

**3D ADVENTURE GAME UNTUK PENGENALAN BUDAYA
WAYANG MENGGUNAKAN ALGORITMA *FIREFLY*
SEBAGAI PEMBANGKIT PERILAKU PADA NPC**

SKRIPSI

Oleh :
EMIL ENAN
NIM 11650042



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2018**

HALAMAN PENGAJUAN

**3D ADVENTURE GAME UNTUK PENGENALAN BUDAYA
WAYANG MENGGUNAKAN ALGORITMA *FIREFLY*
SEBAGAI PEMBANGKIT PERILAKU PADA NPC**

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 :
EMIL ENAN
NIM 11650042**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2018**

LEMBAR PERSETUJUAN

**3D ADVENTURE GAME UNTUK PENGENALAN BUDAYA
WAYANG MENGGUNAKAN ALGORITMA *FIREFLY*
SEBAGAI PEMBANGKIT PERILAKU PADA NPC**

SKRIPSI

Oleh:
EMIL ENAN
NIM 11650042

Telah Diperiksa dan Disetujui untuk Diuji :
Tanggal : 18 Mei 2018

Dosen Pembimbing I

Dosen Pembimbing II

Frey Nugroho, M.T
NIP. 19710722 201101 1 001

Fachrul Kurniawan, M.MT
NIP. 19771020 200912 1 001

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



Wahyu Crysdiyan
NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN

**3D ADVENTURE GAME UNTUK PENGENALAN BUDAYA
WAYANG MENGGUNAKAN ALGORITMA FIREFLY
SEBAGAI PEMBANGKIT PERILAKU PADA NPC**

SKRIPSI

Oleh :
EMIL ENAN
NIM.11650042

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal : Juni 2018

Susunan Dewan Penguji

Penguji Utama : Hani Nurhavati, M.T
NIP. 19780625 200801 2 006

Ketua Penguji : Yunifa Miftachul Arif, M.T
NIP. 19830616 201101 1 004

Sekretaris Penguji : Fresv Nugroho, M.T
NIP. 19710722 201101 1 001

Anggota Penguji : Fachrul Kurniawan, M.MT
NIP. 19771020 200912 1 001

Tanda Tangan

(*[Signature]*)
(*[Signature]*)
(*[Signature]*)
(*[Signature]*)

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



[Signature]
Revo Crysdiyan
NIP. 19740424 200901 1 008

HALAMAN PERNYATAAN ORISINALITAS PENELITIAN

Nama : Emil Enan
NIM : 11650042
Jurusan : Teknik Informatika
Fakultas : Sains dan Teknologi

Judul Skripsi : **3D ADVENTURE GAME UNTUK PENGENALAN
BUDAYA WAYANG MENGGUNAKAN
ALGORITMA *FIREFLY* SEBAGAI
PEMBANGKIT PERILAKU PADA NPC**

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

Malang, 2018
Yang membuat pernyataan,

MIETERAI
TEMPEL
TGL. 20
691CEAFF124848838
6000
ENAM RIBURUPIAH

Emil Enan
NIM.11650042

MOTTO

*“Kalau Mau Jadi Orang Kaya,
Maka Hiduplah Seperti Orang Kaya,
BUKAN MENCONTOH
GAYA HIDUPNYA,
TAPI CONTOHLAH
CARA MEREKA BEKERJA.”*

(Emil Enan)

HALAMAN PERSEMBAHAN

*Dengan rasa syukur seraya mengharap ridho Ilahi kupersembahkan karya ini kepada :
Bapak dan Ibu tercinta*

SABIL HANDALING & BUNGAWALI

Yang senantiasa mencurahkan kasih Sayang, perhatian, Doa, dan bimbingannya dalam setiap langkahku.

Tanpa beliau langkah ini takan pernah sampai pada kampus yang indah ini. Semoga Allah SWT melindungi dan menyayangi keduanya.

*Terimakasih untuk Bapak/Ibu Dosen **Pak Fresy** sebagai dosen pembimbing I dan **Pak Fahrul** sebagai dosen Pembimbing II yang selalu sabar membimbing saya dalam menyelesaikan tugas akhir ini. **Ibu Hani** sebagai wali dosen yang senantiasa mengawasi perkembangan perkuliahan saya selama beberapa tahun ini. Terimakasih juga kepada ibu bapak dosen pengajar, ustad ustadzah yang telah memberikan ilmu dengan keikhlasan, semoga ilmu yang telah beliau beri ini dapat bermanfaat bagi nusa, bangsa dan agama.*

Kepada Keluarga:

Alem Islamy Sabil, Siti Amelia Sabil, Albin Adyaksa Sabil, Siti Amanda Sabli, Mutiawan, Mutiaratih, Mutiasari, Mutiaratih, Mutialdi.

Kepada Sahabat:

Ifa Alif, Hamdi MUSAAD, Wim Sonevel, Catur, Achmad Ihwani, Nafian Wildan, Alriandy Putra, Ulfa, Sigid, Hisbun, Adit, Famie Indra.

yang bersama-sama saling menyemangati satu sama lain dan saling mengingatkan jika lalai.

*Kepada para **sahabat TI** angkatan 2011, yang selalu ada untuk membantu sesama. Dan kepada teman-temanku semua yang tidak bisa kusebutkan satu persatu yang selalu membantuku dan menyemangatiku di saat aku susah dan terpuruk.*

Semoga Allah SWT melindungi, menyayangi dan menempatkan mereka semuanya pada surganya kelak dan melimpahkan rezeki kepada mereka semua...

KATA PENGANTAR



Segala puji bagi Allah SWT yang Maha Pengasih lagi Maha Penyayang atas Rahmat dan Hidayah-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi ini. Sholawat serta Salam tetap tercurahkan kepada junjungan kita, kekasih Allah, Nabi Muhammad SAW, sang pemberi syafaat kelak di hari akhir, beserta seluruh keluarga, sahabat, dan para pengikutnya.

Penelitian skripsi yang berjudul **“3D Adventure Game Untuk Pengenalan Budaya Wayang Menggunakan Algoritma Firefly Sebagai Pembangkit Perilaku Pada NPC”** ini ditulis untuk memnuhi salah satu syarat guna memperoleh gelar Sarjana Strata Satu (S1) Fakultas Sains dan Teknologi Universitas Maulana Malik Ibrahim Malang. Karya penelitian skripsi ini tidak akan pernah ada tanpa bantuan baik moral maupun spiritual dari berbagai pihak yang telah terlibat. Untuk itu dengan segala kerendahan hati, penulis mengucapkan rasa terimakasih yang sebesar-besarnya kepada:

1. Prof. Dr. H. Mudjia Rahardjo, M.Si, selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Bapak Fressy Nugroho, M.T selaku Dosen Pembimbing I yang telah bersedia meluangkan waktu, tenaga dan pikiran untuk memberikan bimbingan, berbagai pengalaman, arahan, nasihat, motivasi dan pengarahan dalam pembangunan program hingga penyusunan skripsi ini.
3. Bapak Fachrul Kurniawan, M.MT selaku dosen pembimbing 2 yang selalu memberi masukan, serta pengarahan dalam penyusunan laporan skripsi ini.
4. Ibu Hani Nurhayati, M.T selaku dosen wali yang juga selalu memberi pengarahan terkait akademik selama masa study.
5. Dr. Cahyo Crysdian selaku ketua jurusan Teknik Informatika yang mendukung dan mengarahkan skripsi ini.

6. Segenap civitas akademika Fakultas Saintek, Universitas Islam Negeri Maulana Malik Ibrahim Malang terutama seluruh dosen, terimakasih atas segala ilmu dan bimbingannya.
7. Ayah, Ibu, Kakak, dan Adik serta seluruh keluarga besar yang selalu memberikan doa, kasih sayang, semangat, dukungan moril, serta motivasi sampai saat ini, terimakasih banyak

Harapan penulis semoga semua amal kebaikan dan jasa-jasa dari semua pihak yang telah membantu hingga skripsi ini selesai diterima oleh Allah SWT, serta mendapatkan balasan yang lebih baik dan berlipat ganda.

Penulis juga menyadari bahwa skripsi ini masih jauh dari kesempurnaan yang disebabkan keterbatasan Harapan penulis, semoga karya ini bermanfaat dan menambah ilmu pengetahuan bagi kita semua, Aamiin.

Malang, 25 Juni 2018
Penulis

Emil Enan

DAFTAR ISI

COVER.....
HALAMAN PENGAJUAN	ii
LEMBAR PERSETUJUAN	iii
LEMBAR PENGESAHAN.....	iv
HALAMAN PERNYATAAN	v
MOTTO.....	vi
HALAMAN PERSEMBAHAN.....	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
ABSTRAK	xvi
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	3
1.5 Manfaat Peneletian	3
BAB II TINJAUAN PUSTAKA	
2.1 Penelitian Terkait	4
2.2 Budaya Wayang.....	6
2.2.1 Macam – Macam Wayang	7
2.2.1.1 Wayang Beber	7
2.2.1.2 Wayang Purwa.....	7
2.2.1.3 Wayang Madya.....	8
2.2.1.4 Wayang Gedog	8

2.2.1.5 Wayang Menak.....	9
2.2.1.6 Wayang Babad.....	9
2.2.1.7 Wayang Modern	9
2.2.1.8 Wayang Topeng.....	10
2.3 Permainan (<i>Game</i>).....	10
2.3.1 Jenis – Jenis Permainan (<i>Game</i>)	11
2.3.1.1 PC Games	12
2.3.1.2 Console Games	12
2.3.1.3 Handheld Games.....	12
2.3.1.4 Mobile Games.....	13
2.4 Artificial Intelligence (AI) Pada Game	17
2.5 Teknik Dalam AI.....	18
2.6 Pathfinding	19
2.7 Algoritma Firefly.....	20
2.7.1 Pengaplikasian Algoritma Firefly	22
2.8 Algoritma Astar (A*)	22
BAB III RANCANGAN DAN PENERAPAN	
3.1 Perancangan Algoritma Firefly	26
3.2 Penerapan Algoritma Firefly	27
3.3 Simulasi Perhitungan Algoritma Firefly.....	27
3.4 Penyelesaian dengan Algoritma Firefly	29
3.5 Perhitungan Menggunakan Matlab.....	29
3.6 Hasil Perhitungan	32
BAB IV HASIL DAN PEMBAHASAN	
4.1 Implementasi	36
4.1.1 Kebutuhan Perangkat Keras	36
4.1.2 Kebutuhan Perangkat Lunak	37
4.1.3 Implementasi Algoritma Firefly Pada NPC	37
4.2 Firefly Sebagai Pathfinding Pada NPC	38

4.3 Pengujian Game.....	41
4.4 Implementasi Aplikasi Game	51
4.4.1 Pembangunan Terrain.....	51
4.4.2 Halaman Splashscreen.....	53
4.4.3 Antarmuka Menu	54
4.4.4 Menu Petunjuk.....	55
4.4.5 Pengaturan Terrain	56
4.4.6 Pembuatan Obstacle	57
4.4.7 Scene Game Pada Bagian Awal	58
4.4.8 Scene Game Menjalankan Misi.....	59
4.4.9 Scene Game Saat NPC Mengejar	60
4.4.10 Scene Game Saat NPC Menyerang	60
4.4.11 Lintasan NPC.....	61
4.4.12 Keterangan Item Wayang	62
4.5 Integrasi Dalam Islam.....	67
BAB V KESIMPULAN DAN SARAN	
5.1 Kesimpulan.....	71
5.2 Saran	71
DAFTAR PUSTAKA	72

DAFTAR GAMBAR

2.1 AI Model	18
2.2 Taksonomi Aplikasi Algoritma Firefly	22
3.1 Perancangan Algoritma Firefly	26
3.2 Contoh Permasalahan	29
3.3 Grafik Algoritma Firefly	32
3.4 Pergerakan Kunang – Kunang (a)	33
3.5 Pergerakan Kunang – Kunang (b)	33
3.6 Pergerakan Kunang – Kunang (c)	36
4.1 Pergerakan Kunang – Kunang (a)	42
4.2 Pergerakan Kunang – Kunang (b)	42
4.3 Pergerakan Kunang – Kunang (c)	42
4.4 Posisi Awal NPC Firefly dan Player	44
4.5 Pergerakan NPC Firefly Mengarah ke Player	44
4.6 NPC Firefly Tiba di Posisi Player	45
4.7 Posisi Awal NPC A* dan Player	45
4.8 NPC A* Bergerak ke Arah Player (a)	46
4.9 NPC A* Bergerak ke Arah Player (b)	46
4.10 NPC A* Berada Pada Posisi Player	47
4.11 Membuat Bukit Pada Terrain	52
4.12 Pemberian Efek Lava Pada Terrain	52
4.13 Tampak Atas Pada Terrain	53
4.14 Tampak Samping Pada Terrain	53
4.15 Halaman Splashscreen	54
4.16 Tampilan Menu Screen.....	55
4.17 Halaman Petunjuk	56
4.18 Pengaturan Permainan	57
4.19 Obstacle Permainan	57
4.20 Tampilan Game Pada Bagian Awal	58

4.21 Tampilan Game Pada Saat Menjalankan Misi	59
4.22 Tampilan Game Pada Saat Menemukan Item Wayang	59
4.23 Tampilan Game Saat NPC Mengejar	60
4.24 Tampilan Game Saat NPC Menyerang	61
4.25 Tampilan Graph Lintasan Yang Bisa Dilalui NPC	61
4.26 Karakter Wayang Bima	62
4.27 Karakter Wayang Arjuna	62
4.28 Karakter Wayang Arjuna 2	63
4.29 Karakter Wayang Sangkuni	63
4.30 Karakter Wayang Nakula	63
4.31 Karakter Wayang Sadewa	64
4.32 Karakter Wayang Duryudana	64
4.33 Karakter Wayang Citraksa	64
4.34 Karakter Wayang Citraksi	65
4.35 Karakter Wayang Durmagati	65
4.36 Karakter Wayang Karna	66
4.37 Karakter Wayang Yudistira	66

DAFTAR TABEL

2.1 Perbandingan Algoritma.....	6
2.2 Algoritma Firefly (Yang, 2009).....	21
2.3 Algoritma Astar (Hart, 1968)	23
3.1 Algoritma Firefly yang Digunakan di Matlab	30
4.1 Kebutuhan Perangkat Keras	36
4.2 Kebutuhan Perangkat Lunak	37
4.3 Keterangan Algoritma Firefly	38
4.4 Hasil Pengujian Pergerakan NPC dengan Firefly.....	48
4.5 Grafik Pergerakan Firefly dipengaruhi Oleh Intesitas Cahaya.....	49
4.6 Perbandingan Tingkat Akurasi	50
4.7 Function Pemanggilan Scene.....	55
4.8 Function Unity Untuk Memunculkan Item Wayang	66

ABSTRAK

Enan Emil. 2018. 3D Adventure Game Untuk Pengenalan Budaya Wayang Menggunakan Algoritma Firefly Sebagai Pembangkit Perilaku Pada NPC. Skripsi. Jurusan Teknik Informatika, Fakultas Saintek, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing : (I) Fresy Nugroho, M.MT, (II) Fachrul Kurniawan, M.MT

Kata kunci : *Adventure Game*, Algoritma *firefly*, Budaya Wayang, Pembangkit Perilaku.

Berbagai metode pencarian terus diteliti dan dikembangkan. Salah satu algoritma terbaik dalam pencarian jalur tercepat adalah *Firefly Algorithm*. Algoritma *Firefly* merupakan algoritma yang terinspirasi dari perilaku kawanan kunang-kunang (*Firefly Behaviour*). Algoritma ini diperkenalkan oleh Xin-She pada tahun 2007.

Algoritma *Firefly* memiliki banyak kemiripan dengan algoritma lain yang didasarkan pada kecerdasan kawanan, seperti *Particle Swarm Optimization* (PSO), *Artificial Bee Colony optimization* (ABC), dan Algoritma *Bacterial Foraging* (BFA), akan tetapi algoritma *Firefly* jauh lebih sederhana baik dalam konsep dan implementasi. Algoritma *Firefly* sangat efisien untuk diterapkan sebagai metode pencarian, dalam berbagai penelitian terbukti lebih baik dari algoritma pencarian lain, seperti jika dibandingkan dengan Algoritma Genetika, Algoritma *Firefly* lebih baik dalam hal memecahkan banyak masalah optimasi.

Penelitian ini membahas mengenai Implementasi Algoritma *Firefly* terhadap NPC musuh Pada Game “3D Wayang Adventure Game” Dengan tujuan sebagai pembangkit perilaku pencarian pada NPC (*Non Player Character*) dalam *3D Wayang Adventure Game* tersebut dan berbasis dekstop.

Hasil dari penelitian ini adalah, implementasi algoritma *firefly* terhadap NPC berhasil diterapkan sebagai pembangkit perilaku. Persentase keberhasilan yang dihasilkan yaitu, rata - rata 90.36% , dengan tingkat keberhasilan terendah 75.71% dan tingkat keberhasilan terbesar 98.57% dengan 20 kali pengujian.

