkan FPS (*Frame Per Second*) rate dan penggunaan memori. *Tools* yang digunakan adalah fitur dari UDK yaitu Stat FPS dan Stat memory.

- Stat FPS
Stat FPS merupakan *tools* untuk memperlihatkan FPS *counter* dan lama peta dijalankan. Perintah yang digunakan adalah ‘stat fps’.
- Stat memory
Stat memory merupakan *tools* untuk memperlihatkan penggunaan memory. Perintah yang digunakan adalah ‘stat memory’.

Aplikasi dijalankan melalui Unreal Editor atau Unreal FrontEnd kemudian menekan tombol *tab* pada *keyboard* dan mengetikkan

perintah sebuah *tools*, maka akan muncul laporannya. Uji coba performa dilakukan pada tiga buah PC (*Personal Computer*) yang masing-masing dilakukan tiga kali. Spesifikasi tiga buah PC yang digunakan untuk uji coba dapat dilihat pada Tabel 5.5, Tabel 5.6, Tabel 5.7 dan Tabel 5.8.

Tabel 5.5. Spesifikasi PC 1

Processor	Intel® Core™ i5-3570 CPU @ 3.4Ghz (4 CPUs), ~3.8GHz
Memori	8192MB RAM
VGA	NVIDIA Geforce GTX660TI
DirectX	DirectX 11
Sistem Operasi	Windows 7 Ultimate 32-bit(6.1, Build 7601)

Tabel 5.6. Spesifikasi PC 2

Processor	Intel® Core™ 2 Duo CPU E7500 @ 2.93Ghz (2 CPUs), ~2.9GHz
Memori	4096MB RAM
VGA	ATI Radeon HD 5700 series 2805
DirectX	DirectX 11
Sistem Operasi	Windows 7 Ultimate 32-bit(6.1, Build 7601)

Tabel 5.7. Spesifikasi PC 3

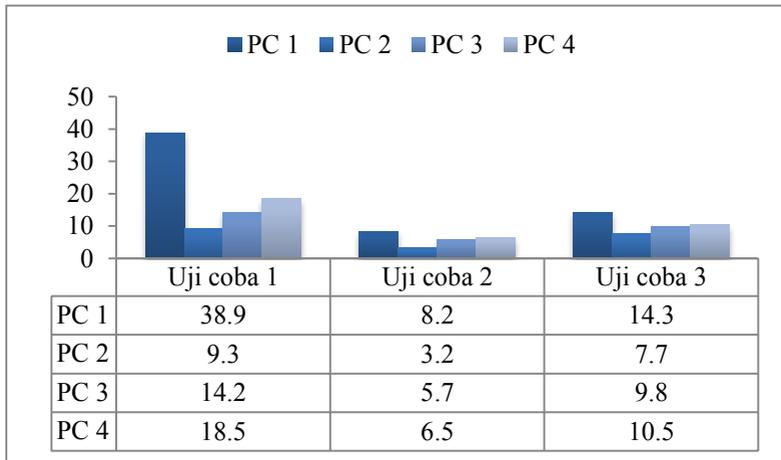
Processor	Intel® Core™ i5-4440 CPU @ 3.1Ghz (4 CPUs), ~3.1GHz
Memori	8192MB RAM
VGA	AMD RADEON HD 7800
DirectX	DirectX 11
Sistem Operasi	Windows 7 Ultimate 64-bit (6.1, Build 7601)

Tabel 5.8. Spesifikasi PC 4

Processor	Intel® Core™ i5-3450 CPU @ 3.1Ghz (4 CPUs), ~3.1GHz
Memori	8192MB RAM
VGA	NVIDIA Geforce GTX560 SE

DirectX	DirectX 11
Sistem Operasi	Windows 7 Ultimate 64-bit (6.1, Build 7601)

Hasil uji coba performa FPS *rate* dirangkum dalam grafik pada Gambar 5.72.



Gambar 5. 72 Grafik perbandingan FPS rate

Keterangan Gambar :

- Ujicoba 1: Keadaan outdoor, pertama kali starting sebelum matahari muncul
- Ujicoba 2 : Keadaan outdoor, siang hari posisi actor di UDK sama dengan ujicoba 1
- Ujicoba 3 : Keadaan Indoor, posisi sama tiap PC

Keterangan FPS:

- FPS > 60, maka spesifikasi tersebut sangat dianjurkan untuk menjalankan aplikasi
- FPS < 60 namun FPS > 30, maka spesifikasi tersebut cukup untuk menjalankan aplikasi
- FPS < 30, maka spesifikasi tersebut tidak dianjurkan untuk menjalankan aplikasi

Hasil analisa :

