



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS 091336

**PENGEMBANGAN PETA INTERAKTIF TIGA
DEMENSI GEDUNG NaSEDEC INSTITUT
TEKNOLOGI SEPULUH NOPEMBER SURABAYA
MENGUNAKAN UNREAL ENGINE**

**ADNAN MAULANA
NRP 5209 100 138**

**Dosen Pembimbing I:
Dr.Eng. Febriliyan Samopa, S.Kom, M.Kom**

**JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014**



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - KS 091336

**DEVELOPMENT OF 3D INTERACTIVE MAP ON
NaSDEC BUILDING INSTITUT TEKNOLOGI
SEPULUH NOPEMBER SURABAYA WITH
UNREAL ENGINE**

**ADNAN MAULANA
NRP 5209 100 138**

**Supervisor I:
Dr.Eng. Febriliyan Samopa, S.Kom, M.Kom**

**INFORMATION SYSTEM DEPARTEMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2014**

**PENGEMBANGAN PETA INTERAKTIF TIGA
DEMENSI GEDUNG NaSDEC INSTITUT
TEKNOLOGI SEPULUH NOPEMBER SURABAYA
MENGUNAKAN UNREAL ENGINE**

Nama Mahasiswa : Adnan Maulana
NRP : 5209 100 138
Jurusan : Sistem Informasi FTIf-ITS
**Dosen Pembimbing: Dr. Eng. Febriliyan Samopa, S.Kom,
M.Kom**

Abstrak

Beberapa tahun terakhir ini, perkembangan dunia bisnis dan teknologi berkembang sangat pesat, sehingga mendukung solusi teknologi informasi sebagai kebutuhan bisnis. Dengan berkembangnya teknologi membuat seseorang bisa membangun aplikasi berbentuk visual tiga dimensi (3D). Aplikasi berbentuk visual 3D ini bisa digunakan sebagai media promosi untuk memperkenalkan sesuatu yang dapat terdeskripsi dengan baik dan jelas bagi beberapa perusahaan.

Dalam tugas akhir ini dibuat aplikasi peta tiga dimensi yang menampilkan gedung NaSDEC ITS dengan menggunakan Unreal Development Kit (UDK). Aplikasi ini dikembangkan dimulai dari survei data, desain aplikasi, pembuatan peta sampai dengan evaluasi hasil aplikasi. Dalam pembuatan peta tiga dimensi ini nantinya akan dibangun dengan aplikasi pendukung seperti 3D Studio Max untuk membuat model benda 3, Adobe Flash untuk membuat tampilan animasi, Adobe Photoshop untuk membuat texture dan material, dan Sketchup untuk membuat peta dua dimensi.

Hasil dari tugas akhir ini berupa informasi-informasi penting mengenai jurusan Gedung NaSDEC ITS, interaksi dengan pengguna, fitur penunjuk arah, dan lain lain. Aplikasi peta 3D ini menyajikan tampilan gedung Gedung NaSDEC ITS yang memudahkan pengguna untuk mengetahui dan mengunjungi gedung tersebut secara virtual.

Kata Kunci : Unreal Engine, Peta Tiga Dimensi, Gedung NaSDEC Institut Teknologi Sepuluh Nopember Surabaya.

DEVELOPMENT OF 3D INTERACTIVE MAP ON NaSDEC BUILDING INSTITUT TEKNOLOGI SEPULUH NOPEMBER SURABAYA WITH UNREAL ENGINE

**Name : Adnan Maulana
NRP : 5209 100 138
Department : Sistem Informasi FTIf-ITS
Supervisor : Dr. Eng. Febriliyan Samopa, S.Kom, M.Kom**

Abstract

The last few years, the development of business and technology is growing very rapidly, so that as information technology solutions to support business needs. With the development of technology makes people can build applications visually in the form of three-dimensional (3D). Application visual 3D can be used as a media promotion to introduce something that can description well for some companies.

In this thesis made an application that displays three-dimensional map building NaSDEC ITS using the Unreal Development Kit (UDK). This application was developed starting from the survey data, the design of the application, map-making application to the evaluation results. In making this three-dimensional map will be constructed with supporting applications such as 3D Studio Max for modeling objects 3, Adobe Flash to create an animated display, Adobe Photoshop to create textures and materials, and Sketchup to make a two-dimensional map.

The results of this thesis in the form of important information about the Department of Building NaSDEC ITS, interaction

with users, features directions, and others. This application presents a 3D map view building NaSDEC ITS that allows users to find and visit the virtual building.

Keywords: Unreal Engine, Three Dimensional Map Building NaSDEC Institute of Technology Surabaya.

**PENGEMBANGAN PETA INTERAKTIF TIGA
DIMENSI GEDUNG NaSDEC INSTITUT
TEKNOLOGI SEPULUH NOPEMBER
MENGUNAKAN UNREAL ENGINE**

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

ADNAN MAULANA

5209 100 138

Surabaya, Juli 2014

**KETUA
JURUSAN SISTEM INFORMASI**

DR.ENG. FEBRILIYAN SAMOPA, S.KOM, M.KOM

NIP 197302191998021001

**PENGEMBANGAN PETA INTERAKTIF TIGA
DIMENSI GEDUNG NaSDEC INSTITUT
TEKNOLOGI SEPULUH NOPEMBER
MENGUNAKAN UNREAL ENGINE**

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

pada

Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

ADNAN MAULANA

5209 100 138

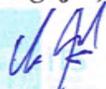
Disetujui Tim Penguji : Tanggal Ujian : 08 Juli 2014
Periode Wisuda : September 2014

Dr.Eng. Febriliyan Samopa, S.Kom, M.Kom (Pembimbing I)

Faizal Johan Atletiko, S.Kom, M.T


(Penguji I)

Nisfu Asrul Sani, S.Kom., M.Sc


(Penguji II)

KATA PENGANTAR

Puji syukur sebesar-besarnya penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya kepada penulis sehingga dapat menyelesaikan Laporan Tugas Akhir ini yang berjudul: “PENGEMBANGAN PETA INTERAKTIF TIGA DIMENSI GEDUNG NaSDEC INSTITUT TEKNOLOGI SEPULUH NOPEMBER MENGGUNAKAN UNREAL ENGINE”. Penulis menyadari bahwa penelitian Tugas Akhir ini masih belum sempurna. Oleh karena itu penulis berharap agar penelitian Tugas Akhir ini dapat dikembangkan lebih baik lagi di kemudian hari. Dalam pengerjaan Tugas Akhir ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

- Orang tua tercinta yang selalu memberikan semangat, doa dan dukungan moral selama penulis berusaha menyelesaikan tugas akhir.
- Bapak Dr. Eng. Febriliyan Samopa, S.Kom, M.Kom selaku pembimbing 1 yang telah meluangkan waktu untuk membimbing penulis dalam menyelesaikan Tugas Akhir penulis.
- Bapak Sholiq, ST. MT selaku dosen wali yang telah memberikan pengarahan kepada penulis dalam merencanakan studi penulis di Jurusan Sistem Informasi selama ini.
- Semua Bapak dan Ibu Dosen Pengajar beserta staf dan karyawan di Jurusan Sistem Infomasi, FTIF ITS

Surabaya yang telah memberikan bantuan kepada penulis selama ini.

- PT Angkasa Pura II yang telah memberikan beasiswa kepada penulis, dengan beasiswa tersebut penulis merasa terbantu dalam menyelesaikan studinya.
- Teman-teman AE9IS yang telah memberikan pertemanan yang begitu erat. Terima kasih atas segala dukungan dan doa yang telah diberikan.
- E-business Family dan Tim INI3D 2013 atas kebersamaannya selama ini mengerjakan tugas akhir bersama-sama yang selama ini telah membantu, berjuang, bergadang dan belajar bersama dalam pengerjaan tugas akhir ini.
- Seluruh kakak, teman, dan adik di Sistem Informasi ITS. Terima kasih atas persaudaraan, dukungan dan doa yang telah diberikan.

Juga terima kasih penulis ucapkan kepada semua pihak yang tidak dapat disebutkan satu persatu yang telah ikut membantu baik secara langsung maupun tidak langsung selama penulisan Tugas Akhir ini.

Surabaya, Juni 2014

Penulis

DAFTAR ISI

Abstrak.....	vii
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xxi
1 BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Permasalahan.....	4
1.3 Batasan Masalah/Ruang Lingkup.....	4
1.4 Manfaat.....	4
1.5 Sistematika Penulisan.....	5
2 BAB II TINJAUAN PUSTAKA	7
2.1 Peta Interaktif Tiga Dimensi ITS.....	7
2.2 Game Engine.....	10
2.3 Unreal Engine	11
2.3.1 Unreal Editor.....	11
2.3.2 Speed Tree Modeler dan Speed Tree Compiler	24
2.3.3 Unreal Frontend	25
2.4 Konfigurasi Aplikasi.....	26
2.5 Perangkat Lunak Pembuat Peta 2D.....	27
2.6 Perangkat Lunak Modeling 3D.....	27
2.7 Perangkat Lunak Pengelola Gambar.....	28
2.8 Perangkat Lunak Pengolah Suara	28
2.9 Perangkat Lunak Pendukung	29
3 BAB III METODOLOGI.....	31
3.1 Studi Literatur	33
3.2 Survey Lokasi dan Pengambilan Data	34
3.3 Perancangan dan Pedesainan Peta.....	34
3.4 Pembuatan Peta.....	34
3.5 Integrasi	35
3.6 Pembuatan Laporan	36

4 BAB IV DESAIN APLIKASI	37
4.1 Interaksi	37
4.2 Domain Model.....	40
4.3 Use Case Diagram.....	42
4.4 Sequence Diagram	43
4.5 Test case	47
4.6 Pemilihan Tombol Navigasi dan Kontrol.....	47
4.7 GUI Story Board Menu Awal.....	51
1. Gui Story Board Menu Awal	51
2. Tampilan Menu Awal	52
3. Tampilan Menu Menjelajahi Peta	52
4. Tampilan Menu Resolusi	53
5 BAB V IMPLEMENTASI DAN UJI COBA SISTEM.....	54
5.1 Lingkungan Implementasi.....	55
5.2 Peta Dua Dimensi.....	56
5.3 Pembuatan Aplikasi.....	56
5.3.1 Pembuatan Level Map	56
5.3.2 Pembuatan dan <i>Import</i> Objek 3D	71
5.3.3 Pengaturan Pencahayaan	73
5.3.4 Penambahan Suara	76
5.3.5 Pembuatan Interaksi	78
5.4 Integrasi	109
5.5 Uji Coba dan Evaluasi.....	110
5.5.1 Uji Coba Fungsional.....	110
5.5.2 Uji Coba Non-Fungsional.....	111
5.5.3 Evaluasi Implementasi	116
6 BAB VI KESIMPULAN DAN SARAN.....	119
6.1 Kesimpulan	119
6.2 Saran	120
DAFTAR PUSTAKA.....	121
BIODATA PENULIS.....	127
LAMPIRAN A DIAGRAM DAN DESKRIPSI USE CASE.1	
A.1 Diagram Use Case.....	A-3
A.2 Deskripsi Use Case Interaksi dengan Obyek ...	A-3

A.3 Deskripsi Use Case Melihat Peta 2 Dimensi.....	A-4
A.4 Deskripsi Use Case Navigasi.....	A-5
A.5 Deskripsi Use Case Menjelajahi Peta.....	A-6
A.6 Deskripsi Use Case Mengaktifkan Layar Informasi	
A-7	
LAMPIRAN B TEST CASE.....	B-1
B.1. Test Case Interaksi Dengan Obyek.....	B-3
B.2. Test Case Melihat Peta 2 Dimensi.....	B-4
B.3. Test Case Navigasi.....	B-4
B.4. Test Case Menjelajahi Peta.....	B-5
B.5. Test Case Mengaktifkan Layar Informasi.....	B-6

DAFTAR TABEL

Tabel 4. 1 Daftar interaksi.....	38
Tabel 4. 2 Analisa pemilihan tombol navigasi dan kontrol ...	48
Tabel 5. 1 Spesifikasi Perangkat Keras dan Sistem Operasi untuk Implementasi Sistem	55
Tabel 5. 2 Perangkat Lunak yang Digunakan	55
Tabel 5. 3 Nilai Vektor Rotation pada Properti Aktor	95
Tabel 5. 4 Unit Test dari Rancangan Test Case pada Lampiran C.....	111
Tabel 5. 5 Spesifikasi Komputer 1.....	111
Tabel 5. 6 Spesifikasi Komputer 2.....	112
Tabel 5. 7 Spesifikasi Komputer 3.....	112
Tabel 5. 8 Spesifikasi Komputer 4.....	112
Tabel 5. 9 Hasil Uji Coba.....	113
Tabel 5. 10. Spesifikasi Rekomendasi untuk Penelitian Selanjutnya	115
Tabel 5. 11 Validasi Peta 3D dengan Keadaan Nyata	117
Tabel A. 1 Deskripsi Use Case Interaksi dengan Obyek.....	3
Tabel A. 2 Deskripsi Use Case Melihat Peta 2 Dimensi	4
Tabel A. 3 Deskripsi Use Case Navigasi	5
Tabel A. 4 Deskripsi Use Case Menjelajahi Peta	6
Tabel A. 5 Deskripsi Use Case Mengaktifkan Layar Informasi	7
Tabel B. 1 Test Case Melihat Peta 2 Dimensi.....	3
Tabel B. 2 Test Case Melihat Peta 2 Dimensi.....	4
Tabel B. 3 Test Case Navigasi.....	4
Tabel B. 4 Test Case Menjelajahi Peta.....	5
Tabel B. 5 Test Case Mengaktifkan Layar Informasi	6

DAFTAR GAMBAR

Gambar 2. 1 Antarmuka pengguna Unreal Editor, editor dari Unreal Development Kit	14
Gambar 2. 2 Contoh <i>sequence</i> sederhana di dalam Kismet ..	15
Gambar 2. 3 <i>Unreal Matinee</i> merupakan editor Matinee di dalam <i>UnrealKismet</i>	16
Gambar 2. 4 Matinee di dalam UnrealKismet	16
Gambar 2. 5 Contoh penggunaan Unreal Static Mesh Editor	17
Gambar 2. 6 Unreal Material Editor.....	19
Gambar 2. 7 Particle system sebuah kunang - kunang.....	20
Gambar 2. 8 Skeletal Mesh aktor yang telah diberi Material	22
Gambar 2. 9 Pembuatan animasi aktor dalam <i>Unreal Animset Editor</i>	23
Gambar 2. 10 Penggunaan <i>Unreal Sound Cue Editor</i>	24
Gambar 2. 11 Penggunaan Speed Tree Modeler untuk pembuatan pohon	25
Gambar 2. 12 Antarmuka Unreal Frontend	26
Gambar 3. 1 Gedung NaSDEC.....	31
Gambar 3. 2 Metodologi Pengerjaan Tugas Akhir	33
Gambar 4. 1 Domain Model Awal.....	40
Gambar 4. 2 Domain model akhir	41
Gambar 4. 3 Diagram Sequence untuk UC01	43
Gambar 4. 4 Diagram Sequence untuk UC02	44
Gambar 4. 5 Diagram Sequence untuk UC03	44
Gambar 4. 6 Diagram Sequence untuk UC04.....	45
Gambar 4. 7 Diagram Sequence untuk UC05	46
Gambar 4. 8 Diagram Sequence untuk UC06.....	47
Gambar 4. 9 GUI story board menu awal.....	51
Gambar 4. 10 Tampilan Menu Awal.....	52

Gambar 4. 11 Tampilan Menu Menjelajahi Peta	52
Gambar 4. 12 Tampilan Menu Resolusi.....	53
Gambar 5. 1 Peta dua dimensi Gedung.....	56
Gambar 5. 2 Built mengalami gangguan.....	57
Gambar 5. 3 Builder Brush Cube.....	59
Gambar 5. 4 Properti Red Builder Brushes	60
Gambar 5. 5 Mode Wireframe Viewport Perspective Unreal Editor	61
Gambar 5. 6 Mode Wireframe Viewport Top Unreal Editor	62
Gambar 5. 7 Contoh Material pada Content browser	63
Gambar 5. 8 Penggunaan Unreal Material Editor untuk Material Dinding.....	64
Gambar 5. 9 Hasil Pemasangan Material	65
Gambar 5. 10 FluidSurface	66
Gambar 5. 11 Air Menggunakan Fluid Surface	67
Gambar 5. 12 Particle System untuk Interaksi Lampu	68
Gambar 5. 13 Particle System untuk Interaksi Objek.....	68
Gambar 5. 14 Particle System untuk informasi	69
Gambar 5. 15 Tanaman Hasil Speed Tree dan Static mesh ..	70
Gambar 5. 16 Aktor Wanita pada SkeletalMesh	71
Gambar 5. 17 Tampilan Default 4 Perspektif Autodesk 3Ds Max.....	72
Gambar 5. 18 Setting Import Objek 3D UDK	73
Gambar 5. 19 Macam-macam Aktor Light pada Unreal Editor	74
Gambar 5. 20 Penggunaan DominantDirectionalLight pada peta gedung NaSDEC ITS	75
Gambar 5. 21 Konfigurasi Dominant Directional Light.....	75
Gambar 5. 22 Melakukan Convert Suara Menggunakan Audacity	76
Gambar 5. 23 File Sound Node Wave dan Sound Cue	77
Gambar 5. 24 Sound Cue pada Unreal Kismet.....	78
Gambar 5. 25 Animasi Flash Layar Informasi	80
Gambar 5. 26 Alur Animasi Flash Layar Informasi.....	81

Gambar 5. 27 Action Script pada Animasi Flash Layar Informasi.....	82
Gambar 5. 28 Kismet Interaksi Prosedur Penyerahan form pelamaran kerja.....	82
Gambar 5. 29 Animasi Flash menyalakan atau mematikan keran air.....	84
Gambar 5. 30 Animasi Flash menyalakan keran air	84
Gambar 5. 31 RemoteEvent pada keran air	85
Gambar 5. 32 ActivateRemoteEvent pada Interaksi keran air	86
Gambar 5. 33 Remote Event pada Interaksi Pintu.....	87
Gambar 5. 34 Active Remote Event pada Interaksi Pintu....	87
Gambar 5. 35 Matinee Interaksi Simulasi Penyewaan Ruang Seminar di gedung NaSDEC ITS	88
Gambar 5. 36 Kismet Interaksi Simulasi Penyewaan Ruang Seminar di gedung NaSDEC ITS	89
Gambar 5. 37. Pengambilan Nilai Properti Location Aktor..	90
Gambar 5. 38. Perbandingan x-axis dan y-axis peta 3D Unreal Engine dan <i>Flash</i> Adobe <i>Flash</i> CS5 (Tim INI3D, 2012).....	91
Gambar 5. 39 Hasil dari rumus (1) pada peta 2D.....	93
Gambar 5. 40 Pengambilan Nilai Properti Rotation Aktor...	94
Gambar 5. 41 Animasi <i>Flash</i> Menu Peta Dua Dimensi	97
Gambar 5. 42 Struktur Utama Proyek Animasi <i>Flash</i> Menu Peta Dua Dimensi.....	98
Gambar 5. 43 Potongan Kode <i>Action Script</i> Animasi <i>Flash</i> Menu Peta Dua Dimensi	99
Gambar 5. 44 Potongan Kode <i>Action Script</i> Animasi <i>Flash</i> Menu Peta Dua Dimensi Scene 1 Layer <i>Action Frame</i> 1	100
Gambar 5. 45 Pilihan Fungsi Teleportasi pada Menu Peta Dua Dimensi.....	101
Gambar 5. 46 Kode <i>Action Script</i> Animasi <i>Flash</i> Menu Peta Dua Dimensi <i>Symbol</i> peta2d_mc Layer <i>Frame</i> 1	102
Gambar 5. 47 Kode <i>Action Script</i> Animasi <i>Flash</i> Menu Peta Dua Dimensi pada <i>Symbol Button</i> Penanda Lokasi	103

Gambar 5. 48 Kode <i>Action Script</i> Animasi <i>Flash</i> Menu Peta Dua Dimensi pada <i>Symbol Button</i> BTeleport	104
Gambar 5. 49 Kismet Teleportasi.....	105
Gambar 5. 50 Kismet Mengambil Posisi Tangga	106
Gambar 5. 51 <i>Static mesh</i> Langit pada Mode Wireframe ...	107
Gambar 5. 52 Matinee Simulasi Pergantian Siang dan Malam	108
Gambar 5. 53. Simulasi Pergantian Siang dan Malam	108
Gambar 5. 54 Integrasi Peta Gedung NaSDEC, ROBOTIKA, dan TC.....	109
Gambar 5. 55 Hasil Integrasi Peta gedung NaSDEC dan ROBOTIKA.....	110
Gambar 5. 56 Grafik Perbandingan Hasil Uji Coba	114
Gambar A. 1 Diagram Use Case.....	3

BAB I

PENDAHULUAN

Pada bab ini, akan dijelaskan tentang Latar Belakang Masalah, Perumusan Masalah, Batasan Masalah, Tujuan Tugas Akhir, dan Relevansi atau Manfaat Kegiatan Tugas Akhir.

1.1 Latar Belakang Masalah

Penggunaan Teknologi Informasi mengalami perkembangan yang pesat saat ini menyebabkan kebutuhan akan penyajian informasi dalam hal bangunan juga mengalami perkembangan. Para pakar-pakar teknologi informasi dituntut untuk menciptakan sebuah karya yang baru sehingga dapat memenuhi kebutuhan yang diinginkan oleh masyarakat. Salah satunya yaitu pengembangan penyajian informasi menggunakan teknologi 3D, mulai dari game, film, termasuk juga dalam hal penyajian informasi bangunan. Penggunaan teknologi 2D memiliki gambaran yang berbeda dengan keadaan yang sebenarnya. Sehingga para pengembang membuat teknologi informasi dengan menggunakan teknik visualisasi 3D untuk menggambarkan keadaan yang terlihat sesuai dengan yang sebenarnya.

Penggunaan teknologi 3D dahulu membutuhkan biaya yang besar dalam penerapannya. Karena sifat keingintahuan manusia sangat besar membuat perusahaan berlomba menciptakan inovasi teknologi yang baru. Dahulu manusia jika ingin mengetahui suatu tempat mereka harus datang ke tempat tersebut. Maka dari itu dibutuhkan suatu teknologi yang dapat membantu masyarakat untuk mengunjungi tempat yang akan dikunjungi tanpa harus datang ke tempat itu. Dengan semakin berkembangnya teknologi penggunaan teknologi 3D bisa didapatkan dengan biaya yang relatif lebih murah atau bisa juga dengan penggunaan aplikasi yang bersifat open source.

Institut Teknologi Sepuluh November (ITS) memiliki *National Ship Design and Engineering Center* (NasDEC) atau Pusat Desain dan Rekayasa Kapal Nasional. Proyek besar itu merupakan hasil kerja sama ITS dengan Departemen Perindustrian. Pada 29/1 2006, gedung NasDEC diresmikan Menteri Perindustrian RI Fahmi Idris. NasDEC di kampus ITS tersebut baru pertama ada di Indonesia. Sampai 2009, NasDEC masih dikelola bersama dengan Departemen Perindustrian.

NasDEC tersebut menyediakan jasa pembuatan desain kapal. Mulai konsep, detail, hingga gambar produksi. Desain kapal itu pun beragam. Di antaranya, kapal nelayan, tanker, penyeberangan, penumpang barang, pengawas, patroli, dan kontainer. Selain desain kapal, NasDEC didirikan untuk penelitian dan pengembangan desain produk kapal yang inovatif.