ABSTRACT

Enan Emil. 2018. **3D Adventure Game For The Introduction Of Wayang Culture Using Firefly Algorithms As A Generating Behavior In NPCs.** Thesis. Department of Informatics, Faculty of Science and technology, State Islamic University of Maulana Malik Ibrahim Malang.

Adviser : (I) Fresy Nugroho, M.MT, (II) Fachrul Kurniawan, M.MT

Keywords : *Adventure Game, Firefly Algorithm, Puppet Culture, Behavior.*

Various search methods are continuously researched and developed. One of the best algorithms in the search for the fastest path is the Firefly Algorithm. Firefly algorithm is an algorithm that is inspired from the behavior of firefly. The algorithm was introduced by Xin-She in 2007.

The Firefly algorithm has many similarities with other algorithms based on swarm intelligence, such as Particle Swarm Optimization (PSO), Artificial Bee Colony optimization (ABC), and Bacterial Foraging Algorithm (BFA), but the Firefly algorithm is much simpler both in concept and implementation. The Firefly algorithm is very efficient to be applied as a search method, in many studies proving to be better than other search algorithms, as compared to Genetic Algorithms, the Firefly Algorithm is better in terms of solving many optimization problems.

This research discusses the Implementation of Firefly Algorithm against enemy NPC In Game "3D Wayang Adventure Game" With the aim of generating search behavior on NPC (Non Player Character) in 3D Wayang Adventure Game and desktop-based.

The result of this research is, firefly algorithm implementation to NPC successfully applied as behavior generator. The resulting percentage of success is, on average 65%, with the lowest success rate of 61% and the greatest success rate of 94% with 20 tests.

ملخص البحث

عنان اميل. ٢٠١٨ . ٣ مغامرة لعبة الأبعاد لإدخال ثقافة العرائس باستخدام خوارزميات يراعة كسلوك توليد في الشخصيات. أطروحة. قسم المعلوماتية ، كلية العلوم والتكنولوجيا ، الجامعة الإسلامية في مولانا مالك إبراهيم مالانج.

فريسي نوجروهو الماجستير و فخر الكرنياوان الماجستير.

الكلمة الرئيسية : لعبة المغامرة ، اليراع خوارزمية ، ثقافة العرائس ، السلوك.

يتم بحث وتطوير طرق البحث المختلفة باستمرار. واحدة من أفضل الخوارزميات في البحث عن أسرع مسار هو خوارزمية يراعة. خوارزمية يراعة هي خوارزمية مستوحاة من سلوك اليراع. تم تقديم الخوارزمية بواسطة شين-هي في عام ٢٠٠٧. ص.

تحتوي خوارزمية يراعة على العديد من أوجه التشابه مع خوارزميات أخرى مبنية على ذكاء سرب ، مثل (PSO)، وتحسين مستعمرة النحل الاصطناعي (ABC)، وخوارزمية الجراثيم البكتيرية (BFA)، ولكن خوارزمية Firefly تكون أبسط بكثير من حيث المفهوم والتنفيذ. تعد خوارزمية يراعة فعالة للغاية ليتم تطبيقها كأسلوب بحث ، في العديد من الدراسات التي تثبت أنها أفضل من غيرها من خوارزميات البحث ، بالمقارنة مع الخوارزميات الجينية ، فإن خوارزمية يراعة هي أفضل من حيث حل العديد من مشكلات التحسين.

يناقش هذا البحث تنفيذ خوارزمية يراعة ضد الشخصيات العدو في اللعبة لعبة مغامرة الدمية ثلاثية الأبعاد بهدف إنشاء سلوك البحث على الشخصيات في عبة مغامرة الدمية ثلاثية الأبعاد و يستند إلى سطح المكتب. نتيجة هذا البحث ، تطبيق خوارزمية اليراع إلى المجلس الوطني لنواب الشعب بنجاح تطبيق كمولد السلوك. نسبة النجاح الناتجة هي ٩٠.٣٦٪ في المتوسط ، مع أدنى معدل نجاح ٧٥.٧١٪ وأعلى معدل نجاح ٩٨.٥٧٪ مع ٢٠ اختبار.

BAB I

PENDAHULUAN

1.1. Latar Belakang

Industri *game* memiliki perkembangan yang sangat pesat dan signifikan. Banyak lahir inovasi dan kreasi mengagumkan yang berhasil mencuri perhatian pengguna. Ada berbagai macam *genre* dari *game*, salah satunya adalah *Role Playing game* (RPG). RPG umumnya memiliki alur cerita yang berbeda dari jenis *game* lain, seperti kisah yang diangkat oleh Ifa Alif dalam penelitiannya, yaitu *game* adventure yang mengangkat budaya Wayang Nusantara. Pada permainan tersebut, ia menggunakan algoritma A* sebagai pembangkit perilaku pencarian pada *Non Player Character* (NPC).

Beberapa metode telah diusulkan untuk menemukan hasil pencarian yang optimal, sebut saja A* dan *Dijkstra*. Berdasarkan penelitian yang dilakukan oleh Willy Setiawan (2010), Algoritma *Dijkstra* mampu menghitung hasil pencarian lebih cepat, tapi bisa saja memberikan hasil yang kurang akurat walau kemungkinan tersebut kecil, sementara Algoritma A* lebih lambat dalam mencari jalur optimum namun hasilnya lebih akurat.

Berbagai metode pencarian terus diteliti dan dikembangkan. Salah satu algoritma terbaik dalam pencarian jalur tercepat adalah *Firefly Algorithm*. Algoritma *Firefly* merupakan algoritma yang terinspirasi dari perilaku kawanan kunang-kunang (*Firefly Behaviour*). Algoritma ini diperkenalkan oleh Xin-She pada tahun 2007. Algoritma *Firefly* memiliki banyak kemiripan dengan algoritma lain yang didasarkan pada kecerdasan kawanan, seperti *Particle Swarm*

Optimization (PSO), *Artificial Bee Colony optimization* (ABC), dan Algoritma *Bacterial Foraging* (BFA), akan tetapi algoritma *Firefly* jauh lebih sederhana baik dalam konsep dan implementasi. Algoritma *Firefly* sangat efisien untuk diterapkan sebagai metode pencarian, dalam berbagai penelitian terbukti lebih baik dari algoritma pencarian lain, seperti jika dibandingkan dengan Algoritma Genetika, Algoritma *Firefly* lebih baik dalam hal memecahkan banyak masalah optimasi.

Pada Tugas Akhir ini *Firefly Algorith*m digunakan untuk menggantikan fungsi A* pada penelitian yang berjudul *3D Wayang Adventure Game* sebagai pembangkit perilaku pencarian pada NPC yang telah diteliti oleh Ifa Alif. *Game* tersebut merupakan objek penelitian dari tugas akhir yang dikerjakan dan telah dipresentasikan pada tanggal 5 November 2015.

1.2. Identifikasi Masalah

Berdasarkan latar belakang di atas, berikut adalah masalah yang dapat teridentifikasi :

- a. Apakah algoritma *Firefly* dapat diterapkan sebagai pembangkit perilaku pencarian pada NPC (*Non Player Character*) dalam *3D Wayang Adventure Game*?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah :

- a. Menerapkan algoritma *Firefly* sebagai pembangkit perilaku pencarian pada NPC (*Non Player Character*) dalam *3D Wayang Adventure Game*

1.4. Batasan Masalah

Batasan masalah pada penelitian ini adalah :

- a. Budaya yang ditampilkan didalam game ini adalah budaya wayang yang berisikan informasi tentang tokoh – tokoh dalam dunia perwayangan
- b. Aplikasi *game* ini berbasis desktop

1.5. Manfaat Penelitian

Manfaat penelitian ini adalah sebagai media edukasi yang inovatif dan menyenangkan (*edutainment*) demi mendukung pelestarian dan pengenalan budaya Indonesia, khususnya wayang. Pada permainan ini, wayang dikreasikan dalam bentuk *adventure game* tiga dimensi. Penerapan algoritma *Firefly* ditujukan sebagai algoritma pencarian.

BAB II

KAJIAN PUSTAKA

2.1. Penelitian Terkait

Fitra Haryadi dalam penelitiannya tentang “Metode Pathfinding pada Map 2D Menggunakan Algoritma Diagonal & Bidirectional BFS” menyatakan bahwa BFS melakukan banyak sekali operasi dan hal tersebut menjadi salah satu kekurangannya, akan tetapi hal tersebut dapat dioptimasi dengan *diagonal search* yang memperpendek rute pencarian dan *bidirectional search* yang menghemat operasi yang dilakukan.

Hendry Setiawan, Lo Hanjaya Hanafi dan Kestrilia Rega Prilianti dalam penelitiannya yang berjudul "Implementasi Algoritma Kunang-Kunang untuk Penjadwalan Mata Kuliah di Universitas Ma Chung", *Firefly algorithm* menjadi sebuah solusi dalam penjadwalan mata kuliah. Berdasarkan hasil uji coba, nilai intensitas cahaya terbaik didapatkan ketika parameter algoritma kunang-kunang, β_0 dimasukkan 1 dan γ dimasukkan 10 hingga didapatkan intensitas sebesar 0,0003831.

Pada penelitian yang berjudul “A Study from the Perspective of Nature-Inspired Metaheuristic Optimization Algorithms” yang dikerjakan oleh Deepthi S dan Aswathy Ravikumar, dijelaskan bahwa Algoritma *Firefly* didasarkan pada perilaku berkedip dari kerumunan kunang-kunang. *Ant Colony Optimization* didasarkan pada interaksi antara kelompok semut. Algoritma *Bat* didasarkan pada perilaku *echolocation* kelelawar yang mencari makan.

Adapun beberapa contoh algoritma atau metode yang sering digunakan atau populer dari algoritma firefly diantaranya :

- b. Ant Colony System (ACS) ACS adalah Algoritma semut merupakan algoritma yang digunakan untuk menyelesaikan masalah-masalah optimasi yang terinspirasi dari perilaku semut.
- c. *Particle Swarm Optimization* (PSO) PSO adalah suatu metode pencarian acak berbasis populasi yang didasarkan pada perilaku perpindahan individu dalam kawanan, seperti kawanan serangga, semut, rayap, lebah atau burung.
- d. Swarm Intelligence adalah salah satu teknik kecerdasan buatan yang berlandaskan kepada perilaku kolektif (collective behaviour) pada sistem yang terdesentralisasi dan dapat mengatur dirinya sendiri (self-organizing).
- e. Artificial Bee Colony (ABC) ABC adalah kecerdasan buatan yang menirukan koloni lebah dalam mencari sumber nektar (sari bunga).

Terdapat beberapa algoritma yang mengadaptasi perilaku pada spesies tertentu. Di antaranya tersaji seperti pada **Tabel 2.1**. *Cuckoo Search* didasarkan pada parasitisme proses pengeraman telur dari spesies Cuckoo yang dititipkan pada spesies lain. Di antara algoritma yang tersaji di **Tabel 2.1**, algoritma *Firefly* dinilai paling baik karena memiliki *running time* minimum dan tingkat keberhasilan yang lebih baik. Sehingga dapat digunakan untuk aplikasi skala besar secara efisien. *Ant Colony Optimization* memiliki tingkat keberhasilan yang lebih baik tapi *running time* yang dilakukan berada pada tingkat *medium* dan

membutuhkan lebih banyak *CPU time* dibandingkan dengan algoritma lain. Ringkasan lengkap mengenai algoritma yang dimaksud, disajikan dalam tabel berikut.(Yang,2009)

Tabel 2.1. Perbandingan Algoritma

Name Of Algorithm	Running Time	Success Rate	Classification	Algorithm Basis
Firefly Algorithm	Minimum	Better	Population Based, Attraction Based Algorithm	Flashing Behavior of Swarming Firefly
Ant Colony Optimization	Medium	Better	Population Based Algorithm	Interaction of Ant Species
Bat Algorithm	Medium	Intermediate	Population Based Algorithm	Echolocation of Foraging Bats
Cuckoo Search	Maximum	Low	Equation Based Algorithm	Brooding Parasitism of Cuckoo Bird

2.2. Budaya Wayang

Budaya wayang merupakan budaya asli Indonesia, khususnya di pulau Jawa. Wayang merupakan salah satu seni budaya bangsa Indonesia yang paling umum dan dikenal diseluruh tanah air Indonesia bahkan sudah di akui dunia diantara banyak karya budaya lainnya. Budaya wayang yang terus berkembang dari zaman ke zaman dan menjadi sebuah media yang digunakan untuk dakwah, pendidikan, pemahaman filsafat, serta hiburan.

Budaya wayang meliputi seni peran, seni suara, seni musik, seni tutur, seni sastra, seni lukis, seni pahat, dan seni perlambang. Keberadaan wayang sudah berabad-abad sebelum agama hindu masuk ke pulau Jawa. cerita wayang yang sangat populer dimasyarakat merupakan adaptasi dari karya Ramayana dan Mahabarata. Untuk menyesuaikan keadaan dan filsafah di Indonesia, cerita - cerita dalam pewayangan itu sendiri banyak mengalami perubahan dan penambahan.

2.2.1. Macam – Macam Wayang

Menurut S. Haryanto (1988: 41-142) wayang dapat dibagi menjadi 8 jenis yang terdiri dari beberapa ragam, yaitu:

2.2.1.1. Wayang Beber

Wayang beber merupakan wayang yang termasuk dalam bentuk wayang yang paling tua usianya dibandingkan jenis lainnya dan merupakan wayang yang berasal dari masa runtuhnya Majapahit di Jawa. Wayang dilukiskan menggunakan media gulungan kertas yang berisikan kejadian – kejadian atau adegan – adegan penting dalam sebuah cerita yang ingin di tunjukkan. Pertunjukannya dilakukan dengan pembacaan cerita dan peragaan gambar-gambar yang telah dilukiskan.

2.2.1.2. Wayang Purwa

Wujudnya berupa wayang kulit, wayang golek, atau wayang wong (orang) dengan mempergelarkan cerita yang bersumber pada kitab Mahabaratha atau Ramayana. Istilah purwa itu sendiri dari pendapat para ahli dinyatakan berasal dari kata „parwa“ yang merupakan bagian dari cerita

Mahabharata atau Ramayana. Selain itu, di kalangan masyarakat Jawa, kata purwa sering diartikan pula dengan purba (jaman dahulu). Oleh karena itu, wayang purwa diartikan pula sebagai wayang yang menyajikan cerita-cerita jaman dahulu (purwa).

2.2.1.3. Wayang Madya

Berusaha menggabungkan semua jenis wayang yang ada menjadi satu kesatuan yang berangkai serta disesuaikan dengan sejarah Jawa sejak beberapa abad yang lalu sampai masuknya agama Islam di Jawa dan diolah secara kronologis. Penggabungan tersebut mengakibatkan terciptanya jenis wayang baru yang menggambarkan dari badan tengah ke atas berwujud wayang purwa, sedangkan dari badan tengah ke bawah berwujud wayang gedog. Wayang Madya ini memakai keris dan dibuat dari kulit, ditatah dan disungging.

2.2.1.4. Wayang Gedog

Arti kata gedog sampai sekarang masih belum dapat ditemukan dengan pasti. Para sarjana barat , gedog ditafsirkan sebagai kandang kuda (bahasa Jawa: gedogan = kandang kuda). Dalam bahasa Kawi, gedog berarti kuda. Sementara pendapat lain menyatakan bahwa „gedog“ itu merupakan batas antara siklus wayang purwa yang mengambil seri cerita Mahabharata dan Ramayana dengan siklus cerita Panji. Ada pula yang menafsirkan bahwa kata gedog berasal dari suara „dog, dog“ yang ditimbulkan dari ketukan sang dalang pada kotak wayang di sampingnya. Namun hingga kini belum dapat yang melanjutkan penelitian, mengapa kata gedog tersebut digunakan untuk

suatu jenis wayang. Ada pula yang menyatakan bahwa wayanag gedog mirip dengan wayang purwa.

2.2.1.5. Wayang Menak

Wayang Menak ini terbuat dari kulit yang ditatah dan disungging sama halnya seperti wayang kulit purwa. Sedangkan wayang Menak yang dibuat dari kayu dan merupakan wayang golek disebut Wayang Tengul. Dalam pementasan wayang menak dijumpai dua macam bentuk wayang, antara lain yang berupa wayang golek dan kulit. Pementasan wayang menak di Jawa Tengah pada umumnya menggunakan wayang golek menak. Sedangkan pementasan wayang kulit menak ini menggunakan kelir dan blencong, sama halnya dengan pementasan wayang kulit purwa, hanya pakemnya berdasarkan pakem Serat Menak.

2.2.1.6. Wayang Babad

Wayang Babad Merupakan penciptaan wayang baru setelah wayang Purwa, Madya dan Gedog yang pementasannya bersumber pada cerita-cerita babad (sejarah) setelah masuknya agama Islam di Indonesia antara lain kisahkisah kepahlawanan dalam masa kerajaan Demak dan Pajang. Wayang ini disebut sebagai wayang Babad atau wayang Sejarah.