1. Dari grafik pada gambar Gambar 5.67, terlihat bahwa aplikasi berjalan dengan kondisi FPS rendah pada komputer 1,3 dan 4 yang sudah menggunakan VGA Card dan di tunjang dengan memory RAM yang besar. *FPS Rate* pada masing-masing komputer masih berkisar <30 FPS. Berbeda dengan *FPS Rate* pada komputer 2 sangat tidak dianjurkan untuk menjalankan aplikasi.
2. Dari ketiga hasil ujicoba didapatkan *FPS Rate* yang berbeda, hal tersebut dipengaruhi adanya Fitur Siang Malam (matahari). Pada Ujicoba 1, matahari belum tampak sehingga masih terasa ringan. Pada Ujicoba 2 dan 3 *FPS Rate* menunjukkan pemakaian yang berat dikarenakan efek bayangan pada objek UDK.
3. Pengambilan data *FPS* dilakukan dengan cara mengarahkan pandangan karakter ke depan, belakang, kiri dan kanan baik itu didalam gedung jurusan maupun diluar gedung jurusan.
4. *FPS* dideteksi dengan fitur dari UDK yaitu Stat *FPS*. Aplikasi dijalankan melalui *Unreal Editor* atau *Unreal FrontEnd* kemudian menekan tombol *tab* pada *keyboard* dan mengetikkan tulisan stat *FPS*, maka akan muncul laporan *FPSrate*.

5.5.3. Evaluasi Implementasi

Evaluasi dilakukan dengan cara validasi peta 3D Unreal Engine dengan memperlihatkan perbandingan gambar pada peta 3D dengan foto pada kondisi nyata. Ukuran kualitatif seperti informatif, interaktif dan yang mendekati nyata dan diukur oleh calon pengguna aplikasi melalui pengadaaan kuesioner sebagai bentuk evaluasi belum dapat dilakukan pada tugas akhir ini.

Pada evaluasi validasi peta 3D digambarkan secara jelas tentang hasil implementasi ruangan yang telah dimodelkan pada peta 3D beserta gambar asli ruangan tersebut. Evaluasi tersebut dapat dilihat pada Tabel 5.9.

Tabel 5.9. Evaluasi implementasi model peta 3D

Ruangan	Kondisi Nyata	Peta 3D
Tampak Samping Gedung Utama PT.Semen Indonesia (Persero) Tbk. Wilayah Tuban		
Ruang Tunggu VIP		
Lobby Gedung Utama		

<p>Pintu Belakang</p>		
<p>Ruang kerja lantai 2</p>		
<p>Lantai 5 Gedung Utama</p>		
<p>Ruang kerja departement internal audit (Lantai 6)</p>		

<p>Tampak samping ruang kerja Departement Keuangan dan Akuntansi (Lantai 3)</p>		
<p>Tampak samping ruang kerja Departement Penjualan (Lantai 4)</p>		
<p>Tampak belakang gedung penunjang A1</p>		

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil pengerjaan tugas akhir yang telah dilakukan, maka dapat diambil beberapa kesimpulan seperti di bawah ini.

1. Standarisasi dalam pengerjaan menggunakan standar aplikasi INI3D yang sudah ada sebelumnya, memudahkan dalam pembuatan peta tiga dimensi dengan menggunakan Unreal Development Kit beserta fitur-fitur yang disediakan.
2. Penggunaan standar ukuran yang sudah ditetapkan juga membantu penulis dalam proses integrasi dengan peta tiga dimensi Gedung Utama PT. Semen Indonesia (Persero) Tbk Wilayah Tuban.
3. Selama penelitian, pemakaian *Brusher* memakan waktu lebih lama saat pertama proses building, namun lebih cepat pada proses building selanjutnya. Berbeda dengan *StaticMesh* yang selalu memakan waktu lebih lama setiap melakukan proses building peta.
4. Keterbatasan dari UDK untuk membuat detail objek yang rumit dapat diselesaikan dengan membuat tiruan dengan detail yang lebih sederhana yang akan lebih mudah jika dibuat di aplikasi modelling 3D seperti 3D Max.
5. Penggunaan VGA Card dan Memory sangat dianjurkan untuk menunjang performa UDK dari pada penggunaan VGA On Board.

6.2. Saran

Pengembangan aplikasi ini tentunya tidak lepas dari batasan batasan dan kesalahan didalam proses pengerjaannya, dan juga masih banyak memiliki kekurangan. Maka penulis dapat memberikan beberapa saran untuk pengembangan aplikasi kedepannya, yaitu :

1. Semua implementasi dalam tugas akhir ini merupakan penelitian dasar yang dilakukan tentang UDK di mana eksplorasi lebih untuk semua implementasi yang telah dilakukan perlu pengembangan aplikasi selanjutnya.
2. Sebaiknya di lakukan pencegahan dini akibat dari aplikasi UDK yang sering terjadi bug atau tiba – tiba aplikasi berhenti secara mendadak adalah dengan mengaktifkan *auto save* atau melakukan *back up* secara berkala.
3. Aktor yang terdapat dalam aplikasi ini masih belum mendekati nyata dalam hal gerakan maupun bentuk beserta materialnya, sehingga perlu dilakukan eksplorasi yang lebih dalam mengenai aktor.
4. Penggunaan LevelLoading untuk setiap ruangan yang ada didalam satu peta untuk mengantisipasi terjadinya *lag* yang sangat parah.
5. Pemasangan Static Mesh tumbuhan bergerak sangat berat dan cukup mengganggu dalam proses pengerjaan maupun pada saat proses building, disarankan untuk membuat Level Map sendiri untuk level taman, sehingga dengan adanya pembagian Level Map ini dapat membantu penulis dalam proses pengerjaan maupun pada saat proses building karena aplikasi berjalan tidak terlalu berat.

