

**PENERAPAN METODE BEHAVIOUR TREE PADA NPC
MUSUH DALAM GAME 2090 SEBAGAI PENDIDIKAN
MORAL PANCASILA**

SKRIPSI

Oleh:
ARINAL RIFQI
NIM. 15650072



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

**PENERAPAN METODE BEHAVIOUR TREE PADA NPC MUSUH
DALAM GAME 2090 SEBAGAI PENDIDIKAN
MORAL PANCASILA**

SKRIPSI

**Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
ARINAL RIFQI
NIM. 15650072**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

LEMBAR PERSETUJUAN
PENERAPAN METODE BEHAVIOUR TREE PADA NPC
MUSUH DALAM GAME 2090 SEBAGAI PENDIDIKAN
MORAL PANCASILA

SKRIPSI

Oleh :
ARINAL RIFQI
NIM. 15650072

Telah Diperiksa dan Disetujui untuk Diuji

Tanggal : juli 2020

Dosen Pembimbing I

Dosen Pembimbing II

Hani Nurhayati, M.T
NIP. 19780625 200801 2 006

Fajar Rohman Hariri, M. Kom
NIP. 198905152018011 001

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdiان
NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN

PENERAPAN METODE BEHAVIOUR TREE PADA NPC MUSUH DALAM GAME 2090 SEBAGAI PENDIDIKAN MORAL PANCASILA

SKRIPSI

Oleh:
ARINAL RIFQI
NIM. 15650072

Telah Dipertahankan di Depan Dewan Penguji
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Tanggal Juli 2020

Susunan Dewan Penguji		Tanda tangan
1. Penguji Utama	<u>Fresy Nugroho, M.T</u> : NIP. 19710722 201101 1 001	()
2. Ketua Penguji	<u>Fachrul Kurniawan, M.MT</u> : NIP. 19771020 200912 1 001	()
3. Sekretaris Penguji	<u>Hani Nurhayati, M.T</u> : NIP. 19780625 200801 2 006	()
4. Anggota Penguji	<u>Fajar Rohman Hariri, M. Kom</u> : NIP. 198905152018011 001	()

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdiان
NIP. 19740424 200901 1 00

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Arinal Rifqi
NIM : 15650072
Fakultas/Jurusan : Sains dan Teknologi/Teknik Infomatika
Judul Skripsi : Penerapan Metode Behaviour Tree Pada NPC
Musuh Dalam Game 2090 Sebagai Pendidikan
Moral Pancasila.

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 16 Juni 2020
Yang membuat pernyataan,



Arinal Rifqi
NIM. 15650072

HALAMAN MOTTO



HALAMAN PERSEMBAHAN

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

Atas kehadiran Allah Subhanahu wa ta'ala, dengan mengucap syukur penulis mempersembahkan sebuah karya untuk orang – orang yang sangat berarti

Terima kasih saya ucapkan kepada kedua orang tua yang selalu memberikan doa dan dukungan, sehingga membentuk individu yang dapat menyelesaikan kewajiban Pendidikan ini.

Terima kasih juga diucapkan kepada Ibu dan Bapak pembimbing yang telah membimbing dalam melakukan penelitian dan memberikan motivasi serta dorongan hingga penelitian terselesaikan dengan lancar.

Terima kasih kepada Tuhan yang maha ESA yang telah memberikan teman dan lingkungan yang sangat mendukung dan berpengaruh baik. Tak lupa pula pada teman Angkatan dan khususnya Risky, Hafizh, Zaky dan Safira yang memberikan dorongan, ajakan, motivasi dan semangat sehingga tugas akhir ini dapat diselesaikan.

Terima kasih untuk orang – orang yang tidak bisa disebutkan satu per satu yang telah mendoakan serta mendukung sehingga dapat terselesaikan skripsi ini.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji dan syukur saya panjatkan ke hadirat Allah subhanahu wa ta'ala yang telah melimpahkan rahmat dan hidayahNya kepada kita, sehingga saya dapat menyelesaikan Skripsi ini. Tujuan dari penyusunan skripsi ini guna memenuhi salah satu syarat untuk bisa menempuh ujian sarjana komputer pada Fakultas Sains dan Teknologi (FSAINTEK) Program Studi Teknik Informatika di Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang. Di dalam pengerjaan skripsi ini telah melibatkan banyak pihak yang sangat membantu dalam banyak hal. Oleh sebab itu, disini penulis sampaikan rasa terima kasih sedalam-dalamnya kepada:

1. Prof. Dr. Abdul Haris, M.Ag selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdiyan, Selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
4. Hani Nurhayati, M.T, selaku Dosen Pembimbing I yang telah membimbing dalam penyusunan skripsi ini hingga selesai.
5. Fajar Rohman Hariri, M. Kom, selaku Dosen Pembimbing II yang telah membimbing dalam penyusunan skripsi ini hingga selesai.
6. Dr. Cahyo Crysdiyan, selaku Dosen Wali yang senantiasa memberikan banyak motivasi dan saran untuk kebaikan penulis.

7. Kedua orang tua tercinta Bapak Bahagia dan Ibu Nurfaridah yang telah banyak memberikan doa dan dukungan kepada penulis secara moril maupun materil hingga Skripsi ini dapat terselesaikan.
8. Anggota keluarga dan kerabat yang senantiasa memberikan doa dan dukungan semangat kepada penulis.
9. Sahabat – sahabat Interface Teknik Informatika 2015 yang selalu memberikan semangat dan doa kepada penulis.
10. Semua pihak yang telah banyak membantu dalam penyusunan skripsi ini yang tidak bisa penulis sebutkan semuanya.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan penulis berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi.

Malang, Juli 2020

Penulis

DAFTAR ISI

HALAMAN PENGAJUAN	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN KEASLIAN TULISAN	Error! Bookmark not defined.
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR IS	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
المخلص	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4

1.5	Manfaat Penelitian.....	4
1.6	Sistematika Penulisan.....	5
BAB 2	TINJAUAN PUSTAKA	6
2.1	Penelitian Terkait.....	6
2.2	Landasan Teori	7
2.2.1	Pancasila.....	7
2.2.2	<i>Artificial Intelligence</i> pada <i>Games</i>	12
2.2.3	<i>Finite State Machine (FSM)</i>	13
2.2.4	<i>Hierarchical Finite State Machine (HFSM)</i>	15
2.2.5	<i>Decision Tree</i>	16
2.2.6	<i>Behaviour tree (BT)</i>	16
BAB 3	DESAIN SISTEM.....	21
3.1	Rancangan <i>Game</i>	21
3.1.1	Deskripsi <i>Game</i>	21
3.1.2	<i>Storyline</i>	21
3.1.3	<i>Storryboard</i>	22
3.1.4	Desain <i>User Interface</i>	23
3.1.5	Deskripsi Karakter dan Objek.....	25
3.2	Perancangan Metode	26
3.3	Rencana Pengujian	31
BAB 4	HASIL DAN PEMBAHASAN	32

4.1	Implementasi Desain	32
4.1.1	Perangkat yang Digunakan	32
4.1.2	Tampilan Antarmuka <i>Game</i>	33
4.2	Implementasi Metode	36
4.2.1	<i>Behaviour tree</i> (BT)	40
4.3	Pengujian	45
4.3.1	Uji Output Perilaku	45
4.3.2	Uji <i>Time Complexity</i> CPU	46
4.4	Integrasi Islam	48
BAB 5	PENUTUP	50
5.1	Kesimpulan	50
5.2	Saran	51
	DAFTAR PUSTAKA	52

DAFTAR GAMBAR

Gambar 2.1 Model AI pada <i>game</i> secara	12
Gambar 2.2 Contoh FSM (Mcquillan, 2015).	14
Gambar 2.3 Contoh HFSM (Mcquillan, 2015)	15
Gambar 2.4 Contoh Decision Tree dalam Penyewaan.....	16
Gambar 2.5 Contoh <i>Behaviour tree</i> (Mcquillan, 2015).	18
Gambar 2.6 Komponen Pada <i>Behaviour tree</i> (Mcquillan, 2015).	19
Gambar 3.1 Alur Perilaku NPC	28
Gambar 3.2 <i>Behaviour</i> Musuh Dalam Bentuk <i>Behaviour tree</i>	29
Gambar 4.1 Menu Utama.....	33
Gambar 4.2 Tampilan Laman Pancasila	33
Gambar 4.3 Tampilan Pembuka Level.....	34
Gambar 4.4 Tampilan Soal	34
Gambar 4.5 Tampilan <i>Gameplay</i> level 1	35
Gambar 4.6 Tampilan <i>Gameplay</i> level 2	35
Gambar 4.7 perbandingan penggunaan CPU untuk rendering dan <i>Scripts</i>	46
Gambar 4.8 perbedaan waktu yang dibutuhkan dalam eksekusi <i>script</i>	48

DAFTAR TABEL

Tabel 2.1 Kelebihan dan Kekurangan <i>Behaviour</i>	20
Tabel 3.1 <i>Storyboard</i>	22
Tabel 3.2 Desain <i>User Interface</i>	23
Tabel 4.3 Tabel keputusan NPC BT	45
Tabel 4.2 Tabel Sampel <i>Time Complexity</i> CPU berdasar jumlah NPC	47



ABSTRAK

Rifqi, Arinal. 2020. *Penerapan Metode Behaviour Tree Pada NPC Musuh Dalam Game 2090 Sebagai Pendidikan Moral Pancasila*. Skripsi. Jurusan Teknik Informatika Fakultas Sains Dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) Hani Nurhayati, M.T. (II) Fajar Rohman Hariri, M. Kom.

Kata Kunci : *Behaviour Tree*, Pancasila, Game, edukasi, NPC.

Pancasila adalah ideologi negara Indonesia yang mana menjadi dasar, pandangan hidup, serta kepribadian bangsa. Pendidikan dan pemahaman Pancasila sangat berpengaruh pada nilai moral dan etika berbangsa. Salah satu Tindakan yang dapat membentuk nilai yang sesuai dengan nilai Pancasila adalah dengan Pendidikan dan pengajaran terhadap Pancasila tersebut. Berdasarkan degradasi moral dan nilai oleh masyarakat terutama remaja berdasarkan kesesuaiannya dengan nilai moral Pancasila. Maka dibuatlah game edukasi Pancasila ini dengan tema petualangan yang berisikan pengajaran akan nilai Pancasila, dalam bentuk game sehingga dapat menarik minat remaja. Pada game ini NPC memiliki perilaku yang beragam, terdiri dari menyerang dalam jarak dekat, menyerang dengan jarak jauh, patroli, mengejar, berdiam, mengeluarkan kemampuan khusus dan meledak. Untuk mengatur dan mengontrol perilaku ini digunakannya Behaviour tree, yang mengatur perilaku tersebut dalam bagan berbentuk pohon dan mengontrolnya menggunakan control flow. Control flow akan mengatur arah dari root node pada bagan pohon tersebut kearah leaf node yang akan dieksekusi.

ABSTRACT

Rifqi, Arinal. 2020. *Application of Behavior Tree Method on NPC Enemies in Game 2090 as Moral Education of Pancasila*. Essay. Department of Informatics Engineering, Faculty of Science and Technology, Islamic State University of Maulana Malik Ibrahim of Malang. Counselor: (I) Hani Nurhayati, M.T. (II) Fajar Rohman Hariri, M. Kom.

Keywords : *fuzzy Sugeno, students, simulator, education, post tsunami mitigation.*

Pancasila is the ideology of the Indonesian state which forms the basis, outlook on life and national personality. Education and understanding of Pancasila determine the moral values and ethics of the nation. One of the actions that can form values in accordance with the values of Pancasila is the Education and teaching of the Pancasila. Based on moral degradation and values by adolescent-specific societies Based on their compatibility with the moral values of Pancasila. So this Pancasila educational game was made with the theme of adventure which contains the value of Pancasila, in the form of a game so that it can attract teenagers. In this game NPC has a variety of behaviors, consisting of attacking at close range, attacking at a distance, patrolling, chasing, idling, special ability and exploding. To regulate and control this behavior a behavior tree is used, which regulates the control tree chart and controls it using the control flow. Flow control will be directed from the root node in the tree chart towards the leaf node to be executed.

الملخص

ارينال، رفق. ٢٠٢٠ . NPC على أعداء **Behaviour tree** تطبيق طريقة في لعبة 2090 كتعليم أخلاقي لبانكاسيلا. أطروحة. الهندسة المعلوماتية ، كلية العلوم والتكنولوجيا ، الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج. المشرف : (1) حاني نورحياتي الماجستير. (٢) فجر الرحمن الحريري الماجستير.