Pembangunan proyek NasDEC itu diharapkan bisa menjadi sarana kegiatan terstruktur, sistematis, dan terorganisasi secara nasional. Desain dan rekayasa kapal tersebut mampu bersaing di pasaran nasional dan internasional dari aspek mutu, biaya, dan waktu,

Pada tugas akhir ini penulis membangun peta interaktif 3D gedung NaSDEC ITS menggunakan teknologi Unreal Development Kit (UDK). UDK merupakan game engine yang mampu membuat peta 3D sesuai dengan dunia nyata. Game engine ini banyak digunakan oleh perusahaan-perusahaan perkembangan game. Engine ini juga bersifat open source sehingga dapat dikembangkan oleh siapa saja. Pada pembuatan tugas akhir ini juga dengan didasarkan dengan sudah dibuatnya peta interaktif 3D dengan menggunakan teknologi yang sama. Sehingga peta 3D ITS bisa menjadi lebih lengkap dan bisa dimanfaatkan. Dengan menggunakan teknologi unreal engine ini diharapkan aplikasi ini dapat memberikan informasi secara akurat dan tepat pada gedung NaSDEC ITS. Sehingga membantu pengguna untuk melihat isi gedung secara keseluruhan tanpa harus datang ke gedung tersebut. Dan diharapkan juga dengan menggunakan engine ini memudahkan pengguna dalam berinteraksi bebas terhadap objek yang ada ada dalam peta.

1.2 Rumusan Permasalahan

Adapun permasalahan yang akan dibahas adalah:

1. Bagaimana membuat peta 3D yang terstandarisasi, sehingga dapat di integrasikan dengan peta lainnya yang sudah dibuat sebelumnya?
2. Bagaimana membangun peta 3D yang informatif dengan menggunakan Unreal Engine?
3. Bagaimana mengembangkan peta 3D yang interaktif, sehingga pengguna dapat berinteraksi dengan obyek-obyek yang ada di dalam peta?

1.3 Batasan Masalah/Ruang Lingkup

Melihat kompleksnya permasalahan yang ada pada Pengembangan Peta Interaktif Tiga Dimensi maka dirasa perlu dilakukan pembatasan masalah, sebagai berikut:

1. Aplikasi yang dikembangkan tidak mencakup hubungan interaksi antar pengguna.
2. Aplikasi yang dikembangkan tidak dapat diubah oleh pengguna.
3. Aplikasi ini tidak akan menggambarkan daerah yang dilarang oleh pihak yang berkaitan (Gedung NaSDEC ITS).
4. Standarisasi aplikasi yang digunakan menggunakan standarisai dengan tim UDK sebelumnya.

1.4 Manfaat

Manfaat dari penulisan tugas akhir ini adalah ITS akan memiliki aset peta 3D yang dapat membantu pengunjung ITS mengelilingi semua area ITS tanpa harus berjalan jauh dan dapat digunakan sebagai media promosi kepada masyarakat jauh yang tidak bisa berkunjung langsung ke ITS, sehingga mereka memiliki pengalaman menjelajah area ITS. Pengalaman dapat diatur sesuai keadaan sebenarnya melalui kesamaan koordinat gedung, pencahayaan matahari, dan tekstur obyek.

1.5 Sistematika Penulisan

Sistematika penulisan buku tugas akhir dibagi menjadi 6 bab sebagai berikut :

BAB I

PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan permasalahan, batasan masalah/ruang lingkup, tujuan, manfaat dan sistematika penulisan.

BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan istilah-istilah yang digunakan pada penulisan buku tugas akhir ini serta dasar teori yang digunakan pada tugas akhir ini.

BAB III

METODOLOGI

Bab ini membahas alur dan tata pengerjaan tugas akhir dari awal sampai selesainya tugas akhir.

BAB IV

DESAIN APLIKASI

Bab ini menjelaskan rancangan desain aplikasi yang dibuat berdasarkan kebutuhan sistem. Desain tersebut digunakan untuk pembangunan aplikasi pada tugas akhir ini.

BAB V

IMPLEMENTASI DAN UJI COBA SISTEM

Bab ini menjelaskan pembangunan aplikasi yang sesuai dengan desain. Selain itu, dijelaskan pula uji coba sistem dalam menjaga performa aplikasi.

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang dapat diambil dari tugas akhir ini dan saran untuk kelanjutan sistem.

BAB II

TINJAUAN PUSTAKA

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

2.1 Peta Interaktif Tiga Dimensi ITS

Peta interaktif 3D ITS atau biasa disebut dengan INI3D merupakan kepanjangannya dari *ITS Now In 3D*. INI3D pertama kali dikembangkan pada tahun 2010 oleh Bagit Airlangga. Peta 3D pertama yang dibuat adalah gedung jurusan Sistem Informasi ITS, dan berlanjut ke pembuatan gedung jurusan-jurusan lainnya oleh pengembang penerusnya. Adapun standarisasi yang digunakan untuk pengembangan INI3D, yaitu :

- Skala 1 meter pada ukuran sebenarnya sama dengan 64 di dalam UDK
- Standarisasi FPS
- UDK yang digunakan adalah UDK versi Februari 2012
- Tampilan langit dengan menggunakan pergantian siang dan malam yang sama dengan waktu 1 jam = 1 menit
- Pembulatan ukuran dibulatkan kebawah tanpa koma
Contoh : 64,45 menjadi 64
- Interaksi standar yang harus ada, antara lain :

- ✓ **Membuka dan menutup pintu**
- ✓ **Menyalakan dan mematikan lampu**

Berikut adalah daftar penelitian yang telah dilakukan oleh tim INI3D sebelumnya :

Tabel 2. 1 Daftar penelitian INI3D sebelumnya

No.	Nama	Lokasi	Referensi
1	Ahmad Bangun Reza Pahlevi	Teknik Fisika	(Pahlevi, 2011)
2	Singgih Setyo Jatmiko	Teknik Elektro	(Jatmiko, 2011)
3	Ade Rachmat Subakti	Teknik Perkapalan	(Subakti, 2012)
4	Panditya Wirangga	Sistem Perkapalan	(Wirangga, 2011)
5	Anita Safitri	Desain Produk	(Safitri, 2011)
6	Nyoman Bagus Prasetia	Teknik Kelautan	(Prasetia, 2011)
7	Dimas Azzahrawani Putra	Puskom dan Gedung BAUK ITS	(Putra D. A., 2011)
8	Ayu Fitri	Teknik Informatika	(Fitri, 2011)
9	Rakhmat Agzati Putra	Teknik Kimia	(Putra R. A., 2011)
10	Safira Nur Assyifa	Teknik Arsitektur	(Assyifa, 2011)
11	Titus Irma Damaiyanti	Teknik Arsiterktur	(Damaiyanti, 2011)

12	Yuli Aria	Rektorat dan Pascasarjana	(Winata, 2011)
13	Azlan Mufti	Teknik Lingkungan	(Mufti, 2011)
14	Fitriannisa Umami	D3 FTI	(Umami, 2011)
15	Bagit Airlangga	Sistem Informasi	(Airlangga, 2011)
16	Zinzia Shavira Pitra Haryananda	Gedung BAAK ITS	(Haryananda, 2011)
17	Fitrah Meilia Purnama	Teknik Sipil	(Purnama, 2011)
18	Chanif Samsyir Saputra	Biologi	(Saputra, 2012)
19	Yeranata Listyadana	Kimia	(Listyadana, 2012)
20	Ludfi Eka	Fisika	(Lesmana, 2012)
21	Kandy Rudyanti	Matematika	(Rudyanti, 2012)
22	Edo Rachmansyah	Graha Sepuluh Nopember ITS dan UPT Bahasa	(Rachmansyah, 2012)
23	Moch. Nur Yasin	Sistem Informasi	(Yasin, 2012)
24	Dimas Aryana	Statistika	(Aryana, 2012)
25	Reza Putra Jawara	Robotika dan Laboratorium Energi	(Jawara, 2013)

26	Made Yudi Pradita	ITS Kampus Manyar	(Pradita, 2013)
27	Mochammad Rahman Irwandi	Darmawanita, TK ITS dan Wisma Yasmin	(Irwandi, 2013)

2.2 GameEngine

Game Engine adalah sistem perangkat lunak yang dirancang untuk menciptakan dan pengembangan video game. Ada banyak mesin permainan yang dirancang untuk bekerja pada konsol permainan video dan sistem operasi desktop seperti Microsoft Windows, Linux, dan Mac OS X. fungsionalitas inti biasanya disediakan oleh mesin permainan mencakup mesin render (“renderer”) untuk 2D atau 3D grafis, mesin fisika atau tabrakan (dan tanggapan tabrakan), suara, script, animasi, kecerdasan buatan, jaringan, streaming, manajemen memori, threading, dukungan lokalisasi, dan grafik (Arita, 2011). Proses pengembangan permainan sering dihemat oleh sebagian besar menggunakan kembali mesin permainan yang sama untuk menciptakan permainan yang berbeda.

Engine bukanlah executable program, artinya engine tidak bisa dijalankan sebagai program yang berdiri sendiri (Bullen, 2006). Diperlukan sebuah program utama sebagai entry point atau titik awal jalannya program. Pada C++, entry point-nya adalah fungsi ‘main().’ Biasanya program utama ini relatif pendek.

Beberapa game dengan jenis dan gameplay yang hampir sama bisa dibuat dengan sedikit usaha bila terlebih dulu dibuat engine-nya. Setelah engine diselesaikan, programmer hanya perlu menambahkan program utama, memakai resources (objek 3D, musik, efek suara) yang baru, dan, jika benar-benar dibutuhkan, sedikit memodifikasi engine sesuai kebutuhan spesifik dari game yang bersangkutan. Program game engine seluruhnya berorientasi objek. Dia lebih bersifat reaktif daripada prosedural. Sulit untuk menggambarkan engine secara keseluruhan dalam flow-chart, karena alur program bisa diatur sesuai dengan keinginan pemakai engine, yaitu game programmer.

2.3 Unreal Engine

Unreal Engine merupakan sebuah engine yang biasanya digunakan untuk membuat suatu game. Unreal Engine merupakan game engine dengan popularitas kedua setelah game engine tiga dimensi serupa yaitu Source Engine. Terdapat beberapa hal yang dipertimbangkan ketika seseorang, instansi atau perusahaan memilih game engine yang akan digunakan sebagai game engine untuk mengembangkan suatu permainan atau visualisasi (Busby, 2004). Beberapa diantaranya dipilih berdasarkan lisensinya, berbayar atau tidak. Unreal Engine merupakan salah satu game engine yang lisensinya adalah proprietary, namun untuk Unreal Development Kit lisensinya adalah tidak berbayar.

Tidak semua *game engine* dapat menangani grafik dua dimensi sekaligus tiga dimensi. Terdapat *game engine* yang hanya menangani grafik dua dimensi, hanya menangani tiga dimensi, atau dapat menangani keduanya.

Unreal Engine memiliki kerangka kerja (*framework*) lengkap untuk pengembangan profesional. Sistem inti *engine* ini menggunakan bahasa pemrograman C++, tetapi untuk pengembangannya digunakan bahasa pemrograman Java. Seperti kebanyakan *game engine* lainnya, Unreal Engine dapat mengolah beberapa data seperti objek tiga dimensi, suara, tekstur, dan lain sebagainya.

Unreal Development Kit menyediakan beberapa alat yaitu *Unreal Editor*, *UnrealFrontend*, *SpeedTreeModeler*, *SpeedTree Compiler* untuk membangun sebuah *game*.

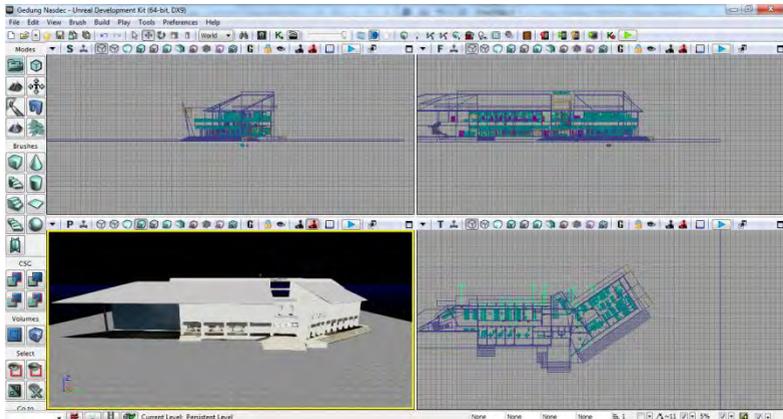
Berikut adalah modul-modul dari Unreal Engine :

- *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 obyek yang terjadi.
- *Input Manager*
Modul ini digunakan untuk mengatur input, seperti tombol ditekan, dan 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++ (Bubsy J, 2004). 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.

2.3.1 Unreal Editor

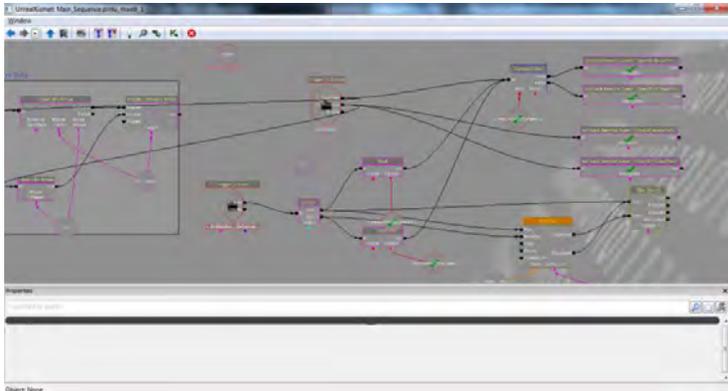
Unreal Development Kit menyediakan alat untuk membuat dunia virtual yaitu Unreal Editor. Editor ini juga bisa melakukan *import* dari perangkat lunak pembuat objek tiga dimensi yang sudah banyak digunakan seperti 3D Studio Max dan Maya. Selain itu, Unreal Editor juga menyediakan cara untuk membuat tekstur, material, suara dan objek yang ada seperti di dunia nyata. Antarmuka penggunaan Unreal Editor dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Antarmuka pengguna Unreal Editor, editor dari Unreal Development Kit

2.3.1.1 Unreal Kismet

Unreal Kismet menjadi solusi bagi pengembang dengan pengetahuan pemrograman sedikit. *Unreal Kismet* merupakan bentuk visual dari script yang sebenarnya kompleks, hingga *Unreal Kismet* menjadi tulang punggung interaksi didalam game. Dengan menggunakan object-object yang tersedia di dalamnya dan salin dihubungkan menjadi suatu modul di dalam *Unreal Kismet*, maka suatu interaksi atau semua yang akan dilakukan di dalam game kit menjadi mudah dan cepat dibuat serta dipahami. Mulai dari sequence yang sederhana yaitu untuk membuat interaksi membuka pintu hingga sequence yang kompleks seperti untuk membuat sequence teleportasi yang komplit. Dengan menyalin sequence kismet yang kita buat ke dalam notepad dan sejenisnya, maka akan terlihat scripts yang sangat kompleks sekalipun itu merupakan sequence yang sederhana.



Gambar 2. 2 Contoh *sequence* sederhana di dalam Kismet

Objek yang tersedia pada Kismet ini saling dihubungkan di dalam suatu modul Kismet. Berkat modul Kismet ini, suatu interaksi di dalam game jadi lebih mudah dan cepat untuk dibuat serta dipahami. Salah satu contoh modul yang ada di dalam Kismet dapat dilihat pada Gambar 2.2.

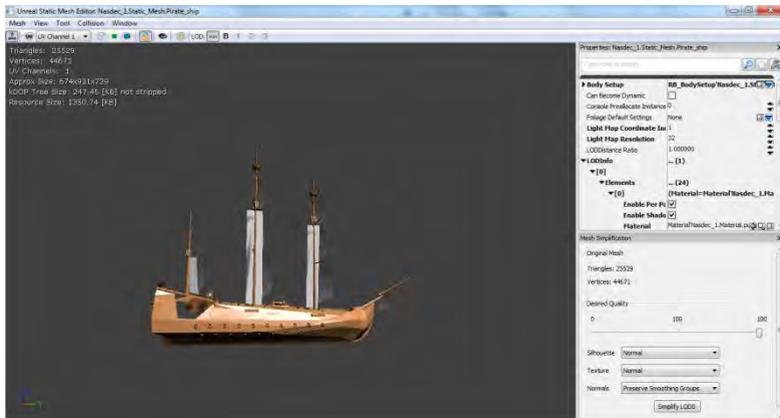
Mulai dari *sequence* yang sederhana yaitu untuk membuat interaksi membuka pintu hingga *sequence* yang kompleks seperti untuk membuat *sequence* teleportasi yang lengkap dapat dilakukan di sini.

2.3.1.2 Unreal Matinee

UnrealMatinee adalah sebuah *tools* yang disediakan oleh Unreal Editor yang dapat digunakan untuk merubah properties object seperti lokasi, rotasi, ukuran, dan lainnya. *Unreal Matinee* juga dapat digunakan untuk mengaktifkan suatu event atau kondisi tertentu dalam sebuah game yang sebelumnya telah diatur di dalam Kismet, seperti memainkan suara, membuat animasi objek, dan lainnya.

2.3.1.3 Unreal Static Mesh Editor

Unreal *StaticMesh* Editor merupakan sebuah tools 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 static mesh, sehingga nantinya bisa melakukan pengaturan properties pada objek seperti mengubah material-nya dan mengatur *collision*-nya.



Gambar 2. 5 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.

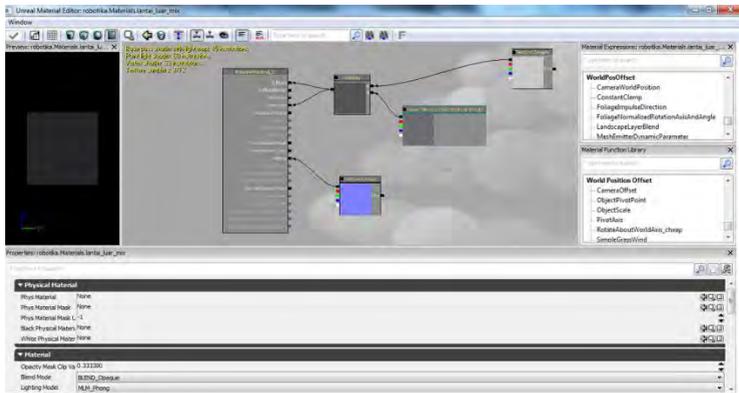
2.3.1.4 Unreal Material Editor

Ketika membuat map didalam *Unreal Engine*, pasti akan terpikirkan untuk memberikan warna atau semacamnya pada map kita agar terlihat sekedar menarik atau bahkan terlihat seperti bentuk nyatanya. Dengan menggunakan *Unreal Material Editor*, kita bisa membuat map kita terlihat menarik dan bahkan seperti bentuk nyatanya.

Kita dapat menempelkan material pada map kita, seperti pada tembok, lantai, object, dan lainnya agar terlihat lebih hidup atau nyata.

Material yang dibuat melalui Unreal Material Editor bisa hanya sekedar warna saja, bisa ditambahkan efek pantulan cahaya seperti pada logam, bisa ditambahkan efek memancarkan cahaya, atau bahkan material yang bisa bergerak. *Material* yang bisa bergerak dapat dibuat dengan

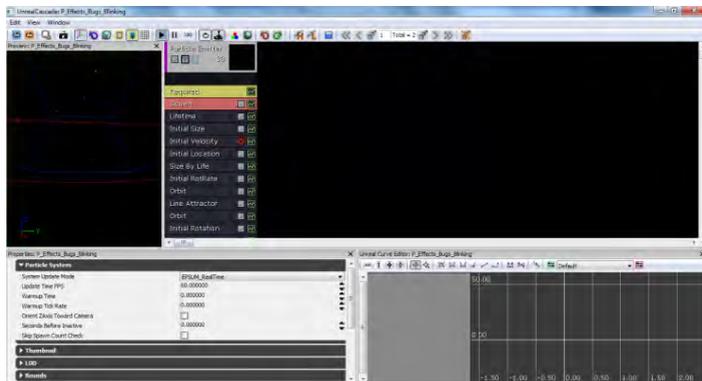
menggunakan *Material Expression* yang tersedia kemudian membuat suatu *sequence* dari kumpulan *expression* yang diatur sedemikian rupa.



Gambar 2. 6 Unreal Material Editor

2.3.1.5 Unreal Cascade

Adanya Unreal Cascade memungkinkan dibuatnya suatu *particle system* dalam sebuah game. Dengan memanfaatkan *tools* yang ada serta *Emitter list* yang tersedia, sebuah *particle system* dapat dibuat. Tapi proses ini juga tidak lepas dari pemanfaatan material dan *static mesh*. Salah satu contoh *particle system* yang dapat dibuat yaitu kunang - kunang seperti pada Gambar 2.8



Gambar 2. 7 Particle system sebuah kunang - kunang

2.3.1.6 Unreal AnimSet Editor

Unreal AnimSet berfungsi untuk mengatur aktor yang akan digunakan di dalam peta 3D. Beberapa pengaturan yang umum dapat dilakukan, seperti material aktor, AnimSet aktor serta lokasi dan rotasi aktor. Dengan memanfaatkan fungsi pengaturan material, tampilan aktor dapat dipercantik dengan memberikan warna baju, kulit, dan lainnya. Sedangkan melalui pengaturan AnimaSet aktor, dapat diberikan animasi untuknya, seperti bergerak maju, mundur, loncat dan lainnya. Selain itu, posisi aktor juga dapat ditentukan melalui pengaturan koordinat x, y dan z. Gambar 2.9 memperlihatkan aktor yang telah diberi material, siap untuk diberi animasi gerakan.

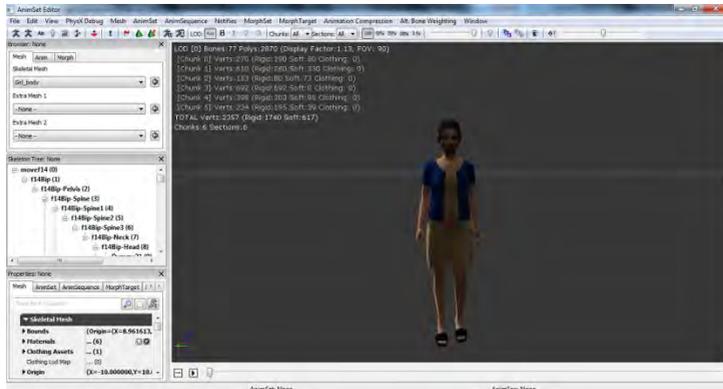
Didalam UDK, aktor yang digunakan adalah aktor bertipe first person shooter, atau dikenal juga sebagai aktor dengan sudut pandang orang pertama. Penggunaan Unreal AnimSet Editor sendiri sudah dilakukan oleh pengembang aplikasi INI3D sebelumnya. Dengan menggunakan model wanita

sebagai objek pembuatan aktor yang nantinya akan dapat dikendalikan didalam UDK maupun aplikasi INI3D. Proses pembuatan aktor didalam UDK terdiri dari beberapa tahap, diantaranya menggunakan Unreal AnimSet untuk mengatur posisi dan letak dari aktor dan Unreal AnimTree untuk mengatur kapan dan bagaimana animasi akan bergerak, serta penggunaan Unreal Script untuk memanggil default aktor, default animasi, mengatur tinggi aktor, dan mengatur kecepatan langkah aktor.