LAMPIRAN A
DESKRIPSI USE CASE

A.1. Deskripsi Use Case Interaksi dengan Obyek

Tabel A.1. Deskripsi *Use Case* Interaksi dengan Obyek

UC01 – Interaksi dengan Obyek	
Primary Actor: Pengguna	Level: User Goal
Pre-conditions: Pengguna berada di halaman Peta 3D.	
Triggers: <ul style="list-style-type: none"> • Pengguna bergerak masuk dalam jangkauan area interaksi suatu obyek. 	
Basic course: Sistem menampilkan pesan interaksi yang dapat terjadi dengan suatu obyek. Pengguna menekan tombol mouse kiri. Sistem akan menjalankan fungsi interaksi pada obyek tersebut.	
Post-conditions: Sistem telah menjalankan fungsi interaksi obyek tersebut dan obyek berubah kondisi sesuai dengan fungsi interaksinya.	
Alternate courses: Jika pengguna tidak menekan tombol apapun: sistem menampilkan pesan interaksi yang dapat terjadi dengan suatu obyek.	

Jika pengguna menekan tombol M pada keyboard: sistem menjalankan UC02

Jika pengguna menekan tombol W/A/S/D/panah atas/panah bawah/panah kiri/panah kanan pada keyboard: sistem menjalankan UC05

A.2. Deskripsi Use Case Melihat Peta 2 Dimensi

Tabel A.2. Deskripsi *Use Case* Melihat Peta 2 Dimensi

UC02 – Melihat Peta 2 Dimensi	
Primary Actor: Pengguna	Level: User Goal
Pre-conditions: Pengguna berada di halaman Peta 3D.	
Triggers: <ul style="list-style-type: none"> • Pengguna menekan tombol M pada keyboard. 	
Basic course: Pengguna menekan tombol M pada keyboard. Sistem menampilkan peta 2 Dimensi.	
Post-conditions: -	
Alternate courses:	

Jika pengguna menekan tombol W/A/S/D/panah atas/panah bawah/panah kiri/panah kanan pada keyboard: sistem menjalankan UC05

Jika pengguna dalam jangkauan areainteraksi suatu obyek dan menekan klik kiri pada mouse: sistem menjalankan UC01

A.3. Deskripsi Use Case Memilih Menu Jelajah

Tabel A.3. Deskripsi *Use Case* Memilih Menu Jelajah

UC03 – Memilih Menu Jelajah	
Primary Actor: Pengguna	Level: User Goal
Pre-conditions: Pengguna berada di halaman Menu Awal.	
Triggers: <ul style="list-style-type: none"> • Pengguna memilih menu Jelajahi Peta dan menekan tombol Enter pada keyboard atau klik kiri pada mouse. 	
Basic course: Pengguna memilih menu Jelajahi Peta dan menekan tombol Enter pada keyboard atau klik kiri pada mouse. Sistem menampilkan halaman Menu Utama.	
Post-conditions: Sistem menampilkan halaman Menu Utama.	

Alternate courses:

Jika pengguna memilih menu Keluar: sistem menampilkan halaman Menu Keluar.

A.4. Deskripsi Use Case Mengubah Resolusi

Tabel A.4. Deskripsi *Use Case* Mengubah Resolusi

<i>UC04</i> – Mengubah Resolusi	
<i>Primary Actor:</i> Pengguna	<i>Level:</i> User Goal
<i>Pre-conditions:</i> Pengguna berada di halaman Menu Utama.	
<i>Triggers:</i> <ul style="list-style-type: none"> • Pengguna memilih menu Pilihan Resolusi dan menekan tombol Enter pada keyboard atau klik kiri pada mouse. 	
<i>Basic course:</i> Pengguna memilih menu Pilihan Resolusi dan menekan tombol Enter pada keyboard atau klik kiri pada mouse. Sistem menampilkan halaman Menu Resolusi. Pengguna memilih salah satu resolusi dan menekan tombol Enter pada keyboard atau klik kiri pada mouse. Sistem menyimpan resolusi yang dipilih oleh pengguna dan mengubah resolusi tampilan sesuai dengan yang dipilih oleh pengguna.	
<i>Post-conditions:</i>	

Sistem menampilkan halaman dengan resolusi yang telah dipilih oleh pengguna.
<i>Alternate courses:</i>
Jika pengguna memilih menu Kembali: sistem menampilkan halaman Menu Utama.

A.5. Deskripsi Use Case Navigasi

Tabel A.5. Deskripsi *Use case* Navigasi

<i>UC05</i> – Navigasi	
<i>Primary Actor:</i> Pengguna	<i>Level:</i> User Goal
<i>Pre-conditions:</i> Pengguna berada di halaman Peta 3D.	
<i>Triggers:</i> -	
<i>Basic course:</i> Jika pengguna menekan W atau panah atas pada keyboard, sistem menggerakkan aktor ke arah depan. Jika pengguna menekan A pada keyboard, sistem menggerakkan aktor ke arah kiri. Jika pengguna menekan D pada keyboard, sistem menggerakkan aktor ke arah kanan.	

