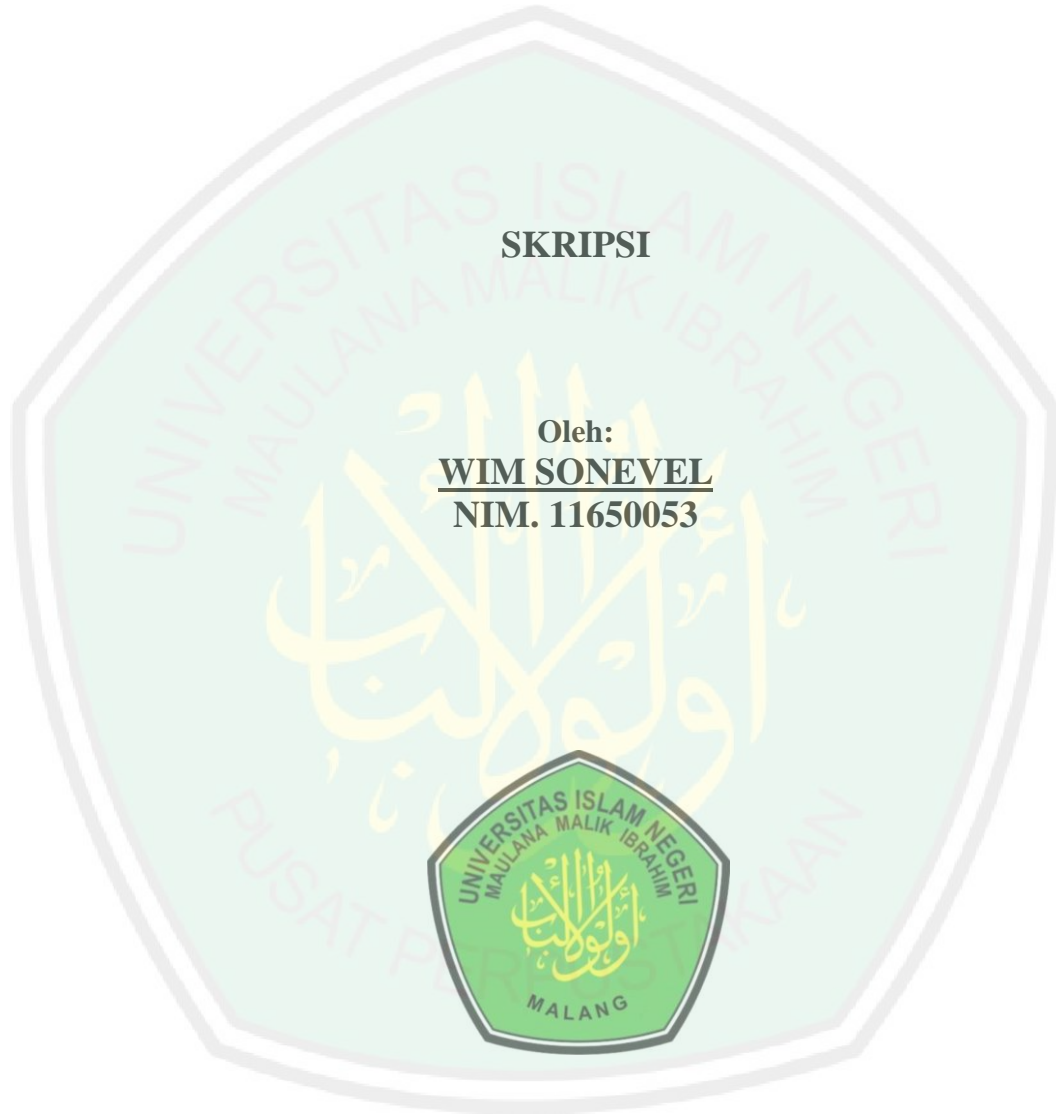


**IMPLEMENTASI FINITE STATE MACHINE DAN LOGIKA FUZZY
UNTUK PERILAKU NON PLAYABLE CHARACTER PADA GAME
KWIKKI JELAJAH NUSANTARA BERBASIS MOBILE
AUGMENTED REALITY**

SKRIPSI

Oleh:

WIM SONEVEL
NIM. 11650053



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2015**

**IMPLEMENTASI FINITE STATE MACHINE DAN LOGIKA
FUZZY UNTUK PERILAKU NON PLAYABLE CHARACTER
PADA GAME KWIKKI JELAJAH NUSANTARA BERBASIS
MOBILE AUGMENTED REALITY**

SKRIPSI

Diajukan kepada:

Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang

Untuk Memenuhi Salah Satu Persyaratan Dalam

Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:

WIM SONEVEL

NIM. 11650053

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2015**

HALAMAN PERSETUJUAN

IMPLEMENTASI FINITE STATE MACHINE DAN LOGIKA FUZZY UNTUK PERILAKU NON PLAYABLE CHARACTER PADA GAME KWIKKI JELAJAH NUSANTARA BERBASIS MOBILE AUGMENTED REALITY

SKRIPSI

Oleh :

Nama : Wim Sonevel
NIM : 11650053
Jurusan : Teknik Informatika
Fakultas : Sains Dan Teknologi

Telah Disetujui, 06 Juli 2015

Dosen Pembimbing I

Dosen Pembimbing II

Hani Nurhayati, M.T

NIP. 19780625 200801 2 006

Fresy Nugroho, M.T

NIP. 19710722 201101 1 001

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian

NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

IMPLEMENTASI FINITE STATE MACHINE DAN LOGIKA FUZZY UNTUK PERILAKU NON PLAYABLE CHARACTER PADA GAME KWIKKI JELAJAH NUSANTARA BERBASIS MOBILE AUGMENTED REALITY

SKRIPSI

Oleh :

Wim Sonevel

NIM. 11650053

Telah Dipertahankan Di Depan Dewan Penguji Skripsi Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal, 06 Juli 2015

Susunan Dewan Penguji :

		Tanda Tangan
1. Penguji Utama	: <u>Yunifa Miftachul Arif, M.T</u> NIP. 19830616 201101 1 004	()
2. Ketua Penguji	: <u>Fachrul Kurniawan, M.MT</u> NIP. 19771020 200901 1 001	()
3. Sekretaris Penguji	: <u>Hani Nurhayati, M.T</u> NIP. 19780625 200801 2 006	()
4. Anggota Penguji	: <u>Fresy Nugroho, M.T</u> NIP. 19710722 201101 1 001	()

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdiان
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertandatangan di bawah ini:

Nama : Wim Sonevel
NIM : 11650053
Fakultas/Jurusan : Sains Dan Teknologi / Teknik Informatika
Judul Penelitian : Implementasi *Finite State Machine* dan Logika
Fuzzy untuk Perilaku *Non Playable Character* pada
Game Kwikki Jelajah Nusantara Berbasis *Mobile*
Augmented Reality

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan 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 di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 06 Juli 2015

Yang Membuat Pernyataan,

Wim Sonevel

11650053

HALAMAN MOTTO

*Selalu berusaha keras, berikhtiar dan berdoa
dalam apa-apa yang kita kerjakan demi masa
depan.*

Karena usaha keras takkan mengkhianati ...

HALAMAN PERSEMBAHAN

Karya ini saya persembahkan :

Kepada Ayah dan Ibu tercinta yang senantiasa mencurahkan kasih sayang,

yang selalu memberikan dukungan, motivasi, nasihat dan doanya,

sehingga saya bisa menyelesaikan skripsi ini tepat waktu.

Kepada keluarga yang juga telah memberikannya baik berupa materi maupun doanya.

Kepada teman-teman khususnya yang telah ikhlas meminjamkan smartphonenya demi kelancaran pengerjaan skripsi ini. Maaf jika selama ini saya banyak merepotkan.

Kepada teman-teman angkatan seperjuangan, adek angkatan maupun kakak angkatan lainnya yang telah menyemangati.

Kepada semuanya yang telah berperan demi kelancaran skripsi ini.

Mungkin terima kasih saja sudah biasa diucapkan, tapi sekali lagi saya ucapkan terima kasih banyak.

*Semoga Allah SWT melimpahkan ridhoNya kepada kita semua.
Amin*

KATA PENGANTAR



Segala puji bagi Allah SWT yang telah melimpahkan rahmat serta karunia-Nya kepada penulis sehingga bisa menyelesaikan skripsi dengan judul “Implementasi *Finite State Machine* dan Logika *Fuzzy* untuk Perilaku *Non Playable Character* pada Game Kwikki Jelajah Nusantara Berbasis *Mobile Augmented Reality*” dengan baik.

Shalawat serta salam semoga tercurah kepada Nabi Agung Muhammad SAW yang telah membimbing umatnya dari gelapnya kekufuran menuju cahaya Islam yang terang benderang.

Penulis menyadari keterbatasan pengetahuan yang penulis miliki, karena itu tanpa keterlibatan dan sumbangsih dari berbagai pihak, sulit bagi penulis untuk menyelesaikan skripsi ini. Maka dari itu dengan segenap kerendahan hati patutlah penulis ucapkan terima kasih kepada:

1. Ayah dan Ibu serta keluarga yang selalu memberikan doa dan motivasinya demi kelancaran dan kemudahan dalam pengerjaan skripsi ini.
2. Hani Nurhayati, M.T, selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan dalam pengerjaan skripsi ini.
3. Fresy Nugroho, M.T, selaku dosen pembimbing II, yang selalu memberikan masukan, nasehat serta petunjuk dalam penyusunan laporan skripsi ini.

4. Dr. Cahyo Crysdiyan, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang mendukung dan mengarahkan dalam pengerjaan skripsi ini.
5. Segenap Dosen Teknik Informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
6. Semua pihak yang tidak mungkin penulis sebutkan satu-persatu, atas segala yang telah diberikan kepada penulis dan dapat menjadi pelajaran.

Sebagai penutup, penulis menyadari dalam skripsi ini masih banyak kekurangan dan jauh dari sempurna. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya. Harapan penulis semoga karya ini bermanfaat bagi kita semua. Amin.

Malang, 06 Juli 2015

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
HALAMAN MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xiv
ABSTRAK	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	6
1.3 Tujuan Penelitian	6
1.4 Batasan Masalah	6
1.5 Manfaat Penelitian	7
BAB II KAJIAN PUSTAKA	8
2.1 Landasan Teori	8
2.1.1 Game	8
2.1.2 Augmented Reality	11
2.1.3 Finite State Machine	12
2.1.4 Logika Fuzzy	16
2.1.5 Android	23
2.1.6 Unity 3D	26

2.1.7	Blender.....	27
2.1.8	Metaio SDK	27
2.2	Penelitian Terkait.....	31
BAB III METODOLOGI PENELITIAN		36
3.1	Desain Penelitian	36
3.2	Prosedur Penelitian	36
3.3	Desain Aplikasi.....	40
3.3.1	Deskripsi Game	40
3.3.2	Storyline	40
3.3.3	Gameplay	41
3.3.4	Skenario Game.....	41
3.3.5	Storyboard	42
3.3.6	Item Permainan	47
3.3.7	Scoring	47
3.3.8	Konten-Konten Pada Aplikasi.....	48
3.4	Rancangan <i>Finite State Machine</i>	50
3.5	Desain Fuzzy	52
3.6	Penerapan 3D Instant Tracking.....	60
BAB IV HASIL DAN PEMBAHASAN		63
4.1.	Implementasi.....	63
4.1.1.	Peralatan yang digunakan.....	63
4.1.2.	Implementasi Finite State Machine	64
4.1.4.	Implementasi Logika Fuzzy	70
4.1.5.	Pengujian Logika Fuzzy	77
4.1.6.	Penerapan 3D Instant Tracking pada Metaio SDK.....	81
4.2.	Hasil Akhir.....	82
4.3.	Integrasi Aplikasi dengan Islam	90

BAB V PENUTUP.....	96
5.1. Kesimpulan	96
5.2. Saran	97

DAFTAR PUSTAKA



DAFTAR GAMBAR

Gambar 2. 1 FSM Klasik	14
Gambar 2. 2 State Diagram FSM.....	15
Gambar 2. 3 Kurva Segitiga.....	22
Gambar 2. 4 Kurva Trapesium.....	23
Gambar 2. 5 Arsitektur Android	26
Gambar 2. 6 Struktur Framework Metaio SDK.....	28
Gambar 2. 7 Sequence Diagram Instant Tracking (Sumber : Metaio Developer Portal).....	31
Gambar 3. 1 Tahap-Tahap Penelitian.....	39
Gambar 3. 2 Tampilan Awal.....	42
Gambar 3. 3 Tampilan Stage Permainan.....	43
Gambar 3. 4 Layar Permainan	44
Gambar 3. 5 Kwikki (Karakter/Tokoh Utama).....	49
Gambar 3. 6 Rabbit	50
Gambar 3. 7 FSM NPC musuh.	50
Gambar 3. 8 <i>State</i> Menyerang pada FSM NPC musuh.....	52
Gambar 3. 9 Desain <i>Fuzzy</i> Sugeno NPC musuh.....	53
Gambar 3. 10 Derajat keanggotaan variabel input jarak player.....	54
Gambar 3. 11 Derajat keanggotaan variabel input kesehatan.....	56
Gambar 3. 12 Keanggotaan variabel output perilaku.....	58
Gambar 3. 13 Flowchart 3D Instant Tracking	61
Gambar 4. 1 Perilaku NPC musuh ketika menyerang jarak jauh.....	79
Gambar 4. 2 Perilaku NPC musuh ketika menyerang jarak dekat.....	80
Gambar 4. 3 Perilaku NPC musuh ketika menjauh.....	80
Gambar 4. 4 Tampilan Menu Awal.....	83
Gambar 4. 5 Tampilan Menu Stage	83
Gambar 4. 6 Tampilan Layar melalui Kamera.....	84
Gambar 4. 7 Tampilan Proses Tracking.....	84
Gambar 4. 8 Tampilan Stage 1.....	85
Gambar 4. 9 Tampilan Stage 2.....	85
Gambar 4. 10 Tampilan Stage 3.....	86
Gambar 4. 11 Tampilan Stage 4.....	86
Gambar 4. 12 Tampilan Stage 5.....	87
Gambar 4. 13 Tampilan Informasi Rumah Gadang.....	88
Gambar 4. 14 Tampilan Informasi Rumah Joglo.....	88
Gambar 4. 15 Tampilan Informasi Rumah Banjar.....	89
Gambar 4. 16 Tampilan Informasi Rumah Honai.....	89
Gambar 4. 17 Tampilan Game Over.....	90

DAFTAR TABEL

Tabel 3. 1 Stage Permainan.....	45
Tabel 3. 2 Input variabel jarak player.	54
Tabel 3. 3 Input variabel kesehatan.....	56
Tabel 3. 4 Input variabel perilaku.	58
Tabel 3. 5 Aturan-aturan fuzzy.	59
Tabel 4. 1 Pengujian perilaku NPC musuh dengan nilai jarak dan kesehatan.....	78



ABSTRAK

Sonevel, Wim. 2015. **Implementasi *Finite State Machine* dan Logika *Fuzzy* untuk Perilaku *Non Playable Character* pada Game “Kwikki Jejalah Nusantara” Berbasis *Mobile Augmented Reality***. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing : (I) Hani Nurhayati, M.T (II) Fresy Nugroho, M.T

Kata Kunci: *Finite State Machine*, *Logika Fuzzy*, *Non Playable Character*

Kecerdasan buatan menjadi sangat penting dalam pengembangan sebuah game, artinya membuat aksi dan reaksi dalam game menjadi lebih realistis khususnya pada agen atau *Non Playable Character* (NPC). Penelitian ini menjelaskan bagaimana merancang perilaku NPC pada sebuah game agar menghasilkan respon yang bervariasi. Game “Kwikki Jelajah Nusantara” adalah sebuah aplikasi permainan berbasis *Augmented Reality* yang mengharuskan untuk bertahan hidup dan mengalahkan semua musuh agar dapat menyelesaikan misi. Musuh merupakan NPC yang dikontrol dengan kecerdasan buatan yang akan dirancang sesuai dengan responnya terhadap pemain. Pada penelitian ini, metode kecerdasan buatan yang digunakan adalah *Finite State Machine* dan logika *fuzzy*. *Finite State Machine* diterapkan untuk mengatur perilaku dari *Non Playable Character* (NPC). Sedangkan logika *fuzzy* digunakan untuk menghasilkan respon yang bervariasi pada aksi menyerang. Dalam pengujian metode *Finite State Machine* berjalan dengan baik dalam mengontrol perilaku. Sedangkan logika *fuzzy* dapat menghasilkan respon yang bervariasi dengan presentase error yang diperoleh sebesar 4,4 %.

ABSTRACT

Sonevel, Wim. 2015. **Implementation of Finite State Machine and Fuzzy Logic for Behaviour of Non Playable Character in the Game "Kwikki Jejalah Nusantara" Based on Mobile Augmented Reality.**

Advisor : (I) Hani Nurhayati, M.T (II) Fresy Nugroho, M.T

Keyword : *Finite State Machine, Fuzzy Logic, Non Playable Character*

Artificial intelligence is very important in development of game. that means is making action and reaction in the game becomes more realistic, especially on the agent or Non Playable Character (NPC). This paper describes how to design a behavior of NPC in a game in order to produce varies responses. Game "Kwikki Jelajah Nusantara" is an Augmented Reality Games which require to survive and defeat all the enemies in order to complete the mission. An enemy NPC controlled by artificial intelligence that will be designed according to its response to the player. An artificial intelligence method that used is Finite State Machine and fuzzy logic. Finite State Machine implemented to regulate the behavior of Non-Playable Character (NPC). While fuzzy logic is used to generate varied responses on attacking action. In the test of methods Finite State Machine runs well in controlling behavior. Meanwhile, fuzzy logic can produce responses that vary with the percentage error of 4.4% was obtained.

مستخلص البحث

ويم سونيفيل ، 2015 ، تنفيذ محدود آلة الدولة ومنطق ضبابي على اجراء الحرف غير القابل للتشغيل في لعبة " كويكك المبحرة الأرخبيل " استنادا موبايل مزيدة واقع

المستشار : (1) هاني نور البيولوجية الماجستير (2) فاريش نوغروهو الماجستير

الكلمات المفتاحية : آلة الدولة المحدودة ، المنطق الضبابي ، الحرف غير القابل للتشغيل

الذكاء الاصطناعي مهم جدا في تطوير لعبة ، يعني جعل الفعل ورد الفعل في اللعبة يصبح أكثر واقعية ولا سيما في الحرف غير القابل للتشغيل . وتصف هذه الورقة كيفية تصميم السلوك الحرف غير القابل للتشغيل في لعبة من أجل إنتاج الاستجابات التي تختلف . لعبة " كويكك المبحرة الأرخبيل " القائم على لعبة مزيدة واقع الأمر الذي يتطلب البقاء على قيد الحياة ، وهزيمة كل الأعداء من أجل اكمال المهمة. العدو الحرف غير القابل للتشغيل التي يسيطر عليها الذكاء الاصطناعي التي سيتم تصميمها وفقا للاستجابة لالعاب . في هذه الدراسة ، طريقة الذكاء الاصطناعي المستخدم هو آلة الدولة المحدودة والمنطق الضبابي . آلة الدولة المحدودة تطبيق لتنظيم سلوك الحرف غير القابل للتشغيل . في حين يتم استخدام المنطق الضبابي ل تنتج الاستجابات التي تختلف في الهجوم . في وطريقة الاختبار آلة الدولة المحدودة يعمل بشكل جيد في السيطرة على السلوك . بينما المنطق الضبابي يمكن أن تنتج استجابات مختلفة مع نسبة الخطأ التي تم الحصول عليها في % 4,4

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di zaman modern sekarang ini, globalisasi menjadi tantangan bagi aspek kehidupan yang terkait dengan kebudayaan. Budaya tradisional yang sejak dulu sudah melekat pada masyarakat tidak mampu bertahan dalam era global. Era globalisasi menuntut kesiapan kita untuk berubah sesuai dengan perkembangan zaman. Kebudayaan tradisional lama kelamaan mulai mengalami erosi. Orang-orang terutama anak muda lebih suka menghabiskan waktunya untuk bermain gadget, mengakses internet dari pada mempelajari kultur budaya daerah sendiri.

Kebudayaan adalah kompleks yang mencakup pengetahuan, kepercayaan, kesenian, moral, hukum, adat istiadat dan kemampuan lain serta kebiasaan yang didapatkan oleh manusia sebagai anggota masyarakat. Budaya sangat erat kaitannya dengan kehidupan masyarakat di sebuah negara, terlebih bagi negara yang dikenal sebagai negara multikultural seperti Indonesia. Segala aspek dalam kehidupan masyarakat dapat dikatakan sebagai wujud dari kebudayaan, misalnya gagasan atau pikiran, aktivitas, maupun karya yang dihasilkan.

Budaya sebenarnya adalah suatu identitas bangsa yang harus dijaga dan dihormati oleh para penerus bangsa. Di negara tercinta kita Indonesia memiliki keberanekaragaman budaya lokal yang terdiri dari banyak suku, bahasa, dan kesenian. Dengan potensi yang dimiliki Indonesia sebagai negara multikultural dapat mendefinisikan apa yang disebut kebudayaan bangsa, seperti yang terdapat

pada penjelasan Pasal 32 UUD 1945, yang berbunyi: “Kebudayaan bangsa (Indonesia) adalah puncak-puncak kebudayaan di daerah”. Keanekaragaman budaya lokal yang di miliki Indonesia menjadi sebuah aset berharga dan tak ternilai harganya.

Budaya daerah lambat laun mulai hilang dikarenakan masuknya budaya asing. Oleh karena itu kita sebagai warga negara Indonesia seharusnya melestarikannya apa yang menjadi milik kita yang diwariskan oleh nenek moyang kita. Dahulu, budaya daerah di negara begitu banyak ragamnya, mulai dari tarian daerah, bahasa daerah, alat musik tradisional, rumah adat dan masih banyak lagi. Tetapi sekarang budaya daerah di Indonesia mulai sedikit dan terancam punah. Banyak kasus-kasus yang terjadi terkait masalah budaya. Salah satunya adalah rumah adat tradisional yang terancam punah disebabkan pengaruh globalisasi. Mulai dari Rumah Joglo, Rumah Kudus, Rumah Mamasa, Rumah Kerinci dan lainnya. Sejumlah alat musik tradisional dari bambu juga mulai terancam punah. Selain itu juga pakaian adat sudah mulai terlupakan akibat sudah jarang dipakai.