2.2.1.7. Wayang Modern

Ketika wayang-wayang purwa, madya dan gedog sudah tidak sesuai lagi untuk keperluan yang khusus, maka untuk kebutuhan masyarakat akan sarana komunikasi sosial dengan media wayang semakin meningkat, maka diciptakanlah wayang baru lagi yang dapat memadai faktor-faktor komunikasi tersebut.

2.2.1.8. Wayang Topeng

Wayang Topeng adalah wayang yang dimainkan oleh orang dengan menggunakan topeng yang menutupi wajah. Wayang Topeng ini dimainkan dengan iringan gamelan dan tari-tarian. Wayang Topeng ini selain ditampilkan dalam pagelaran budaya, biasanya juga digunakan dalam pesta pernikahan untuk menghibur para tamu undangan sekitar 20 sampai 30 menit dalam pementasannya. (Indah Turita S. 2008)

2.3. Permainan (*Game*)

Game atau dalam bahasa Indonesia berarti permainan, merupakan sebuah aktifitas yang bertujuan untuk memberikan kesenangan dan memiliki aturan yang menjadi acuan bagi para pemain. Beda permainan memiliki aturan main yang berbeda pula. Aturan dan kompetisi menjadi salah satu daya tarik dari sebuah *game*. Adapun definisi *game* menurut para ahli sebagai berikut :

a. Clark Ab

Menurut Clark Abt (1970), *Game* adalah kegiatan yang melibatkan keputusan pemain serta berupaya mencapai suatu tujuan dengan dibatasi dengan aturan atau konteks tertentu.

b. Salen & Zimmerman

Menurut Salen & Zimmerman (2004), *game* adalah sistem di mana pemain terlibat dalam konflik buatan, yang ditentukan oleh sebuah aturan, yang memiliki hasil yang dapat dihitung.

c. Chris Crawford

Chris Crawford berpendapat bahwa permainan, pada intinya adalah interaktif, aktivitas yang berpusat pada pencapaian, ada pemain aktif, dan ada lawan.

d. Roger Caillois

Menurut Roger Caillois dalam bukunya yang berjudul “*Les jeux et les hommes*” menyatakan game adalah kegiatan yang mencakup karakteristik berikut: fun (bermain bebas adalah pilihan tetapi bukan kewajiban), terpisah (terpisah), tidak pasti, tidak produktif, diatur oleh aturan (tidak ada aturan), fiktif (pura-pura).

e. Greg Costikyan

Menurut Greg Costikyan, Game adalah "suatu bentuk karya seni di mana para peserta, yang disebut *Player*, membuat keputusan untuk mengelola sumber dayanya melalui benda-benda dalam permainan untuk mencapai tujuan".

Dewasa ini, bermain *game* sudah menjadi rutinitas berbagai kalangan, bahkan beberapa *game* khusus dirancang berdasarkan usia dan minat para pemainnya.

2.3.1 Jenis – Jenis Permainan (Game)

Kemajuan teknologi khususnya dibidang industry *game* tidak lepas dari minat masyarakat dalam bermain *game* yang semakin hari semakin tinggi, sehingga mulailah bermunculan banyak sekali perangkat lunak ataupun perangkat keras untuk membantu memberikan pengalaman yang lebih kepada para pemain

game. jenis – jenis *game* yang saat ini beredar juga sangat banyak, dan dibagi berdasarkan *platform* dan *genre*. Adapun jenis *game* berdasarkan *platform* yaitu :

2.3.1.1. PC Games

PC Games atau *Personal Computer Games* adalah *game* yang dimainkan di perangkat computer, bisa berupa *online game* (terhubung dengan koneksi internet) ataupun *offline game* (tidak terhubung dengan koneksi internet), contoh *game* yang biasa dimainkan di *PC Games* seperti *PlayerUnknown's Battlegrounds*, *Divinity*, *Wolfenstein*, *Fortnite*, dan *Cuphead*.

2.3.1.2. Console Games

Console Games atau konsol permainan adalah *game* yang biasa dimainkan di perangkat - perangkat konsol seperti Xbox (buatan Microsoft), Playstation (buatan Sony) dan Nintendo.

2.3.1.3. Handheld Games

Handheld Games adalah *game* konsol yang layar, speaker dan control berada dalam satu perangkat yang dapat digenggam dengan kedua tangan seperti Nintendo Switch, Nvidia Shield, Nintendo 3DS, Xperia Play, Playstation Vita, dan Razer Switchblade.

2.3.1.4. Mobile Games

Mobile Games adalah *game* yang dimainkan di perangkat *mobile phone* atau *handphone* dan seiring perkembangan zaman saat ini lebih sering kita dengar dengan sebutan *smartphone* yang didukung berbagai macam teknologi sehingga mampu memainkan *game* dalam spesifikasi tinggi. Adapun contoh *mobile games* yang sering dimainkan seperti *mobile legends*, *clash of clans*, *clash royale* dan *pubg*.

Dilihat dari jenis *platform*, *game* itu sendiri dibagi menjadi 4 macam, sedangkan jika dilihat dari *genre*, *game* terbagi atas 11 macam, Adapun jenis *game* berdasarkan *genre* yaitu :

a. Action Game

Action game adalah *game* yang menyajikan fitur utama berupa aksi, dimana pemain akan diuji keterampilannya serta tingkat reaksi yang cukup cepat dalam melawan musuh atau NPC dalam sebuah *game*. Dan saat ini *action game* sudah sangat beragam dengan kualitas grafik yang tinggi. Adapun contoh *action game* seperti *overkill 3*, *Angel Stone*, *Warhammer: freeblade*, *guardian stone* dan *walking war robot*.

b. Fighting Game

Sesuai dengan namanya *fighting game* yang berarti permainan pertarungan, dimana pada *genre* ini memberikan pengalaman kepada *player* untuk melakukan pertarungan satu lawan satu atau lebih dengan memilih *character* atau *hero* yang memiliki masing – masing kemampuan,

contoh *fighting game* yaitu mortal kombat, tekken, street fighter, fight club dan injustice.

c. First Person Shooter (FPS)

First person shooter merupakan *game* aksi di mana pemain berada "di belakang mata" dari karakter permainan dalam perspektif orang pertama. Meskipun sejumlah *game* FPS juga mendukung pandangan orang ketiga. Adapun contoh *game* yang berjenis FPS seperti Counter Strike, Far Cry, Deep Rock Galactic, DayZ dan Midair.

d. Third Person Shooter (TPS)

Third Person Shooter adalah genre *game* aksi 3D di mana karakter pemain terlihat di layar, berbeda dengan *first Person Shooter* pada TPS, *player* dapat melihat tampilan dari karakter, contoh *game* yang termasuk dalam *Third Person Shooter* seperti Mass Effect, Splatoon, Star Wars, Drawn to Death dan School Girl.

e. Real Time Strategy (RTS)

Real Time Strategy atau RTS adalah *game* yang berjenis strategi dimana pemain membutuhkan cara tertentu untuk menyelesaikan tantangan disetiap tingkatan level atau map, dan RTS sendiri umumnya memiliki banyak tujuan dan para pemain akan saling bersaing dengan pemain lainnya ataupun melawan komputer untuk mencapai tujuan tersebut. Contoh *game* yang berjenis *real time strategy* adalah halo wars, total war, steel division, starcraft dan spellforce.

f. Role Playing Game (RPG)

Role Playing Game atau RPG adalah jenis permainan petualangan khusus yang biasanya menggabungkan tiga elemen utama yaitu pencarian spesifik, *leveling* (proses untuk mengembangkan karakter melalui pengalaman untuk meningkatkan kemampuannya menangani musuh yang mematikan) dan menyelesaikan misi atau *quest*. Karena itu, *game* jenis ini memiliki banyak variasi dan tampilan. Contoh *game* yang termasuk dalam *game* RPG seperti monster hunter, vampyr, kingdom hearts, Greedfall dan Code Vein.

g. Adventure

Adventure game adalah *game* yang bersifat petualangan, dimana pemain akan menjalankan tugas atau misi berupa pemecahan misteri yang harus diselesaikan, dimana pemain dituntut untuk mengumpulkan sejumlah barang atau menandai sebuah objek sebagai syarat dalam menyelesaikan misi tersebut, dan didalam *adventure game* juga terdapat unsur pertarungan tapi porsinya tidak banyak. Adapun contoh *adventure game* saat ini seperti monster hunter, a way out, the last night, jenny LeClue dan Harry Potter.

h. Simulation

Simulation adalah jenis *game* yang berupaya untuk memberikan pengalaman kepada pemain secara akurat dalam bentuk simulasi. seperti pada *game* SimCity yang memberikan pengalaman kepada pemain bagaimana membuat atau manajemen sebuah kota agar mampu berkembang, atau lebih mengandalkan kelincihan tangan seperti mengemudikan pesawat atau mobil.

i. Sport Game

Sesuai dengan artinya yaitu permainan olahraga, *Sport Game* memberikan pengalaman sama halnya dengan *simulation game* namun lebih ke arah *sport* seperti sepakbola, basket dan bola voli yang memberikan pengalaman kepada pemain untuk memajemen sebuah tim atau menjadi pemain di dalam tim itu sendiri. Adapun contoh *game* yang merupakan *sport game* yaitu NBA 2K18, FIFA 2018, Pro Evolution Soccer, Everybody's Golf dan Rugby 18.

j. Racing Game

Racing game adalah *game* yang memberikan pengalaman seperti halnya *simulation game* dan pada umumnya memiliki *camera view* seperti pada *game* FPS dan TPS, perbedaannya terletak pada objeknya yang berupa benda atau kendaraan seperti mobil, motor, pesawat dan sepeda, tantangannya adalah pemain dituntut mendapatkan catatan waktu terbaik dalam melewati garis *finish*. Contoh dari *racing game* seperti Pole Position, Mario Kart, Gran Turismo, Need for Speed dan GTR.

k. Multiplayer Game

Multiplayer game adalah permainan yang dimainkan dua atau lebih pemain, dimana para pemain dalam sebuah permainan dapat bekerjasama atau bahkan saling bersaing untuk menyelesaikan permainan tersebut. Conoh *multiplayer game* seperti Fornite, PUBG, Destiny 2, For Honor dan Absolver. (Lindsay Grace, 2005)

2.4. Artificial Intelligence (AI) Pada *Game*

Perkembangan *game* saat ini tidak lepas dari *Artificial Intelligence* (AI). Kecerdasan buatan merupakan salah satu bagian dari ilmu komputer yang membuat mesin (komputer) dapat melakukan pekerjaan seperti manusia dan memungkinkan komputer untuk dapat berpikir.

Perkembangan *game* yang sangat pesat pada masa kini, tidak dipungkiri bahwa dibutuhkan sesuatu yang berbeda pada *rule* permainan yang hendak dibangun. Di situlah salah satu peran kecerdasan buatan. Kecerdasan buatan memiliki kemampuan menjadikan *game* lebih seru. Adapun pengertian kecerdasan buatan menurut beberapa ahli :

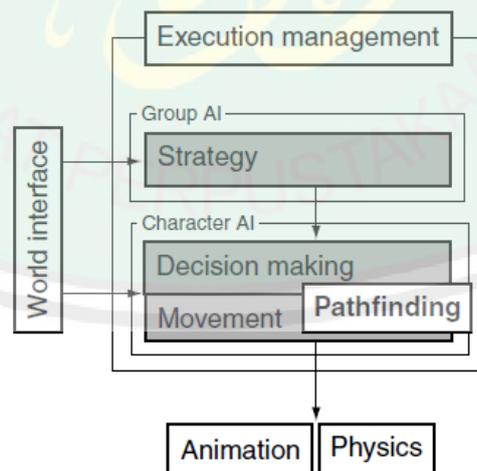
- a. *H.A. Simon* (1987) mengatakan bahwa kecerdasan buatan merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas.
- b. Menurut *Sri Kusumadewi* (2003), Kecerdasan buatan merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia.
- c. *John McCarthy* mendefinisikan, AI merupakan cabang dari ilmu komputer yang berfokus pada pengembangan komputer untuk dapat memiliki kemampuan dan berperilaku seperti manusia”.
- d. Menurut *Anita Desiani dan Muhammad Arhami* (2006), Kecerdasan buatan adalah Bagian dari komputer sehingga harus didasarkan pada Teori Suara dan prinsip – prinsip dibidangnya. Prinsip – prinsip tersebut meliputi struktur data yang digunakan dalam representasi pengetahuan,

algoritma yang diperlukan untuk mengaplikasikan pengetahuan tersebut, serta bahasa dan teknik pemrograman yang digunakan dalam mengimplementasikannya.

Seiring dengan perkembangan *game* dan kecerdasan buatan yang diterapkan pada *game*, sistem dalam permainan juga dapat belajar mengenali pola permainan dari pemain. Sebagai contoh, ketika pemain tersebut memulai permainan kembali, maka sistem ini akan menggunakan *rule* yang berbeda, sehingga *game* menjadi lebih menarik dan menantang untuk dimainkan. (Luqmannul, 2010)

2.5. Teknik dalam AI

Pada buku yang berjudul “*Artificial Intellegnce for Games*” disebutkan bahwa ada beberapa macam teknik yang digunakan dalam penerepan AI pada sebuah *game*, di antaranya *movement*, *pathfinding*, *decision making*, *tactical and strategic AI*, *learning* dan *board games*. Setiap teknik tersebut memiliki peran yang sangat penting bagi keberhasilan sebuah *game*.



Gambar 2.1. AI Model

Pada **gambar 2.1.** dapat dilihat bahwa setiap teknik dalam AI memiliki perannya masing-masing dan saling terkait. Hal tersebutlah yang membuat sebuah *game* berjalan dengan baik dan sebagaimana mestinya. (Husni, 2009)

2.6. Pathfinding

Pada permainan yang memanfaatkan medan yang luas dan membutuhkan pencarian lokasi mengaplikasikan *pathfinding*. *Pathfinding* berfungsi untuk mencari rute terdekat, penerapan *pathfinding* dipengaruhi beberapa faktor seperti map dan objek yang terdapat pada suatu *game* yang berkaitan dengan letak dan posisi.

A star (A*) dan Dijkstra adalah algoritma yang paling sering digunakan dalam penerapan *pathfinding* pada sebuah *game*. Algoritma A* ditemukan oleh Peter Hart, Nils Nilsson, dan Bertram Raphael pada tahun 1968. A* menggunakan *best first search* dan bobot minimal yang diberikan dari simpul awal ke simpul tujuan. Sedangkan Dijkstra adalah sebuah algoritma yang dikembangkan oleh Edsger Dijkstra, seorang ilmuwan komputer dari Belanda.

Menurut Willy Setiawan (2010), A* menggunakan jarak ditambah biaya yang dinyatakan dengan $f(x)$ dari fungsi heuristik untuk menentukan urutan pencarian mencapai simpul mana pada sebuah pohon, sedangkan algoritma Dijkstra adalah sebuah algoritma yang menyelesaikan pencarian jalur terpendek pada graf dengan nilai non negatif untuk bobot setiap simpul, menghasilkan pohon jalur terpendek.

Penggunaan algoritma A* cocok diaplikasikan ke dalam aplikasi permainan yang membutuhkan pencarian rute oleh AI, karena fungsi *heuristic*. Fungsi

heuristik merupakan perilaku menebak-nebak jalan yang harus dilalui. Pada penelitian ini digunakan algoritma *Firefly* sebagai *pathfinding* untuk menggantikan peran A* yang telah diterapkan peneliti sebelumnya. (Pramono, 2005)

2.7. Algoritma Firefly

Firefly algorithm atau disebut sebagai algoritma kunang-kunang merupakan algoritma yang terinspirasi dari perilaku kunang-kunang pada saat mencari cahaya. Algoritma ini pertama kali ditemukan oleh Dr. Xin-She Yang di Universitas Cambridge pada tahun 2007. Terdapat 3 acuan dasar dari algoritma kunang-kunang yaitu :

1. Semua kunang-kunang adalah unisex, jadi satu kunang-kunang tertarik dengan kunang-kunang lain terlepas dari jenis kelamin mereka.
2. Daya tarik sebanding dengan kecerahan, maka kunang-kunang dengan kecerahan lebih redup akan bergerak ke arah kunang-kunang dengan kecerahan lebih terang dan kecerahan berkurang seiring dengan bertambah jarak. Apabila tidak ada kunang-kunang yang memiliki kecerahan paling cerah maka kunang-kunang akan bergerak random.
3. Kecerahan kunang-kunang dideterminasikan oleh tempat dari fungsi objektif kunang-kunang.

Berdasarkan tiga aturan dasar di atas, langkah-langkah algoritma *Firefly* dapat diringkas menjadi *pseudo code* yang dapat dilihat pada Tabel 2.2.