Proses pembuatan aktor didalam Unreal AnimSet sendiri, juga memerlukan tahapan tahapan yang berurutan, yaitu :

- Skin aktor yang digunakan dalam peta INI3D merupakan hasil pencarian dari free source 3D Actor Modelling
- Melakukan eksport skin aktor kedalam UDK dengan menggunakan plugin Unreal ActorX
- Setelah membuat skin aktor, dilakukan pembuatan gerakan yang nantinya akan digunakan dalam skin aktor tersebut dalam UDK
- Pemberian material adalah langkah terakhir dalam pembuatan aktor didalam Unreal AnimSet

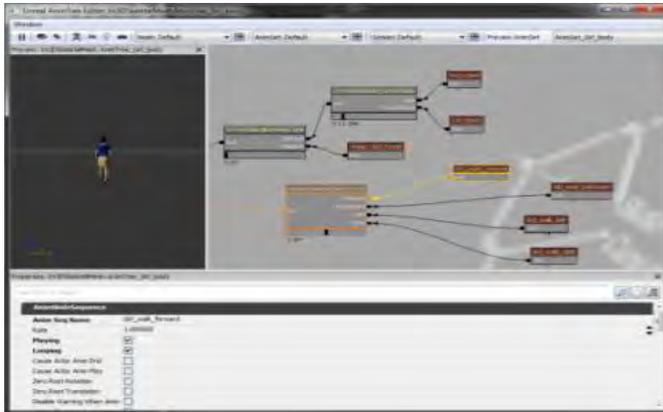
Setelah semua langkah diatas dilakukan, maka aktor yang akan digunakan didalam UDK akan tampak seperti gambar dibawah ini.



Gambar 2. 8 Skeletal Mesh aktor yang telah diberi Material

2.3.1.7 Unreal AnimTree Editor

Setelah pembuatan aktor didalam Unreal AnimSet selesai, maka selanjutnya adalah pembuatan animasi aktor dengan menggunakan Unreal AnimTree Editor. Penggunaan AnimTree untuk penambahan animasi ketika aktor berjalan dapat dilihat pada gambar 2.10, dimana pada gambar tersebut terlihat diagram Kismet yang digunakan untuk dapat membuat aktor dapat melakukan animasi berjalan.



Gambar 2. 9 Pembuatan animasi aktor dalam *Unreal Animset Editor*

2.3.1.8 Unreal Sound Cue Editor

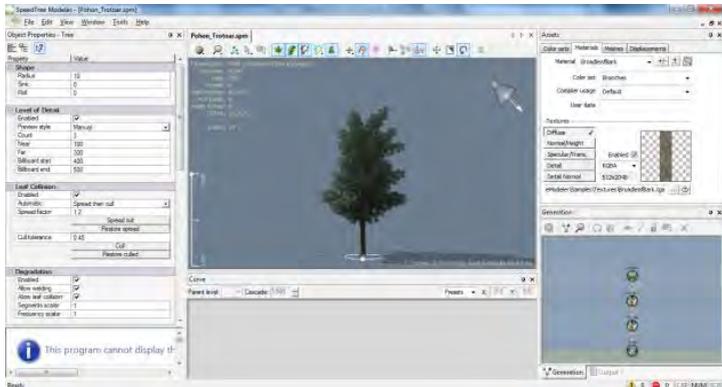
Unreal Sound Cue Editor digunakan untuk memainkan suara di dalam peta 3D. Suara yang dibuat di dalam editor ini berasal suara dengan tipe .WAV yang di-import ke dalam Content browser menjadi Sound Wave. Sound Cue ini dapat berasal dari lebih dari satu *Sound Wave* dan memberikan efek-efek yang tersedia di dalam editor ini seperti efek attenuation, random, looping dan efek lainnya sehingga terbentuk bunyi baru yang siap digunakan di dalam peta 3D. Gambar 2.11 adalah contoh penggunaan Unreal Sound Cue Editor.



Gambar 2. 10 Penggunaan *Unreal Sound Cue Editor*

2.3.2 Speed Tree Modeler dan Speed Tree Compiler

Unreal Development Kit dalam situs web resminya (*Unreal Development Kit*, 2010) juga menyediakan perangkat lunak tambahan yaitu *SpeedTree Modeler* yang berfungsi untuk membuat objek tiga dimensi tumbuh-tumbuhan. Sedangkan *Speed Tree Compiler* dapat digunakan sebagai compile objek tiga dimensi tumbuh-tumbuhan hasil dari pemodelan yang dilakukan di *Speed Tree Modeler*. Khusus untuk UDK versi Februari 2012, modul *Speed Tree Modeler* dan *Speed Tree Compiler* tidak dapat dipakai sehingga harus menggunakan file *Eyecommon.dll* dari modul *Speed Tree* yang sebelumnya. Gambar 2.12 adalah contoh penggunaan *Speed Tree Modeler*.



Gambar 2. 11 Penggunaan Speed Tree Modeler untuk pembuatan pohon

2.3.3 Unreal Frontend

Unreal Development Kit juga menyediakan Unreal Frontend (UFE). UFE merupakan sebuah alat yang menyediakan cara beragam untuk melakukan banyak tugas-tugas umum dalam ekosistem Unreal, misalnya :

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

Tanpa menggunakan UFE banyak dari tugas-tugas ini akan membutuhkan file batch terpisah yang tidak perlu dan akan meningkatkan kompleksitas alur kerja.

2.5 Perangkat Lunak Pembuat Peta 2D

Peta 2D dibuat sebelum membuat peta 3D. Berikut adalah sedikit ulasan 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 bentuk .PNG untuk dimasukkan pada Adobe *Flash* yang berfungsi sebagai penunjuk arah 2D.

2.6 Perangkat Lunak Modeling 3D

Perangkat lunak modelling 3D yang dimaksud disini adalah perangkat lunak untuk membuat objek tiga dimensi yang nantinya akan dimasukkan ke 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 modeling 3D, animasi, hingga rendering. Perangkat lunak ini adalah salah satu aplikasi modeling 3D berbayar.

2.7 Perangkat Lunak Pengelola Gambar

Pada pembuatan tugas akhir ini juga dibutuhkan perangkat lunak pengolah gambar untuk membuat material dan texture 2D dari benda-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 yang merupakan aplikasi yang digunakan untuk membuat atau editing gambar dan memberikan efek didalamnya. Photoshop ini akan menjadi salah satu tool dalam proses pembuatan tekstur peta maupun obyek peta nantinya.

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.** 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 obyek penelitian serta langkah-langkah penelitian yang dilakukan. Penelitian diawali dengan melakukan studi literatur, kemudian dilanjutkan dengan peninjauan lokasi demikian seterusnya hingga tercapai simpulan dari penelitian berbentuk buku tugas akhir.

Objek yang digunakan sebagai penelitian tugas akhir ini adalah wilayah Gedung NaSDEC ITS dimana terdapat batasan-batasan wilayah yang akan digunakan pada objek penelitian ini. Hal penting yang menjadi perhatian utama dalam pengerjaan tugas akhir ini adalah interaksi yang dapat terjadi di dalam wilayah tersebut oleh pengguna dengan benda-benda yang terdapat di dalamnya.



Gambar 3. 1 Gedung NaSDEC

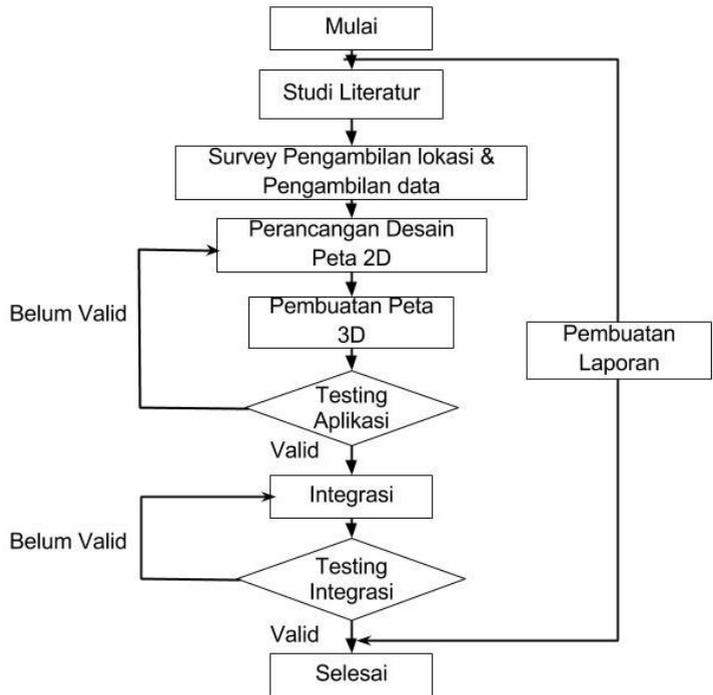
Gedung NasDEC adalah jasa pembuatan desain Kapal. Gedung NasDEC diresmikan Menteri Perindustrian RI Fahmi Idris. NasDEC di kampus ITS tersebut baru pertama ada di Indonesia. Sampai 2009, NasDEC masih dikelola bersama dengan Departemen Perindustrian.

NasDEC tersebut menyediakan jasa pembuatan desain kapal. Mulai konsep, detail, hingga gambar produksi. Desain kapal itu pun beragam. Di antaranya, kapal nelayan, tanker, penyeberangan, penumpang barang, pengawas, patroli, dan kontainer. Selain desain kapal, NasDEC didirikan untuk penelitian dan pengembangan desain produk kapal yang inovatif.

Interaksi yang dimasukkan ke dalam gedung NaSDEC:

- Buka dan tutup pintu.
- Menyalakan dan mematikan lampu.
- Interaksi di Ruang Seminar Gedung NaSDEC.
- Interaksi di Kantor NaSDEC.
- Interaksi di Ruang Lobby NaSDEC.

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 peninjauan lokasi demikian seterusnya hingga tercapai simpulan dari penelitian berbentuk buku tugas akhir. Adapun langkah-langkah penelitian yang dilakukan oleh penulis dapat dijelaskan sebagai berikut:



Gambar 3. 2 Metodologi Pengerjaan Tugas Akhir

3.1 Studi Literatur

Studi Literatur yang dilakukan dalam pembuatan tugas akhir ini 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. Studi pendahuluan dan literatur ini tidak terbatas hanya bersumber dari

buku dan jurnal, tetapi juga dari media lain seperti video tutorial ataupun sumber lain dari internet

3.2 Survey Lokasi dan Pengambilan Data

Pada tahap ini akan dilakukan pengambilan data dan survey pada lokasi gedung, ini dilakukan dengan cara memfoto 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 Perancangan dan Pedesainan 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.

3.4 Pembuatan Peta

Pada tahap ini dilakukan pembuatan peta dalam versi 3D secara keseluruhan. Semua gedung, serta obyek-obyek yang ada divisualisasikan sehingga mirip seperti aslinya, serta ditambahkan interaksi interaksi yang dapat terjadi pada setiap obyek. Semua hal tersebut dilakukan menggunakan Unreal Development Kit dari Unreal Engine. Selain pembuatan peta 3D akan ditanamkan juga interaksi-interaksi yang bisa dilakukan dengan obyek

yang ada. Berikut urutan pembuatan aplikasi ini antara lain adalah :

1. Menentukan batasan-batasan dari aplikasi yang akan dibuat,
2. Membuat level map Gedung NaSDEC ITS dengan menggunakan Unreal Editor,
3. Membuat tekstur dari bangunan menggunakan Adobe Photoshop atau aplikasi sejenis lalu memasang tekstur tersebut pada peta dengan menggunakan Unreal Editor,
4. Mengumpulkan/membuat obyek 3D menggunakan Blender atau 3D Studio Max atau mencari obyek dari internet lalu memasang obyek pada peta dengan menggunakan Unreal Editor,
5. Membuat suara dari material yang ada menggunakan Audacity atau Adobe Audition lalu memasang suara tersebut pada peta menggunakan Unreal Editor,
6. Membuat *gameplay*, menu dan logika *game* menggunakan Unreal Kismet,
7. Membuat karakter dengan menggunakan Unreal PhAT,
8. Melakukan *scripting* menggunakan bahasa pemrograman Unreal Script sehingga *default* aplikasi sesuai dengan yang penulis rencanakan,
9. Testing aplikasi,
10. Packaging aplikasi menggunakan Unreal Frontend.

3.5 Integrasi

Langkah berikutnya, adalah menyatukan semua peta menjadi satu. Dan terakhir melakukan packaging

keseluruhan peta menjadi sebuah aplikasi. Disini peta 3D berdiri sendiri-sendiri dalam satu kesatuan aplikasi yang di hubungkan dengan link ketempat-tempat lain. Integrasi bersifat tidak wajib dilaksanakan saat kondisi peta 3D yang lain tersebut belum dapat dikatakan sama standarnya atau belum terselesaikan.

3.6 Pembuatan Laporan

Tahap ini dilaksanakan ketika semua tahap metode pengerjaan diatas sudah diselesaikan. Pembuatan laporan ini dimaksudkan untuk mendokumentasikan langkah-langkah pekerjaan yang sudah dilakukan sehingga dapat memberikan informasi yang berguna bagi yang membacanya, selain itu, laporan ini juga berguna bagi pengembangan aplikasi di lain waktu. Laporan ini juga berguna untuk mengetahui apakah permasalahan yang dialami penulis sudah terselesaikan atau belum, tujuan yang diinginkan tercapai atau tidak. Selain itu laporan ini juga berisi kesimpulan dan saran untuk pengembangan aplikasi ini kedepannya.

BAB IV DESAIN APLIKASI

Pada bab ini akan dijelaskan mengenai desain aplikasi yang akan dibangun pada tugas akhir ini. Desain sistem dibuat dengan mengacu pada kebutuhan aplikasi yang dibagi menjadi 2, yaitu fungsionalitas dan non fungsionalitas. Kebutuhan fungsionalitas aplikasi didefinisikan sebagai berikut:

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

Kebutuhan non-fungsional didefinisikan seperti berikut:

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

Berikut ini desain aplikasi yang akan dibangun pada tugas akhir ini :

4.1 Interaksi

Terdapat beberapa interaksi yang harus ada pada setiap peta tiga dimensi interaktif yang dibuat.

Tabel 4. 1 Daftar interaksi

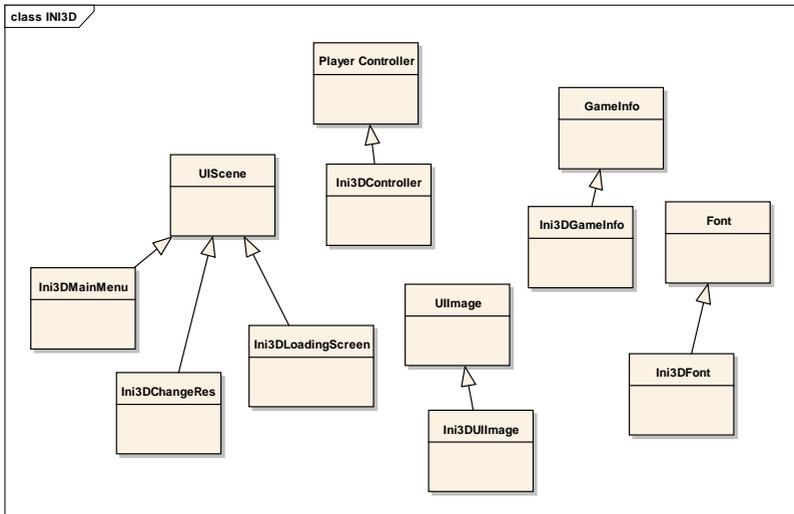
No.	Interaksi	Deskripsi
1.	Membuka pintu	Pintu dapat terbuka
2.	Menutup pintu	Pintu dapat tertutup
3.	Menyalakan lampu	Lampu dapat menyala
4.	Mematikan lampu	Lampu dapat mati
5.	Informasi ruangan	Pada setiap ruang yang memiliki interaksi terdapat penjelasan singkat dari ruangan tersebut
6.	Simulasi proses masuk gedung NaSDEC	<p>Simulasi ini dilakukan pada ruang lobby gedung NaSDEC bagi pengunjung sebagai berikut:</p> <ol style="list-style-type: none"> 1. Memberikan Kartu Tanda Penduduk(KTP) 2. Mengisi data Formulir 3. Petugas memberikan kartu pengunjung
7.	Simulasi Sewa Ruang Seminar	<p>Simulasi ini dilakukan pada ruang seminar gedung NaSDEC, dengan urutan prosedur sebagai berikut :</p> <ol style="list-style-type: none"> 1. Membuat dan menyerahkan surat izin peminjaman ruang seminar 2. Pihak pengurus melakukan verifikasi jadwal terlebih dahulu

		<ol style="list-style-type: none"> 3. Mengisi Form peminjaman yang telah disediakan 4. Membayar uang kebersihan sebesar Rp.100.000,~ 5. Ruang Seminar siap digunakan
8.	Simulasi sewa kantor	<p>Simulasi dilakukan ruang sewa kantor gedung Nasdec, dengan urutan prosedur sebagai berikut :</p> <ol style="list-style-type: none"> 1. Meminta form pengajuan form sewa kantor 2. Isikan data-data form yang tersedia 3. Baca peraturan –peraturan persyaratan sesuai form tersebut 4. Tanda tangan untuk menyetujui persyaratan yang sesuai dengan form tersebut
9.	Simulasi melihat Video Ruang Seminar	<p>Simulasi dilakukan pada gedung NaSDEC pada ruangan Seminar. Video ini tentang acara ITS sebagai salah satu institusi yang identik dengan kelautan</p>

10.	Simulasi melihat Video Profil gedung NaSDEC	Simulasi dilakukan pada gedung NaSDEC. Video ini tentang sejarah berdirinya gedung NaSDEC
-----	---	---

4.2 Domain Model

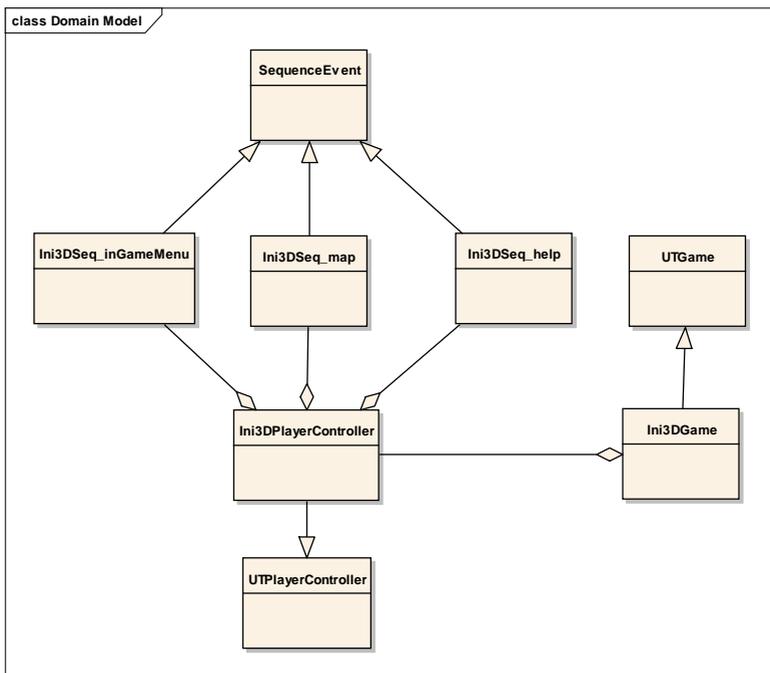
Pendefinisian *domain model* pada pengerjaan tugas akhir ini sangat penting, dikarenakan domain model menggambarkan obyek-obyek utama yang akan digunakan. Domain model dapat berubah seiring dengan pengembangan desain dan aplikasi, sehingga obyek-obyek yang digambarkan pada domain model akan semakin lengkap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



Gambar 4. 1 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. 2 Domain model akhir

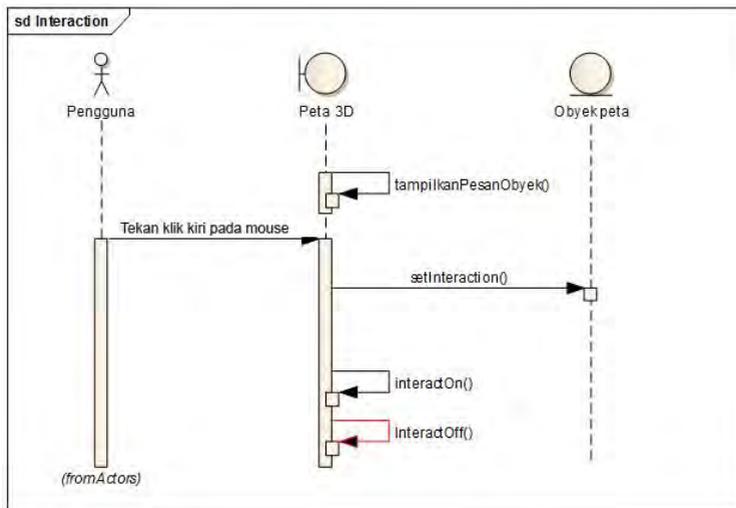
UT Player Controller adalah *class* bawaan unreal yang mengatur navigasi untuk tipe permainan UT Game. Karena terdapat beberapa navigasi tambahan, maka terdapat *class* baru dengan *parent class* UT Player Controller dengan nama Ini3D player Controller. Nantinya, *class* baru ini akan memanfaatkan *class* 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.

4.3 Use Case Diagram

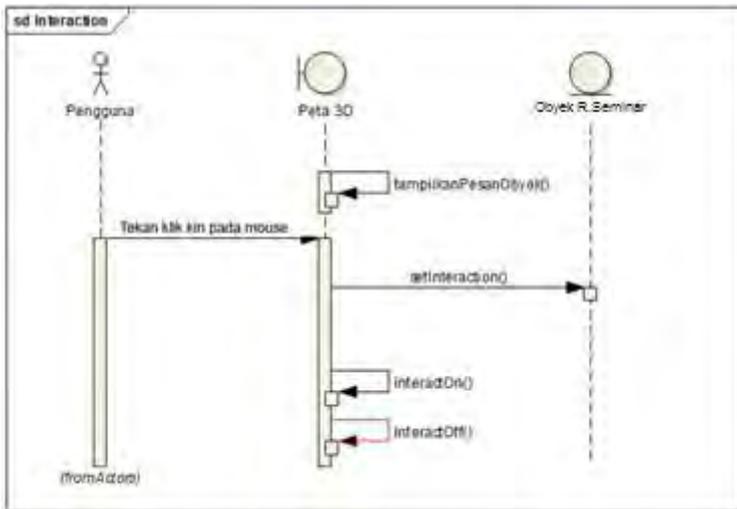
Use case yang dirancang sepatutnya memenuhi kebutuhan yang telah disebutkan sebelumnya, terutama dalam kebutuhan fungsional. Aplikasi ini memiliki *use case* standar berdasarkan penelitian peta tiga dimensi interaktif ITS yang sebelumnya telah dibuat. Rancangan diagram dan deskripsi *use case* fungsional dapat dilihat pada lampiran A.

4.4 Sequence Diagram

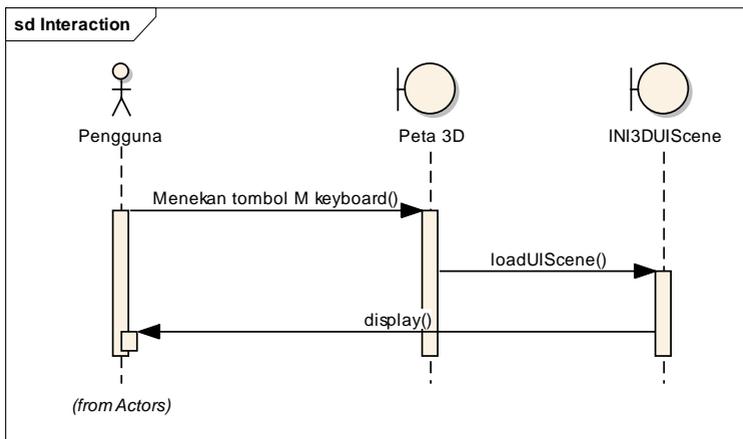
Sequence diagram memuat alur dalam use case dengan penjelasan yang mengarah pada pemrograman aplikasi, sehingga sebelum merancang sequence diagram diharuskan mengerti tentang teknologi yang akan diterapkan pada aplikasi.