Ajaran Islam mendorong umatnya agar mengerahkan segala daya dan upaya untuk kebaikan dan kesejahteraan umat manusia, termasuk dalam pengembangan kebudayaan. Upaya-upaya tersebut kemudian telah menghasilkan suatu prestasi peradaban baru yang tinggi yang dikenal dengan “peradaban Islam” yang dalam sejarahnya telah memberikan andil yang cukup besar bagi kemajuan peradaban dunia. Islam menempatkan akal fikiran dalam posisi yang tinggi, seperti yang tercantum dalam surat Ali Imran ayat 190-191 :

إِنَّ فِي خَلْقِ السَّمَوَاتِ وَالْأَرْضِ وَأَخْتِلَافِ اللَّيْلِ وَالنَّهَارِ لَآيَاتٍ لِّأُولِي الْأَلْبَابِ ۝ ١٩٠ الَّذِينَ
يَذْكُرُونَ اللَّهَ قِيَمًا وَقُعُودًا وَعَلَىٰ جُنُوبِهِمْ وَيَتَفَكَّرُونَ فِي خَلْقِ السَّمَوَاتِ وَالْأَرْضِ
رَبَّنَا مَا خَلَقْتَ هَذَا بَطْلًا سُبْحَانَكَ فَقِنَا عَذَابَ النَّارِ ۝ ١٩١

Artinya: Sesungguhnya dalam penciptaan langit dan bumi, dan silih bergantinya malam dan siang terdapat tanda-tanda bagi orang-orang yang berakal, (yaitu) orang-orang yang mengingat Allah sambil berdiri atau duduk atau dalam keadaan berbaring dan mereka memikirkan tentang penciptaan langit dan bumi (seraya berkata): "Ya Tuhan kami, tiadalah Engkau menciptakan ini dengan sia-sia. Maha Suci Engkau, maka peliharalah kami dari siksa neraka (Q.S.Ali Imran:190,191).

Menurut Tafsir Jalalain Jilid 1 halaman 308-309, ayat di atas dapat ditafsirkan : “Sesungguhnya di dalam penciptaan langit dan bumi” berikut keajaiban-keajaiban yang ada di dalamnya “dan silih bergantinya malam dan siang” dengan datang dan pergi, bertambah dan berkurang “benar-benar terdapat tanda-tanda” bukti-bukti yang menunjukkan kekuasaan Allah “bagi orang-orang yang berakal”. Yakni orang-orang yang memiliki akal sehat.

Perkembangan teknologi saat ini semakin berkembang pesat dan telah memberikan dampak yang signifikan terhadap kehidupan masyarakat. Salah satu teknologi yang memiliki peran yang sangat penting adalah teknologi multimedia. Multimedia memiliki banyak manfaat bagi kehidupan masyarakat di berbagai bidang. Contohnya pemanfaatannya adalah di bidang media hiburan maupun

media edukasi. Salah satu bentuk multimedia yang digunakan adalah berupa *game* dengan berbagai konten di dalamnya. Bermain *game* pada umumnya bertujuan hanya semata-mata sebagai sarana hiburan.

Game pada umumnya menyajikan konten-konten menarik yang bersifat menghibur, namun ada juga menyajikan konten-konten edukasi. Namun, *game* yang menyajikan tentang konten budaya lokal masih sangat jarang ditemui. Padahal, konten-konten tersebutlah yang membuat perbedaan dari jenis *game* lainnya.

Semakin berkembangnya teknologi di bidang komputer grafis saat ini maka muncullah teknologi baru yang disebut *Augmented Reality* (AR). Teknologi ini menggabungkan benda dua dimensi maupun tiga dimensi yang ada di dunia maya ke dunia nyata secara *realtime*. Dapat disebut bahwa teknologi *augmented reality* merupakan gambaran dunia nyata diintegrasikan bersama materi-materi virtual, seperti gambar, suara, video, bahkan lokasi. Hal ini menjadikan teknologi ini sebagai teknologi kreatif yang cukup diminati saat ini. Penggunaan teknologi *augmented reality* telah merambah ke berbagai bidang diantaranya industri, marketing, otomotif, sales, edukasi, *game*, *entertainment* dan sebagainya. Salah satu penggunaannya adalah pada *game*.

Teknologi *augmented reality* saat ini telah banyak diimplementasi dalam berbagai platform, contohnya pada perangkat *smartphone*. *Augmented Reality* dapat diimplementasikan berupa sebuah *mobile game*. Dengan demikian, *mobile game* yang berbasis *augmented reality* dapat memvisualisasi objek-objek dalam *game* menjadi nyata, variatif dan menarik. Namun penggunaannya masih

tergolong minim. Sedangkan penggunaan *smartphone* saat ini sudah mencapai angka yang cukup tinggi. Seperti dikutip pada artikel di website SWA Media Grup yaitu menurut Anil Anthony, Executive Director Consumer Insight Nielsen, angka pengguna *smartphone* di Indonesia akan terus meningkat dan diproyeksikan tahun ini akan menembus angka 28-30%. “Saat ini pengguna *smartphone* di Indonesia baru mencapai 23% dari total pengguna ponsel. Selain itu, menurut survey yang dilakukan oleh Nielsen On Device Meter (ODM) pada february tahun 2014 lalu mencatat bahwa pengguna *smartphone* di Indonesia menghabiskan waktu rata-rata 140 menit per hari untuk menggunakan *smartphone* mereka. Alokasi waktu penggunaan *smartphone* untuk bermain *game* adalah 17 menit per hari. Oleh karena itu, *game* mempunyai potensi yang besar untuk memperkaya konten multimedia menjadi lebih interaktif. Apalagi dengan *game* yang termuat konten edukasi di dalamnya.

Dari permasalahan tersebut maka untuk tujuan dari penelitian ini adalah mengangkat tema permasalahan mengenai kebudayaan lokal yang hampir punah dan menyajikannya dalam bentuk *game* dengan konten kebudayaan nusantara. Selain itu juga dapat mengenalkan kepada dunia bahwa Indonesia sebenarnya memiliki keanekaragaman luar biasa dan tidak ternilai harganya. Walaupun dengan kemajuan teknologi yang begitu pesat, peneliti berharap agar budaya-budaya lokal di Indonesia tidak tergeser oleh budaya asing. Pada penelitian ini, peneliti membuat sebuah *game* atau permainan berbasis *Augmented Reality* (AR) yang menyajikan konten-konten budaya lokal. *Game* ini bernama “Kwikki Jelajah Nusantara” yang menceritakan petualangan Si Kwikki dalam menggali informasi

tentang rumah adat tradisional yang ada di suatu daerah. Metode *Finite State Machine* dan Logika *Fuzzy* adalah metode kecerdasan buatan pada game ini yang diletakkan pada *Non Playable Character* (NPC) agar respon yang berbeda.

1.2 Identifikasi Masalah

1. Bagaimana cara mengimplementasikan *Finite State Machine* dan logika *Fuzzy* untuk perubahan perilaku dan pengambilan keputusan oleh NPC (*Non Playable Character*) ?
2. Bagaimana membuat *game* berbasis *Augmented Reality* berisi konten budaya lokal di Indonesia ?

1.3 Tujuan Penelitian

1. Mengimplementasikan *Finite State Machine* dan logika *Fuzzy* untuk perubahan perilaku dan pengambilan keputusan oleh NPC (*Non Playable Character*).
2. Membuat *game* berbasis *Augmented Reality* berisi konten budaya lokal di Indonesia.

1.4 Batasan Masalah

1. *Finite State Machine* dan logika *fuzzy* diimplementasikan pada NPC musuh
2. *Tracking* dalam *Augmented Reality* menggunakan mode *3D Instant Tracking* yang sudah ada di Metaio SDK.

3. Permukaan yang *di-tracking* berdasarkan tingkat kekontrasan dan permukaan benda.
4. Konten budaya lokal berupa rumah adat.

1.5 Manfaat Penelitian

1. Manfaat pada *game* adalah agar menghasilkan perilaku NPC musuh yang bervariasi.
2. Manfaat dari penelitian ini adalah untuk mengenalkan budaya lokal di Indonesia melalui *game* agar dapat menambah wawasan kita tentang keanekaragaman budaya lokal yang harus kita jaga dan lestarikan sebagai penerus bangsa.

BAB II

KAJIAN PUSTAKA

2.1 Landasan Teori

2.1.1 Game

Game berasal dari bahasa Inggris yang artinya permainan. Definisi *game* dalam kamus *Cambridge Learners Dictionary* adalah “*an entertaining activity or sport that people play, usually needing some skill and played according to rules*”, yang berarti aktifitas hiburan atau olahraga yang dimainkan oleh orang, biasanya membutuhkan keterampilan tertentu dan dimainkan sesuai aturan. Permainan merupakan suatu aktifitas atau kegiatan dengan tujuan hiburan. Permainan memiliki aturan tertentu dalam bermain yang dapat membangun situasi bersaing antara dua atau beberapa orang(kelompok). Aturan-aturan dalam game menentukan tindakan dari pemain maupun lawan. Berikut adalah beberapa pengertian *game* menurut para ahli antara lain :

1. Menurut Chris Crawford, seorang *computer game designer* mengemukakan bahwa *game* merupakan sebuah interaktif, aktivitas yang berpusat pada sebuah pencapaian, ada perilaku aktif dan ada lawan Anda.
2. Menurut Ivan C. Sibero, *game* merupakan aplikasi yang paling banyak digunakan dan dinikmati oleh pengguna media elektronik saat ini.

3. Menurut Samuel Henry, *game* merupakan bagian yang tidak dapat terpisahkan dari keseharian anak-anak.
4. Menurut John Naisbitt, *game* merupakan sistem partisipatoris dinamis karena *game* memiliki tingkat penceritaan yang tidak dimiliki film.
5. Menurut Gred Costikyan, *game* merupakan sebuat karya seni dimana peserta yang disebut sebagai pemain, membuat keputusan untuk mengelola sumber daya yang dimilikinya melalui benda di dalam *game* demi mencapai tujuan.

Computer game merupakan *game* yang dimainkan di komputer. Dalam kamus *Cambridge Learners Dictionary*, definisi *computer game* adalah “*a game that is played on a computer, in which the pictures on the screen are controlled by pressing keys*” yang berarti permainan yang dimainkan di komputer, di mana gambar pada layar komputer dikendalikan dengan menekan tombol. Dalam setiap *game* terdapat peraturan yang berbeda - beda dalam memainkannya sehingga membuat jenis *game* semakin bervariasi. Jenis-jenis *game* antara lain (Lindsay Grace, 2005) :

1. Action

Jenis *game* ini menawarkan aksi atau tindakan sebagai daya tarik utama. Pemain harus mempunyai kemampuan respon reflex yang baik sebagai keterampilan utama yang dibutuhkan untuk bermain *game* ini. Contoh dari *game* ini : Doom dan Metal Gear.

2. Adventure

Jenis *game* ini menawarkan eksplorasi dan memecahkan teka-teki sebagai daya tarik utama. Kemampuan penalaran, kreativitas dan rasa ingin tahu sangat diperlukan dari pemain. Contoh game jenis ini : *Myst* dan *Syberia*.

3. Puzzle

Jenis *game* ini menawarkan teka-teki sebagai daya tarik utama. *Game-game* yang sukses dari jenis ini diantaranya *Tetris*, *Lemmings* dan *Minesweeper*.

4. Role Playing

Jenis *game* ini menawarkan pemain kesempatan untuk membenamkan diri dalam situasi karakter pemain. *Role Playing Game* (RPG) membuat sebuah cerita dalam *game* dengan cara-cara yang inovatif dan bervariasi dalam menjalankan cerita. Karakter dapat berubah dan berkembang ke arah yang diinginkan pemain misalnya menjadi semakin hebat, semakin kuat, dan semakin berpengaruh, permainan yang panjang serta dapat manajemen karakter. Contoh dari *game* ini diantaranya : *Baldour's Gate*, *Fable*, *Might and Magic*, *Neverwinter Nights*, *Ultima*, and *World of Warcraft*.

5. Simulation

Unsur utama permainan simulasi adalah kemampuannya untuk menyesuaikan situasi dunia nyata. Simulasi berusaha untuk memberikan kenikmatan melalui pemeragaan. Simulasi tempur dan simulasi mobil balap relatif populer dalam jenis game ini. Simulasi juga dapat mencakup simulasi situasi sosial seperti pada game Sims dan Leisure Suit Larry.

6. Strategy

Permainan Strategi menawarkan hiburan melalui penalaran dan pemecahan masalah. Permainan strategi tidak memerlukan banyak cerita, meskipun permainan yang lebih baru sangat bergantung pada kualitas narasi. Contoh dari jenis *game* ini salah satunya adalah Command dan Conquer.

2.1.2 Augmented Reality

Augmented Reality adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata.

Ronald T. Azuma (1997) mendefinisikan *Augmented Reality* sebagai penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata, dan terdapat integrasi antarbenda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata.

Augmented Reality berbeda dengan *Virtual Reality*. *Virtual reality* membawa pengguna dalam dunia yang dihasilkan komputer sedangkan *Augmented Reality* menggabungkan dunia nyata dengan grafika komputer. Oleh sebab itu, *Augmented Reality* membawa dunia komputer ke dalam dunia kita. Tidak seperti *virtual reality* yang membutuhkan peralatan khusus yang harus disediakan, AR hanya membutuhkan cara untuk menangkap dunia di sekitar kita, biasanya dengan overlay grafika komputer dari kamera. AR didefinisikan sebagai teknologi yang adalah sebagai berikut (Lester Madden, 2011) :

- Menggabungkan dunia nyata dengan grafika komputer
- Menyediakan interaksi dengan benda-benda di sekitar secara real-time
- *Tracking* objek secara *real-time*
- Menyediakan pengenalan citra atau benda
- Menyediakan konteks *real-time* atau data

2.1.3 Finite State Machine

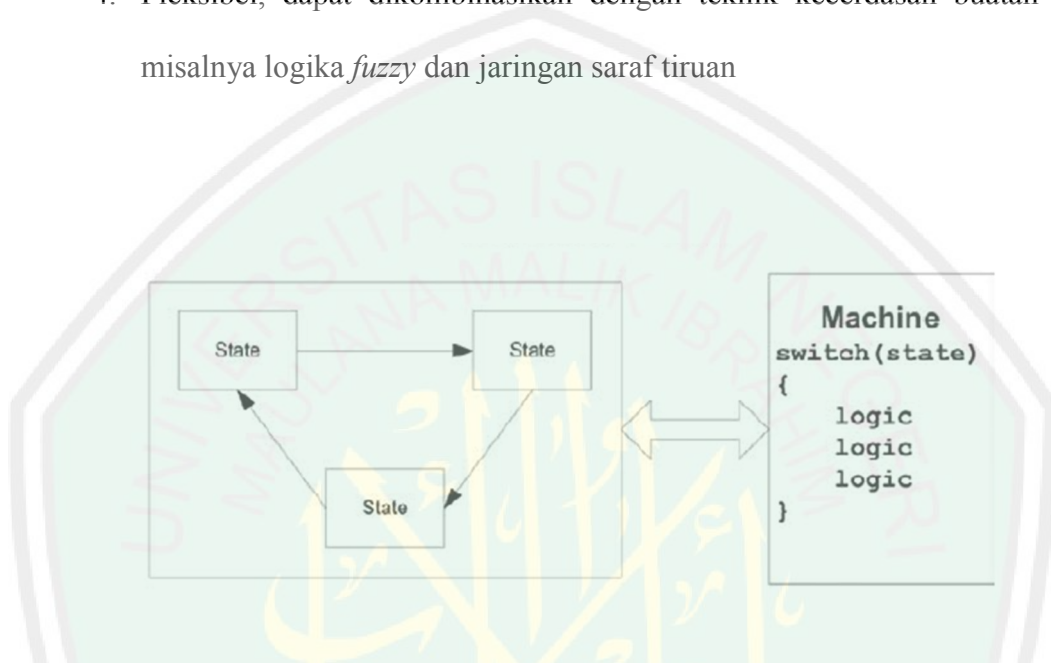
Dalam bidang *Artificial Intelligence* (AI) suatu *state machine* yang terdiri dari bermacam-macam struktur yang disebut *Finite State Machine* (Ian Millington, 2006). Pada intinya, sebuah *State Machine* adalah struktur data yang memodelkan perilaku suatu sistem. *Finite State Machines* (FSM) merupakan sebuah perancangan sistem kontrol yang memodelkan perilaku atau prinsip kerja dari sebuah sistem yang terdiri dari tiga hal yaitu *State* (*Keadaan*), *Event* (*kejadian*) dan *Action* (*aksi*) (Brian Schwab, 2009).

FSM adalah struktur data yang alami. FSM diinisialisasi terlebih dahulu dengan menyatakan semua *state* yang ada, kemudian menyatakan transisi masing-masing *state* dengan kondisi yang diperlukan. Pada saat dalam periode waktu tertentu, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat berpindah atau bertransisi menuju *state* lainnya jika mendapatkan masukan atau *event* tertentu sesuai dari kondisi yang diperlukan. Transisi keadaan umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relative kompleks. Berdasarkan sifatnya, metode FSM ini sangat cocok digunakan sebagai basis perancangan perangkat lunak pengendalian yang bersifat reaktif dan *real time*. Salah satu keuntungan nyata penggunaan FSM adalah kemampuannya dalam mendekomposisi aplikasi yang relative besar dengan hanya menggunakan sejumlah kecil item *state*. Selain untuk bidang control. Penggunaan metode ini pada kenyataannya juga umum digunakan sebagai basis untuk perancangan protokol-protokol komunikasi, perancangan perangkat lunak *game*, aplikasi web dan sebagainya.

Finite State Machine di dunia kecerdasan buatan merupakan salah satu teknik yang paling sering digunakan. Alasannya yaitu :

1. Implementasinya mudah dan cepat
2. Memudahkan proses debugging. Karena telah dipecah menjadi kepingan yang lebih kecil.

3. Proses komputasi yg minimal, karena FSM hanyalah pernyataan kondisi yang dikemas dalam bentuk yang lebih elegan.
4. Fleksibel, dapat dikombinasikan dengan teknik kecerdasan buatan lain misalnya logika *fuzzy* dan jaringan saraf tiruan

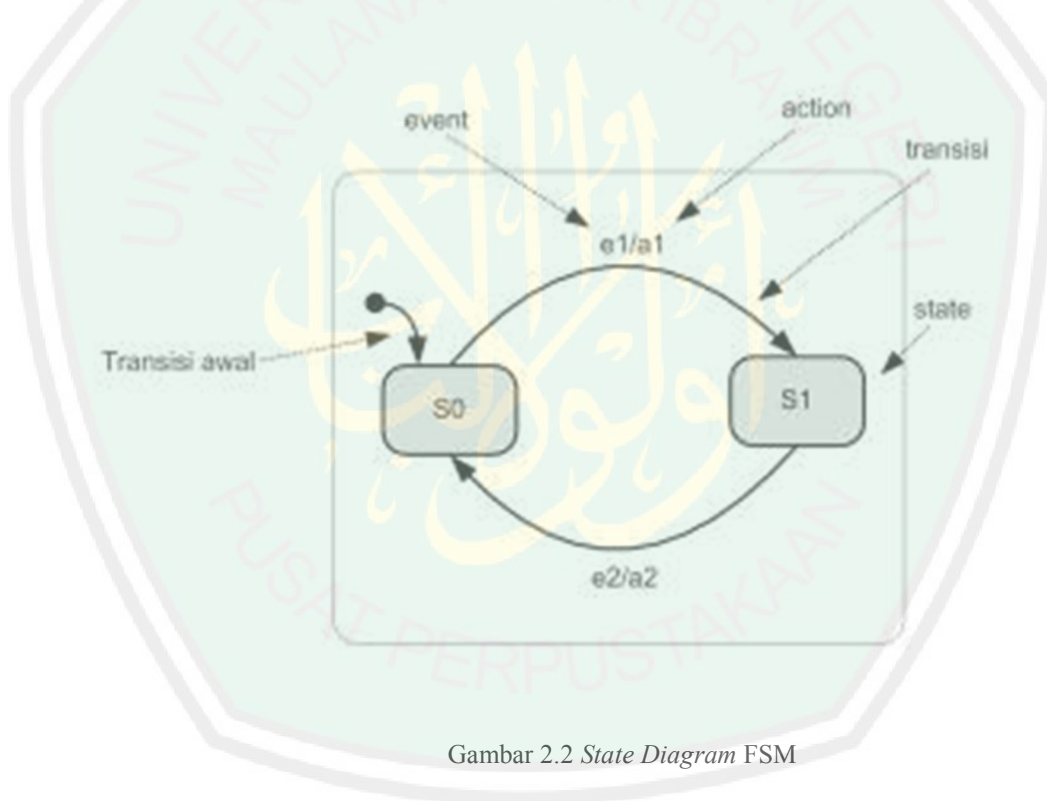


Gambar 2.1 FSM klasik

Dalam metode FSM klasik, transisi dinyatakan sebagai event, biasanya berupa daftar *enumerated* dari beberapa jenis yang dapat digunakan untuk memicu transisi. Pemeriksaan transisi kemudian dilakukan dengan mengirimkan semua state ke mesin input *event* tertentu dan menentukan apakah setiap state memiliki transisi yang dapat merespon setiap input *event*.

State diagram (diagram keadaan) umumnya merupakan salah satu bentuk representasi dari FSM. Diagram ini secara visual menggambarkan tingkah laku yang dimiliki oleh sistem kontrol yang kompleks kedalam bentuk yang lebih sederhana dan relative mudah dipahami. *State-state* yang terdapat pada sebuah sistem digambarkan sebagai lingkaran yang diberi label unik,

sedangkan transisi *state* yang diakibatkan oleh *event* tertentu direpresentasikan sebagai anak panah yang berasal dari *state* yang ditinggalkan menuju *state* yang aktif. Setiap transisi yang terjadi umumnya juga diikuti oleh aksi yang dilakukan oleh sistem yang dirancang. Secara praktis setiap diagram *state* yang dirancang akan selalu memiliki sebuah transisi awal (inisial) yang menuju salah satu *state* sejak sistem kontrol tersebut mulai dihidupkan. Gambar berikut memperlihatkan contoh penggambaran *diagram state*:



Gambar 2.2 State Diagram FSM

Diagram tersebut memperlihatkan FSM dengan dua buah *state* (S0 dan S1) dan dua buah input (e1 dan e2) serta dua buah aksi (a1 dan a2) output yang berbeda : seperti terlihat pada gambar, ketika sistem mulai dihidupkan, sistem akan bertransisi menuju *state0*, pada keadaan ini sistem akan menghasilkan *Action2* jika terjadi masukan *Event2*, sedangkan jika terjadi

Event1 maka *Action1* akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan *State1* dan seterusnya.

2.1.4 Logika Fuzzy

Istilah *fuzzy* dalam kamus Oxford didefinisikan sebagai *blurred* (kabur), *indistinct* (tidak jelas), *imprecisely defined* (didefinisikan secara tidak presisi), *confused* (membingungkan), atau *vague* (tidak jelas). Logika adalah dasar pemikiran, logika secara klasik berkaitan dengan proposisi yang dapat bernilai 1 (benar) atau 0 (salah). Jadi, logika *fuzzy* merupakan perluasan logika klasik dan teori himpunan. Nilai kebenarannya berupa variabel linguistic “benar”, tidak seperti logika klasik yang memiliki nilai logika benar dan salah. Nilai kebenaran sebuah pernyataan atau proposisi pada logika *fuzzy* berada pada *range* interval $[0,1]$ (Dwi Ana Ratna Wati, 2011).

Logika *Fuzzy* memungkinkan nilai keanggotaan antara 0 dan 1, tingkat keabuan dan juga hitam dan putih, dan dalam bentuk linguistik, konsep tidak pasti seperti "sedikit", "lumayan", dan "sangat". Logika ini berhubungan dengan *set fuzzy* dan teori kemungkinan. Logika *fuzzy* diperkenalkan oleh Dr. Lotfi Zadeh dari Universitas California, Berkeley pada 1965.