Pancasila هي إيديولوجية الدولة الإندونيسية التي تشكل الأساس والنظرة إلى الحياة والشخصية الوطنية. تعليم وفهم Pancasila تحديد القيم والأخلاق الأخلاقية للأمة. أحد الإجراءات التي يمكن أن تشكل القيم وفقاً لقيم Pancasila هو تعليم وتعليم Pancasila. بناءً على الانحطاط الأخلاقي والقيم من قبل المجتمعات الخاصة بالمراهقين بناءً على توافقها مع القيم الأخلاقية لبانكاسيلا. لذلك تم صنع لعبة Pancasila التعليمية هذه بموضوع المغامرة الذي يحتوي على قيمة Pancasila ، في شكل لعبة بحيث يمكنها جذب المراهقين. في هذه اللعبة NPC لديها مجموعة متنوعة من السلوكيات ، التي تتكون من مهاجمة من مسافة قريبة ، مهاجمة عن بعد ، دوريات ، مطاردة ، التباطؤ ، قدرة خاصة والانفجار. لتنظيم هذا السلوك والتحكم فيه ، يتم استخدام شجرة السلوك التي تنظم مخطط شجرة التحكم وتحكم فيه باستخدام تدفق التحكم. سيتم توجيه التحكم في التدفق من العقدة الجذرية في المخطط الشجري باتجاه عقدة الورقة التي سيتم تنفيذها

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pesatnya Perkembangan teknologi pada abad 21 menciptakan tatanan kebiasaan baru dalam kehidupan. Perkembangan teknologi tersebut searah dengan pesatnya globalisasi. Arus globalisasi yang sering kali dijelaskan sebagai sebuah konsep abstrak yang menggambarkan berbagai proses ekonomi, sosial dan budaya yang berbeda beredar secara masif dan cepat seluruh dunia. Globalisasi tersebut ikut membawa nilai, pola pikir, pandangan, kebiasaan dan gaya hidup. Globalisasi dan perkembangan teknologi tidak hanya memiliki dampak negatif, layaknya pisau bermata dua (Nurhaidah, 2015). Cara kita dalam memanfaatkan hal tersebut yang menjadi penentu baik buruknya perkembangan teknologi tersebut dan globalisasi.

Globalisasi yang memberikan dampak pluralitas menjadi penyebab dari dekadensi moral yang dialami masyarakat, terutama generasi muda yang tidak mendapat Pendidikan karakter yang baik, Mereka akan kehilangan arah. Dekadensi moral terjadi ketika pengaruh globalisasi tidak sejalan dengan nilai-nilai Pancasila, dan ternyata nilai yang berasal dari luar lebih dominan (Nurwardani, 2016).

Pada penelitian terdahulu dalam usaha menumbuhkan moralitas bangsa untuk mengatasi dekadensi yang terjadi adalah dengan Pendidikan. Baik Pendidikan formal, nonformal dan informal bersinergi secara efektif menanamkan nilai-nilai moralitas dan tatakrma serta budi pakerti yang luhur. Ketiga lembaga tersebut saling mengisi dalam membangun anak didik, sebagai generasi masa depan yang

bermoral budi pekerti yang luhur. Tetapi cara tersebut, hanya dapat diterapkan pada masa Pendidikan. Hanya dapat dilakukan pada saat proses ajar-mengajar berlangsung. Pendidikan juga akan berhenti ketika masa Pendidikan berakhir (Iskarim, 2016).

Dengan ulasan serta pendekatan yang telah dilakukan diatas, maka perlunya agen globalisasi yang dapat memberikan nilai-nilai moral serta mendidik, bersifat fleksibel dan ikut serta dalam perkembangan teknologi itu sendiri. Salah satu dari agen dan media dari globalisasi terbesar adalah *Game*. *game* juga dapat ikut seta sebagai media edukasi. Dimana *game* itu sendiri tidak terbatas pada waktu dan tempat, sehingga bersifat fleksibel (Putra, Nugroho and Puspitarini, 2016).

ادْعُ إِلَى سَبِيلِ رَبِّكَ بِالْحُكْمِ وَالْمَوْعِظَةِ الْحَسَنَةِ وَجَادِلْهُمْ بِالَّتِي هِيَ أَحْسَنُ إِنَّ رَبَّكَ هُوَ أَعْلَمُ بِمَنْ ضَلَّ عَنْ سَبِيلِهِ وَهُوَ أَعْلَمُ بِالْمُهْتَدِينَ

Artinya: *Serulah (manusia) kepada jalan Tuhanmu dengan hikmah dan pelajaran yang baik dan bantahlah mereka dengan cara yang baik. Sesungguhnya Tuhanmu Dialah yang lebih mengetahui tentang siapa yang tersesat dari jalan-Nya dan Dialah yang lebih mengetahui orang-orang yang mendapat petunjuk.*(Q.S An-Nahl: 125).

Berdasarkan Tafsir Al-Jalalain ayat ini berupa ajakan kepada suatu kaum dengan cara yang sesuai dengan keadaan objek dakwah. Manusia tidak bertugas memberikan hidayah, melainkan hanya sebatas penyampai. Menyuru pada manusia ke jalan yang baik dengan hikmah, pelajaran yang baik atau nasihat yang lemah lembut dan membantah dengan bantahan yang baik pula (As-suyuthi, Muhammad and Ahmad, 1505).

Sebagai media edukasi dalam rangka memperbaiki dekadensi moral. *Game* ini yang berjudul 2090, memuat edukasi moral dan etika sesuai dengan butir-butir pada sila pertama Pancasila. *game* ini disegmentasikan pada kamu muda yang sudah tidak memiliki kewajiban dalam belajar Pendidikan kewarganegaraan. Berjudul 2090 karena bertema *fantasy* pada latar tahun tersebut. *Game* ini hadir petualangan yang mengedukasi, sehingga tidak mengurangi fungsinya sebagai media hiburan. Dalam membangun *game* petualangan yang menarik, maka memerlukan pengaturan NPC yang kompleks dengan perilaku yang variatif (Sekhavat, 2017).

Penggunaan perilaku NPC yang kompleks dan variatif membutuhkan banyak *state/task* serta memiliki transisi yang lebih rumit dari *game* pada umumnya. Pada umumnya NPC menggunakan metode FSM sebagai metode *Decision making*, tetapi dengan *state* yang banyak akan sangat sulit untuk diatur dan dikontrol (Sekhavat, 2017). Walaupun permasalahan ini dapat diatasi dengan menggunakan metode HFSSM, tetapi tetap saja akan tetap sulit untuk mengaturnya. FSM dan HFSSM yang menggunakan transisi berulang bukan merupakan solusi yang ideal, terutama modulnya yang tidak bersifat *reusable*. (Chang and Zhu, 2017)

Membutuhkan metode dinamis yang mendukung *state* yang banyak, maka pada penelitian ini peneliti menggunakan *Behaviour tree* sebagai metode *Decision making* NPC. Dengan adanya penelitian dan *game* ini diharapkan memberikan kemudahan dalam memahami cara kerja *Behaviour tree*.

1.2 Identifikasi Masalah

1. Bagaimana perilaku NPC yang diimplementasikan Behaviour tree sebagai metode Decision makingnya.
2. Bagaimana pengaruh *Behaviour tree* pada *time complexity* CPU.

1.3 Batasan Masalah

1. Berbasis Dekstop dan menggunakan platform windows.
2. Penelitian Terbatas pada performa CPU dan alur dari perilaku NPC.
3. Penelitian ini terbatas pada metode *Behaviour tree* sebagai *decision making*,
4. Penelitian ini tidak berfokus pada fungsi *game* sebagai media edukasi.

1.4 Tujuan Penelitian

Tujuan dilakukan penelitian ini menghasilkan perilaku NPC yang adaptif menggunakan metode *Behaviour tree*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi pada perkembangan metode perilaku (*decision making*) pada NPC, serta dapat dimanfaatkan sebagai metode dalam mengatur strategi.

1.6 Sistematika Penulisan

Uraian dalam laporan skripsi penulis menyusun dengan sistematika penulisan sebagai berikut :

BAB I : PENDAHULUAN

Pendahuluan berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Tinjauan pustaka berisikan tentang teori yang berhubungan dengan permasalahan penelitian dari *game* yang dibuat, yang selanjutnya digunakan dalam bagian pembahasan dan sebagai dasar dalam pembuatan sistem.

BAB III : METODOLOGI PENELITIAN

Metodologi penelitian berisikan tentang perancangan sebuah *game* 3D yang meliputi metode penelitian yang digunakan, perancangan *game* 3D yang akan dibuat.

BAB IV : HASIL DAN PEMBAHASAN

Analisa dan perancangan berisikan tentang analisa sistem aplikasi dan perancangannya

BAB V : PENUTUPAN

Pada bab terakhir berisi kesimpulan dan saran berdasarkan hasil yang telah dicapai dari pembahasan.

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Berikut penelitian yang terkait dengan penelitian, yaitu:

(Mcquillan, 2015) dalam jurnalnya yang berjudul “*A Survey of Behaviour trees and their Applications for Game AI*” ia membandingkan FSM, HFSM dengan BT, komponen dasar pembangun BT, kelebihan dan kekurangan BT, serta penggunaan BT dalam *game development* dan menyinggung topik lanjut dari BT yang dapat dikombinasikan dengan metode lainnya.

(Olsson, 2016) pada penelitiannya yang berjudul “*Behaviour trees for decision-making in autonomous driving*” ia menggunakan BT sebagai arsitektur untuk *behavioral layer* pada *autonomous driving*. BT yang secara umum digunakan pada *video games* mendapat perhatian untuk penelitian robotika.

(Sekhavat, 2017) dalam penelitiannya yang berjudul “*Behaviour tree for Computer Games*” menjelaskan, walaupun sebuah *finite state machine* (FSM) mudah untuk diimplementasikan pada NPC dalam *computer games*, tetapi sangat sulit untuk mengontrol *behaviour* nya terutama karena pertumbuhan jumlah *states*. *Behaviour tree* (BT) yang mana merupakan *hierarchical nodes* yang berbentuk pohon untuk mengontrol alur *decision making* yang mana secara luas digunakan untuk mengatasi permasalahan ukuran dari *behaviour*. Sekhavat juga menjelaskan akan kelemahan serta kelebihan dari BT tersebut.

(Ryan Keith Marcotte Regina, 2017) pada penelitiannya yang berjudul “*Modelling Artificial Intelligence in Games Using Mindset Behaviour trees*” ia

berpendapat bahwa popularitas BT dikarenakan dari keunggulannya pada *maintainability, scalability, reusability and extensibility*.

2.2 Landasan Teori

2.2.1 Pancasila

Pancasila adalah ideologi dari negara Indonesia. Pancasila berasal dari bahasa sansakerta yaitu *panca* yang berarti lima, serta *sila* yang berarti asas atau dasar. Pancasila dicetuskan oleh para leluhur sebagai ideologi negara, yang mana menjadi dasar, pandangan hidup, serta kepribadian bangsa. Oleh sebab itu sebagai ideologi, Pancasila juga disebut sebagai sumber dari segala hukum (Asmaroini, 2017).

Berikut merupakan lima sila dalam Pancasila serta butir-butir Pancasila, berdasarkan Tap MPR no. I/MPR/2003.

1. Sila kesatu, Ketuhanan Yang Maha Esa

Berlambang bintang. Bintang yang Menyimbolkan Ketuhanan Yang Maha Esa. Maksudnya rakyat Indonesia mempercayai Tuhan yang satu, Tuhan Yang Maha Esa.

- a. Bangsa Indonesia menyatakan kepercayaannya dan ketaqwaannya kepada Tuhan Yang Maha Esa.
- b. Manusia Indonesia percaya dan taqwa terhadap Tuhan Yang Maha Esa, sesuai dengan agama dan kepercayaannya masing-masing menurut dasar kemanusiaan yang adil dan beradab.
- c. Mengembangkan sikap hormat menghormati dan bekerjasama anatra pemeluk agama dengan penganut kepercayaan yang berbeda-beda terhadap Tuhan Yang Maha Esa.

- d. Membina kerukunan hidup di antara sesama umat beragama dan kepercayaan terhadap Tuhan Yang Maha Esa
 - e. Agama dan kepercayaan terhadap Tuhan Yang Maha Esa adalah masalah yang menyangkut hubungan pribadi manusia dengan Tuhan Yang Maha Esa.
 - f. Agama dan kepercayaan terhadap Tuhan Yang Maha Esa adalah masalah yang menyangkut hubungan pribadi manusia dengan Tuhan Yang Maha Esa.
 - g. Mengembangkan sikap saling menghormati kebebasan menjalankan ibadah sesuai dengan agama dan kepercayaannya masing masing
 - h. Tidak memaksakan suatu agama dan kepercayaan terhadap Tuhan Yang Maha Esa kepada orang lain.
2. Sila kedua, Kemanusiaan Yang Adil Dan Beradab
- Berlambang rantai. Rantai yang Menyimbolkan Kemanusiaan Yang Adil Dan Beradab. Maksudnya Negara Indonesia menjamin rakyatnya untuk mendapatkan Hak hidup, Hak kebebasan, persamaan hak, serta hak untuk memiliki tanpa ada rasa.
- a. Mengakui dan memperlakukan manusia sesuai dengan harkat dan martabatnya sebagai makhluk Tuhan Yang Maha Esa.
 - b. Mengakui persamaan derajat, persamaan hak dan kewajiban asasi setiap manusia, tanpa membeda-bedakan suku, keturunan, agama, kepercayaan, jenis kelamin, kedudukan sosial, warna kulit dan sebagainya.
 - c. Mengembangkan sikap saling mencintai sesama manusia.
 - d. Mengembangkan sikap saling tenggang rasa dan tepa selira.
 - e. Mengembangkan sikap tidak semena-mena terhadap orang lain.
 - f. Menjunjung tinggi nilai-nilai kemanusiaan.

- g. Gemar melakukan kegiatan kemanusiaan.
- h. Berani membela kebenaran dan keadilan.
- i. Bangsa Indonesia merasa dirinya sebagai bagian dari seluruh umat manusia.
- j. Mengembangkan sikap hormat menghormati dan bekerjasama dengan bangsa lain.

3. Sila ketiga, Persatuan Indonesia (Kebangsaan)

Berlambang pohon beringin. Pohon Beringin yang Menyimbolkan Persatuan Indonesia Artinya, meskipun Indonesia terdiri dari banyak sekali agama, ras, suku, budaya, dan adat istiadat yang berbeda. Namun, Indonesia harus tetap bersatu. Lebih mementingkan kesatuan dan persatuan bangsa, serta tidak mementingkan individu atau golongan di atas kepentingan bersama sebagai Negara Kesatuan Republik Indonesia (NKRI).