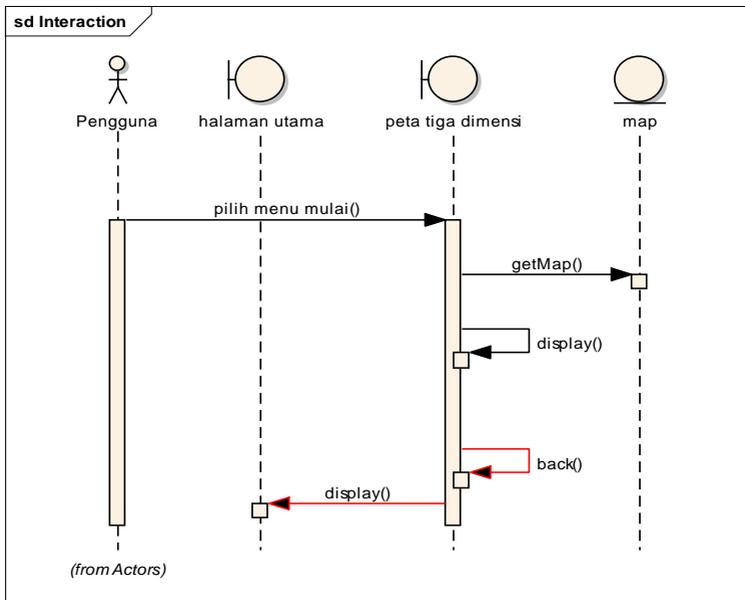
Gambar 4. 3 Diagram Sequence untuk UC01



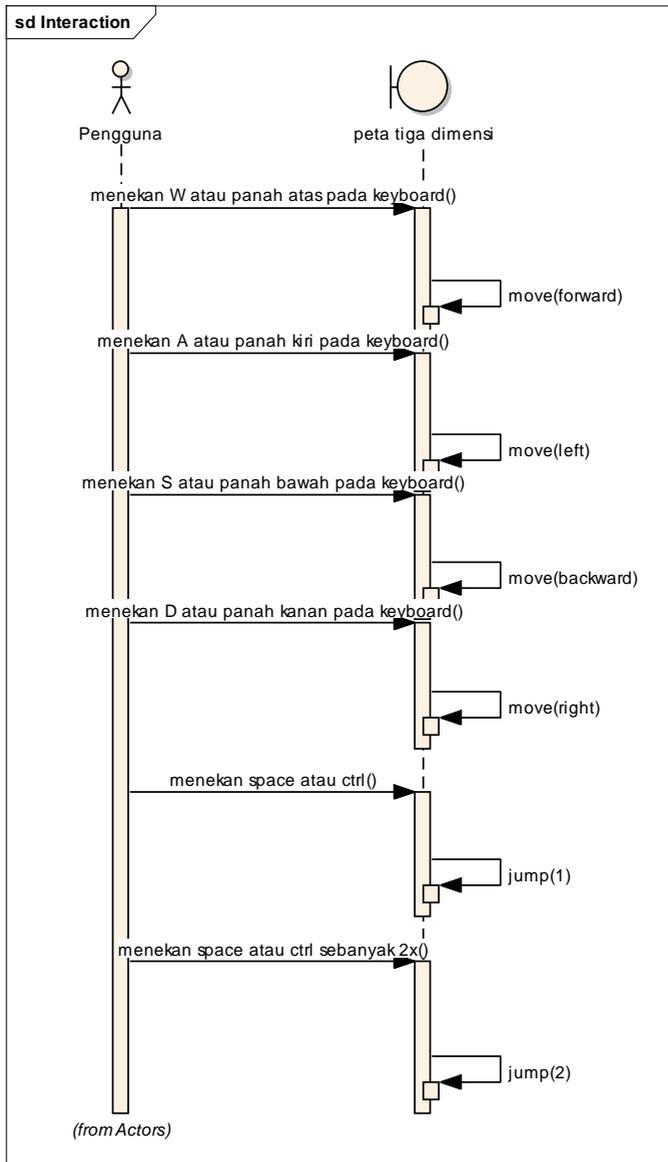
Gambar 4. 4 Diagram Sequence untuk UC02



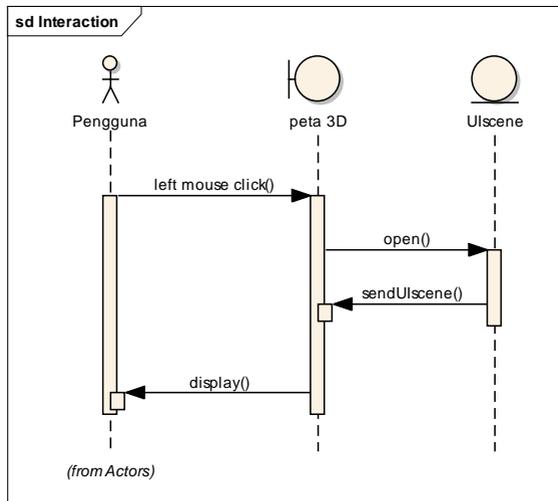
Gambar 4. 5 Diagram Sequence untuk UC03



Gambar 4. 6 Diagram Sequence untuk UC04



Gambar 4. 7 Diagram Sequence untuk UC05



Gambar 4. 8 Diagram Sequence untuk UC06

4.5 Test case

Test case dirancang untuk mengarah pada performa aplikasi agar sesuai dengan desain yang dibuat. Jadi test case akan dijalankan dengan beberapa skenario yang sesuai dengan rancangan pada diagram use case pada lampiran A. Untuk rancangan skenario dan test dapat dilihat pada lampiran B. Test case nantinya akan diuji coba berupa unit test.

4.6 Pemilihan Tombol Navigasi dan Kontrol

Peran tombol navigasi dan kontrol dalam sebuah aplikasi terutama dalam game itu sangat penting, dikarenakan peta tiga dimensi interaktif ini dikembangkan dengan game engine dan berbentuk aplikasi pembelajaran. Pada penelitian sebelumnya telah dibuat standarisasi pemilihan tombol navigasi dan kontrol. Sebagai bagian dari keseluruhan

penelitian peta tiga dimensi interaktif ITS, maka aplikasi dari penelitian ini juga menggunakan standarisasi tombol navigasi dan kontrol tersebut.

Tabel 4. 2 Analisa pemilihan tombol navigasi dan kontrol

No	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	Mem-bungkuk	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 lebih mudah untuk ditekan.
10	Melompat	Spasi	Menggerakkan tampilan seakan aktor pengguna sedang melompat	Umum dipakai pada permainan tiga dimensi

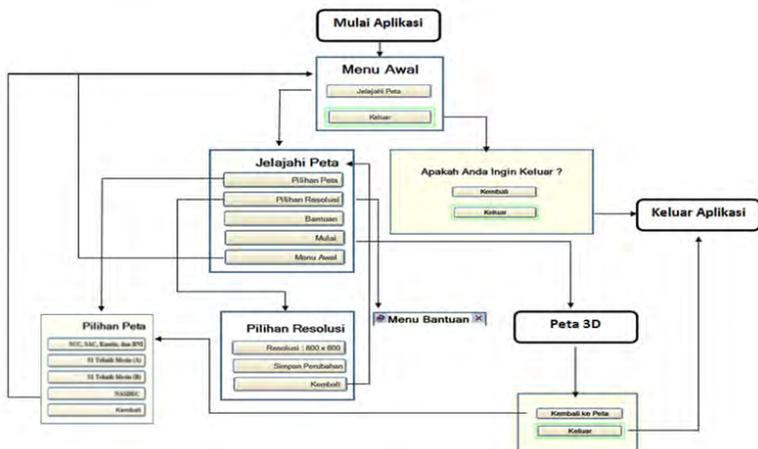
11	Ber-interaksi dengan objek peta	Enter/ Klik kiri	Menggerakkan tampilan sesuai dengan interaksi objek peta	Dipilih karena interaksi adalah hal yang paling sering digunakan dan pada game umumnya digunakan tombol mouse kiri untuk hal yang paling sering digunakan tersebut.
B	Kontrol Tingkat Peta			
1	Menu Bantuan	F1	Membuka menu Bantuan. Tekan Tombol sekali lagi untuk keluar dari menu	Umum dipakai pada permainan
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
--	--	--	--	--------------------------

4.7 GUI Story Board Menu Awal

Peran GUI Story Board yaitu memuat tampilan dan alur bagaimana aplikasi dijalankan. GUI Story Board dalam aplikasi ini memuat beberapa tampilan static dan tampilan peta tiga dimensi yang dinamis yang bisa dilihat pada lampiran G.

1. Gui Story Board Menu Awal



Gambar 4. 9 GUI story board menu awal

2. Tampilan Menu Awal



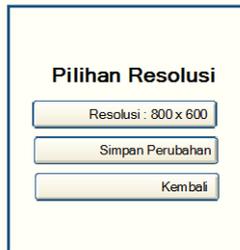
Gambar 4. 10 Tampilan Menu Awal

3. Tampilan Menu Menjelajahi Peta



Gambar 4. 11 Tampilan Menu Menjelajahi Peta

4. Tampilan Menu Resolusi



Gambar 4. 12 Tampilan Menu Resolusi

BAB V IMPLEMENTASI DAN UJI COBA SISTEM

5.1 Lingkungan Implementasi

Aplikasi ini diimplementasikan pada computer client. Spesifikasi sistem operasi dan perangkat keras yang digunakan dalam pengembangan dan pengimplementasiannya dapat dilihat pada Tabel 5.1.

Tabel 5. 1 Spesifikasi Perangkat Keras dan Sistem Operasi untuk Implementasi Sistem

Spesifikasi
Prosesor: Intel Core i5-3450 @3.10 Ghz
Memori: 8192MB RAM
VGA: GeForce GTX 560 SE
Sistem Operasi: Windows 7 Ultimate 64-bit (6.1, Build 7601)

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

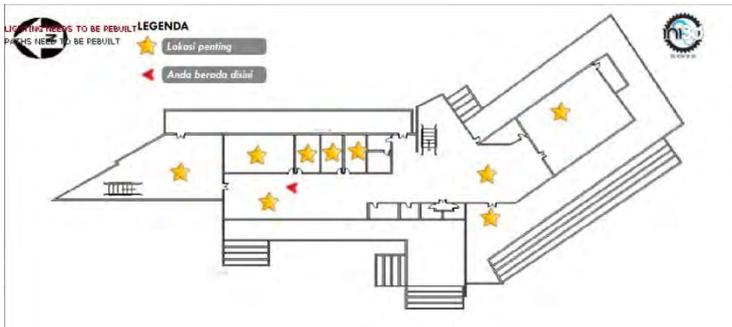
Tabel 5. 2 Perangkat Lunak yang Digunakan

Teknologi	Versi
Editor	Unreal Development Kit 64bit (Februari'12)
3D Editor	Autodesk 3ds Max 2013

Animation Editor	Adobe Flash CS5
Sound Editor	Audacity
Texture Editor	Adobe Photoshop CS5
Video Editor	IMovie

5.2 Peta Dua Dimensi

Area dari gedung NaSDEC dari peta 2D..



Gambar 5. 1 Peta dua dimensi Gedung

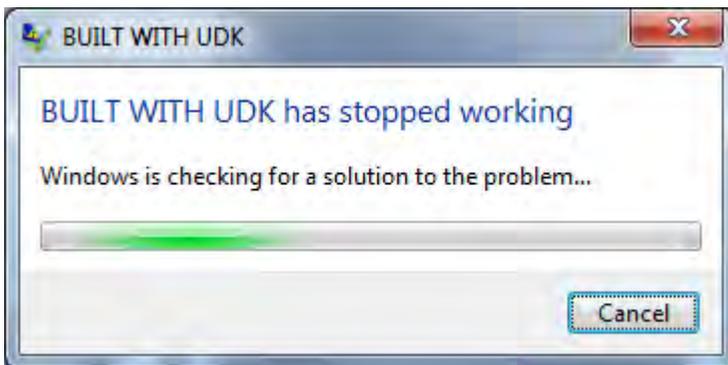
5.3 Pembuatan Aplikasi

Pada sub bab ini berisi tentang deskripsi pembuatan aplikasi berupa pembuatan *Level Map* , pembuatan objek, peletakan objek, penambahan interaksi, pencahayaan, dan penambahan suara.

5.3.1 Pembuatan Level Map

Hal pertama yang dilakukan dalam pembuatan aplikasi ini dimulai dengan pembuatan *Level Map*. *Level Map*

dibentuk berdasarkan data yang didapatkan dari hasil survey, yang berupa foto dan juga foto/video dari gedung NaSDEC ITS. Pembuatan *Level Map* ini mencakup pembuatan geometri dan pemberian material pada geometri yang dibuat. Pembuatan level map bergantung pada banyaknya bagian yang akan di pisah kan .Semakin banyak static mesh dari objek yang dibuat dan disimpan didalam level maka semakin berat aplikasi yang akan dilakukan proses built dalam aplikasi Unreal Engine. Proses Built akan mengalami gangguan berjalan nya aplikasi Unreal Engine.



Gambar 5. 2 Built mengalami gangguan

5.3.1.1 Pembuatan Constructive Solid Geometry

Pembuatan Constructive Solid Geometry (CSG) adalah hal pertama yang dilakukan pada pembuatan *Level Map*. CSG pada UDK berfungsi untuk membentuk model tiga dimensi dari peta, yang mencakup bangunan dan semua permukaan bangunan peta, seperti tangga, tembok, atap,

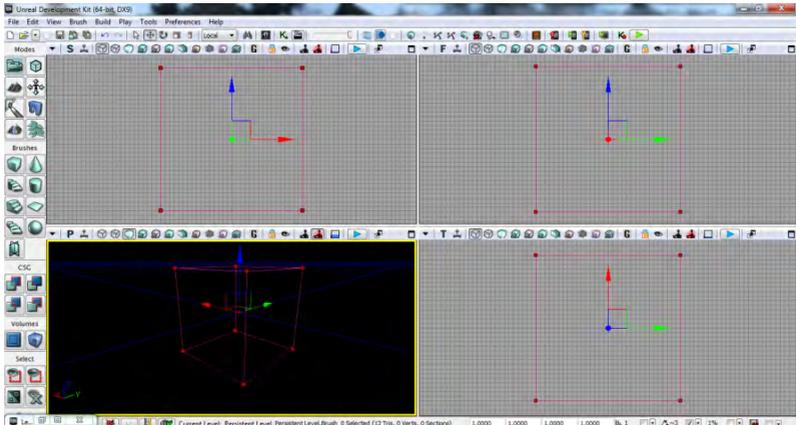
permukaan tanah/dasar gedung dan beberapa bagian dari peta yang menggunakan.

CSG dalam UDK terdapat 2 mode utama ketika membuat sebuah *Level Map* baru, yaitu *subtract*, dan *additive*. Terdapat 2 mode yang biasanya digunakan dalam membuat CSG dalam UDK Editor yaitu *additive* dan *subtract*. Untuk membuat CSG tiga dimensi kita menggunakan *additive* dan untuk memotong atau menghilangkan bagian geometri tersebut kita menggunakan *subtract*. Dalam membangun CSG *Level Map*, dalam UDK dikenal dengan istilah *brushes*. *Brushes* ini memiliki banyak bentuk seperti *cube*, *cone*, *curved staircase*, *cylinder*, *linear staircase*, *sheet*, *spiral staircase*, *tetrahedron* dan *cards*.

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

1. Menentukan *brushes* yang akan digunakan

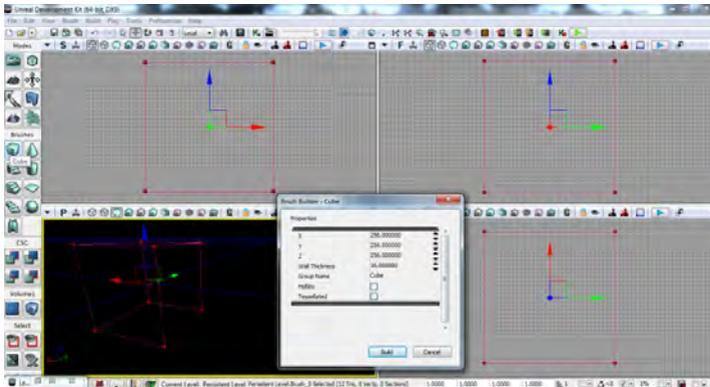
Dalam menentukan *brushes*, terlebih dahulu dipertimbangkan bentuk permukaan *Level Map* yang akan dibuat dengan *red builder brushes* dalam UDK. Seperti bentuk tembok yang harus dibuat dalam bentuk *cube* seperti pada gambar 5.2.



Gambar 5.3 Builder Brush Cube

2. Menentukan ukuran brushes

Untuk menentukan ukuran brush yang sesuai dengan ukuran *Level Map* berdasarkan hasil survey dan desain dari peta 2D yaitu dengan cara menghitung berdasarkan skala yang sudah distandarisasi. Menentukan ukuran brushes dilakukan dengan cara mengganti properti red builder brushes, seperti pada gambar 5.3.



Gambar 5. 4 Properti Red Builder Brushes

3. Menentukan jenis brushes

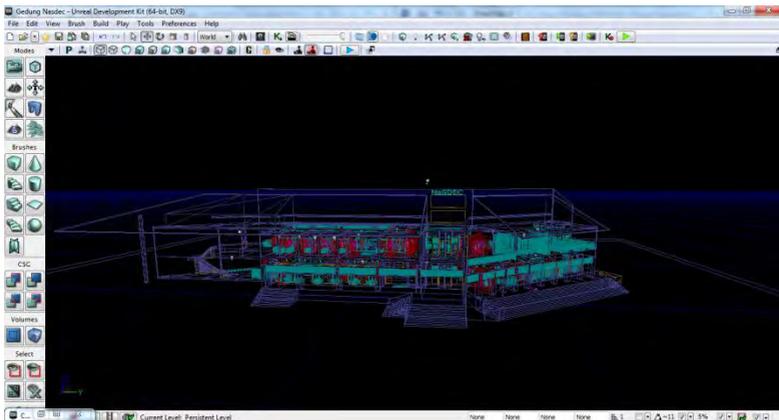
Cara menentukan jenis brushes yaitu dengan melihat fungsi dari 3 jenis brushes yang ada dalam CSG :

- **CSG add**
CSG add adalah brush dengan permukaan yang terisi penuh, dimana brush ini tidak memiliki rongga sama sekali. CSG add ini dipakai dalam membentuk permukaan dan juga tembok dari *Level Map*.
- **CSG subtract**
CSG subtract adalah brush yang digunakan untuk membuat rongga dari brush add, yang fungsinya untuk memberi lubang pada dinding untuk membuat jendela dan pintu.
- **Special brush**
Selain CSG add dan CSG subtract, terdapat Special brush seperti Lightmass Volume dan Trigger Volume. Dimana Lightmass Volume

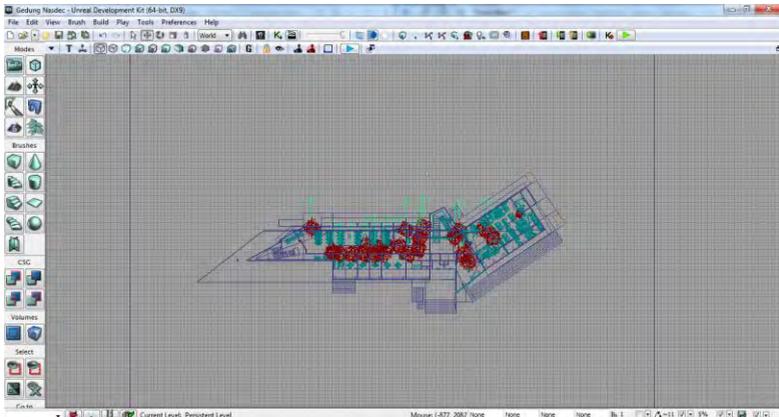
akan dijelaskan pada sub bab lightning dan Trigger Volume akan dijelaskan pada sub bab interaksi.

4. Geometry Build

Cara untuk melihat hasil brush yang di-add dalam geometri *Level Map*, harus dilakukan build geometry, sehingga akan terlihat geometri *Level Map* yang telah dibuat. Mode view dalam UDK dapat di-set ke dalam mode perspektif dan wireframe, seperti yang terlihat pada gambar 5.4 dan gambar 5.5.



Gambar 5. 5 Mode Wireframe Viewport Perspective Unreal Editor



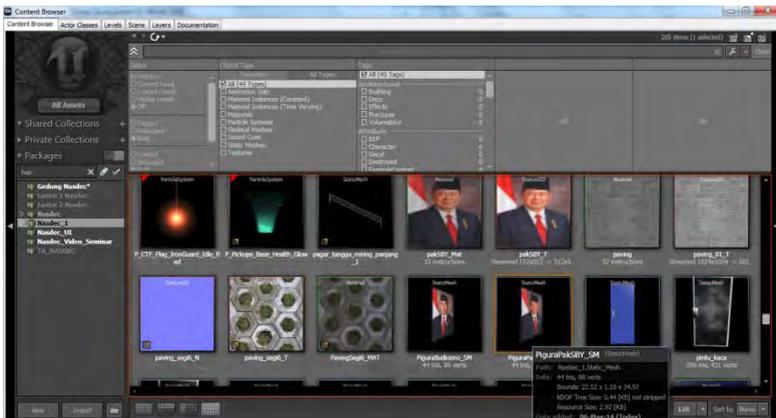
Gambar 5. 6 Mode Wireframe Viewport Top Unreal Editor

5.3.1.2 Pemberian Material

Setelah kerangka gedung jadi yang sebelumnya telah dibuat dengan brushes, maka selanjutnya adalah pembuatan tekstur dan material. Tekstur adalah bagian dari material, jadi nantinya tekstur dimasukkan ke dalam material agar bisa menempel pada brushes. Hasil jadi tekstur dan material akan disimpan ke dalam *package* UDK, jadi sebelumnya *package* harus ada. File *package* akan disimpan dalam format *.upk dan direktori *package* sendiri yaitu pada UDK\UDKGame\Content.

Tekstur berasal dari file image yang berformat *.png, kemudian di-*import* ke dalam *package* UDK. Agar hasil *import* berhasil sempurna, maka sebelumnya tekstur harus berukuran kelipatan kuadrat 2 yaitu seperti 64x64 pixel, 512x512 pixel, dan seterusnya.

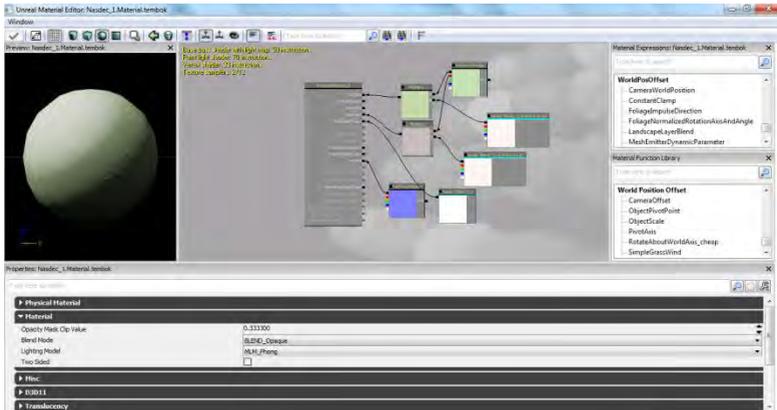
Tekstur yang dapat dipakai dalam material, dalam Unreal Material Editor terdapat di dalam channel texture sample. Tekstur ini digabungkan dalam channel diffuse dan normal untuk texture sampe normal yang di-generate dari gambar normal. Gambar 5.6 merupakan contoh tekstur yang telah di-import ke dalam package.



