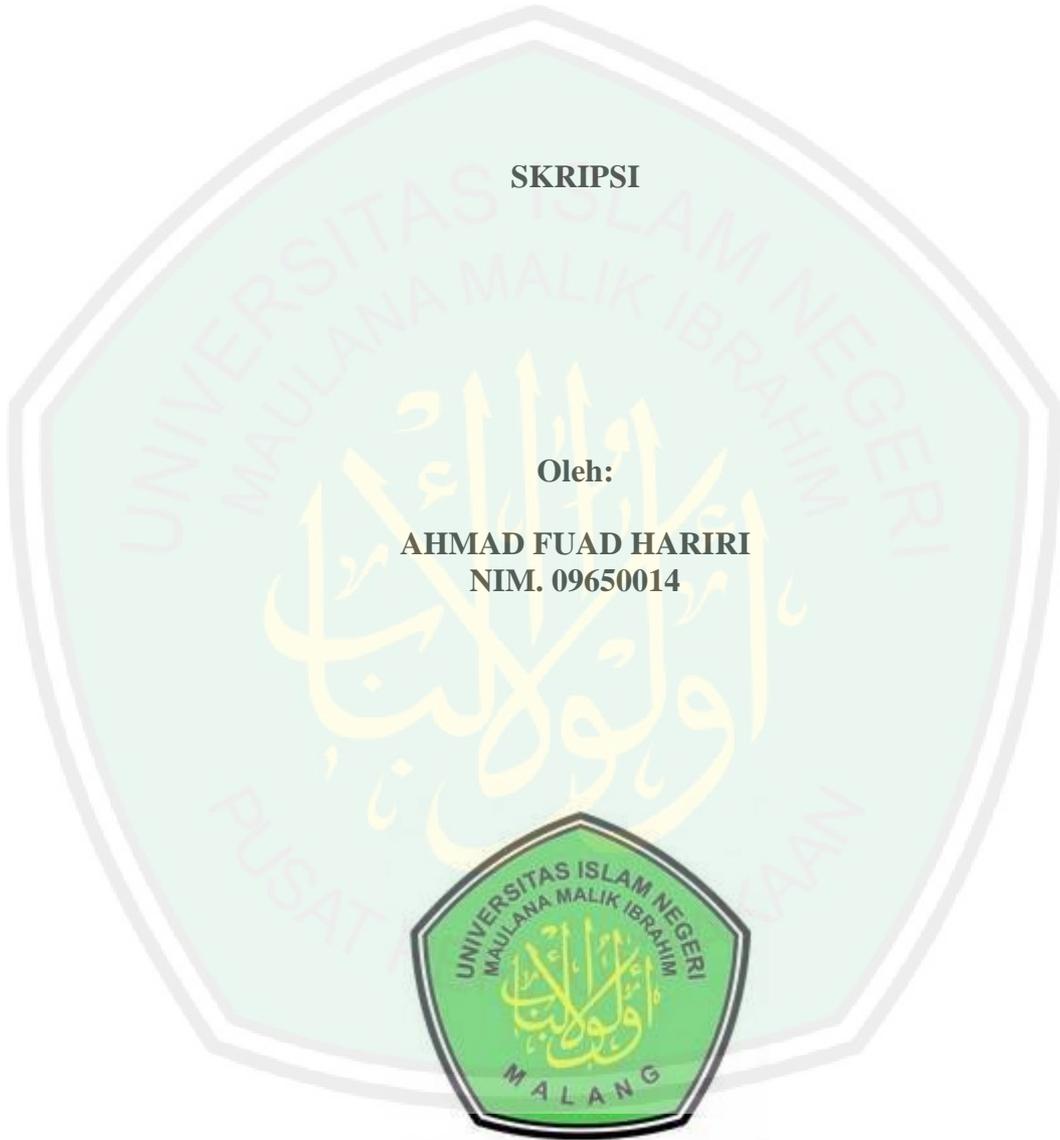


***DRIVING SIMULATION MENGGUNAKAN METODE PATHFINDING
DENGAN STUDI KASUS DI KAMPUS UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG***

SKRIPSI

Oleh:

**AHMAD FUAD HARIRI
NIM. 09650014**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM MALANG
2013**

***DRIVING SIMULATION MENGGUNAKAN METODE PATHFINDING
DENGAN STUDI KASUS DI KAMPUS UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG***

SKRIPSI

**Diajukan Kepada:
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim (UIN Maliki) Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
AHMAD FUAD HARIRI
NIM. 09650014**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM MALANG
2013**

LEMBAR PENGESAHAN
DRIVING SIMULATION MENGGUNAKAN METODE PATHFINDING
DENGAN STUDI KASUS DI KAMPUS UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG

SKRIPSI

Oleh

AHMAD FUAD HARIRI
NIM. 09650014

Diajukan Kepada:
Dekan Fakultas Sains dan Teknologi
Universitas Islam Maulana Malik Ibrahim (UIN Maliki) Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)
September 2013

Susunan Dewan Penguji:	Tanda Tangan
1. Penguji Utama : Hani Nurhayati, M.T NIP. 197107222011011001	()
2. Ketua Penguji : Yunifa Miftachul Arif, M.T NIP. 198306162011011004	()
3. Sekretaris Penguji : Fresy Nugroho, M.T NIP. 196805192003122001	()
4. Anggota Penguji : Irwan Budi Santoso, M.Kom NIP. 197710202009121001	()

Mengetahui dan Mengesahkan,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crvsdian, MCS
NIP.19740424 200901 1 008

HALAMAN PERSEMBAHAN

Pertama, Kepada Orang tua saya, Masyhadi dan Mudlikah, yang telah memberikan segala kehidupannya untuk mendidik anak - anaknya menjadi anak yang dirahmati oleh Allah.

Kedua, Kepada seluruh adik - adik saya Shofi auliah wardah dan Andini Salsa billa yang telah mendukung dan menerima saudaranya dengan segala kekurangannya.

Ketiga, Kepada Keluarga Besar saya yang telah mendukung dan memberi motifasi dalam pengerjaan skripsi ini.

Keempat, kepada Semua Guru -Guru saya yang telah bersedia untuk membimbing saya dan sudah saya anggap sebagai orang tua kedua saya.

Kelima, Kepada Jodoh saya yang telah ditetapkan di lauh mahfuz yang sudah bersedia menunggu saya sampai saat dipertemukan oleh Allah.

Keenam, kepada seluruh teman - teman saya terutama TI 2009 yang telah membantu dan saling mendukung selama proses pembelajaran di UIN Malang. Terima kasih

Motto

Permudahlah orang lain, maka engkau akan di permudah oleh Allah.



PERNYATAAN KEASLIAN TULISAN

Saya yang bertandatangan dibawah ini:

Nama : Ahmad Fuad Hariri

NIM : 09650014

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul Penelitian : ***DRIVING SIMULATION MENGGUNAKAN METODE PATHFINDING DENGAN STUDI KASUS DI KAMPUS UNIVERSITAS ISLAM MAULANA MALIK IBRAHIM MALANG***

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 25 September 2013

Yang membuat pernyataan,

Ahmad fuad hariri

NIM. 09650014

KATA PENGANTAR

Segala puji syukur dipanjatkan bagi Allah S.W.T atas limpahan rahmat, taufik, dan hidayah-Nya sehingga skripsi yang berjudul ***DRIVING SIMULATION MENGGUNAKAN METODE PATHFINDING DENGAN STUDI KASUS DI KAMPUS UIVERSITAS ISLAM MAULANA MALIK IBRAHIM***

MALANG. Skripsi ini disusun sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik Informatika di Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Terima kasih diucapkan kepada seluruh pihak yang telah meringankan, membantu, dan memberikan semangat kepada penulis. Ucapan terima kasih disampaikan kepada:

1. Bapak Fresy Nugroho, M.T, selaku Pembimbing I yang telah meluangkan waktu untuk membimbing, mengarahkan dan memberi masukan dalam proses pengerjaan skripsi ini.
2. Bapak Irwan Budi Santoso, M.Kom, selaku Pembimbing II yang selalu memberikan masukan, nasehat, serta petunjuk penyusunan skripsi ini.
3. Bapak Prof. Dr. H. Mudjia Raharjo, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Seluruh Dosen Fakultas Sains dan Teknologi Universitas Islam Negeri Mulana Malik Ibrahim Malang, khususnya Dosen Jurusan Teknik Informatika, yang telah membimbing, mengajarkan, dan mencurahkan ilmu-ilmunya kepada penulis.

5. Untuk segenap keluarga besar dan kerabat penulis. Terima kasih atas dukungan moral maupun spiritual sehingga penulisan skripsi ini dapat terselesaikan. Semoga skripsi ini menjadi khasanah kepustakaan baru yang akan memberikan manfaat bagi semua pihak.

Malang, September 2013

Penulis



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	iii
LEMBAR PENGESAHAN	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTTO	vi
PERNYATAAN KEASLIAN TULISAN	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN.....	xiii
ABSTRAK	
.....	xi
v	
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	5
1.3. Batasan Masalah	5
1.4. Tujuan.....	6
1.5. Manfaat	6
1.6. Sistematika Penelitian	7
1.7. Metode Penelitian.....	8
BAB II TINJAUAN PUSTAKA	11
2.1. Simulasi Mengemudi	11
2.2. <i>Pathfinding</i>	12
2.2.1 <i>Onewaypath</i>	14
2.3. <i>Line Detection</i>	15
2.4. <i>Unreal Developmen Kit (UDK)</i>	16
2.4.1 <i>Unreal Script</i>	17

2.5. Tinjauan <i>Game</i> Dari Sudut Pandang Islam	18
BAB III ANALISI DAN PERANCANGAN SISTEM.....	20
3.1. Analisis dan Perancangan Sistem.....	20
3.1.1 Keterangan Umum	20
3.1.2 Gambaran Umum Aplikasi.....	21
3.2. Perancangan Aplikasi.....	21
3.2.1 Perancangan Sistem	23
3.2.2 Use Case Diagram.....	24
3.2.3 Perancangan <i>Pathfinding</i>	25
3.2.4 Perancangan Antarmuka	27
3.2.5 Perancangan Map.....	29
3.2.6 Perancangan Detail Mobil.....	31
3.2.7 Perancangan <i>Role</i> Dalam <i>Game</i>	36
3.3. Kebutuhan Sistem.....	42
BAB IV HASIL DAN PEMBAHASAN	44
4.1. Implementasi <i>Pathfinding</i>	44
4.1.1 <i>Onewaypath</i>	44
4.2. Implementasi Aplikasi.....	47
4.3. Ujicoba Aplikasi.....	55
4.3.1 Pengujian <i>Pathfinding</i> Terhadap <i>Game</i>	55
4.3.2 Ujicoba Pengguna	59
BAB V PENUTUP.....	64
5.1 Kesimpulan.....	64
5.2 Saran	64

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 1.1 Gambar Simulasi Mengemudi.....	11
Gambar 2.1 Gambar <i>Pathnode</i> dan <i>Pathfinding</i>	14
Gambar 2.2 Gambar <i>Onewaypath</i> Bagian dari <i>Pathnode</i>	14
Gambar 2.3 Gambar <i>Navigasi</i> dan <i>Land Area</i>	15
Gambar 2.4 Gambar Perpotongan Bangun Yang Terdeteksi.....	16
Gambar 2.5 Gambar Konten Hasil <i>Unreal Script</i>	17
Gambar 3.1 Gambar <i>Activity Diagram</i> Aplikasi.....	23
Gambar 3.2 Gambar Use Case Diagram Sistem.....	24
Gambar 3.3 Gambar <i>Flowchard</i> cara kerja <i>Pathfinding</i>	25
Gambar 3.4 Gambar Cara Kerja <i>ActorFactory</i>	26
Gambar 3.5 Gambar Menu Utama.....	27
Gambar 3.6 Gambar Halaman Pengaturan Resolusi	28
Gambar 3.7 Gambar Halaman Tampilan Game.....	29
Gambar 3.8 Gambar Desain Map UIN Malang.....	30
Gambar 3.9 Gambar Desain Kampus UIN Malang.....	31
Gambar 3.10 Gambar Desain <i>Geometri</i>	32
Gambar 3.11 Gambar Peletakan <i>Bone</i>	33
Gambar 3.12 Gambar Peletakan <i>Mainbone</i>	34
Gambar 3.13 Gambar Koneksi Antar <i>Bone</i>	34
Gambar 3.14 Gambar Komponen Berat Pada Mobil.....	35
Gambar 3.15 Gambar <i>Flowchard</i> Penunjuk Arah.....	36
Gambar 3.16 Gambar Rancangan Penempatan <i>Trigger</i>	37
Gambar 3.17 Gambar Penempatan <i>Trigger</i> Salah Jalan.....	38
Gambar 3.18 Gambar <i>Flowchard</i> <i>Trigger</i> Salah Jalan.....	39
Gambar 3.19 Gambar Posisi Matahari Dalam Satuan Menit.....	40
Gambar 3.20 Gambar Role Perpindahan Jalur <i>Pathnode</i>	41
Gambar 4.1 Gambar <i>Source Pathnode</i>	44
Gambar 4.2 Gambar <i>Source Class</i> Lalulintas.....	45
Gambar 4.3 Gambar <i>Source</i> Definisi Jarak.....	45
Gambar 4.4 Gambar <i>Source</i> Destinasi <i>Pathnode</i>	46

Gambar 4.5	Gambar Halaman Menu pada <i>Game</i>	47
Gambar 4.6	Gambar Halaman <i>Setting</i>	48
Gambar 4.7	Gambar Halaman Bantuan.....	48
Gambar 4.8	Gambar Perkenalan Pemandu Karakter.....	49
Gambar 4.9	Gambar Penunjuk Arah Quest.....	50
Gambar 4.10	Gambar Perintah Kepada Player.....	50
Gambar 4.11	Gambar HUD Pada Mobil.....	51
Gambar 4.12	Gambar Penunjuk Arah.....	52
Gambar 4.13	Gambar Perintah Penunjuk Arah.....	53
Gambar 4.14	Gambar <i>Reward</i> Mobil.....	53
Gambar 4.15	Gambar <i>Traffic System</i>	54
Gambar 4.16	Gambar Percobaan <i>Pathfinding</i> 200.....	55
Gambar 4.17	Gambar Percobaan <i>Pathfinding</i> 500.....	56
Gambar 4.18	Gambar Percobaan <i>Pathfinding</i> 1000.....	57
Gambar 4.19	Gambar Percobaan <i>Line Detection</i>	57
Gambar 4.20	Gambar Agen <i>Sharing</i>	58

DAFTAR TABEL

Tabel 1.1	Data Kecelakaan Lalu Lintas	1
Tabel 4.1	Data Hasil Ujicoba Perangkat.....	59
Tabel 4.2	Rekapitulasi Pendapat Responden.....	60
Tabel 4.3	Rekapitulasi Prosentase Hasil Ujicoba	61

DAFTAR LAMPIRAN

Lampiran 1 Kode Program

Lampiran 2 Sampel blanko Responden Terisi

ABSTRAK

Hariri, Ahmad Fuad . 2013. 09650014. **DRIVING SIMULATION MENGGUNAKAN METODE PATHFINDING DENGAN STUDI KASUS DI KAMPUS UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG.** Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (I) FresyNugroho, M.T, (II) Irwan Budi Santoso, M.Kom

Kata Kunci: *DrivingSimulation, Pathfinding, UDK*

Mengemudi merupakan aktifitas yang lumrah bagi kebanyakan orang. Namun aktifitas ini memiliki cukup masalah untuk di diselesaikan. Masalah ini terbentuk dari masyarakat, karena kebanyakan dari masyarakat tersebut tidak disiplin dalam mengemudikan kendaraan mereka.

Driving simulation merupakan sebuah simulasi yang menggambarkan kondisi sebenarnya dari lalu lintas kedalam suatu game. Game ini dibuat untuk membantu orang lain memahami disiplin. Khususnya di kampus Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Game ini menggunakan Pathfinding sebagai metode didalam game. Pathfinding sendiri berfungsi sebagai navigasi untuk mengemudikan mobil di dalam NPC sistem. Peneliti menggunakan UDK untuk membangun game ini, karena UDK merupakan versi gratis dari unreal engine.

Selain menampilkan cara mengemudi. Penelitian ini juga mengangkat sisi agama dengan menambahkan doa-doa atau hal lain, yang diharapkan dapat meningkatkan disiplin pengguna, sehingga manfaat dari game ini bisa dirasakan.

ABSTRACT

Hariri, Ahmad Fuad . 2013. 09650014. **DRIVING SIMULATION USING PATHFINDING METHOD CASE STUDY IN STATE ISLAMIC UNIVERSITY OF MAULANA MALIK IBRAHIM MALANG.**
Departement of Informatics Engineering, Faculty of Sience and Technology, State Islamic University of Maulana Malik Ibrahim Malang.
Advisers (I) Fresy Nugroho, M.T, (II) Irwan Budi Santoso, M.Kom

Keyword: *Driving Simulation, Pathfinding, UDK*

Driving is habitual activity of many people. However, this activity has problems to be handled. In fact, the problems emerge from the society, because they are not discipline during driving.