Tabel 2.2. Algoritma *Firefly* (Yang, 2009)

```

Begin
  1) Objective function:  $f(x)$ ;
  2) Generate an initial population of fireflies  $x_i$ ;
  3) Formulate light intensity  $I$  so that it is
associated with  $f(x_i)$ 
  (for example, for maximization problems,
  or simply  $I_i = f(x_i)$ ;)
  4) Define absorption coefficient  $\gamma$ 

  While ( $t < \text{MaxGeneration}$ )
    for  $i = 1 : n$  (all  $n$  fireflies)
      for  $j = 1 : i$  ( $n$  fireflies)
        if ( $r < b_{ij}$ ),
          Vary attractiveness with distance  $r$  via
           $a_{ij} = a_0 \exp(-\gamma r)$ ;
          move firefly  $i$  towards  $j$ ;
          Evaluate new solutions and update light
intensity;
        end if
      end for  $j$ 
    end for  $i$ 
    Rank fireflies and find the current best;
  end while

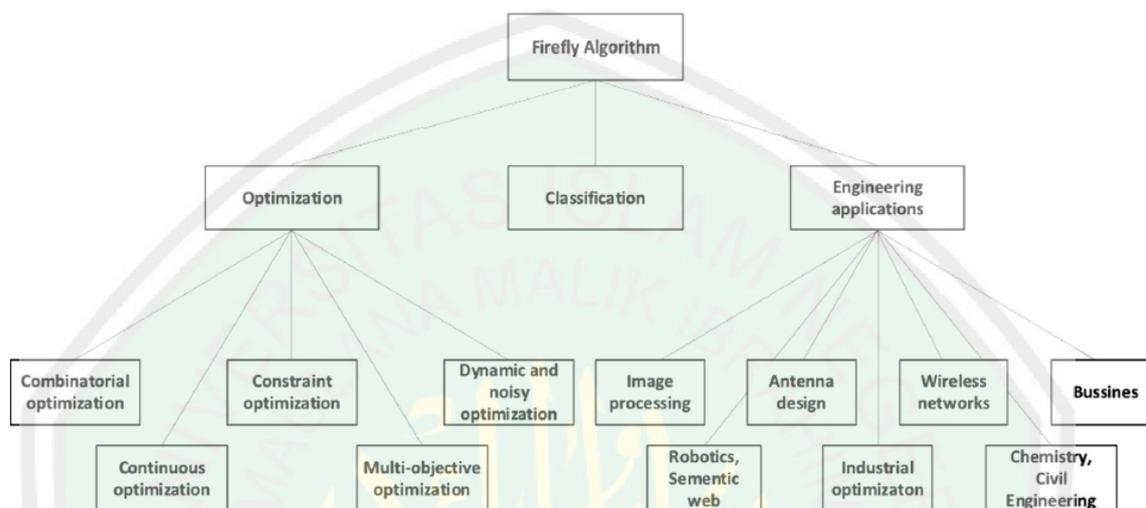
  Post-processing the results and visualization;

End

```

2.7.1. Pengaplikasian Algoritma *Firefly*

Pada hakikatnya terdapat tiga pengaplikasian algoritma *Firefly* secara garis besar sesuai pada **gambar 2.2**, yaitu *optimization*, *classification* dan *engineering applications*.



Gambar 2.2. Taksonomi Aplikasi Algoritma *Firefly*

Pada penelitian ini, algoritma *Firefly* menjurus kepada *robotics* yang merupakan salah satu cabang dari pengaplikasian algoritma *Firefly* yaitu pada *engineering applications*, dan akan diterapkan dalam *pathfinding* pada NPC dan menggantikan fungsi dari A* yang telah di terapkan pada *3D Wayang Adventure Game*, sehingga nantinya dapat dilihat seberapa optimum penggunaan algoritma *Firefly* dalam mengukur jarak terpendek pada NPC musuh untuk bergerak menuju *player*. (Yang, 2009)

2.8. Algoritma Astar (A*)

Algoritma A star menggunakan dua senarai yaitu *OPEN* dan *CLOSED*. *OPEN* adalah senarai yang digunakan untuk menyimpan simpul-simpul yang

pernah dibangkitkan dan nilai heuristiknya telah dihitung tetapi belum terpilih sebagai simpul terbaik atau *best node* dengan kata lain, *OPEN* berisi simpul-simpul masih memiliki peluang untuk terpilih sebagai simpul terbaik, sedangkan *CLOSED* adalah senarai untuk menyimpan simpul-simpul yang sudah pernah dibangkitkan dan sudah pernah terpilih sebagai simpul terbaik. (Riyadi, 2009)

Tabel 2.3. Algoritma *Astar* (Hart, 1968)

```

function A*(start, goal)
  // The set of nodes already evaluated
  closedSet := {}

  // The set of currently discovered nodes that are
  // not evaluated yet.
  // Initially, only the start node is known.
  openSet := {start}

  // For each node, which node it can most
  // efficiently be reached from.
  // If a node can be reached from many nodes,
  // cameFrom will eventually contain the
  // most efficient previous step.
  cameFrom := an empty map

  // For each node, the cost of getting from the
  // start node to that node.
  gScore := map with default value of Infinity

  // The cost of going from start to start is zero.
  gScore[start] := 0

  // For each node, the total cost of getting from
  // the start node to the goal
  // by passing by that node. That value is partly
  // known, partly heuristic.
  fScore := map with default value of Infinity

```

```

    // For the first node, that value is completely
    heuristic.
    fScore[start] := heuristic_cost_estimate(start,
    goal)

    while openSet is not empty
        current := the node in openSet having the
lowest fScore[] value
        if current = goal
            return reconstruct_path(cameFrom, current)

        openSet.Remove(current)
        closedSet.Add(current)

        for each neighbor of current
            if neighbor in closedSet
                continue // Ignore the neighbor
which is already evaluated.

            if neighbor not in openSet // Discover a
new node
                openSet.Add(neighbor)

                // The distance from start to a neighbor
//the "dist_between" function may vary as
per the solution requirements.
                tentative_gScore := gScore[current] +
dist_between(current, neighbor)
                if tentative_gScore >= gScore[neighbor]
                    continue // This is not a better
path.

                // This path is the best until now. Record
it!
                cameFrom[neighbor] := current
                gScore[neighbor] := tentative_gScore
                fScore[neighbor] := gScore[neighbor] +
heuristic_cost_estimate(neighbor, goal)

        return failure

function reconstruct_path(cameFrom, current)

```

```
total_path := [current]
while current in cameFrom.Keys:
    current := cameFrom[current]
    total_path.append(current)
return total_path
```

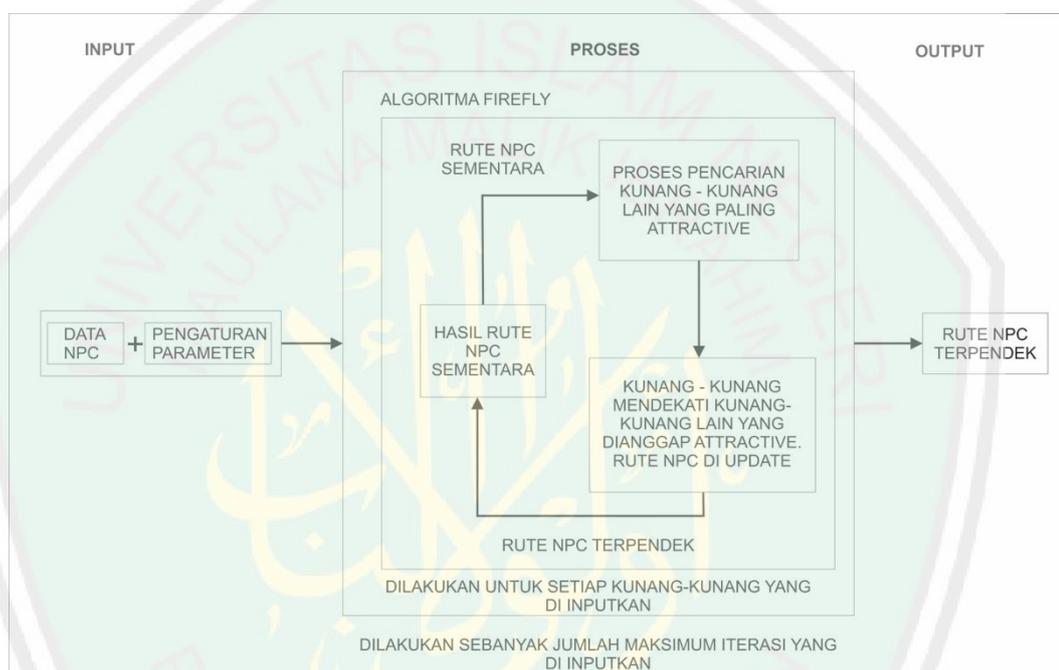


BAB III

PERANCANGAN DAN PENERAPAN

3.1. Perancangan Algoritma Firefly

Berikut perancangan algoritma *Firefly* yang akan diterapkan pada NPC :



Gambar 3.1. Perancangan Algoritma Firefly

Perancangan algoritma *Firefly* diterapkan pada NPC (musuh) yang bergerak menuju *player* dengan rute terpendek sesuai dengan **gambar 3.1**. Berikut detail langkah penerapan *algoritma Firefly*:

1. Input data NPC dan pengaturan inialisasi populasi Kunang-kunang.
2. Inialisasi fungsi *fitness* awal sebagai penentuan tingkat intensitas cahaya awal.

3. Melakukan proses pencarian *Firefly*. Kunang-kunang akan mendekati *Firefly* lainnya yang dianggap paling *attractive*.
4. Update pergerakan tiap *Firefly*.
5. Lakukan sampai batas iterasi atau sampai mendapatkan *Firefly* dengan fungsi *fitness* yang cukup baik.

3.2. Penerapan Algoritma Firefly

Game ini dimainkan secara *single player* dan termasuk dalam kategori *Adventure game*, dalam *game* terdapat karakter utama yang digunakan *player* untuk menjalankan misi dan terdapat karakter NPC yang berperan sebagai musuh. NPC menggunakan algoritma *Firefly* dalam mencari posisi *player*, di mana hal tersebut menjadi objek penelitian dalam permainan ini.

3.3. Simulasi Perhitungan Algoritma Firefly

Parameter yang terdapat dalam algoritma *Firefly* adalah :

1. Intensitas cahaya dan Keatraktifan

Intensitas cahaya menyatakan tingkat kecerahan dari Kunang-kunang. Semakin cerah seekor Kunang-kunang, maka semakin bagus solusi Kunang-kunang tersebut. Intensitas cahaya proporsional dengan fungsi obyektif. Intensitas cahaya dalam kasus ini adalah nilai *error* masing-masing Kunang-kunang.

Tingkat keatraktifan β bernilai relatif. Hal ini disebabkan intensitas cahaya dinilai oleh kunang-kunang yang lain. Hasil penilaian akan berbeda tergantung jarak antara kunang-kunang yang satu dengan yang lain (r_{ij}).

Fungsi keatraktifan dapat dirumuskan sebagai berikut:

$$\beta(r) = \beta_0 e^{-\gamma r^m}, \quad (m \geq 1).$$

dimana γ adalah tingkat absorpsi.

2. Jarak

Setiap kunang-kunang memiliki posisi yang berbeda. Jarak antar masing-masing kunang-kunang diukur menggunakan diagram Cartesian dan diformulasikan menggunakan rumusan berikut:

$$r_{ij} = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}$$

Di mana :

r_{ij} = jarak antara kunang-kunang i dan j

$x_{i,k}$ = nilai dimensi k pada kunang-kunang i

$x_{j,k}$ = nilai dimensi k pada kunang-kunang j

3. Pergerakan

Berikut merupakan rumus pergerakan dari kunang-kunang :

$$\mathbf{x}_i = \mathbf{x}_i + \beta_0 e^{-\gamma r_{ij}^2} (\mathbf{x}_j - \mathbf{x}_i) + \alpha \left(\text{rand} - \frac{1}{2} \right)$$

Di mana :

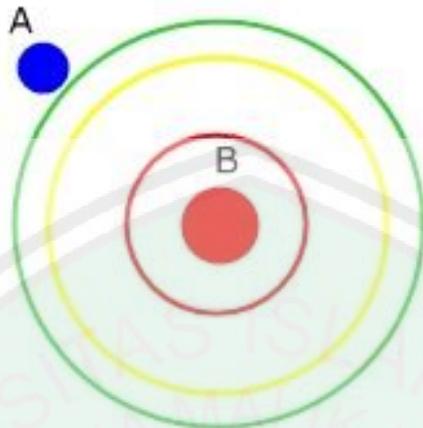
\mathbf{x}_i = posisi awal kunang-kunang i

β_0 = tingkat keatraktifan kunang-kunang pada posisi 0

γ = koefisien penyerapan cahaya

α = koefisien bilangan random antara 0 dan 1

r_{ij} = jarak antara kunang-kunang i dan j



Gambar 3.2. Contoh Permasalahan

3.4. Penyelesaian dengan Algoritma Firefly

Diberikan sebuah permasalahan, yaitu bagaimana cara NPC musuh bergerak dari titik A menuju titik B. Sesuai pada gambar 3.2. Berdasarkan permasalahan tersebut, digambarkan bagaimana letak awal NPC musuh sehingga nantinya dapat bergerak dari titik A menuju titik B, dalam kasus ini menggunakan 10 iterasi. Maksimum daya serap cahaya 2 dan 5 populasi. Setelah proses pencarian rute telah mencapai iterasi yang diinginkan, maka dihasilkan output sistem berupa rute NPC terpendek.

3.5. Perhitungan Menggunakan Matlab

Berikut adalah *method* dari *firefly Algorithm* yang digunakan dalam aplikasi matlab untuk menghitung akurasi dari algoritma tersebut.

Tabel 3.1 Algoritma *Firefly* yang digunakan di Matlab

```

function [best]=firefly_simple(instr)
% n=number of fireflies
% MaxGeneration=number of pseudo time steps
if nargin<1, instr=[12 50]; end
n=instr(1); MaxGeneration=instr(2);
% Show info
help firefly_simple.m
rand('state',0); % Reset the random generator
% ----- Four peak functions -----
str1='exp(-(x-4)^2-(y-4)^2)+exp(-(x+4)^2-(y-4)^2)';
str2='+2*exp(-x^2-(y+4)^2)+2*exp(-x^2-y^2)';
funstr=strcat(str1,str2);
% Converting to an inline function
f=vectorize(inline(funstr));
% range=[xmin xmax ymin ymax];
range=[-5 5 -5 5];

% -----
alpha=0.2; % Randomness 0--1 (highly random)
gamma=1.0; % Absorption coefficient
delta=0.97; % Randomness reduction (similar to
            % an annealing schedule)
% -----
% Grid values are used for display only
Ngrid=100;
dx=(range(2)-range(1))/Ngrid;
dy=(range(4)-range(3))/Ngrid;
[x,y]=meshgrid(range(1):dx:range(2),...
               range(3):dy:range(4));
z=f(x,y);
% Display the shape of the objective function
figure(1); surfc(x,y,z);

% -----
% generating the initial locations of n fireflies
[xn,yn,Lightn]=init_ffa(n,range);
% Display the paths of fireflies in a figure with
% contours of the function to be optimized
figure(2);
% Iterations or pseudo time marching
for i=1:MaxGeneration, %%%% start iterations
% Show the contours of the function
contour(x,y,z,15); hold on;
% Evaluate new solutions
zn=f(xn,yn);

% Ranking the fireflies by their light intensity

```

```

[Lightn, Index]=sort(zn);
xn=xn(Index); yn=yn(Index);
xo=xn; yo=yn; Lighto=Lightn;
% Trace the paths of all roaming fireflies
plot(xn,yn, '.', 'markersize',10, 'markerfacecolor','g');
% Move all fireflies to the better locations
[xn,yn]=ffa_move(xn,yn,Lightn,xo,yo,Lighto,alpha,gamma,
range);
drawnow;
% Use "hold on" to show the paths of fireflies
hold off;

% Reduce randomness as iterations proceed
alpha=newalpha(alpha,delta);

end % end of iterations
best(:,1)=xo'; best(:,2)=yo'; best(:,3)=Lighto';

% ----- All subfunctions are listed here -----
% The initial locations of n fireflies
function [xn,yn,Lightn]=init_ffa(n,range)
xrange=range(2)-range(1);
yrange=range(4)-range(3);
xn=rand(1,n)*xrange+range(1);
yn=rand(1,n)*yrange+range(3);
Lightn=zeros(size(yn));

% Move all fireflies toward brighter ones
function [xn,yn]=ffa_move(xn,yn,Lightn,xo,yo,...
Lighto,alpha,gamma,range)
ni=size(yn,2); nj=size(yo,2);
for i=1:ni,
% The attractiveness parameter beta=exp(-gamma*r)
for j=1:nj,
r=sqrt((xn(i)-xo(j))^2+(yn(i)-yo(j))^2);
if Lightn(i)<Lightn(j), % Brighter and more attractive
beta0=1; beta=beta0*exp(-gamma*r.^2);
xn(i)=xn(i).*(1-beta)+xo(j).*beta+alpha.*(rand-0.5);
yn(i)=yn(i).*(1-beta)+yo(j).*beta+alpha.*(rand-0.5);
end
end % end for j
end % end for i
[xn,yn]=findrange(xn,yn,range);

% Reduce the randomness during iterations
function alpha=newalpha(alpha,delta)
alpha=alpha*delta;

% Make sure the fireflies are within the range