Gambar 5.7 Contoh Material pada Content browser

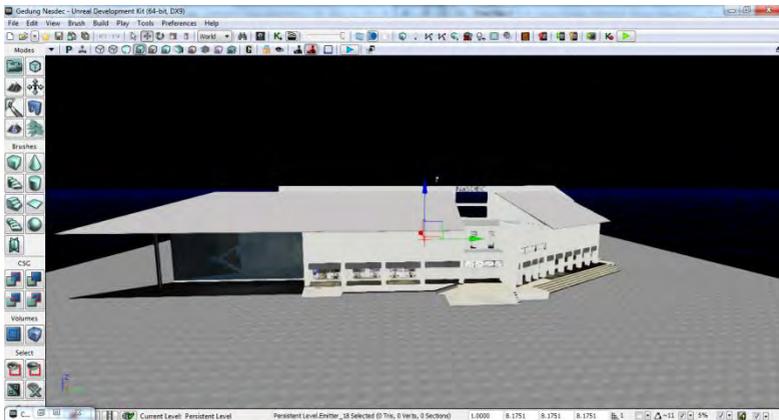
Tekstur yang di-import ini dapat diatur kembali untuk warna dan koordinat dengan cara menambahkan channel multiply dan texture coordinat. Untuk membuat Material dasar atau material sederhana, material yang melibatkan hanya satu channel yaitu texture sample yang dihubungkan ke channel diffuse material. Gambar 5.7 adalah contoh dari pengaturan dan penggunaan dari Unreal Material Editor. Untuk membuat material yang lebih kompleks, seperti material yang memiliki pantulan cahaya atau merubah warna dari material tersebut dapat

dibuat dengan menggunakan multiple channel yang ada dalam Unreal Material Editor.



Gambar 5. 8 Penggunaan Unreal Material Editor untuk Material Dinding

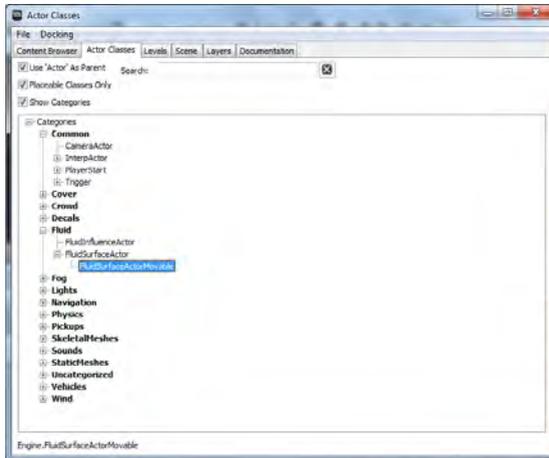
Material yang telah jadi akan diaplikasikan ke permukaan brush dan objek. Untuk menambahkan material pada permukaan brush, dilakukan dengan apply material yang terpilih pada permukaan brush. Pemasangan material pada masing-masing brush ditunjukkan pada gambar 5.8.



Gambar 5. 9 Hasil Pemasangan Material

5.3.1.3 Pemberian Fluid Surface

Pada Unreal Engine, tersedia *Fluid Surface* (pada gambar 5.9) yang berguna untuk menggambarkan efek air di *Level Map*, penggunaannya pada kolam dan selokan air yang ada di gedung NaSDEC ITS. Selain menambahkan efek tampilan air, *Fluid Surface* juga memiliki efek berupa riakan air sehingga air yang ada di *Level Map* terlihat lebih nyata. *Fluid Surface* 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. 10 FluidSurface

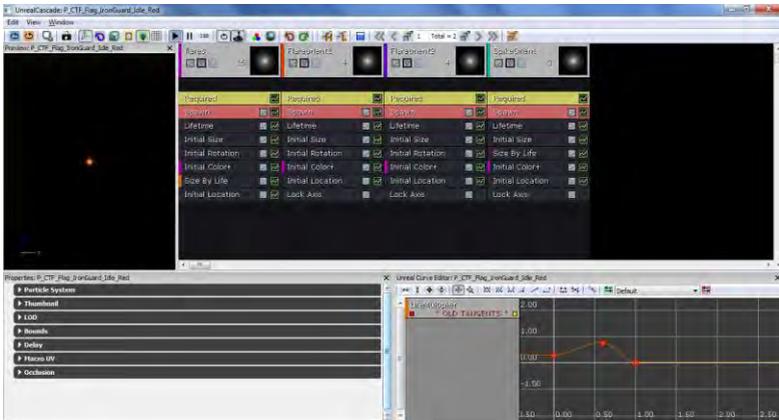
Setelah menambahkan Fluid Surface Actor ke dalam *Level Map*, selanjutnya melakukan pemberian material yang sudah disediakan pada Actor tersebut. Selain itu juga dapat melakukan banyak pengaturan lain seperti mengatur nilai Fluid Noise Frequency yaitu riakan dari cairan tersebut, memberikan Terrain, dan pengaturan lainnya di dalam Fluid Surface Actor Properties. Actor ini digunakan untuk memberikan efek air pada saluran air dan kolam yang dapat dilihat contohnya untuk kolam pada gambar 5.10.



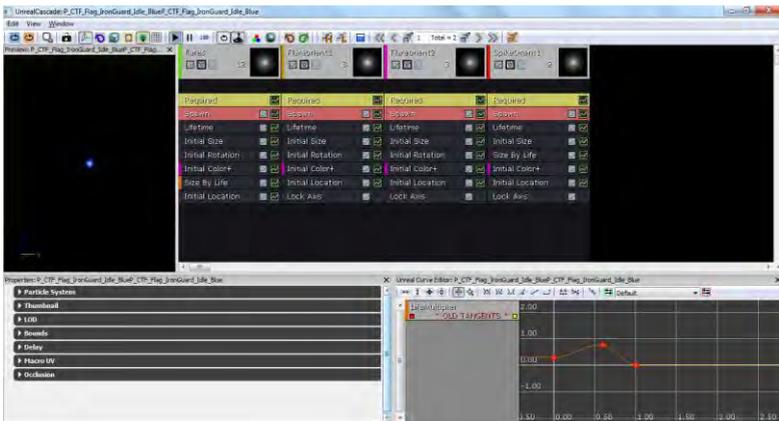
Gambar 5. 11 Air Menggunakan Fluid Surface

5.3.1.4 Pemberian Tanda Interaksi

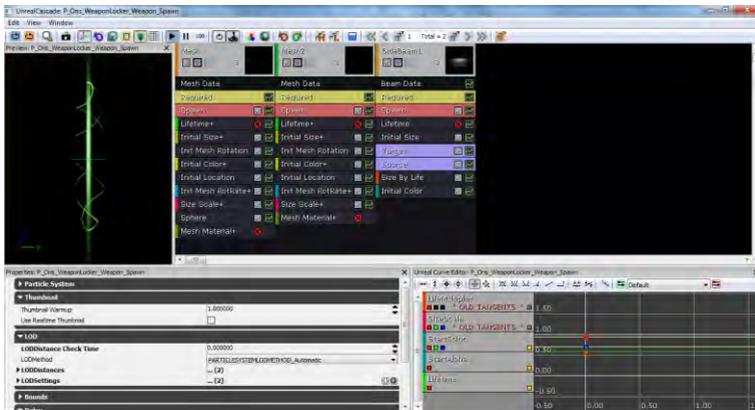
Agar pengguna dapat mengetahui bahwa suatu tempat pada *Level Map* memiliki interaksi, maka tiap tempat yang memiliki interaksi akan ditandai menggunakan particle system. Particle system ini dibuat menggunakan Unreal Cascade Particle system yang dibuat ada dua buah yaitu untuk penanda interaksi informasi ruangan dan penanda interaksi objek. Gambar 5.11 menunjukkan penanda interaksi lampu, gambar 5.10 menunjukkan penanda interaksi objek dan gambar 5.13 menunjukkan penanda untuk informasi.



Gambar 5. 12 Particle System untuk Interaksi Lampu



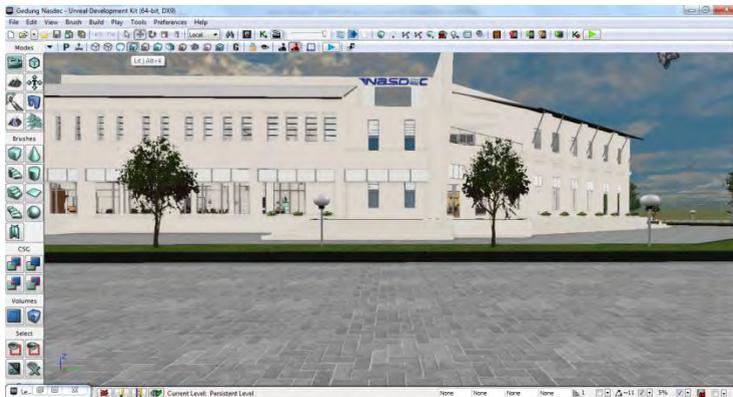
Gambar 5. 13 Particle System untuk Interaksi Objek



Gambar 5. 14 Particle System untuk informasi

5.3.1.5 Pemberian Tanaman dan Pepohonan

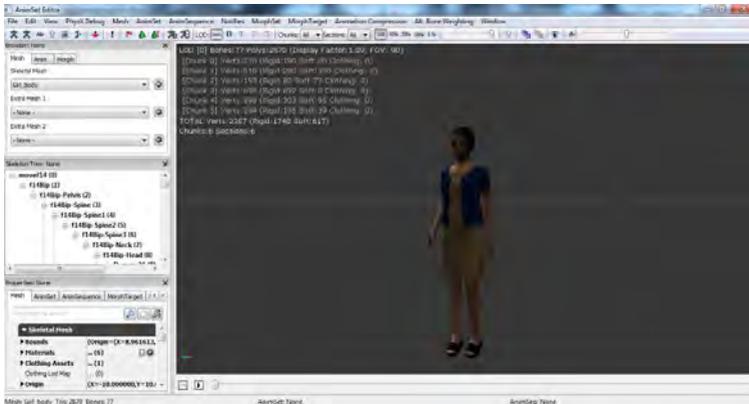
Penambahan *Static mesh* yang berbentuk tanaman juga dilakukan pada *Level Map* agar terlihat lebih nyata, dengan menggunakan *Static mesh* dan menggunakan objek tanaman yang dibuat melalui Speed Tree Modeler dan Speed Tree Compiler. Pemberian dengan hasil dari Speed Tree lebih terlihat hidup karena hasil tersebut mirip dengan pohon asli dari kondisi nyata dan terdapat efek bergerak terkena hembusan angin, berbeda dengan *Static mesh* yang tidak bergerak. Contoh tanaman dan pepohonan bisa dilihat pada gambar 5.14.



Gambar 5. 15 Tanaman Hasil Speed Tree dan Static mesh

5.3.1.6 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 adalah seorang karakter wanita seperti yang dapat dilihat pada gambar 5.15.



Gambar 5. 16 Aktor Wanita pada SkeletalMesh

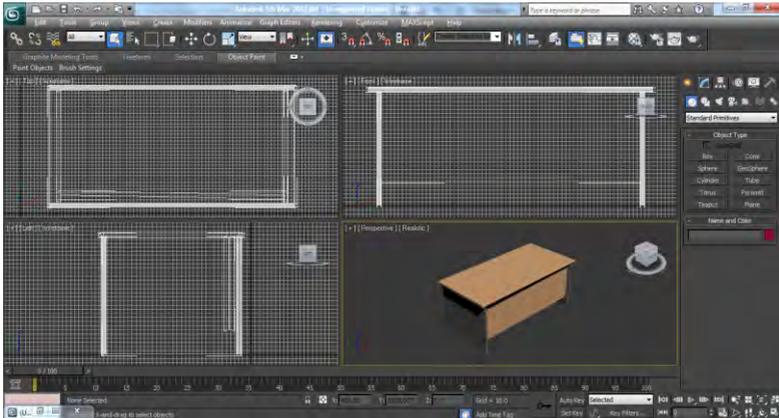
5.3.2 Pembuatan dan *Import* Objek 3D

Tahap pembuatan dan *import* objek pada *Level Map* dimulai dari pembuatan objek 3D, selanjutnya peng-*import*-an objek 3D ke dalam UDK.

5.3.2.1 Pembuatan Objek 3D

Pembuatan Objek 3D dalam pembuatan map ini semua menggunakan aplikasi Autodesk 3ds Max. Dalam Autodesk 3ds Max terdapat 4 tampilan perspektif, yang berbeda dari 4 tampilan perspektif itu ialah fungsinya untuk mempermudah kita dalam pengaturan terhadap suatu objek, yaitu dari atas, samping kanan, depan, dan perspective. Sebenarnya selain dari 4 tampilan perspektif itu ada tampilan lainnya, tetapi 4 tampilan perspektif ini adalah tampilan default dari awal membuka Autodesk 3Ds Max. Selain itu suatu objek tiga dimensi memiliki

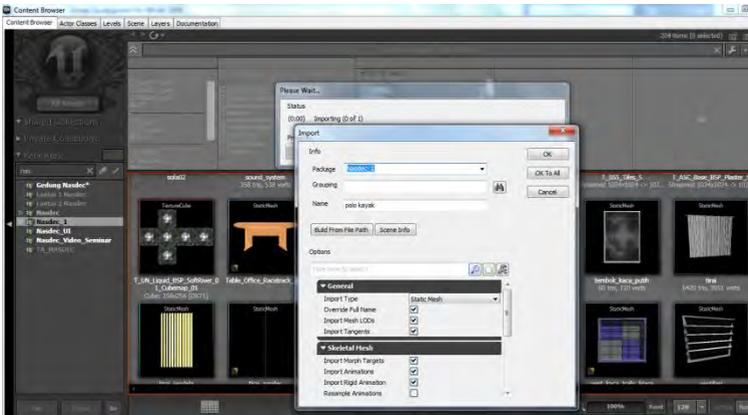
posisi x , y dan z . Tampilan perspektif dan hasil dari Autodesk 3Ds Max dapat dilihat pada gambar 5.16.



Gambar 5. 17 Tampilan Default 4 Perspektif Autodesk 3Ds Max

5.3.2.2 Import Objek 3D

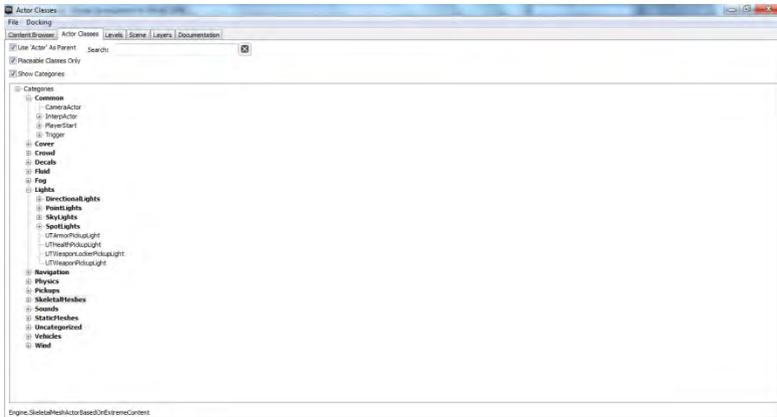
Setelah objek 3D dibuat menggunakan Autodesk 3Ds Max, maka objek disimpan dalam format *.fbx, hal ini dilakukan agar objek 3D bisa di-*import* ke dalam UDK. Objek yang kita buat dapat diatur material-nya, collision-nya, maupun LOD-nya pada *content browser*. Harus dipastikan saat meng-*import* objek, tipe yang dipilih *Staticmesh*. Didalam *content browser*, agar objek yang sudah di-*import* tertata rapi, maka dibuatkan *package*, dan atau group untuk objek tersebut. Gambar 5.17 menunjukkan *setting import* ke dalam Unreal Editor.



Gambar 5. 18 Setting Import Objek 3D UDK

5.3.3 Pengaturan Pencahayaan

Fungsi pengaturan pencahayaan ini untuk membuat objek 3D terlihat nyata, sehingga efek bayangan dan warna pada objek 3D akan terlihat lebih jelas. Pengaturan cahaya di UDK dilakukan melalui beberapa kelas aktor light (cahaya). Seperti pada gambar 5.18 terdapat beberapa jenis kelas aktor light yang ada, yaitu *Directional Light*, *Point Light*, *Sky Light*, dan *Spot Light*.



Gambar 5. 19 Macam-macam Aktor Light pada Unreal Editor

Masing-masing aktor light dapat dikonfigurasi sesuai dengan kebutuhan, seperti *brightness*, light color, dan radius. Berdasarkan pada keadaan yang sebenarnya, dalam aplikasi ini hanya menggunakan Dominant *Directional Light* dan Point Light. Dominant *Directional Light* merupakan aktor light yang memiliki cahaya yang lurus untuk menerangi bagian peta tertentu dan memiliki efek pencahayaan yang mirip dengan matahari sehingga Dominant *Directional Light* dipilih untuk merepresentasikan efek cahaya matahari, contoh penggunaannya pada peta gedung NaSDEC ITS dapat dilihat pada gambar 5.19.



Gambar 5. 20 Penggunaan DominantDirectionalLight pada peta gedung NaSDEC ITS

Konfigurasi dari aktor Dominant *Directional Light* agar terlihat lebih nyata dengan penyesuaian warna dan tingkat *brightness* dapat dilihat pada gambar 5.20.



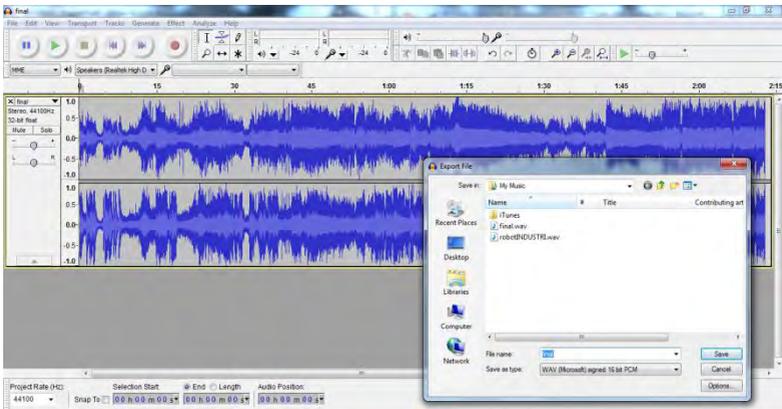
Gambar 5. 21 Konfigurasi Dominant Directional Light

5.3.4 Penambahan Suara

Tahap penambahan suara pada *Level Map* dimulai dari meng-*convert* suara menggunakan Audacity, selanjutnya meng-*import* suara ke dalam UDK.

5.3.4.1 Convert Suara

Audacity berfungsi sebagai aplikasi *convert* suara, dikarenakan awal dari suara yang didapat itu formatnya *.mp3 dan akan diubah menjadi *.wav, agar bisa dimasukkan ke dalam UDK. Pada gambar 5.21 menunjukkan aplikasi Audacity pada saat melakukan *convert* suara.



Gambar 5. 22 Melakukan Convert Suara Menggunakan Audacity

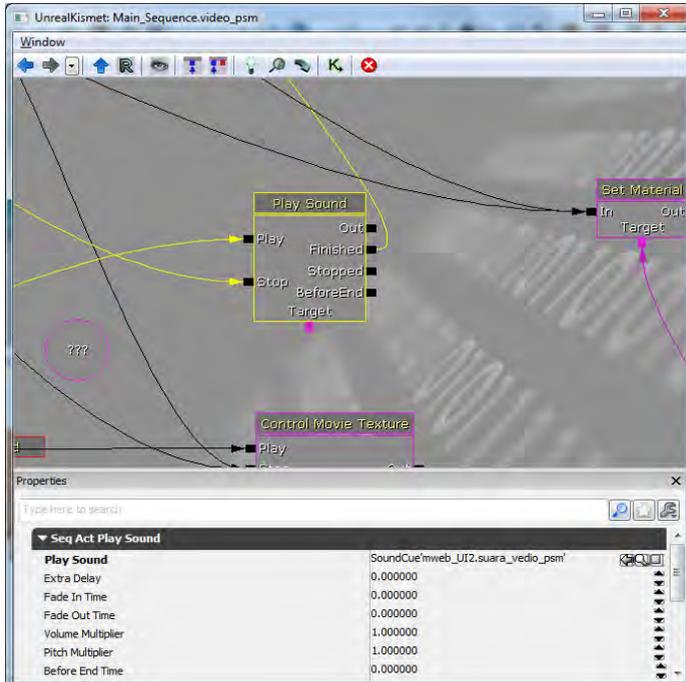
5.3.4.2 Import Suara

Selanjutnya hasil dari aplikasi Audacity yang berupa file *.wav itu kemudian *di-import* ke dalam *content browser* UDK. File hasil *import* tersebut berubah menjadi Sound Node Wave. Contoh file Sound Node Wave yang sudah berhasil *di-import*. Supaya dapat digunakan untuk dimasukkan ke dalam Unreal Kismet atau Unreal Matinee, maka dibutuhkan Sound Cue. Sound Cue merupakan gabungan dari Sound Node Wave. Contoh file Sound Node Wave dan Sound Cue dapat dilihat pada gambar 5.22.



Gambar 5. 23 File Sound Node Wave dan Sound Cue

Sound Cue yang dibuat pada Unreal Kismet dan Unreal Matinee digunakan untuk memberikan efek suara pada saat tertentu. Contoh *Sound Cue* pada Unreal Kismet pada salah satu interaksi yaitu interaksi video dapat dilihat pada gambar 5.23.



Gambar 5. 24 Sound Cue pada Unreal Kismet

5.3.5 Pembuatan Interaksi

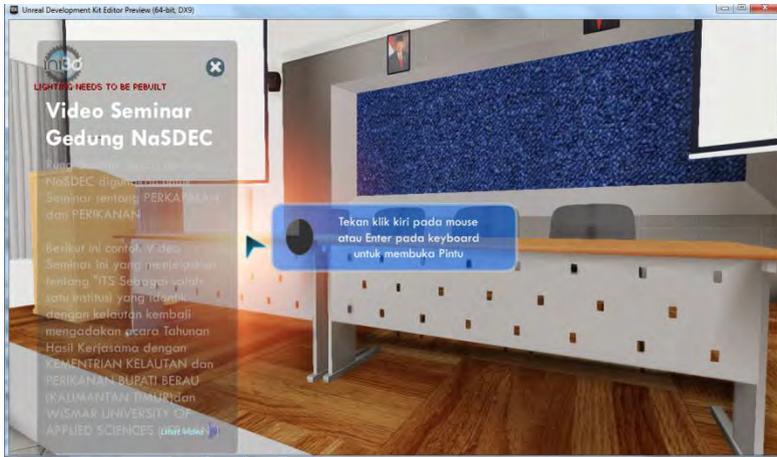
Pembuatan interaksi dalam UDK diatur dalam Unreal Kismet yang didalamnya terdapat *matinee* untuk membuat gerakan-gerakan dari objek 3D dalam map, selain itu interaksi dapat juga berupa tampilan animasi *flash*. Tampilan *Matinee* atau gabungan antara keduanya dimana interaksi tersebut dilengkapi dengan *flash* untuk membuat tampilan informasi dari ruangan atau objek 3D yang ada di dalam map. Namun, interaksi banyak

didominasi oleh tampilan animasi *flash*. Interaksi dibedakan menjadi tiga kategori 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*.