Driving Simulation is a simulation which represents actual situation of traffic into kind of game. This game is invented to help people in conceiving discipline, especially in State Islamic University of Maulana Malik Ibrahim Malang.

This game uses pathfinding method in which its function is as the navigation for driving car in NPC system. In addition, UDK is used by the researcher to create this game, because UDK is free version from unreal engine.

Besides displaying how to drive, this study also raises the religious side by providing a prayers etc. which is expected to improve the discipline of this game users, so the benefits can be felt from it.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Meningkatnya kesejahteraan masyarakat saat ini tentunya merupakan suatu hal yang patut disyukuri, karena hal itu dapat berdampak tercukupinya kebutuhan masyarakat tersebut. Namun hal itu juga memiliki beberapa dampak negatif, salah satunya adalah dalam segi transportasi, dampak tersebut dikarenakan banyaknya orang tua yang memberikan kendaraan bermotor pada anak berusia di bawah umur tanpa memperhatikan kedisiplinan anak dalam berkendara. Hal itu terbukti dari meningkatnya kecelakaan transportasi dari tahun ke tahun seperti data yang diungkapkan BPS mengacu dari kepolisian berikut ini.

Tabel 1 : Data Kecelakaan Lalu lintas

Tahun	Jumlah Kecelakaan	Korban Mati	Luka Berat	Luka Ringan	Kerugian Materi (Juta Rp)
1992	19 920	9 819	13 363	14 846	15 077
1993	17 323	10 038	11 453	13 037	14 714
1994	17 469	11 004	11 055	12 215	16 544
1995	16 510	10 990	9 952	11 873	17 745
1996	15 291	10 869	8 968	10 374	18 411
1997	17 101	12 308	9 913	12 699	20 848
1998	14 858	11 694	8 878	10 609	26 941
1999	12 675	9 917	7 329	9 385	32 755

2000	12 649	9 536	7 100	9 518	36 281
2001	12 791	9 522	6 656	9 181	37 617
2002	12 267	8 762	6 012	8 929	41 030
2003	13 399	9 856	6 142	8 694	45 778
2004	17 732	11 204	8 983	12 084	53 044
2005	91 623	16 115	35 891	51 317	51 556
2006	87 020	15 762	33 282	52 310	81 848
2007	49 553	16 955	20 181	46 827	103 289
2008	59 164	20 188	23 440	55 731	131 207
2009	62 960	19 979	23 469	62 936	136 285
2010	66 488	19 873	26 196	63 809	158 259
2011	108 696	31 195	35 285	108 945	217 435
2012	109.038	25.131	36.710	118.152	277.18

Sumber : Badan Statistik Nasional tahun 2011

Dari data diatas dapat diketahui, terdapat peningkatan kecelakaan lalu lintas tiap tahunnya. Sudah sepatutnya tiap pengendara kendaraan bermotor mematuhi aturan yang diberlakukan pada tiap - tiap wilayah yang dilaluinya, hal tersebut perlu dilakukan untuk menjaga kedisiplinan pengendara dan diharapkan dapat meminimalisir kemungkinan terjadinya kecelakaan. Selain dari segi kedisiplinan, pengendara juga harus memperhatikan transportasi dipandang dari segi agama dan moral. karena hal tersebut sangat mempengaruhi sikap dari pengendara dalam mengendarai kendaraanya, salah satu langkah tersebut dapat

dilakukan dengan memulai segala sesuatu dengan berdoa yang telah diajarkan pada tiap-tiap agama. Dalam islam berdoa merupakan hal yang sangat dianjurkan, dan dipertegas dalam Al-Qur'an :

وَقَالَ رَبُّكُمْ ادْعُونِي أَسْتَجِبْ لَكُمْ ۗ

Berdoalah kepada-Ku, niscaya akan Kuperkenankan bagimu. (Al Ghaafir, ayat : 40).

Pada dasarnya doa merupakan komponen yang dianjurkan dalam melakukan sesuatu pekerjaan termasuk pada saat mengemudikan suatu kendaraan. Dalam ilmu nahwu, kata ادْعُونِي Merupakan *fi'il amr* yang berarti perintah dan dapat diartikan sebagai sebuah hal yang dianjurkan. Kalimat tersebut memakai wazan faala dengan kata awal دَعَا – يَدْعُو – اُدْعُ. Hal tersebut berarti Allah SWT benar-benar menyuruh makhluknya untuk berdoa dan semua doa yang dilakukan akan dikabulkan, namun dalam konteks ini para ulama berpendapat bahwa Allah tetap menyesuaikan dengan kebutuhan kita. Oleh karena itu kita tetap harus tetap melaksanakan doa pada setiap pekerjaan yang kita lakukan. Karena pada dasarnya setiap bencana terjadi, tentunya tidak luput dari campur tangan Allah SWT. Namun sebagai manusia kita tetap wajib melakukan hal yang dapat meminimalisir terjadinya bencana yang ada. Hal itu dipertegas dalam ayat berikut:

الَّذِي خَلَقَ الْمَوْتَ وَالْحَيَاةَ لِيَبْلُوَكُمْ أَيُّكُمْ أَحْسَنُ عَمَلًا ۗ وَهُوَ الْعَزِيزُ الْعَفُورُ

"Yang menjadikan mati dan hidup, supaya Dia menguji kamu, siapa di antara kamu yang lebih baik amalnya. Dan Dia Maha Perkasa lagi Maha Pengampun."

(QS.67. Al Mulk:2)

Dalam ayat *الَّذِي خَلَقَ الْمَوْتَ وَالْحَيَاةَ* jelas-jelas disampaikan bahwa setiap hidup mati seseorang adalah kuasa Allah, namun ayat tersebut tidak berhenti begitu saja, namun masih terdapat ayat *لِيَبْلُوَكُمْ أَيُّكُمْ أَحْسَنُ عَمَلًا* yang juga menuntut manusia untuk tetap berusaha untuk memperbaiki setiap perilaku yang dilakukannya. Dari ayat diatas dapat digunakan untuk mempertegas, bahwa perlu adanya pendidikan kedisiplinan berkendaraan yang digabungkan dengan sisi keagamaan yang berlaku dalam masyarakat .

Untuk mengakomodir semua hal yang telah dipaparkan diatas tentunya perlu adanya suatu metode yang mendukung dan mampu menjawab masalah diatas. Oleh sebab itu peneliti menggunakan dua metode yang saling berketerkaitan yakni *line detection* dan *pathfinding*, kedua metode ini sangat berketerkaitan karena memang kedua metode ini yang membuat game tersebut bisa dijalankan.

Sedangkan untuk pengembangan peneliti menggunakan UDK sebagai alternatif untuk mengembangkan game simulasi ini, dengan alasan UDK adalah *Open Source* Sehingga tidak menyalahi aturan yang ada. Selain itu UDK juga digunakan di berbagai *game* yang sudah terbukti layak untuk dijual.

Dengan berbagai penjelasan diatas maka peneliti berkeinginan untuk membuat sebuah simulasi berbasis PC Platform yang mampu membangkitkan kesadaran Masyarakat terutama remaja Dalam Mengendarai Kendaraanya. Sehingga terbentuk Karakter yang sesuai dengan semboyang UIN Malang Yakni Ulul Albab.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan dapat dirumuskan rumusan masalahnya sebagai berikut:

1. Bagaimana membangun sebuah aplikasi yang dapat mengakomodasi visualisasi 3D yang informatif ?
2. Bagaimana membentuk kedisiplinan dan kepatuhan pemain dalam berlalu lintas ?
3. Bagaimana membuat penunjuk arah dalam kampus uin malang ?

1.3 Batasan Masalah

Mengingat luasnya rute yang dapat di visualisasikan, maka perlu adanya batasan masalah yang ada :

1. Peta yang akan divisualisasikan meliputi kampus UIN Malang dan daerah di sekitar wilayah tersebut. Untuk lingkungan kampus akan dimiripkan bentuk fisik bangunan yang ada.

2. *Game* ini dikerjakan dengan lisensi *freeware* dan terdapat beberapa fitur yang tidak dapat dijalankan karena tidak mendapatkan lisensi.

1.4 Tujuan

Tujuan dilakukannya penelitian ini adalah :

1. Aspek Dasar
 - Memberikan simulasi rute yang tepat kampus Uin Malang.
 - Memperkenalkan kampus UIN Malang dan aturan yang diberlakukan pada kampus tersebut.
2. Aspek Yang Ditekankan
 - Membentuk karakter dan kedisiplinan pengguna yang sesuai dengan sisi keagamaan.

1.5 Manfaat

Hasil dari penelitian ini diharapkan akan memberikan manfaat antara lain :

1. Memudahkan pihak-pihak yang membutuhkan informasi mengenai area dan fasilitas di UIN Malang sesuai dengan penggambaran yang telah disesuaikan dengan keadaan yang sebenarnya.
2. Memberikan informasi mengenai rute di kampus UIN Malang yang sesuai dengan rute yang ada .
3. Mengajarkan Kedisiplinan dalam mengemudikan kendaraan bermotor dilingkungan kampus.

4. Memiliki kebiasaan baik saat mengemudikan kendaraan bermotor.
5. Dapat dijadikan dasar sebagai dasar untuk penelitian selanjutnya.

1.6 Sistematika Penulisan

Adapun sistematika pada skripsi ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan tentang teori-teori yang terkait dengan permasalahan yang diambil.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang analisa yang dilakukan dalam merancang dan membuat aplikasi *Game* simulasi.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas tentang implementasi dari aplikasi yang dibuat secara keseluruhan. Serta melakukan pengujian terhadap aplikasi yang dibuat untuk mengetahui aplikasi tersebut telah dapat berjalan dengan baik dan dapat memberikan solusi dari permasalahan yang dihadapi.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dan saran yang diharapkan dapat bermanfaat untuk pengembangan pembuatan aplikasi selanjutnya.

1.7 Metode Penelitian

Pembuatan skripsi ini terbagi menjadi beberapa tahap pengerjaan yang tertera sebagai berikut:

1. Pengumpulan data-data yang diperlukan

Beberapa metode yang akan dipakai dalam pengumpulan data:

a. Studi Literatur

Pada metode ini penulis akan melakukan pencarian, pembelajaran dari berbagai macam literatur dan dokumen yang menunjang pengerjaan skripsi ini khususnya yang berkaitan dengan Simulasi Mengemudi beserta metode yang diambil.

b. Observasi

Melakukan pengamatan terhadap lingkungan dan situasi yang ingin disimulasikan beserta alur yang mungkin dimasukan kedalam *game*.

c. Browsing

Melakukan pengamatan ke berbagai macam website di internet yang menyediakan informasi yang relevan dengan permasalahan dalam pembuatan aplikasi ini.

2. Analisa data yang telah dikumpulkan

Membuat analisa data yang telah diperoleh dari hasil observasi yaitu dengan menggabungkan dengan laporan survei dan kebijakan pemakai menjadi spesifikasi yang terstruktur dengan menggunakan pemodelan.

3. Perancangan Sistem

Memahami dan melakukan perancangan sebuah sistem yang dapat memberikan pengalaman dan pembelajaran terhadap pengguna.

4. Pembuatan Aplikasi

Melakukan pembuatan aplikasi berbasis dekstop dengan menggunakan UDK sebagai *Game Engine* serta memasukan *Pathfinding* sebagai metode yang dipakai.

5. Ujicoba dan Evaluasi

Menguji seluruh spesifikasi terstruktur dan sistem secara keseluruhan. Pada tahap ini, dilakukan uji coba sistem yang telah sesuai dibangun menggunakan kuesioner. Proses ini dilakukan guna untuk memastikan,

apakah aplikasi sudah berjalan dengan baik dan sesuai desain sistem yang sebelumnya.

6. Penyusunan Buku Skripsi

Tahap terakhir ini merupakan tahap dokumentasi pelaksanaan skripsi. Diharapkan, buku skripsi ini bermanfaat bagi pembaca yang ingin mengembangkan sistem ini lebih lanjut maupun pada lain kasus.



BAB II

TINJAUAN PUSTAKA

2.1 Simulasi Mengemudi

Simulation game merupakan *game* yang mensimulaikan suatu masalah yang dihadapi dan dibuat dalam keadaan mirip dengan dunia nyata sehingga pemain dapat merasakan keadaan sebenarnya pada saat memainkan *game* tersebut. Selain itu sebuah simulasi *game* menuntuk penggunanya untuk disiplin dalam menjalankan *game* yang dimainkannya, karena tujuan sebenarnya dari proses pembuatan *game* tersebut adalah untuk bisa diaplikasikan dalam kehidupan nyata. (Roger McHaney, 2009: 9)

Contoh dari *game* simulasi ini adalah *Microsoft Flight simulation*, *The Sims*, maupun *Football Manager*. Kelebihan dari *game* jenis ini adalah pengguna dapat menerima keahlian yang diajarkan secara bertahap oleh *game* tersebut.



Gambar 1.1 : Gambar Sebuah simulasi Mengemudi.

Terdapat beberapa model yang mendefinisikan simulasi, meskipun model ini mirip dengan simulasi dalam pemrograman, namun model tersebut juga dapat digunakan dalam simulasi sebuah *game*.

Simulasi Statis adalah representasi sistem pada suatu waktu tertentu, atau model yang digunakan untuk merepresentasikan sistem dimana waktu tidak mempunyai peranan, contoh simulasi Monte Carlo (simulasi perilaku sistem fisika dan matematika). Simulasi Dinamis adalah representasi sistem sepanjang pergantian waktu ke waktu, contohnya sistem conveyor di pabrik. Deterministik adalah model simulasi yang tidak mengandung komponen yang sifatnya probabilistik (random) dan *output* telah dapat ditentukan begitu sejumlah input dan hubungan tertentu dimasukkan. Model Simulasi Stokastik adalah model simulasi yang mengandung *input-input* probabilistik (random) dan *output* yang dihasilkan pun sifatnya random (probabilistik). Simulasi Kontinue adalah model simulasi dimana *state* (status) dari sistem berubah secara kontinue karena berubahnya waktu (*continuous change state variables*), contohnya simulasi populasi penduduk. Simulasi Diskrit adalah model suatu sistem dimana perubahan *state* terjadi pada satuan-satuan waktu yang diskrit sebagai hasil suatu kejadian (*event*) tertentu (*discrete-change state variables*), contohnya simulasi antrian. (Roger McHaney, 2009:15-17)

2.2 *Pathfinding*

Pathfinding merupakan satu elemen penting dalam *agen movement* karena *pathfinding* merupakan awal dari pergerakan dari *agen* tersebut. *Pathfinding*

sendiri digunakan untuk mencari rute dari starting poin yang telah ditentukan (Steve Rabin, 2001).

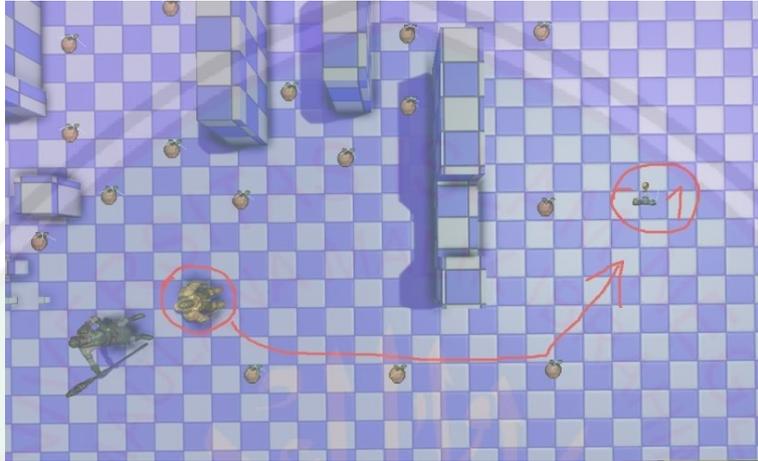
Basic Pathfinding, pada *basic pathfinding* koordinat NPC akan berjalan mengikuti increment dari koordinat x, dan y dari objek. jika koordinat x telah tercapai, namun y belum tercapai, maka y akan dilakukan *increment* hingga mencapai tujuan. Hal sebaliknya juga terjadi pada algoritma ini. Kelemahan algoritma ini adalah jika NPC menemui *obstacle* pada saat dia berjalan.

Random Movement Obstacle Avoidance, algoritma ini mengatasi kekurangan pada *basic pathfinding*. pada saat *basic pathfinding* menemui *obstacle*, maka NPC akan diberikan random arah, sehingga tidak menemukan *obstacle*.