- a. Mampu menempatkan persatuan, kesatuan, serta kepentingan dan keselamatan bangsa dan negara sebagai kepentingan bersama di atas kepentingan pribadi dan golongan.
- b. Sanggup dan rela berkorban untuk kepentingan negara dan bangsa apabila diperlukan.
- c. Mengembangkan rasa cinta kepada tanah air dan bangsa.
- d. Mengembangkan rasa kebanggaan berkebangsaan dan bertanah air Indonesia.
- e. Memelihara ketertiban dunia yang berdasarkan kemerdekaan, perdamaian abadi dan keadilan sosial.
- f. Mengembangkan persatuan Indonesia atas dasar Bhinneka Tunggal Ika.
- g. Memajukan pergaulan demi persatuan dan kesatuan bangsa.

4. Sila keempat, Kerakyatan Yang Dipimpin Oleh Hikmat Kebijaksanaan Dalam Permusyawaratan/Perwakilan

Berlambang kepala banteng. Kepala Banteng yang Menyimbolkan Kerakyatan Yang Dipimpin Oleh Hikmat Kebijaksanaan, Dalam Permusyawaratan Perwakilan maksudnya di sini, kita sebagai rakyat Indonesia tidak boleh egois dengan hanya mementingkan kepentingan pribadi dan tidak mengutamakan kepentingan negara dan masyarakat.

- a. Sebagai warga negara dan warga masyarakat, setiap manusia Indonesia mempunyai kedudukan, hak dan kewajiban yang sama.
- b. Tidak boleh memaksakan kehendak kepada orang lain.
- c. Mengutamakan musyawarah dalam mengambil keputusan untuk kepentingan bersama.
- d. Musyawarah untuk mencapai mufakat diliputi oleh semangat kekeluargaan.
- e. Menghormati dan menjunjung tinggi setiap keputusan yang dicapai sebagai hasil musyawarah.
- f. Dengan i'tikad baik dan rasa tanggung jawab menerima dan melaksanakan hasil keputusan musyawarah.
- g. Di dalam musyawarah diutamakan kepentingan bersama di atas kepentingan pribadi dan golongan.
- h. Musyawarah dilakukan dengan akal sehat dan sesuai dengan hati nurani yang luhur.
- i. Keputusan yang diambil harus dapat dipertanggungjawabkan secara moral kepada Tuhan Yang Maha Esa, menjunjung tinggi harkat dan martabat

manusia, nilai-nilai kebenaran dan keadilan mengutamakan persatuan dan kesatuan demi kepentingan bersama.

- j. Memberikan kepercayaan kepada wakil-wakil yang dipercayai untuk melaksanakan pemusyawaratan.

5. Sila kelima, Keadilan Sosial Bagi Seluruh Rakyat Indonesia

Kapas yang Menyimbolkan Keadilan Sosial Bagi Seluruh Rakyat Indonesia maksudnya, untuk menjadi warga Negara yang baik, kita harus senantiasa mengembangkan perbuatan-perbuatan luhur, sehingga dapat mencerminkan sikap serta suasana kekeluargaan dan gotong royong. Bersikap adil pada semua golongan dengan menjaga keseimbangan antara hak dan kewajiban.

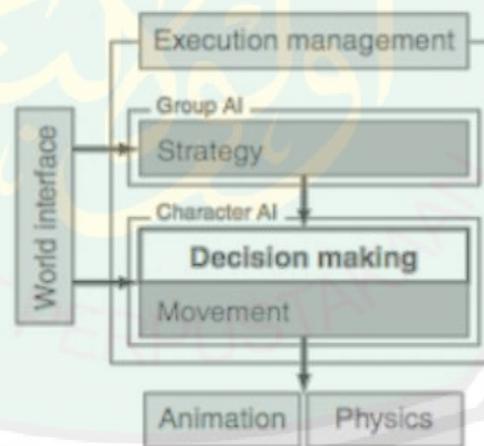
- a. Mengembangkan perbuatan yang luhur, yang mencerminkan sikap dan suasana kekeluargaan dan kegotongroyongan.
- b. Mengembangkan sikap adil terhadap sesama.
- c. Menjaga keseimbangan antara hak dan kewajiban.
- d. Menghormati hak orang lain.
- e. Suka memberi pertolongan kepada orang lain agar dapat berdiri sendiri.
- f. Tidak menggunakan hak milik untuk usaha-usaha yang bersifat pemerasan terhadap orang lain
- g. Tidak menggunakan hak milik untuk hal-hal yang bersifat pemborosan dan gaya hidup mewah.
- h. Tidak menggunakan hak milik untuk bertentangan dengan atau merugikan kepentingan umum.
- i. Suka bekerja keras.

- j. Suka menghargai hasil karya orang lain yang bermanfaat bagi kemajuan dan kesejahteraan bersama.
- k. Suka melakukan kegiatan dalam rangka mewujudkan kemajuan yang merata dan berkeadilan sosial.

2.2.2 Artificial Intelligence pada Games

(Kirby, 2011) mendefinisikan bahwa AI adalah “kemampuan untuk bertindak secara cerdas pada kondisi keadaan yang berubah-ubah”. Pada *games*, AI harus mampu bertindak dengan cara tertentu, aksi yang dilakukan AI harus dapat dirasakan oleh *player*. AI harus cerdas, tidak hanya melakukan aksi secara acak. AI dapat bereaksi pada perubahan kondisi. Kita setuju, yang membuat sebuah *game* menarik adalah tingginya interaktivitas yang dilakukan *player*.

Sangat banyak pendekatan pada *game* AI, secara umum bentuk dari kebanyakan system AI pada *Game* sebagai berikut (Mcquillan, 2015).



Gambar 2.1 Model AI pada *game* secara umum (Mcquillan, 2015)

Penggunaan AI dalam *game* terdapat pada beberapa layer. *Execution management* AI ini merupakan jenis AI yang berkaitan dengan pengaturan dari NPC. Terdapat *decision making* (FSM, FuSM dan Bt) dan *movement* (A*). Serta

group AI yang mengatur NPC secara jamak atau dalam jumlah besar, yang biasanya digunakan dalam *game* RTS untuk mengatur strategi dan posisi dari NPC. Pada penelitian ini akan membahas AI dalam layer *decision making*.

Beberapa pendekatan pada AI. *Scripted AI*, atau juga dikenal dengan *hard-coded AI*. Merupakan pendekatan paling tradisional dan masih banyak digunakan. Mulai dari kondisi yang sederhana (*if-else*) sampai pada yang lebih berorientasi objek, pendekatan *data-driven*, sederhana, cepat dan efektif dengan sedikit proses komputasi dan eksekusi yang cepat.

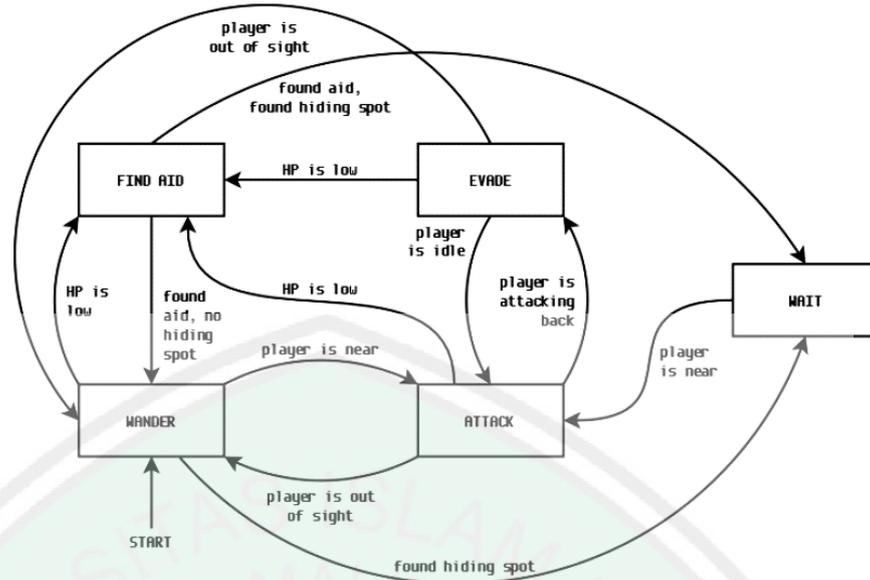
Rule Based System, terdiri dari beberapa aturan dan *framework*. ia akan memeriksa setiap *state* nya agar dapat sesuai dengan aturan yang berlaku pada tempatnya. Sistemnya tidak terlalu terikat dibanding pendekatan *state-based*.

Random and probabilistic systems, sangat efektif dalam mensimulasikan tingkah laku nyata, yang mana biasanya tidak terduga. metode ini dikenal aneh untuk menyeimbangkan isu *predictability vs feasibility*. Contoh dari metode ini adalah monte-carlo, *precomputing* dan *'faking it'*.

Machine Learning, menggunakan Teknik seperti algoritma genetika, jarang digunakan diluar dunia akademisi. Keluarannya sangat tidak dapat diprediksi, cocok untuk beberapa *game* tapi sangat sulit untuk dikontrol. Terdapat beberapa pendekatan lainnya, seperti STRIPS, GOAP, HTN, A* dan *Behaviour tree*.

2.2.3 Finite State Machine (FSM)

Finite State Machine (FSM). Merupakan metode yang dapat digunakan dalam merancang perilaku NPC yang memiliki kelebihan pada kesederhanaan komputasinya dan kemudahan dalam pemahaman komputasinya (Bimantoro and Haryanto, 2016).



Gambar 2.2 Contoh FSM (Mcquillan, 2015).

FSM terdiri dari beberapa komponen. Yang berbentuk kotak merupakan *task* atau yang disebut sebagai *state*. Kemudian garis yang menghubungkan 2 *state* disebut sebagai *transition*, serta teks yang terdapat pada transisi tersebut disebut sebagai *action/event*.

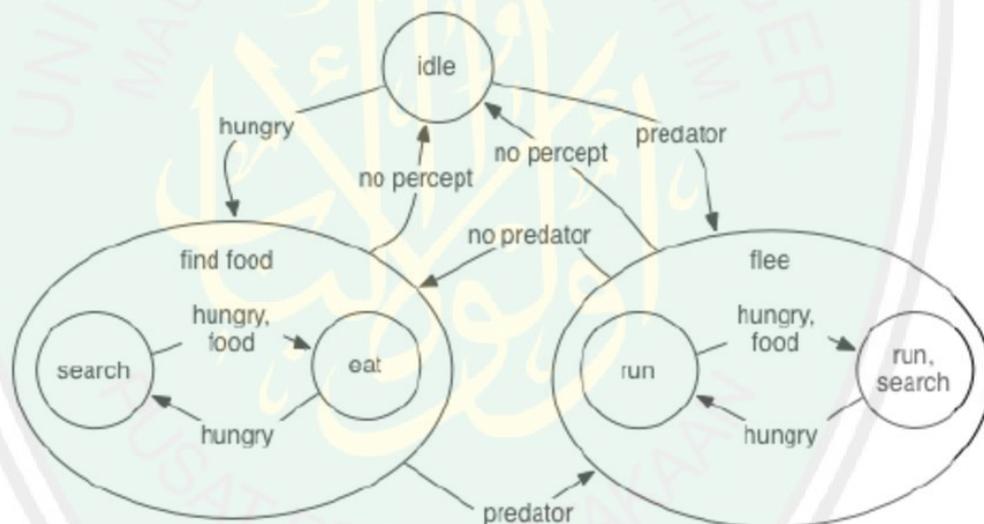
Berikut merupakan permasalahan pada FSM (Sekhavat, 2017).

- Maintainability*, ketika menambah atau menghapus suatu *state*, terdapat kemungkinan untuk mengubah kondisi dari semua *state*.
- Scalability*, FSM dengan penggunaan *state* yang banyak dapat menghilangkan keunggulannya dalam *graphical readability*, sehingga menjadi sangat sulit untuk dibaca dan dikontrol.
- Reusability*, hampir tidak memungkinkan untuk menggunakan *behavior* yang sama untuk beberapa projek yang sama.

2.2.4 Hierarchical Finite State Machine (HFSM)

Metode FSM yang sudah sangat dikenal akan kehandalannya jika diterapkan dalam *game*. Namun, sangat sulit dalam memperlihatkan agen yang terdiri dari beberapa perilaku. Salah satu solusi dalam permasalahan ini adalah dengan menggunakan metode HFSM (Mcquillan, 2015).

Pada HFSM, *state* FSM kembali didefinisikan dalam FSM lain. FSM yang berada didalam dapat disebut *slave* dan FSM yang ada diluar dapat disebut *master*. HFSM tidak menambahkan model komputasi. Tetapi HFSM bisa mengurangi jumlah transisi secara signifikan dan membuat FSM lebih intuitif dan lebih dipahami (Fauzi, 2015).