Secara umum, logika *fuzzy* adalah sebuah metodologi berhitung dengan variabel linguistik. Variabel linguistik digunakan sebagai pengganti berhitung dengan bilangan. Linguistik yang digunakan dalam logika *fuzzy* dianggap lebih dekat dengan intuisi manusia. Logika *fuzzy* dapat mengeksploitasi toleransi terhadap ketidakpresisian. Logika *fuzzy*

membutuhkan biaya yang lebih murah dalam memecahkan berbagai permasalahan yang bersifat *fuzzy* (tidak jelas). Dengan logika *fuzzy*, sistem kepakaran manusia bisa diimplementasikan ke dalam bahasa mesin secara mudah dan efisien.

Menurut Loffi A. Zadeh, dalam hampir setiap kasus yang memakai logika *fuzzy* dapat membangun sistem yang lebih cepat dan efisien. Berikut adalah alasan mengapa menggunakan logika fuzzy (Agus Naba, 2009) :

1. Konsep logika *fuzzy* sangat sederhana sehingga mudah dipahami. Kelebihannya disbanding konsep lain yang bukan pada kompleksitasnya, tetapi pada *naturalness* pendekatannya dalam memecahkan masalah.
2. Logika *fuzzy* adalah fleksibel, dalam arti dapat dibangun dan dikembangkan dengan mudah dengan mudah tanpa harus memulainya dari nol.
3. Logika *fuzzy* memberikan toleransi terhadap ketidakpastian data. Segala sesuatu di alam ini relatif tidak presisi, meskipun diamati secara lebih dekat dan hati-hati.
4. Pemodelan atau pemetaan untuk mencari hubungan data input-output dari sembarang sistem black-box bisa dilakukan dengan memakai sistem *fuzzy*.
5. Pengetahuan atau pengalaman dari para pakar dapat dengan mudah dipakai untuk membangun logika *fuzzy*.

6. Logika *fuzzy* dapat diterapkan dalam desain sistem control tanpa harus menghilangkan teknik desain sistem kontrol konvensional yang sudah terlebih dahulu ada.
7. Logika *fuzzy* berdasarkan pada bahasa manusia.

a. Himpunan Fuzzy

Himpunan *fuzzy* adalah sebuah himpunan di mana keanggotaan dari tiap elemennya tidak mempunyai batas yang jelas. Himpunan *fuzzy* mendasari konsep logika *fuzzy* yang menyatakan bahwa kebenaran dari sembarang pernyataan hanyalah masalah derajat. Pernyataan yang serupa juga telah dikemukakan oleh Lotfi A. Zadeh : “ As complexity rises, precise statements lose meaning and meaningful statements lose precision” (Agus Naba, 2009).

Himpunan *fuzzy* didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian sampai fungsi tersebut mencakup nilai pada interval $[0,1]$. Nilai keanggotaan *fuzzy* menunjukkan bahwa suatu item tidak hanya berada pada 0 atau 1, melainkan juga terletak diantaranya. Artinya, masih ada nilai-nilai yang terletak antara benar dan salah (Sri Kusumadewi, 2002).

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy* (Aplikasi logika *fuzzy* untuk pendukung keputusan, Sri Kusumadewi, Hari Purnomo, Edisi kedua, Graha Ilmu, 2010), yaitu:

a. Variable *Fuzzy*

Variable *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*. Contoh: umur, temperature, permintaan, dsb.

b. Himpunan *Fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*.

c. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Ada kalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

d. Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif dan bilangan negatif.

b. Fungsi Keanggotaan

Fungsi keanggotaan atau lebih sering disebut *membership function* mendefinisikan bagaimana tiap titik dalam ruang input dipetakan menjadi bobot atau derajat keanggotaan antara 0 dan 1. Dalam teori himpunan, ruang input dikenal sebagai *universe of discourse*.

Himpunan *fuzzy* merupakan pengembangan dari sebuah himpunan klasik. Jika X adalah *universe of discourse* dan elemen-elemennya dinotasikan dengan x , maka sebuah himpunan *fuzzy* A dalam X didefinisikan dengan :

$$A = \{x, \mu_A(x) \mid x \in X\}$$

$\mu_A(x)$ adalah fungsi keanggotaan dari x dan A . Fungsi keanggotaan memetakan tiap elemen dari x menjadi derajat keanggotaan antara 0 dan 1. Beberapa kesimpulan dari himpunan dan fungsi keanggotaan *fuzzy* :

1. Himpunan *fuzzy* menekankan konsep variabel samar (*vague of fuzzy variable*) seperti variabel hari akhir minggu, suhu panas, pelari cepat dan lain-lain.
2. Himpunan *fuzzy* mengizinkan keanggotaan parsial dari suatu himpunan seperti hari Jumat yang dianggap sebagai hari akhir minggu namun dengan derajat di bawah 1.

3. Derajat keanggotaan *fuzzy* dalam *fuzzy* set berkisar antara 0 sampai 1.
4. Tiap fungsi keanggotaan μ berasosiasi dengan sebuah himpunan *fuzzy* tertentu dan memetakan suatu nilai input ke nilai derajat keanggotaan yang sesuai. Misalnya dalam kasus himpunan *fuzzy* orang berbadan “tinggi” mempunyai fungsi keanggotaan sendiri, yaitu μ_{tinggi} , yang berbeda dengan fungsi keanggotaan dari himpunan *fuzzy* orang berbadan “rendah”, yaitu μ_{rendah} .

Fungsi keanggotaan dapat dibuat dalam berbagai bentuk. Biasanya dalam bentuk suatu kurva yang menunjukkan pemetaan titik-titik input data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan yaitu (Kusumadewi S, Purnomo H, 2010) :

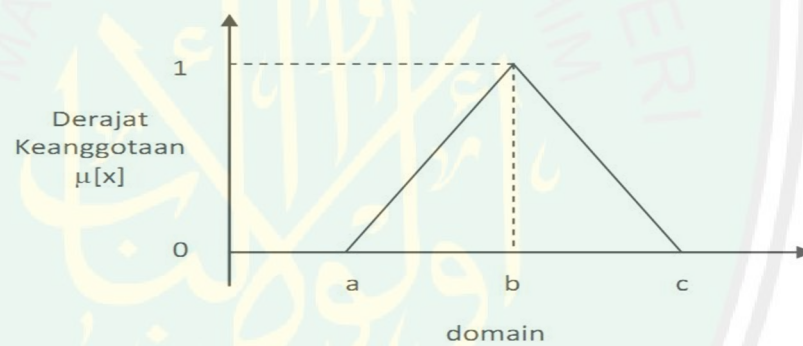
- a. Representasi Linear

Pada representasi linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan himpunan *fuzzy* yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju ke nilai domain yang memiliki derajat

keanggotaan lebih tinggi. Kedua, merupakan kebalikan dari yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah.

b. Representasi Kurva Segitiga

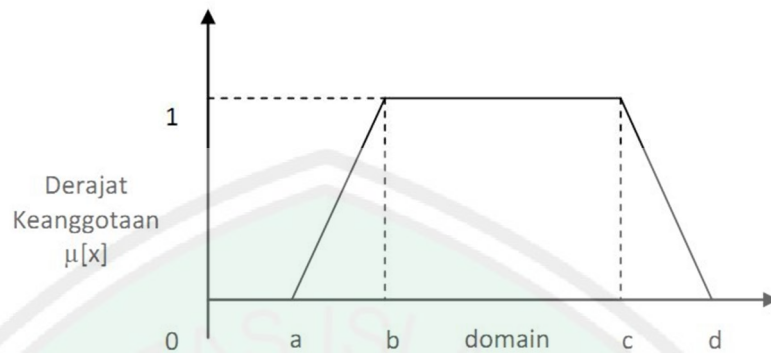
Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linear).



Gambar 2.3 Kurva segitiga

c. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada titik yang memiliki nilai keanggotaan.



Gambar 2.4 Kurva Trapesium

2.1.5 Android

Android adalah sebuah sistem operasi berbasis kernel linux pada perangkat mobile yang mencakup sistem operasi, middleware dan aplikasi. Android dibeli oleh Google Inc. yang kemudian dilakukan pengembangan. Perilisan perdana Android dilakukan pada 5 November 2007. Selain itu, Google merilis kode-kode Android di bawah lisensi Apache. Android merupakan generasi baru platform mobile dilisensikan di bawah GNU General Public Lisensi Versi 2 (GPLv2) yang memperbolehkan pihak ketiga untuk melakukan pengembangan dengan ketentuan harus terus jatuh di bawah *terms* yang sama. Android didistribusikan di bawah Lisensi Apache Software (ASL/Apache2), yang dapat memungkinkan distribusi kedua dan seterusnya. Pengembang diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apa pun yang diinginkan (Nazruddin Safaat, 2012:1-3).

Ada beberapa keunggulan Android diantaranya (Nazruddin Safaat, 2012:3):

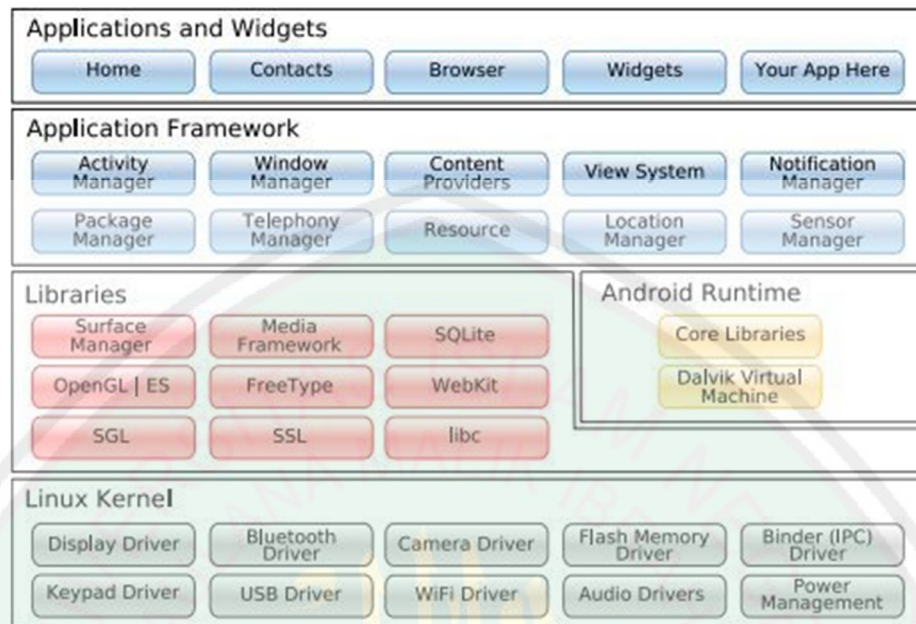
- a. Lengkap (Complete Platform) : Para desainer dapat melakukan pendekatan secara komprehensif dalam melakukan pengembangan platform Android. Android telah banyak menyediakan tools dalam membangun aplikasi berbasis Android.
- b. Terbuka (Open Source Platform) : Pengembang dapat dengan bebas untuk mengembangkan aplikasi mereka.
- c. Free (Free Platform) : Android adalah platform/aplikasi yang bebas untuk pengembangan. Tidak ada lisensi atau biaya royalty. Tidak ada biaya keanggotaan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Semua aplikasi berbasis Android bebas untuk didistribusikan dan diperdagangkan dalam bentuk apa pun.

Pada dasarnya, Android memiliki arsitektur yang dijelaskan dan digambarkan sebagai berikut (Nazruddin Safaat, 2012:7-8):

- a. *Application dan Widgets*, merupakan *layer* dimana kita berhubungan dengan aplikasi saja, seperti aplikasi untuk *browsing*. Selain itu, fungsi-fungsi seperti telepon dan sms juga terdapat pada layer ini.
- b. *Application Frameworks*, merupakan layer dimana para pembuat aplikasi melakukan pengembangan/ pembuatan aplikasi yang akan dijalankan di sistem operasi Android. Beberapa komponen yang

terdapat pada layer ini adalah, *Views*, *Content Provider*, *Resource Manager*, *Notification Manager* dan *Activity Manager*.

- c. *Libraries*, merupakan layer dimana fitur-fitur Android berada yang dapat digunakan untuk menjalankan aplikasi. *Library* yang disertakan seperti *library* untuk pemutaran audio dan video, tampilan, grafik, SQLite, SSL dan Webkit, dan 3D.
- d. *Android Run Time*, merupakan layer yang berisi *Core Libraries* dan *Dalvik Virtual Machine (DVK)*. *Core libraries* berfungsi untuk menerjemahkan bahasa Java/C. Sedangkan DVK merupakan sebuah virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien.
- e. *Linux Kernel*, merupakan *layer* yang berfungsi sebagai abstraction/ pemisah antara hardware dan software. Linux kernel inilah yang merupakan inti sistem operasi dari Android yang berfungsi untuk mengatur *sistem proses*, *memory*, *resource*, dan *driver*. *Linux kernel* yang digunakan Android adalah *linux kernel release 2.6*.



Gambar 2.5 Arsitektur Android

2.1.6 Unity 3D

Unity 3D adalah sebuah teknologi yang memungkinkan pengembangan game menjadi lebih mudah dan lebih baik. Unity adalah *game engine* yang dapat digunakan untuk membangun sebuah video *game*. Dengan menggunakan Unity, kita dapat membangun video game dengan lebih cepat dan mudah dari sebelumnya. Unity juga menambahkan tools untuk pembuatan game yang dimainkan di iPhone, iPod, iPad, Android, Xbox Live Arcade, PS3 dan Nintendo WiiWare. Masing-masing tools tersebut adalah *add-on* atau fungsi tambahan yang disediakan dalam paket Unity dengan biaya tambahan.

2.1.7 Blender

Blender adalah sebuah perangkat lunak komputer grafis 3D untuk mengembangkan film animasi, efek visual, game 3D dan software. Blender merupakan program yang *open source* sehingga para pengembang dapat dengan bebas menggunakan atau mendistribusikannya. Selain itu, banyak fitur-fitur yang ditawarkan seperti 3D modeling, texturing, rigging, skinning, fluid, smoke simulation, animation, rendering, compositing, motion tracking dan video editing.

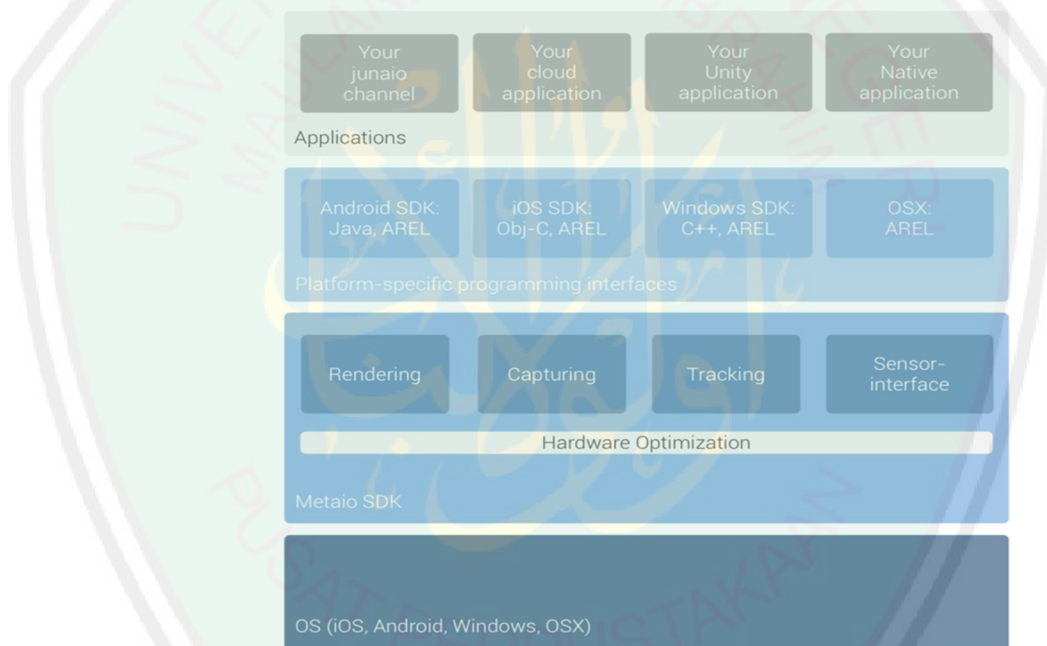
Blender bersifat cross platform dan berjalan dengan baik di Linux, Windows dan Macintosh.

2.1.8 Metaio SDK

Metaio SDK merupakan framework modular yang mencakup *capture component*, *sensor interface component*, *rendering component*, *tracking component* dan *Metaio SDK interface*. Interface menyediakan interaksi antara aplikasi dan empat komponen modular. Pada konfigurasi framework ini, rincian mengenai implementasi di dalamnya disembunyikan sehingga pengembang tidak perlu khawatir bagaimana proses capturing, rendering, sensor atau tracking. Fungsi- fungsi utamanya di wujudkan melalui API SDK dengan setiap bagian-bagian di dalamnya sehingga dapat memudahkan implementasi pada aplikasi berbasis *Augmented Reality*.

Metaio SDK kompatibel dengan berbagai *platform* yang meliputi Android, IOS, Unity3D dan Windows. Antarmuka spesifik pada tiap platform dapat dengan mudah berinteraksi dengan lingkungan pengembangan. Aplikasi *Augmented Reality* terbentuk dari kombinasi antara Metaio SDK dan platform SDK.

Framework Metaio SDK dapat digambarkan sebagai berikut :



Gambar 2.6 Struktur Framework Metaio SDK

a. Instant Tracking

Teknologi *Instant Tracking* tidak memerlukan penentuan terlebih dahulu untuk melakukan tracking. Apapun dapat digunakan sebagai media tracking. Dalam berbagai kasus, teknologi *instant tracking* yang

digunakan memiliki berbagai jenis yang berbeda-beda. Berikut adalah jenis-jenis *instant tracking* :

1. Instant Tracking (2D)

Instant tracking jenis ini adalah instant tracking biasa yang bisa melakukan pelacakan dalam ruang dua dimensi (2D). Mekanisme dari instant tracking jenis ini memungkinkan pengembang aplikasi untuk mengambil gambar dari suatu obyek dan menggunakannya sebagai referensi pelacakan markerless. Permukaan objek yang paling cocok untuk jenis ini yaitu datar, kontras yang tinggi dan perangkat pengguna harus sejajar dengan permukaan benda.

2. Instant Tracking dengan *Gravity Sensor* (Sensor Gravitasi)

Instant tracking jenis ini memiliki fungsi yang sama dengan Instant Tracking 2D seperti yang dijelaskan di atas, tetapi dengan menambahkan informasi gravitasi dari sensor gravitasi perangkat pengguna ke dalam skenario AR, selanjutnya akan menstabilkan posisi objek. *Instant tracking* jenis ini pengguna bisa mendapatkan preview dari daerah dilacak sebelum mengaktifkan pelacakan untuk mendapatkan hasil pelacakan yang lebih baik.

3. Instant Tracking dengan SLAM

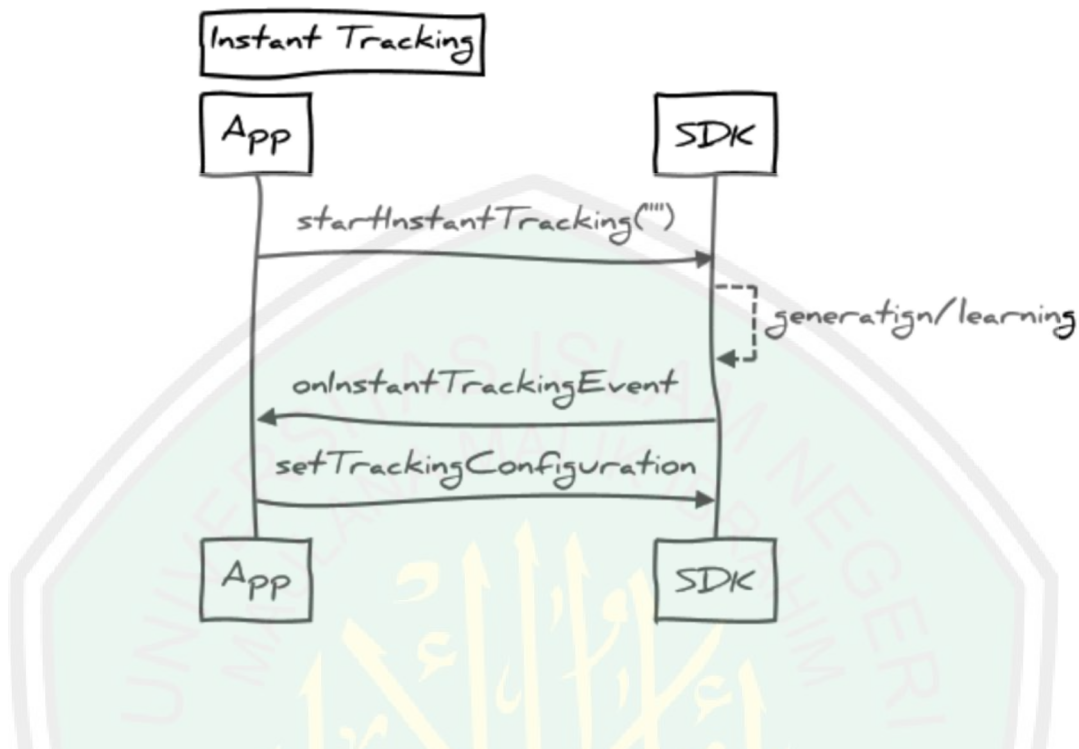
Instant tracking jenis ini merupakan instant tracking yang bisa melakukan pelacakan dalam ruang tiga dimensi (3D). Mekanisme dari

instant tracking jenis ini memungkinkan pengembang aplikasi untuk menginisialisasi pelacakan di permukaan apapun. Kontras tinggi objek tiga dimensi adalah permukaan yang paling cocok untuk jenis ini. Teknologi ini sering disebut sebagai SLAM (Simultaneous Localization And Mapping) yang dapat membuat peta 3D dan diperbarui secara *real time*, berdasarkan fitur dari adegan dilihat.

4. Instant Tracking dengan SLAM *Extension* dan *Gravity Sensor*

Instant tracking jenis ini memiliki fungsi yang sama dengan *Instant Tracking* 3D seperti yang sudah dijelaskan di atas, akan tetapi juga dapat menginisialisasi pelacakan secara langsung dengan mengambil gambar dari objek 2D. Yang paling cocok untuk inisialisasi pelacakan jenis ini adalah permukaan objek 2D yang datar dan kontras tinggi. Selain jenis ini menambahkan informasi gravitasi dari sensor gravitasi perangkat pengguna dalam skenario AR, tetapi juga akan menstabilkan posisi objek. *Instant tracking* jenis ini pengguna bisa mendapatkan preview dari daerah dilacak sebelum mengaktifkan pelacakan untuk mendapatkan hasil pelacakan yang lebih baik.