Tracing Around Obstacles, berbeda dengan RMOA, algoritma ini melakukan penelusuran area ke arah tertentu baik ke kiri, atau ke kanan *obstacle*. algoritma TAO sederhana memiliki kekurangan pada pemberhentian tracing around, maka dari itu bisa digabungkan dengan algoritma yang lainnya.

Pathfollowing adalah algoritma yang menjalankan NPC mengikuti arah/pola jalan tertentu. algoritma ini cocok dipergunakan sebagai contoh NPC diharuskan berlari, berjalan mengikuti alur/*track*/jalan tertentu. *Pathfinding* sendiri saling terkait dengan *pathnode*, jika *pathfinding* merupakan rute yang terbentuk maka *pathnode* merupakan titik pembentuk dari rute tersebut, pada gambar 2.1 *pathnode* didefinisikan sebagai node yang ditempel pada map,

sedangkan, pathfinding didefinisikan dengan garis berwarna merah sebagai arah navigasi.

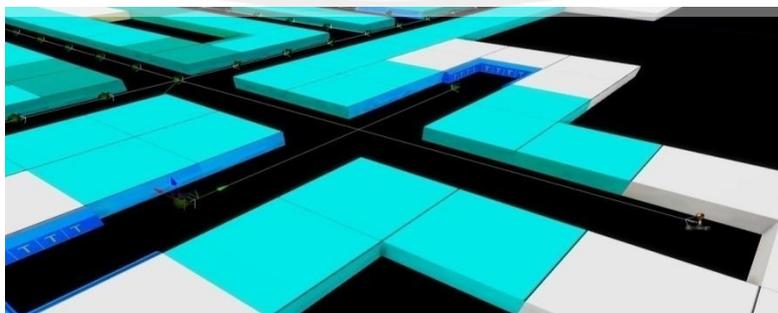


Gambar 2.1 : Gambar *Pathnode* dan *Pathfinding*.

2.2.1 *Onewaypath*

Onewaypath merupakan bagian dari *pathfinding* yang memungkinkan *pathfinding* untuk mendeteksi *pathfinding* lain dengan satu jalur. Fitur ini sangat penting digunakan untuk membuat *simple traffic* dari *pathfinding* tersebut.

(Robert Chin, 2011:141)



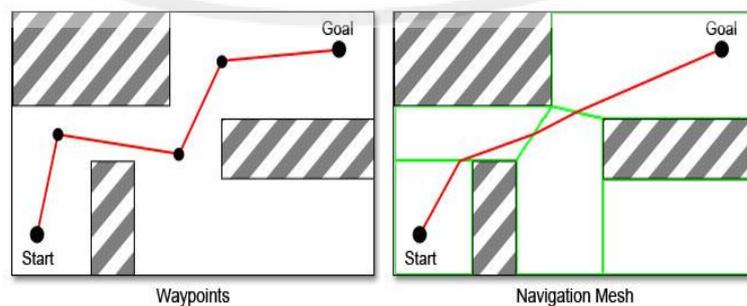
Gambar 2.2 : Gambar *Onewaypath* bagian dari *Pathnode*.

Onewaypath juga dapat digunakan untuk mengisolasi jarak yang terbentuk akibat *pathfinding* yang terlalu dekat dengan *pathfinding* lain yang tidak diinginkan. Karena pada *onewaypath* terdapat fitur untuk memutuskan rantai navigasi yang dibentuk. Namun terdapat kelemahan dalam fitur ini karena programmer dituntut untuk bekerja secara ekstra dalam memutus navigasi yang sering keluar dari titik yang telah didefinisikan.

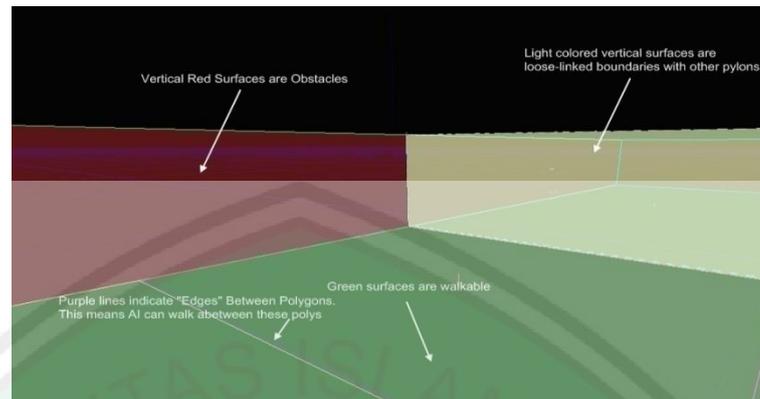
2.3 Line Detection

Line Detection merupakan pendeteksian Garis tepi yang banyak digunakan dalam pencitraan indra maupun gambar. Namun dalam game ini *line detection* digunakan untuk mendeteksi garis dari point yang terbentuk dari *land area* yang telah ditentukan. Dalam udk hal tersebut hanya bisa didefinisikan jika user telah mengatur unreal kismet yang telah disediakan dalam menu yang ada. (Rachel Cordone, 2011: 283)

Land area itu sendiri juga ditentukan oleh banyaknya *pathnode*. Semakin banyak *pathnode*, maka semakin banyak pula *Land Area* yang terbentuk.



Gambar 2.3 : Gambar Navigasi dan Land Area.



Gambar 2.4 : Gambar Perpotongan Bangun Yang Terdeteksi..

Dari gambar 2.3 dan 2.4 dapat dicermati bahwa setiap perpotongan antar bangun dapat dijadikan garis dan nantinya akan dapat dilalui oleh agen, Sehingga secara otomatis agen dapat mengeneralisasi rute yang dibentuk oleh garis tersebut.

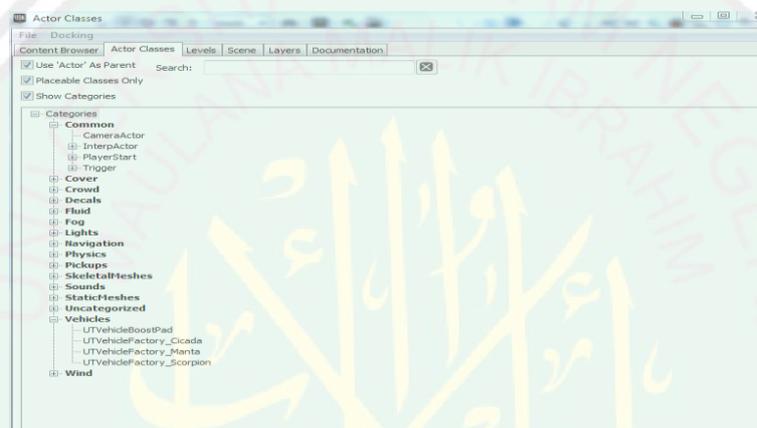
2.4 *Unreal Developmen Kit (UDK)*

UDK merupakan sebuah *Game Engine* yang berbasis dekstop dan IOS *game*. Dimana *engine* ini merupakan versi gratis dari *unreal engine*. Setiap orang dapat menggunakan *engine* ini secara gratis dengan syarat tidak di perjual belikan. Peralatan yang diberikan sama persis dengan *Unreal engine* namun ada beberapa batasan dalam sistem navigasi.

Jika developer ingin menggunakan sistem navigasi maka mereka harus mendefinisikan sendiri sistem tersebut mulai dari awal. Dengan sistem *class* yang telah disediakan.

2.4.1 Unreal Script

Unreal Script merupakan bagian pemrograman dari *unreal* UDK, dimana dalam fitur ini pemrogram dapat melakukan penambahan, deklarasi maupun modifikasi dari obyek programming yang ingin ditambahkan pada *Game Engine* tersebut.



Gambar 2.5 : Gambar Konten dari hasil *Unreal Script*..

Sebagai contoh, jika kita ingin menambahkan mobil didalam *engine* , maka kita harus membuat dan mendeklarasikan *class Vehicle* dan menyambungkannya dengan *class* yang kita buat. Ketika sudah disimpan, maka secara otomatis mobil yang didefinisikan akan tampil di konten browser di dalam UDK.

2.5 Tinjauan *Game* Dari Sudut Pandang Islam

Adapun keterkaitan aplikasi tersebut di lihat dari sudut pandang Islam, mengenai manfaat yang terkandung dari beberapa ayat al Qur'an maupun dari al hadits, antara lain:

1. Islam mengajarkan manusia untuk selalu bersungguh-sungguh dalam suatu kebaikan di jalan Allah. Bahwasannya Allah tidak akan merubah nasib seseorang, jika orang tersebut tidak ingin merubah dirinya sendiri. Ini kaitannya erat dengan sebuah kreatifitas maupun inovasi dalam penerapan teknologi informasi dengan pendidikan, khususnya mengenai pembelajaran suatu ibadah.

Allah SWT berfirman :

لَهُ مُعَقَّبَاتٌ مِّنْ بَيْنِ يَدَيْهِ وَمِنْ خَلْفِهِ يَحْفَظُونَهُ مِنْ أَمْرِ اللَّهِ ۗ إِنَّ اللَّهَ لَا يُغَيِّرُ مَا بِقَوْمٍ حَتَّىٰ يُغَيِّرُوا مَا بِأَنْفُسِهِمْ ۗ وَإِذَا أَرَادَ اللَّهُ بِقَوْمٍ سُوءًا فَلَا مَرَدَّ لَهُ ۗ وَمَا لَهُمْ مِّنْ دُونِهِ مِنْ وَالٍ

“Sesungguhnya Allah tidak merubah keadaan sesuatu kaum sehingga mereka merubah keadaan yang ada pada diri mereka sendiri. dan apabila Allah menghendaki keburukan terhadap sesuatu kaum, Maka tak ada yang dapat menolaknya; dan sekali-kali tak ada pelindung bagi mereka selain Dia.”(QS.ar Ra’d: 11)

2. Islam mengajarkan kita untuk disiplin dalam melakukan segala sesuatu, hal itu tercermin dalam surat An Nisa ayat 59 yang secara harfiah menyuruh kita untuk taat dan berlaku disiplin.

يَا أَيُّهَا الَّذِينَ آمَنُوا أَطِيعُوا اللَّهَ وَأَطِيعُوا الرَّسُولَ وَأُولِي الْأَمْرِ مِنْكُمْ

“Wahai orang-orang yang beriman kepada ajaran yang dibawa Muhammad, taatilah Allah, rasul-rasul-Nya dan penguasa umat Islam yang mengurus urusan kalian dengan menegakkan kebenaran, keadilan dan melaksanakan syariat” (QS.An Nisa 59, Tafsir AL Misbah, Zekr Recitation).

Meskipun dalam arti tidak menyebutkan kata disiplin sama sekali, namun kata أَطِيعُوا merupakan sebuah fi'il atau kata perintah yang juga menuntut kedisiplinan. Kata taatlah disini dapat diartikan melakukan apa yang menjadi peraturan yang diberlakukan. Namun terdapat taat diatas juga terdapat batasan, kita diperbolehkan taat kepada pemerintah yang jujur dan dapat menjadi tauladan bagi rakyatnya, seperti masa Rosulullah Saw.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis dan Perancangan Sistem

3.1.1 Keterangan Umum

Simulasi *Game* ini dikembangkan dalam *desktop platform* dengan menggunakan *UDK engine*. *Engine* ini digunakan untuk mempermudah pembuatan *game* 3D dengan perbandingan luas gedung $\frac{1}{3}$ luas gedung seluruhnya. Terjadinya penyusutan luas gedung ini karena minimnya spesifikasi yang digunakan untuk membangun cahaya di dalam *game*.

Metode yang digunakan pada *game* ini adalah *Pathfinding* dimana metode ini berkaitan dengan arah mobil *traffic* yang sudah didefinisikan oleh *pathfinding* tersebut. Sistem *pathfinding* tersebut menggunakan *navigasi onewaypath* dikarenakan mobil hanya boleh melaju satu arah.

Sedangkan *ActorFactory* digunakan untuk mengaitkan mobil dengan *pathnode* yang digunakan, kemudian fungsi *EnterVehicle* digunakan untuk memasukan *player traffic* kedalam mobil.

3.1.2 Gambaran Umum Aplikasi

Peneliti membuat *game* yang mensimulasikan arah yang dilalui kendaraan di kampus UIN Maulana Malik Ibrahim Malang. Mulai dari pintu masuk sampai pintu keluar. Yang mana hal itu akan diperlihatkan dengan *traffic system* menggunakan *pathfinding*. Selain itu pengguna bisa melaksanakan beberapa *quest* untuk menyelesaikan misi yang diberikan dan mendapat *reward* terhadap beberapa mobil yang disediakan.

Selain itu pengguna juga dapat mempelajari beberapa kegiatan yang diharapkan diharapkan mempunyai nilai positif bagi perilaku pengguna. Mulai dari sholat membaca bismillah saat mengendarai mobil dan beberapa hal lain yang tidak dapat disebutkan secara terperinci pada bab ini.

3.2 Perancangan Aplikasi

Perancangan aplikasi digunakan untuk menentukan kinerja aplikasi yang akan dibuat. Perancangan aplikasi bertujuan untuk menentukan *input*, *proses* maupun *output* yang akan dilaksanakan dalam aplikasi yang sedang dikerjakan. Selain itu perancangan aplikasi juga bertujuan untuk menentukan spesifikasi dan kemampuan aplikasi dalam menangani permasalahan yang telah dibebankan kepada aplikasi tersebut.

Hal penting lainnya yang menjadi perhatian pada perancangan sistem ini adalah bahwa merancang yang dibuat diharapkan dapat digunakan dengan mudah oleh para pengguna dan tidak hanya digunakan bagi yang ahli saja, sehingga

sistem dapat digunakan secara menyeluruh. Selain itu beberapa hal yang harus diperhatikan antara lain adalah kinerja program yang baik dalam mengoperasikan aplikasi yang dibuat.

Secara garis besar perancangan sistem aplikasi ini terdiri dari beberapa bagian, diantaranya *input*, *proses* penilaian, dan *output popup screen game* tersebut.

1. *Input*

Inputan terdapat pada input arah mulai dari tombol W untuk maju S untuk mundur A untuk belok kiri D untuk belok kanan.

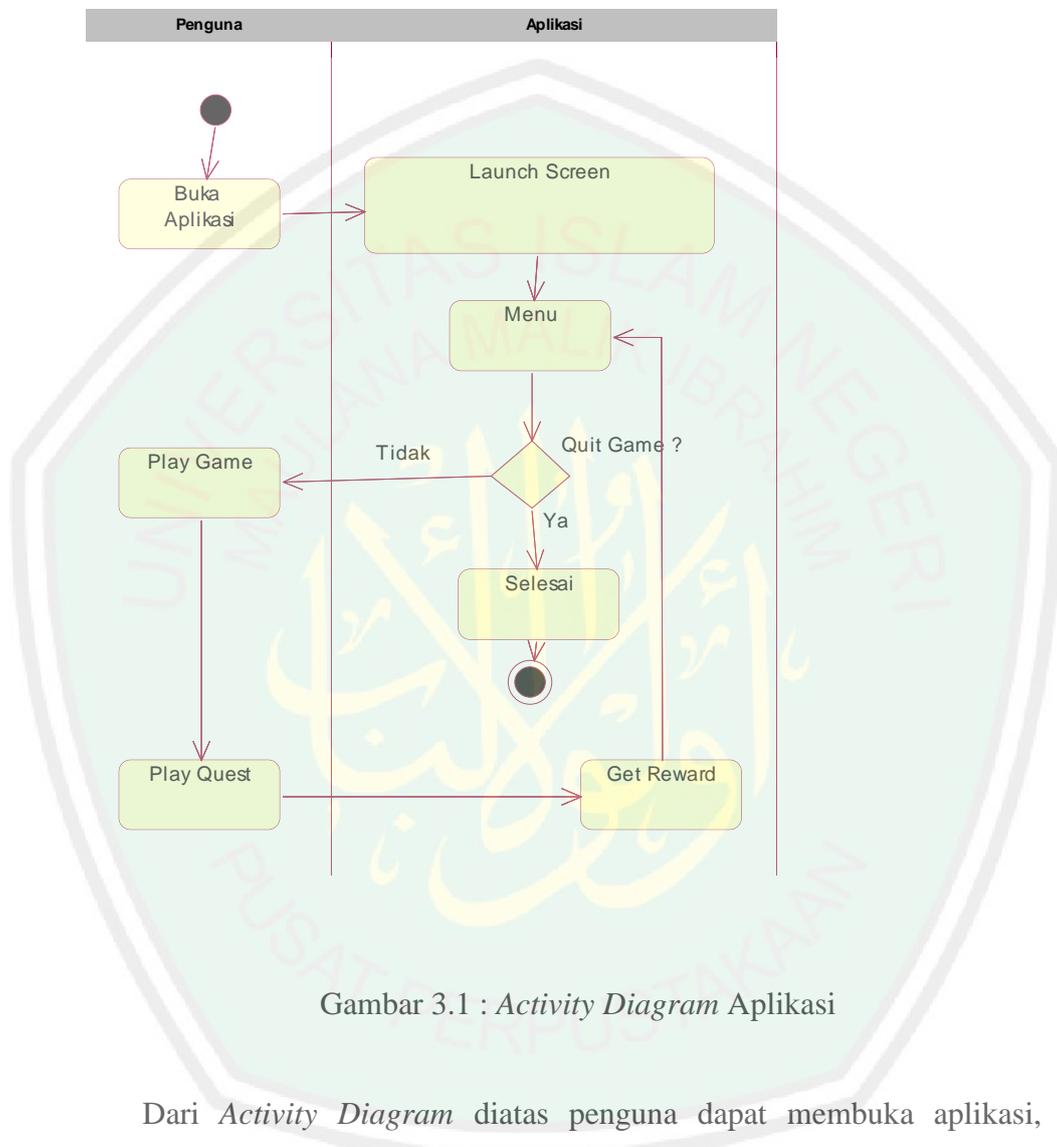
2. *Proses*

Pada proses ini memunculkan *spawnpoint* dan *rule game* yang telah diatur di *game kismet*.