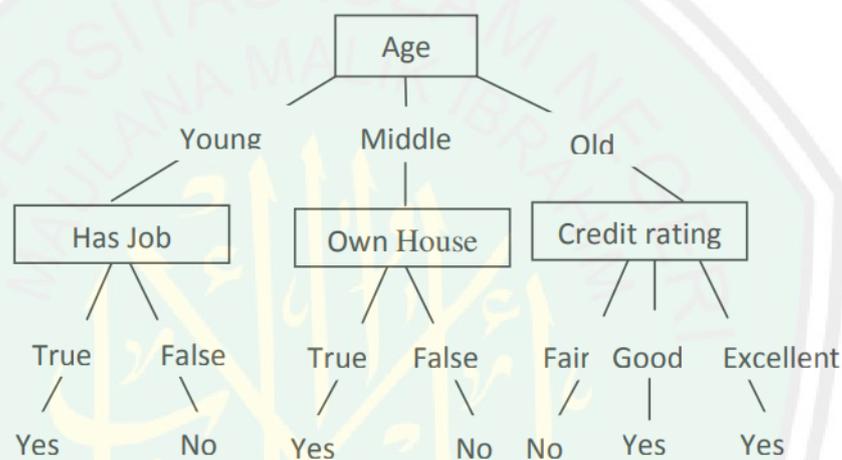


Gambar 2.3 Contoh HFSM (Mcquillan, 2015)

Cara kerja dari HFSM sama seperti FSM, hanya *state* yang berisikan FSM lain yang membedakannya. Pada gambar 2.3 tersebut, “*find food*” dan “*flee*” menjadi *master* dari *state* “*search*”, “*eat*”, “*run*” dan “*run search*”.

2.2.5 Decision Tree

Decision Tree merupakan bagan struktur dalam bentuk pohon, seperti flowchart. Dengan node berbentuk kotak. Setiap node mempresentasikan sekumpulan data. Root node merupakan node paling atas, node tanpa parent. Decision tree digunakan untuk menemukan jalan terbaik untuk mengeksekusi suatu *class/task/node*. terdapat lima macam algoritma decision tree yang banyak dipakai, ID3, CART, CHAID, C4.5 dan J48 (Saini, Rai and Jain, 2014).



Gambar 2.4 Contoh Decision Tree dalam Penyewaan
(Saini, Rai and Jain, 2014)

Pada *decision tree* hanya terdapat *true* dan *false*, berbeda dengan *Behaviour tree* yang terdapat *success*, *failure* dan *running*. serta pada *decision tree* tidak terdapat *control flow* (Sekhavat, 2017).

2.2.6 Behaviour tree (BT)

Behaviour tree adalah model matematis yang digunakan untuk mengeksekusi dan menyusun peralihan antara berbagai *task* yang berbeda pada *autonomous agent* (Colledanchise and Ögren, 2017). *Behaviour tree* adalah sebuah model untuk pengekseskuan rencana yang secara grafis dipresentasikan sebagai sebuah pohon

(Marcotte and Hamilton, 2017). *Behaviour tree* memungkinkan untuk mengontrol karakter pada *games* dan menjelaskan hirarki keputusan dan aksi (Sekhavat, 2017). Sehingga dapat dimengerti bahwa *Behaviour tree* adalah model yang memiliki struktur seperti pohon yang dapat digunakan untuk mengontrol *behaviour* NPC.

BT muncul sebagai *tool* untuk pemodelan *artificial intelligence* (AI) pada *games* sejak digunakannya pada *game* Halo 2. BT sering digunakan sebagai *decision making* pada NPC. BT juga populer digunakan sebagai metode yang efektif untuk mengontrol robot. Biasanya, sebuah *game engine* secara berulang mengeksekusi sebuah *game loop*, dimana didalamnya terdapat *sub-engine* untuk *physics*, *artificial intelligence*, grafis, dan sebagainya (Marcotte, 2017)

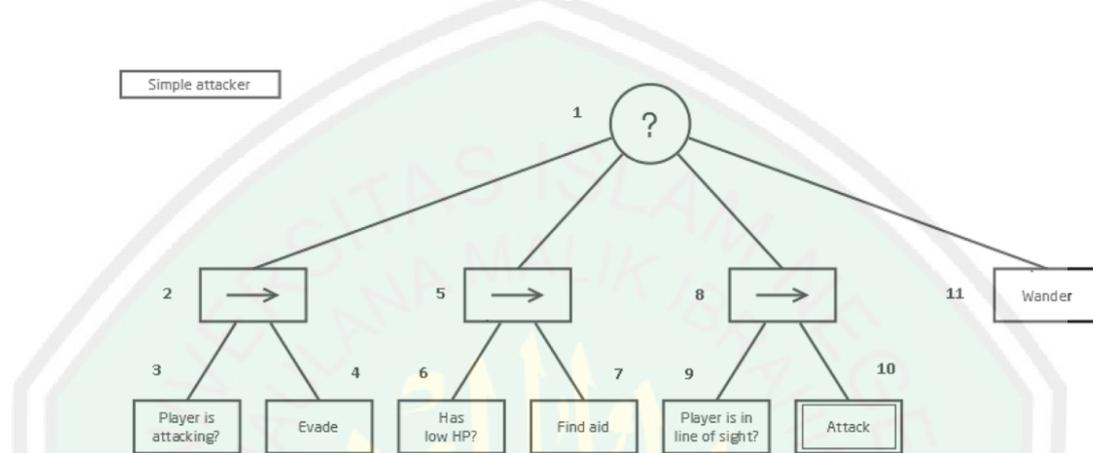
Selama *game* berjalan, maka *tree* akan terus “ticked”, yang artinya ia akan terus bergerak dari *root* melalui semua *child node* semakin ke bawah, jika *node* memiliki *child*. Menggunakan algoritma DFS (*Depth First Search*), *ticked* bergerak dari atas ke bawah dan dari kiri ke kanan (Sekhavat, 2017).

Setiap *node* memiliki aksi dan kembalian (*return*) tersendiri kepada *parent* nya, diantaranya:

- *Success*: *node* selesai melakukan tugasnya
- *Failure*: *node* gagal menyelesaikan tugasnya
- *Continue*: *node* belum menyelesaikan tugasnya

Root pada BT adalah *node* tanpa *parent*. Sedangkan, *node* yang tidak memiliki *child* adalah leaf (daun). Pada standar BT, *non-leaf node* bisa berupa *selector node* atau *sequence node*. *Selector node* digunakan ketika kita ingin mengeksekusi *child* pertama yang dapat berjalan tanpa gagal (*failure*). Pada *sequence node*, ia akan menjalankan semua *child* nya secara berurutan, *success* jika

semua *child node* dapat berjalan secara *success*. Sedangkan pada *leaf*, dapat berupa *action* atau *condition*. *Action node* dapat berupa dijalankannya animasi, merubah *state* karakter, atau apapun yang dapat mengubah *state* pada *game*. Sedangkan *condition node* pada umumnya digunakan untuk *testing* suatu nilai. *Condition* dikatakan *success* jika syarat nilai yang ditentukan sesuai (Sekhavat, 2017).



Gambar 2.5 Contoh *Behaviour tree* (Mcquillan, 2015).

Behaviour tree memiliki berbagai macam komponen, tetapi dasar yang harus dan pasti dimiliki oleh BT adalah sebagai berikut (Sekhavat, 2017).

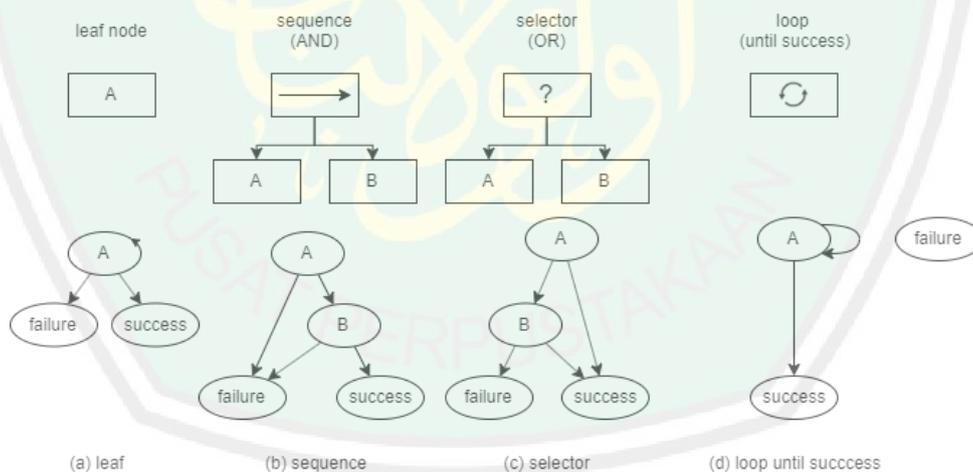
- a. *Leaf node* atau *action node (state)*, merupakan komponen yang menjalankan sebuah aksi. Seperti memukul, kabur, lari, jalan, menjalankan clip suara, animasi, ledakan dan sebagainya. Jika komponen ini berhasil dijalankan maka akan mengembalikan kembalian *success*. Sedangkan jika komponen selesai dijalankan dan tidak berhasil maka akan mengembalikan kembalian *failure*. Jika komponen masih dalam keadaan berjalan maka akan mengembalikan kembalian *running*. pada contoh gambar 2.3. 'wander' merupakan komponen ini.
- b. *Condition component (if-else)*, komponen ini berupa *Boolean question*, dimana mengembalikan kembalian *true* atau *false*. Tidak mungkin untuk kembalian

running. pada gambar 2.3. digambarkan dalam bentuk pertanyaan. ‘*player is attacking?*’

- c. *Control Flow (Selector dan Sequence)*, merupakan kelompok komponen *child* dan menentukan urutan eksekusi diantaranya. Terdapat dua macam *control flow* secara umum.

Selector, merupakan komponen yang berjalan dengan cara membuat keputusan. Berbeda dengan *decision*. berjalan dari kiri ke kanan. Contoh pada gambar 2.5 berupa lingkaran dengan tanda tanya. cara kerjanya dapat dilihat pada gambar 2.6.

Sequence, komponen yang memuat sederet *child* yang dieksekusi dari kiri ke kanan. Contohnya dapat dilihat pada gambar 2.3.berbentuk seperti anak panah. Cara kerjanya dapat dilihat pada gambar 2.6.



Gambar 2.6 Komponen Pada *Behaviour tree* (Mcquillan, 2015).

Tabel 2.1 Kelebihan dan Kekurangan *Behaviour tree* (Mcquillan, 2015)

	Simplicity	Complexity	Readability	Debugability	Scalability	Modularity	Reusability
Strength	Elegant & intuitive	Powerful	Non-programmer Designers	Simple to follow	No fixed limit to number of levels	Aids reusability, hybridity	Increased productivity
Limitation		Can affect performance					
	Flexibility	Composability	Hierarchy	Dependency	Nuance	Hybridity	Efficiency
Strength	Many applications & hybrid approaches	Sub-trees handle complexity, & hybridity, while maintaining a single shared, intuitive interface	Improvement on FSMs, HSMs for managing complexity		Bottom Up can make more informed, better decisions	Eminently suitable for FSMs, Planners, & most others; Can result in best of both worlds	Top Down (Fast) Data-oriented (Cache) Event-driven (Reduced workload)
Limitation	Capable of most AI needs, but not ALWAYS the very best option		Static prioritisation can increase dependencies	Reliance on fluid externalities can lead to "stupid" behaviour/ loops	Can require knowledge of changing external systems, which means constant updates	Need a good understanding of what works best when, to take full advantage of hybridity	Bottom Up (Increased processing power required)

Pada tabel tersebut dapat dimengerti bahwa, *Behaviour tree* merupakan metode yang elegan, memudahkan pengguna dalam mengatur *task*. *Behaviour tree* menggunakan alur desain umum, sehingga penggunaanya tidak dibatasi hanya oleh *programmer*. Dengan alur desain tersebut juga dapat memudahkan dalam *debugging*. Bergantung pada tampilan pada alur *task*.

Dalam menangani *task* yang rumit serta Jumlah dari *task* dan turunan *child* yang digunakan tidak dibatasi. *Behaviour tree* tetap mampu menanganinya, walaupun berpengaruh terhadap performa dari *game*.

Keunggulan dari *Behaviour tree* yang tidak dapat dipungkiri juga terdapat pada sifatnya yang modular, Dapat digunakan berulang, dapat dihibridisasi dengan metode lain, dapat digunakan dalam banyak hal, tidak dibatasi hanya untuk *game*, Kompabilitas yang baik.

BAB 3

DESAIN SISTEM

3.1 Rancangan *Game*

3.1.1 Deskripsi *Game*

Penelitian ini berupa *game adventure* yang memuat edukasi secara tersirat dari keadaan dan cerita yang disajikan. Memuat cerminan dari akhlak *player* yang menyuarakan butir-butir pancasila. Memakai sudut pandang orang ketiga. *Game* disegmentasikan kepada anak muda yang sudah tidak memiliki kewajiban dalam Pendidikan kewarganegaraan.

Game ini berlatar pada tahun 2090 dengan *art style* yang minimalis serta memiliki efek yang hidup. *Game* bercerita tentang Nusantara melawan penjajahan dari makhluk asing yang datang ingin merebut kekuasaan Nusantara. *Player* diharuskan menyelesaikan setiap misi yang diberikan untuk menamatkan *game*. Terdapat beberapa misi, dimana pada setiap misi memiliki *objective* yang berbeda-beda, terdiri dari penjagaan, penyelamatan dan penyerangan.

3.1.2 *Storyline*

Game ini menceritakan keadaan Indonesia pada tahun 2090. Dimana keadaan masyarakatnya telah berubah, manusia hidup berdampingan dengan robot serta memiliki AI yang cukup cerdas. Misi akan menggambarkan perilaku yang sesuai dengan 45 butir Pancasila. Memberikan pedoman melalui tingkah laku yang baik yang disajikan pada cerita *game*.