```

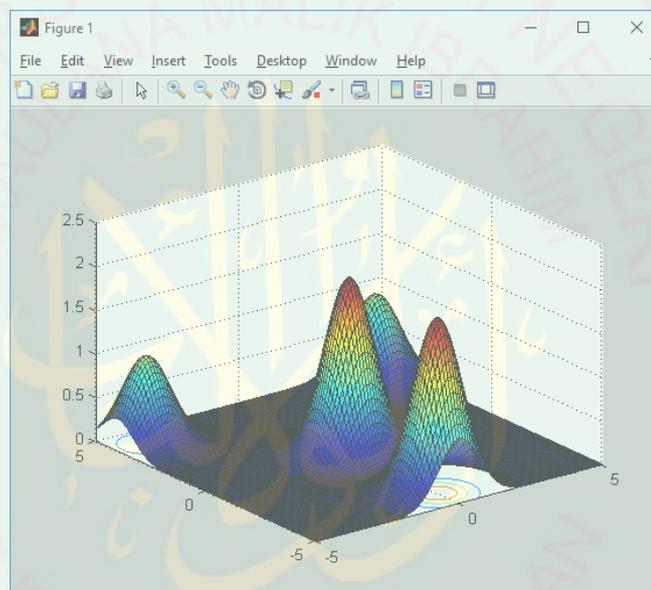
```

function [xn,yn]=findrange(xn,yn,range)
for i=1:length(yn),
    if xn(i)<=range(1), xn(i)=range(1); end
    if xn(i)>=range(2), xn(i)=range(2); end
    if yn(i)<=range(3), yn(i)=range(3); end
    if yn(i)>=range(4), yn(i)=range(4); end
end
% ===== end
=====

```

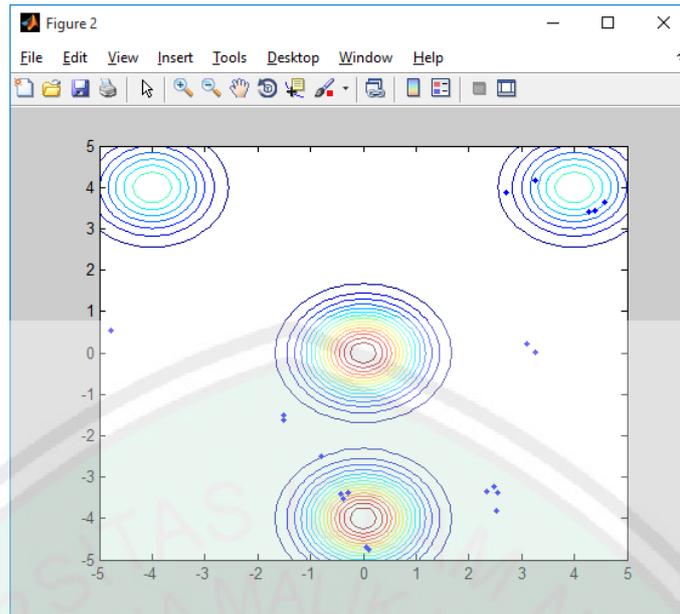
3.6. Hasil Perhitungan

1. Hasil simulasi perhitungan algoritma *Firefly* menggunakan Matlab 2011.

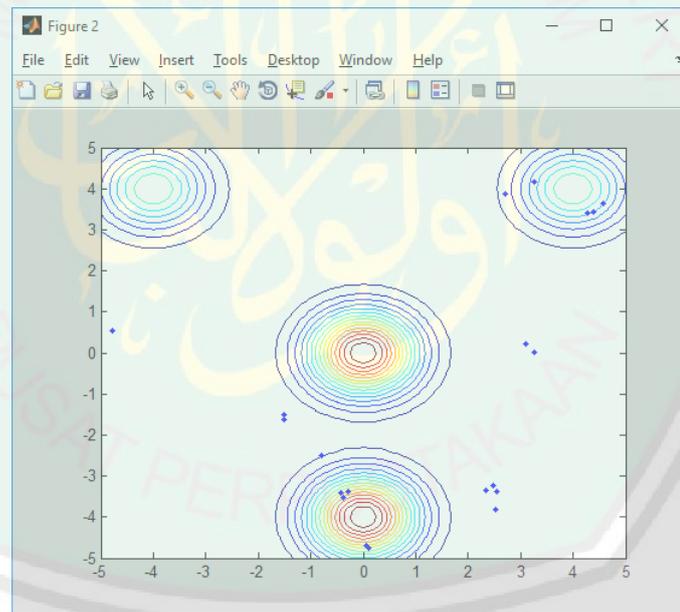


Gambar 3.3. Grafik Algoritma *Firefly*

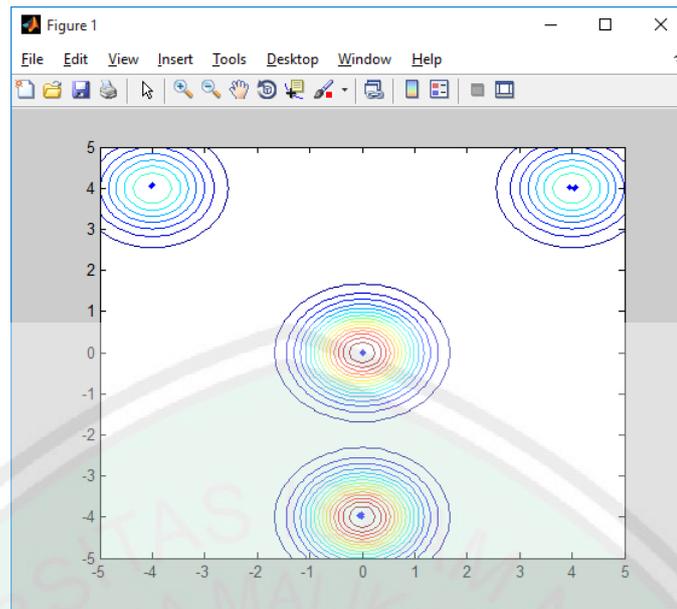
Pada **Gambar 3.3.** menampilkan grafik dari perhitungan *Firefly*. Grafik tersebut menunjukkan ketinggian tingkat intensitas cahaya yang dimiliki oleh kunang – kunang dan membuat kunang - kunang yang lain bergerak menuju kearah kunang – kunang tesebut.



Gambar 3.4. Pergerakan kumbang-kumbang (a)



Gambar 3.5. Pergerakan kumbang-kumbang (b)



Gambar 3.6. Pergerakan kunang-kunang (c)

Pada gambar **gambar 3.4** menunjukkan posisi awal dari gerombolan kunang – kunang, kemudian kunang – kunang tersebut mulai merespon dengan melakukan pergerakan menuju ke arah kunang – kunang yang memiliki intensitas cahaya yang tinggi sesuai pada **gambar 3.5**, setelah itu kunang – kunang akan berkumpul di titik dimana intensitas cahaya yang tinggi tersebut berada sesuai dengan **gambar 3.6**.

Tabel 3.2 Hasil Perhitungan Algoritma *Firefly*

No.	Location X	Location Y	Light	Iteration
1	-4.0065	4.0831	0.9930	50
2	4.0767	4.0237	0.9935	50
3	4.0536	4.0052	0.9970	50
4	3.9603	4.0272	0.9976	50
5	-4.0207	4.0392	0.9980	50
6	4.0356	3.9920	0.9986	50
7	-0.0670	-3.9611	1.9880	50
8	-0.0162	-3.9442	1.9932	50
9	0.01795	-0.0160	1.9988	50
10	-0.0208	-3.9926	1.9990	50
11	0.0067	0.0153	1.9994	50
12	0.0003	0.0047	v1.9999	50

BAB IV

HASIL DAN PEMBAHASAN

2.1. Implementasi

Pada bab ini, dibahas mengenai penerapan dari perencanaan yang sudah diusulkan. Implementasi merupakan suatu hal yang sangat penting dalam penelitian dan menjadi tolok ukur untuk mengetahui seberapa baik atau efektif penelitian tersebut. Implementasi melingkupi tahap uji coba pada *game* untuk mengukur apakah *game* yang dimaksud sesuai dengan tujuan penelitian yang ingin dicapai.

4.1.1. Kebutuhan Perangkat Keras

Pada penelitian ini digunakan perangkat keras sebagai media untuk mengimplementasikan *game*, adapun perangkat keras yang digunakan sebagai berikut:

Tabel 4.1 Kebutuhan Perangkat Keras

No.	Perangkat Keras	Spesifikasi
1.	Processor	2.7 GHz Intel Core i5
2.	RAM	8 GB 1867 MHz DDR3
3.	VGA	Intel Iris Graphics 6100 1536 MB
4.	SSD	125 GB
5.	Monitor	13.3 - inch
6.	Speaker	On
7.	Mouse & Keyboard	On

4.1.2. Kebutuhan Perangkat Lunak

Adapun perangkat lunak yang diperlukan untuk mengimplementasikan *game* ini, diuraikan sebagai berikut :

Tabel 4.2 Kebutuhan Perangkat Lunak

No.	Perangkat Lunak	Spesifikasi
1.	Sistem Operasi	macOS Sierra
2.	<i>Game Engine</i>	Unity3D 5.3.4f1
3.	Konsep desain 2D	Adobe Illustrator
4.	Desain 3D	Blender 2.7
5.	<i>Script Writer</i>	Mono Develop

2.1.3. Implementasi Algoritma Firefly pada NPC

Proses implementasi algoritma *Firefly* adalah proses penerapan sebuah sistem yang telah dibangun berdasarkan rancangan yang telah diusulkan. Pada tahap ini dilakukan implementasi kecerdasan buatan pada NPC dengan menggunakan algoritma *Firefly*. Algoritma *Firefly* digunakan sebagai algoritma *pathfinding* atau pencarian rute pada NPC. Posisi awal NPC akan dijadikan sebagai *starting point* dan posisi *player* menjadi target yang diinisiasi memiliki intensitas cahaya paling tinggi. Perhitungan algoritma *Firefly* dikalkulasi secara berkala menyesuaikan dengan posisi pemain.

2.2. Firefly Sebagai Pathfinding pada NPC

Pada bagian ini membahas tentang penerapan algoritma *Firefly* pada NPC sebagai *pathfinding*. Pada *game* ini NPC memiliki kemampuan untuk menemukan posisi *player*. Ketika NPC berada pada posisi yang sama dengan *player*, secara otomatis NPC akan melakukan serangan yang berakibat pada berkurangnya darah pemain. Misi dari *player* adalah melarikan diri dan bersembunyi dari kejaran NPC dan menyelesaikan tantangan pada *game*. Penjelasan penerapan *source code* algoritma *Firefly* diuraikan sebagai berikut :

Tabel 4.3 Keterangan Algoritma *Firefly*

No	Method / Fungsi	Keterangan
1.	<pre>function Awake() { anakbuah = transform; } function Start() { playertarget = GameOb ject.FindWithTag("Player") .transform; }</pre>	<p>Method yang digunakan untuk mengatur atau menentukan <i>player</i> sebagai target dari NPC</p>

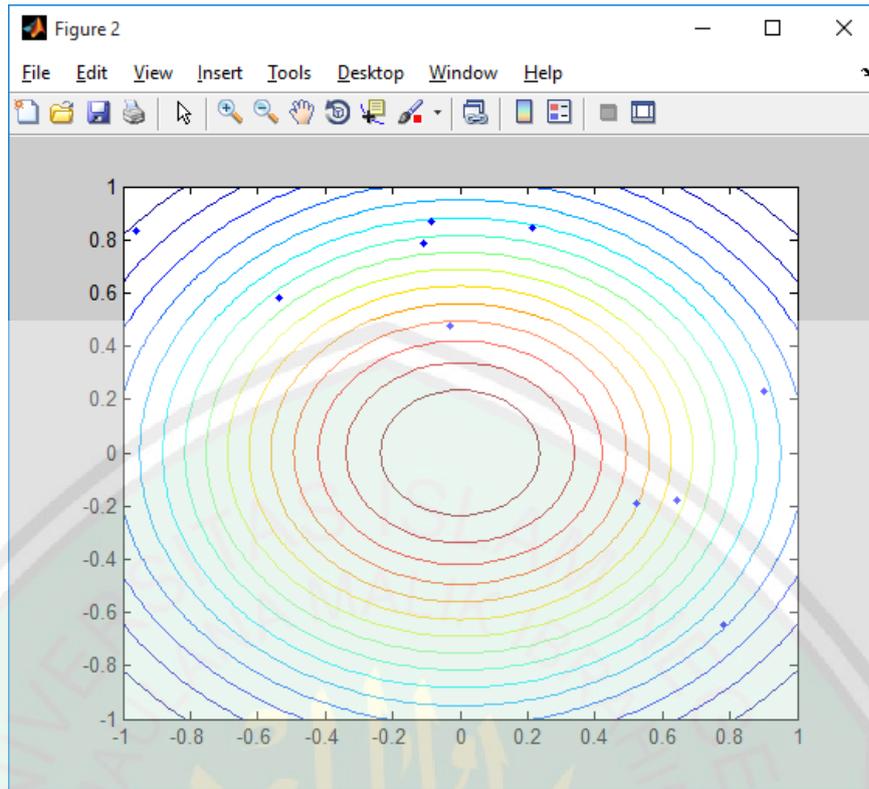
2.	<pre>function Update () { var distance = parseInt((p layertarget.position - anakbuah.position).magnit ude); if (distance<=1) { player_inrange('1'); }else if (distance<5) { player_inrange('1'); }else if (distance<10) { player_inrange('2'); }else if (distance<20) { player_inrange('3'); }else{ monster_random(); } energyBar.SendMessage ("Se tValueCurrent", bardarah); }</pre>	<p>Method yang digunakan untuk menentukan luas intensitas cahaya <i>firefly</i> pada <i>player</i></p>
3.	<pre>function enemy_attack(){ if(bardarah<=0){ Debug.Log("Kalah"); }else{ bardarah = bardarah -5; } } function player_inrange(id _speed){ anakbuah.rotation = Quater nion.Slerp(anakbuah.rotati on, Quaternion.LookRotation(pl ayertarget.position - anakbuah.position), rotat ionSpeed*Time.deltaTime); var fp=findplayer(); if(fp==1){ monster_to_player(id_speed);</pre>	<p>Method yang digunakan pada NPC untuk melakukan serangan kepada <i>Player</i> dengan</p>

	<pre> }else{ } } </pre>	
4.	<pre> function monster_to_player (id_speed){ var speedtime=slv3; if(id_speed=="1"){ speedtime=slv1; }else if(id_speed=="2"){ speedtime=slv2; }else{ speedtime=slv3; } anakbuah.rotation = Quater nion.Slerp(anakbuah.rotati on, Quaternion.LookRotation(pl ayertarget.position - anakbuah.position), rotat ionSpeed*Time.deltaTime); </pre>	<p><i>Method</i> yang digunakan untuk mengatur kecepatan dan membuat NPC (<i>firefly</i>) melihat ke arah <i>player</i> ketika berada pada jarak</p>
5.	<pre> anakbuah.position += anakb uah.forward * speedtime * Time.deltaTime; } function monster_random(){ anakbuah.Translate(Vector3 (0, 0, 2)*speedrandommove* Time.deltaTime); var randomint=Random.Range (0,360); var fwd=anakbuah.Transform Direction(Vector3.forward) ; var hit:RaycastHit; Debug.DrawRay(anakbuah.pos ition,fwd*3,Color.red); if(Physics.Raycast(anakbua h.position,fwd,hit,3)){ print(hit.collider.tag); if(hit.collider.tag=="Peng halang" hit.collider.ta g=="Untagged"){ anakbuah.Rotate(0,randomin t,0); } } } </pre>	<p><i>Method</i> yang digunakan untuk menggerakkan NPC (<i>firefly</i>) secara random ketika tidak melihat cahaya dan mengarahkan NPC ke arah <i>Player</i> jika sudah berada pada jangkauan <i>player</i></p>