3. *Output*

Output merupakan tampilan *game* yang menampilkan beberapa perintah dan alur *game*. Dalam hal ini adalah semua *quest* yang menjadikan *game* memiliki alur cerita.

3.2.1 Perancangan Sistem

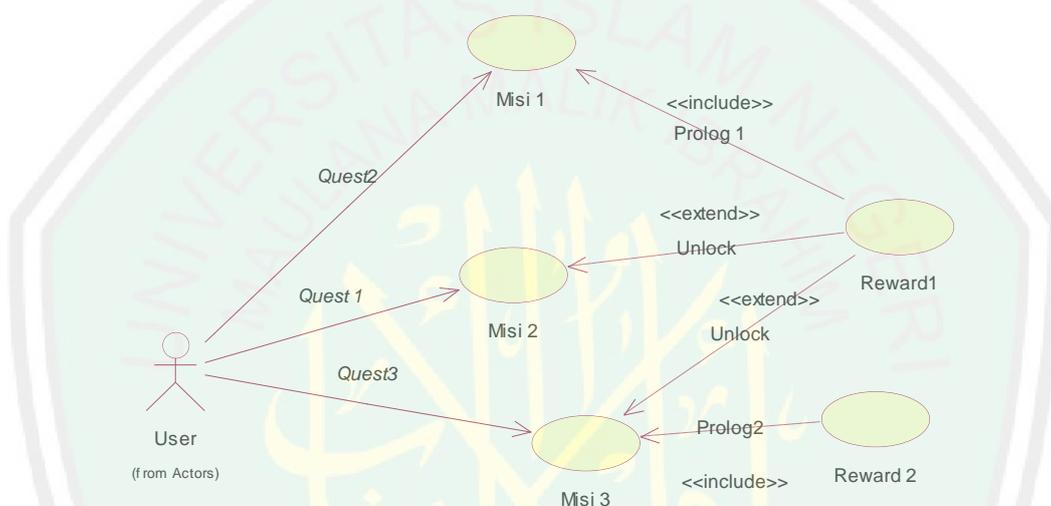


Gambar 3.1 : Activity Diagram Aplikasi

Dari *Activity Diagram* diatas pengguna dapat membuka aplikasi, lalu pengguna dapat memilih di *main* menu untuk memainkan *game*, ataupun mengahiri *game* dengan menekan tombol *exit*. Jika pengguna memainkan *game*, maka pengguna tersebut akan memainkan *quest* dengan imbalan *reward* yang telah didefinisikan pada *game*.

Selain itu pengguna juga dapat melanjutkan lagi ke *quest* lain dan bisa mengahiri *game* malalui menu yang disediakan. Didalam menu tersebut terdapat beberapa bagian yang dimasukan pada sub menu.

3.2.2 Use Case Diagram



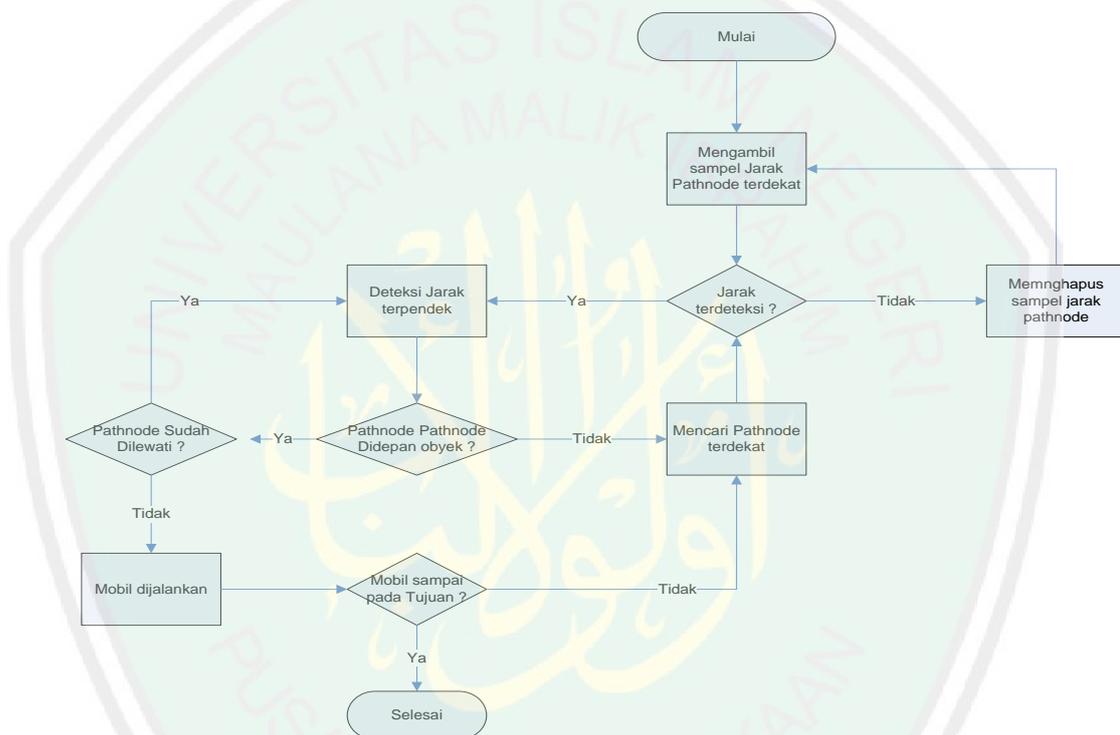
Gambar 3.2 : Use Case Diagram Sistem

Use case class digunakan untuk memodelkan dan menyatakan unit fungsi atau layanan yang disediakan oleh sistem, *use case* dibuat agar pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

Dari *Use case* diatas bahwa aktor utama yang berperan dalam penggunaan *game* tersebut adalah *user*. Dimana *user* dapat mengakses *Quest 1* sebagai perkenalan terhadap kampus dan areanya. *Quest 1* akan menampilkan *reward* Sehingga dapat untuk membuka *quest* kedua dan ketiga. Sedangkan untuk *quest* kedua tidak terdapat *reward*. Sehingga bisa langsung untuk menjalankan *quest*

ketiga. Untuk *Quest* ketiga akan mendapatkan *reward* yang bisa digunakan untuk membuka *Quest* lainnya. Setiap *quest* akan berisikan beberapa macam perintah yang disesuaikan dengan situasi di *game* tersebut.

3.2.3 Perancangan *PathFinding*



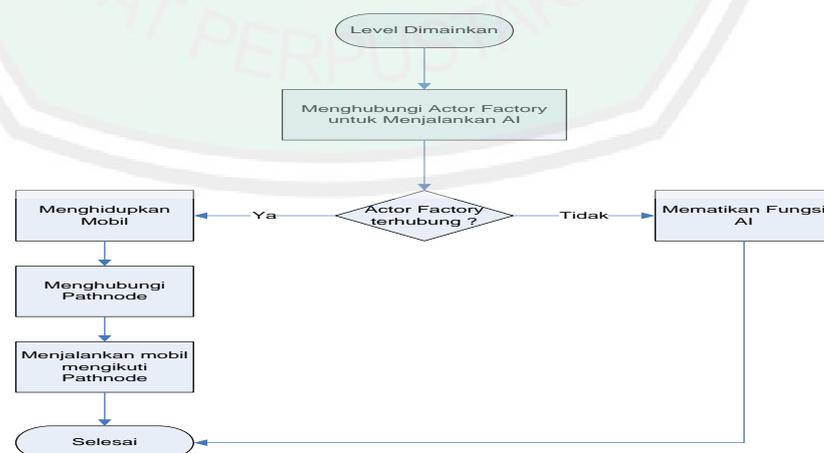
Gambar 3.3 : *Flowchart* Cara Kerja *Pathfinding* terhadap *Pathnode*.

Dari *flowchart* diatas dapat dilihat bahwa fungsi *pathfinding* adalah mencari jarak yang paling dekat dan belum dilalui. Pada saat *game* mulai di jalankan, *engine* akan menyimpan koordinat x dan y dari seluruh *pathfinding* yang ada pada level *game* tersebut kedalam *CollisionCylinder*. Koordinat x merupakan lebar *pathfinding* dan koordinat y adalah tinggi *pathfinding*. Keakurasian *pathfinding* tergantung dari berapa jarak dari *pathnode* diletakan dan berapa nilai

x yang didefinisikan pada *pathfinding* tersebut. Dalam metode ini juga mengharuskan posisi *pathfinding* berada di depan *pathfinding* yang sedang dilewati. Hal tersebut digunakan untuk mengurangi kemungkinan mobil untuk mundur dan keluar dari jalur yang telah ditetapkan.

Ketika *pathnode* tujuan sudah terdeteksi oleh mobil dan *pathfinding* sudah menjalankan fungsi navigasi, maka secara langsung *pathfinding* akan mencari *pathnode* didepanya tanpa harus mencari jalur *pathfinding* terdekat. Hal tersebut berguna untuk mengurangi waktu *spawn pathnode*.

Ketika *Pathnode* tujuan belum terdeteksi oleh mobil, maka mobil akan melakukan pencarian *pathnode* terdekat lainnya, Sehingga mobil secara acak akan mengulangi proses pencarian *pathnode* terdekat. Jika jarak *pathnode* hilang maka secara otomatis akan merestart *ActorFactory* yang menjalankan fungsi *Pathfinding*.



Gambar 3.4 : Cara kerja *ActorFactory* terhadap *Pathnode*.

ActorFactory merupakan elemen yang sangat penting dalam UDK, karena *ActorFactory* berfungsi untuk menjalankan pembangkit kecerdasan yang digunakan oleh *pathfinding*. Ketika *game* dijalankan *ActorFactory* akan dipanggil oleh *super class* bernama *LevelLoaded*. Ketika *ActorFactory* telah berjalan pada sistem *game*. Maka *ActorFactory* akan membangkitkan *pathfinding* dan mendefinisikan fungsi *pathfinding* kedalam mobil yang sedang dijalankan oleh AI tersebut. Namun jika *ActorFactory* tidak bisa dijalankan, maka otomatis semua AI akan dimatikan, dan *traffic* tidak akan berjalan.

3.2.4 Perancangan Antarmuka

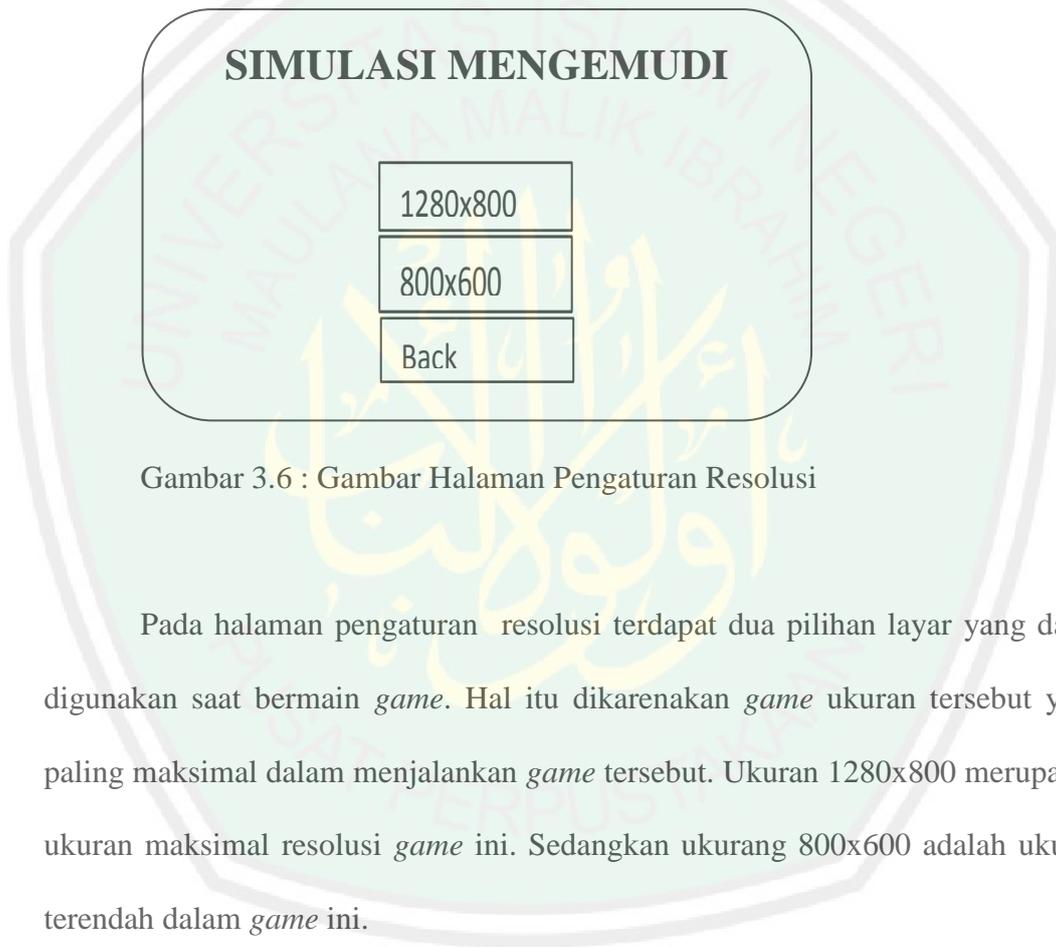
Dalam aplikasi ini terdapat beberapa halaman berikut:

1. *Main Menu*



Gambar 3.5 : Gambar Menu Utama

Pada *Main menu* akan ditampilkan beberapa pilihan kepada *user* yang memungkinkan *user* untuk masuk kedalam *game*. Mengatur resolusi layar dan keluar dari *game* tersebut. Pada halaman ini tidak dipasang *save game* karena memang di dalam *game* ini tidak didukung untuk menyimpan permainanya.



Gambar 3.6 : Gambar Halaman Pengaturan Resolusi

Pada halaman pengaturan resolusi terdapat dua pilihan layar yang dapat digunakan saat bermain *game*. Hal itu dikarenakan *game* ukuran tersebut yang paling maksimal dalam menjalankan *game* tersebut. Ukuran 1280x800 merupakan ukuran maksimal resolusi *game* ini. Sedangkan ukuran 800x600 adalah ukuran terendah dalam *game* ini.

2. Tampilan *Game*



Gambar 3.7 : Halaman Tampilan *Game*

Pada halaman ini akan menampilkan beberapa tampilan tergantung situasi dari *game* tersebut.

- a. Jika *player* berada dalam *vehicle* maka akan menampilkan *spedoometer*.
- b. Jika *player* tidak menaiki *vehicle* maka HUD akan kosong dan tidak menampilkan apa-apa.
- c. Dalam waktu tertentu akan ditampilkan doa-doa dalam tampilan layar.

3.2.5 Perancangan Map

Sebelum melakukan pembuatan map, perlu dilakukan perancangan map terlebih dahulu hal tersebut dimaksudkan agar map dapat menyerupai dengan

bentuk asli dari lokasi yang dipetakan. Dalam *game* ini peneliti menggunakan standar peta arsitektur uin malang yang dikeluarkan oleh kampus dan sudah disesuaikan dengan gedung sebenarnya.



Gambar 3.8 : Desain Map UIN Malang.

Desain diatas dibuat dengan format cdr dan sudah dilakukan pengukuran secara tepat terhadap lokasi antar bangunan. Selain peta penelti juga menggunakan gambar asli tiap – tiap gedung di uin malang sebagai acuan pembuatan model 3D, hal tersebut dilakukan untuk mendapatkan detail model yang maksimal.



Gambar 3.9 : Desain kampus UIN Malang.