Sequence diagram dari penggunaan *Instant Tracking* dalam Metaio SDK dapat digambarkan sebagai berikut :



Gambar 2.7 Sequence Diagram Instant Tracking (Sumber : Metaio Developer Portal)

2.2 Penelitian Terkait

Penelitian yang memiliki keterkaitan dengan penelitian ini antara lain:

1. Penelitian oleh Trond Nilsen *et all* (2004) tentang *game* berbasis Augmented Reality. Pada penelitiannya membahas mengenai kekuatan dan kelemahan dari augmented reality dalam model informal pengalaman *gaming* yang dibagi menjadi empat aspek yaitu fisik, mental, sosial dan emosional. Game yang dibangun berbasis Augmented Reality dapat berbaur untuk meningkatkan gaya permainan yang sudah ada dan menciptakan sesuatu yang baru. Implementasi dari penelitiannya adalah dengan membangun *game* AR Worms yang

merupakan implementasi ulang dari game klasik Worm menggunakan Augmented Reality. Hasil dari penelitiannya adalah beberapa ide untuk game berbasis AR dengan jenis game strategy dan role playing game.

2. Penelitian oleh Troels L. Andersen *et all* (2004) membahas tentang desain BattleBoard 3D (BB3D) yang merupakan prototipe game berbasis ARToolkit dengan menggunakan LEGO sebagai markernya. BB3D adalah jenis baru dari game AR *board* tradisional dengan fitur dari game komputer. Pada penelitian ini juga membahas konsep dalam permainan, isu-isu implementasi, pengaturan fisik untuk permainan, user interface, serta potongan tailorable dan prajurit. Percobaan penelitian ini telah menjanjikan hasil berupa jenis baru pada game AR board ini dan rencana untuk membuat penelitian lebih lanjut dan evaluasi yang sistematis untuk mengembangkan *game* generasi berikutnya.
3. Penelitian oleh Fekolkin Roman (2013) membahas analisis tentang game augmented reality pada platform android. Dalam penelitiannya seputar Augmented Reality di game di Android Platform dilakukan dengan menguji 108 game dari Google Play Market dan dengan menganalisis ratusan tinjauan pengguna untuk menentukan tingkat penerimaan dan tingkat stabilitas teknis dari game mobile berbasis teknologi augmented reality tersebut.
4. Penelitian oleh Yunifa Mifhachul Arif *et all* (2011) membahas tentang metode Hierarchy Finite State Machine dan Logika Fuzzy untuk

strategi menyerang pada game FPS. Pada penelitiannya menjelaskan tentang bagaimana membuat strategi menyerang yang dilakukan oleh NPC dengan menggunakan Hierarchy Finite State Machine untuk mendesain perilaku. Selanjutnya digunakan logika fuzzy untuk menentukan respon perilaku terhadap kondisi yang dihadapi adalah menyerang brutal, menyerang, bertahan dan melarikan diri. Strategi menyerang dalam penelitian ini mempunyai tingkat kemenangan hingga 80% ketika diuji coba dengan musuh yang mempunyai perilaku umum yaitu menyerang dan menghindari tembakan.

5. Penelitian oleh Surya Adi Wijaya *et all* (2009) membahas tentang metode Fuzzy State Machine (FuSM) untuk menghasilkan variasi respon pada NPC. Pada penelitiannya bertujuan untuk mengetahui adanya variasi respon dari NPC jika dibandingkan dengan hanya menggunakan FSM. Dari hasil percobaannya didapatkan bahwa FuSM bisa menghasilkan variasi respon NPC yang lebih banyak. Variasi respon tersebut bisa berbeda ketika berinteraksi dengan obyek lain.
6. Penelitian oleh Miftakhur Rokhman (2011) membahas tentang metode *Finite State Machine* sebagai agen cerdas animasi wajah pada game Assyuhada. Pada penelitiannya menjelaskan tentang implementasi *Finite State Machine* untuk menghasilkan animasi-animasi wajah yang bervariasi. Hasil percobaannya berjalan dengan baik sehingga dapat dikatakan metode *Finite State Machine* cocok diterapkan pada animasi wajah dalam game Assyuhada.

7. Penelitian oleh Galih Suyoga (2012) membahas tentang kombinasi dua metode kecerdasan buatan yaitu FSM dan Fuzzy Logic atau sering juga disebut dengan FuSM (Fuzzy State Machine). Logika fuzzy dalam penelitiannya digunakan untuk pengambilan keputusan dari Non-Playable Character (NPC) agar perilakunya lebih realistis dan lebih dinamis. Hasil dari penelitian tersebut adalah game memiliki respon yang dinamis dengan memanfaatkan nilai fuzzy. Pemanfaatan fungsi keanggotaan seperti segitiga dan trapesium mempengaruhi hasil keluaran dari sistem fuzzy, dimana jika menggunakan fungsi keanggotaan trapesium, maka respon objek lebih mudah ditebak daripada fungsi segitiga. Inferensi mamdani dapat lebih banyak menghasilkan variasi respon daripada dengan menggunakan inferensi sugeno.
8. Penelitian oleh Ady Wicaksono (2013) membahas tentang metode Fuzzy State Machine (FuSM) dipergunakan sebagai metode untuk merancang variasi respon serangan agen pada game FPS melalui pemilihan jenis senjata yang sesuai dengan jarak jangkanya. Perilaku agen dengan variasi respon yaitu serangan jarak dekat, jarak menengah dan jarak jauh. Perilaku ini digunakan oleh tiga agen pada saat bertempur dengan bergantung pada perubahan jarak musuh (pemain), serta tingkat kesehatan (hp) dari masing-masing agen. Selanjutnya perilaku disimulasikan dengan First Person Shooter game engine untuk dapat diamati perilakunya. Hasil simulasi agen 1 menunjukkan variasi

respon menyerang jarak dekat dan menengah, dengan kecenderungan serangan jarak dekat hingga 68%. Agen 2 menunjukkan variasi respon menyerang jarak menengah dan jarak jauh dengan kecenderungan serangan jarak menengah hingga 78%. Agen 3 menunjukkan variasi respon serangan lengkap, mulai dari serangan jarak dekat, jarak menengah dan jarak jauh dengan kecenderungan serangan jarak jauh hingga 65%. Agen dengan perilaku serangan berbasis fuzzy lebih kompetitif dibandingkan agen tanpa logika fuzzy dengan perbandingan prosentase kemenangan 60%:20%.

BAB III

METODOLOGI PENELITIAN

3.1 Desain Penelitian

Penelitian ini menggunakan metode kuantitatif, deskriptif, dan empirikal. Metode kuantitatif ini dilakukan dengan melakukan pengukuran dijabarkan ke dalam beberapa komponen masalah, variabel dan indikator. Setiap variabel yang ditentukan dan diukur dengan memberi simbol-simbol angka yang berbeda-beda sesuai dengan kategori informasi yang berkaitan dengan variabel tersebut. Kedalaman penelitian berupa dalam bentuk deskriptif dimana game ini merupakan bentuk aplikasi yang di dalamnya terdapat kontinuitas.

Analisis menggunakan deskriptif dimana dari teori yang sudah ada dilakukan penerapan terhadap suatu aplikasi agar lebih mudah untuk memahami suatu konsep atau algoritma yang ada ketika digunakan untuk menyelesaikan suatu permasalahan. Objektivitasnya menggunakan empirikal, yakni dengan pengujian *visibility* atau kelayakan suatu metode atau algoritma yang diterapkan dalam menyelesaikan suatu masalah

3.2 Prosedur Penelitian

Prosedur penelitian diperlukan agar tahapan pengerjaan dapat dilakukan secara berurutan. Berikut ini adalah tahap-tahap pembuatan game pada penelitian yang dilakukan antara lain :

3.1.1 Studi Literatur

Pada tahap ini dimulai dengan studi literatur tentang pembuatan game. Pengumpulan informasi-informasi yang bisa dijadikan referensi dalam pengembangan game pada penelitian ini. Referensi atau informasi yang diperlukan adalah seperti pengembangan game dengan Unity 3D, pengembangan Augmented Reality dengan Metaio SDK dan metode *Finite State Machine* serta logika *fuzzy* sebagai referensi terkait dengan metode kecerdasan buatan yang digunakan pada penelitian ini.

3.1.2 Analisis Parameter yang Dibutuhkan

Tahap ini merupakan tahap untuk menganalisis sejumlah parameter yang dibutuhkan dalam penelitian ini. Adapun parameter yang digunakan pada penelitian ini merupakan parameter yang digunakan untuk implementasi dan pengujian metode yang digunakan. Hasil keluaran dari metode ditentukan oleh parameter-parameter yang diinputkan. Sehingga akan diketahui hasil uji coba dari metode tersebut.

3.1.3 Desain *Finite State Machine* dan Logika *Fuzzy*

Tahap ini merupakan tahap untuk mendesain atau merancang metode yang digunakan. Pertama adalah membuat rancangan *Finite State Machine* dalam bentuk *state* diagram. Kemudian mendesain perilaku pada *state* menyerang dengan menggunakan logika *fuzzy* agar menjadi bervariasi.

3.1.4 Coding dan Implementasi

Tahap ini merupakan tahap implementasi metode yang digunakan, kemudian akan dimasukkan ke dalam *game* yang telah dirancang. *Game* yang akan dibangun pada perangkat *smartphone* berbasis android. Pada tahap ini pula metode kecerdasan buatan akan diterapkan akan diimplementasi ke dalam bahasa pemrograman C#.

3.1.5 Menampilkan Objek dengan *Augmented Reality*

Tahap ini merupakan tahap untuk menampilkan objek-objek pada *game* agar dapat tampil di lingkungan nyata melalui perantara kamera perangkat *smartphone*. Selain itu juga bagaimana mengintegrasikan antara kamera *smartphone* dengan *game* agar dapat berjalan dengan baik

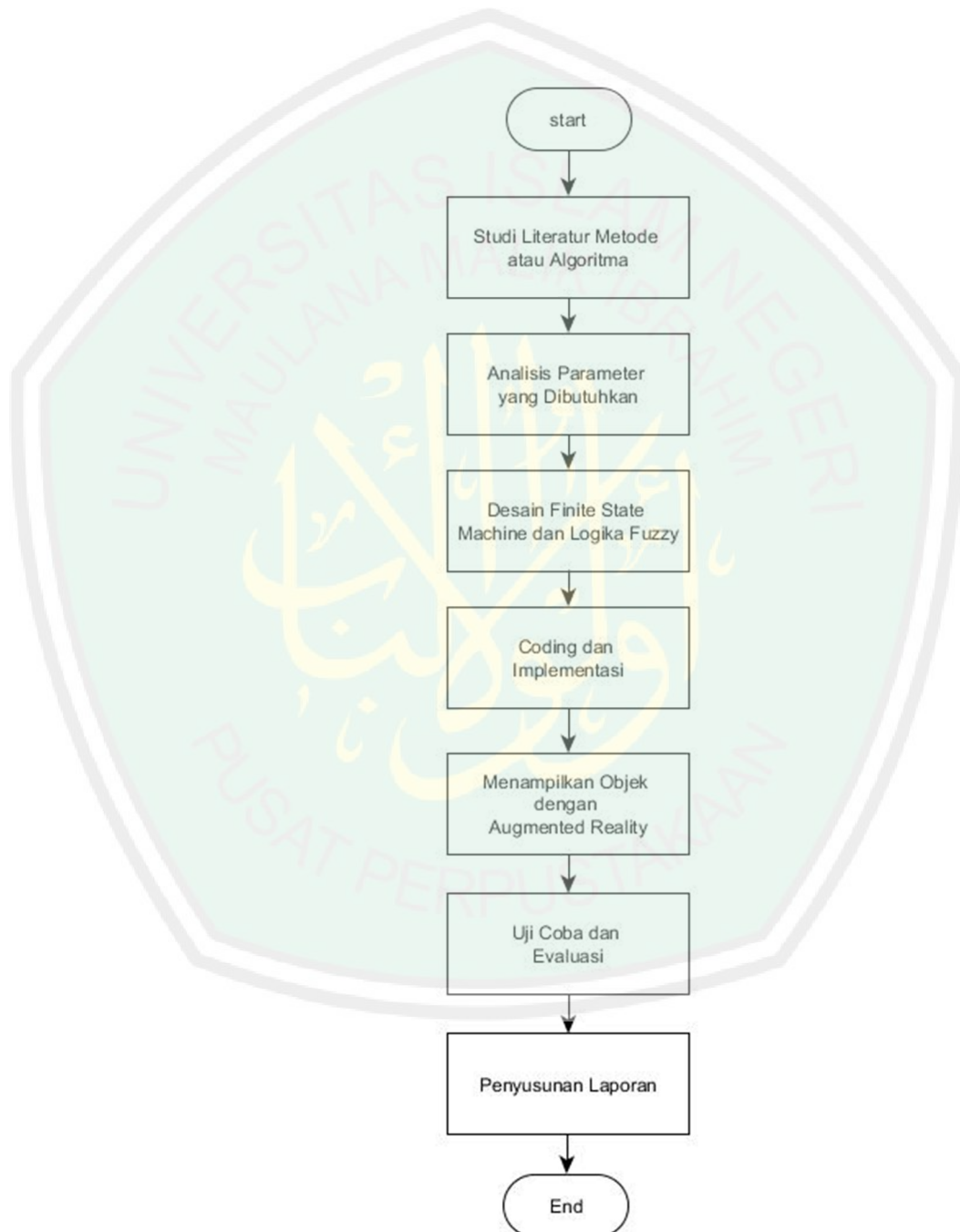
3.1.6 Uji Coba dan Evaluasi

Tahap ini merupakan tahap untuk menguji coba *game* yang sudah selesai dibuat, kemudian melakukan evaluasi terhadap metode kecerdasan buatan yang digunakan. Evaluasi dilakukan untuk mengukur seberapa baik metode *Finite State Machine* dan logika *Fuzzy* dalam mengatur dan mengontrol perilaku NPC pada *game*.

3.1.7 Penyusunan Laporan

Tahap yang terakhir adalah penyusunan laporan yang berisi hasil dari seluruh dokumentasi dari keseluruhan pelaksanaan penelitian. Hasil penelitian diharapkan bisa bermanfaat dan bisa dilakukan penelitian-penelitian lebih lanjut terkait dengan penelitian ini.

Berikut adalah permodelan dari dari tahap-tahap pengerjaan penelitian.



Gambar 3.1 Tahap-Tahap Penelitian

3.3 Desain Aplikasi

3.3.1 Deskripsi Game

Game ini merupakan game 3D ber-genre *Survival Adventure*. *Game* ini berbasis *Augmented Reality*. *Game* ini dibuat pada perangkat mobile yang berplatform *Android*. Dalam *game* ini disisipi konten budaya lokal yaitu rumah adat. Sasaran pengguna *game* ini adalah semua umur terutama kepada kalangan remaja dan anak-anak.

3.3.2 Storyline

Game ini menceritakan tentang petualangan kwikki dalam menjelajahi pulau-pulau di Indonesia. Penjelajahannya dimulai dari pulau Sumatera, Jawa, Kalimantan, Sulawesi dan Papua. Setiap pulau yang dikunjungi akan dikunjungi akan mengambil latar belakang permainan di sebuah daerah dengan beragam kebudayaannya. Misinya adalah menjelajahi setiap daerah yang dikunjungi dan mendapatkan informasi mengenai rumah adat di daerah tersebut. *Game* ini berbasis *Augmented Reality* sehingga objek-objek di dalam *game* nantinya akan ditampilkan dalam bentuk nyata melalui kamera. Dalam permainan ini, pemain akan menemukan rintangan di sepanjang perjalanan seperti musuh-musuh yang siap menghadang. Pemain dapat melakukan aksi menyerang musuh dengan menembak atau memukulnya. Jika pemain diserang musuh atau terkena serangan musuh, maka nyawa akan berkurang. *Game* ini memiliki lima *stage*, setiap *stage* berdasarkan pulau yang dikunjungi.

3.3.3 *Gameplay*

Pemain berusaha memenangkan permainan dengan cara mengalahkan musuh-musuh dan mendapatkan informasi mengenai rumah adat di setiap daerah yang dikunjungi. Setelah pemain memenangkan permainan, maka akan lanjut ke *stage* berikutnya. Setiap *stage* merupakan daerah-daerah yang dikunjungi di setiap pulau. Berikut penjelasan setiap daerah yang dikunjungi :

- Stage 1 (Sumatera) : Sumatera Barat
- Stage 2 (Jawa) : Yogyakarta
- Stage 3 (Kalimantan) : Kalimantan Selatan
- Stage 4 (Sulawesi) : Sulawesi Selatan
- Stage 5 (Papua) : Papua

3.3.4 *Skenario Game*

Skenario pada *game* ini dimulai dari pemain dan NPC berada pada posisi *idle*. Berikut adalah skenario *game* :

- Player dan musuh sama-sama berada pada posisi *idle*.
- Setiap NPC Musuh akan muncul (*spawn*) secara acak.
- Player berjalan mengelilingi arena untuk mengumpulkan item-item dan alat musik.
- NPC Musuh berada pada *state* berkelana yakni berkelana mengelilingi arena sampai berpindah ke *state* berikutnya.
- NPC Musuh dikontrol dengan menggunakan FSM dan logika fuzzy.

- Jika pemain bertemu dengan NPC Musuh, maka NPC Musuh akan merespon pemain dengan aksi seperti mengejar, menyerang dan menghindar.
- Pemain dan NPC Musuh memiliki parameter berupa jarak, kesehatan dan aksi player.
- Jika kesehatan musuh memburuk maka NPC akan kabur ketika melihat pemain dan keluar dari *state* menyerang.
- Pemain dapat menyerang NPC Musuh. Jika NPC Musuh kalah maka akan mendapatkan poin, jika pemain kalah maka permainan selesai.
- Ketika pemain telah berhasil mengalahkan NPC musuh, maka akan menampilkan profil rumah adat pada *stage* tersebut.

3.3.5 Storyboard

- Tampilan Menu Utama

Untuk tampilan menu dari *game* ini ditunjukkan seperti gambar 3.2.



Gambar 3.2 Tampilan Awal

- Tampilan *Stage* Permainan

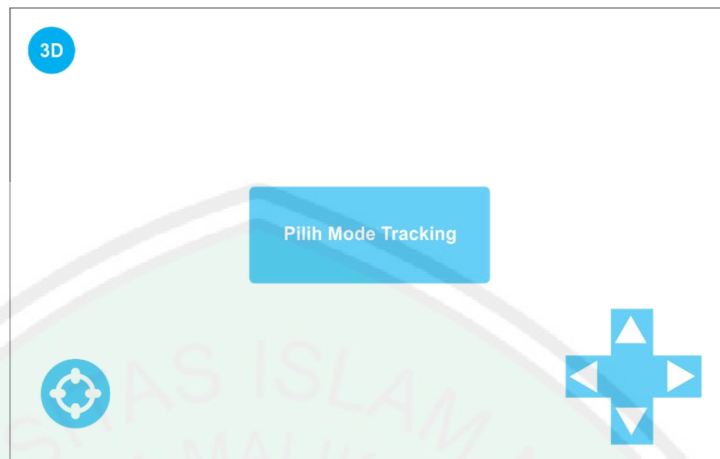
Pada tampilan ini akan ditampilkan stage - stage yang akan dimainkan. Stage mengambil latar belakang dari lima provinsi yang ada di Indonesia seperti yang ditunjukkan oleh gambar 3.3.



Gambar 3.3 Tampilan Stage Permainan

- Memulai permainan

Tampilan ini merupakan tampilan pada saat bermain. Layar permainan menggunakan kamera dari smartphone, kemudian pemain melakukan pelacakan dengan mode tracking 3D. Ketika memilih mode tracking maka otomatis sistem akan melakukan proses *scanning* terhadap permukaan yang akan dijadikan *marker*. Objek-objek dalam game nantinya akan muncul melalui layar kamera. Berikut ini ditunjukkan oleh gambar 3.4.