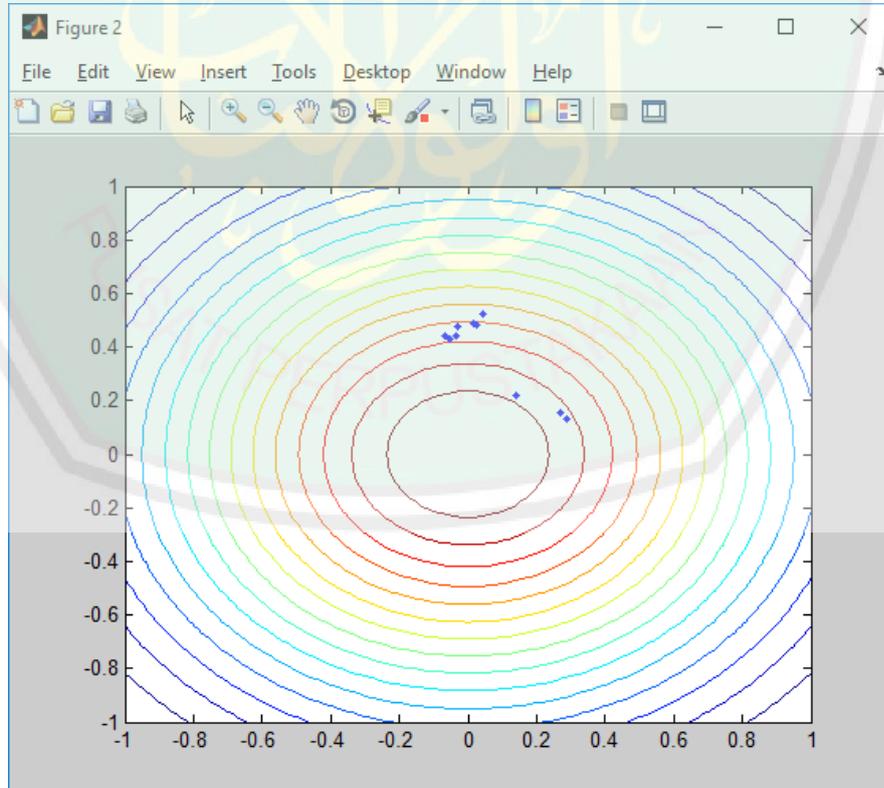
<pre> } function findplayer(){ var objectplayer=0; var hit : RaycastHit; if(Physics.Raycast(anakbuah.position, anakbuah.TransformDirection(Vector3.forward) , hit, 20)) { if(hit.collider.gameObject.tag == "Player"){ objectplayer=1; }else{ objectplayer=0; } } return objectplayer; } </pre>	
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

4.3 Pengujian Game

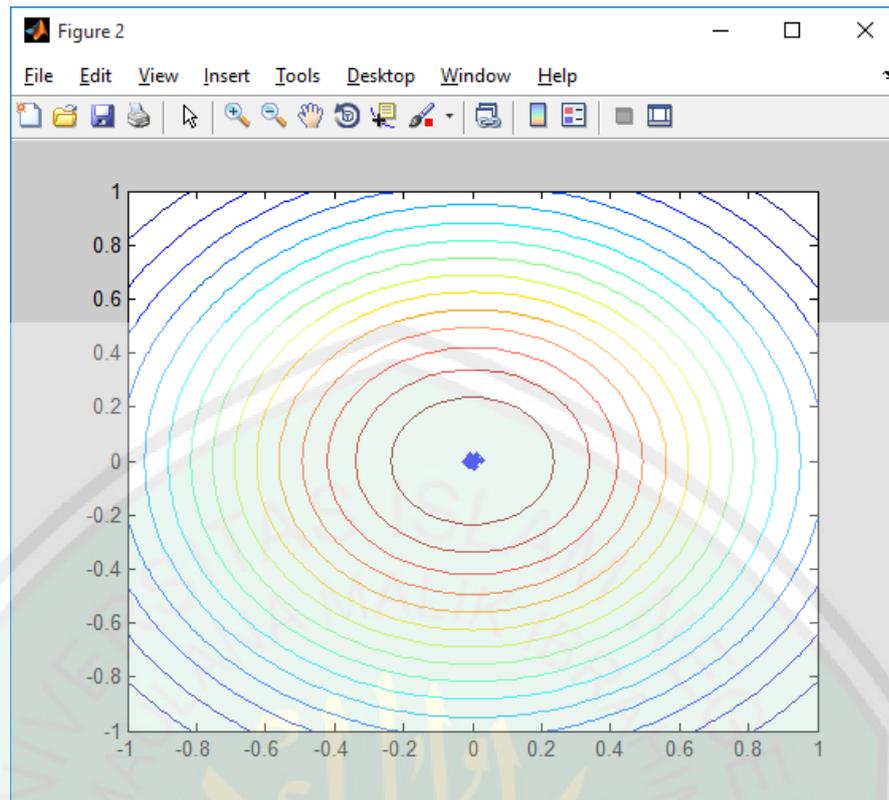
Pengujian ini dilakukan untuk mengetahui bagaimana NPC melakukan pergerakan menuju ke tempat *player* berada. Pengujian ini dilakukan menggunakan 11 NPC yang terdiri dari 10 NPC menggunakan algoritma *firefly* dan 1 NPC menggunakan algoritma *Astar* sebagai perbandingan, dan seluruh NPC tersebar di dalam map dimana setiap NPC yang menggunakan algoritma *firefly* memiliki kecepatan bergerak yang relatif sama dan akan bergerak lebih cepat jika sudah berada pada jarak tertentu.



Gambar 4.1 Pergerakan kunang-kunang (a)

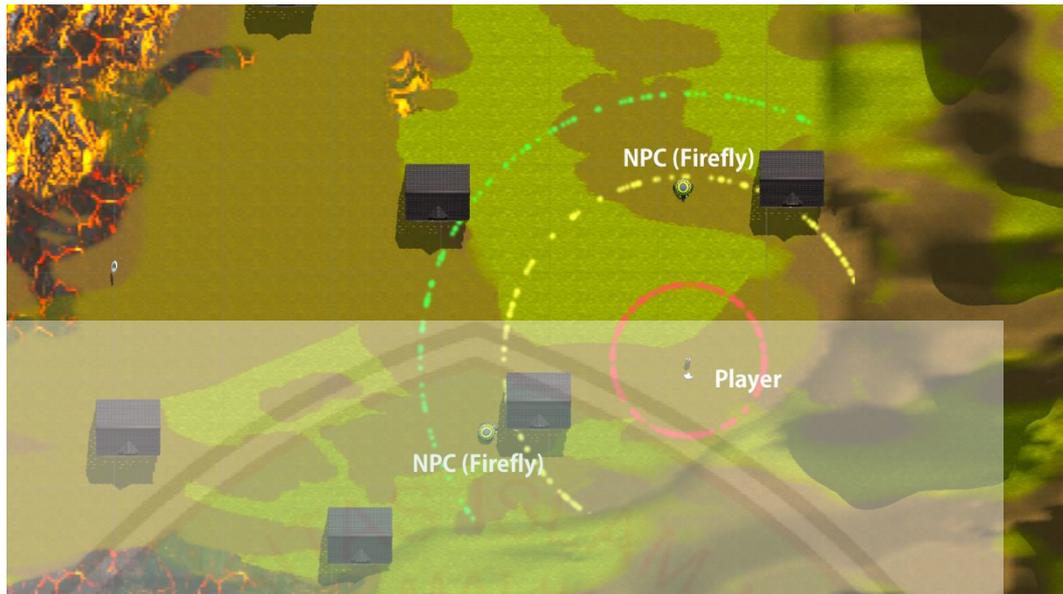


Gambar 4.2 Pergerakan kunang-kunang (b)

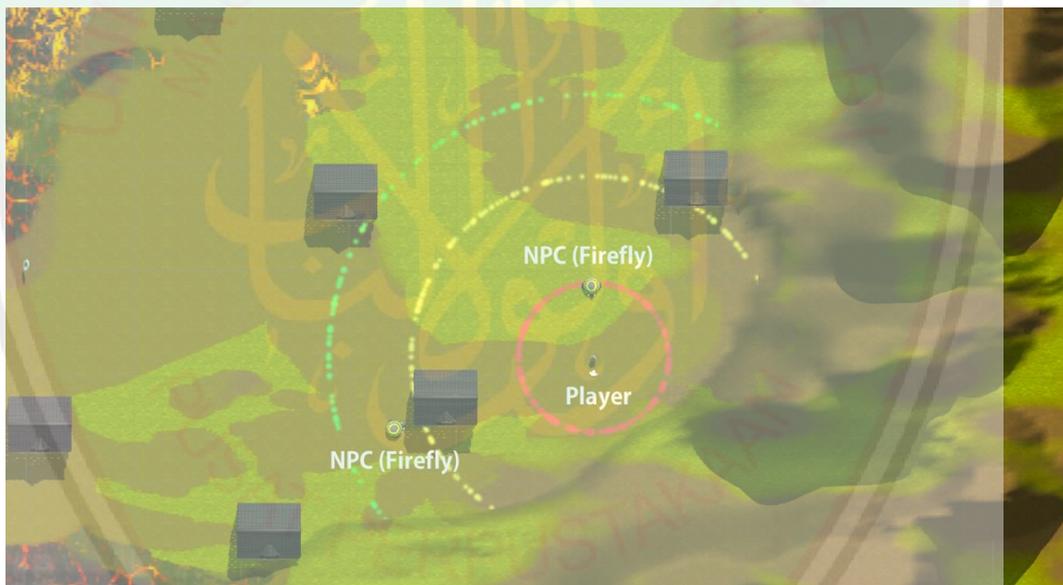


Gambar 4.3 Pergerakan kunang-kunang (c)

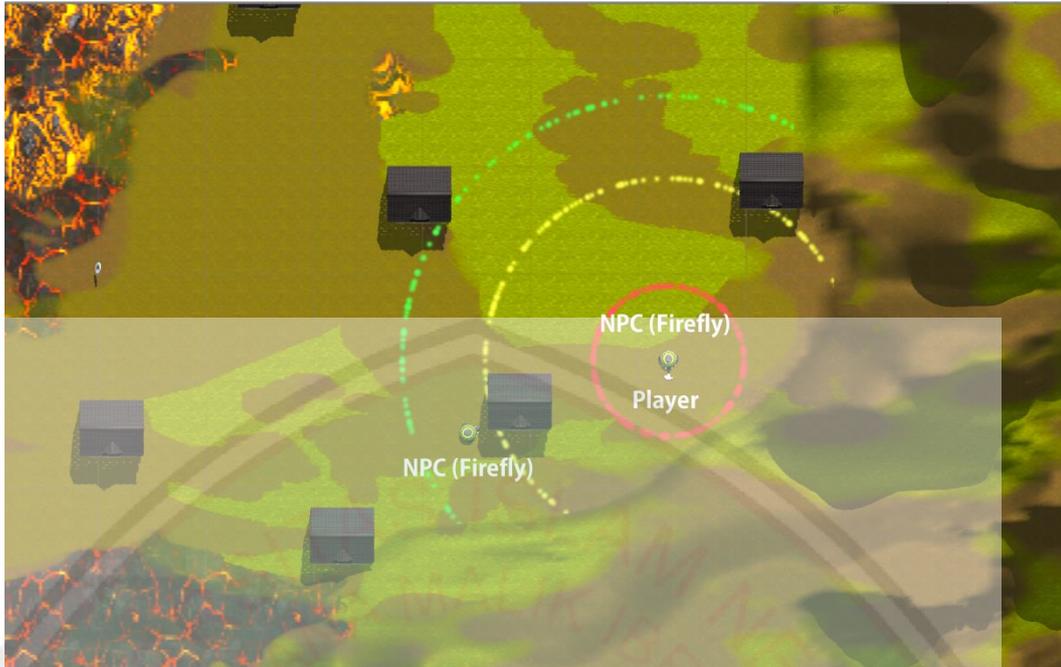
Grafik pada **gambar 4.1** hingga **gambar 4.3** merupakan visualisasi hasil perhitungan algoritma *firefly* menggunakan matlab dan kemudian di visualisasikan pada *game* 3D Adventure Wayang, dalam percobaan tersebut memberikan gambaran bahwa kunang – kunang bergerak ke arah intensitas cahaya yang paling terang, dimana cahaya tersebut dalam *game* ini adalah *player* sedangkan kunang – kunang yang sedang bergerak menuju ke arah cahaya adalah NPC atau Musuh.



Gambar 4.4 Posisi Awal NPC *firefly* dan *Player*



Gambar 4.5 Pergerakan NPC *firefly* mengarah ke *Player*



Gambar 4.6 NPC *firefly* tiba di posisi *Player*



Gambar 4.7 Posisi Awal NPC A* dan *Player*



Gambar 4.8 NPC A* bergerak ke arah *Player* (a)



Gambar 4.9 NPC A* Bergerak Ke Arah *Player* (b)



Gambar 4.10 NPC A* Berada Pada Posisi *Player*

Dengan menerapkan algoritma kunang - kunang diharapkan NPC tersebut dapat bergerak menuju ke lokasi *player* jika sudah berada pada jangkauan *player*. Pengujian tersebut dilakukan untuk mengetahui apakah algoritma telah berfungsi sehingga dapat mempengaruhi perilaku atau pergerakan pada NPC.

NPC dapat mengejar *player* ketika berada dalam jangkauan (intensitas cahaya) *player* sesuai pada **gambar 4.4**, terlihat sebuah NPC bergerak ke arah *player* dan sebuah NPC yang tidak bergerak ke arah *player* meskipun berada dalam jangkauan, karena NPC tersebut terhalang oleh objek, sedangkan NPC lainnya bergerak secara acak karena tidak berada dalam jangkauan. Hal ini menunjukkan bahwa untuk setiap iterasinya, NPC telah melakukan pergerakan menuju posisi yang lebih optimal.

Sebagai pembanding maka diterapkan pula algoritma *Astar* pada salah satu NPC untuk mengetahui tingkat efisiensi dari algoritma *firefly*. Sesuai pada

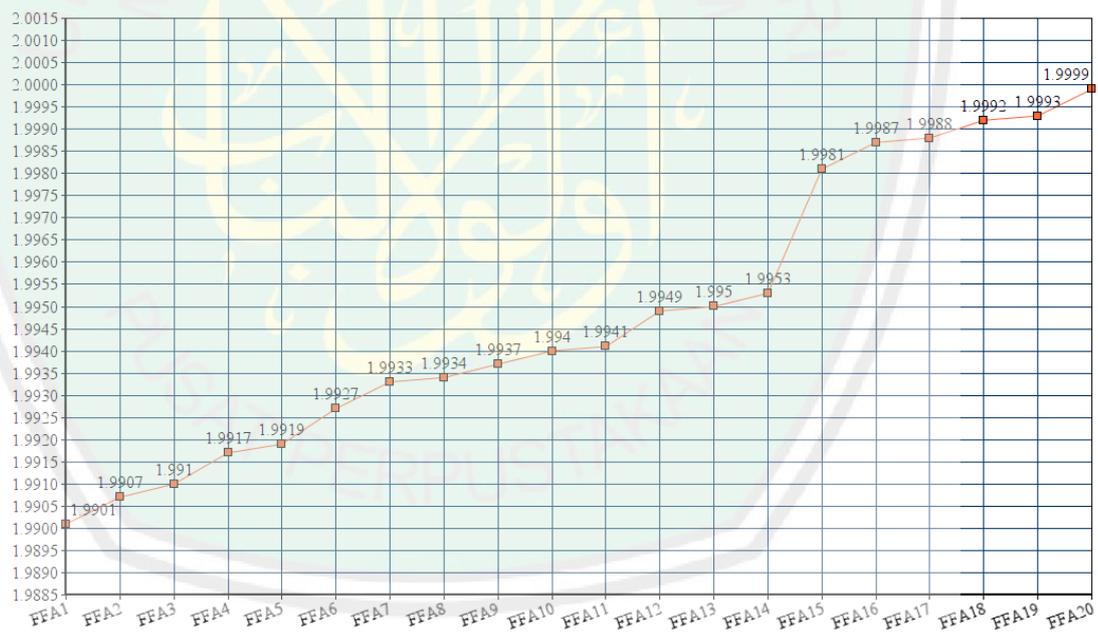
gambar 4.7 sampai dengan **gambar 4.10** yang menggambarkan visualisasi pengaplikasian algoritma *Astar* pada NPC dari *start position* hingga NPC berhasil berada pada posisi *player*, dimana *start position* dari NPC tersebut lebih jauh dari NPC lainnya.