<p>Jika pengguna menekan S atau panah bawah pada keyboard, sistem menggerakkan aktor ke arah belakang.</p> <p>Jika pengguna menekan panah kiri pada keyboard, sistem mengarahkan pandangan aktor ke kiri.</p> <p>Jika pengguna menekan panah kanan pada keyboard, sistem mengarahkan pandangan aktor ke kanan.</p> <p>Jika pengguna menekan C pada keyboard, sistem menggerakkan aktor pada posisi jongkok.</p> <p>Jika pengguna menekan F pada keyboard, sistem menggerakkan aktor pada posisi tidur.</p> <p>Jika pengguna menekan Spasi pada keyboard, sistem menggerakkan aktor untuk melompat.</p>
<p><i>Post-conditions:</i></p> <p>Sistem menggerakkan aktor sesuai dengan arah navigasi dan menyesuaikan tampilan dengan pandangan aktor pada posisi barunya.</p>
<p><i>Alternate courses:</i></p> <p>Jika pengguna dalam jangkauan areainteraksi suatu obyek dan menekan klik kiri pada mouse: sistem menjalankan UC01</p> <p>Jika pengguna menekan tombol M pada keyboard: sistem menjalankan UC02</p>

A.6. Deskripsi Use Case Memilih Peta

Tabel A.6. Deskripsi *Use Case* Memilih Peta

<i>UC06</i> – Memilih Peta	
<i>Primary Actor:</i>	<i>Level:</i>

Pengguna	User Goal
<p><i>Pre-conditions:</i></p> <p>Pengguna berada di halaman Menu Utama.</p>	
<p><i>Triggers:</i></p> <ul style="list-style-type: none"> • Pengguna memilih menu Pilihan Peta dan menekan tombol Enter pada keyboard atau klik kiri pada mouse. 	
<p><i>Basic course:</i></p> <p>Sistem menampilkan halaman Pilihan Peta. Pengguna memilih salah satu peta dan menekan tombol Enter pada keyboard atau klik kiri pada mouse. Sistem menyimpan pilihan peta yang dipilih oleh pengguna dan menampilkan halaman Menu Utama.</p>	
<p><i>Post-conditions:</i></p> <p>Sistem menyimpan pilihan peta yang dipilih oleh pengguna dan menampilkan halaman Menu Utama.</p>	
<p><i>Alternate courses:</i></p> <p>Jika pengguna memilih menu Kembali: sistem menampilkan halaman Menu Utama.</p>	

A.7. Deskripsi Use Case Menjelajahi Peta

Tabel A.7. Deskripsi *Use Case* Menjelajahi Peta

UC07 – Menjelajahi Peta

Primary Actor: Pengguna	Level: User Goal
Pre-conditions: Pengguna berada di halaman Menu Utama.	
Triggers: <ul style="list-style-type: none"> • Pengguna memilih menu Mulai dan menekan tombol Enter pada keyboard atau klik kiri pada mouse. 	
Basic course: Sistem me-load pilihan peta aktif dan menampilkan halaman Peta 3D sesuai dengan pilihan peta aktif.	
Post-conditions: Sistem menampilkan halaman Peta 3D sesuai dengan pilihan peta aktif.	
Alternate courses: Jika pengguna memilih menu Kembali: sistem menampilkan halaman Menu Utama.	

A.8. Deskripsi Use Case Melihat Bantuan

Tabel A.8. Deskripsi *Use Case* Melihat Bantuan

UC08 – Melihat Bantuan	
Primary Actor: Pengguna	Level: User Goal

<i>Pre-conditions:</i>	
Pengguna berada di halaman Menu Utama atau di halaman Peta 3D.	
<i>Triggers:</i>	
<ul style="list-style-type: none">• Pengguna berada di halaman Menu Utama kemudian memilih menu Bantuan dan menekan tombol Enter pada keyboard atau klik kiri pada mouse.• Pengguna berada di halaman Peta 3D dan menekan tombol ESC pada keyboard kemudian memilih menu Bantuan dan menekan tombol Enter pada keyboard atau klik kiri pada mouse.	
<i>Basic course:</i>	
Sistem menampilkan halaman Bantuan. Pengguna melihat halaman Bantuan.	
<i>Post-conditions:</i>	
-	
<i>Alternate courses:</i>	
Jika pengguna memilih menu Kembali: sistem menampilkan halaman Menu Utama.	

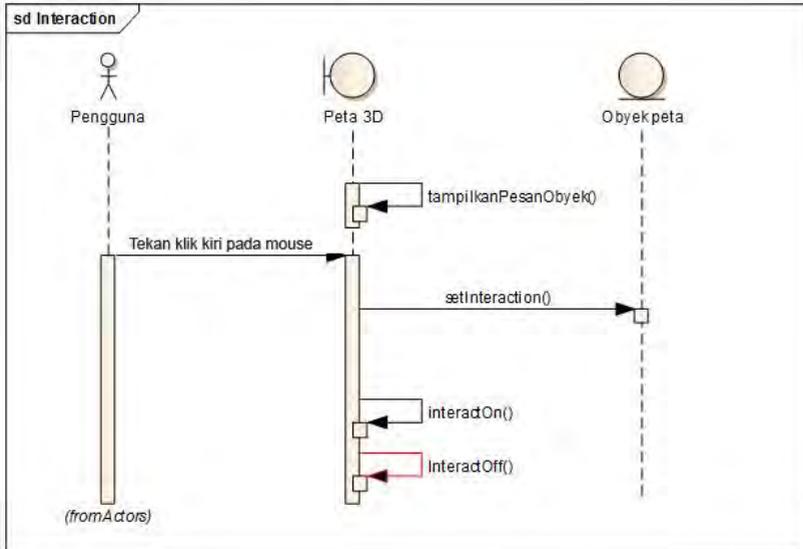
A.9. Deskripsi Use Case Mengaktifkan Layar Informasi