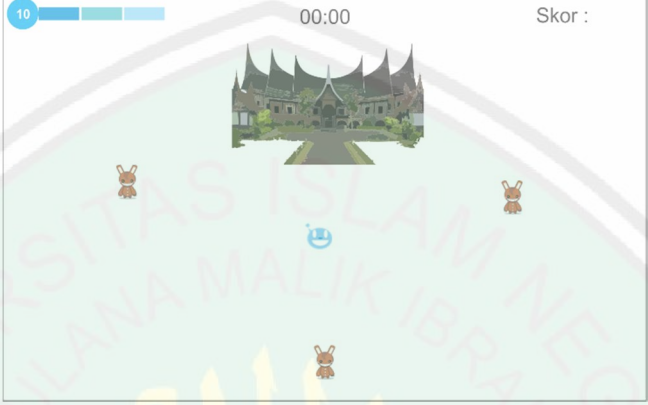
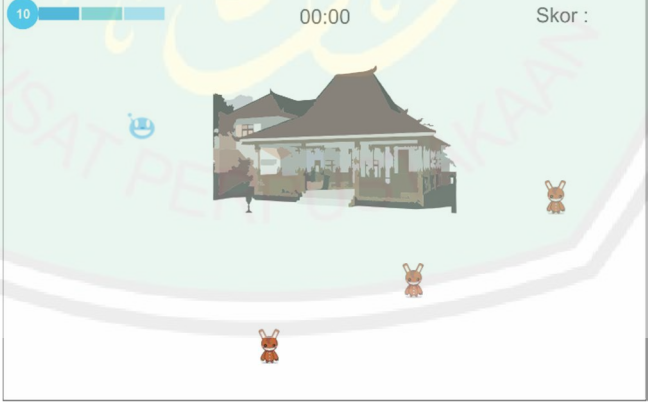


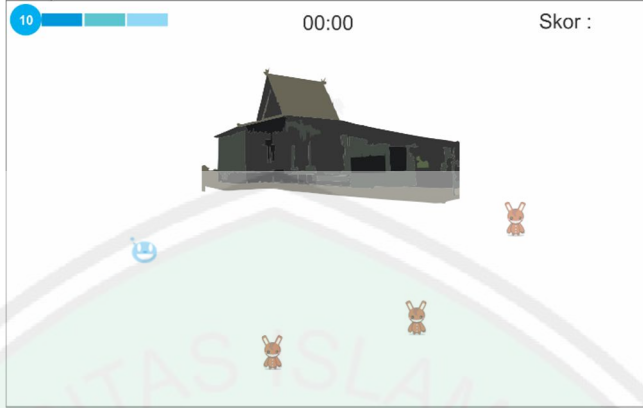
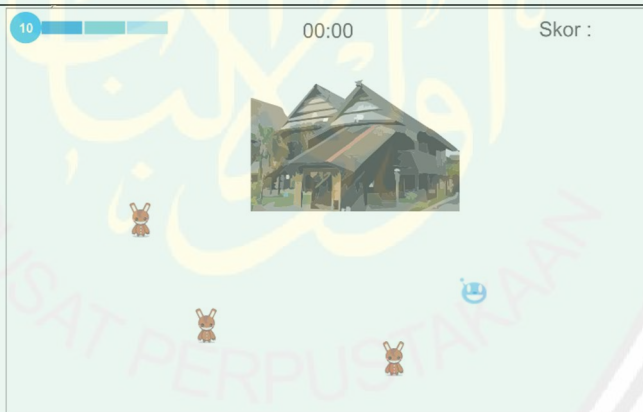
Gambar 3.4 Layar Permainan

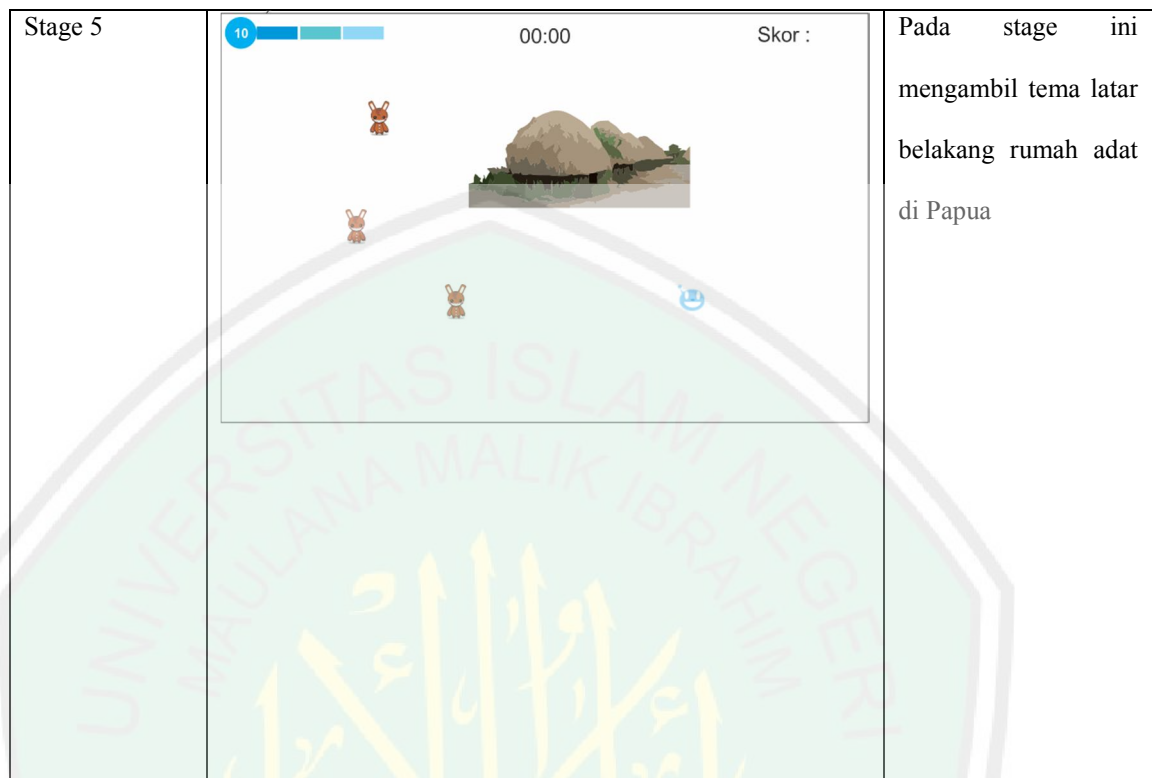
- *Stage* permainan

Tampilan ini merupakan tampilan ketika proses tracking berhasil dan akan muncul objek-objek berupa arena permainan dan karakter-karakter dalam *game* pada layar kamera. Pada arena permainan, latar belakang akan dibuat seperti sedang berada di daerah dengan kultur kebudayaan seperti rumah adat yang ada disekitarnya. Permainan memiliki lima stage diantaranya Stage 1 (Sumatra Barat), Stage 2 (Yogyakarta), Stage 3 (Kalimantan Selatan), Stage 4 (Sulawesi Selatan) dan Stage 5 (Papua).

Tabel 3. 1 Stage Permainan

Nama Stage	Tampilan	Keterangan
Stage 1		<p>Pada stage ini mengambil tema latar belakang rumah adat di Sumatra Barat.</p>
Stage 2		<p>Pada stage ini mengambil tema latar belakang rumah adat di Daerah Istimewa Yogyakarta</p>

<p>Stage 3</p>		<p>Pada stage ini mengambil tema latar belakang rumah adat di Kalimantan Selatan</p>
<p>Stage 4</p>		<p>Pada stage ini mengambil tema latar belakang rumah adat di Sulawesi Selatan</p>



3.3.6 Item Permainan

Item-item dalam permainan ini terdiri dari :

- Koin

3.3.7 Scoring

Sistem scoringnya berasal dari jumlah musuh yang kalah dan item-item yang ditemukan di sepanjang permainan. Masing-masing mempunyai bobot skor sebagai berikut :

- Musuh kalah : 10 poin
- Item berupa koin-koin : 5 poin

3.3.8 Konten – Konten Pada Aplikasi

Konten-konten yang terdapat pada game ini adalah :

- Konten Budaya Lokal

Di dalam *game* ini disisipi konten-konten budaya lokal adalah rumah adat tradisional. Selain itu juga terdapat keterangan mengenai budaya lokal tersebut. Budaya-budaya lokal diambil dari beberapa daerah yang ada di Indonesia, contohnya seperti di pulau Sumatera yaitu Sumatera Barat. Penyajian konten ini berupa objek-objek yang dijadikan sebagai pendukung skenario dalam *game*, artinya hanya sebagai objek-objek sesuai dengan latar belakang game ini. Konten ini akan disajikan berupa bentuk 3D dan ditampilkan melalui kamera *smartphone*. Setiap selesai mengalahkan musuh, maka akan tampil profil mengenai rumah adat tersebut.

Berikut ini adalah konten-konten rumah adat yang akan disajikan :

1. Rumah Gadang (Sumatra Barat)
2. Rumah Joglo (Yogyakarta)
3. Rumah Banjar (Kalimantan Selatan)
4. Rumah Balla Lompoa (Sulawesi Selatan)
5. Rumah Honai (Papua)

- Latar Belakang (*Background*)

Latar Belakang atau *background* dari *game* ini adalah di tampilan tiga dimensi bentuk rumah adat tradisional yang menggambarkan kultur kebudayaan di daerah tersebut.

- Tokoh utama

Tokoh utama adalah karakter yang dimainkan oleh pemain. Tokoh utama dari *game* ini bernama “kwikki” yang merupakan maskot dari sebuah jejaring sosial lokal di Indonesia.



Gambar 3.5 Kwikki (Karakter/Tokoh Utama)

- *Non Playable Character (NPC)* Musuh

NPC merupakan karakter-karakter yang tidak dikendalikan oleh pemain, melainkan karakter yang diberi kecerdasan buatan untuk mengontrol perilaku seperti aksi dan respon pada karakter utama. Dalam hal ini, metode *Finite State Machine* digunakan untuk mengontrol perilaku NPC tersebut. NPC dalam *game* ini adalah musuh. Berikut ini adalah karakter NPC musuh dalam *game*

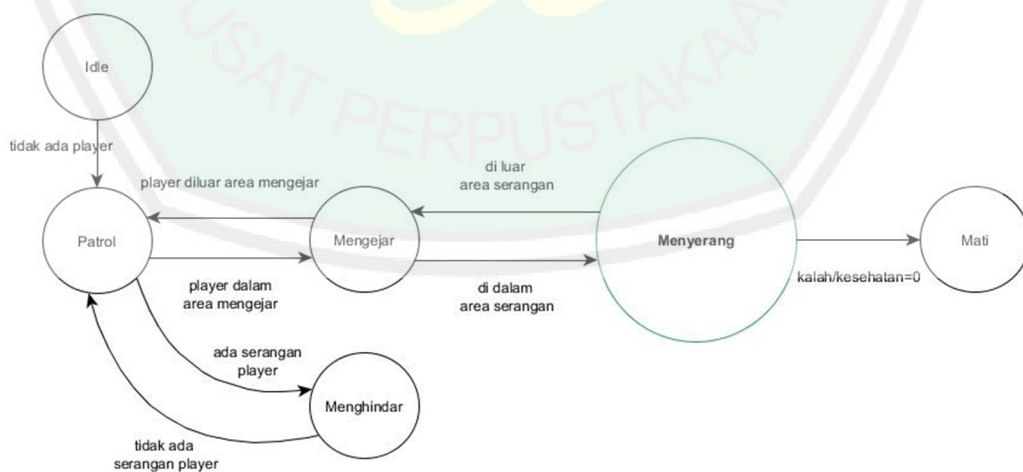
:



Gambar 3.6 Rabbit

3.4 Rancangan *Finite State Machine*

Dari skenario sebelumnya, maka dapat dibuat alur *Finite State Machine* (FSM) untuk NPC musuh. FSM terdiri dari beberapa *state* yang dapat bertransisi satu sama lain. Berikut adalah FSM NPC musuh ditunjukkan pada gambar 3.7.



Gambar 3.7 FSM NPC musuh

Berikut adalah *state-state* pada game ini :

1. Idle

Posisi awal NPC musuh.

2. Patrol

NPC musuh akan bergerak random jika jarak atau jangkauan player terhadapnya sangat jauh.

3. Mengejar

NPC akan mengejar target yang ada di range tertentu. Jika jarak atau jangkauan *player* terhadap NPC musuh pada jarak pandang, maka NPC musuh akan merespon dengan mengejar *player*. Akan tetapi jika kesehatan NPC musuh buruk maka state ini tidak aktif.

4. Menyerang

Player berada pada jangkauan serangan, maka NPC musuh akan menyerang player.

5. Menghindar

NPC musuh dapat menghindari jika jarak serangan di luar jangkauan serangan, akan tetapi tidak dapat menghindari jika jangkauan serangan sangat dekat.

6. Mati

NPC musuh kalah atau kesehatan sama dengan nol dan mati .

Peletakan logika *fuzzy* terdapat pada *state* “menyerang” yang terdiri dari tiga aksi yang dapat dilakukan oleh NPC musuh yaitu menyerang jarak dekat, menyerang jarak jauh dan menjauh. Aksi-aksi pada state ini akan dijalankan

berdasarkan logika *fuzzy*. *State* ini merupakan *state* yang memiliki logika *fuzzy* di dalamnya. *State* menyerang pada FSM ditunjukkan pada gambar 3.8.



Gambar 3.8 *State* Menyerang pada FSM NPC musuh

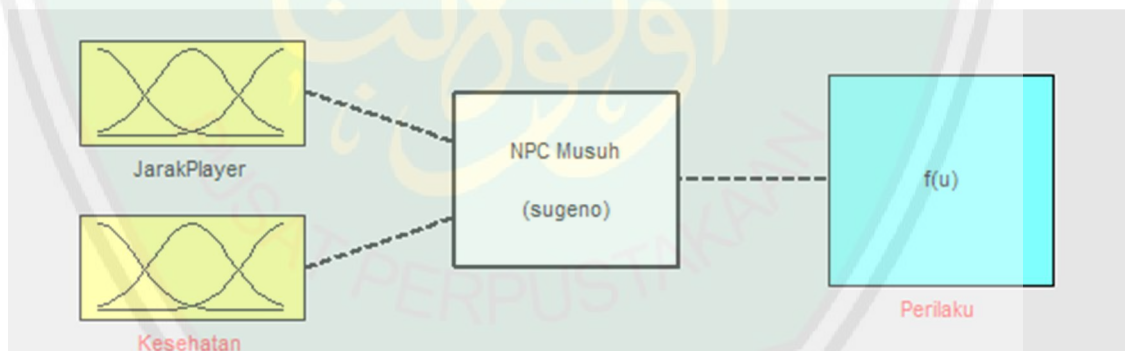
3.5 Desain *Fuzzy*

Logika *fuzzy* digunakan untuk menentukan perilaku NPC agar menjadi bervariasi. Metode *fuzzy* yang digunakan adalah *fuzzy* Sugeno. Pada rancangan *Finite State Machine*, logika *fuzzy* diletakkan pada *state* “menyerang”. Variabel yang digunakan sebagai variabel input ada dua yaitu jarak dan kesehatan. Kedua variabel tersebut akan menentukan perilaku NPC musuh. Sedangkan untuk variabel outputnya yaitu perilaku. Nilai dari perilaku akan ditentukan oleh input yang berupa jarak dan kesehatan.

Dari variabel-variabel yang digunakan, maka nilai linguistiknya dari masing-masing variabel tersebut adalah sebagai berikut :

1. Variabel Jarak dibagi menjadi lima yaitu : Sangat Dekat, Dekat, Menengah, Jauh dan Sangat Jauh.
2. Variabel Kesehatan dibagi menjadi empat yaitu : Buruk, Lemah, Sedang, Sehat
3. Variabel Perilaku dibagi menjadi tiga yaitu : Menyerang Jarak Dekat, Menyerang Jarak Jauh dan Menjauh

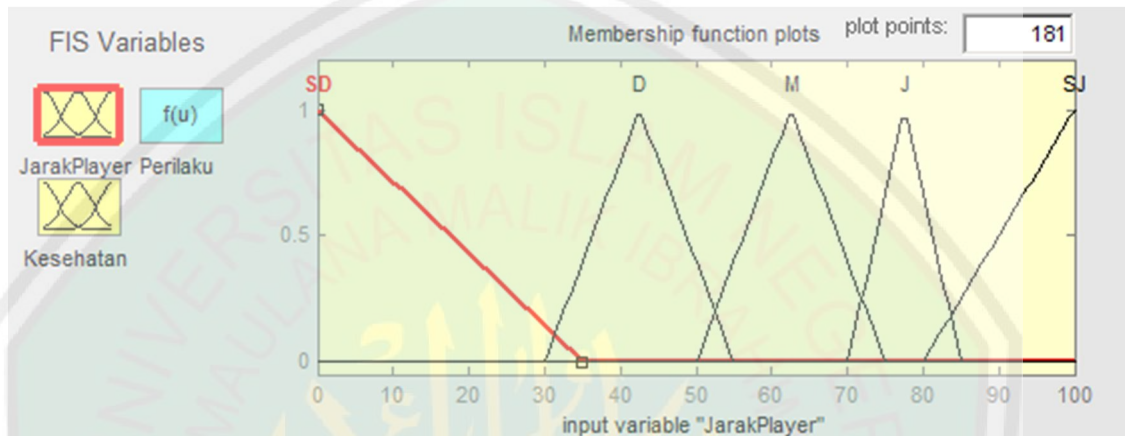
Desain fuzzy untuk menghasilkan perilaku yang bervariasi terdiri atribut yaitu jarak terhadap player dan kesehatan. Dari masing-masing atribut menggunakan gabungan fungsi keanggotaan linear turun, segitiga, dan naik. Kemudian nantinya akan menghasilkan output berupa perilaku. Seperti ditunjukkan pada gambar 3.9 berikut ini :



Gambar 3.9 Desain *Fuzzy* Sugeno NPC musuh

Berdasarkan desain fuzzy maka dapat dilakukan pemetaan himpunan fuzzy. Variabel “jarak player” mempunyai lima himpunan fuzzy yakni Sangat Dekat, Dekat, Menengah, Jauh dan Sangat Jauh. Derajat keanggotaan (*Membership Degree*) untuk variabel “jarak player” mempunyai nilai yang berada

pada interval 0 sampai 100. Nilai tersebut berupa bilangan bertipe *float*, sehingga untuk penentuan nilai rentan jarak diambil berdasarkan ukuran tinggi dan lebar dari karakter.



Gambar 3.10 Derajat keanggotaan variabel input jarak player

Fungsi keanggotaan mempunyai nilai variabel linguistik serta notasinya dan mempunyai interval yang berbeda. Nilai variabel linguistik berada pada rentang interval dari 0 sampai 100.

Tabel 3.2 Input variabel jarak player

Input variabel jarak player		
Variabel	Notasi	Nilai
SD	Sangat Dekat	0 – 35
D	Dekat	30 – 55
M	Menengah	50 – 75
J	Jauh	70 – 85
SJ	Sangat Jauh	80 – 100

Setiap nilai linguistik dari variabel “jarak player” mempunyai nilai fuzzyfikasi. Perhitungan nilai fuzzyfikasi didapatkan dari beberapa fungsi. Fungsi yang digunakan yaitu fungsi linear turun, segitiga dan linear naik. Berikut perhitungan manual dari fungsi tersebut :

$$\mu[x]_{SD} = \begin{cases} 0, & x \geq 35 \\ \frac{(35 - x)}{(35 - 0)}, & 0 < x < 35 \\ 1, & x \leq 0 \end{cases}$$

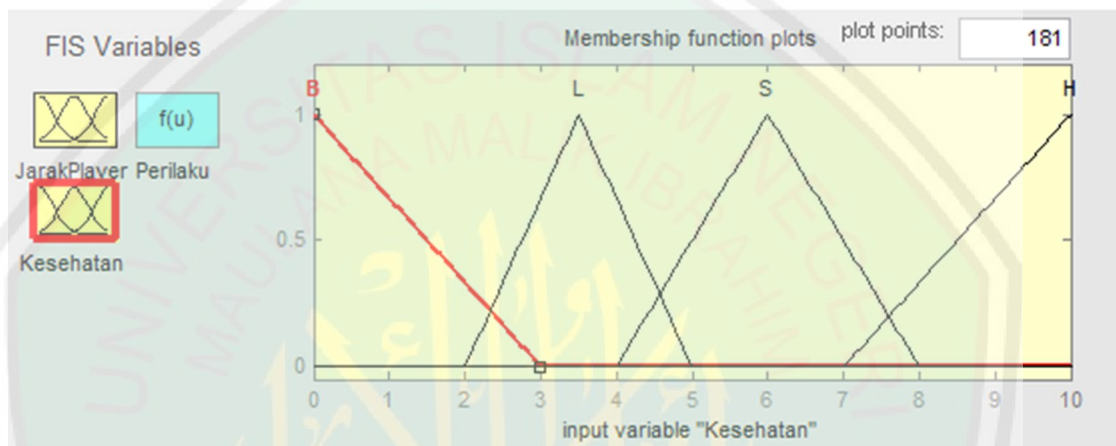
$$\mu[x]_D = \begin{cases} 0, & x \geq 55 \text{ atau } x \leq 30 \\ \frac{(x - 30)}{(42.5 - 30)}, & 30 < x \leq 42.5 \\ \frac{(55 - x)}{(55 - 42.5)}, & 42.5 < x < 55 \end{cases}$$

$$\mu[x]_M = \begin{cases} 0, & x \geq 75 \text{ atau } x \leq 50 \\ \frac{(x - 50)}{(62.5 - 50)}, & 50 < x \leq 62.5 \\ \frac{(75 - x)}{(75 - 62.5)}, & 62.5 < x < 75 \end{cases}$$

$$\mu[x]_J = \begin{cases} 0, & x \geq 85 \text{ atau } x \leq 70 \\ \frac{(x - 70)}{(77.5 - 70)}, & 70 < x \leq 77.5 \\ \frac{(85 - x)}{(85 - 77.5)}, & 77.5 < x < 85 \end{cases}$$

$$\mu[x]_{SJ} = \begin{cases} 0, & x \leq 80 \\ \frac{(x - 80)}{(100 - 80)}, & 80 < x < 100 \\ 1, & x \geq 100 \end{cases}$$

Variabel “kesehatan” mempunyai empat himpunan fuzzy yakni Buruk, Lemah, Sedang dan Sehat. Derajat keanggotaan (*Membership Degree*) untuk variabel “kesehatan” mempunyai nilai yang berada pada interval 0 sampai 10. Sama halnya dengan nilai variabel jarak yakni berupa bilangan bertipe *float*.



Gambar 3.11 Derajat keanggotaan variabel input kesehatan

Fungsi keanggotaan mempunyai nilai variabel linguistik serta notasinya dan mempunyai interval yang berbeda. Nilai variabel linguistik berada pada rentang interval dari 0 sampai 10.

Tabel 3.3 Input variabel kesehatan

Input variabel kesehatan		
Variabel	Notasi	Nilai
B	Buruk	0 – 3
L	Lemah	2 – 5
S	Sedang	4 – 8
H	Sehat	7 – 10

Setiap nilai linguistik dari variabel “jarak player” mempunyai nilai fuzzyfikasi. Perhitungan nilai fuzzyfikasi didapatkan dari beberapa fungsi. Fungsi yang digunakan yaitu fungsi linear turun, segitiga dan linear naik. Berikut perhitungan manual dari fungsi tersebut:

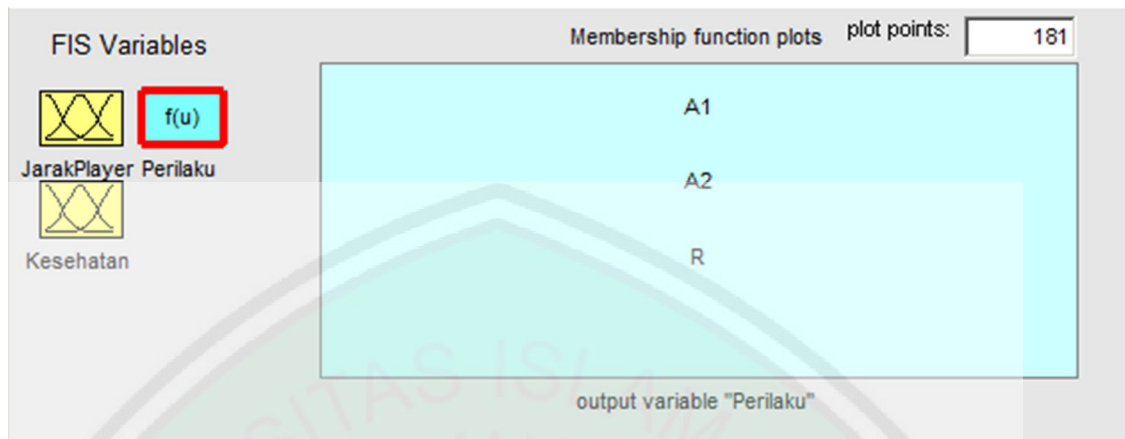
$$\mu[x]_B = \begin{cases} 0, & x \geq 3 \\ \frac{(3-x)}{(3-0)}, & 0 < x < 3 \\ 1, & x \leq 0 \end{cases}$$

$$\mu[x]_L = \begin{cases} 0, & x \geq 2 \text{ atau } x \leq 5 \\ \frac{(x-2)}{(3.5-2)}, & 2 < x \leq 3.5 \\ \frac{(5-x)}{(5-3.5)}, & 3.5 < x < 5 \end{cases}$$

$$\mu[x]_S = \begin{cases} 0, & x \geq 4 \text{ atau } x \leq 8 \\ \frac{(x-4)}{(6-4)}, & 4 < x \leq 6 \\ \frac{(8-x)}{(8-6)}, & 6 < x < 8 \end{cases}$$

$$\mu[x]_H = \begin{cases} 0, & x \leq 7 \\ \frac{(x-7)}{(10-7)}, & 7 < x < 10 \\ 1, & x \geq 10 \end{cases}$$

Untuk output dari variabel “perilaku” NPC Musuh, nilai linguistiknya dibagi menjadi yakni Menyerang Jarak Dekat, Menyerang Jarak Jauh dan Menjauh.