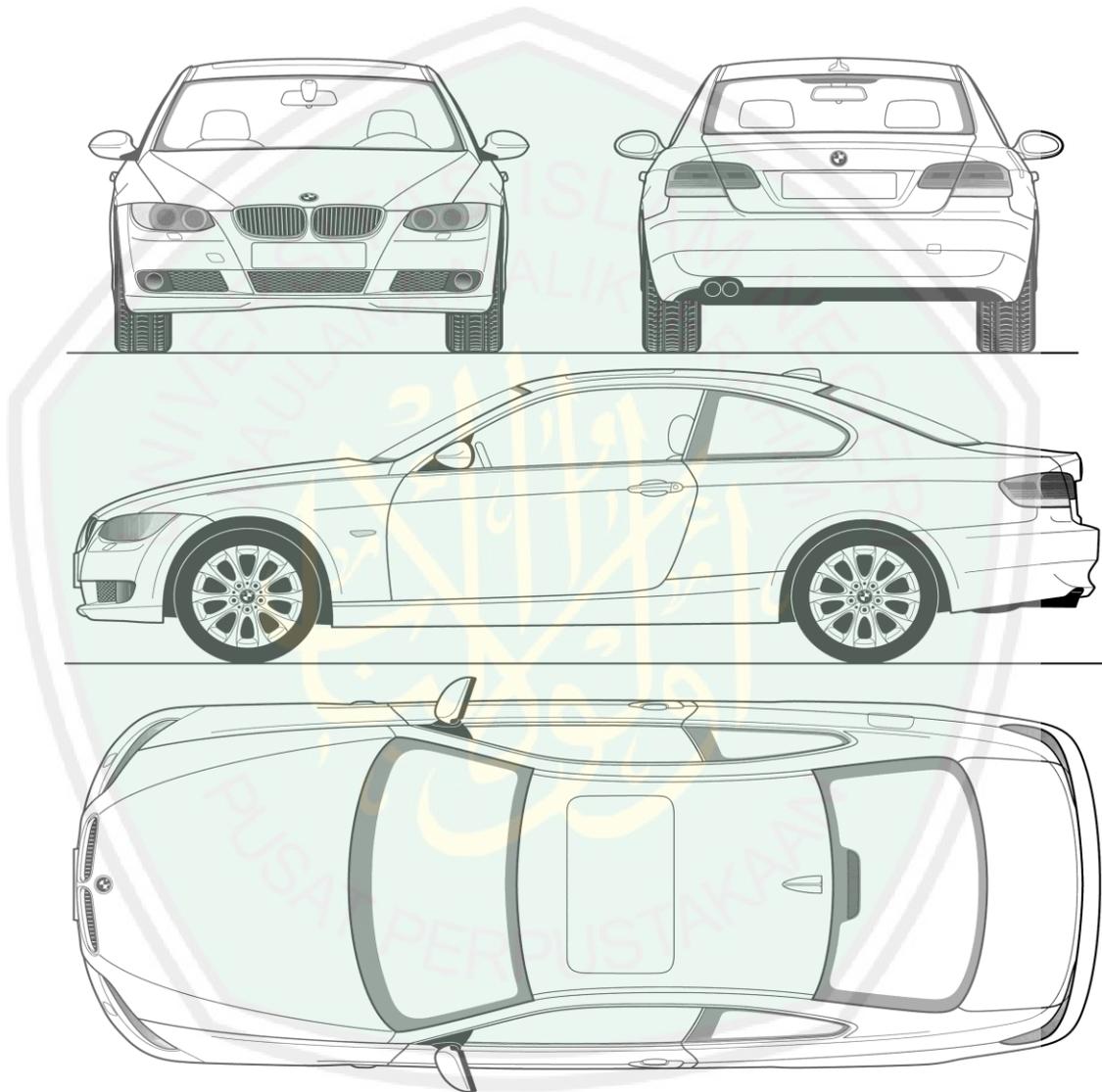
3.2.6 Perancangan Detail Mobil

Pada tahap ini akan dilakukan beberapa pembuatan mobil, hal tersebut dilakukan untuk membuat mobil bisa terdeteksi di UDK sebagai *mesh*. Sehingga dapat dikonfigurasi dengan *unreal script* yang telah dibuat. Terdapat beberapa tahapan yang harus dilakukan dalam pembuatan mobil yang dapat diimport ke dalam UDK. Tahapan tersebut akan dijelaskan sebagai berikut :

1. *Modeling Geometri*

Pada tahap ini perlu dilakukan permodelan mobil yang digunakan. Hal tersebut akan menentukan bentuk dan detail dari mobil yang akan diimport. Dalam permodelan tersebut model *geometri* dibatasi harus tidak

lebih dari 6500 *poly*. Jika lebih dari 6500 secara otomatis udk akan mengagalkan proses *import*.



Gambar 3.10 : Desain *Geometri*

Hal lain yang harus diperhatikan adalah posisi ban dan *geometri* ban harus terpisah dengan *main body* dari mobil tersebut, agar *bone* dapat diletakan di masing-masing ban tanpa tercampur dengan *bone* yang lain

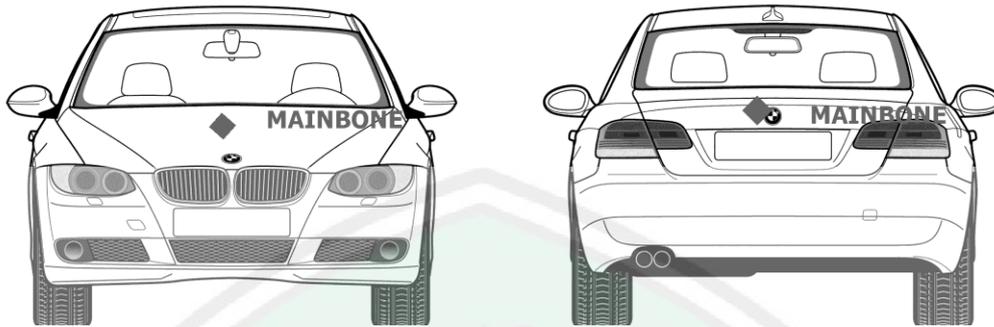
2. Memasang *Bone* dalam Mobil

Tahap selanjutnya adalah pemasangan *bone* didalam ban mobil. Hal ini diperlukan untuk dapat menjalankan animasi dan mengontrol arah laju dari mobil tersebut. Selain itu *bone* juga berfungsi untuk menentukan keseimbangan berat dari mobil. Tiap satu mobil terdapat minimal 5 *bone*, yakni *MAINBONE*, *DKA*, *DKI*, *BKA* dan *BKI*. *Mainbone* untuk bodi mobil. *DKA* berarti depan kanan sedangkan *DKI* depan kiri.



Gambar 3.11 : Peletakan *Bone* Pada Roda Mobil

Dalam peletakan main *bone* pada bodi mobil harus tepat ditengah, hal tersebut sangat berpengaruh dengan berat yang terdapat dalam mobil. Jika *mainbone* tidak tepat maka mobil bisa saja terbalik karena keseimbangannya tidak sama.



Gambar 3.12 : Peletakan *Mainbone*

Selain itu kondisi *bone* juga harus saling terkoneksi satu sama lain agar ketika mobil dijalankan tidak terjadi *bone lost* yang mengakibatkan mobil kehilangan bentuk utamanya. Cara yang paling mudah dalam menghubungkan antar *bone* adalah menggunakan *link tools*.

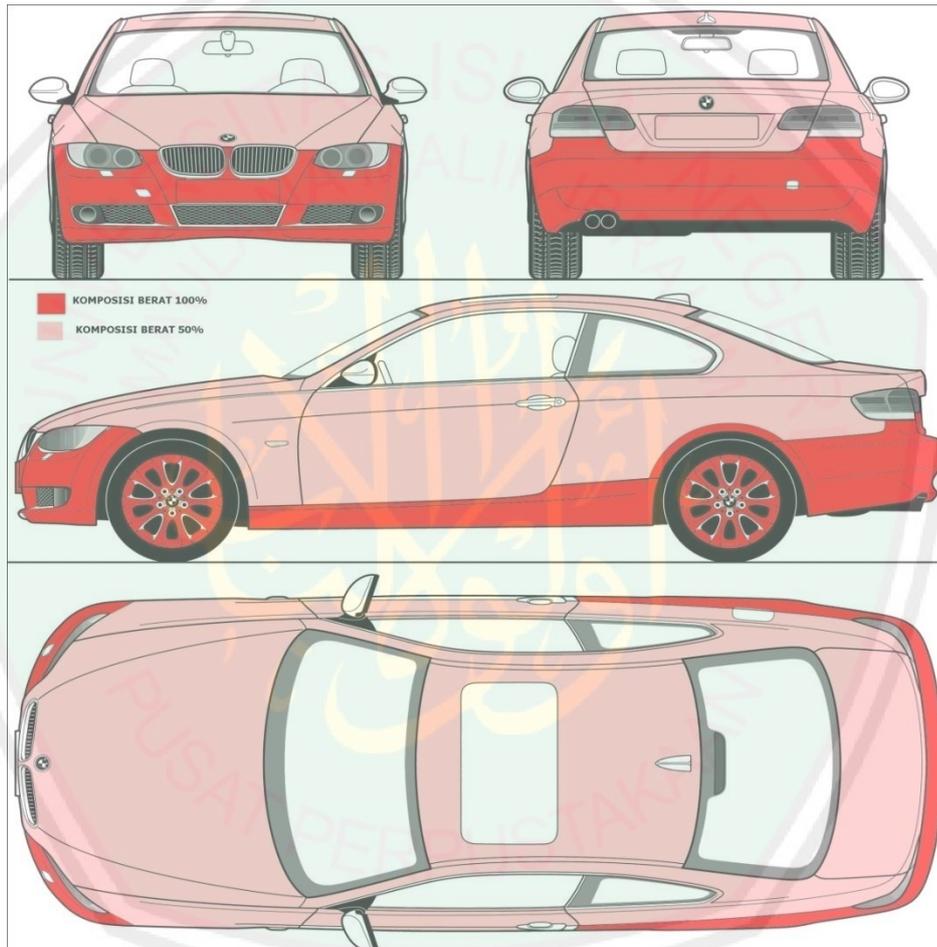


Gambar 3.13 : Koneksi antar *bone*

3. Perancangan Berat Mobil

Setelah melakukan pengaturan *bone* dan *geometri*, maka mobil harus diatur berat di tiap – tiap *geometri* yang terhubung dengan *bone*. Hal ini

dilakukan untuk mengurangi kemungkinan mobil kehilangan keseimbangan. Selain itu berat pada mobil akan mempengaruhi *handling* pada mobil tersebut.



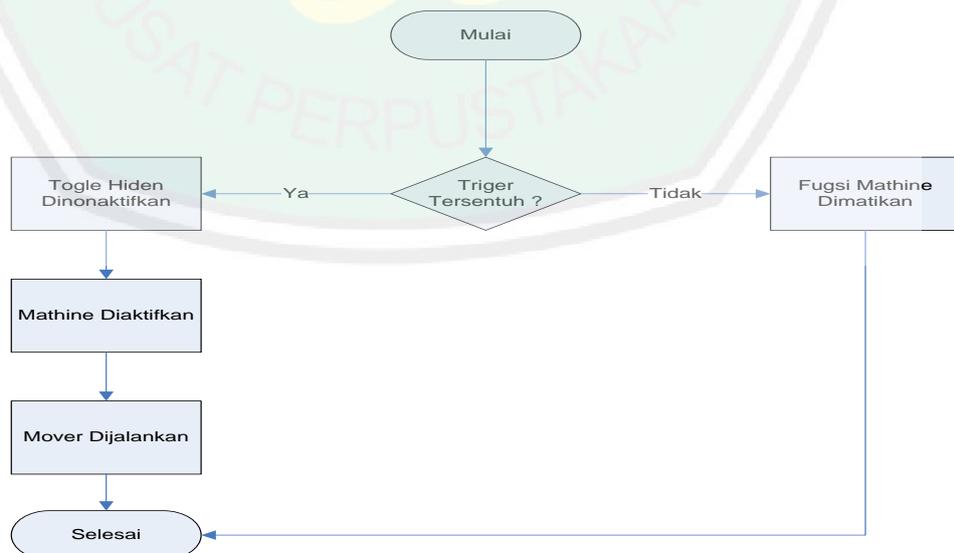
Gambar 3.14 : Komponen Berat Pada Mobil

Terdapat dua tipe berat yang ada di dalam desain mobil, berat tersebut direpresentasikan dengan warna merah muda dan warna merah. Semakin berat suatu *volume* mobil tersebut maka warna didalam desain akan berwarna semakin merah.

3.2.7 Perancangan *Role* Dalam *Game*

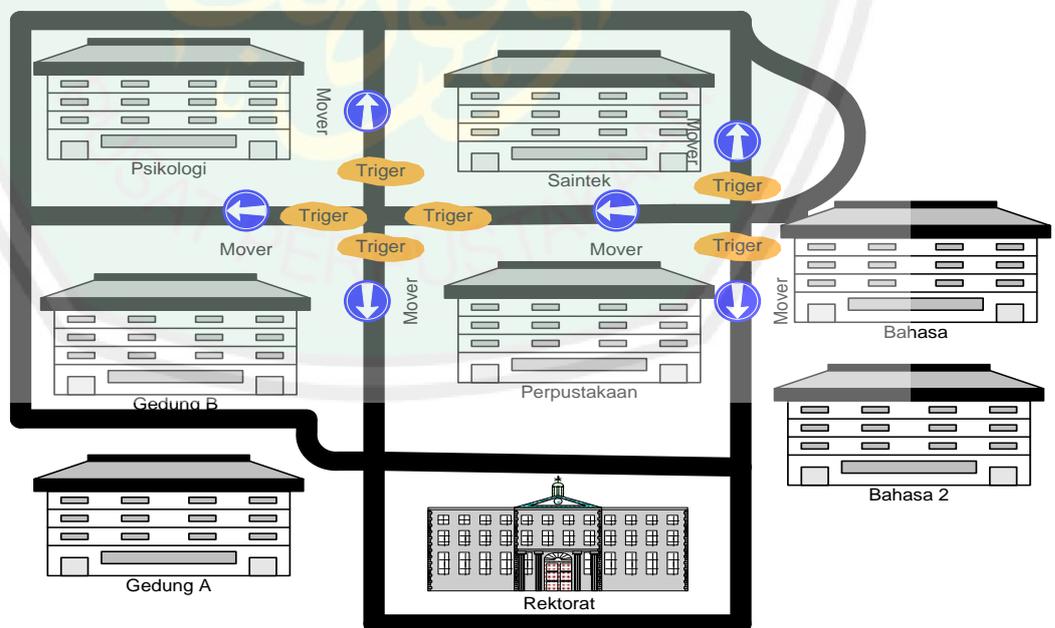
1. Sistem Penunjuk Arah

Di dalam *game* ini terdapat sistem penunjuk arah, sistem ini bermanfaat untuk menampilkan suatu penunjuk arah yang digunakan untuk menuntun *user* mengikuti arah lalu lintas yang terdapat pada kampus uin malang. Untuk membangkitkan sistem ini *player* harus melalui *trigger* sebagai pembangkit dari penunjuk arah. Setiap *trigger* memiliki tinggi dan lebar area yang telah ditentukan, apabila *trigger* tersebut dilalui maka otomatis akan membangkitkan fungsi yang telah didefinisikan oleh programmer. Setiap *trigger* dapat diatur berapa detik *trigger* tersebut melakukan *delay*. *Delay* disini digunakan untuk memperlambat reaksi dari *trigger*, didalam beberapa kasus hal tersebut sangat bermanfaat dalam memperluas waktu user terhadap interaksi sekitar.