Tabel 4.4 Hasil Pengujian Pergerakan NPC dengan Firefly

No.	Location X	Location Y	Light	Jarak Tempuh (m)	Iteration
1	-0.0207	-0.0670	1.9901	87	17
2	0.0504	-0.0456	1.9907	85	16
3	0.0382	0.0548	1.9910	83	16
4	0.0625	0.0143	1.9917	82	15
5	0.0518	-0.0367	1.9919	82	15
6	-0.0132	-0.0586	1.9927	80	15
7	0.0226	-0.0530	1.9933	77	15
8	0.0555	0.0145	1.9934	76	15
9	0.0538	0.0155	1.9937	76	14
10	0.0226	-0.0495	1.9940	75	14
11	0.0437	0.0316	1.9941	75	14
12	0.0439	-0.02472	1.9949	75	13

13	-0.0255	-0.0425	1.9950	74	13
14	-0.0249	-0.04150	1.9953	74	12
15	-0.0086	-0.0287	1.9981	73	12
16	-0.0206	-0.0141	1.9987	73	11
17	-0.0100	-0.0214	1.9988	73	10
18	0.0112	0.0163	1.9992	72	10
19	-0.0136	0.0116	1.9993	72	10
20	-0.0028	-0.0025	1.9999	71	10

Tabel 4.5 Grafik Pergerakan firefly dipengaruhi oleh intensitas cahaya



Tabel 4.6 Perbandingan Tingkat Akurasi

No.	Pengujian	Tingkat Akurasi Firefly (%)	Tingkat Akurasi Astar (%)
1	Pengujian ke 1	75.71	75.22
2	Pengujian ke 2	78.57	77.53
3	Pengujian ke 3	81.43	80.68
4	Pengujian ke 4	82.86	81.55
5	Pengujian ke 5	82.86	81.94
6	Pengujian ke 6	85.71	83.64
7	Pengujian ke 7	90.00	84.97
8	Pengujian ke 8	91.43	85.24
9	Pengujian ke 9	91.43	87.94
10	Pengujian ke 10	92.86	88.89
11	Pengujian ke 11	92.86	89.24
12	Pengujian ke 12	92.86	89.89
13	Pengujian ke 13	94.29	90.94
14	Pengujian ke 14	94.29	91.40
15	Pengujian ke 15	95.71	91.83
16	Pengujian ke 16	95.71	93.82
17	Pengujian ke 17	95.71	93.97
18	Pengujian ke 18	97.14	94.24
19	Pengujian ke 19	97.14	94.92
20	Pengujian ke 20	98.57	95.19

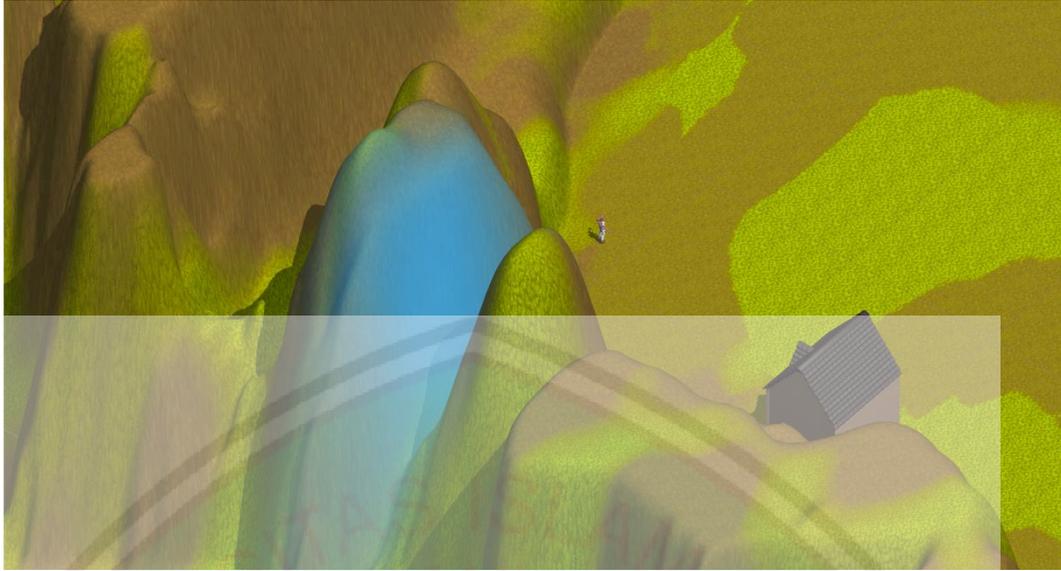
Tabel 4.4. merupakan hasil perhitungan pada algoritma *firefly*, pada percobaan tersebut menggunakan 10 kunang – kunang dengan 15 iterasi di mana X adalah koordinat pada sumbu X, dan Y adalah koordinat dari sumbu Y yang merupakan *best location* bagi setiap *firefly*, Sedangkan *Light* merupakan tingkat intensitas cahaya yang nilai maksimalnya adalah 2. Sesuai pada **tabel 4.5** menunjukkan grafik tingkat akurasi dari setiap *firefly* dimana menunjukkan peningkatan yang setiap nilainya mendekati 2 dan merupakan tingkat maksimum dari intensitas cahaya. Berdasarkan tabel tersebut dapat disimpulkan bahwa pergerakan dari *firefly* sangat dipengaruhi oleh tingkat intensitas cahaya dan Hal tersebutlah yang menunjukkan tingkat akurasi dari sebuah *Firefly*.

4.4 Implementasi Aplikasi Game

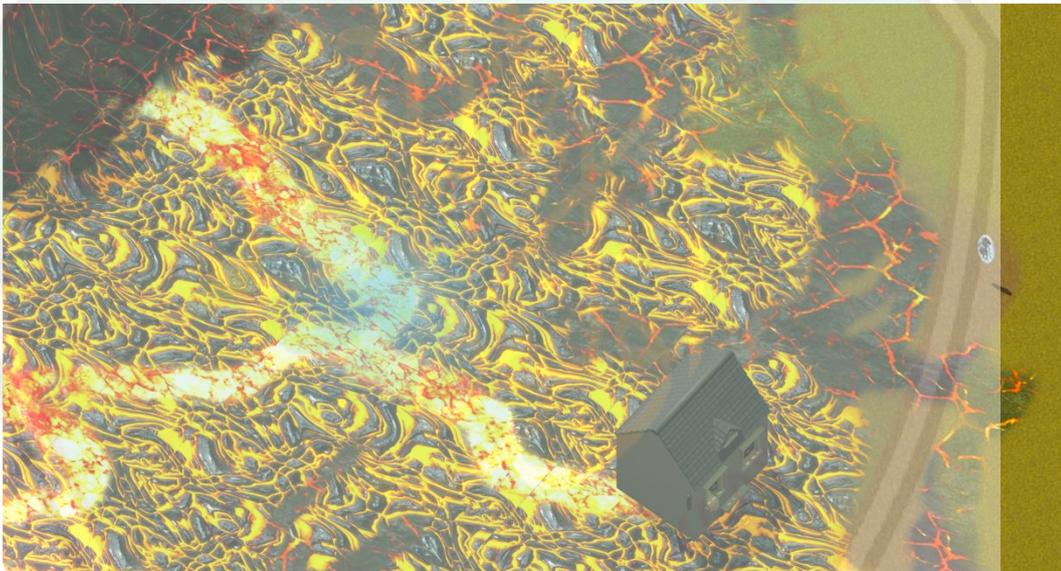
Implementasi aplikasi *game* merupakan tahapan pembangunan komponen-komponen utama suatu sistem. Komponen yang dimaksud dikembangkan berdasarkan hasil perancangan yang telah diusulkan sebelumnya.

4.4.1 Pembangunan Terrain

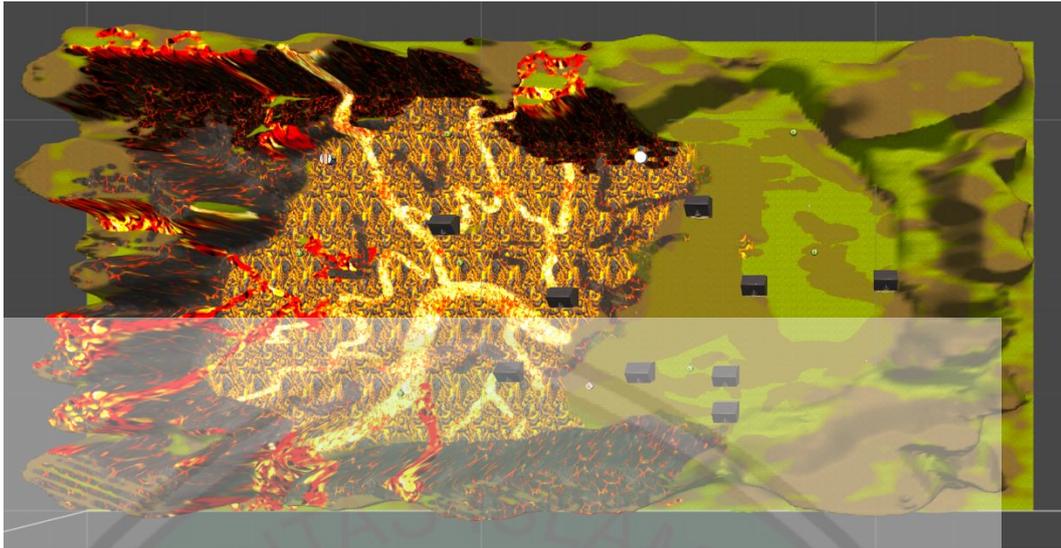
Terrain merupakan sebuah arena yang menjadi *stage* dari sebuah *game*. Karakter dari *player* dan musuh akan diletakkan dalam *terrain* tersebut. Pada pembangunan *terrain* ditentukan juga posisi awal dari *player* dan juga musuh ketika memulai *game* pertama kali.



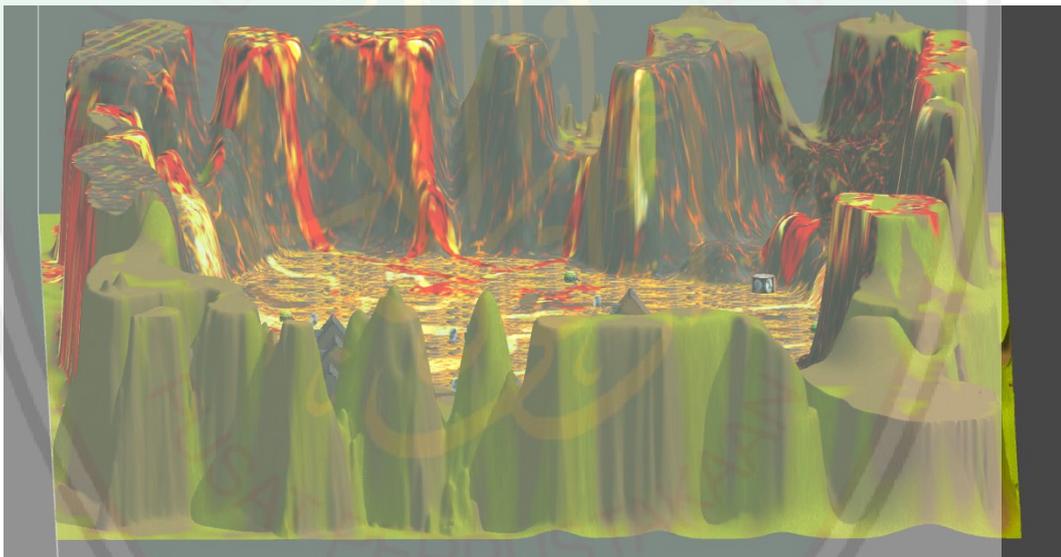
Gambar 4.11 membuat bukit pada *Terrain*



Gambar 4.12 Pemberian Efek Lava Pada *Terrain*



Gambar 4.13 Tampak Atas Pada *Terrain*



Gambar 4.14 Tampak Samping Pada *Terrain*

4.4.2 Halaman *Splashscreen*

Halaman *splashscreen* digunakan sebagai transisi ketika memulai *game* pada saat pertama kali, *splashscreen* membuat tampilan awal pada *game* terlihat lebih menarik dan memberikan kesan yang lebih dramatis ketika *game* dijalankan.



Gambar 4.15 Halaman *Splashscreen*

4.4.3 Antarmuka Menu

Tampilan antarmuka menu pada *game* ini terdapat 4 *button* sebagai menu utama, antara lain *Play*, *Gallery*, *Option* dan *Guideline*. Adapun masing – masing *button* berfungsi sebagai berikut :

1. *Play* berfungsi untuk memulai *game* ketika kita sudah berada di halaman antarmuka menu.
2. *Gallery* berisi informasi tentang beberapa tokoh wayang
3. *Option* berfungsi sebagai menu pengaturan pada *game*
4. *Guideline* berisi petunjuk cara memainkan *game*



Gambar 4.16 Tampilan Menu Screen

Main menu atau menu utama sendiri berfungsi sebagai gambaran atau petunjuk kepada *user* tentang profil sebuah *game* dan juga tempat *user* mengakses berbagai macam informasi pada *game* tersebut.

Tabel 4.7 Function Pemanggilan Scene

No.	Method / Function	Keterangan
1	<pre>function PlayGame () { Application.LoadLevel (1) ; } function QuitGame () { Application.Quit () ; }</pre>	<p>Function yang digunakan untuk memulai dan mengakhiri scene pada game</p>

4.4.4 Menu Petunjuk

Pada menu petunjuk dijelaskan beberapa hal tentang isi dari *game* seperti misi yang harus dicapai ketika bermain dan apa saja yang harus dilakukan dalam *game* tersebut.



Gambar 4.17 Halaman Petunjuk

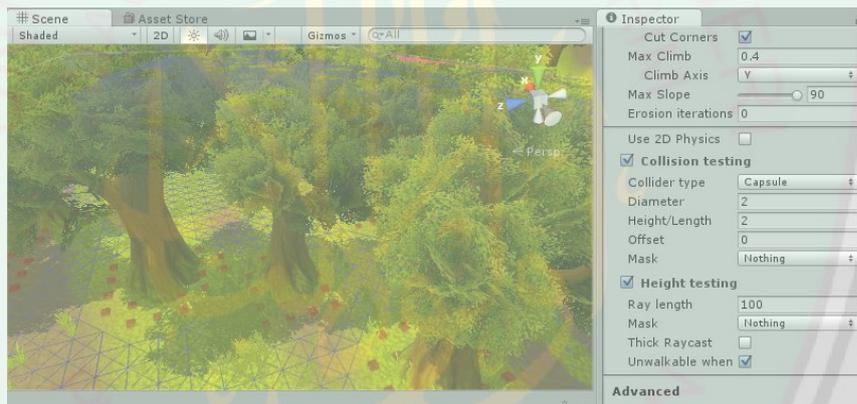
4.4.5 Pengaturan Terrain

Terrain berperan penting dalam sebuah *game*. Maka dari itu perlu dilakukan beberapa pengaturan agar sesuai dengan alur *game* yang sedang dibangun, pengaturan tinggi dan rendah suatu terrain, penambahan objek sebagai *obstacle* seperti pohon, rumah, dan rumput juga penting sebagai bahan penilaian untuk menguji kinerja kecerdasan buatan yang diterapkan. Di sisi lain, penambahan objek tersebut juga menambah nilai estetika pada *game*.



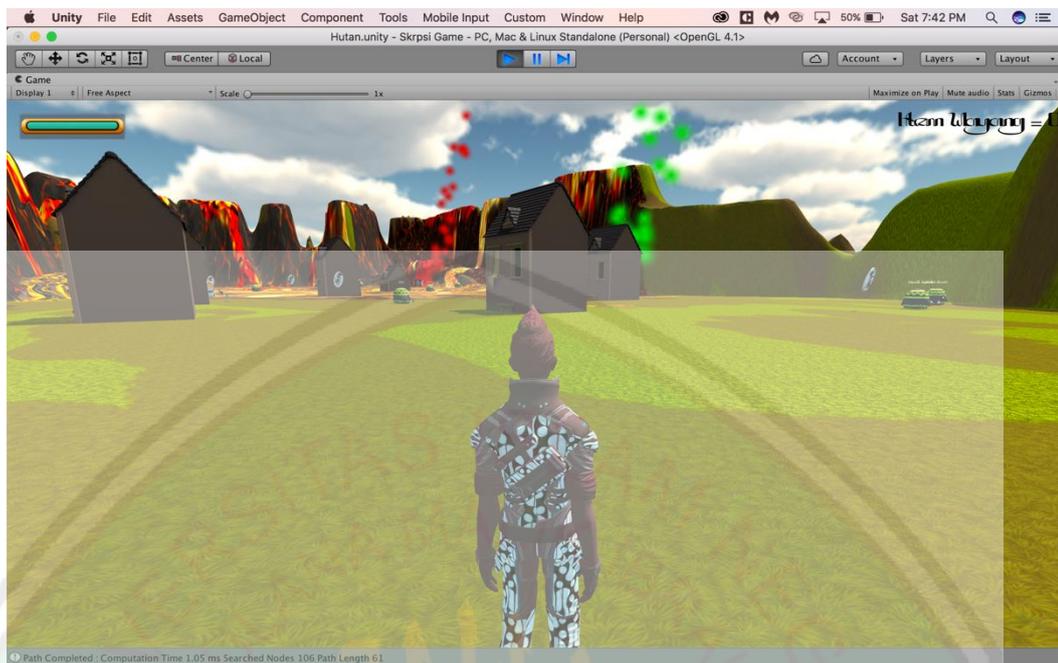
Gambar 4.18 Pengaturan *Terrain*

4.4.6 Pembuatan *Obstacle*



Gambar 4.19 *Obstacle* Permainan

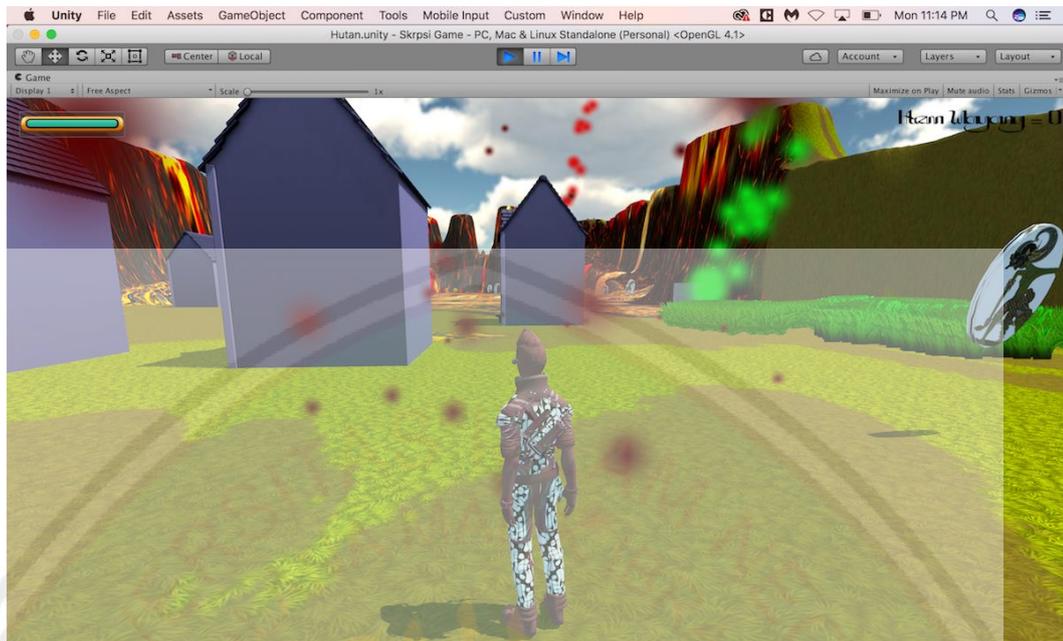
4.4.7 Scene Game pada Bagian Awal



Gambar 4.20 Tampilan Game Pada bagian Awal

Pada bagian awal permainan, *player* akan berada pada titik awal, dan menjalankan misi berupa mencari *item* wayang, pada **gambar 4.20** menunjukkan tampilan awal pada saat *game* telah berjalan dan terlihat *player* memasuki map. *Player* dapat bergerak bebas di area map dan menghindari NPC dengan cara berlari atau bahkan terbang sebelum NPC melakukan *attack* kepada *player*.