Gambar 3.12 Keanggotaan variabel output perilaku

Keanggotaan dari variabel “perilaku” mempunyai nilai linguistik serta notasinya. Nilai linguistik Menyerang Jarak Dekat bernilai 0, Menyerang Jarak Jauh bernilai 1 dan Menjauh bernilai 2. Nilai-nilai pada variabel “perilaku” bersifat konstan.

Tabel 3.4 Input variabel perilaku

Input variabel perilaku		
Variabel	Notasi	Nilai
A1	Menyerang Jarak Dekat	0
A2	Menyerang Jarak Jauh	1
R	Menjauh	2

Selanjutnya dari variabel input dan output maka akan disusun sejumlah aturan *fuzzy* untuk menentukan perilaku NPC musuh. Aturan *fuzzy* disusun berupa matrik berdasarkan nilai linguistic variabel jarak dan variabel kesehatan.

Tabel 3.5 Aturan-aturan fuzzy

	Jarak					
	Variabel	SD	D	M	J	SJ
Kesehatan	B	R	R	R	R	R
	L	R	R	R	R	A2
	S	R	A1	A2	A2	A2
	H	A1	A1	A2	A2	A2

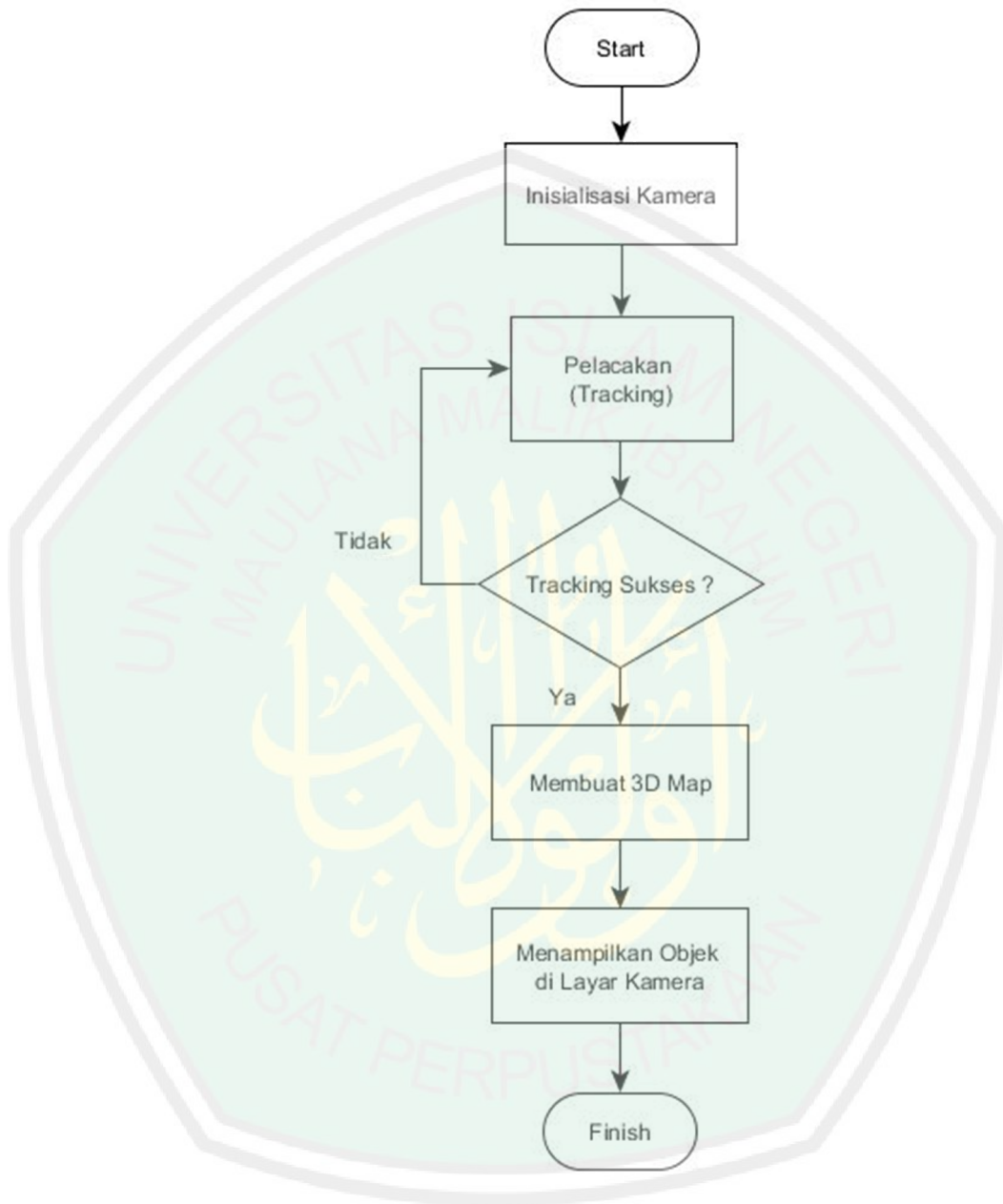
Dari tabel di atas maka terbentuklah aturan-aturan *fuzzy* yang diterapkan dalam penentuan perilaku NPC musuh. Aturan *fuzzy* yang terbentuk berjumlah 20 yang tersusun sebagai berikut :

1. IF (Jarak Player is SD) AND (Kesehatan is B) THEN (Perilaku is R)
2. IF (Jarak Player is SD) AND (Kesehatan is L) THEN (Perilaku is R)
3. IF (Jarak Player is SD) AND (Kesehatan is S) THEN (Perilaku is R)
4. IF (Jarak Player is SD) AND (Kesehatan is H) THEN (Perilaku is A1)
5. IF (Jarak Player is D) AND (Kesehatan is B) THEN (Perilaku is R)
6. IF (Jarak Player is D) AND (Kesehatan is L) THEN (Perilaku is R)
7. IF (Jarak Player is D) AND (Kesehatan is S) THEN (Perilaku is A1)
8. IF (Jarak Player is D) AND (Kesehatan is H) THEN (Perilaku is A1)
9. IF (Jarak Player is M) AND (Kesehatan is B) THEN (Perilaku is R)
10. IF (Jarak Player is M) AND (Kesehatan is L) THEN (Perilaku is R)
11. IF (Jarak Player is M) AND (Kesehatan is S) THEN (Perilaku is A2)

12. IF (Jarak Player is M) AND (Kesehatan is H) THEN (Perilaku is A2)
13. IF (Jarak Player is J) AND (Kesehatan is B) THEN (Perilaku is R)
14. IF (Jarak Player is J) AND (Kesehatan is L) THEN (Perilaku is R)
15. IF (Jarak Player is J) AND (Kesehatan is S) THEN (Perilaku is A2)
16. IF (Jarak Player is J) AND (Kesehatan is H) THEN (Perilaku is A2)
17. IF (Jarak Player is SJ) AND (Kesehatan is B) THEN (Perilaku is R)
18. IF (Jarak Player is SJ) AND (Kesehatan is L) THEN (Perilaku is A2)
19. IF (Jarak Player is SJ) AND (Kesehatan is S) THEN (Perilaku is A2)
20. IF (Jarak Player is SJ) AND (Kesehatan is SH) THEN (Perilaku is A2)

3.6 Penerapan 3D Instant Tracking

Konsep dari 3D tracking adalah membuat 3D map di sebuah *scene* (layar) dengan menggunakan suatu teknologi yang dinamakan SLAM (Simultaneous Localization And Mapping). Dengan menggunakan mode *3D Instant Tracking* maka dapat membentuk sebuah map *real time* sehingga tidak memerlukan marker untuk menampilkan objek-objek digital. Semua objek digital akan di-*mapping* ke dalam sebuah map 3D.



Gambar 3.13 Flowchart 3D Instant Tracking

Dalam *game* ini menggunakan *3D Instant Tracking* sebagai teknik atau metode untuk menampilkan objek dengan tanpa marker. Jadi, istilah markerless pada *game* ini adalah dengan membentuk sebuah 3D map yang akan digunakan untuk menampilkan objek pada *scene* (layar) secara realtime. Proses yang terjadi

saat melakukan pelacakan dengan *3D Instant Tracking* keseluruhannya diatur dalam *framework* Metaio SDK.



BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi

Pada Bab ini membahas tentang implementasi dan hasil pengujian metode yang digunakan pada game yang telah dibuat. Pengujian metode bertujuan untuk mengetahui apakah metode yang digunakan dapat berpengaruh pada NPC musuh dan menghasilkan perilaku yang bervariasi. Selain itu, mengukur seberapa baik metode yang digunakan dari sisi ketepatan dan keakuratan dalam mengontrol perilaku NPC musuh.

4.1.1 Peralatan yang digunakan

Peralatan yang akan digunakan dalam implementasi adalah perangkat dengan spesifikasi yang mendukung baik dari segi *hardware* maupun *software*. Berikut adalah spesifikasi yang akan digunakan :

Hardware :

1. Notebook Emachines Aces : Processor Intel Dual Core 2.2 GHz, RAM 2 GB, Hardisk 320 GB
2. Tablet Asus K016

Software :

1. Unity 3D 4.6.1
2. Blender 2.67
3. Metaio SDK 5
4. Android Jelly Bean 4.2

4.1.2 Implementasi *Finite State Machine*

Berdasarkan rancangan *Finite State Machine* yang dibuat sebelumnya, maka akan dilakukan pengujian terlebih dahulu pada NPC musuh. Untuk berpindah dari satu *state* ke *state* yang lain ditentukan berdasarkan parameter input berupa jarak, kesehatan dan aksi *player*. Nilai dari parameter jarak dan kesehatan bervariasi, sehingga dapat mengetahui *state* mana yang akan aktif terlebih dahulu sebelum berpindah ke *state* selanjutnya. *State* yang telah dirancang sebelumnya berjumlah 5 diantaranya idle, patrol, menghindar, mengejar, menyerang dan mati. Beberapa *state* tersebut diimplementasikan ke dalam sebuah variabel yang bertipe *enum*.

```
enum States {IDLE, PATROL, MENGEJAR, MENYERANG, MENGHINDAR,  
MATI};
```

Kemudian untuk melakukan transisi dari satu *state* ke *state* lainnya adalah mengaktifkan *state* sesuai dengan aksi yang diberikan dalam proses transisi. Aksi dalam transisi merupakan kondisi yang harus dipenuhi agar *state*

dapat berpindah ke *state* berikutnya. Berikut adalah *pseudo-code* untuk transisi

state :

```

class FiniteStateMachine:

//list of states
States : enum

//initial state
state : States

Start():
//initial state from idle
state = States.IDLE
End

UpdateState():
//calculate distance between player and enemy
distance = Vector3.Distance(player.position, enemy.position)

//apply transition
switch(state) :
    case States.PATROL:
        Patrol()
        break

    case States.MENGEJAR:
        Mengejar()
        break

    case States.MENYERANG:
        Menyerang()
        break

    case States.MENGHINDAR:
        Menghindar()
        break

    case States.MATI:
        Mati()
        break

    End
End

```

Berdasarkan *pseudo-code* tersebut, *state-state* yang ada pada variabel bertipe *enum* digunakan untuk sebagai kondisi aktif atau tidaknya sebuah *state*.

Jika kondisi state sama dengan PATROL, maka *state* yang aktif adalah *state* berkelana. Jika kondisi *state* sama dengan MENGEJAR, maka *state* yang aktif adalah *state* MENGEJAR. Begitu seterusnya untuk semua *state*.

Masing-masing terdiri dari beberapa fungsi yaitu **Patrol()**, **Mengejar()**, **Menyerang()**, **Menghindar()**, **Mati()**. Fungsi **Patrol()** merupakan fungsi untuk melakukan aksi berkelana atau patrol. Ketika *player* memasuki area *state* PATROL, maka NPC akan melakukan pergerakan secara acak. Jarak antara *player* dengan NPC musuh akan dilakukan pengecekan secara realtime untuk mendapatkan posisi antara *player* dan NPC musuh. Jika jarak antara posisi *player* dengan NPC musuh mencapai jarak minimum (jarak lebih kecil sama dengan 200) dari *range* state PATROL, maka *state* akan berpindah ke *state* MENGEJAR.

Selain itu, NPC musuh akan mengecek secara realtime apakah *player* melakukan aksi serangan berupa menembak, maka jarak antara *projectile* dengan NPC musuh akan dihitung sampai mencapai area dari *state* MENGHINDAR. Selanjutnya jika *projectile* mencapai area dari *state* MENGHINDAR, maka *state* akan berpindah ke *state* MENGHINDAR.

Berdasarkan penjelasan di atas, maka fungsi **Patrol()** sebagai berikut :

Patrol() :

```

//check with minimum distance
If distance <= 200
    state = States.MENGEJAR
End

If bullet != null
    //calculate distance between bullet and NPC
    distanceBullet = Vector3.Distance(bullet.position,
    enemy.position)

    //check distance the bullet
    If distanceBullet <= 60
        state = States.MENGHINDAR
    End
End
End

```

Selanjutnya, pada fungsi **Mengejar()** merupakan fungsi untuk melakukan aksi mengejar. Jarak antara *player* dengan NPC musuh akan dihitung secara realtime. Jika *player* memasuki area (jarak lebih kecil sama dengan 200) dari *state* MENGEJAR, maka *state* MENGEJAR akan aktif. Seterusnya akan dilakukan pengecekan posisi antara *player* dan NPC musuh, Jika jarak antara posisi *player* dengan NPC musuh mencapai jarak minimum (jarak lebih kecil sama dengan 100) dari *range* state MENGEJAR, maka *state* akan berpindah ke *state* MENYERANG. Sedangkan, jika jarak antara posisi *player* dengan NPC musuh mencapai jarak maksimum (jarak lebih besar dari 200) dari *range* state MENGEJAR, maka *state* akan berpindah ke *state* PATROL.

Berikut adalah implementasi fungsi **Mengejar()** dalam *pseudo-code* program.

```
Mengejar() :

If distance > 200
    state = States.PATROL
Else If distance <= 100
    state = States.MENYERANG
End
```

Selanjutnya, pada fungsi **Menyerang()** merupakan fungsi untuk melakukan aksi menyerang. Pada fungsi ini tempat mengimplementasikan logika *fuzzy* yang akan menghasilkan tiga aksi yang berbeda. Diantaranya ada aksi menyerang jarak jauh, menyerang jarak dekat dan menjauh. Jika output *fuzzy* yang dihasilkan sama dengan 0 maka akan melakukan aksi menyerang jarak dekat. Jika output *fuzzy* yang dihasilkan sama dengan 1 maka akan melakukan aksi menyerang jarak jauh. Sedangkan, jika output *fuzzy* yang dihasilkan sama dengan 2 maka akan melakukan aksi menjauh. Jika jarak antara posisi *player* dengan NPC musuh mencapai jarak maksimum (jarak lebih besar sama dengan 100) dari *range* state MENYERANG, maka *state* akan berpindah ke *state* MENGEJAR.

Berikut adalah implementasi fungsi **Menyerang()** berupa *pseudo-code* program.

```

Menyerang() :

//hold the output of fuzzy sugeno
fuzzy

//check the fuzzy output
If fuzzy.result() == 1
    Attack2()
Else If fuzzy.resul() == 2
    RunAway()
Else
    Attack1()
End

```

Selanjutnya, pada fungsi **Menghindar()** merupakan fungsi untuk melakukan aksi menghindar. Pada fungsi ini akan dilakukan pengecekan apakah ada tembakan dari *player*, jika ada tembakan maka akan melakukan pengecekan jarak *projectile* dengan NPC musuh. Jika *projectile* memasuki area *state* menghindar, maka NPC musuh akan melakukan aksi menghindar. Setelah itu maka NPC musuh akan berpindah ke *state* PATROL. Berikut adalah implementasi fungsi **Menghindar()** berupa *pseudo-code* program.

```

Menghindar() :

if bullet != null
    distanceBullet = Vector3.Distance(bullet.position,

```

```

    enemy.position)

    if distanceBullet > 60
        state = States.PATROL
    End
End

```

Terakhir adalah fungsi **Mati()** yaitu jika NPC musuh kalah atau mati.

```

Mati () :
state = States.MATI

```

4.1.3 Implementasi Logika *Fuzzy*

Pada pengujian ini dilakukan pengujian menggunakan logika *fuzzy* pada NPC musuh. Implementasi dari logika *fuzzy* sendiri diletakkan pada salah satu *state* di *Finite State Machine* yaitu *state* Menyerang. Hasil dari logika *fuzzy* akan menentukan variasi perilaku dari NPC musuh. Beberapa parameter yang diujikan diantaranya adalah jarak dan kesehatan. Nilai dari masing-masing variabel yang diuji dengan nilai bervariasi. Metode *fuzzy* yang digunakan adalah *Fuzzy Sugeno*, karena hasil output dari defuzzifikasi berupa konstanta yang dapat langsung menyatakan perilaku dari NPC musuh.

Dalam implementasi metode *Fuzzy Sugeno*, langkah pertama yang dilakukan adalah memasukkan variabel *fuzzy* yang sudah dimodelkan. Ada 3 variabel yang digunakan yaitu jarak, kesehatan dan perilaku. Dari variabel – variabel tersebut maka diimplementasikan masing-masing dalam bentuk *class*. Untuk variabel jarak adalah sebagai berikut :

```

class Jarak:

// deklarasi variabel
nilai_jarak : float
SD_a : float
SD_b : float
D_a : float
D_b : float
D_c : float
M_a : float
M_b : float
M_c : float
J_a : float
J_b : float
J_c : float
SJ_a : float
SJ_b : float

// inisialisasi nilai
SD_a = 0
SD_b = 35
D_a = 30
D_b = 42.5f
D_c = 55
M_a = 50
M_b = 62.5f
M_c = 75
J_a = 70
J_b = 77.5f
J_c = 85
SJ_a = 80
SJ_b = 100

MiuSangatDekat() :
    If nilai_jarak <= SD_a
        return 1
    Else If nilai_jarak > SD_a && nilai_jarak < SD_b
        return (SD_b - nilai_jarak) / (SD_b - SD_a)
    Else
        return 0

MiuDekat() :
    If nilai_jarak > D_b && nilai_jarak < D_c
        return (D_c - nilai_jarak) / (D_c - D_b)
    Else If nilai_jarak > D_a && nilai_jarak <= D_b
        return (nilai_jarak - D_a) / (D_b - D_a)
    Else
        return 0

```

```

MiuMenengah() :
    If nilai_jarak > M_b && nilai_jarak < M_c
        return (M_c - nilai_jarak) / (M_c - M_b)
    Else If nilai_jarak > M_a && nilai_jarak <= M_b
        return (nilai_jarak - M_a) / (M_b - M_a)
    Else
        return 0

MiuJauh() :
    If nilai_jarak > J_b && nilai_jarak <= J_c
        return (J_c - nilai_jarak) / (J_c - J_b)
    Else If nilai_jarak > J_a && nilai_jarak <= J_b
        return (nilai_jarak - J_a) / (J_b - J_a)
    Else
        return 0

MiuSangatJauh() :
    if nilai_jarak >= SJ_b
        return 1
    Else If nilai_jarak > SJ_a && nilai_jarak < SJ_b
        return (nilai_jarak - SJ_a) / (SJ_b - SJ_a)
    Else
        return 0

```

Berdasarkan *pseudo-code* di atas, yang pertama kali dilakukan adalah menginisialisasi nilai keanggotaan dari himpunan fuzzy variabel jarak. Kemudian perhitungan nilai fuzzyfikasi berdasarkan fungsi keanggotaan yang dimiliki. Dari hasil fuzzyfikasi nantinya akan didapatkan nilai keanggotaan fuzzy sesuai dengan nilai jarak yang diinputkan.

Kemudian untuk variabel kesehatan adalah sebagai berikut :

```

class Kesehatan :
// deklarasi variabel
nilai_kesehatan : float
B_a : float
B_b : float
L_a : float
L_b : float
L_c : float
S_a : float
S_b : float
S_c : float

```

```

H_a : float
H_b : float
H_c : float

// inisialisasi
B_a = 0
B_b = 3
L_a = 2
L_b = 3.5f
L_c = 5
S_a = 4
S_b = 6
S_c = 8
H_a = 7
H_b = 10

MiuBuruk() :
    If nilai_kesehatan<= B_a
        return 1
    Else If nilai_kesehatan> B_a && nilai_kesehatan< B_b
        return (B_b - nilai_kesehatan) / (B_b - B_a)
    Else
        return 0

MiuLemah() :
    If nilai_kesehatan> L_b && nilai_kesehatan< L_c
        return (L_c - nilai_kesehatan) / (L_c - L_b)
    Else If nilai_kesehatan> L_a && nilai_kesehatan<= L_b
        return (nilai_kesehatan- L_a) / (L_b - L_a)
    Else
        return 0

MiuSedang() :
    If nilai_kesehatan> S_b && nilai_kesehatan< S_c
        return (S_c - nilai_kesehatan) / (S_c - S_b)
    Else If nilai_kesehatan> S_a && nilai_kesehatan<= S_b
        return (nilai_kesehatan - S_a) / (S_b - S_a)
    Else
        return 0

MiuSehat() :
    If nilai_kesehatan >= H_b
        return 1;
    Else If nilai_kesehatan > H_a && nilai_kesehatan < H_b
        return (nilai_kesehatan - H_a) / (H_b - H_a)
    Else
        return 0

```

Berdasarkan *pseudo-code* di atas sama halnya dengan variabel jarak yaitu menginisialisasi nilai keanggotaan. Kemudian perhitungan nilai fuzzyfikasi berdasarkan fungsi keanggotaan yang dimiliki. Variabel yang terakhir adalah variabel perilaku sebagai berikut :

```
class Perilaku :
    variabel = [1...n]
    A1 : float
    A2 : float
    R : float
    A1 : 0 // Menyerang Jarak Dekat (A1)
    A2 : 1 // Menyerang Jarak Jauh (A2)
    R : 2 // Menjauh (R)
    variabel[0] = A1
    variabel[1] = A2
    variabel[2] = R
```

Nilai keanggotaan dari variabel perilaku berupa nilai konstanta yang berjumlah 3 yaitu Menyerang Jarak Dekat (A1) bernilai 0, Menyerang Jarak Jauh (A2) bernilai 1 dan Menjauh (R) bernilai 2.

Langkah selanjutnya dalam implementasi adalah mengaplikasikan operator logika fuzzy. Dengan melakukan operasi logika maka kita dapat mengkombinasikan dua atau lebih himpunan fuzzy. Nilai keanggotaan hasil operasi logika dari dua atau lebih himpunan fuzzy disebut *fire strength* atau α -predikat. Dalam metode *Fuzzy Sugeno* menggunakan operator AND (Intersection) untuk mendapatkan irisan dari dua himpunan fuzzy yang minimum. Rumus untuk mencari nilai minimum dari dua himpunan fuzzy adalah :

$$\mu_{A \cap B} [x] = \min(\mu_A[x], \mu_B[x])$$

Langkah selanjutnya yaitu mengaplikasikan fungsi implikasi. Bentuk umum aturan yang digunakan dalam fungsi implikasi adalah :

IF x is A THEN y is B

dengan x dan y adalah skalar, A dan B adalah himpunan fuzzy. Semua aturan menggunakan fungsi implikasi MIN (minimum). Kemudian kita mencari *fire strength* atau α -predikat untuk semua aturan yang sudah dibuat.