Zico merupakan pemuda yang menemukan rekaman yang berisi pesan pada tahun 2020, rekaman berisi peringatan akan kehancuran nusantara akibat ulah masyarakatnya sendiri, serta ancaman akan kekuatan Robot super AI yang

diciptakan untuk menjadi pemimpin masyarakat nusantara pada tahun tersebut, dimana kemudian bertindak diktatorat. Zico yang tersadar mulai melakukan pergerakan, dengan mengedepankan nilai Pancasila yang ia dapat. Pada misi pertama Zico membakar media propaganda yang diciptakan oleh super AI. propaganda yang menantang Pancasila yang memuat pemahaman.

3.1.3 Storyboard

Berikut *storyboard* pada *game* ini:

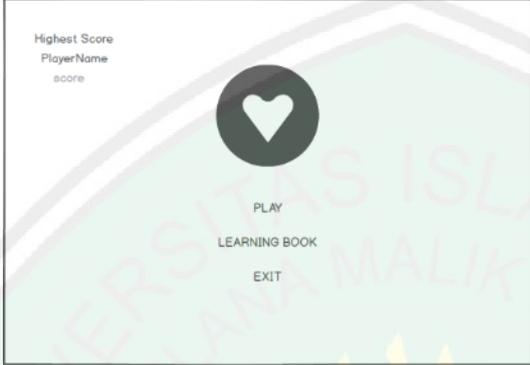
Tabel 3.1 *Storyboard*

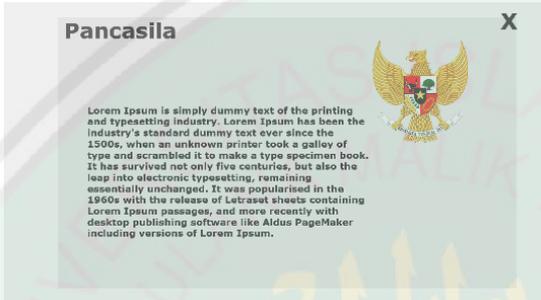
No.	Gambar	Keterangan
1.		<p>pada awal level terdapat pemberian materi sekilas tentang Pancasila.</p> <p>Serta pada akhir level terdapat soal-soal yang berkaitan dengan Pancasila.</p>
2.		<p><i>player</i> memiliki misi untuk menghancurkan media propaganda yang disebarakan pada map.</p>
3.		<p>Misi dikatakan selesai jika semua objektif pada misi telah dapat dipenuhi.</p>

3.1.4 Desain User Interface

Berikut penjelasan desain *interface* pada *game* ini:

Tabel 3.2 Desain User Interface

No	Frame	Isi	Keterangan
1.		<p>Frame awal</p> <p>berisi:</p> <ul style="list-style-type: none"> ➤ play ➤ learning book ➤ exit <p>highest score</p>	<ul style="list-style-type: none"> ➤ <i>Button play</i> untuk memulai permainan ➤ <i>Button learning book</i>, berisikan pengetahuan tentang Pancasila. ➤ <i>Button exit</i>, mengakhiri permainan atau keluar dari permainan. ➤ <i>Highest score</i>, menunjukkan nilai tertinggi yang telah

No	Frame	Isi	Keterangan
			<p>dimainkan oleh <i>player</i> tertentu</p>
2.	 <p>Pancasila X</p> <p> Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p>	<p><i>Frame</i> <i>learning</i> <i>book,</i></p>	<p>berisikan pelajaran tentang Pancasila serta ayat-ayat yang berhubungan dengan pancasila. Mirip seperti <i>collection</i> yang dimiliki oleh <i>game</i> biasanya.</p>

No	Frame	Isi	Keterangan
3.		Frame pada <i>game play</i>	<p>➤ terdapat HP bar yang menginfokan keadaan dari HP <i>player</i> Energy bar, yang menginfokan keadaan dari energy atau mana dari <i>player</i> yang digunakan untuk mengeluarkan <i>skill</i>.</p>

3.1.5 Deskripsi Karakter dan Objek

Rancangan Karakter Utama Karakter utama yang dijalankan oleh *player* bernama zico. Dia merupakan seorang anak kecil yang dikemudian pada akhir cerita menjadi seorang pahlawan. Pada *Game* ini memiliki kemampuan untuk memanipulasi partikel sebagai peluru dari senjata yang merupakan tangannya sendiri. NPC musuh NPC musuh merupakan NPC yang berperan sebagai musuh

dari Zico, dimana merupakan makhluk asing penjajah yang berniat menguasai Indonesia pada tahun 2090. NPC musuh terdiri dari beberapa macam, ada yang menyerang secara fisik langsung maupun menyerang dengan tembakan. Terdapat juga NPC musuh yang berperan sebagai pimpinannya, sehingga memiliki beberapa kemampuan unik. NPC baik NPC baik adalah NPC yang berperan sebagai rakyat Indonesia, dapat berupa pejuang maupun netral. NPC baik pada beberapa misi merupakan sandera perang. Potongan Pancasila Potongan Pancasila adalah potongan dari lambing Pancasila yang mana menjadi objektif pada setiap misi. *Player* diharuskan dapat mengumpulkan kelima potongan Pancasila agar dapat menyelesaikan *game* ini.

3.2 Perancangan Metode

Pada NPC akan diberikan beberapa *task*. Setiap NPC akan memiliki *task* serta prioritas dari *task* tersebut berbeda-beda. Berikut merupakan *task* pada NPC diurutkan dari yang biasa sampai yang diprioritaskan.

a. *Patrol*

task default, yaitu *task* dimana NPC melakukan *patroli*. Jika jarak NPC terhadap *player* lebih dari 20, serta HP NPC lebih dari 80.

b. *Pursue*

yaitu *task* dimana NPC mengejar *Player* tanpa menyerangnya. Jika jarak NPC terhadap *player* lebih dari 10 dan kurang dari 20. Dengan HP lebih dari 80.

c. *Range Attack*

yaitu *task* dimana NPC melakukan serangan berupa tembakan, serta mendekat kearah *player*. Dengan damage 1 pada setiap serangan. Jika jarak NPC terhadap *player* lebih dari 4 dan kurang 10 dan HP NPC lebih dari 40.

d. *Melee Attack*

task dimana NPC melakukan serangan jarak dekat. Dengan damage 3 pada setiap serangan. Jika NPC terhadap *player* kurang dari 4 dan HP NPC lebih dari 40.

e. *Unique Skill*

yaitu *task* dimana NPC melakukan serangan berupa asap racun, Dengan radius 10. Jika cooldown selama 20 detik selesai. Dengan jarak NPC terhadap *player* kurang dari 10.

f. *Flee*

yaitu *task* dimana NPC melarikan diri dari *player*. Jika HP NPC kurang dari 40 dan jarak NPC terhadap *player* kurang dari 40.

g. *Heal*

yaitu *task* dimana NPC melakukan pemulihan HP. Jika HP NPC kurang dari 80

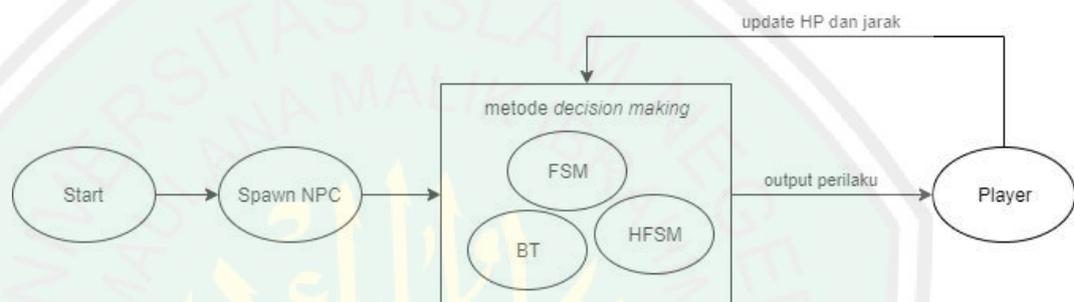
h. *Explode*

yaitu *task* dimana NPC meledakan diri serta memberikan damage dengan radius 10 dan damage 8. Jika HP NPC 0. Setelah memasuki *state* ini NPC akan langsung mati.

Berdasarkan *task* yang dimiliki oleh NPC tersebut, behavior NPC yang diharapkan dapat berlaku sebagai berikut. *Dead* merupakan *task* yang paling prioritas, jikalau HP NPC mencapai 0 atau dibawahnya, maka NPC akan langsung mengeksekusi *task* ini, tidak terkecuali apapun *task* sebelumnya. Setelah *dead task* berikutnya yang menjadi prioritas adalah *explode*, *task* ini merupakan perintah agar NPC melakukan bunuh diri, dengan meledakan dirinya sendiri serta memberi damage kepada *player*, setelah *task* ini dieksekusi, maka *task* selanjutnya harus *dead*. Kemudian *task flee*, jika HP low (berkisar 10% kebawah), maka NPC melarikan diri, jika sudah memiliki jarak yang jauh dari *Player*, maka NPC akan mengeksekusi *task healing up*, dimana *player* melakukan regenerasi HP nya sampai

80%, jika saat *task* ini dieksekusi jarak dengan *player* < 30 , maka NPC akan mengeksekusi *task flee*. Kemudian *task Attack*, *task* ini terdiri dari 3 jenis, berupa *melee attack* dengan jarak serang pendek dan damage lebih besar, *range attack* memiliki jarak lebih besar tetapi damage yang lebih kecil dan *unique skill*, *skill* ini berupa *attack* unique yang memiliki damage paling besar dengan jarak serang tidak terlalu besar, tetapi memiliki cooldown yang lama.

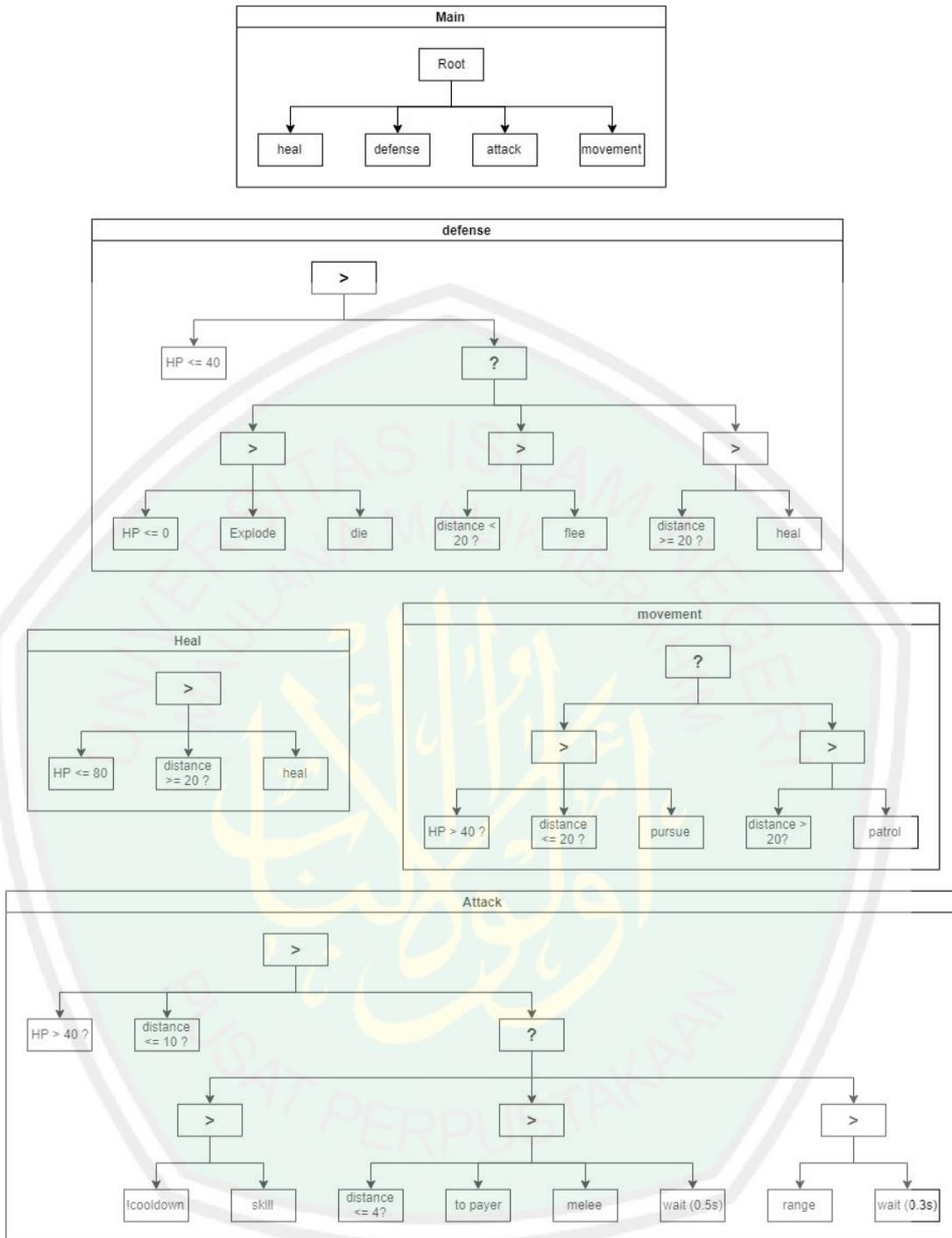
Berikut merupakan alur dari NPC.



Gambar 3.1 Alur Perilaku NPC

NPC muncul menggunakan metode *Behaviour tree*. Kemudian NPC mengecek informasi berupa HP nya dan jarak ke *Player*. Berdasarkan input tersebut NPC memilih perilaku yang sesuai. Proses eksekusi perilaku dan update informasi terus berulang.

Pada penelitian ini antara ketiga metode diharuskan untuk memiliki output yang sama sehingga dapat dibandingkan dengan perbedaan sekecil mungkin.