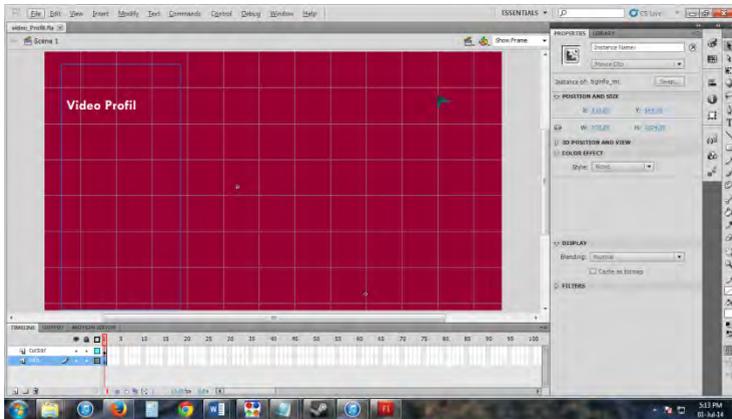
5.3.5.1 Layar Informasi

Layar informasi yang dimaksud disini adalah sebuah interaksi menggunakan animasi *flash*, dimana yang memuat informasi suatu tempat dalam map. Layar informasi muncul setiap aktor melewati tempat-tempat penting yang perlu diketahui oleh pengguna, contohnya aula, laboratorium, atau kantor. Tempat-tempat penting ini juga dapat dilihat oleh pengguna melalui Menu Peta Dua Dimensi. Layar informasi akan menghilang dengan sendiri setiap aktor memasuki tempat tersebut atau menjauh dari tempat tersebut.



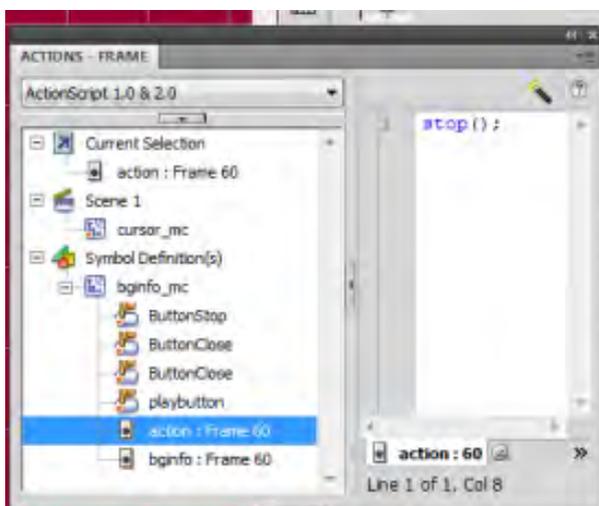
Gambar 5. 25 Animasi Flash Layar Informasi

Informasi mengenai tempat tersebut akan muncul di samping kiri layar Peta 3D yang sedang aktif. Animasi yang dibawa adalah transisi transparansi atau beningnya gambar grafik. Gambar 5.24 menunjukkan bagaimana layar informasi diletakkan berdasarkan layar yang sedang aktif, sedangkan animasi transisi ditunjukkan pada gambar 5.25.

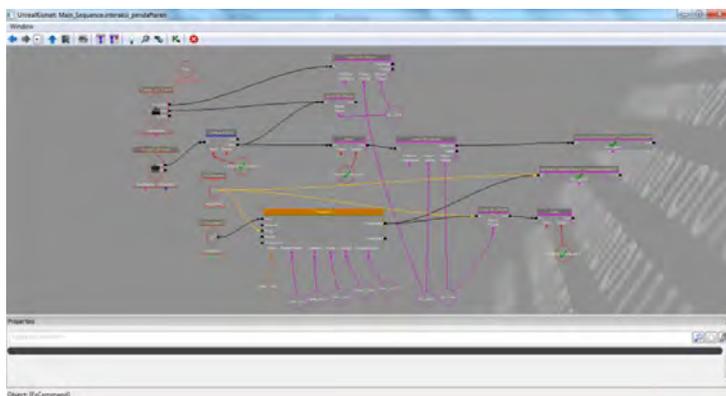


Gambar 5. 26 Alur Animasi Flash Layar Informasi

Terdapat *Action Script* yang dijalankan pada layer *action frame*. *Action Script* tersebut menghentikan jalannya animasi *flash* ketika animasi transisi transparansi telah dilakukan. Potongan kode *Action Script* tersebut dapat dilihat pada gambar 5.26. Sedangkan Kismet-nya pada Unreal Editor dapat dilihat pada gambar 5.27.



Gambar 5. 27 Action Script pada Animasi Flash Layar Informasi



Gambar 5. 28 Kismet Interaksi Prosedur Penyerahan form pelamaran kerja

5.3.5.2 Informasi Objek

Informasi objek merupakan interaksi yang menggunakan animasi *flash* dan matinee yang menampilkan informasi dari sebuah objek 3D, contohnya informasi membuka pintu, memadamkan keran atau simulasi kegiatan pada ruangan.

Untuk membuat Kismet yang dapat mengatur keran air bisa menyala atau padam, konsepnya adalah sebagai berikut. Kondisi awal adalah lampu mati, 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 menyala. Kemudian ketika aktor berinteraksi dengan saklar lampu, maka keranyang merupakan *Static mesh* dengan tipe objek InterpActor di *Level Map* akan diganti materialnya menjadi material menyala.

Dua animasi *flash* yang dibutuhkan untuk interaksi informasi objek menyalakan keran dan memadamkan keran sebagian besar adalah sama, yang berbeda hanya pada tulisan informasi yang ditampilkan pada pengguna. Teks yang ditampilkan untuk interaksi informasi objek menyalakan lampu adalah “Tekan klik kiri pada mouse atau Enter pada keyboard untuk menyalakan atau mematikan keran.” seperti pada gambar 5.28.



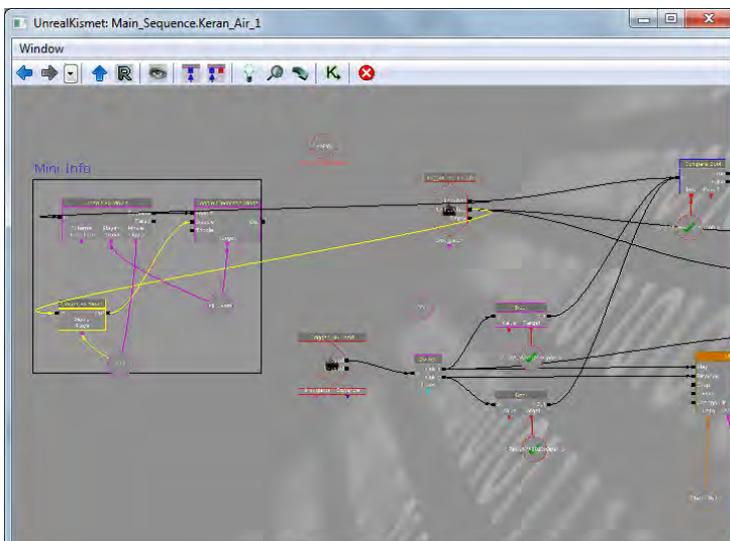
Gambar 5. 29 Animasi Flash menyalakan atau mematikan keran air

Sedangkan untuk memadamkan lampu yaitu “Tekan klik kiri pada mouse atau Enter pada keyboard untuk memadamkan lampu.” seperti gambar 2.29.



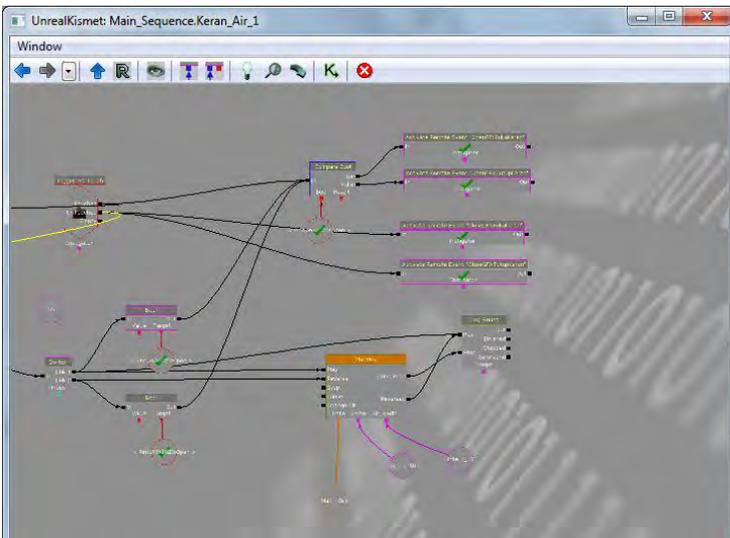
Gambar 5. 30 Animasi Flash menyalakan keran air

Terdapat bagian yang sama pada setiap Kismet untuk interaksi menyalakan dan memadamkan lampu yaitu menampilkan dan menutup dua animasi *flash*, animasi *flash* berinteraksi menyalakan dan animasi *flash* berinteraksi memadamkan keran air. Hal ini dapat terjadi dan didukung oleh Unreal Kismet dengan objek Kismet Remote Event. Cara kerja Remote Event adalah menghubungkan dua objek Kismet yang berada saling berjauhan. Remote Event dipakai untuk menampilkan atau menutup animasi *flash* yang diinginkan ketika dibutuhkan di banyak tempat. gambar 5.28 adalah RemoteEvent pada Kismet yang dapat dipanggil sesuai kebutuhan untuk menampilkan atau menutup animasi *flash* tertentu.



Gambar 5. 31 RemoteEvent pada keran air

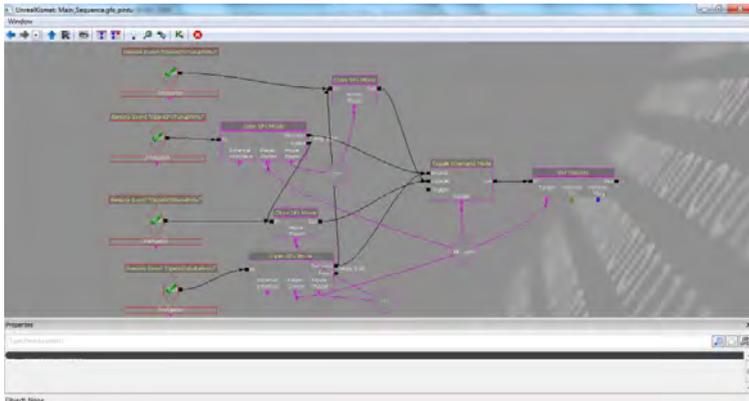
Setelah terdapat Remote Event yang menyediakan Kismet untuk menampilkan atau menutup animasi *flash* tertentu, maka pada setiap Trigger saklar yang ada, Kismet dapat dibuat dan Remote Event dapat dipanggil sesuai kebutuhan dengan menggunakan Activate Remote Event. Gambar 5.30 memperlihatkan penggunaan Activate Remote Event.



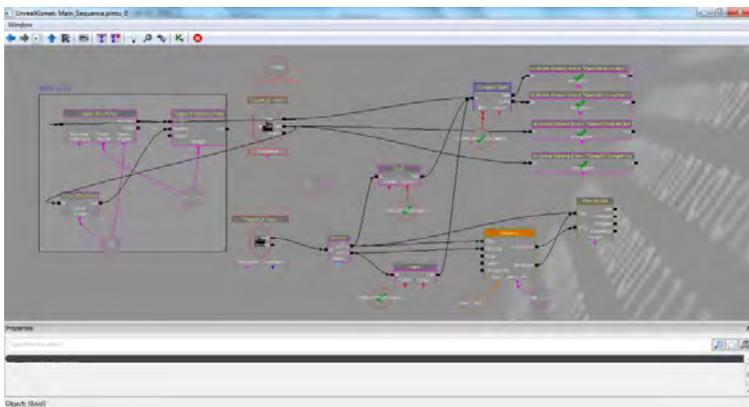
Gambar 5. 32 ActivateRemoteEvent pada Interaksi keran air

Konsep interaksi pintu sama dengan interaksi lampu yaitu dengan penggunaan *flash* sebagai penanda area jangkauan interaksi, Remote Event dan Activate Remote Event. Gambar 5.31 memperlihatkan penggunaan Remote Event pada interaksi pintu. Sedangkan untuk penggunaan

Activate Remote Event pada interaksi pintu dapat dilihat pada gambar 5.32.



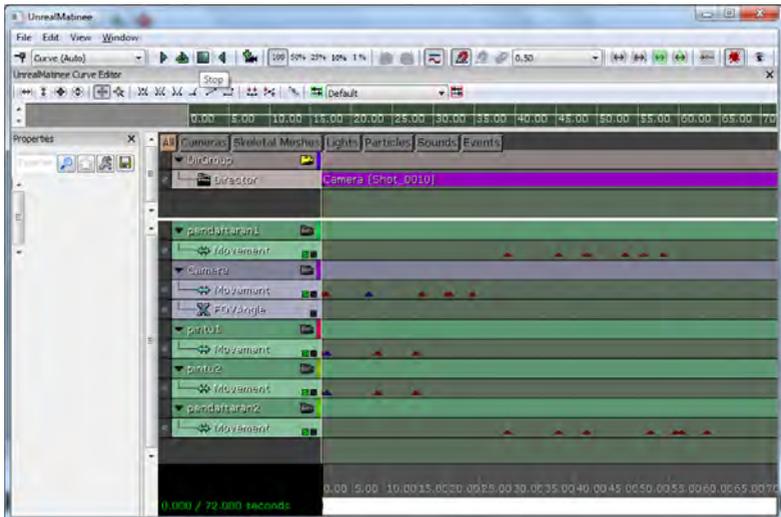
Gambar 5. 33 Remote Event pada Interaksi Pintu



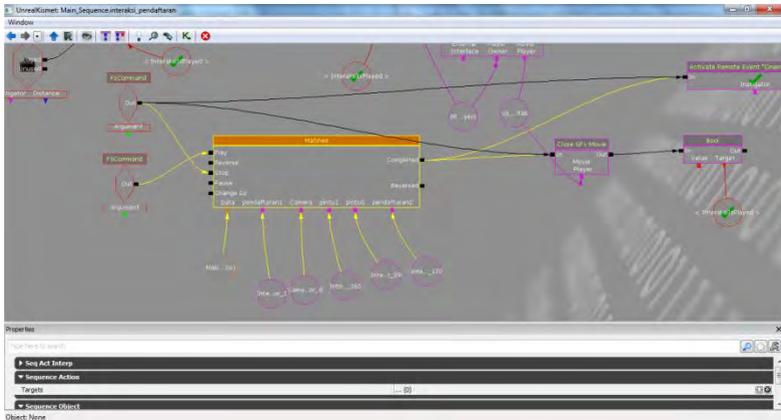
Gambar 5. 34 Active Remote Event pada Interaksi Pintu

Pada interaksi simulasi pada laboratorium 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 alat pada laboratorium dan terdapat tombol untuk menjalankan simulasi serta tombol untuk menghentikan simulasi yang sedang berjalan. Contoh tampilan matinee untuk interaksi penyerahan form dan berkas-berkas pelamaran kerja dapat dilihat pada gambar 5.34.



Gambar 5. 35 Matinee Interaksi Simulasi Penyewaan Ruang Seminar di gedung NasDEC ITS



Gambar 5. 36 Kismet Interaksi Simulasi Penyewaan Ruang Seminar di gedung NaSDEC ITS

5.3.5.3 Peta Dua Dimensi

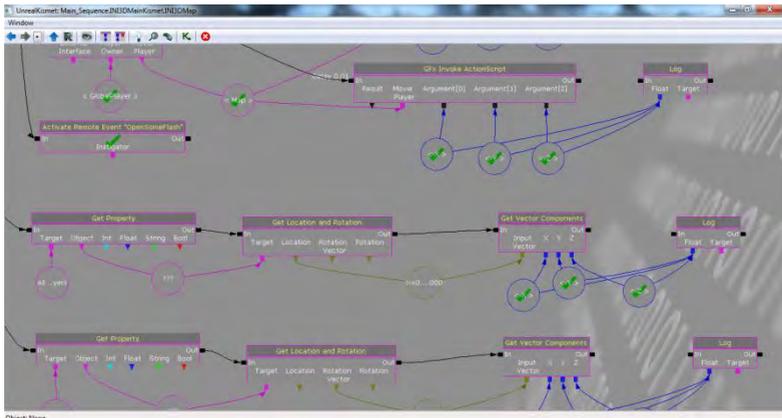
Menu peta dua dimensi menampilkan peta 2D yang sesuai dengan peta 3D, berguna untuk memberikan informasi lokasi-lokasi penting dari peta dan juga posisi dari aktor. Tampilan menu peta dua dimensi berupa animasi *flash* yang mempunyai tiga fungsi yaitu mengetahui posisi aktor, teleportasi ke suatu tempat dan menunjukkan arah menuju suatu tempat.

Lokasi-lokasi penting diberi simbol khusus agar pengguna langsung dapat mengakses tempat tersebut. Akses yang diberikan adalah aktor dapat langsung memasuki atau menempati lokasi tersebut dengan fungsi teleportasi, aktor dapat menuju tempat dengan bantuan penunjuk arah, selain itu aktor juga dapat memilih peta 2D bangunan lantai. 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.

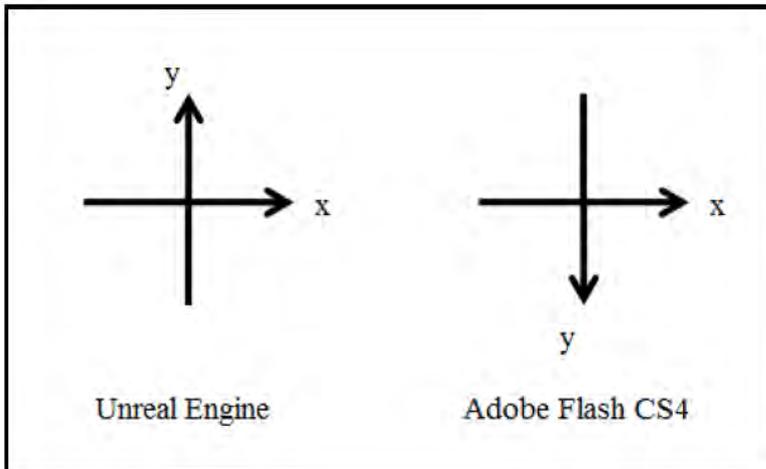
Aktor yang digunakan pada aplikasi pada dasarnya juga adalah sebuah objek yang memiliki properti sehingga dapat diketahui nilainya. Untuk mengetahui posisi aktor dengan cara location dan rotation.

Properti location yang dimiliki aktor memiliki tiga nilai yaitu X, Y, Z yang dapat diambil berupa vektor. Kismet untuk mengambil nilai properti tersebut dapat dilihat pada Gambar gambar 5.36. 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. 37. Pengambilan Nilai Properti Location Aktor

Pada gambar 5.37 menunjukkan perbedaan yang cukup signifikan mengenai x-axis dan y-axis dimana y-axis pada aplikasi animasi *flash* terbalik dibandingkan dengan y-axis peta 3D Unreal Engine. Hal ini menunjukkan bahwa sistem koordinat peta 3D Unreal Engine dan animasi *flash* Adobe *Flash* CS5 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. 38. Perbandingan x-axis dan y-axis peta 3D Unreal Engine dan *Flash* Adobe *Flash* CS5 (Tim INI3D, 2012)

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.

$$\text{Perbandingan Skala} = \frac{A-B}{C-D} \quad (1) \quad (\text{Tim INI3D, 2012})$$

Keterangan rumus (1):

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

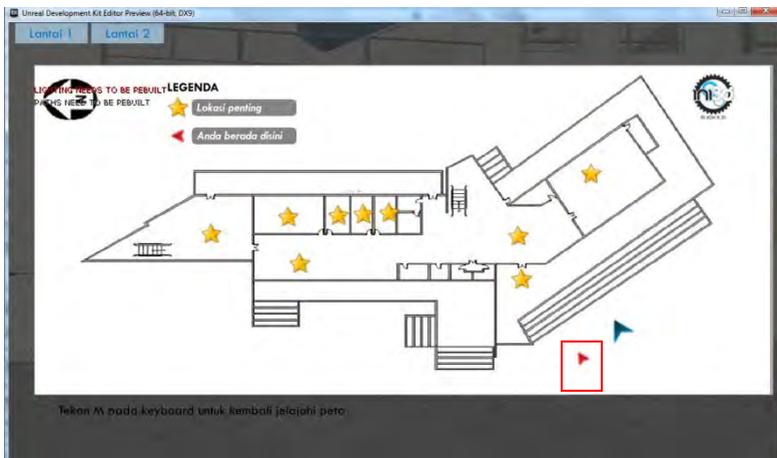
B = koordinat x-axis titik tempat kedua aplikasi animasi *flash*

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. Hasil dari rumus (1) pada peta 2D yang dapat dilihat pada gambar 5.38, jadi pada peta 3D karakter aktor didekatkan dengan tangga dan panah merah pada dua dimensi akan mengikuti juga mendekati tangga.



Gambar 5. 39 Hasil dari rumus (1) pada peta 2D

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 gambar 5.39.



Gambar 5. 40 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 dapat dilihat pada Tabel 5.3.

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

Kedua properti tersebut dapat diambil nilainya dalam Unreal Kismet 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 diambil melalui Get Location and Rotation. Kedua vektor akhirnya menghasilkan enam nilai yang menunjukkan posisi dan rotasi aktor pada peta 3D.

Peta 2D dibuat menggunakan tampilan animasi *flash* yang menggunakan aplikasi Adobe *Flash* CS5. Untuk itu, rotasi yang dikenal oleh aplikasi tersebut, sama dengan yang biasa dipakai sehari-hari mengenai rotasi yaitu rotasi

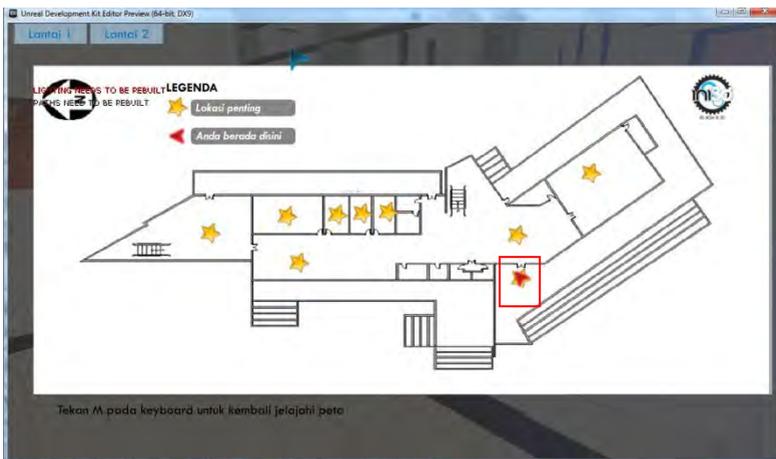
berdasarkan derajat. Padahal, sistem rotasi yang ada pada peta 3D Unreal Engine tidak menggunakan sistem rotasi derajat, berbeda dengan sistem rotasi pada aplikasi animasi *flash*. Unreal Engine memiliki ukuran sendiri mengenai rotasi. Selain itu, perlu mengetahui sistem koordinat pada aplikasi tersebut.

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 Yaw ke derajat, yaitu rumus (2), terlihat bahwa nilai Yaw pada Unreal Engine (*yaw*) 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*).

$$deg = \arccos(yaw) * \frac{A-B}{C-D} \quad (2) \quad (\text{Tim INI3D, 2012})$$

Pada tugas akhir ini, peta 2D dibuat dengan arah utara yang menuju pada y-axis negatif sistem koordinat aplikasi

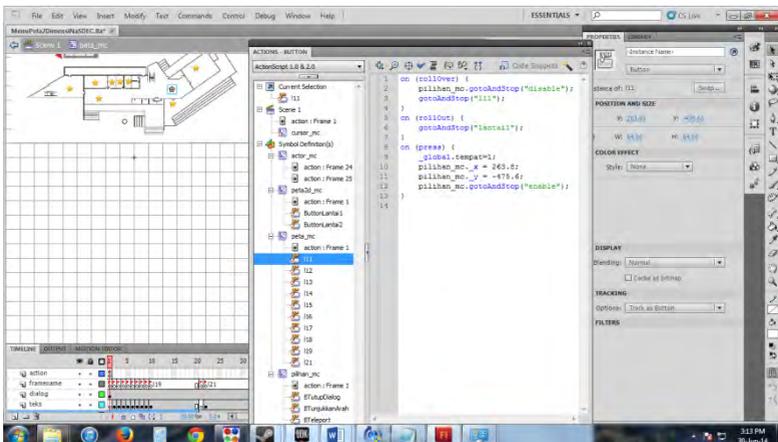
animasi *flash* Adobe *Flash* CS5 atau sama dengan sistem koordinat kartesian menuju pada y-axis negatif. Sedangkan, arah utara peta 3D adalah menuju pada y-axis positif sistem koordinat kartesian. Pada gambar 5.40 memperlihatkan hasil rumus (2) yaitu pada peta 3D yang diarahkan ke y-axis positif dapat dilihat dengan arah utara pada peta 2D yang telah menuju pada y-axis negatif.