Tabel A.9. Deskripsi *Use Case* Mengaktifkan Layar Informasi

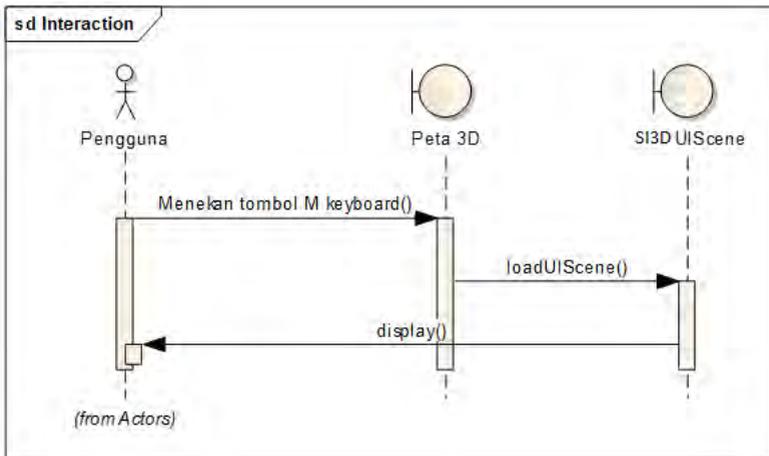
UC09 – Mengaktifkan Layar Informasi	
Primary Actor: Pegguna	Level: User Goal
Pre-conditions: Pegguna berada di halaman peta 3D.	
Triggers: <ul style="list-style-type: none"> • Pegguna bergerak masuk dalam jangkauan area interaksi suatu obyek 	
Basic course: Pegguna bergerak masuk dalam jangkauan area interaksi suatu obyek. Pegguna menekan tombol mouse kiri. Sistem menampilkan layar informasi. Pegguna melakukan informasi sesuai dengan alur interaksi.	
Post-conditions: -	
Alternate courses:	