Gambar 3.15 : *Flowcard* Sistem Penunjuk Arah menggunakan *Trigger*

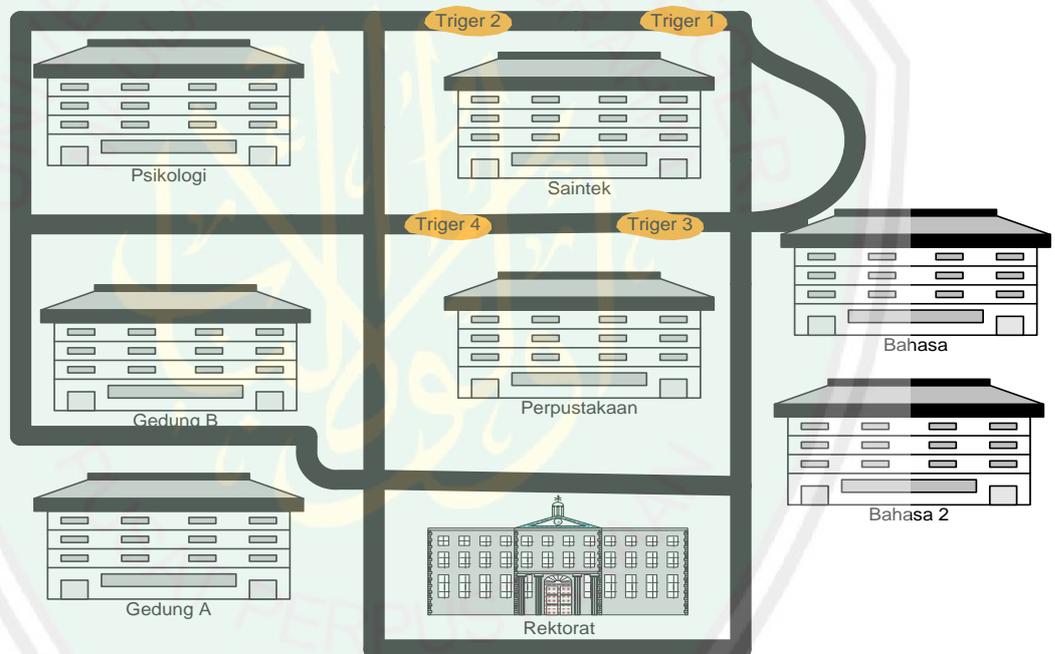
Didalam *game* ini terdapat lima *trigger* yang berfungsi untuk menampilkan penunjuk arah. Setiap *trigger* tersebut akan menampilkan penunjuk yang berbeda, ketika *trigger* tersentuh *player* atau *traffic* maka *toggle hidden* akan dinonaktifkan dan penunjuk arah dapat dilihat oleh *player*, selain *toggle hidden trigger* juga akan mengaktifkan *mathine* yang berisikan animasi arah jalan yang sesuai. Ketika *player* sudah tidak didalam area *trigger* maka *toggle hidden* dinonaktifkan Sehingga *player* tidak dapat melihat penunjuk arah tersebut. Dan *mathine* akan segera dinonaktifkan. Untuk menonaktifkan atau mengaktifkan semua diatas, *mathine* dan *toggle hidden* perlu terhubung dengan *mover*, *mover* adalah hasil konversi dari *Static Mesh*.



Gambar 3.16 : Rancangan Penempatan *Trigger*

2. Sistem Salah Jalan

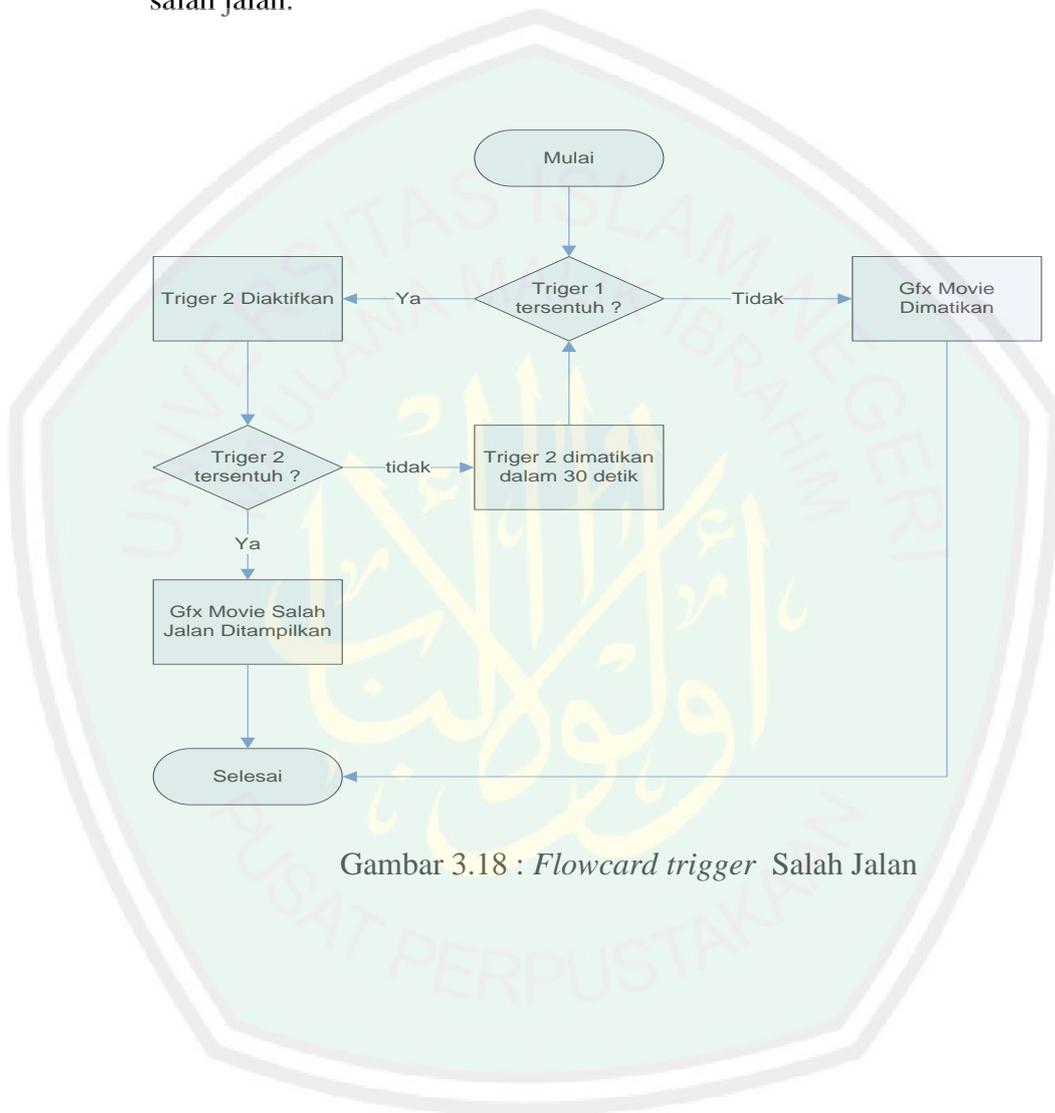
Sama seperti sistem penunjuk arah, sistem salah jalan juga menggunakan *trigger* sebagai penggerak sistem. Pada sistem ini *trigger* akan terhubung dengan *trigger* lainnya. *Trigger* pertama akan dihubungkan dengan *trigger* kedua, dengan menggunakan *toggle*, *toggle* disini berfungsi untuk menonaktifkan *trigger* kedua.



Gambar 3.17 : Penempatan *Trigger* Salah Jalan

Ketika *trigger* pertama dilewati oleh mobil *player* maka *trigger* kedua akan diaktifkan oleh *trigger* pertama. Jika pada keadaan aktif dan *trigger* dua di lalui oleh mobil, maka secara otomatis akan menampilkan keadaan salah jalan. Tampilan salah jalan hanya akan tampil apabila mobil tersebut melalui *role* yang sudah ditentukan, Yakni *trigger* satu lebih dulu

dilewati, apabila mobil melewati *trigger* dua terlebih dahulu maka *trigger* tersebut tidak akan menampilkan apapun karena mobil tidak mengalami salah jalan.



Gambar 3.18 : *Flowcard trigger* Salah Jalan

3. Sistem Siang Malam

Pada sistem ini *mathine* digunakan untuk mengatur *dominan light* dengan drajat perputaran 90 drajat tiap 15 menit. Karena pada *game* ini 24 jam didefinisikan 1 jam dalam *game*. Untuk memutar posisi matahari dibutuhkan *movement track* di dalam *mathine*. Setelah *movement track* dibuat maka perlu didefinisikan *key* pada *track* tersebut, terdapat lima *key*

track dimana tiap – tiap *key* tersebut dipasang pada setiap posisi +15 menit dalam satu jam. Di dalam *game* ini juga akan menampilkan Azan yang disesuaikan dengan waktu sebenarnya dalam satu jam tersebut.

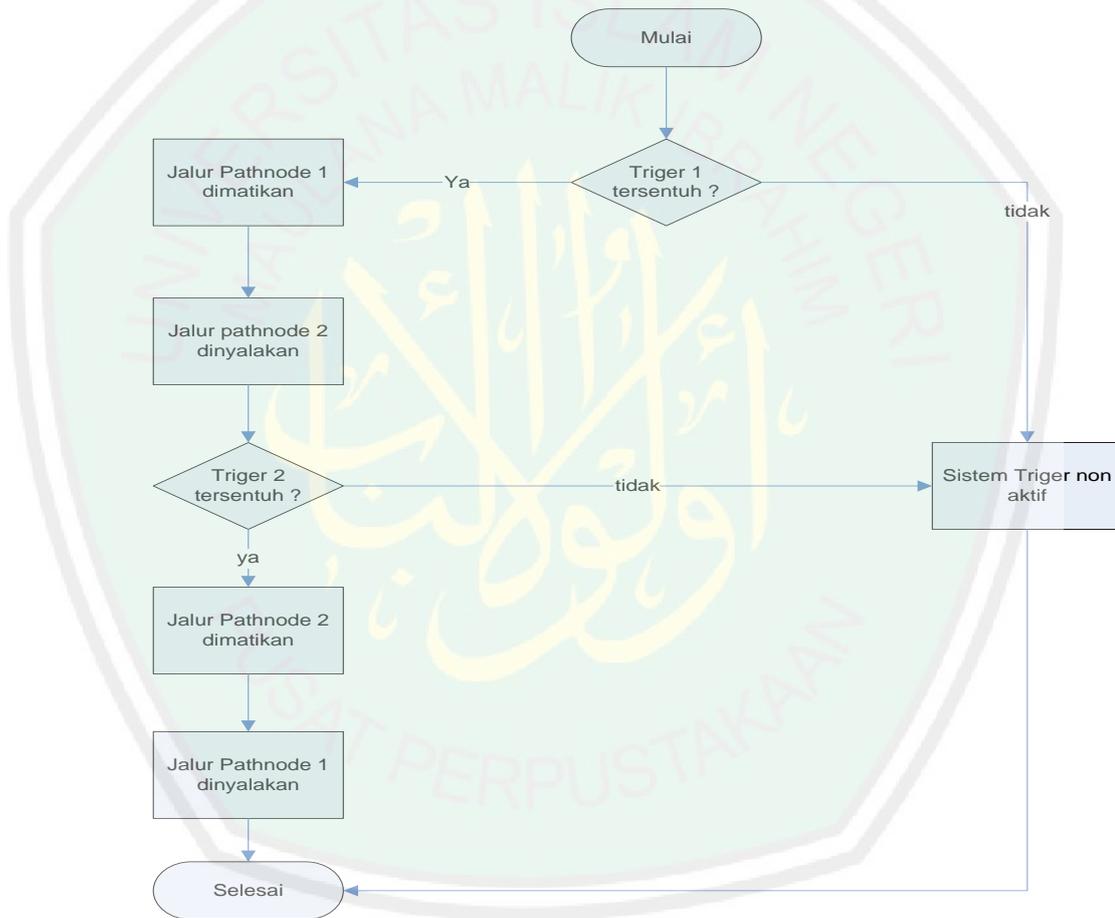


Gambar 3.19 : Gambar Posisi Matahari Dalam Satuan Menit

4. Sistem NPC

NPC pada *game* ini adalah mobil *traffic* yang digerakan oleh *actorfactory* melalui jalur *pathfinding*. Pada NPC inilah metode *pathfindinig* dijalankan. Dalam sistem ini ketika *game* dimulai *actorfactory* akan menghubungi mobil dengan *pathfinding* dan yang telah didefinisikan sebelumnya. Lalu *actorfactory* akan melakukan *spawn* kedalam mobil yang disediakan oleh sistem. Ketika mobil sudah siap maka mobil akan diatur untuk melalui jalur *pathnode* yang telah disediakan. Selain itu pada *game* ini juga terdapat *role* untuk berpindah jalur. *Role* ini

ditanamkan pada jalur di perempatan perpustakaan uin malang, dimana mobil *traffic* dapat melalui jalur lurus ke saintek atau lewat jalur selatan psikologi. Perpindahan jalur tersebut menggunakan *trigger switch* yang telah didefinisikan oleh peneliti.



Gambar 3.20 : *Role* Perpindahan Jalur *Pathnode*

3.3 Kebutuhan Sistem

Berikut ini beberapa perangkat keras maupun lunak yang dibutuhkan untuk mendukung pembuatan dan uji coba *Game* Simulasi Mengemudi.

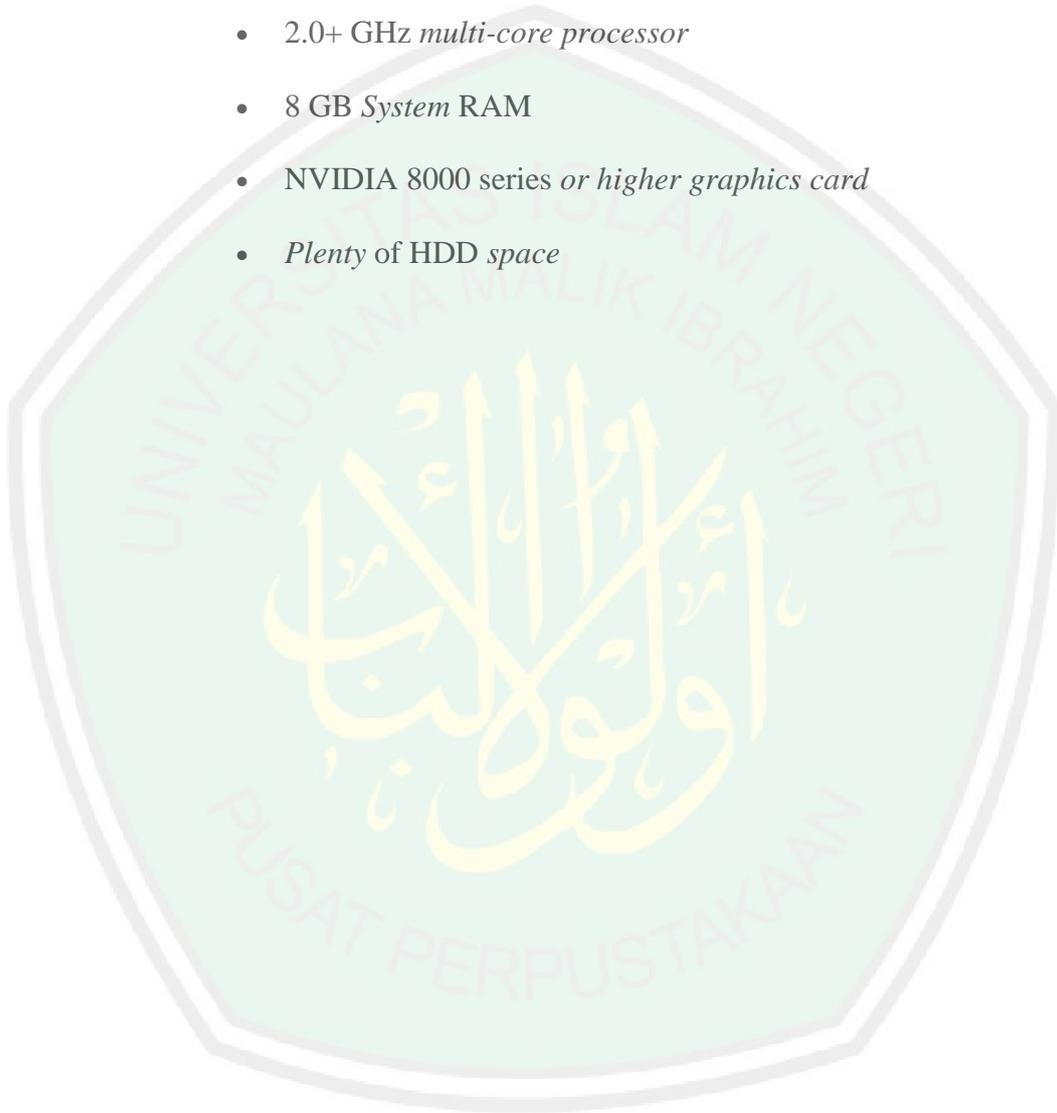
1. Perangkat Keras (*Hardware*)
 - a. PC / Laptop dengan spesifikasi minimal : *Processor* Intel(R) Core(TM) 2 Duo T6600 @ 2.20GHz (2 CPUs) dan *Memory* 4000 MB RAM, digunakan untuk pembuatan aplikasi, VGA Nvidia, Ati 512 Mb.
 2. Perangkat Lunak (*Software*)
 - a. UDK. *Update* Juni 2013 digunakan untuk membuat visual game. Dan melakukan pemasangan fungsi sistem yang ada.
 - b. *Notepad++*, digunakan untuk melakukan *Scripting Pathnode* dan *Vehicle* dan beberapa tambahan lain di UDK.
- Untuk UDK sendiri memasang spesifikasi minimal untuk direx 9

Minimum:

- Windows XP SP3 (32-bit only), Windows Vista, or Windows 7
- 2.0+ GHz processor
- 2 GB *system* RAM
- SM3-compatible video card
- 3 GB *free hard drive space*

Recommended for Content Development:

- Windows 7 64-bit
- 2.0+ GHz *multi-core processor*
- 8 GB System RAM
- NVIDIA 8000 series *or higher graphics card*
- *Plenty of HDD space*



BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi *Pathfinding*

4.1.1 *OnewayPath*

Pathfinding merupakan sebuah metode navigasi dimana agen akan dibimbing dari starting poin menuju ke beberapa area yang telah didefinisikan oleh pengembang aplikasi. Dalam game ini akan digunakan *pathfinding* berjenis *onewaypath*. Dimana jalan yang digunakan hanya satu arah. Dan *pathfinding* hanya akan mencari jalur didepanya. Hal ini digunakan untuk menstabilkan jalan kendaraan yang akan melalui *pathfinding* ini.

```

class Pathnodekedua extends PathNode;

Begin Object Name=CollisionCylinder
    CollisionRadius=+150.000000
    CollisionHeight=+00100.000000
Begin Object Name=Arrow
}

```

Gambar 4.1 : *Source Pathnode*

Dari kode diatas *class Pathnodekedua* akan dideklarasikan dengan *pathnode* sehingga dapat menjadi anak *class* dari *pathnode* itu sendiri. Hal ini dikarenakan *pathnode* adalah *class* dari *engine* yang tidak dapat dirubah sehingga harus dibuat anak *class* yang sama seperti kelas induk. Sedangkan *CollisionCylinder* merupakan penampung definisi dari seberapa luas *pathnode* yang akan dibentuk.