Gambar 3.2 Behaviour Musuh Dalam Bentuk Behaviour tree

Node akan di *ticked* dari arah kiri ke kanan, dari arah atas ke bawah. Node akan dicek dan dijalankan. Node *Root* adalah node yang tidak memiliki *parent*. Node dengan simbol “>” adalah sequence dan node dengan simbol “?” adalah node selector. Untuk node dengan akhiran tanda tanya “...?” merupakan condition

node, node ini berfungsi untuk mengecek kondisi. Jika kondisi terpenuhi akan memberikan kembalian *true*. Dan jika salah maka akan memberikan kembalian *false*. Untuk node tanpa tanda tersebut merupakan *action* atau *task* atau *leaf node*, yaitu node yang berisi fungsi perilaku. Contoh node ini adalah *patrol*.

Contoh untuk *node defense* pada behaviour tree. Node pertama adalah *sequence*, yang mana berarti jika *child* dari *sequence* bernilai *false* maka *sequence* selesai, node akan berpindah kepada node setelah *sequence* tersebut, yang mana pada root berupa node *attack*. Pada node *defense*, pertama dicek node *condition* “apakah HP lebih kecil sama dengan dari 40?”. jika benar maka akan masuk pada node *selector*, pada node jika terdapat *child* yang bernilai *false*, maka akan berpindah pada node *child* setelahnya. *Child* dari node *selector* ini adalah 3 *sequence*. *Sequence* akan berakhir dan pindah pada node setelahnya jika *child* dari node *sequence* memiliki kembalian *false*. Pada node *sequence* pertama, dicek “apakah HP lebih kecil sama dengan dari 0?”. Jika salah maka akan memiliki kembalian *false* dan akan dicek node *sequence* kedua. Pada node kedua, “apakah jarak NPC ke player lebih kecil sama dengan dari 20?”. Jika memiliki kembalian *true*, maka akan dieksekusi task *flee*. yang mana dapat disimpulkan sebagai berikut. HP lebih kecil sama dengan dari 40 dan jarak terhadap player kurang dari 20 maka NPC akan kabur (*flee*).

3.3 Rencana Pengujian

Terdapat beberapa langkah pada pengujian. Langkah pertama, ketiga NPC yang memiliki metode berbeda tersebut divalidasi kesamaan perilakunya menggunakan *black box testing*, sehingga dapat diketahui perbedaan pengeksekusian task dari metode. Pengujian ini diperlukan agar pemilihan task pada perilaku tidak berbeda pada setiap metode behaviour. Sehingga setiap metode behaviour berada pada kondisi *input output* yang sama.

Langkah kedua, membandingkan rata-rata *time complexity* dari CPU dalam memproses metode tersebut dengan jumlah NPC yang berbeda, untuk mengukur kemampuan pengolahan *script* dengan jumlah yang besar. Pengujian ini hanya pada CPU, karena proses eksekusi *script* hanya membutuhkan CPU, tanpa pengaruh memori, khususnya pada *game engine* Unity.

NPC pada penelitian ini dikatakan adaptif jika dapat mencapai target dari peneliti, yaitu dapat mengeksekusi task yang sesuai pada desain dari behaviour NPC tersebut.

BAB 4

HASIL DAN PEMBAHASAN

Pada bab ini akan membahas hasil pembuatan *game* 2090 dalam usaha mendidik moral pancasila. Pada bab ini pula dibahas validasi rancangan yang telah didesain pada bab sebelumnya. Sehingga secara transparan dapat diketahui bagaimana proses perkembangan *game* ini dilakukan.

4.1 Implementasi Desain

Pada subbab ini akan dijelaskan pengimplementasian yang telah dilakukan dalam *game*, terdiri dari pengimplementasian pada grafis visual dan metode yang digunakan.

4.1.1 Perangkat yang Digunakan

Uji coba dilakukan pada PC dengan spesifikasi perangkat keras dan perangkat lunak sebagai berikut.

1. Processor : Intel Core i5-4210U up to 2.7 GHz
2. RAM : 8 GB
3. Storage : 500 GB HDD
4. Operating System : Windows 10 Pro 64-bit
5. *Game* Engine : Unity 3D 2019.2.8f1
6. 3D Asset : Blender and Maya (FBX)
7. *Script* Editor : Microsoft Visual Studio 2019
8. 2D concept : Photoshop CC 2017

4.1.2 Tampilan Antarmuka *Game*

Berikut tampilan UI serta *gameplay* pada *game* ini.

Menu utama memiliki 3 button. Play untuk menuju gamplay, Pancasila untuk membuka laman tentang pengetahuan Pancasila.



Gambar 4.1 Menu Utama

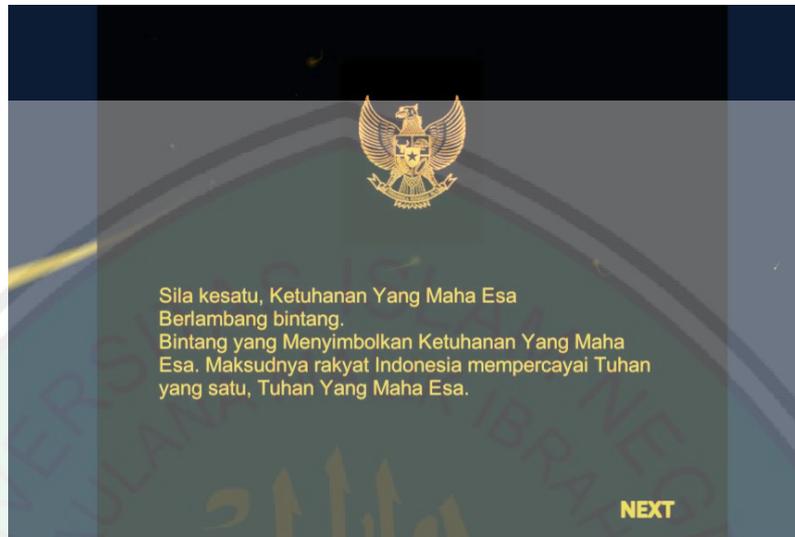
Pada laman Pancasila memuat berbagai macam pengetahuan dasar Pancasila.

Yang pada awal laman akan memperlihatkan 5 sila pada Pancasila.



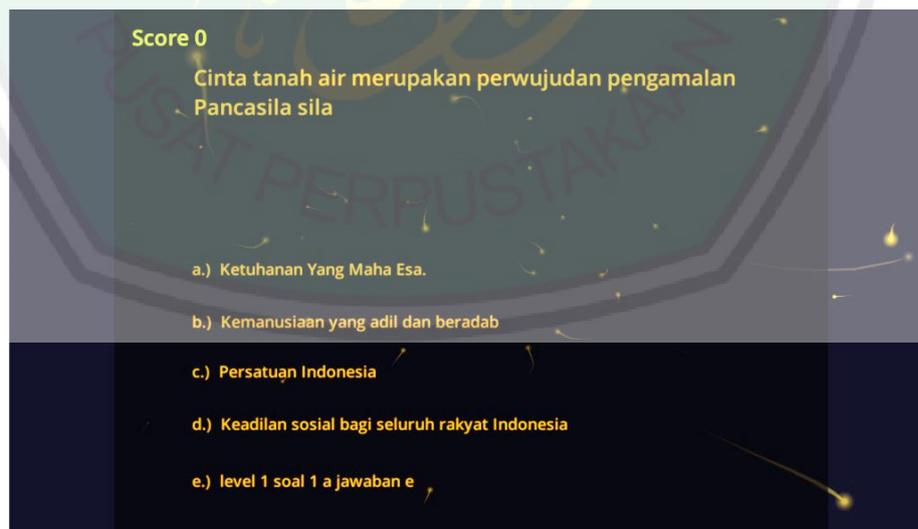
Gambar 4.2 Tampilan Laman Pancasila

Setelah masuk pada menu utama. Kemudian menekan button play, maka akan masuk pada laman pembuka level *game*, dimana membahas sila pada level tersebut, serta objektif dari level.



Gambar 4.3 Tampilan Pembuka Level

Kemudian ketika menekan button next, maka akan masuk pada laman butir-butir pada sila tersebut. Yang memuat contoh dari moral serta etika yang sesuai dengan sila berdasar pada level.



Gambar 4.4 Tampilan Soal

Pada game ini terdapat 2 level, level pertama membahas Pancasila sila pertama dan kedua. Level kedua membahas Pancasila ketiga, keempat dan kelima.

berikut merupakan tampilan dari *gameplay*.



Gambar 4.5 Tampilan *Gameplay* level 1

Pada level pertama berlokasi pada pulau apung berada pada lautan lepas. Dan pada level kedua berlokasi pada perumahan industry yang sudah rusak dan terendam banjir.



Gambar 4.6 Tampilan *Gameplay* level 2

Pada kedua level memiliki misi yang sama, yaitu membakar propaganda. Tetapi memiliki soal pada akhir game yang berbeda.

4.2 Implementasi Metode

Pada *Game* ini, *input* berupa HP, jarak dan *cooldown skill*. Berikut merupakan proses implementasi metode pada NPC berdasarkan prioritas dan alur dari keputusan menggunakan metode *Behaviour tree*.

Pada tahap awal, dibangun terdahulu *tasks* yang dibutuhkan oleh NPC, diurutkan berdasarkan event atau input sebagai trigger *action*, serta prioritas dari setiap *action*. Setelah dibangun *task* yang diperlukan, maka bangun sistem proses dari metode tersebut.

Pada metode *behaviour tree* saya menggunakan *script C#*. menggunakan *script* saya membangun satu kelas induk sebagai tempat proses logika dari *behaviour tree*, satu kelas untuk menyusun dan mengatur *behaviour* dan kelas *behaviour* untuk mengisi fungsi dari aksi NPC yang digunakan.

Berikut merupakan gambaran dari *script* yang dibangun.

a. Task Patrol

```
void GetDestinationPatrol()
{
    Vector3 testposition = (transform.position + (transform.forward * 4f)) + new Vector3(x:
UnityEngine.Random.Range(-4.5f, 4.5f), y: UnityEngine.Random.Range(-4.5f, 4.5f), z:
UnityEngine.Random.Range(-4.5f, 4.5f));

    _destination = new Vector3(testposition.x, testposition.y, testposition.z);

    _direction = Vector3.Normalize(_destination - transform.position);
    _direction = new Vector3(_direction.x, _direction.y, _direction.z);
    _desiredRotation = Quaternion.LookRotation(Vector3.Lerp(this.transform.position,
_direction, 1f));
    //output arah secara random tanpa gerak
    inpoState = "Patrol";
}
```

b. Task Pursue

```
void GetTowardPlayer()
{
    _destination = player.transform.position;
    _direction = Vector3.Normalize(_destination - transform.position);
    _direction = new Vector3(_direction.x, y: 0f, _direction.z);
    _desiredRotation = Quaternion.LookRotation(Vector3.Lerp(this.transform.position,
_direction, 4f));
    inpoState = "Toward Player (pursue)";
}
```

c. Task Bergerak

```
public void Move()
{
    //transform.rotation = _desiredRotation; //kaku buat rubah direction
    Debug.DrawLine(transform.position, _destination, IsPathEnded() ? Color.red :
Color.green);
    if (IsPathEnded())
    {
        GetDestinationPatrol();
    }

    transform.rotation = Quaternion.Lerp(this.transform.rotation, _desiredRotation,
Time.deltaTime * 2);
    transform.Translate(Vector3.forward * Time.deltaTime * moveSpeed);
}
```

d. Mengecek Jalan

```
bool IsPathEnded() //didepan ga ada navmesh
{
    blocked = NavMesh.Raycast(transform.position, _destination, out navHit,
NavMesh.AllAreas);
    return blocked;
}
```

e. Task Meledak

```

public void StartExplode(GameObject Explosive)
{
    inpoState = "Explode";
    float explosionRadius = 20f;
    float explodeDamage = 15f;
    Collider[] colliders = Physics.OverlapSphere(transform.position, explosionRadius);
    foreach (Collider collider in colliders)
    {
        if (collider.tag == "Player") //tag jgn lupa wkww
        {
            Damage(collider.transform, explodeDamage);
        }
    }
    GameObject b = Instantiate(Explosive, transform.position, transform.rotation);
}

```

f. Task Flee

```

void GetDestinationFlee() //BELUM KELARR
{
    if (pPlayer == null)
    {
        return;
    }
    Vector3 dirtopLayer = Vector3.Normalize(player.transform.position - transform.position);
    //arah ke player
    Vector3 testposition = (transform.position + (-dirtopLayer)) + new Vector3(x:
    UnityEngine.Random.Range(-1.5f, 1.5f), y: UnityEngine.Random.Range(-4.5f, 4.5f), z:
    UnityEngine.Random.Range(-1.5f, 1.5f));

    _destination = new Vector3(testposition.x, y: 1f, testposition.z);

    _direction = Vector3.Normalize(_destination - transform.position);
    _direction = new Vector3(_direction.x, y: 0f, _direction.z); // ga mikirin tinggi dulu
ya...
    _desiredRotation = Quaternion.LookRotation(Vector3.Lerp(this.transform.position,
    _direction, 1f));
}

```

g. Task Melee Attack

```

void MeleeAttack()
{
    //anim.SetBool("Attack", true);
    float attackDamage = 3;
    MeleTimeAttack = 1;

    inpoState = "Melee Attack";

    Damage(player.transform, attackDamage);
    GameObject b = Instantiate(MeleeAttackParticle, gun.transform.position,
    gun.transform.rotation);
    //}
}

```

h. *Task Range Attack*

```

void RangeAttack()
{
    GameObject b = Instantiate(enemybullet, gun.transform.position,
enemybullet.transform.rotation);
    //b.GetComponent<Rigidbody>().AddForce(gun.transform.forward * 1);
    EnemyBullet bullet = b.GetComponent<EnemyBullet>();

    inpoState = "Range Attack";

    if (bullet != null)
        bullet.Seek(player.transform);
}

```

i. *Task Skill*

```

public void UniqueSkill()
{
    GameObject b = Instantiate(Skill, gun.transform.position, Skill.transform.rotation);
    //player take damage
    Collider[] colliders = Physics.OverlapSphere(transform.position, 8);
    foreach (Collider collider in colliders)
    {
        if (collider.tag == "Player") //tag jgn lupa wkkwk
        {
            Damage(collider.transform, 5);
        }
    }
    inpoState = "Skill";
    cd = 0;
}

```

j. *Task menerima damage*

```

public void TakeDamage(float dmg)
{
    GameObject b = Instantiate(TakeDamageParticle, gun.transform.position,
TakeDamageParticle.transform.rotation);
    HP -= dmg;
}

```

k. *Task memberi damage*

```

void Damage(Transform player, float dmg)
{
    if (player.GetComponent<MoveBehaviour>().GetHP() == 0)
    {
        return;
    }

    if (HP <= 0)
    {
        return;
    }
    MoveBehaviour PlayerHealth = player.GetComponent<MoveBehaviour>();
    PlayerHealth.TakeDamage(dmg);
}

```

4.2.1 Behaviour tree (BT)

Behaviour tree dibangun menggunakan *script*. Memiliki kelas induk berupa *static class* yang tidak *extends* pada monobehaviour. Kelas induk ini akan dipanggil oleh *script* yang digunakan untuk mengatur task yang telah dibuat pada subab sebelumnya. Pada kelas induk, akan dijabarkan bagaimana sequence, selector serta komponen lainnya pada BT bekerja.