LAMPIRAN B
SEQUENCE DIAGRAM

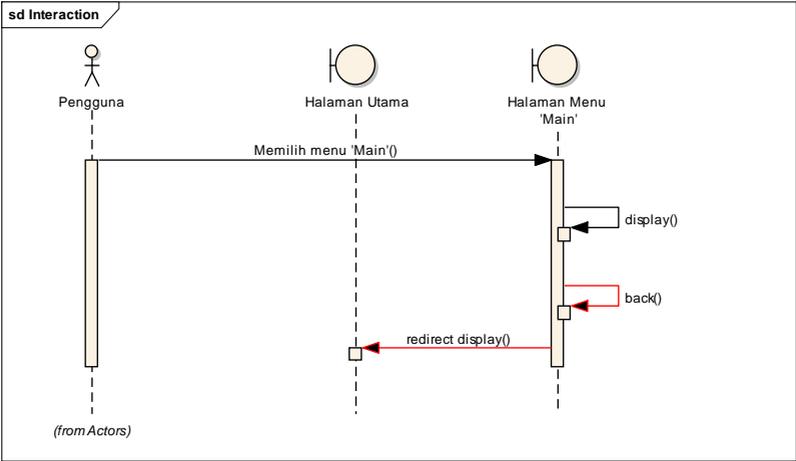
B.1 Sequence Diagram



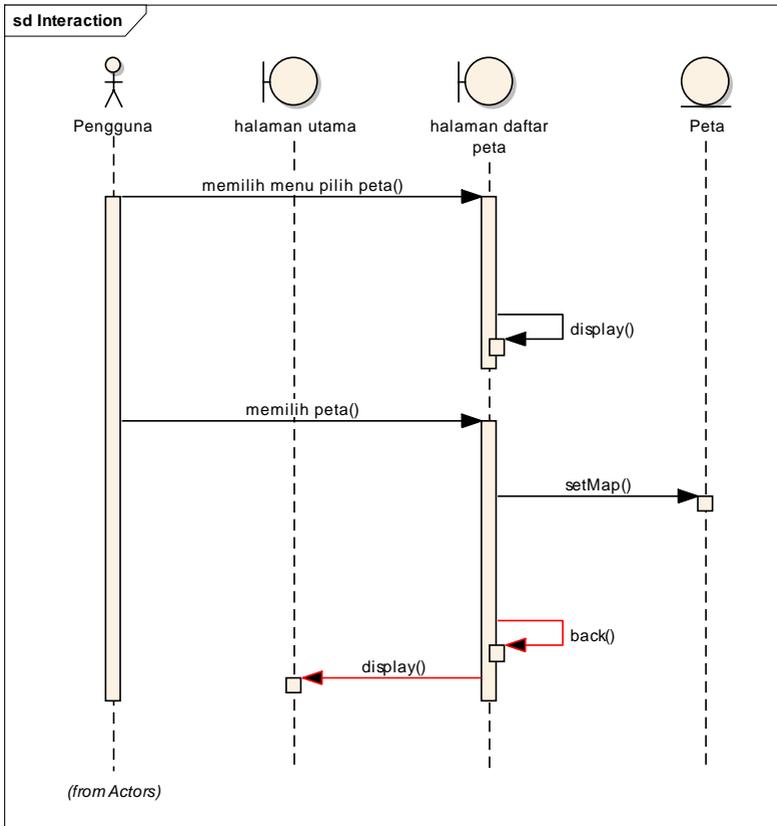
Gambar B 1 Diagram Sequence untuk UC01



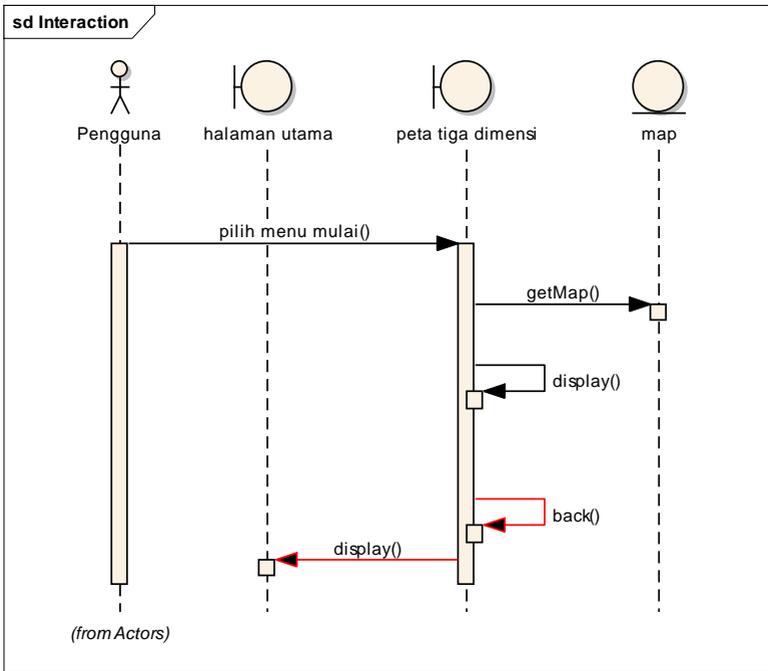
Gambar B 2 Diagram Sequence untuk UC02



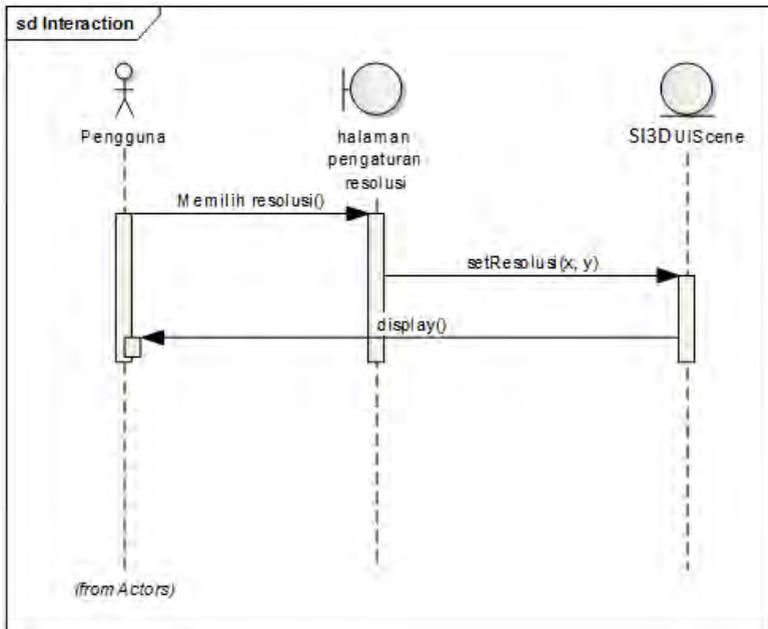
Gambar B 3 Diagram Sequence untuk UC03



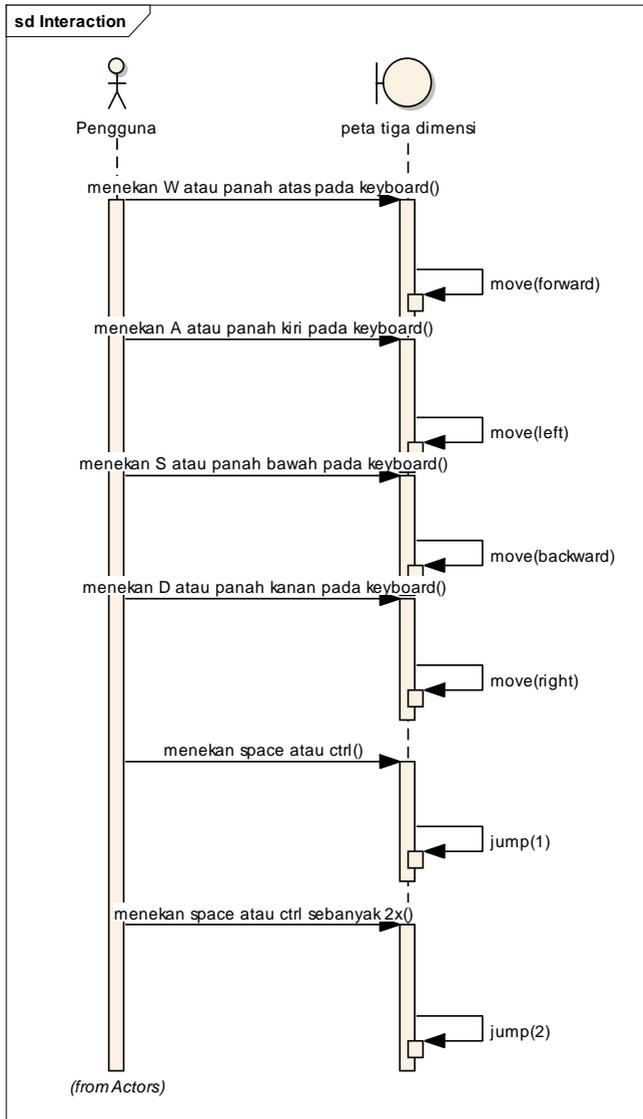
Gambar B 4 Diagram Sequence untuk UC04



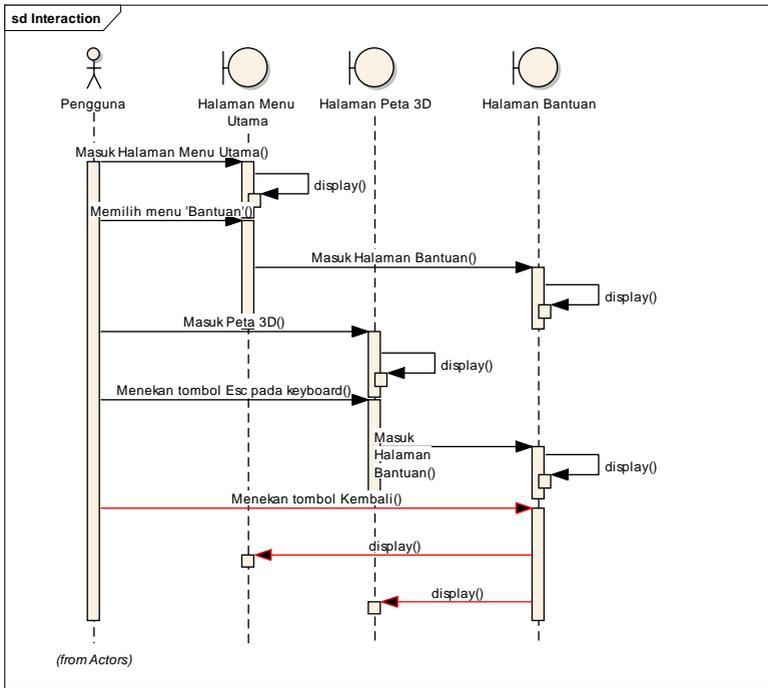
Gambar B 5 Diagram Sequence untuk UC05



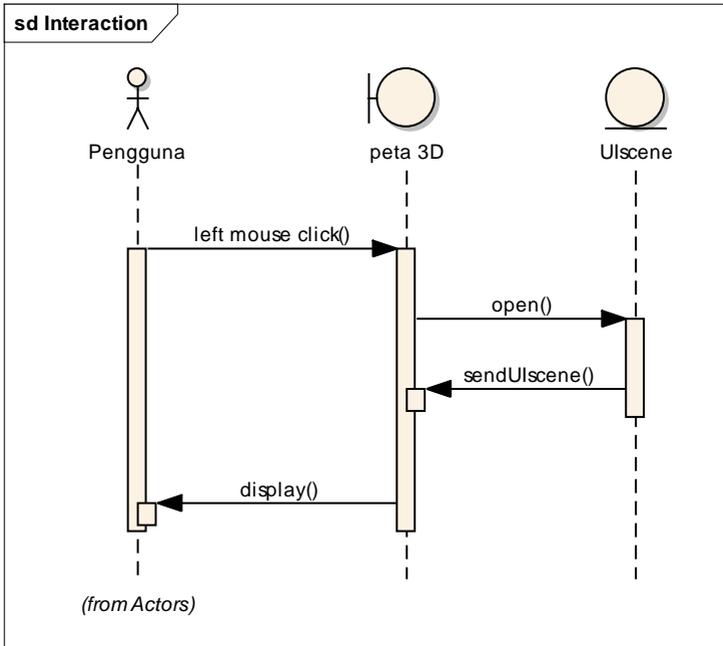
Gambar B 6 Diagram Sequence untuk UC06



Gambar B 7 Diagram Sequence untuk UC07



Gambar B 8 Diagram Sequence untuk UC08



Gambar B 9 Diagram Sequence untuk UC09

LAMPIRAN C
TEST CASE

C.1. Test Case Interaksi Dengan Obyek

Tabel C.1. Test Case Interaksi Dengan Obyek

ID	Skenario	Masuk Peta 3D	Menekan tombol mouse kiri	Hasil
TC01	Pengguna berhasil berinteraksi dengan obyek	V	V	Sistem menampilkan pesan interaksi yang dapat terjadi dengan suatu obyek. Sistem akan menjalankan fungsi interaksi pada obyek tersebut.
TC02	Pengguna tidak menekan tombol apapun	V	N/A	Sistem menampilkan informasi, tetapi pengguna tidak dapat berinteraksi dengan obyek.