Berikut adalah implementasi untuk mencari nilai *fire strength* atau α -predikat dengan memasukkan sejumlah aturan.

```
ApplyRules() :
alfa_predikat[0] = Min(Miu_Jarak_SD, Miu_Kesehatan_B)
z_rule[0] = perilaku.variabel[2];

alfa_predikat[1] = Min(Miu_Jarak_SD, Miu_Kesehatan_L)
z_rule[1] = perilaku.variabel[2]

alfa_predikat[2] = Min(Miu_Jarak_SD, Miu_Kesehatan_S)
z_rule[2] = perilaku.variabel[2]

alfa_predikat[3] = Min(Miu_Jarak_SD, Miu_Kesehatan_H)
z_rule[3] = perilaku.variabel[0]

alfa_predikat[4] = Min(Miu_Jarak_D, Miu_Kesehatan_B)
z_rule[4] = perilaku.variabel[2]

alfa_predikat[5] = Min(Miu_Jarak_D, Miu_Kesehatan_L)
z_rule[5] = perilaku.variabel[2]
```

```

alfa_predikat[6] = Min(Miu_Jarak_D, Miu_Kesehatan_S)
z_rule[6] = perilaku.variabel[0]

alfa_predikat[7] = Min(Miu_Jarak_D, Miu_Kesehatan_H)
z_rule[7] = perilaku.variabel[0]

alfa_predikat[8] = Min(Miu_Jarak_M, Miu_Kesehatan_B)
z_rule[8] = perilaku.variabel[2]

alfa_predikat[9] = Min(Miu_Jarak_M, Miu_Kesehatan_L)
z_rule[9] = perilaku.variabel[2]

alfa_predikat[10] = Min(Miu_Jarak_M, Miu_Kesehatan_S)
z_rule[10] = perilaku.variabel[1]

alfa_predikat[11] = Min(Miu_Jarak_M, Miu_Kesehatan_H)
z_rule[11] = perilaku.variabel[1]

alfa_predikat[12] = Min(Miu_Jarak_J, Miu_Kesehatan_B)
z_rule[12] = perilaku.variabel[2]

alfa_predikat[13] = Min(Miu_Jarak_J, Miu_Kesehatan_L)
z_rule[13] = perilaku.variabel[2]

alfa_predikat[14] = Min(Miu_Jarak_J, Miu_Kesehatan_S)
z_rule[14] = perilaku.variabel[1]

alfa_predikat[15] = Min(Miu_Jarak_J, Miu_Kesehatan_H)
z_rule[15] = perilaku.variabel[1]

alfa_predikat[16] = Min(Miu_Jarak_SJ, Miu_Kesehatan_B)
z_rule[16] = perilaku.variabel[2]

alfa_predikat[17] = Min(Miu_Jarak_SJ, Miu_Kesehatan_L)
z_rule[17] = perilaku.variabel[1]

alfa_predikat[18] = Min(Miu_Jarak_SJ, Miu_Kesehatan_S)
z_rule[18] = perilaku.variabel[1]

alfa_predikat[19] = Min(Miu_Jarak_SJ, Miu_Kesehatan_H)
z_rule[19] = perilaku.variabel[1]

```

Langkah berikutnya adalah melakukan defuzzyfikasi dengan menggunakan metode centroid. Hasil output z akan berupa nilai konstanta tegas (crisp) sebagai berikut :

```
ComputeOutput() :
```

```
temp_1 : float
temp_2 : float

temp_1 = 0;
temp_2 = 0;

for i = 0 to z_rule.length do
temp_1 += alfa_predikat[i] * z_rule[i];
temp_2 += alfa_predikat[i]

z = (temp_1 / temp_2)
return z
```

4.1.4 Pengujian Logika *Fuzzy*

Pengujian dilakukan untuk menghasilkan nilai output perilaku yang bervariasi dari NPC Musuh. Parameter yang diujikan terdiri dari variabel jarak dan kesehatan NPC Musuh. Untuk pengujian ini mengambil beberapa nilai variabel jarak dan kesehatan dari nilai minimum sampai maksimum.

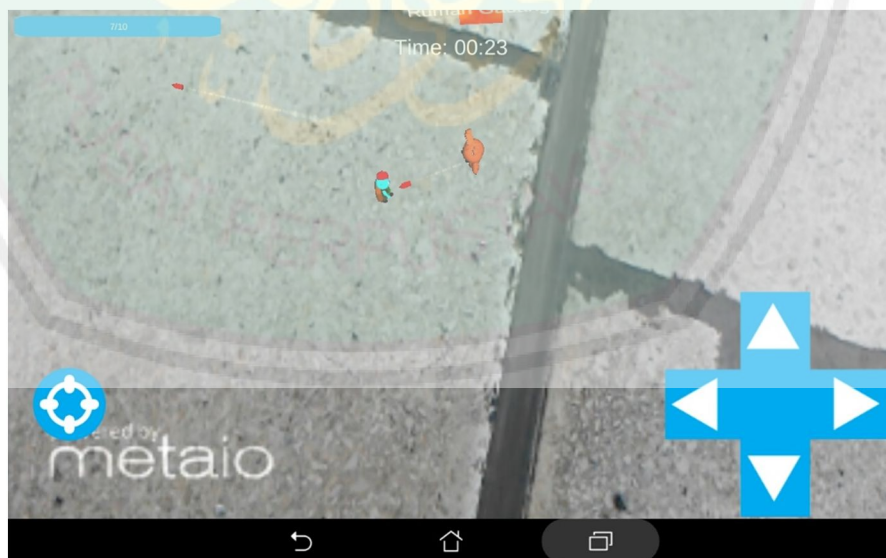
Tabel 4.1 Pengujian Perilaku NPC musuh dengan nilai jarak dan kesehatan

Kategori	Kesehatan		Jarak	
	0	1	0	1
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0
16	0	0	0	0
17	0	0	0	0
18	0	0	0	0
19	0	0	0	0
20	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0
26	0	0	0	0
27	0	0	0	0
28	0	0	0	0
29	0	0	0	0
30	0	0	0	0
31	0	0	0	0
32	0	0	0	0
33	0	0	0	0
34	0	0	0	0
35	0	0	0	0
36	0	0	0	0
37	0	0	0	0
38	0	0	0	0
39	0	0	0	0
40	0	0	0	0
41	0	0	0	0
42	0	0	0	0
43	0	0	0	0
44	0	0	0	0
45	0	0	0	0
46	0	0	0	0
47	0	0	0	0
48	0	0	0	0
49	0	0	0	0
50	0	0	0	0
51	0	0	0	0
52	0	0	0	0
53	0	0	0	0
54	0	0	0	0
55	0	0	0	0
56	0	0	0	0
57	0	0	0	0
58	0	0	0	0
59	0	0	0	0
60	0	0	0	0
61	0	0	0	0
62	0	0	0	0
63	0	0	0	0
64	0	0	0	0
65	0	0	0	0
66	0	0	0	0
67	0	0	0	0
68	0	0	0	0
69	0	0	0	0
70	0	0	0	0
71	0	0	0	0
72	0	0	0	0
73	0	0	0	0
74	0	0	0	0
75	0	0	0	0
76	0	0	0	0
77	0	0	0	0
78	0	0	0	0
79	0	0	0	0
80	0	0	0	0
81	0	0	0	0
82	0	0	0	0
83	0	0	0	0
84	0	0	0	0
85	0	0	0	0
86	0	0	0	0
87	0	0	0	0
88	0	0	0	0
89	0	0	0	0
90	0	0	0	0
91	0	0	0	0
92	0	0	0	0
93	0	0	0	0
94	0	0	0	0
95	0	0	0	0
96	0	0	0	0
97	0	0	0	0
98	0	0	0	0
99	0	0	0	0
100	0	0	0	0

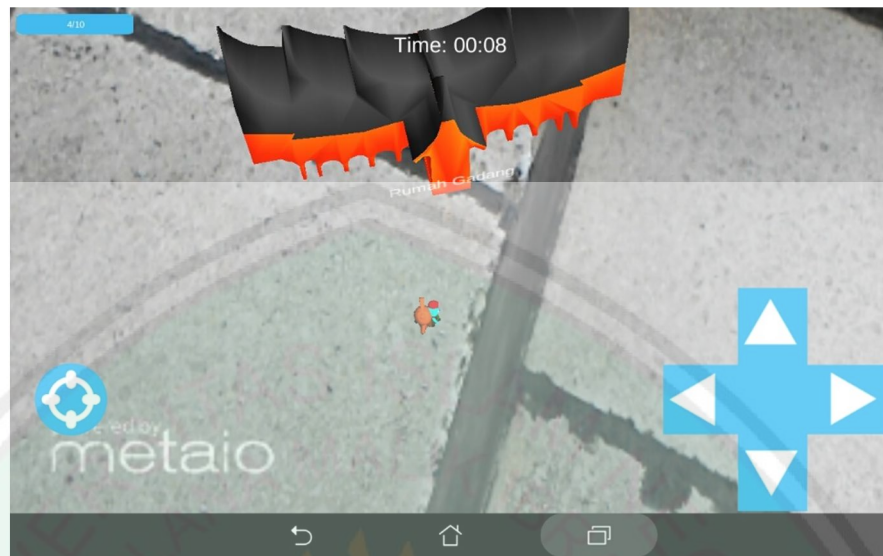


Berdasarkan tabel pengujian logika *fuzzy* menunjukkan variasi perilaku yang dihasilkan dari input parameter berupa jarak dan kesehatan. Perilaku yang dihasilkan berupa 3 model perilaku yaitu menyerang jarak jauh bernilai 1, menyerang jarak dekat bernilai 0 dan menjauh bernilai 2. Dari data pengujian tabel di atas bahwa perilaku yang dihasilkan mempunyai tingkat error sebesar 4,4 %. Data yang error akan dimasukkan ke dalam aksi yang ditentukan agar tidak terjadi *blank* respon terhadap *player*. Error yang bernilai kurang dari 0,5 akan dimasukkan ke kategori perilaku menyerang jarak dekat, error yang bernilai 0,6 – 1,5 akan dimasukkan ke kategori perilaku menyerang jarak jauh, sedangkan error yang bernilai lebih dari 1,5 akan dimasukkan ke kategori menjauh.

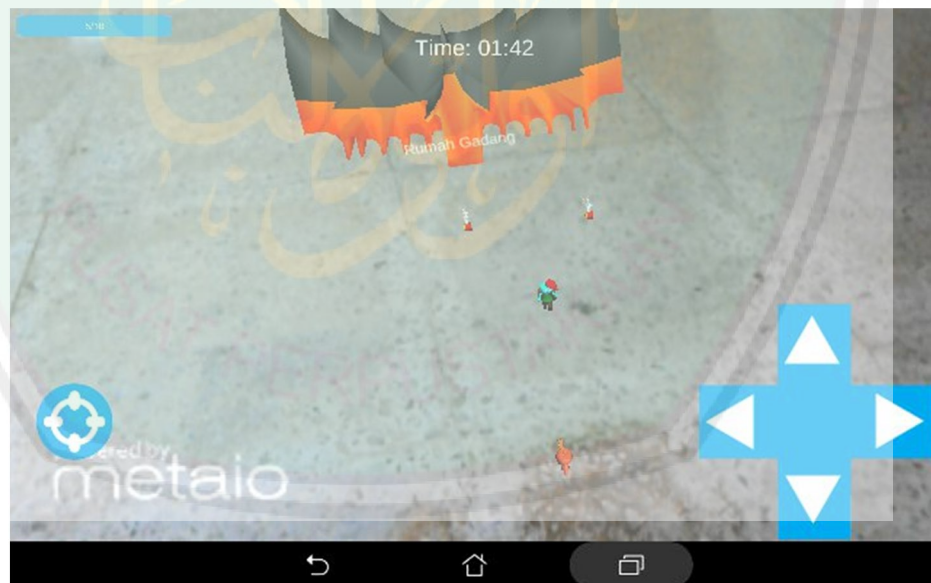
Masing-masing perilaku dapat dilihat pada gambar 4.1, 4.2 dan 4.3



Gambar 4.1 Perilaku NPC musuh ketika menyerang jarak jauh



Gambar 4.2 Perilaku NPC musuh ketika menyerang jarak dekat



Gambar 4.3 Perilaku NPC musuh ketika menjauh

4.1.5 Penerapan 3D Instant Tracking pada Metaio SDK

Dalam implementasinya cukup memanggil sebuah fungsi yang ada di dalam Metaio SDK sebagai berikut :

```
metaioSDK.startInstantTracking("INSTANT_3D");
```

Ketika proses *instant tracking* berjalan sukses maupun gagal, maka kita akan mendapat pesan callback melalui fungsi MetaioCallback. Fungsi callback di-*override* ke dalam game ini yaitu :

- *onInstantTrackingEvent(string)* merupakan transisi *triggers* dari state “Loading Configuration”
- *onTrackingEvent(string)* merupakan *report* dari tracking, transisi *triggers* antara state ”Running” dan “Paused”.

Berikut adalah implementasinya dalam *listing* program :

```
override protected void onTrackingEvent(String trackingEvent) {
    Debug.Log(trackingEvent);
    if (trackingEvent == "ETS_FOUND"){
        onRestoredTracking();
    } else {
        onLostTracking();
    }
}
```

```
    }

    override protected void onInstantTrackingEvent(string filepath)
    {
        lostTracking = true;
        lostTrackingTime = 0;

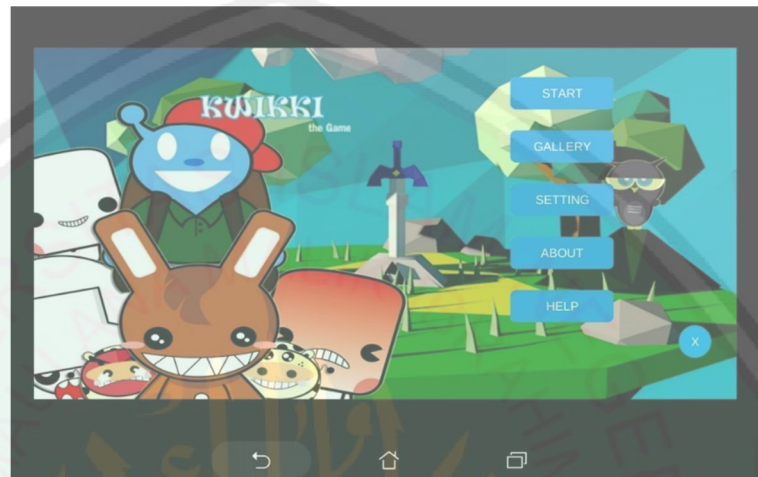
        // if succeeded, set new tracking configuration
        if (filepath.Length > 0){
            onPrepareGame();
            int result =
metaioSDK.SetTrackingConfiguration(filepath,1);
        } else {
            gameState = States.INITIAL;
        }
    }
}
```

4.2 Hasil Akhir

Setelah melakukan serangkaian tahapan penelitian dan uji coba, game ini dapat diselesaikan dengan baik. Hasil dapat dilihat dari gambar berikut ini.

1. Tampilan Menu Awal

Tampilan ini merupakan tampilan menu awal pada game.



Gambar 4.4 Tampilan Menu Awal

2. Tampilan Menu Pilihan Stage

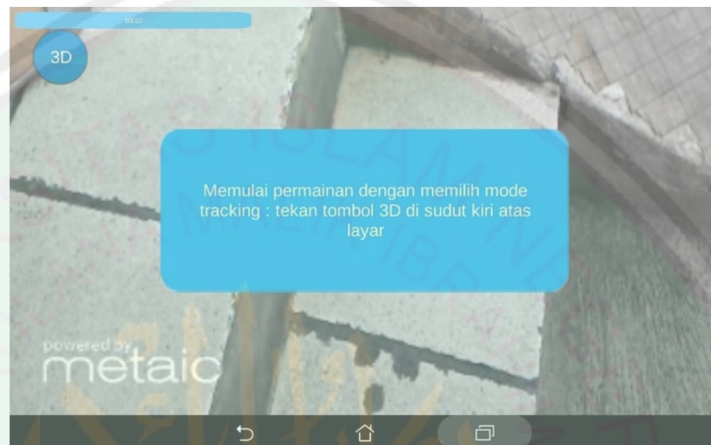
Tampilan ini merupakan tampilan untuk memilih stage yang akan dimainkan.



Gambar 4.5 Tampilan Menu Stage

3. Tampilan Layar melalui Kamera

Tampilan ini merupakan tampilan awal ketika akan memulai layar permainan. Layar akan disajikan melalui kamera perangkat.



Gambar 4.6 Tampilan Layar melalui Kamera

4. Tampilan Proses Tracking

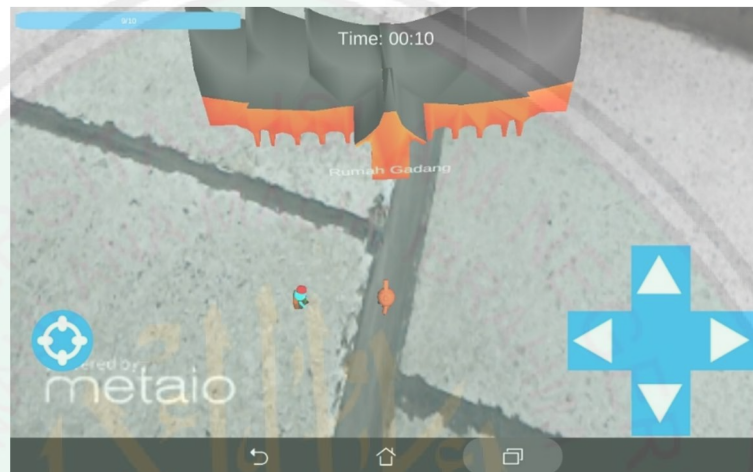
Tampilan ini merupakan tampilan ketika melakukan proses tracking pada permukaan objek dengan sedikit menggeser perangkat secara perlahan.



Gambar 4.7 Tampilan Proses Tracking

5. Tampilan Stage 1

Tampilan ini merupakan tampilan stage 1 yang menampilkan latar belakang rumah adat di Sumatra Barat.



Gambar 4.8 Tampilan Stage 1

6. Tampilan Stage 2

Tampilan ini merupakan tampilan stage 2 yang menampilkan latar belakang rumah adat di Daerah Istimewa Yogyakarta.



Gambar 4.9 Tampilan Stage 2

7. Tampilan Stage 3

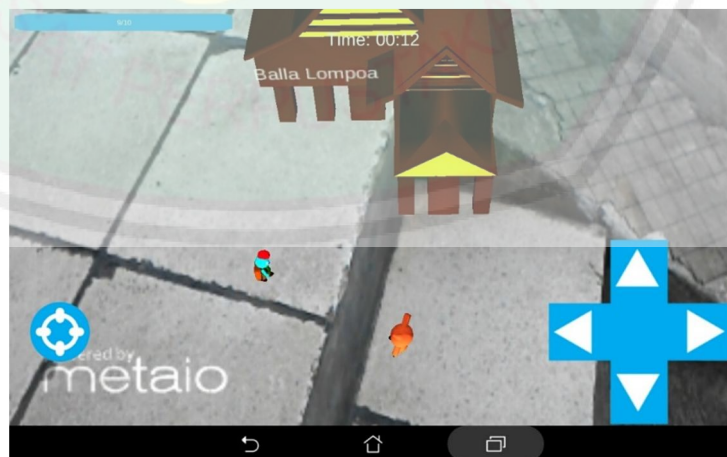
Tampilan ini merupakan tampilan stage 3 yang menampilkan latar belakang rumah adat di Kalimantan Selatan.



Gambar 4.10 Tampilan Stage 3

8. Tampilan Stage 4

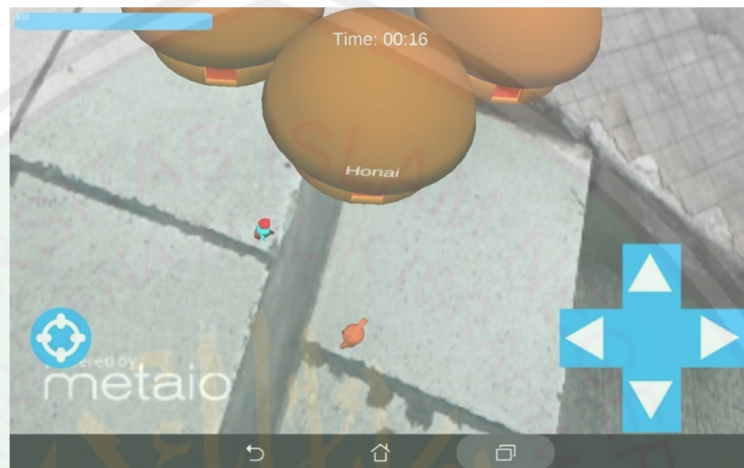
Tampilan ini merupakan tampilan stage 4 yang menampilkan latar belakang rumah adat di Sulawesi Selatan.



Gambar 4.11 Tampilan Stage 4

9. Tampilan Stage 5

Tampilan ini merupakan tampilan stage 5 yang menampilkan latar belakang rumah adat di Papua.



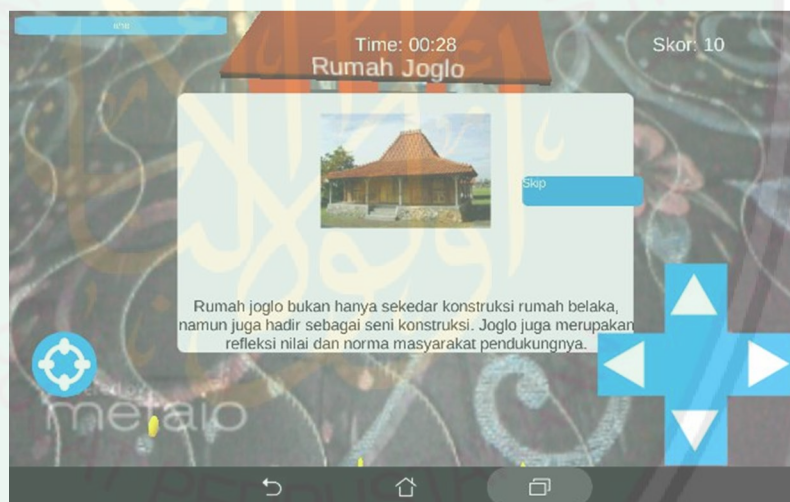
Gambar 4.12 Tampilan Stage 5

10. Tampilan Informasi Rumah Adat

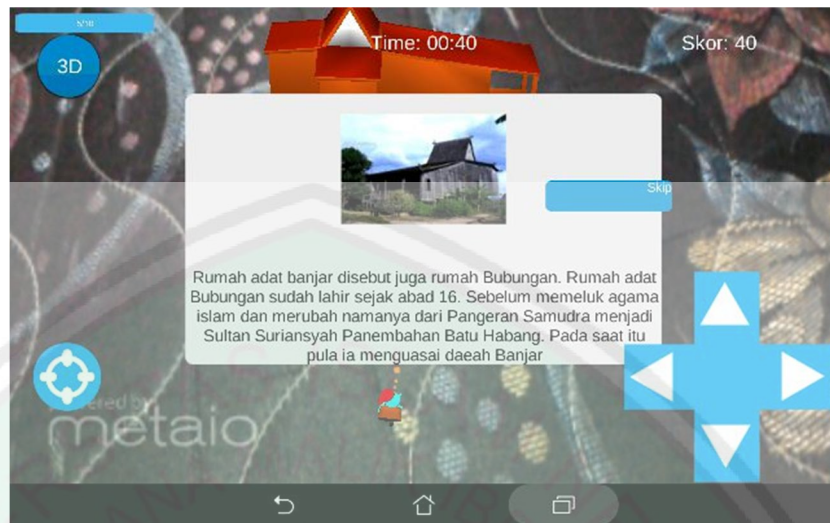
Tampilan ini merupakan tampilan informasi seputar rumah adat di salah satu *stage* tertentu. Ketika berhasil mengalahkan musuh maka akan muncul informasi misalnya karakteristik, bentuk atau filosofi dari rumah adat. Berikut ada beberapa tampilan informasi rumah adat.