Kelas yang memanggil kelas induk ini. Berupa kelas behaviour dari NPC, dimana otak behaviour tersebut adalah kelas induk, serta motorik dari kelas behaviour ini adalah kelas yang berisi task.

Berikut merupakan kelas induk *Behaviour tree* (BT).

a) Fungsi *public* yang dipanggil dari kelas *child*

```
public static class BT
{
    public static Root Root() { return new Root(); }
    public static Sequence Sequence() { return new Sequence(); }
    public static Selector Selector(bool shuffle = false) { return new Selector(shuffle); }
    public static Action RunCoroutine(System.Func<IEnumerator<BTState>> coroutine) { return
new Action(coroutine); }
    public static Action Call(System.Action fn) { return new Action(fn); }
    public static ConditionalBranch If(System.Func<bool> fn) { return new
ConditionalBranch(fn); }
    public static While While(System.Func<bool> fn) { return new While(fn); }
    public static Condition Condition(System.Func<bool> fn) { return new Condition(fn); }
    public static Repeat Repeat(int count) { return new Repeat(count); }
    public static Wait Wait(float seconds) { return new Wait(seconds); }
    public static Trigger Trigger(Animator animator, string name, bool set = true) { return
new Trigger(animator, name, set); }
    public static WaitForAnimatorState WaitForAnimatorState(Animator animator, string name,
int layer = 0) { return new WaitForAnimatorState(animator, name, layer); }
    public static SetBool SetBool(Animator animator, string name, bool value) { return new
SetBool(animator, name, value); }
    public static SetActive SetActive(GameObject gameObject, bool active) { return new
SetActive(gameObject, active); }
    public static WaitForAnimatorSignal WaitForAnimatorSignal(Animator animator, string
name, string state, int layer = 0) { return new WaitForAnimatorSignal(animator, name, state,
layer); }
    public static Terminate Terminate() { return new Terminate(); }
    public static Log Log(string msg) { return new Log(msg); }
    public static RandomSequence RandomSequence(int[] weights = null) { return new
BTAI.RandomSequence(weights); }
}
```

Kelas induk merupakan kelas dimana *Behaviour tree* di definisikan dan dideskripsikan. Berisi bermacam fungsi yang memuat cara kerja dari *action* node *condition* node, *decorator* dan *control flow*.

b) fungsi *control flow sequence*

```
public class Sequence : Branch
{
    public override BTState Tick()
    {
        var childState = children[activeChild].Tick();
        switch (childState)
        {
            case BTState.Success:
                activeChild++;
                if (activeChild == children.Count)
                {
                    activeChild = 0;
                    return BTState.Success;
                }
                else
                    return BTState.Continue;
            case BTState.Failure:
                activeChild = 0;
                return BTState.Failure;
            case BTState.Continue:
                return BTState.Continue;
            case BTState.Abort:
                activeChild = 0;
                return BTState.Abort;
        }
    }
}
```

Fungsi *control flow* terdapat dua macam, yaitu *sequence* dan *selector*. *Sequence* bersifat *And*, yang mana akan memberikan return *true* jika semua anak bernilai *true*, sehingga jika terdapat satu anak yang *false* maka proses *sequence* akan berakhir dan melanjutkan pada proses selanjutnya.

c) fungsi *control flow selector*

```

public class Selector : Branch
{
    public Selector(bool shuffle)
    {
        if (shuffle)
        {
            var n = children.Count;
            while (n > 1)
            {
                n--;
                var k = Mathf.FloorToInt(Random.value * (n + 1));
                var value = children[k];
                children[k] = children[n];
                children[n] = value;
            }
        }
    }
}

```

Berbeda dengan sequence, selector memiliki sifat *Or* yang mana jika anak bernilai *false*, maka akan mengecek anak selanjutnya pada node.

d) fungsi *condition*

```

public class Condition : BTNode
{
    public System.Func<bool> fn;

    public Condition(System.Func<bool> fn)
    {
        this.fn = fn;
    }
    public override BTState Tick()
    {
        return fn() ? BTState.Success : BTState.Failure;
    }

    public override string ToString()
    {
        return "Condition : " + fn.Method.ToString();
    }
}

```

fungsi condition merupakan fungsi yang memberikan return Boolean berupa *true* dan *false*. Digunakan untuk memberikan suatu kondisi terhadap control flow agar dapat berjalan.

e) fungsi *Action*

```

public class Action : BNode
{
    System.Action fn;
    System.Func<IEnumerator<BTState>> coroutineFactory;
    IEnumerator<BTState> coroutine;
    public Action(System.Action fn)
    {
        this.fn = fn;
    }
    public Action(System.Func<IEnumerator<BTState>> coroutineFactory)
    {
        this.coroutineFactory = coroutineFactory;
    }
    public override BTState Tick()
    {
        if (fn != null)
        {
            fn();
            return BTState.Success;
        }
        else
        {
            if (coroutine == null)
                coroutine = coroutineFactory();
            if (!coroutine.MoveNext())
            {
                coroutine = null;
                return BTState.Success;
            }
            var result = coroutine.Current;
            if (result == BTState.Continue)
                return BTState.Continue;
            else
            {
                coroutine = null;
                return result;
            }
        }
    }
    public override string ToString()
    {
        return "Action : " + fn.Method.ToString();
    }
}

```

Fungsi *action* adalah fungsi dimana task dieksekusi, fungsi ini juga disebut dengan leaf node. Fungsi ini diisi dengan fungsi yang berfungsi sebagai perilaku dari NPC, memberikan return success atau continue. Success jika task berhasil dieksekusi dan continue jika task masih dalam proses eksekusi.

Berikut merupakan kelas *behaviour* dalam bentuk *Behaviour tree* yang digambarkan dalam *script*. Dimana desain sesuai dengan bentuk gambar pada bab tiga.

```

Root.OpenBranch(
    BT.Sequence().OpenBranch( //defense
        BT.Condition(() => { return m_Drone.HPdrone <= 40; }},
        //BT.Call(m_Drone.a),
        BT.Selector().OpenBranch(
            BT.Sequence().OpenBranch(
                BT.Condition(() => { return m_Drone.Distance < 20; }},
                BT.Call(m_Drone.Flee)
            ),
            BT.Sequence().OpenBranch(
                BT.Condition(() => { return m_Drone.HPdrone <= 0; }}, //mati
                BT.Call(m_Drone.StartExplode),
                BT.Call(m_Drone.Die)
            ),
            BT.Sequence().OpenBranch(
                BT.Condition(() => { return m_Drone.Distance >= 20; }}, //heal
                BT.Call(m_Drone.Heal)
            )
        )
    ),
    BT.Sequence().OpenBranch(
        BT.Condition(() => { return m_Drone.HPdrone <= 80; }},
        BT.Condition(() => { return m_Drone.Distance >= 20; }},
        BT.Call(m_Drone.Heal)
    ),
    BT.Sequence().OpenBranch( //attack ...contoh konkrit selector
        BT.Condition(() => { return m_Drone.HPdrone > 40; }},
        BT.Condition(() => { return m_Drone.Distance <= 10; }},
        BT.Selector().OpenBranch(
            BT.Sequence().OpenBranch(
                BT.Condition(() => { return m_Drone.CD >= 20; }},
                BT.Call(m_Drone.UniqueSkill)
            ),
            BT.Sequence().OpenBranch(
                BT.Condition(() => { return m_Drone.Distance <= 4; }}, //melee
                BT.Call(m_Drone.GetTowardPlayer),
                BT.Call(m_Drone.MeleeAttack),
                BT.Wait(0.5f)
            ),
            BT.Sequence().OpenBranch(
                BT.Call(m_Drone.RangeAttack), // nembak
                BT.Wait(0.3f)
            )
        )
    ),
    // termasuk dalam range attack
    BT.Sequence().OpenBranch( // pursue
        BT.Condition(() => { return m_Drone.HPdrone > 40; }},
        BT.Condition(() => { return m_Drone.Distance <= 20; }},
        BT.Call(m_Drone.GetTowardPlayer)
    ),
    BT.Sequence().OpenBranch( //patrol
        BT.Condition(() => { return m_Drone.Distance > 20; }},
        BT.Condition(() => { return m_Drone.HPdrone > 40; }},
        BT.Wait(1f),
        BT.Call(m_Drone.GetDestinationPatrol)
    )
);

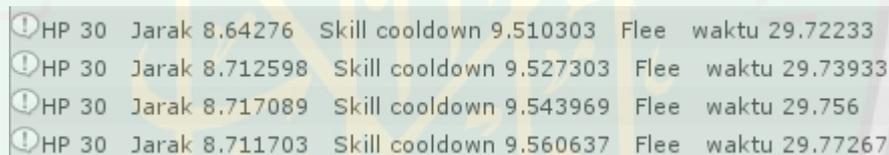
```

4.3 Pengujian

Pengujian melalui 2 tahap, pertama pengujian ketepatan perilaku. Pengujian ini dilakukan agar dapat memastikan bahwa pengambilan keputusan oleh NPC menggunakan 3 metode tersebut Memiliki output yang sama. Pengujian ini menggunakan console yang terdapat pada Unity. Kedua, pengujian *time complexity* CPU dalam menjalankan algoritma metode. Pengujian ini untuk membandingkan jumlah waktu yang dibutuhkan oleh CPU untuk mengeksekusi algoritma pada beberapa scenario yang berbeda.

4.3.1 Uji Output Perilaku

Pengujian kesesuaian perilaku menggunakan metode blackbox testing. Pengujian ini secara langsung menggunakan console log pada unity.



```

!HP 30 Jarak 8.64276 Skill cooldown 9.510303 Flee waktu 29.72233
!HP 30 Jarak 8.712598 Skill cooldown 9.527303 Flee waktu 29.73933
!HP 30 Jarak 8.717089 Skill cooldown 9.543969 Flee waktu 29.756
!HP 30 Jarak 8.711703 Skill cooldown 9.560637 Flee waktu 29.77267
  
```

Gambar 4.7 event dan output NPC pada Console Log

Berikut merupakan beberapa sampel dari event dan perilaku yang dihasilkan oleh ketiga metode. Testing secara langsung menggunakan console.

Tabel 4.1 Tabel keputusan NPC BT

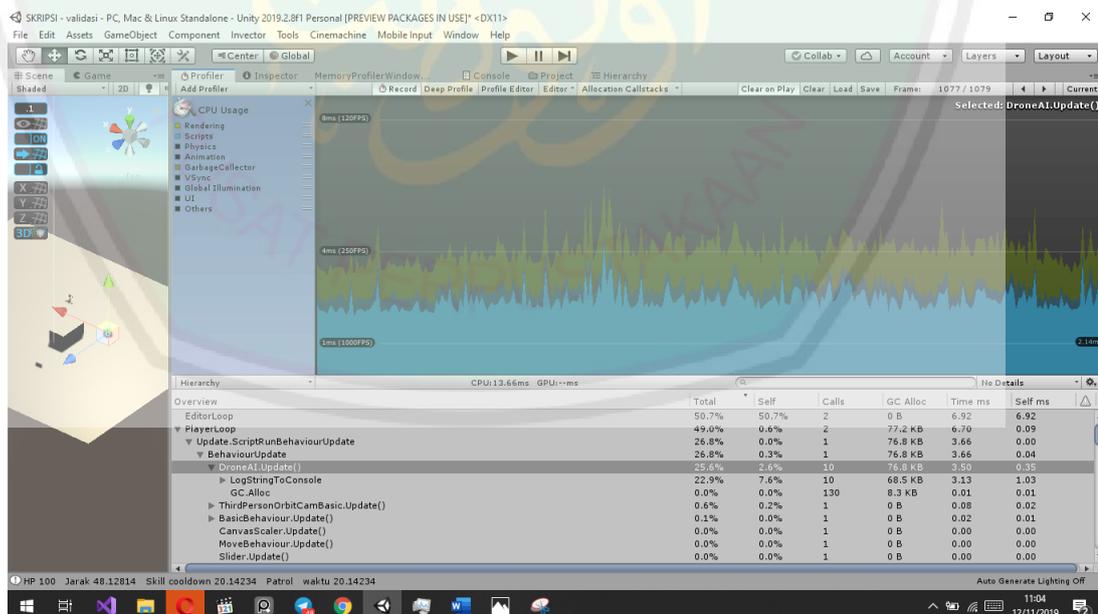
No	HP	Jarak (meter)	Skill (detik)	Output perilaku
1	90	12.32	5.98	<i>Pursue</i>
2	100	14.87	9.82	<i>Pursue</i>
3	80	5.32	14.58	<i>Range</i>
4	60	20.12	20.4	<i>Heal</i>
5	40	2.65	5.99	<i>Melee</i>
6	50	12.33	9.83	<i>Pursue</i>

7	70	14.88	14.59	<i>pursue</i>
8	70	5.33	20.5	<i>Range</i>
9	90	20.13	5.10	<i>Patrol</i>
10	0	6.66	9.84	<i>Explode</i>
11	10	12.34	14.60	<i>flee</i>

Dari hasil pengujian peneliti dapat menyimpulkan bahwa output perilaku yang dikeluarkan oleh ketiga jenis NPC tersebut adalah sama.