Selain *CollisionCylinder* terdapat juga *MaxSpeed* yang berfungsi untuk mengatur kecepatan dari mobil tersebut. Sehingga mobil tidak dapat melaju dengan kecepatan penuh dan keluar dari *pathnode*.

```

lass Lalulintas2 extends UTBot;

    Pawn.SetMovementPhysics();
    if (Pawn.Physics == PHYS_Walking)
        Pawn.SetPhysics(PHYS_Falling);

```

Gambar 4.2 : *Source class* lalulintas

Selain *Class pathnode* diperlukan juga Pengatur lalu lintas yang bertugas untuk mengatur jalan dari mobil tersebut. *Class* tersebut berfungsi untuk menentukan ke arah manakah mobil akan berjalan. *Pawn* sendiri berfungsi untuk mengaitkan antara satu *pathnode* dengan *pathnode* lainnya. *Class* ini juga mendefinisikan bagaimana mobil akan berjalan, Terbang atau berjalan seperti biasa.

```

if (Jarakkita < AwasTabraan)
{
    `log("MobilJalan PlayerTerlihat CLOSE");
}
if (Jarakkita > AwasTabraan)
{
    `log("MobilJalan PlayerTerlihat FAR");
}

```

Gambar 4.3 : *Source* definisi jarak

Ketika mobil dalam keadaan berjalan maka secara otomatis mobil akan mendeteksi apasaja yang ada didepannya. Jika ada sesuatu penghalang maka

mobil akan berusaha menghindari dari halangan tersebut. Hal itu didefinisikan dengan *variable* Jarakkita dan AwasTabrakan.

```

    if (Pawn.ReachedDestination(Tujuan))
    {
        TujuanTerahir = Tujuan;
    }
    Tujuan = FindRandomDest();

    if (Tujuan != TujuanTerahir)
    {
        Tujuan = FindRandomDest();
        `log("MobilJalan Picked Same Again");
    }
    if (!ActorReachable(Tujuan))
    {
        `log("MobilJalan not reachable");
    }
    if (ActorReachable(Tujuan))
    {
        `log("MobilJalan found a reachable PATH");
    }
}

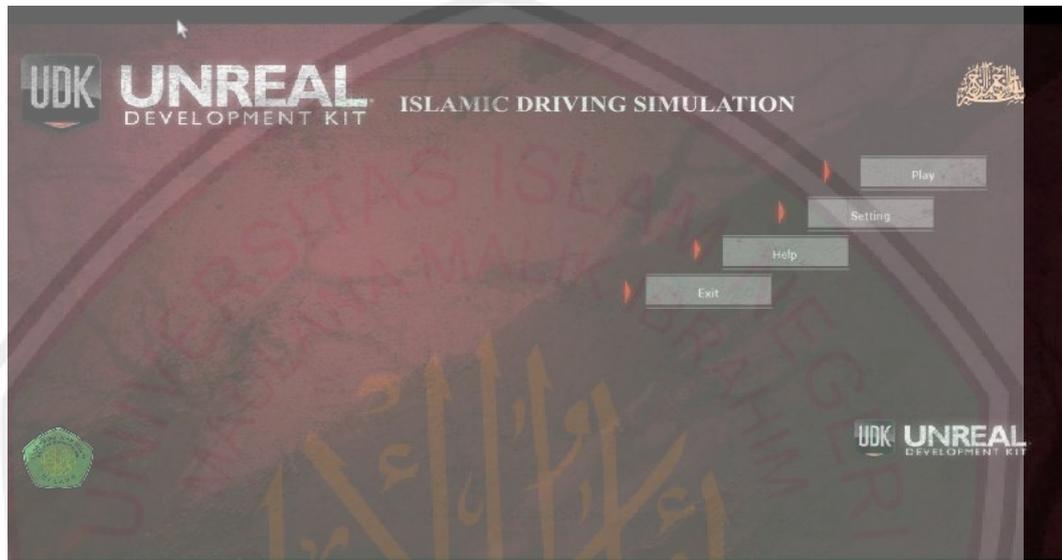
```

Gambar 4.4 : *Source Destinasi Pathnode*

Jika tujuan tidak tercapai maka mobil akan mencari tujuan lain yang terdekat secara random. Hal ini didefinisikan dengan `if (Tujuan == None){Tujuan = FindRandomDest();}` maka secara acak mobil akan mencari target *pathnode*, namun tetap pada satu jalur dengan menggunakan *onewaypath*. Namun sebelum kita memasang metode tersebut, tentunya kita perlu mengatur mobil yang akan digunakan dalam *game* tersebut agar bisa berjalan di dalam *pathfinding*. Dan kode diatas merupakan kode dasar yang harus ada dalam setiap mobil yang digunakan.

4.2 Implementasi Aplikasi

Berikut merupakan Implementasi dari Simulasi Mengemudi.



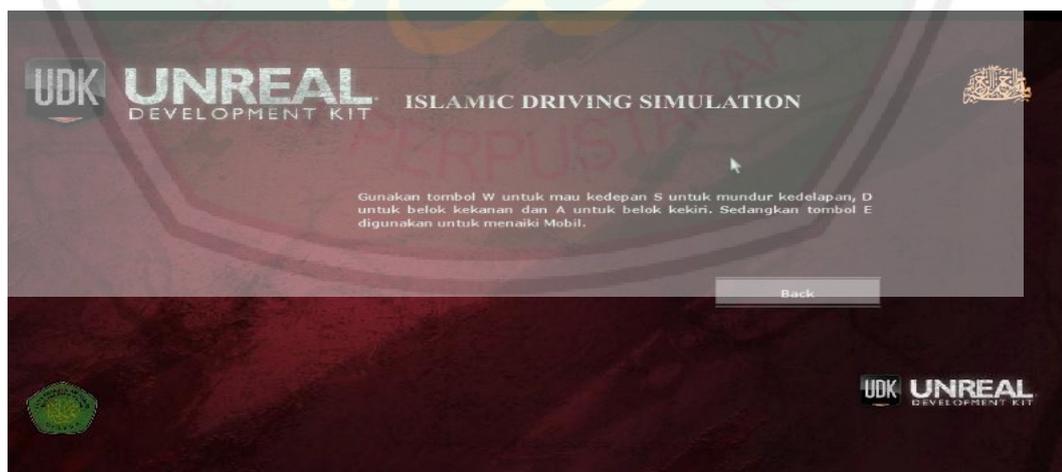
Gambar 4.5 : Halaman Menu pada Game

Gambar diatas merupakan tampilan awal dari *game* simulasi mengemudi, menu diatas memiliki empat pilihan, yakni *play* untuk memainkan *game*, *setting* untuk mengatur resolusi *game*, *help* untuk bantuan dan *exit* untuk mengahiri *game*. Sebelum masuk kedalam menu ini pemain juga akan melihat *splash screen* default udk. *Splash screen* ini ter *setting default* karena versi yang dipakai untuk membuat *game* ini adalah versi *freeware*.



Gambar 4.6 : Halaman Setting

Halaman *Setting* digunakan untuk mengatur *resolusi game* yang akan dijalankan, disini hanya mengatur 2 jenis resolusi yakni 800x600 dan 1280x800 karena *game* yang dijalankan akan bagus pada resolusi tersebut.



Gambar 4.7 : Halaman Bantuan

Halaman ini digunakan untuk membantu pengguna untuk memahami bagaimana cara untuk menjalankan mobil yang ada didalam *game* tersebut. Sehingga pengguna dapat langsung memainkan *game* tersebut.

Ketika pemain masuk kedalam *game* maka secara otomatis ditampilkan cerita dengan memperkenalkan karakter ahmad yang akan membimbing pemain untuk melakukan pekerjaan yang sesuai dengan jalan cerita yang telah didefinisikan.



Gambar 4.8 :Perkenalan Pemandu karakter

Karakter tersebut akan memerintahkan kepada *player* untuk mencari petunjuk yang akan menampilkan cerita berikutnya. Petunjuk tersebut berada pada sekitar rumah *player* itu sendiri Sehingga lebih mudah dicari. Pada saat itulah, sekitar 10 detik setelah *game* dimainkan akan terdengar suara adzan, adzan juga

akan berkumandang selama lima kali sesuai dengan jumlah sholat. Dalam game ini 24 jam sama dengan satu jam permainan *game*.



Gambar 4.9 : Penunjuk Arah *Quest*.

Pada saat *player* menemukan dan mendekati petunjuk yang ditampikan dengan petunjuk warna kuning, maka *player* akan menerima misi secara *parallel*. Oleh karena itu *player* harus menyelesaikan misi tersebut secara bertahap.



Gambar 4.10 : Perintah setelah *player* menyentuh petunjuk arah

Setelah *player* telah mendapatkan misi maka *player* telah siap untuk mengendarai kendaraan yang telah disediakan oleh *system game* tersebut. Kendaraan tersebut berupa mobil dengan *sterring* yang kurang baik, hal itu dikarenakan *player* masih dalam level awal. Mobil yang lebih bagus dapat didapatkan apabila *player* sudah menyelesaikan misi pengenalan.



Gambar 4.11 : HUD akan ditampilkan apabila *player* berada dalam mobil.

Ketika *player* telah mengunakan mobil yang telah disediakan maka dalam tampilan layar akan menampilkan *spedoometer* guna mengingatkan *player* agar tidak terlalu cepat, karena dengan sengaja *handling* pada mobil awal dibuat lebih sensitif, Sehingga *player* akan muda kehilangan kendali saat mengemudi terlalu cepat. Hal itu dilakukan untuk mengurangi kecendrungan *player* untuk mamacu

mobilnya dengan cepat, Sehingga melenceng dari tujuan *game* ini dibuat. Selain itu misi-misi *game* ini akan terlalu cepat berakhir jika *player* terlalu cepat memacu mobilnya.



Gambar 4.12 : Perintah setelah *player* menyentuh penunjuk arah.

Jika *player* telah memasuki kawasan kampus uin malang, maka *player* akan bertemu navigasi yang terpasang di beberapa tempat guna menunjukan kemana *player* boleh melewati jalan tersebut. Dan *player* harus hati hati karena mobil lain dikendalikan oleh AI. Yang akan kacau apabila tersengol. Namun dalam definisi AI tersebut sudah ada sistem menghindar.



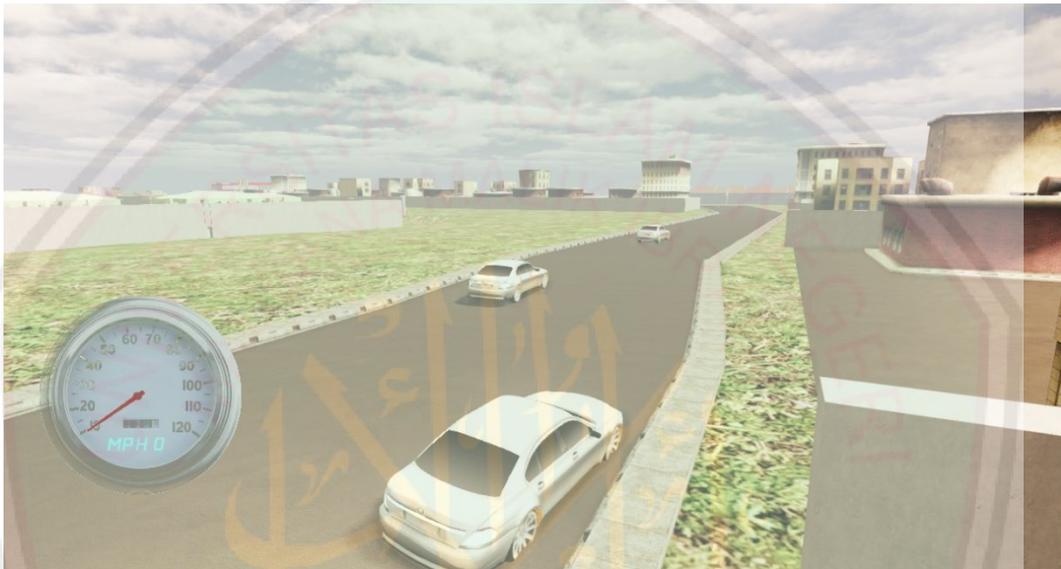
Gambar 4.13 : Perintah setelah *player* menyentuh penunjuk arah.

Sistem akan menyuruh *player* untuk sholat karena pada saat itu sudah selesai azan nduhur, namun ketika sholat *video* yang telah terpasang tidak bisa dimainkan karena *engine* yang digunakan adalah versi *freeware*, jika di *running* dengan *engine* terlisensi fungsi sholat tersebut baru akan dijalankan.



Gambar 4.14 : Gambar *player* telah menyelesaikan misi dan mendapat mobil.

Jika *player* telah selesai dalam tugas pengenalan maka *player* akan mendapatkan *reward* yakni mobil baru yang bisa digunakan langsung oleh *player* tersebut.



Gambar 4.15 : *traffic System* menggunakan *Pathfinding*.

Game ini juga menggunakan *traffic system* menggunakan metode *pathfinding*, dimana mobil akan mengikuti rute *pathfinding* yang sudah didefinisikan oleh *system*. *Pathfinding* yang digunakan adalah *pathfinding onewaypath*, dimana mobil hanya dapat melalui 1 jalur yang telah ditentukan.

4.3 Pengujian Aplikasi

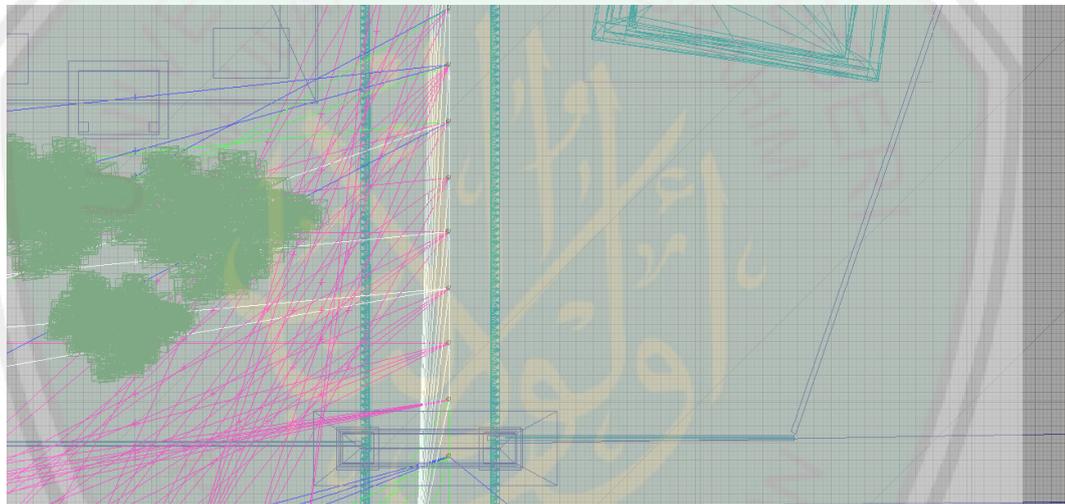
4.3.1 Pengujian *Pathfinding* terhadap *Game*.

Pada tahap ini telah dilakukan pengujian metode dengan berbagai pengaturan luas area pathnode, dimana *pathnode* tersebut diberi nilai 200, 500, dan 1000. Hal tersebut digunakan untuk mengetahui akurasi dari metode ini dalam menjalankan mobil sesuai dengan lintasan, dengan catatan jarak sebenarnya antar *pathnode* adalah 150. Pada tahap pertama digunakan nilai 200. Ternyata dari percobaan tersebut menghasilkan akurasi 75 % pada bidang X dan 100% pada bidang Y, dengan perhitungan matematis $150/200 \times 100\% = 75\%$. Dalam percobaan diatas mobil berjalan sangat lancar dan mengikuti arah path finding. Dan tidak terjadi kesalahan sama sekali. Ketika mobil dengan sengaja ditabrak dengan mobil lain secara reflek mobil akan segera kembali ke jalurnya.