Gambar 5. 41 Animasi *Flash* Menu Peta Dua Dimensi

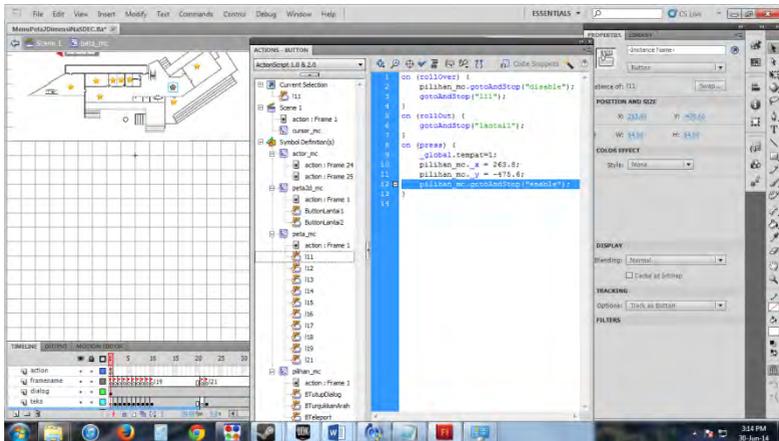
Pada gambar 5.41 terlihat bahwa terdapat *Action Script* yang dijalankan pada layer *action* frame 1 dan layer *cursor* frame 1 Symbol *cursor_mc*. *Action Script* pertama dapat dilihat pada gambar 5.47 sedangkan *Action Script* kedua dapat dilihat gambar 5.42. Fungsi *setPosition* mencakup konversi koordinat peta 3D Unreal Engine kedalam koordinat peta 2D aplikasi animasi *flash* dan meletakkan Symbol *actor_mc* yang merupakan representasi aktor pada peta 3D Unreal Engine sesuai

dengan posisi yang diinginkan pada peta 2D aplikasi animasi *flash*. Fungsi tersebut juga mengatur peta 2D lantai bangunan mana yang ditampilkan sesuai dengan posisi aktor pada peta 3D Unreal Engine. Fungsi `setRotation` menetapkan bagaimana `actor_mc` sebagai representasi aktor pada peta 3D Unreal Engine melakukan rotasi persis seperti rotasi aktor pada peta 3D Unreal Engine.



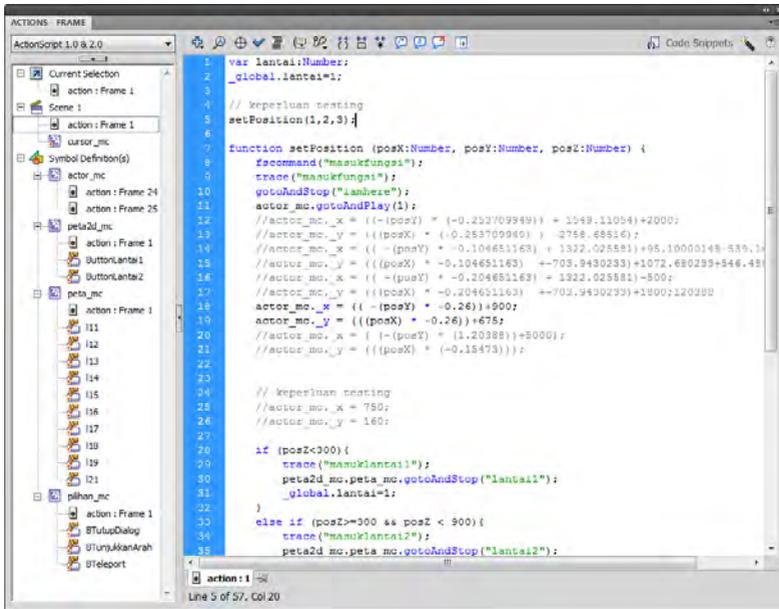
Gambar 5. 42 Struktur Utama Proyek Animasi *Flash* Menu Peta Dua Dimensi

Sebuah *Symbol Movie Clip* `cursor_mc` digunakan pada animasi *flash* Menu Peta Dua Dimensi untuk merepresentasikan kursor mouse. Pada gambar 5.42 menunjukkan *Action Script* agar *Symbol* `cursor_mc` dapat beranimasi sesuai dengan pointer mouse yang sedang aktif.



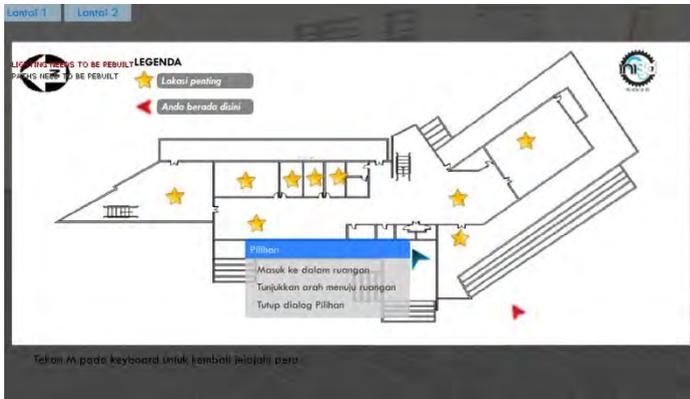
Gambar 5. 43 Potongan Kode Action Script Animasi Flash Menu Peta Dua Dimensi

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. Pengguna memilih tempat yang akan dituju dengan melakukan klik pada *Symbol Button* penanda tempat berbentuk bintang. Selanjutnya pilihan menu untuk tempat tersebut akan muncul, pengguna dapat memilih menu Masuk ke dalam ruangan untuk melakukan teleportasi. *Symbol Movie Clip* bernama *pilihan_mc* merupakan dialog pilihan menu yang muncul ketika penanda ruangan berbentuk bintang ditekan. Potongan kode Action Script animasi flash Menu Peta Dua Dimensi Scene 1 layer action frame 1 seperti gambar 5.43.



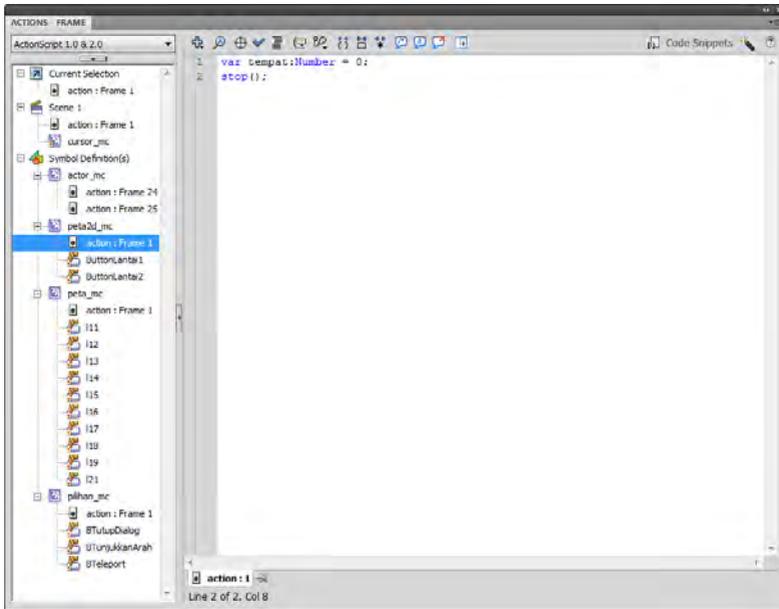
Gambar 5. 44 Potongan Kode Action Script Animasi Flash Menu Peta Dua Dimensi Scene 1 Layer Action Frame 1

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 5.44 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 *Action Script*.



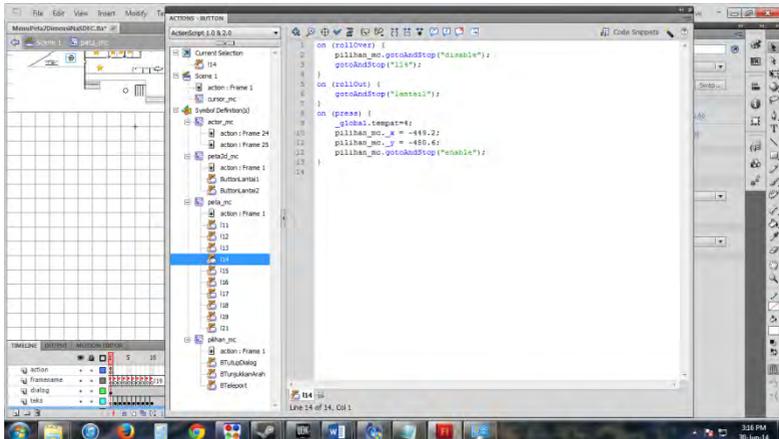
Gambar 5. 45 Pilihan Fungsi Teleportasi pada Menu Peta Dua Dimensi

Konfigurasi pertama berada pada *Symbol Movie Clip* peta2d_mc layer *action* frame 1. Gambar 5.45 memperlihatkan *Action Script* dengan melakukan inialisasi variabel tempat yang nantinya akan berubah ketika suatu *Symbol Button* penanda lokasi diklik dan variabel tempat akan disesuaikan.



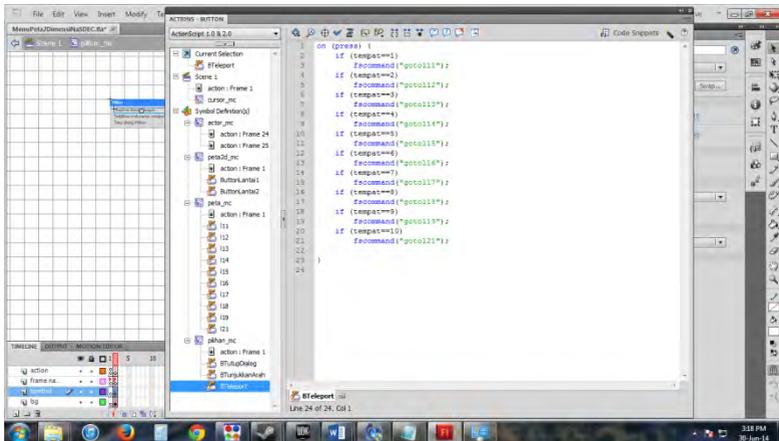
Gambar 5. 46 Kode Action Script Animasi Flash Menu Peta Dua Dimensi Symbol peta2d_mc Layer Frame 1

Konfigurasi kedua berada pada pengaturan setiap *Symbol Button* penanda lokasi yang ada pada animasi *flash*. *Symbol Button* tersebut memiliki *ActionScript* yang berisi inialisasi variabel tempat dengan mengatur nilai awalnya. Pada gambar 5.46 memperlihatkan *ActionScript* melakukan perubahan variabel global gedung SCC ITS 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. 47 Kode Action Script Animasi Flash Menu Peta Dua Dimensi pada Symbol Button Penanda Lokasi

Konfigurasi ketiga untuk fungsi teleportasi ada pada *Symbol Button* Bteleport, yaitu menu “Masuk ke dalam ruangan”. gambar 5.47 menunjukkan *Action Script* pada button tersebut dimana sebuah *fsccommand* (fungsi yang dapat memanggil Actor *Fsccommand* pada *UnrealKismet*) akan aktif berdasarkan variabel tempat.



Gambar 5. 48 Kode Action Script Animasi Flash Menu Peta Dua Dimensi pada Symbol Button BTeleport

Untuk dapat mengintegrasikan fungsi fscCommand teleportasi yang ada pada ActionScript animasi flash Menu Peta Dua Dimensi, konfigurasinya adalah membuat Kismet untuk teleportasi yang khusus diletakkan dalam satu Sequence bernama Teleport. Kismet tersebut dibuat untuk masing-masing fungsi fscCommand sebanyak lokasi-lokasi penting yang ingin dibuatkan teleportasi agar pengguna mudah mengakses masuk lokasi tersebut. Pada peta 3D Unreal Engine perlu meletakkan PlayerStart yang lokasinya direpresentasikan pada peta 2D animasi flash.

Masing-masing FscCommand pada Kismet mengacu pada Action Script untuk nilai variabel tempat yang diatur ketika melakukan klik pada Symbol Button penanda lokasi pada animasi flash Menu Peta Dua Dimensi, sedangkan sebuah PlayerStart yang berada pada sebuah lokasi pada peta 3D Unreal Engine dijadikan Variable Objek dan

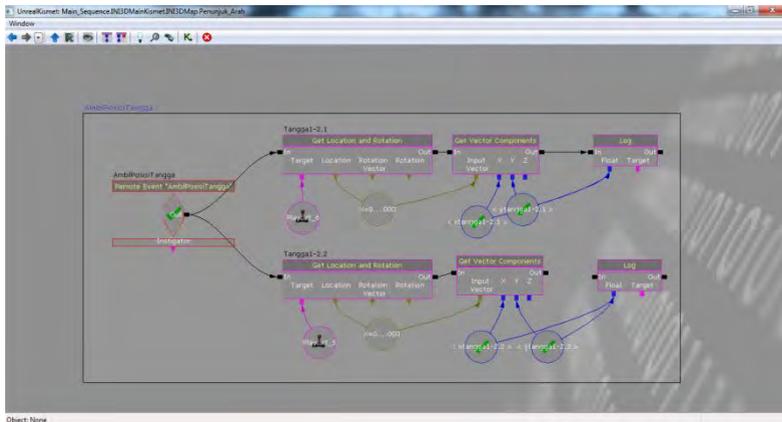
ditugaskan sebagai tujuan teleportasi seperti pada gambar 5.48. Konfigurasi Kismet teleportasi ini sama untuk semua ruangan atau lokasi, yang membedakan hanyalah Actor Fscmmand-nya karena setiap Fscmmand akan merepresentasikan pemanggilan fungsi fscmmand pada animasi *flash* dengan tujuan lokasi ruangan yang berbeda.



Gambar 5. 49 Kismet Teleportasi

Untuk fungsi penunjuk arah, logika didominasi oleh perhitungan dalam Kismet. Sebuah tampilan *flash* yang bergambar panah digunakan sebagai tampilan penunjuk arah dengan mengambil input dari peta 3D melalui *Action Script*. Jika untuk melakukan sesuatu pada peta 3D, *flash* bisa menggunakan fungsi fscmmand, 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 Action Script*. Pertama-tama, perlu dilakukan

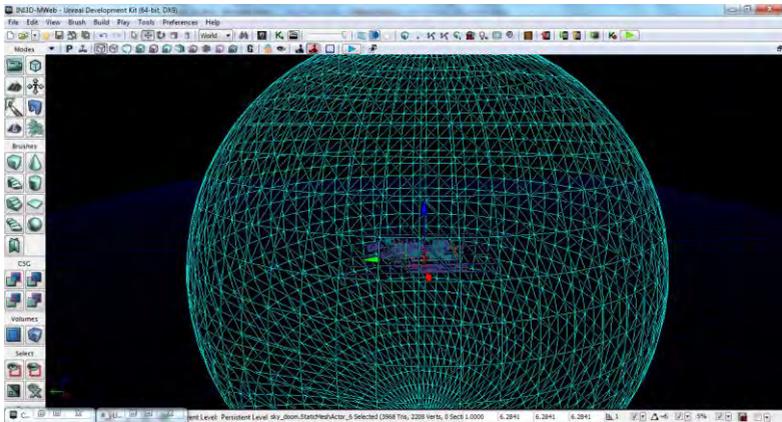
pengambilan posisi setiap tangga yang ada pada peta 3D sehingga dapat menghubungkan lokasi pada lantai bangunan yang berbeda. Kismet tersebut dapat dilihat pada gambar 5.50. Setelah posisi tangga diketahui, selanjutnya adalah mengetahui posisi aktor berada.



Gambar 5. 50 Kismet Mengambil Posisi Tangga

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 *static mesh* langit. *Staticmesh* ini berbentuk kubah setengah bola yang digunakan sebagai latar belakang langit yang mengelilingi keseluruhan peta dan dapat dilihat pada gambar 5.50.



Gambar 5. 51 *Static mesh* Langit pada Mode Wireframe

Selain menggunakan *static mesh* langit, digunakan juga 3 aktor berrfungsi pada fungsinya masing-masing yaitu pertama aktor *Dynamic Directional Light* 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 dan rumus *playrate* yang digunakan:

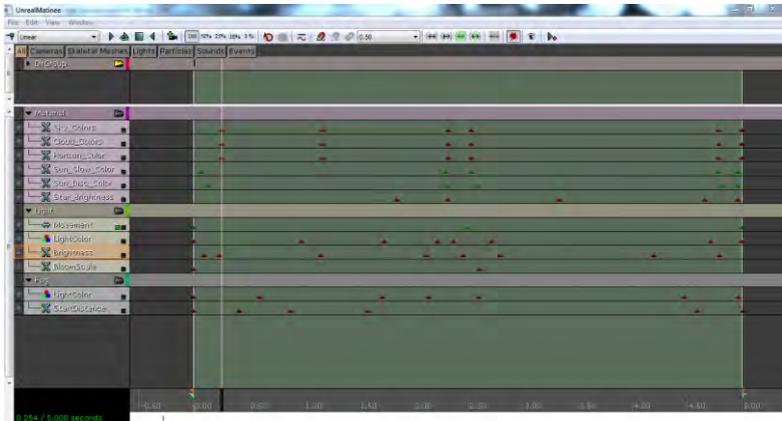
$$\text{Playrate} = \frac{1}{\frac{\text{Lama siklus yang diinginkan}}{\text{Durasi matinee}}} \quad (3) \quad (\text{Tim INI3D, 2012})$$

Keterangan rumus (3):

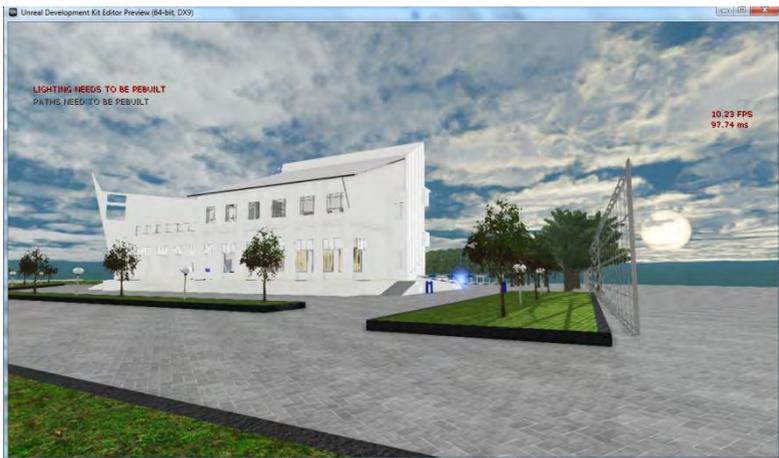
Lama siklus yang diinginkan = 24 menit

Durasi matinee = 5 detik

Hasil dari perhitungan *playrate* menampilkan matinee yang mengatur proses diatas yang dapat dilihat pada gambar 5.52 dan hasil di dalam 3D dapat dilihat pada gambar 5.53.



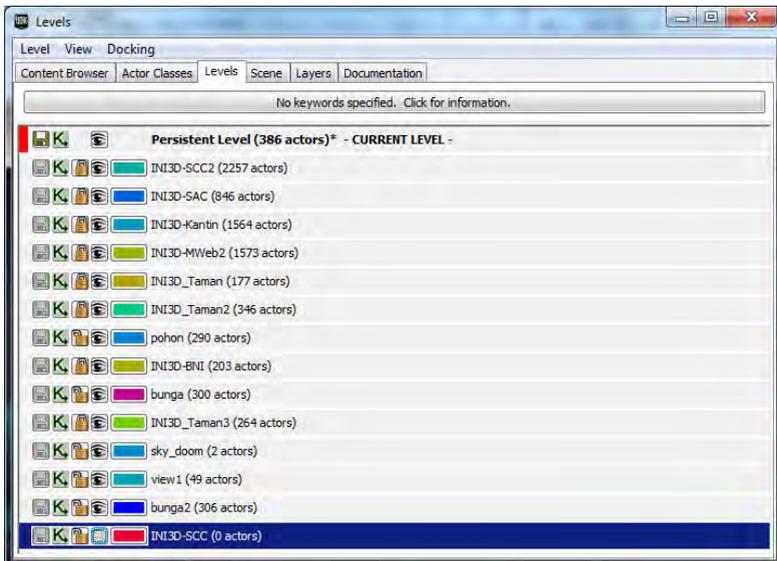
Gambar 5. 52 Matinee Simulasi Pergantian Siang dan Malam



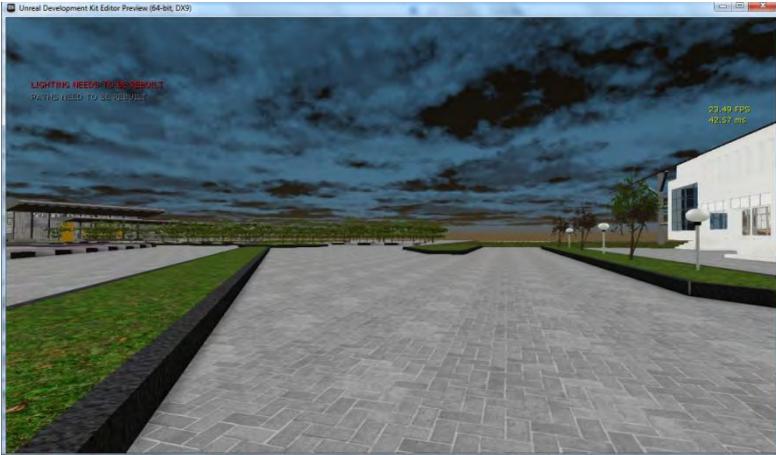
Gambar 5. 53. Simulasi Pergantian Siang dan Malam

5.4 Integrasi

Integrasi antar peta di UDK dilakukan dengan dua cara yaitu Level Streaming. Level Streaming artinya integrasi peta dilakukan secara live tanpa perantara (loading). Dalam Level Streaming, cara untuk menggabungkan antar peta adalah pilih peta yang akan digabung dengan Add Existing Level dan pilih dengan Always Visible. Dalam hal ini peta yang akan digabung yaitu peta gedung NaSDEC ITS. Lokasi peta dapat diatur sesuai kebutuhan sehingga posisi peta dapat disesuaikan letaknya. Level yang dimasukkan dapat dilihat pada gambar 5.53 dan hasil pada 3D dapat dilihat pada gambar 5.54.



Gambar 5. 54 Integrasi Peta Gedung NaSDEC, ROBOTIKA, dan TC



Gambar 5. 55 Hasil Integrasi Peta gedung NasDEC dan ROBOTIKA

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 unit test dari rancangan test case yang telah dirancang pada lampiran E. Setiap skenario pada test case dijalankan dan hasil yang ada pada test case dibandingkan dengan hasil aplikasi. Unit test case dan hasilnya tersebut dapat dilihat pada tabel 5.4.

Tabel 5. 4 Unit Test dari Rancangan Test Case pada Lampiran C

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 cara mengukur performa yang dihasilkan oleh sistem perangkat keras dalam menjalankan aplikasi. Ketentuan-ketentuan yang dipakai dalam uji coba sebagai berikut :

1. Spesifikasi komputer yang digunakan ada 4 spesifikasi yang ditampilkan pada tabel 5.5, tabel 5.5, tabel 5.7 dan tabel 5.8.