4.3.2 Uji Time Complexity CPU

Pengujian ini dilakukan menggunakan Unity Editor, menggunakan profiler yang telah disediakan oleh unity sendiri. Pada profiler output dari kalkulasi peforma berupa FPS yang dapat dicapai saat *game* berjalan. Salah satu komponen dalam mendongkrak FPS adalah CPU. pada penelitian ini saya mendapati *Script* menjadi penyumbang kedua terbesar setelah rendering dalam penggunaan CPU.



Gambar 4.7 perbandingan penggunaan CPU untuk rendering dan *Scripts*

Berdasarkan cara percobaan tersebut peneliti mengambil sampel pada setiap percobaan. Tipe data berupa *millisecond* (ms). Yaitu jumlah waktu yang dibutuhkan oleh CPU untuk menjalankan *script* tersebut.

**Tabel 4.2 Tabel Sampel *Time Complexity* CPU
berdasar jumlah NPC**

1	10	20	30	40	50	60	70	80	90	100
0,78	4,85	7,35	11,56	8,41	13,9	19,13	21,23	27,11	32,05	41,76
1,32	2,99	5,29	8,48	10,1	16,46	15,34	15,79	21,15	27,28	30,9
0,89	3,99	5,28	9,01	12,49	18	18,29	24,73	21,48	21,55	36,7
1,03	5,14	5,98	7,27	11,64	15,59	13,02	23,6	19,46	32,33	43,76
0,55	5,73	7,31	9,92	12,07	18,16	15,2	24,86	28,81	28,7	42,65
0,98	4,84	6,04	9,28	12,69	13,58	14,78	17,87	26,67	30,32	32,11
1,07	4,13	7,45	7,97	8,27	16,45	20,49	19,76	18,22	28,47	41,57
0,49	4,23	6,38	8,93	11,72	16,63	19,1	19,66	21,7	21,37	37,06
0,55	4,67	5,07	9,47	12,45	13,57	15,39	17,99	17,71	27,68	32,23
1,39	3,6	6,97	9,8	9,53	18,48	16,61	22,17	22,9	26,96	28,24
0,98	5,39	5,97	9,98	10,08	15,44	17,64	16,58	17,52	22,09	42,03
1,28	3,21	5,54	9,35	11,07	13,73	21,11	21,75	26,22	23,24	23,75
0,56	2,67	5,93	7,12	10,94	18,29	15,51	25,28	21,25	24,19	42,06
0,78	5,09	7,39	10,38	10,64	13,7	20,09	25,43	29,31	24,59	36,64
0,8	3,76	4,73	11,6	14,48	11,42	21,46	24,46	27,15	23,3	23,38
0,48	4,21	5,06	11,21	12,08	16,76	18,97	22,29	25,35	24,76	37,54
Rata-rata										
0.87	4.28	6.1	9.45	11.16	15.63	17.62	21.46	23.25	26.18	35.77

Pada saat time complexity CPU mencapai 16 ms, maka game akan menyentuh 60 FPS. Dan ketika game mencapai 33 ms, maka game akan menyentuh 30 FPS.

Pada gambar berikut. Dapat dilihat bahwa waktu yang dibutuhkan oleh CPU dalam mengeksekusi *script* NPC hanya sebagian kecil dari waktu yang dibutuhkan oleh CPU untuk mengeksekusi *script* player.



Gambar 4.8 perbedaan waktu yang dibutuhkan dalam eksekusi *script*

4.4 Integrasi Islam

Sangat jelas tidak ada pertentangan antara Islam dan ilmu pengetahuan. Islam sebagai agama yang damai tidak membatasi suatu pemikiran akan ilmu pengetahuan. Tidak memaksakan pemahaman pada kaum intelektual. sebagai agama yang memiliki kitab suci Al-Quran, kitab yang menjelaskan bagaimana proses makhluk hidup dilahirkan sebelum ilmu pengetahuan melakukan hipotesa evolusinya. Kitab yang menjelaskan bagaimana setiap makhluk hidup mendapat rezeki nya, sebelum ilmu pengetahuan menyadari akan keseimbangan penciptaanya. Tidak ada pertentangan melainkan perbedaan hipotesa pemikir sebagai makhluk dengan fakta yang Al-Quran ungkapkan.

Bukti dari damainya agama islam dalam menanggapi perbedaan pemikiran. Sebagaimana dalam firman Allah pada surah An-Nahl ayat 125.

ادْعُ إِلَى سَبِيلِ رَبِّكَ بِالْحُكْمَةِ وَالْمَوْعِظَةِ الْحَسَنَةِ ۗ وَجَادِلْهُمْ بِالَّتِي هِيَ أَحْسَنُ ۚ إِنَّ

رَبَّكَ هُوَ أَعْلَمُ بِمَنْ ضَلَّ عَنْ سَبِيلِهِ ۗ وَهُوَ أَعْلَمُ بِالْمُهْتَدِينَ

Artinya: *Serulah (manusia) kepada jalan Tuhanmu dengan hikmah dan pelajaran yang baik dan bantahlah mereka dengan cara yang baik. Sesungguhnya Tuhanmu Dialah yang lebih mengetahui tentang siapa yang tersesat dari jalan-Nya dan Dialah yang lebih mengetahui orang-orang yang mendapat petunjuk.*

(Q.S An-Nahl: 125).

Menjadi substansi perkembangan zaman, zaman dimana pengetahuan menjadi identitas diri yang hidup. Zaman ini menuntut agar manusia terus belajar, tidak batasi hanya pengetahuan agama semata. Terlebih pengertian ilmu pengetahuan umum yang dipisahkan dengan pengetahuan agama hanyalah pemisah yang dibuat oleh manusia (Mustopo, 2017). Dimana menjadi muslim merupakan tindakan yang *kaffah* yang artinya utuh, Tidak hanya sebahagian. Tidak dibatasi oleh pemikiran manusia akan perbedaan ilmu pengetahuan. Yang mana ilmu pengetahuan itu sendiri berupa ilmu yang bertujuan untuk mengenal lebih dekat kepada pencipta dan ciptaanya.

Pancasila berlaku sebagai aturan, norma, moral dan etika tidak dapat menyaingi aturan dari tuhan. Sehingga pada titik tertentu aturan tuhan harus lebih diutamakan. Sebagai buah pemikiran masyarakat Indonesia, yang didominasi oleh muslim, maka pada poin Pancasila tertuang pula pemikiran aturan seorang yang bertuhan. Sehingga tidak didapati pada setiap sila-nya yang bertentangan dengan aturan tuhan.

Dapat kita lihat bagaimana game mempengaruhi pola pikir, moral etika dan pengetahuan masyarakat Indonesia. Banyak generasi muda yang terpengaruh olehnya. Game sebagai media, dapat pula digunakan sebagai mediasi untuk hal yang positif. Kembali pada ayat Al-Quran yang tertera di atas.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan game yang telah dibuat dan diuji pada penelitian ini, peneliti menyimpulkan sebagai berikut.

1. Behaviour tree mampu menciptakan NPC yang adaptif pada keadaan tertentu. Menggunakan input berupa HP, jarak dan Cooldown skill. Memberikan output berupa pilihan perilaku, dari 9 perilaku antaranya *patrol*, *pursue*, *flee*, *heal*, *range attack*, *range attack*, *skill attack* dan *explode*. Dimana menggunakan behaviour tree dapat secara langsung mengontrol perilaku NPC.
2. Berdasarkan uji coba time complexity CPU dalam mengeksekusi *script* NPC dari metode BT. Peneliti menyimpulkan bahwa besar waktu yang dibutuhkan oleh CPU untuk mengeksekusi *script* NPC berbedanding lurus dengan jumlah NPC, semakin banyak jumlah NPC maka semakin besar pula waktu yang dibutuhkan untuk mengeksekusinya. Dan Waktu eksekusi *script* sangat dipengaruhi juga oleh *script action* pada NPC tersebut, misalkan pada *script action* untuk menyerang yang lebih rumit daripada *script* lainnya akan diproses lebih lama pula dari yang lainnya.

5.2 Saran

Peneliti sadar akan banyaknya kekurangan pada game ini. Peneliti menyimpulkan kekurangan yang dapat dikembangkan pada penelitian berikutnya, sebagai berikut.

1. Membuat level design, karakter, animasi, efek dan kebutuhan visual lainnya menjadi lebih menarik.
2. Menggunakan jumlah materi dan soal yang lebih besar, serta memberikan algoritma agar soal dapat dirandom secara optimal. Serta menambah jumlah level pada game.
3. Menggunakan *hybrid behaviour tree*, mengingat metode *behaviour tree* sedang sangat pesat perkembangannya dan menjadi default logaritma pada game engine Unreal.
4. Meneliti performa *behaviour tree* pada tingkat per perilaku NPC, sehingga dapat dimengerti optimasi pada desain *behaviour tree* yang paling sesuai dan memakan waktu eksekusi paling sedikit.

DAFTAR PUSTAKA

- As-suyuthi, J., Muhammad, J. and Ahmad, I. (1505) 'Tafsir jalalain', pp. 1–402.
- Asmaroini, A. P. (2017) 'MENJAGA EKSISTENSI PANCASILA DAN PENERAPANNYA BAGI MASYARAKAT DI ERA GLOBALISASI', 1(2).
- Bimantoro, T. and Haryanto, H. (2016) 'Pemodelan Perilaku Musuh Menggunakan Finite State Machine (FSM) Pada *Game* Pengenalan Unsur Kimia', 1(3), pp. 210–219.
- Chang, K. and Zhu, D. (2017) 'Hierarchical Finite State Machine (HFSM) Problems of FSM possible Transitions'.
- Colledanchise, M. and Ögren, P. (2017) 'Behavior Trees in Robotics and AI: An Introduction'. Available at: <http://arxiv.org/abs/1709.00084>.
- Fauzi, R. (2015) *NPC BEHAVIOR VARIATION AT COMBAT DEFENSE USING HIERARCHICAL FINITE STATE MACHINE (HFSM)*.
- Iskarim, M. (2016) 'Dekadensi Moral di Kalangan Pelajar (Revitalisasi Strategi PAI dalam Menumbuhkan Moralitas Generasi Bangsa)', 1, pp. 1–20.
- Kirby, N. (2011) *Introduction to Game AI, Learning*.
- Marcotte, R. and Hamilton, H. J. (2017) 'Behavior Trees for Modelling Artificial Intelligence in *Games: A Tutorial*', *The Computer Games Journal*. Springer New York, 6(3), pp. 171–184. doi: 10.1007/s40869-017-0040-9.
- Marcotte, R. K. (2017) 'Modelling Artificial Intelligence in *Games* Using Mindset Behavior Trees', (May).
- Mcquillan, K. (2015) 'A Survey of *Behaviour trees* and their Applications for *Game AI*'.
- Mustopo, A. (2017) 'Integrasi agama dan ilmu pengetahuan', *al-Afkar*, 5, no.2.
- Nurhaidah, M. I. M. (2015) 'Pembaharuan Dalam Pemanfaatan Hasil-Hasil Teknologi.', *Jurnal Pesona Dasar*, 3(3), pp. 1–14. Available at: <http://www.jurnal.unsyiah.ac.id/PEAR/article/view/7506>.
- Nurwardani, P. (2016) *Pendidikan pendidikan untuk perguruan tinggi*. 1st edn. jakarta.
- Olsson, M. (2016) 'Behavior Trees for decision-making in Autonomous Driving'. Available at: <http://www.divaportal.org/smash/get/diva2:907048/FULLTEXT01.pdf>.

Putra, D. W., Nugroho, A. P. and Puspitarini, E. W. (2016) 'GAME EDUKASI BERBASIS ANDROID SEBAGAI MEDIA PEMBELAJARAN UNTUK ANAK USIA DINI', 1(1), pp. 46–58.

Saini, P., Rai, S. and Jain, A. K. (2014) 'Decision Tree Algorithm Implementation Using Educational Data', 1(1).

Sekhvat, Y. A. (2017) 'Behavior Trees for Computer Games', *International Journal on Artificial Intelligence Tools*, 26(02), p. 1730001. doi: 10.1142/s0218213017300010.