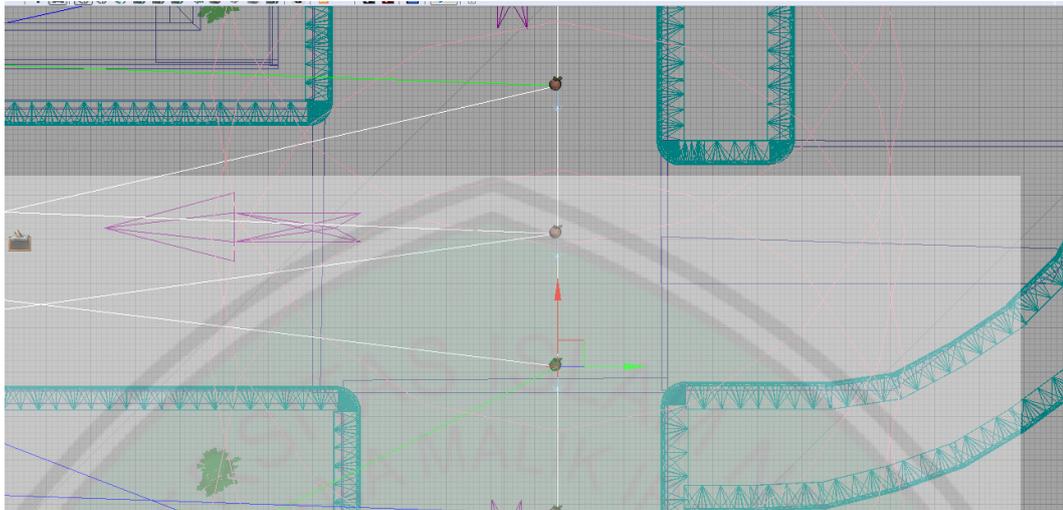
Gambar 4.16 : percobaan 1 *Pathfinding* dengan menggunakan luas jangkauan 200.

Pada percobaan kedua *pathfinding* telah di setting dengan jarak jangkauan 500. Dengan ketentuan letak *pathfinding* berjarak 150 satu sama lain. Pada percobaan tersebut menghasilkan akurasi 30% hal itu karena jangkauan *pathnode* terlalu luas, dan menyebabkan *pathfinding* mendapatkan target tujuan lebih dari satu. Ketika *pathfinding* mendapatkan target lebih dari satu maka *pathfinding* akan berjalan random ke segala arah.



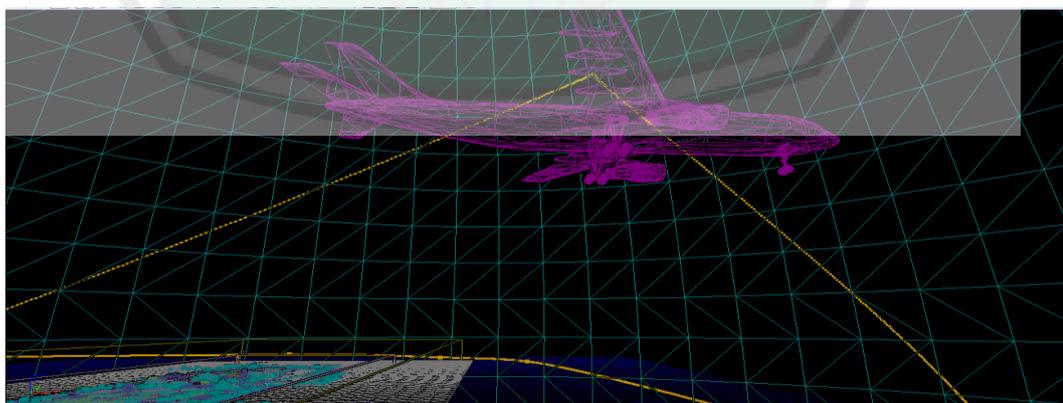
Gambar 4.17 : percobaan 2 *Pathfinding* dengan menggunakan luas jangkauan 500.

Pada percobaan ketiga *pathfinding* telah diatur pada nilai 1000, maka akurasi kurang dari 10% dan mobil akan berjalan dengan random dan terbalik karena tidak bisa mengendalikan sistem navigasinya. Hal tersebut sangat tidak efisien dan menyebabkan komputer yang digunakan untuk percobaan mengalami kelebihan beban. Luasnya perpotongan diatas dapat terlihat dalam gambar dibawah ini.



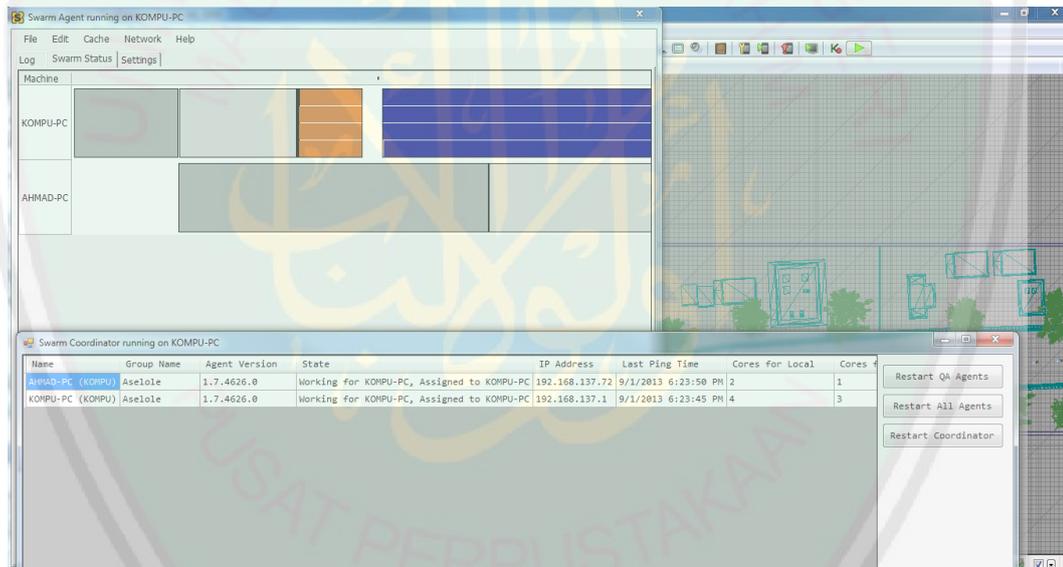
Gambar 4.18 : percobaan 3 *Pathfinding* dengan menggunakan luas jangkauan 1000.

Selain metode *pathfinding*, game ini juga menggunakan *line detection*. Dimana obyek dapat berjalan sesuai dengan garis yang telah didefinisikan, metode ini mempunyai akurasi 100% namun terdapat kelemahan yang sangat merugikan dalam metode ini, kelemahan tersebut adalah obyek tersebut akan bergerak secara statis, dan kurang baik untuk sebuah simulasi.



Gambar 4.19 : percobaan 2 *Line detection* pada pesawat.

Semua percobaan diatas dilakukan dengan menggunakan beberapa komputer yang disambungkan oleh agen. Hal tersebut dilakukan untuk mengetahui kemampuan komputer – komputer lain dalam menjalankan aplikasi ini. Namun dari semua komputer tersebut terdapat satu komputer yang menjadi remote agen. Adapun spesifikasi remote agen diatas adalah PC spesifikasi: Processor Intel(R) Core(TM) 5 Duo T6600 @ 2.20GHz (2 CPUs) dan Memory 4000 MB RAM,VGA Nvidia 9800M.



Gambar 4.20 : Agen Komputer *Sharing* performa.

Dari gambar diatas sangat terlihat ketimpangan antara Kompu-PC dengan Ahmad-pc, dimana Kompu pc lebih cepat dari pada ahmad pc, karena Kompu pc menggunakan core i5 sedangkan Ahmad-pc menggunakan *dualcore*. Namun ketika menjalankan *game* ini, sama sama bisa menjalankan metode yang telah disisipkan kedalam *system*.

Tabel 4.1 Hasil ujicoba pada perangkat lain

Nomor	Komputer	Windows	Hasil Uji Coba
1	Pentium D/256 Nvidia	Windows 7	Cukup dengan res 800x600
2	Core2Duo/intel	Windows 7	Berat/res 800x600
3	Corei5/nvidia 1gb	Windows 7	Sangat Baik
4	Core2Duo/512 Nvidia	Windows 7	Baik
5	Corei3/ati 512	Windows 7	Sangat baik.

4.3.2 Ujicoba Pengguna

Penguji mengambil 10 orang yang diambil saat mengikuti praktikum di kampus UIN MALANG dan 5 orang saat berada di Lingkungan belajar dengan menggunakan random sampling, yakni memilih secara acak orang yang berada dalam lab pemoraman untuk memberikan pendapat tentang game yang dibuat.

Dari pengujian diatas manghasilkan :

Tabel 4.2 Rekapitulasi Pendapat Responden

Nomor	Angka Pendapat Responden				
	Sangat Setuju (SS)	Setuju (S)	Kurang Setuju (KS)	Tidak Setuju (TS)	Jumlah
1	4	10	1	-	15
2	7	8	-	-	15
3	9	3	3	-	15
4	11	4	-	-	15
5	7	5	3	-	15
6	9	6	-	-	15
7	8	7	-	-	15
8	7	6	2	-	15
9	13	1	1	-	15
10	0	3	12	-	15

Dari tabel diatas menghasilkan sebuah presentase yang menggambarkan ujicoba diatas, berikut ini adalah presentase tersebut:

Tabel 4.3 Rekapitulasi Prosentase Hasil Ujicoba

Nomor	Uraian	Pendapat Responden			
		Sangat Setuju	Setuju	Kurang Setuju	Tidak Setuju
1.	Desain <i>Game</i> Menarik	26.66667	66.66	6.666	-
2.	Menu Mudah Dimengerti	46.66667	53.33	-	-
3.	Mudah Mengenali <i>PopUp</i>	60	20	20	-
4.	Faham Dengan Cerita <i>Game</i>	73.33333	26.66	-	-
5.	Memberikan Manfaat terhadap mahasiswa	46.66667	33.33	20	-
6.	<i>Game</i> tersebut menarik dan menyenangkan.	60	40	-	-
7.	<i>Game</i> tersebut lebih menarik daripada <i>Game</i> 2D	53.33	46.66	-	-
8.	Lingkungan Dalam <i>Game</i> menarik	46.66667	40	13.33333	-

9.	Konten Agama Bermanfaat	86.66667	6.666	6.666667	-
10	Sudah Sepadan Dengan Game Sejenis	0	20	80	-

Berdasarkan tabel diatas dapat diketahui bahwa sebagian besar responden menanggapi positif dalam penggunaan *game* tersebut, hal itu dibuktikan dengan banyaknya presentasi responden yang setuju dengan *game* tersebut.

Responden beranggapan *game* tersebut cukup menarik dalam segi desain menu maupun desain popup yang ditampilkan dalam *game*. Meskipun hanya *game* sederhana namun responden menganggap hal tersebut sudah cukup untuk dianggap menarik, senada dengan desain alur *game* juga mendapat respon positif, hal tersebut dibuktikan dengan didapaknya 73% responden yang merasa mengaerti dengan alur cerita. Alur cerita tersebut dianggap cukup sederhana namun dapat menampilkan tokoh-tokoh yang seolah-olah berbicara dengan diri mereka.

Selain itu tampilan *game* yang serba 3D menjadi suatu hal yang mendapat sambutan yang cukup positif, mereka merasa *game* tersebut seperti keadaan yang sebenarnya. Karena itulah pada poin ini mendapat presentase diatas 50%. Perpaduan 3D dan beberapa konten agama juga dirasa cukup baik dan mendapatkan 86% dari presentase responden.

Namun karena keterbatasan game ini, maka responden juga berpendapat bahwa game ini masih jauh dari game yang sejenis, yakni game-game simulasi yang telah diproduksi oleh developer lainnya. Responden tidak satupun setuju apabila game ini disejajarkan dengan game yang sudah lebih baik.



BAB V

PENUTUP

5.1 Kesimpulan

Dari berbagai penjelasan diatas dapat disimpulkan bahwa aplikasi diatas dapat berjalan cukup baik *desktop platform*, namun masih terdapat beberapa kekurangan yang harus dibenahi. selain itu metode *pathfinding* navigasi terbukti sangat baik digunakan dalam udk. Karena dalam pengujian tidak terdapat masalah yang berarti dari metode ini.

Dari segi manfaat, *game* ini sangat bermanfaat karena secara tidak langsung dapat membentuk karakter penggunanya untuk lebih berusaha untuk tertib dalam berlalulintas. Hal tersebut ditunjukkan dengan mematuhi petunjuk arah yang diatur di kampus uin malang.

5.2 Saran

Tentunya masih banyak kekurangan dalam penelitian *game* simulasi mengemudi ini. Oleh karena itu penulis menyarankan beberapa hal untuk bahan pengembangan selanjutnya, diantaranya:

1. Mengembangkan *game* simulasi ini dengan menambahkan fitur-fitur lain yang lebih optimal.
2. Menambahkan beberapa navigasi dan metode-metode lain yang mendukung terjadinya sebuah simulasi yang menarik.

DAFTAR PUSTAKA

- Pereira, José Luis Ferrás. 2011. *An Integrated Architecture for Autonomous Vehicles Simulation*. FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO.
- Cutilli, Benjamin. 2010. *Computer Vision and its Application in Self-Driving Cars*. Haverford College: USA.
- Assyifa, Nur Safira. 2009. *Pengembangan Peta Interaktif Tiga Dimensi Menggunakan Unreal Engine*. Institut Teknologi Sepuluh Nopember Surabaya.
- Rabin, Steve. 2002. *AI Programming Wisdom*. CHARLES RIVER MEDIA: USA.
- Pangajow, Frank Albert. 2008. *RPG STUDIO Every one can develop Game*. PT ELEX MEDIA KOMPUTINDO: Jakarta.
- Cordone, Rachel. 2011. *Unreal Development Kit Game Programming with UnrealScript*. Packt Publishing: USA.
- Wibowo, Agung Toto. *Jurnal Perkuliahan. Game Programming IT TELKOM*
- Bahreisy, Salim. 1987. *Tafsir Ibnu Katsir jilid 1*. PT BINA ILMU: Surabaya.
- Muhammad, Abu Abdulloh. 2010. *Matan Ajrumiyyah*. Kakilangit: Langitan Tuban
- Chin, Robert. 2012. *Beginning IOS 3D Unreal Games Developmen*. Apress: USA.
- McHaney, Roger. 2009. *Understanding Computer Simulation*. Venus Publishing : USA.
- Tegleg. (2010). *Tegleg Pathnode Tutorial*. [Online]. Tersedia: <http://forums.epicgames.com/threads/804767-Teglegs-Tutorials-Code-and-Stuff> [12 Agustus 2013].

DAFTAR PUSTAKA

- Pereira, José Luis Ferrás. 2011. *An Integrated Architecture for Autonomous Vehicles Simulation*. FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO.
- Cutilli, Benjamin. 2010. *Computer Vision and its Application in Self-Driving Cars*. Haverford College: USA.
- Assyifa, Nur Safira. 2009. *Pengembangan Peta Interaktif Tiga Dimensi Menggunakan Unreal Engine*. Institut Teknologi Sepuluh Nopember Surabaya.
- Rabin, Steve. 2002. *AI Programming Wisdom*. CHARLES RIVER MEDIA: USA.
- Pangajow, Frank Albert. 2008. *RPG STUDIO Every one can develop Game*. PT ELEX MEDIA KOMPUTINDO: Jakarta.
- Cordone, Rachel. 2011. *Unreal Development Kit Game Programming with UnrealScript*. Packt Publishing: USA.
- Wibowo, Agung Toto. *Jurnal Perkuliahan. Game Programming IT TELKOM*
- Bahreisy, Salim. 1987. *Tafsir Ibnu Katsir jilid 1*. PT BINA ILMU: Surabaya.
- Muhammad, Abu Abdulloh. 2010. *Matan Ajrumiyah*. Kakilangit: Langitan Tuban
- Chin, Robert. 2012. *Beginning IOS 3D Unreal Games Developmen*. Apress: USA.
- McHaney, Roger. 2009. *Understanding Computer Simulation*. Venus Publishing : USA.
- Tegleg. (2010). Tegleg Pathnode Tutorial. [Online]. Tersedia: <http://forums.epicgames.com/threads/804767-Teglegs-Tutorials-Code-and-Stuff> [12 Agustus 2013].