C.2. Test Case Melihat Peta 2 Dimensi

Tabel C.2. Test Case Melihat Peta 2 Dimensi

ID	Skenario	Masuk Peta 3D	Menekan tombol M	Hasil
TC01	Melihat peta 2 dimensi	V	V	Sistem menampilkan peta 2 dimensi.

C.3. Test Case Memilih Menu Jelajah

Tabel C.3. Test Case Memilih Menu Jelajah

ID	Skenario	Masuk halaman utama	Memilih menu main	Memilih menu keluar	Memilih menu kembali	Hasil
TC01	Memilih menu	V	V	N/A	N/A	Sistem menampilkan menu main (untuk melakukan pemilihan Peta 3D yang akan dilihat)
TC02	Memilih menu keluar	V	N/A	V	N/A	Sistem menutup halaman session kemudian keluar aplikasi.

TC03	Memilih menu kembali	V	N/A	N/A	V	Sistem menutup pilihan menu kemudian kembali ke halaman aplikasi
------	----------------------	---	-----	-----	---	--

C.4. Test Case Mengubah Resolusi

Tabel C.4. Test Case Mengubah Resolusi

ID	Skenario	Masuk halaman utama	Memilih menu Resolusi	Memilih resolusi yang digunakan	Menekan tombol OK	Menekan tombol Batal	Hasil
TC01	Berhasil mengubah resolusi	V	V	V	V	N/A	Sistem akan memproses perubahan resolusi kemudian kembali ke halaman utama.
TC02	Batal mengubah resolusi	V	V	V	N/A	V	Sistem tidak melakukan proses perubahan resolusi kemudian mengembalikan pengguna ke halaman utama.

C.5. Test Case Navigasi

Tabel C.5. Test Case Navigasi

ID	Skenario	Masuk Peta 3D	Menekan arrow up	Menekan arrow left	Menekan arrow right	Menekan arrow down	Hasil
TC01	Navigasi depan	V	V	N/A	N/A	N/A	Aktor pengguna dalam peta bergerak maju.
TC02	Navigasi samping kanan	V	N/A	N/A	V	N/A	Aktor pengguna dalam peta bergerak ke kanan
TC03	Navigasi samping kiri	V	N/A	V	N/A	N/A	Aktor pengguna dalam peta bergerak ke kiri
TC04	Navigasi samping bawah	V	N/A	N/A	N/A	V	Aktor pengguna dalam peta bergerak mundur

C.6. Test Case Memilih Peta

Tabel C.6. Test Case Memilih Peta

ID	Skenario	Masuk Peta 3D	Memilih menu peta	Memilih menu kembali	Hasil
TC01	Memilih peta	N/A	V	N/A	Sistem menampilkan daftar peta kemudian memindah pengguna ke peta yang telah dipilih
TC02	Kembali ke Peta 3D	V	N/A	V	Sistem tidak melakukan proses pemindahan lokasi pengguna dalam peta kemudian kembali ke halaman peta yang sedang aktif.

C.7. Test Case Menjelajahi Peta

Tabel C.7. Test Case Menjelajahi Peta

ID	Skenario	Halaman Utama	Memilih menu Mulai	Hasil
TC01	Pengguna mulai eksplorasi peta	V	V	Sistem akan meload peta (default).

C.8. Test Case Melihat Bantuan

Tabel C.8. Test Case Melihat Bantuan

ID	Skenario	Masuk Halaman Peta 3D	Masuk Halaman Menu Utama	Menekan tombol Esc pada keyboard	Memilih Menu Bantuan	Memilih Menu Kembali	Hasil
TC01	Melihat Halaman Bantuan dari Halaman Peta 3D	V	N/A	V	N/A	N/A	Sistem menampilkan halaman Bantuan.
TC02	Melihat Halaman Bantuan dari Halaman Menu Utama	N/A	V	N/A	V	N/A	Sistem menampilkan halaman Bantuan.
TC03	Kembali ke Peta 3D dari Halaman Bantuan	N/A	N/A	V	N/A	N/A	Sistem menampilkan halaman Peta 3D.

TC04	Kembali ke Menu Utama	N/A	N/A	N/A	N/A	V	Sistem menampilkan halaman Menu Utama.
------	-----------------------	-----	-----	-----	-----	---	--

C.9. Test Case Mengaktifkan Layar Informasi

Tabel C.9. Test Case Mengaktifkan Layar Informasi

ID	Skenario	Masuk Peta 3D	Menekan tombol mouse kiri	Hasil
TC01	Pengguna berhasil mengaktifkan layar informasi	V	V	Sistem menampilkan layar informasi berupa alur interaksi obyek.
TC02	Pengguna tidak menekan tombol apapun	V	N/A	Sistem hanya menampilkan informasi