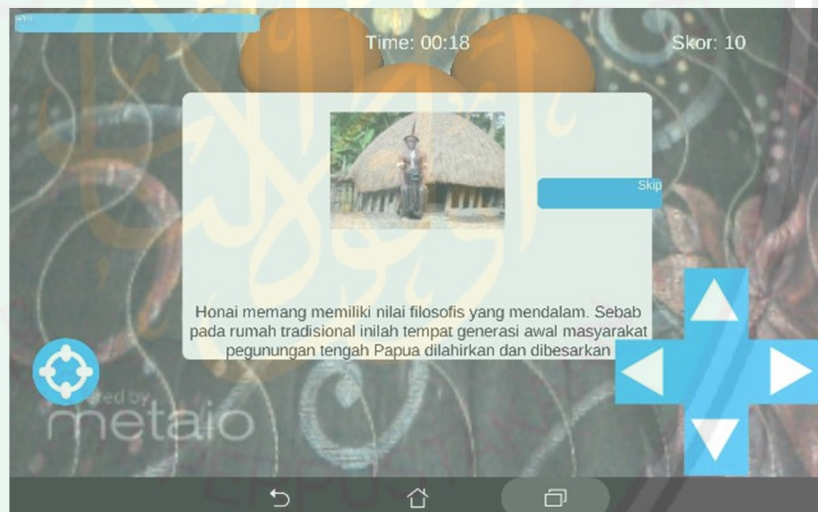
Gambar 4.13 Tampilan Informasi Rumah Gadang



Gambar 4.14 Tampilan Informasi Rumah Joglo



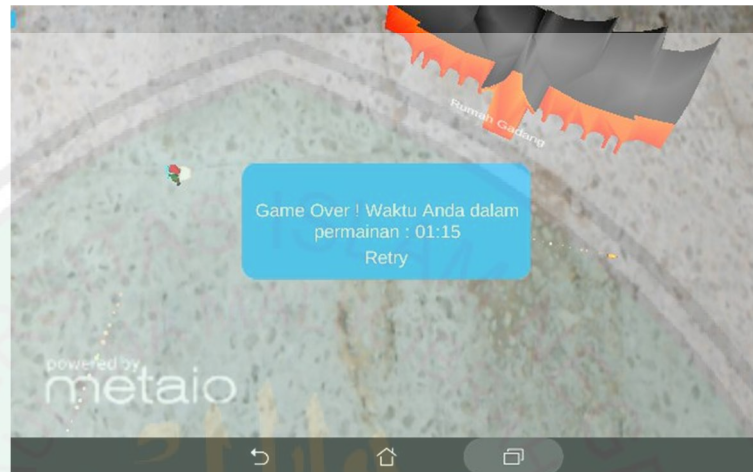
Gambar 4.15 Tampilan Informasi Rumah Banjar



Gambar 4.16 Tampilan Informasi Rumah Honai

11. Tampilan Game Over

Tampilan ketika game over.



Gambar 4.17 Tampilan Game Over

4.3 Integrasi Aplikasi dengan Islam

Ayat-ayat Al-Quran banyak memberikan dorongan kepada umat manusia bagi pengembangan kebudayaan. Ada sejumlah prinsip dasar yang terkandung di dalam Alquran dan hadis, sehingga umat Islam dapat mengembangkan kebudayaan secara maksimal. Islam menempatkan akal fikiran dalam posisi yang tinggi, seperti yang tercantum dalam surat Ali Imran ayat 190-191 :

إِنَّ فِي خَلْقِ السَّمَوَاتِ وَالْأَرْضِ وَاخْتِلَافِ اللَّيْلِ وَالنَّهَارِ لَآيَاتٍ لِّأُولِي الْأَلْبَابِ ۝ ١٩٠ الَّذِينَ يَذْكُرُونَ اللَّهَ قِيَمًا وَقُعُودًا وَعَلَىٰ جُنُوبِهِمْ وَيَتَفَكَّرُونَ فِي خَلْقِ السَّمَوَاتِ وَالْأَرْضِ رَبَّنَا مَا خَلَقْتَ هَذَا بَطْلًا سُبْحَانَكَ فَقِنَا عَذَابَ النَّارِ ۝ ١٩١

Artinya: “Sesungguhnya dalam penciptaan langit dan bumi, dan silih bergantinya malam dan siang benar-benar terdapat tanda-tanda bagi orang-orang yang berakal, (yaitu) orang-orang yang mengingat Allah sambil berdiri, duduk atau dalam keadaan berbaring dan mereka memikirkan tentang penciptaan langit dan bumi (seraya berkata): “Ya Tuhan kami, tiadalah Engkau menciptakan ini dengan sia-sia. Maha Suci Engkau, maka peliharalah kami dari siksa neraka.” (Q.S.Ali Imran:190,191)

Menurut Tafsir Jalalain Jilid 1 halaman 308-309, ayat di atas dapat ditafsirkan : “Sesungguhnya di dalam penciptaan langit dan bumi” berikut keajaiban-keajaiban yang ada di dalamnya “dan silih bergantinya malam dan siang” dengan datang dan pergi, bertambah dan berkurang “benar-benar terdapat tanda-tanda” bukti-bukti yang menunjukkan kekuasaan Allah “bagi orang-orang yang berakal”. Yakni orang-orang yang memiliki akal sehat.

“Yaitu orang-orang yang” ini adalah na’at (sifat) bagi apa yang disebutkan sebelumnya atau menjadi badal (keterangan pengganti) “mengingat Allah sambil berdiri, duduk atau berbaring” merebah, maksudnya dalam segala keadaan dan ada riwayat dari Ibnu Abbas yang menyebutkan : “Mereka melaksanakan shalat seperti itu menurut kemampuan” “dan mereka memikirkan tentang penciptaan langit dan bumi” agar bisa mereka jadikan sebagai petunjuk atas kekuasaan penciptanya seraya berkata “Ya Tuhan kami, Engkau tidak menciptakan ini”, yakni makhluk yang kami lihat ini “dengan sia-sia. “Ini berkedudukan sebagai haal (keterangan keadaan). Maksudnya dengan main-main. Tetapi merupakan

bukti yang menunjukkan kesempurnaan kekuasaanMu. “Maha Suci Engkau.” Untuk mensucikanMu dari tindakan main-main (sia-sia). “Maka periharalah kami dari siksa Neraka.”

Tentang kejadian langit dan bumi dan pertikaian (tidak sama) malam dan siang, menjadi bukti atas kekuasaan Allah bagi orang-orang yang berakal, yakni orang-orang yang selalu ingat akan Allah, baik diwaktu berdiri, duduk atau berbaring, serta memikirkan kejadian langit dan bumi. Mereka mengaku, bahwa semuanya itu dijadikan Allah, bukanlah dengan percuma, melainkan mengandung rahasia-rahasia yang ajaib, sebagai bukti bahwa yang menjadikannya dan yang mengaturnya ialah Allah Yang Maha Kuasa.

Manusia dikaruniakan akal budi oleh Allah SWT. Dengan akal budi manusia mampu memikirkan konsep-konsep maupun menyusun prinsip-prinsip umum yang diikhtiarkan dari berbagai pengamatan dan percobaan. Dengan akal budinya penciptaan alam semesta seluruhnya dan ciptaan kekuasaan-Nya. Allah telah memberikan dorongan kepada manusia untuk memikirkan alam semesta, mengadakan pengamatan terhadap berbagai gejala alam, merenungkan keindahan ciptaan-Nya dan mengungkapkan hukum-hukum-Nya di alam semesta ini. Seruan untuk mengadakan tinjauan, pemikiran, penelitian dan pembahasan ilmiah dapat ditemukan dalam berbagai ayat dalam Al-Quran. Ayat yang membahas hal tersebut ada pada surat Al-Ankabut ayat 20, Yunus ayat 101 dan Al-Hajj ayat 46.

فَلَّ سِيرُوا فِي الْأَرْضِ فَانظُرُوا كَيْفَ بَدَأَ الْخَلْقَ ثُمَّ اللَّهُ يُنشِئُ النَّشْأَةَ الْآخِرَةَ إِنَّ اللَّهَ

عَلَىٰ كُلِّ شَيْءٍ قَدِيرٌ ٢٠

Artinya : “Katakanlah : ‘Berjalanlah kamu di muka bumi, lalu perhatikanlah bagaimana Allah menciptakan makhluk dari awal kejadian, kemudian Allah membangkitkannya untuk terakhir kali. Sesungguhnya Allah Maha Kuasa atas segala sesuatu.’” (Q.S.Al-Ankabut:29,20)

Menurut Tafsir Jalalain Jilid 2 halaman 797, ayat di atas dapat ditafsirkan : “Katakanlah : ‘Berjalanlah kamu di muka bumi, lalu perhatikanlah bagaimana Allah menciptakan makhluk dari awal” bagi orang-orang yang ada sebelum kamu dan mematikan mereka, “kemudian Allah membangkitkannya untuk yang terakhir kali.” “Sesungguhnya Allah Maha Kuasa atas segala sesuatu.” Termasuk memulai penciptaan dan mengembalikannya.

فَلَّ أَنْظُرُوا مَاذَا فِي السَّمَوَاتِ وَالْأَرْضِ وَمَا نُعْتِي الْآيَاتِ وَالنُّذُرِ عَن قَوْمٍ لَّا يُؤْمِنُونَ

١٠١

Artinya : “Katakanlah : ‘Perhatikanlah apa yang ada di langit dan di bumi. Dan tidaklah tanda-tanda dan Rasul-Rasul yang memberi peringatan itu berguna bagi orang-orang yang tidak beriman.’” (Q.S.Yunus:10,101)

Menurut Tafsir Jalalain Jilid 2 halaman 64, ayat di atas dapat ditafsirkan : “Katakanlah” kepada orang-orang kafir Makkah : “Perhatikanlah apa”, yakni

segala sesuatu “yang ada di langit dan di bumi.” Yaitu ayat-ayat (tanda-tanda) yang menunjukkan keesaan Allah. “ Dan tidaklah tanda-tanda dan para pemberi peringatan itu cukup” maksudnya para Rasul “bagi orang-orang yang tidak beriman” dalam pengetahuan Allah. Maksudnya hal itu tidak ada gunanya bagi mereka.

أَفَلَمْ يَسِيرُوا فِي الْأَرْضِ فَتَنُّونَ لَهُمْ قُلُوبَ يَعْقِلُونَ بِهَا أَوْ آدَانٍ يَسْمَعُونَ بِهَا فَإِنَّهَا لَا تَعْمَى الْأَبْصَارُ وَلَكِنْ تَعْمَى الْقُلُوبُ الَّتِي فِي الصُّدُورِ ٤٦

Artinya : “Maka tidakkah mereka berjalan di muka bumi, lalu mereka mempunyai hati yang dapat mereka gunakan untuk memahami, atau telinga yang dapat mereka gunakan untuk mendengar? Karena sesungguhnya bukanlah mata yang buta, tetapi hati yang ada di dalam dada lah yang buta.” (Q.S.Al-Hajj:22,46)

Menurut Tafsir Jalalain Jilid 2 halaman 541, ayat di atas dapat ditafsirkan : “Maka tidakkah mereka berjalan”, yakni orang-orang kafir Makkah “di muka bumi, lalu mereka mempunyai hati yang dapat mereka gunakan untuk memahami” apa yang telah diturunkan kepada para pendusta sebelum mereka “atau telinga yang dapat mereka gunakan untuk mendengar” kisah-kisah mereka tentang pembinasaaan dan penghancuran tempat tinggal mereka, sehingga mereka bisa mengambil pelajaran yang berharga? “Karena sesungguhnya” kisah itu bukanlah mata yang buta, tetapi hati yang ada di dalam dada lah yang buta.“

Seruan al-Quran untuk berpikir tampak jelas dari banyaknya ayat-ayat yang memuat ungkapan-ungkapan seperti “apakah kamu tidak memikirkan”, “apakah mereka tidak berpikir”, “agar kamu mengerti”, “agar kamu berpikir”, “apakah mereka tidak merenungkannya”, “apakah mereka tidak mau mengambil pelajaran”.

Dengan seruan Allah itu manusia sebagai khalifah di bumi, dengan akal budi dan ilmu pengetahuan yang diajarkan Allah dan dari sesama manusia, manusia dituntut untuk mampu menciptakan piranti kehidupannya, yaitu kebutuhan rohani seperti ilmu, seni, budaya, bahasa, sastra. Dengan karunia Allah dan akal budi serta cipta rasa dan karsa manusia mampu menghasilkan kebudayaannya. Di sini tampak jelas hubungan antara manusia dengan kebudayaan, bahwa manusia sebagai penciptanya sesudah Tuhan, juga manusia sebagai pemakai kebudayaan maupun sebagai pemelihara atau sebaliknya sebagai perusakannya.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari implementasi dan pengujian yang dilakukan oleh peneliti, maka dapat ditarik kesimpulan bahwa metode *Finite State Machine* dan logika *fuzzy* cocok diaplikasikan pada *game* ini. Metode diimplementasikan pada NPC sesuai rancangan model perilaku yang telah dibuat, kemudian diaplikasikan pada *game*. Bentuk dari representasi *Finite State Machine* berupa *State Diagram* yang menggambarkan tingkah laku oleh sistem kontrol ke dalam bentuk yang sederhana dan mudah dipahami, sehingga memudahkan untuk diimplementasikan ke dalam bahasa pemrograman. Sedangkan logika *fuzzy* digunakan untuk menghasilkan respon yang bervariasi pada NPC berdasarkan variabel-variabel *fuzzy*. Dari variabel-variabel *fuzzy* yang ada maka dapat dikombinasikan dan terbentuk sejumlah aturan-aturan *fuzzy* yang mana akan menentukan respon perilaku seperti menyerang jarak dekat, menyerang jarak jauh dan menjauh. Logika *fuzzy* dapat diterapkan untuk mengatur perilaku NPC musuh pada *state* menyerang di rancangan *Finite State Machine* sehingga dapat menghasilkan respon yang bervariasi. Logika *fuzzy* berjalan dengan baik dengan presentase error yang diperoleh sebesar 4,4 %.

Game ini berhasil dibuat dan berjalan dengan baik pada *smartphone* berbasis *android* dengan menggunakan Metaio SDK yang merupakan *framework*

untuk pengembangan aplikasi berbasis *augmented reality*. Selain itu, game ini berisi konten budaya lokal berupa rumah adat yang ada di Indonesia, misalnya menampilkan informasi baik berupa ciri-ciri, karakteristik maupun filosofinya. Game ini juga berjalan dengan baik saat pengujian di perangkat *smartphone* berbasis *android*.

5.2 Saran

Dalam pengembangan *game* ini masih banyak kekurangan. Oleh karena itu, penulis menyarankan untuk pengembang selanjutnya, diantaranya:

1. Mengembangkan *game* ini dengan penambahan level, karakter musuh dan konten nusantara agar permainan ini menjadi lebih menarik dan bercita rasa nusantara sehingga diharapkan agar *game* ini mempunyai daya saing yang tinggi dalam industri kreatif.
2. Dapat dijalankan di perangkat mobile yang lain (seperti iOS, Windows Phone, Blackberry) maupun pada perangkat desktop.

DAFTAR PUSTAKA

1. Al-Quranul Karim.
2. Al-Imam Jalaluddin Muhammad (2010), *Tafsir Jalalain : Jilid 1*, Surabaya : PT. eLBA Fitrah Mandiri Sejahtera.
3. Al-Imam Jalaluddin Muhammad (2010), *Tafsir Jalalain : Jilid 2*, Surabaya : PT. eLBA Fitrah Mandiri Sejahtera.
4. Rohiman Notowidagdi (1996), *Ilmu Budaya Dasar Berdasarkan Al-Quran da Hadits*, Jakarta : PT RajaGrafindo Persada.
5. Pemanfaatan multimedia,
<https://agungborn91.wordpress.com/2013/03/16/pemanfaatan-multimedia/>, diakses pada tanggal 29 Desember 2014 pukul 11.36.
6. [Dadi A. Salim](#), Pengguna Smartphone Menghabiskan Waktu 140 Menit Per Hari <http://swa.co.id/business-research/pengguna-smartphone-menghabiskan-waktu-140-menit-per-hari>, diakses pada tanggal 4 Januari 2015 pukul 09.00.
7. Kebudayaan lokal Indonesia,
<http://pariwisata.rejanglebongkab.go.id/kebudayaan-lokal-di-indonesia/>, diakses pada tanggal 4 Maret 2015 pukul 20.10.
8. Rumah Adat Kudus Terancam Punah,
<http://www.suaramerdeka.com/v1/index.php/read/news/2013/07/13/164334/Rumah-Adat-Kudus-Terancam-Punah>, diakses pada tanggal 4 Maret 2015 pukul 20.15.

9. Rumah tradisional kerinci makin langka,
<http://www.kerincizone.com/2013/03/rumah-tradisional-kerinci-makin-langka.html>, diakses pada tanggal 4 Maret 2015 pukul 21.00.
10. Rumah Joglo Kotagede Terancam Punah,
<http://www.tempo.co/read/news/2009/06/08/058180668/Rumah-Joglo-Kotagede-Terancam-Punah>, diakses pada tanggal 4 Maret 2015 pukul 21.25.
11. Alat Musik Bambu Asli Indonesia yang Hampir Punah,
<http://edukasi.kompasiana.com/2012/10/03/25-alat-musik-bambu-asli-indonesia-yang-hampir-punah-492537.html>, diakses pada tanggal 4 Maret 2015 pukul 21.25.
12. Pakaian Adat Palembang Terancam Punah,
<http://beritapagi.co.id/read/pakaian-adat-palembang-terancam-punah.html>,
diakses pada tanggal 4 Maret 2015 pukul 21.30.
13. <http://www.mandalamaya.com/pengertian-game-menurut-para-ahli/>,
diakses pada tanggal 21 Januari 2015 pukul 17.30.
14. www.scribd.com/doc/194987976/Ejarah-Game-Dan-Definisi-Menurut-Para-Ahli, diakses pada tanggal 12 Januari 2015 pukul 17.58.
15. http://dictionary.cambridge.org/dictionary/learner-english/game_1,
diakses pada tanggal 21 januari 2015 pukul 21.18
16. <http://dictionary.cambridge.org/dictionary/learner-english/computer-game>,
diakses pada tanggal 21 januari 2015 pukul 21.20.

17. <http://technologies-it.blogspot.com/p/finite-state-machines.html>, diakses pada tanggal 21 Mei 2015 pukul 19.00.
18. Azuma, Ronald T. (1997). "A Survey of Augmented Reality". *Presence: Teleoperators and Virtual Environments* **6** (4): 355–385.
19. Lester Madden (2011), "Professional Augmented Reality Browser for Smartphones : Programming for junaio, Layar, and Wikitude", United Kingdom : Wiley Publishing, Inc.
20. Brian Schwab (2009), *AI Game Engine Programming : Second Edition*, Canada : Course Technology
21. Ian Millington dan John Funge (2009), *Artificial Intelligence For Games : Second Edition*, United States : Elsevier
22. Nazruddin Safaat (2011), *Android, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Bandung: Informatika.
23. Dwi Ana Ratna Wati (2011), *Sistem Kendali Cerdas : Fuzzy Logic Controller (FLC), Jaringan Syaraf Tiruan (JST), Algoritma Genetika (AG) dan Algoritma Particle Swarm Optimization(PSO)*, Yogyakarta : Graha Ilmu
24. Agus Naba (2009), *Belajar Cepat Fuzzy Logic Menggunakan MATLAB*, Yogyakarta : ANDI
25. Sri Kusumadewi (2002), *Analisis dan Desain Sistem Fuzzy Menggunakan Toolbox Matlab*, Yogyakarta : Graha Ilmu
26. Ryan Henson Creighton (September, 2010), *Unity 3D Game Development by Example : Beginner's Guide*, Packt Publishing Ltd. Birmingham

27. Blender, <http://www.computerhope.com/jargon/b/blender.htm>, diakses pada tanggal 16 Februari 2015 pukul 14.20.
28. Blender, <http://www.blender.org/about/>, diakses pada tanggal 16 Februari 2015 pukul 14.30.
29. Metaio SDK Framework, <http://dev.metaio.com/sdk/documentation/metaio-sdk-framework/>, diakses pada tanggal 16 Februari 2015 pukul 16.00.
30. Trond Nilsen, Steven Linton dan Julian Looser (2004), *Motivations of Augmented Reality Gaming*, NZGDC'04, June 26–29, 2004, Dunedin, New Zealand.
31. Troels L. Andersen, Sune Kristensen, Bjørn W. Nielsen, and Kaj Grønbaek (2004), *Designing Augmented Reality Board Games: The BattleBoard 3D experience*, Conference Papers and Proceedings
32. Fekolkina Roman (2013), *Analysis of Augmented Reality Games in Android Platform*, S-801 76 Gävle, Sweden.
33. Surya Adi Wijaya, Susi Juniastuti, Supeno Mardi SN dan Moch. Hariadi (2009), *Desain Fuzzy State Machine untuk Menghasilkan Variasi Respon NPC pada Sebuah Game*, Prosiding Seminar Nasional Manajemen Teknologi IX, Program Studi MMT-ITS, Surabaya.
34. Yunifa Mifthachul Arif, Mochamad Hariadi, Supeno Mardi S.N (2011), *Strategi Menyerang pada Game FPS Menggunakan Hierarchy Finite State Machine dan Logika Fuzzy*, Tugas Akhir Tesis Mahasiswa ITS Surabaya Jurusan Teknik Elektro.

35. Miftakhur Rokhman (2011), *Finite State Machine sebagai agen cerdas animasi wajah pada game Assyuhada*, Skripsi Mahasiswa Teknik Informatika UIN Malang.
36. Galih Suyoga (2012), *Pengaplikasian Fuzzy State Machine untuk Menentukan Behavior Non Playable Character (NPC) Pada Game Hungry Pigs*, Tugas Akhir Mahasiswa Telkom University.
37. Ady Wicaksono (2013), *Respon Agen Menggunakan Fuzzy State Machine Pada Game First Person Shooter*, Tugas Akhir Tesis Mahasiswa ITS Surabaya Jurusan Teknik Elektro.