Tabel 5. 5 Spesifikasi Komputer 1

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

Tabel 5. 6 Spesifikasi Komputer 2

Spesifikasi
Prosesor : Intel® Core™ i5-4440 CPU @ 3.1Ghz (4 CPUs), ~3.1GHz
Memori : 8192MB RAM
VGA: AMD RADEON HD 7800
OS : Windows 7 Ultimate 64-bit (6.1, Build 7601)

Tabel 5. 7 Spesifikasi Komputer 3

Spesifikasi
Prosesor : Intel® Core™ 2 Duo CPU E7500 @ 2.93Ghz (2 CPUs), ~2.9GHz
Memori : 4096MB RAM
VGA : NVIDIA Geforce GTX650TI
OS : Windows 7 Ultimate 64-bit (6.1, Build 7601)

Tabel 5. 8 Spesifikasi Komputer 4

Spesifikasi
Prosesor: : Intel® Core™ i7-2670M CPU @ 2.2Ghz (4 CPUs), ~3.1GHz
Memori : 8192MB RAM
VGA : NVIDIA Geforce GT555M
OS : Windows 7 Ultimate 64-bit (6.1, Build 7601)

2. Uji coba menggunakan 4 spesifikasi komputer seperti diatas dikarenakan dianggap mewakili segmentasi komputer yang ada. Spesifikasi komputer 1 merupakan PC gaming yang memiliki spesifikasi tinggi. Spesifikasi komputer 2 dan 3 merupakan komputer yang digunakan sehari-hari yang memiliki spesifikasi sedang. Beda dari spesifikasi 2 dan 3 adalah VGA yang dimiliki.

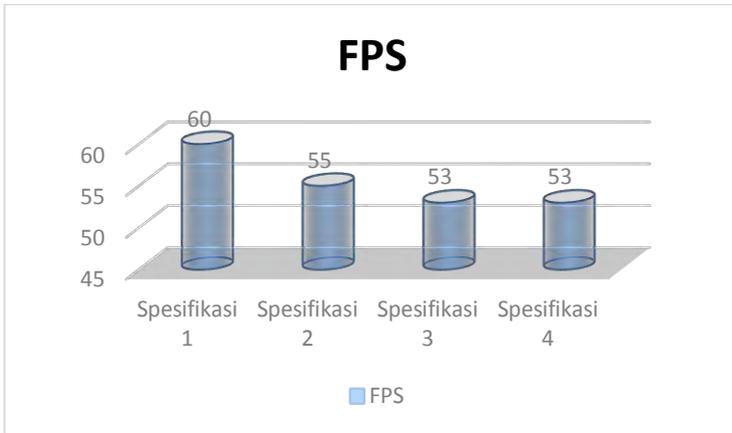
Pada spesifikasi 2 adalah VGA onboard dari Intel yang memiliki spesifikasi rendah sedangkan pada spesifikasi 3 menggunakan VGA standalone yang memiliki spesifikasi tinggi. Sedangkan spesifikasi 4 adalah PC Netbook yang memiliki spesifikasi rendah.

3. Uji coba yang dilakukan menggunakan peta tiga dimensi interaktif yang dibuat pada tugas akhir ini yang mencakup gedung NaSDEC
4. Pengambilan data FPS dilakukan dengan cara mengarahkan pandangan karakter ke depan, belakang, kiri dan kanan baik itu didalam gedung maupun diluar gedung.
5. 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 FPS *rate*.

Hasil uji coba FPS dapat dilihat pada tabel 5.9.

Tabel 5.9 Hasil Uji Coba

Spesifikasi	FPS	Keterangan
Spesifikasi 1	60	Baik
Spesifikasi 2	55	Baik
Spesifikasi 3	53	Baik
Spesifikasi 4	53	Baik



Gambar 5. 56 Grafik Perbandingan Hasil Uji Coba

Keterangan :

Standart yang digunakan dalam penentuan grafik hasil uji coba menggunakan standarisasi INI3D yaitu ; 50 fps dan juga 25 fps. Hal ini dikarenakan rata-rata spesifikasi computer untuk menjalankan sebuah game hanya memiliki minimal 30 fps.

- FPS > 50, maka spesifikasi tersebut sangat dianjurkan untuk menjalankan aplikasi
- FPS < 50 namun FPS > 25, maka spesifikasi tersebut cukup untuk menjalankan aplikasi
- FPS < 25, maka spesifikasi tersebut tidak dianjurkan untuk menjalankan aplikasi

Analisa dari hasil uji coba adalah :

- Spesifikasi yang sangat dianjurkan adalah spesifikasi 1, dikarenakan pada saat menjalankan di spesifikasi 1, aplikasi berjalan sangat lancar,
- untuk spesifikasi 2 dan 3 bisa dikatakan cukup untuk menjalankan aplikasi, tetapi lebih disarankan yang menggunakan VGA standalone dikarenakan memberikan hasil FPS yang lebih tinggi,
- sedangkan untuk spesifikasi 4 tidak dimungkinkan untuk menjalankan aplikasi 3D interaktif ini karena spesifikasi yang terlalu rendah

Berdasarkan data dari tabel uji coba performa diatas, maka rekomendasi spesifikasi komputer yang dapat digunakan sebagai acuan untuk melakukan pengembangan yang lebih lanjut pada tabel 5.10 dibawah ini :

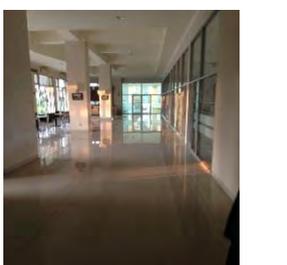
Tabel 5. 10. Spesifikasi Rekomendasi untuk Penelitian Selanjutnya

Spesifikasi
Prosesor : Intel Pentium i5 4430 ~ 3,0 Ghz
Memori : 8 GB - 16 GB RAM
VGA : NVIDIA Geforce GTX660TI
Sistem Operasi : Windows 7 Ultimate 64-bit(6.1, Build 7601)

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. Pada evaluasi ini akan digambarkan secara jelas tentang hasil implementasi ruangan yang telah dimodelkan pada peta 3D beserta gambar asli ruangan tersebut. Evaluasi tersebut dapat dilihat pada tabel 5.11.

Tabel 5. 11 Validasi Peta 3D dengan Keadaan Nyata

Ruangan	Kondisi Nyata	Peta 3D
Gedung NaSDEC		
Keterangan	Tampilan gedung NaSDEC dari depan Pada peta 3D sudah terlihat lebih mirip dari kondisi nyatanya dari bentuk.	
Ruang Seminar		
Keterangan	Tampilan ruang Seminar di gedung NaSDEC dari samping. Pada peta 3D sudah terlihat lebih mirip dari kondisi nyatanya dari bentuk, dan material.	
Café NaSDEC		
Keterangan	Tampilan café nasdec sudah terlihat mirip dengan kondisi nyatanya.	

Kantor Sewa	
Keterangan	Tampilan kantor sewa di gedung NaSDEC di lantai 2 sudah lebih mirip dengan kondisinya nyata

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. Dengan adanya standarisasi dari aplikasi INI3D yang sudah ada sebelumnya, pembuatan peta tiga dimensi dapat dilakukan dengan mudah dengan menggunakan Unreal Development Kit beserta fitur-fitur yang disediakan karena telah ada standarisasi yang bisa digunakan pedoman untuk pembuatan aplikasi.
2. Penggunaan standarisasi juga membantu dalam proses integrasi dengan peta tiga dimensi dengan menggunakan metode loading, sehingga sistem tidak terlalu berat.
3. Lebih mengutamakan *brusher* daripada *staticmesh*, karena menggunakan *brusher* hanya berat diawal proses building peta, tetapi building selanjutnya akan lebih ringan, dan sebaliknya, menggunakan *staticmesh* akan berat di setiap building peta.
4. Penggunaan VGA Card dan Memory sangat dianjurkan untuk menunjang performa UDK dari pada penggunaan VGA On Board.

6.2 Saran

Pengembangan aplikasi INI3D, memiliki batasan-batasan tertentu dalam pengerjaannya. Hal ini menyebabkan perlu adanya pengembangan lebih lanjut. Beberapa saran yang dapat dilakukan untuk pengembangan aplikasi yang dapat dilakukan adalah sebagai berikut :

1. Pengembangan aplikasi INI3D sebaiknya dilakukan pada komputer yang memang mendukung untuk proses pembuatan game.
2. Interaksinya lebih dikembangkan lagi atau lebih unik lagi dalam menampilkannya.
3. Untuk penelitian selanjutnya lebih mengutamakan *brusher* daripada *staticmesh* agar resource yang dibutuhkan oleh aplikasi menjadi lebih rendah.
4. Melakukan back up secara berkala untuk mengantisipasi jikat sewaktu waktu terjadi kesalahan atau bug dan manajemen *resource* yang baik karena UDK menghabiskan *resource storage* yang sangat besar terutama jika fitur auto save diaktifkan.

DAFTAR PUSTAKA

Airlangga, B. (2011). *Pembangunan Peta Tiga Dimensi Informatif Pada Jurusan Sistem Informasi Institut Teknologi Sepuluh Nopember Dengan Menggunakan Unreal Engine*. Institut Teknologi Sepuluh Nopember. Surabaya: ITS.

Arita. 2011. *Game Engine*. Retrieved December 2, 2012. From <http://bordearita.wordpress.com/2011/06/08/game-engine>.

Aryana, D. (2012). *Pembangunan Peta Tiga Dimensi Informatif Pada Jurusan Statistika Institut Teknologi Sepuluh Nopember Dengan Menggunakan Unreal Engine*. Surabaya: ITS.

Assyifa, S. N. (2011). *Pengembangan Peta Interaktif Tiga Dimens Jurusan Arsitektur Institut Teknologi Sepuluh Nopember Surabaya Menggunakan Unreal Engine*. Institut Teknologi Sepuluh Nopember. Surabaya: ITS.

Bubsy J, P. Z. (2004). *Mastering Unreal Technology, Volume I Introduction to Level Design with Unreal Engine* 3. Indianapolis.

Bullen, T., Katchabaw, M. J. & Dyer-Witheyford, N. (2006). *Instrumentation of Video Game Software to Support Automated Content Analyses*. Makalah disajikan dalam 2006 GameOn North America Conference, Monterey, California, September 2006.

Damaiyanti, T. I. (2011). *Pengembangan Peta Interaktif Tiga Dimensi Jurusan Arsitektur Institut Teknologi Sepuluh Nopember Menggunakan Unreal Engine*. Institut Teknologi Sepuluh Nopember. Surabaya: ITS.

Epic Games, Inc. (2012, December). Diambil kembali dari Unreal Technology Product: www.unreal.com

Fitri, A. (2011). *Pengembangan Peta Interaktif Tiga Dimensi Gedung Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya Menggunakan Unreal Engine*. Institut Teknologi Sepuluh Nopember. Surabaya: ITS.

Fritsch D, K. M. (2004). "Visualisation using Game Engines" *ISPRS Commission, Vol 5*.

Haryananda, Z. S. (2011). *Pengembangan Peta Interaktif Tiga Dimensi Gedung BAAK Institut Teknologi Sepuluh Nopember Surabaya menggunakan Unreal Engine*. Institut Teknologi Sepuluh Nopember. Surabaya: ITS.

Jatmiko, S. S. (2011). *Pengembangan Peta Tiga Dimensi Interaktif Gedung Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya Menggunakan Unreal Engine*. Undergraduate Thesis, Institut Teknologi Sepuluh Nopember, Surabaya.

Juarez, J. (2011). *Dynamic Sky Cycle Tutorial*. Dipetik April 10, 2012, dari Julio Juarez Portfolio: <http://3dbrushwork.com/tutorials/>

Lesmana, L. E. (2012). *Pengembangan Peta Tiga Dimensi Interaktif Jurusan Fisika Institut Teknologi Sepuluh Nopember Surabaya Menggunakan Unreal Engine.* Undergraduate Thesis, Institut Teknologi Sepuluh Nopember, Surabaya.

Listyadana, Y. (2012). *Pengembangan Peta Tiga Dimensi Gedung Jurusan Kimia Institut Teknologi Sepuluh Nopember Interaktif Menggunakan Unreal Engine.* Undergraduate Thesis, Institut Teknologi Sepuluh Nopember, Surabaya.

Mufti, A. (2011). *Pengembangan Peta Interaktif Tiga Dimensi Jurusan Teknik Lingkungan Institut Teknologi Sepuluh Nopember Surabaya Menggunakan Unreal Engine.* Undergraduate Thesis, Institut Teknologi Sepuluh Nopember, Surabaya.

NaSDEC Indonesia (2012). Retrieved March 2013. From <http://www.nasdec-indonesia.com>

Pahlevi, A. B. (2011). *Pengembangan Peta Tiga Dimensi Interaktif Jurusan Teknik Fisika Institut Teknologi Sepuluh Nopember Surabaya.* Undergraduate Thesis, Institut Teknologi Sepuluh Nopember, Surabaya.

Prasetya, N. B. (2011). *Pemetaan Digital Secara Tiga Dimensi pada Gedung Teknik Kelautan Institut Teknologi Sepuluh Nopember Surabaya Menggunakan Unreal Engine.* Undergraduate Thesis, Institut Teknologi Sepuluh Nopember, Surabaya.

Purnama, F. M. (2011). *Pengembangan Peta Interaktif Tiga Dimensi Jurusan Teknik Sipil Institut Teknologi Sepuluh Nopember Menggunakan Unreal Engine.* Institut Teknologi Sepuluh Nopember. Surabaya: ITS.

Putra, D. A. (2011). *Pembuatan Peta Tiga Dimensi Wilayah Puskom dan Gedung BAUK Institut Teknologi Sepuluh Nopember Surabaya Dengan Menggunakan Unreal Engine.* Institut Teknologi Sepuluh Nopember. Surabaya: ITS.

Putra, R. A. (2011). *Pengembangan Peta Tiga Dimensi Gedung Jurusan Teknik Kimia Institut Teknologi Sepuluh Nopember Interaktif Menggunakan Unreal Engine.* Institut Teknologi Sepuluh Nopember. Surabaya: Undergraduate thesis.

Rachmansyah, E. (2012). *Pengembangan Peta Tiga Dimensi Interaktif Jurusan Grha Sepuluh Nopember ITS dan UPT Bahasa Instritut Teknologi Sepuluh Nopember Surabaya Menggunakan Unreal Engine.* Undergraduate Thesis, Institut Teknologi Sepuluh Nopember, Surabaya.

Rudyanti, K. (2012). *Pengembangan Peta Tiga Dimensi Interaktif Jurusan Matematika Instritut Teknologi Sepuluh Nopember Surabaya Menggunakan Unreal Engine.* Undergraduate Thesis, Institut Teknologi Sepuluh Nopember, Surabaya.

Safitri, A. (2011). *Penerapan Unreal Engine Pada Pemetaan Digital Tiga Dimensi Gedung Jurusan Desain Produk Industri Institut Teknologi Sepuluh Nopember*

Surabaya. Institut Teknologi Sepuluh Nopember. Surabaya: ITS.

Saputra, C. S. (2012). *Pengembangan Peta Tiga Dimensi Gedung Jurusan Biologi Institut Teknologi Sepuluh Nopember Interaktif Menggunakan Unreal Engine*. Undergraduate Thesis, Institut Teknologi Sepuluh Nopember, Surabaya.

Shiratuddin M F, F. D. (2007). *Utilizing 3D Games Development Tool For Architectural Design in a Virtual Environment*. The University of Southern Mississippi.

Shiratuddin M F, T. M. (2002). *Virtual Office Walkthrough Using a 3D Game Engine*. Department of Building Construction.

Subakti, A. R. (2012). *Penggunaan Unreal Engine Untuk Aplikasi Peta 3D Interaktif pada Jurusan Teknik Perkapalan Institut Teknologi Sepuluh Nopember*. undergraduate Thesis, Institut Teknologi Sepuluh Nopember, Surabaya.

Tim INI3D. (2011). *Pengembangan Peta Interaktif Tiga Dimensi Institut Teknologi Sepuluh Nopember Menggunakan Unreal Engine*. Surabaya.

Tim INI3D. (2012). *Pengembangan Peta Interaktif Tiga Dimensi Institut Teknologi Sepuluh Nopember Menggunakan Unreal Engine*. Surabaya.

Umami, F. (2011). *Pengembangan Peta Interaktif Tiga Dimensi Program Studi D3 Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember Menggunakan Unreal Engine*. Undergraduate Thesis, Institut Teknologi Sepuluh Nopember, Surabaya.

Unreal Development Kit. (2010, May). Dipetik January 30, 2011, dari www.udk.com

Winata, Y. A. (2011). *Pengembangan Peta Interaktif Tiga Dimensi Rektorat dan Pascasarjana Institut Teknologi Sepuluh Nopember Surabaya Menggunakan Unreal Engine*. Undergraduate Thesis, Institut Teknologi Sepuluh Nopember, Surabaya.

Wirangga, P. (2011). *Pengembangan Peta Tiga Dimensi Interaktif Jurusan Sistem Perkapalan Instritut Teknologi Sepuluh Nopember Surabaya Menggunakan Unreal Engine*. Institut Teknologi Sepuluh Nopember. Surabaya: ITS.

Yasin, M. N. (2012). *Pembangunan Peta Tiga Dimensi Informatif Pada Jurusan Sistem Informasi Institut Teknologi Sepuluh Nopember Dengan Menggunakan Unreal Engine*. Surabaya: ITS.

BIODATA PENULIS



Penulis dilahirkan di Perancis pada tanggal 21 Agustus 1990. Penulis merupakan anak kedua dari 3 bersaudara. Penulis telah menempuh pendidikan formal yaitu di TK Hidayatullah Surabaya, SD Muhammadiyah 4 Surabaya,

SMPN 19 Surabaya, dan SMA 2 Surabaya Surabaya .

Pada tahun 2009 penulis diterima di jurusan Sistem Informasi – Institut Teknologi Sepuluh Nopember (ITS) Surabaya melalui jalur SNMPTN dan terdaftar dengan NRP 5209100138

Selama menempuh perkuliahan di Sistem Informasi ITS, penulis aktif mengikuti kegiatan kemahasiswaan baik akamedik maupun non-akademik ditambah beberapa kali mendapat project di luar kampus.

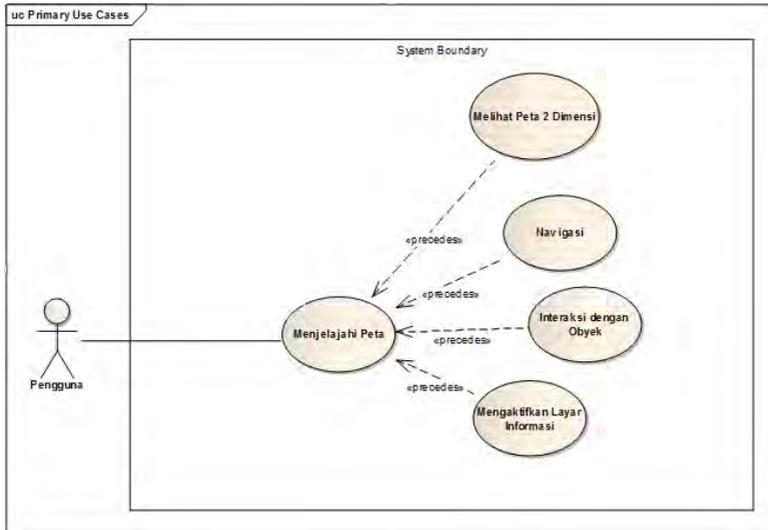
Tugas akhir yang dipilih penulis di Jurusan Sistem Informasi ini merupakan salah satu bidang minat E-Bisnis. Jika ada Pertanyaan tentang tugas akhir ini Penulis dapat dihubungi melalui e-mail adnanmaulana.mail@gmail.com.

LAMPIRAN A
DIAGRAM DAN DESKRIPSI USE CASE

A-2

Halaman ini sengaja dikosongkan.

A.1 Diagram Use Case



Gambar A. 1 Diagram Use Case

A.2 Deskripsi Use Case Interaksi dengan Obyek

Tabel A. 1 Deskripsi Use Case Interaksi dengan Obyek

<i>UC01 – Interaksi dengan Obyek</i>	
<i>Primary Actor:</i> Pengguna	<i>Level:</i> User Goal
<i>Pre-conditions:</i> Pengguna berada di halaman Peta 3D.	
<i>Triggers:</i> <ul style="list-style-type: none"> • Pengguna bergerak masuk dalam jangkauan area interaksi suatu obyek. 	
<i>Basic course:</i> Sistem menampilkan pesan interaksi yang dapat terjadi	

dengan suatu obyek. Pengguna menekan tombol mouse kiri. Sistem akan menjalankan fungsi interaksi pada obyek tersebut.
<i>Post-conditions:</i> Sistem telah menjalankan fungsi interaksi obyek tersebut dan obyek berubah kondisi sesuai dengan fungsi interaksinya.
<i>Alternate courses:</i> 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.3 Deskripsi Use Case Melihat Peta 2 Dimensi

Tabel A. 2 Deskripsi Use Case Melihat Peta 2 Dimensi

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

-
<p><i>Alternate courses:</i></p> <p>Jika pengguna menekan tombol W/A/S/D/panah atas/panah bawah/panah kiri/panah kanan pada keyboard: sistem menjalankan UC05</p> <p>Jika pengguna dalam jangkauan areainteraksi suatu obyek dan menekan klik kiri pada mouse: sistem menjalankan UC01</p>

A.4 Deskripsi Use Case Navigasi

Tabel A. 3 Deskripsi Use Case Navigasi

<i>UC05 – Navigasi</i>	
<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> <p>Jika pengguna menekan W atau panah atas pada keyboard, sistem menggerakkan aktor ke arah depan.</p> <p>Jika pengguna menekan A pada keyboard, sistem menggerakkan aktor ke arah kiri.</p> <p>Jika pengguna menekan D pada keyboard, sistem menggerakkan aktor ke arah kanan.</p> <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.5 Deskripsi Use Case Menjelajahi Peta

Tabel A. 4 Deskripsi Use Case Menjelajahi Peta

UC07 – Menjelajahi Peta	
<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 Mulai dan menekan tombol Enter pada keyboard atau klik kiri pada mouse. 	

<p>Basic course: Sistem me-load pilihan peta aktif dan menampilkan halaman Peta 3D sesuai dengan pilihan peta aktif.</p>
<p>Post-conditions: Sistem menampilkan halaman Peta 3D sesuai dengan pilihan peta aktif.</p>
<p>Alternate courses: Jika pengguna memilih menu Kembali: sistem menampilkan halaman Menu Utama.</p>

A.6 Deskripsi Use Case Mengaktifkan Layar Informasi

Tabel A. 5 Deskripsi Use Case Mengaktifkan Layar Informasi

UC09 – Mengaktifkan Layar Informasi	
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: Pengguna bergerak masuk dalam jangkauan area interaksi suatu obyek. Pengguna menekan tombol mouse kiri. Sistem menampilkan layar informasi. Pengguna melakukan informasi sesuai dengan alur interaksi.	
Post-conditions: -	
Alternate courses: -	

A-8

Halaman ini sengaja dikosongka.

LAMPIRAN B
TEST CASE

B-2

Halaman ini sengaja dikosongkan.

B.1. Test Case Interaksi Dengan Obyek

Tabel B. 1 Test Case Melihat Peta 2 Dimensi

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.

B.2. Test Case Melihat Peta 2 Dimensi

Tabel B. 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.

B.3. Test Case Navigasi

Tabel B. 3 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

B.4. Test Case Menjelajahi Peta

Tabel B. 4 Test Case Menjelajahi Peta

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