4.4.8 Scene Game menjalankan misi



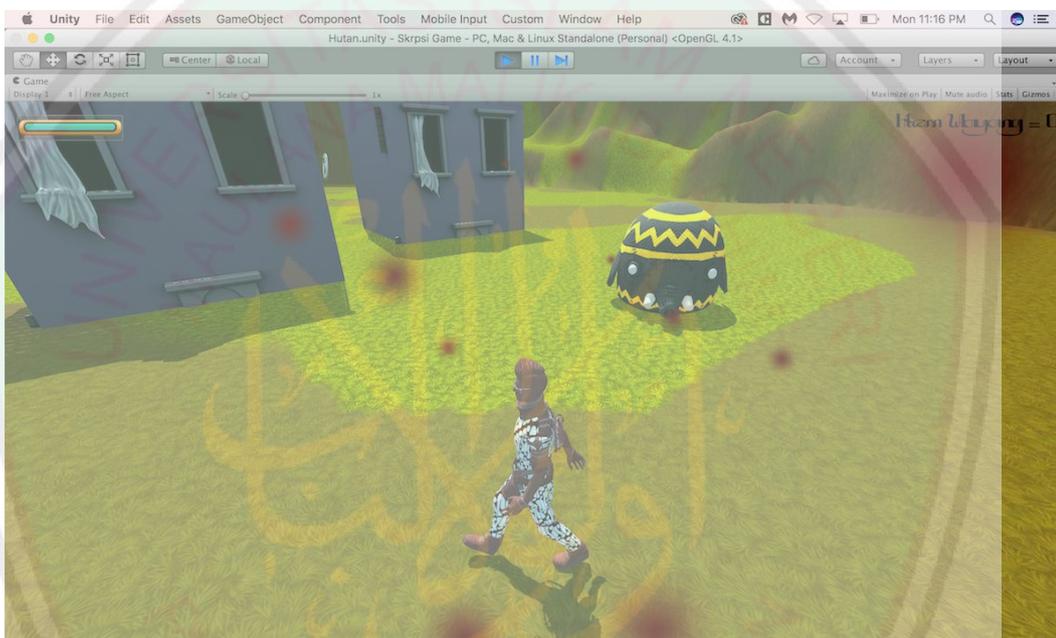
Gambar 4.21 Tampilan *Game* Pada Saat Menjalankan Misi



Gambar 4.22 Tampilan *Game* Pada Saat Menemukan Item Wayang

4.4.9 Scene Game Saat NPC Mengejar

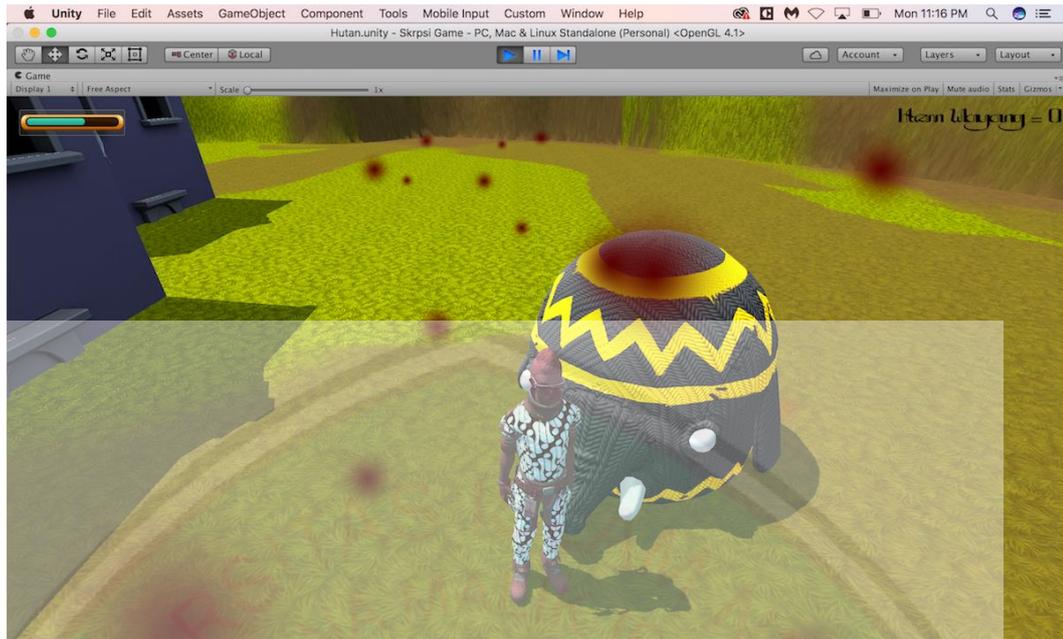
Pada saat pemain berada dalam jangkauan NPC, maka NPC akan mengamati dan mulai mengejar, dan pada jarak tertentu kecepatan NPC akan semakin bertambah ketika mengejar, ketika NPC terhalang dengan objek tertentu maka secara otomatis NPC tidak akan mengejar *player*, setelah *player* bergerak menjauh dari NPC maka secara langsung NPC akan bergerak secara acak di dalam map.



Gambar 4.23 Tampilan Game Saat NPC Mengejar

4.4.10 Scene Game Saat NPC Menyerang

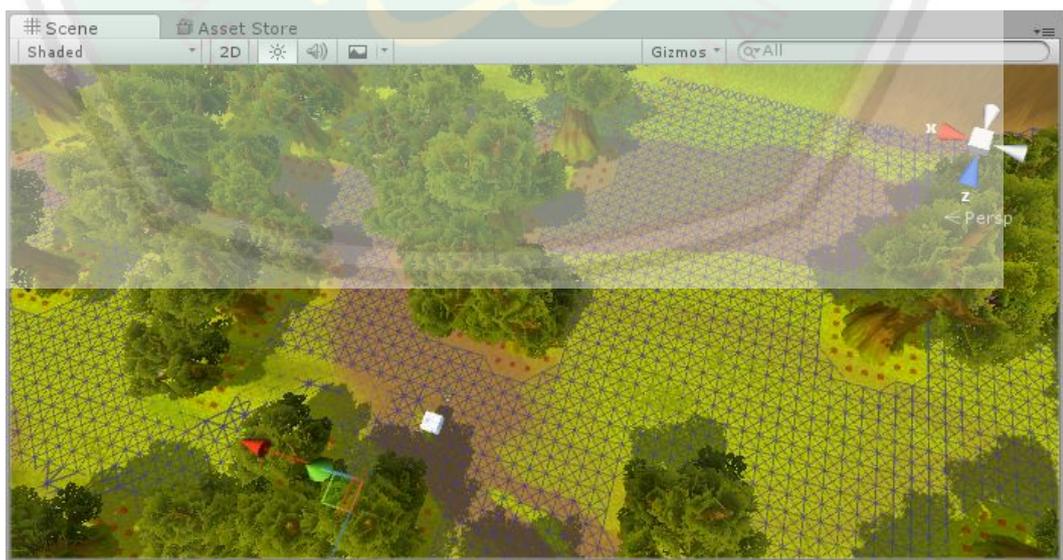
Jarak merupakan salah satu faktor yang menjadi *trigger* bagi NPC untuk mengejar *player*. Apabila *player* berada dalam jangkauan serang dari NPC, maka secara langsung NPC akan menyerang dan membuat sejumlah HP atau darah dari *player* berkurang.



Gambar 4.24 Tampilan *Game* Saat NPC Menyerang

Pada **gambar 4.24** NPC melakukan *attack* kepada *player* dan tampak sejumlah darah dari *player* berkurang secara berkala sesuai dengan nilai kekuatan *attack* dari NPC tersebut. Adapun *source code* pada saat NPC melakukan *attack* kepada *player*, sebagai berikut :

4.4.11 Lintasan NPC



Gambar 4.25 Tampilan *Graph* Lintasan yang Bisa Dilalui NPC

4.4.12 Keterangan Item Wayang

Berikut adalah beberapa item yang berisi informasi terkait wayang – wayang yang terdapat di dalam *game*, ketika *player* berhasil menemukan item tersebut maka akan tampil informasi mengenai karakter wayang berupa kekuatan ataupun karakteristik dari tokoh wayang tersebut.

a. Bima



Gambar 4.26 Karakter Wayang Bima

b. Arjuna



Gambar 4.27 Karakter Wayang Arjuna



Gambar 4.28 Karakter Wayang Arjuna 2

c. Sangkuni



Gambar 4.29 Karakter Wayang Sangkuni

d. Nakula



Gambar 4.30 Karakter Wayang Nakula

e. Sadewa



Gambar 4.31 Karakter Wayang Sadewa

f. Duryudana



Gambar 4.32 Karakter Wayang Duryudana

g. Citraksa



Gambar 4.33 Karakter Wayang Citraksa

h. Citraski



Gambar 4.34 Karakter Wayang Citraksi

i. Durmagati



Gambar 4.35 Karakter Wayang Durmagati

j. Karna



Gambar 4.36 Karakter Wayang Karna

k. Yudistira



Gambar 4.37 Karakter Wayang Yudistira

Tabel 4.8 *Function* Unity Untuk Memunculkan Item Wayang

No.	Method / Function	Keterangan
1	<pre> var itemwayang = 0; var style : GUIStyle; function OnTriggerEnter(item : Collider) { if (item.tag == "Item") { itemwayang += 1; } </pre>	Method atau Function ini digunakan untuk menampilkan <i>item</i> wayang ketika

<pre> } function OnGUI () { GUI.Label (Rect (Screen.width - 270,30,100,50), "Item Wayang = " + itemway ang, style); } </pre>	<p><i>player</i> menyentu h <i>item</i> yang tersebar di map</p>
--------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------

Pada saat *player* menemukan *item* wayang maka akan muncul gambar atau *image* tentang informasi terkait wayang tersebut dan untuk menampilkannya digunakan *source code* sesuai pada **table 4.6**.

4.5 Integrasi Dalam Islam

Belajar merupakan hal yang utama bagi seluruh umat manusia khususnya yang beragama Islam. Manusia memiliki kewajiban untuk senantiasa belajar mulai dari lahir hingga akhir hayat. Tugas utama manusia adalah menjadi manusia yang memahami perannya sebagai manusia. Peran yang terkandung dalam hubungan manusia secara vertikal (*hablum minallah*) dan hubungan manusia secara horizontal (*hablum minannas*). Inti sari hubungan manusia dengan tujuannya secara sempurna sudah terkandung dalam *al-Quran* dan *al-Hadist*. Sementara itu, kajian mendalam terkait *hablum minannas* salah satu perwujudannya adalah cinta terhadap tanah air. Pelestarian budaya merupakan bagian dari cinta tanah air tersebut.

Nenek Moyang suku Jawa membagi kebijaksanaan hidup mengenai pelestarian ke dalam kata bijak yang berbunyi : *mamayu hayuning bawana* yang berarti bahwa manusia berkewajiban untuk selalu memperindah dunia yang

sejatinya sudah sangat cantik dengan menjaga dan menjadikannya lestari, untuk diwariskan ke generasi lanjut. Pelestarian yang dimaksud tentunya juga termasuk segala macam karya, budaya dan sejarah panjang tanah air yang mampu terus dikaji dan dipelajari untuk senantiasa membangkitkan kebijaksanaan.

Al-Quranul kariim juga membawa napas yang sama terkait pengkajian untuk terus belajar menemukan hikmah. Allah *Subhanahu wa Ta'ala* membingkainya dalam kandungan Qs Az Zumar ayat 9.

أَمَّنْ هُوَ قَانِتٌ آنَاءَ اللَّيْلِ سَاجِدًا وَقَائِمًا يَحْذَرُ الْآخِرَةَ وَيَرْجُو رَحْمَةَ رَبِّهِ ۗ قُلْ هَلْ يَسْتَوِي الَّذِينَ يَعْلَمُونَ وَالَّذِينَ لَا يَعْلَمُونَ ۗ إِنَّمَا يَتَذَكَّرُ أُولُو الْأَلْبَابِ

Artinya :

(Apakah kamu hai orang-orang musyrik yang lebih beruntung) ataukah orang yang beribadah di waktu-waktu malam dengan sujud dan berdiri, sedangkan ia takut kepada (azab) akhirat dan mengharapkan rahmat Tuhannya? Katakanlah, "Adakah sama orang-orang yang mengetahui dengan orang-orang yang tidak mengetahui?" Sesungguhnya orang yang berakallah yang dapat menerima pelajaran.

Bersama lantunan indah ayat ke sembilan surat Az Zumar, Allah SWT memberikan keistimewaan kepada siapa saja yang menanamkan keinginan untuk mencari tahu demi menemukan kebenaran melalui proses belajar.

Di antara manfaat mempelajari kisah-kisah terdahulu adalah sebagai salah satu wujud usaha mencari jalan meraih ridhaNya. Adapun salah satu hadits yang diriwayatkan oleh Imam Muslim dalam shahihnya, dari hadits Abu Hurairah

radhiyallahu ‘anhu. Secara khusus membahas menuntut ilmu, sesungguhnya Nabi shallallahu ‘alaihi wa sallam bersabda :

وَمَنْ سَلَكَ طَرِيقًا يَلْتَمِسُ فِيهِ عِلْمًا سَهَّلَ اللَّهُ لَهُ بِهِ طَرِيقًا إِلَى الْجَنَّةِ، وَمَا اجْتَمَعَ قَوْمٌ فِي بَيْتٍ مِنْ بُيُوتِ اللَّهِ يَتْلُونَ كِتَابَ اللَّهِ وَيَتَدَارَسُونَهُ بَيْنَهُمْ إِلَّا نَزَلَتْ عَلَيْهِمُ السَّكِينَةُ، وَعَشِيَتْهُمُ الرَّحْمَةُ، وَحَفَّتْهُمُ الْمَلَائِكَةُ، وَذَكَرَهُمُ اللَّهُ فِيمَنْ عِنْدَهُ

Artinya :

“Barangsiapa yang menempuh suatu perjalanan dalam rangka untuk menuntut ilmu maka Allah akan memudahkan baginya jalan ke surga. Tidaklah berkumpul suatu kaum di salah satu masjid di antara masjid-masjid Allah, mereka membaca Kitabullah serta saling mempelajarinya kecuali akan turun kepada mereka ketenangan dan rahmat serta diliputi oleh para malaikat. Allah menyebut-nyebut mereka dihadapan para malaikat.”

Pada sejarah panjang umat terdahulu, tercatat banyak sekali nilai yang dapat dijadikan teladan untuk menempuh perjalanan selanjutnya. Salah satu manfaat yang luar biasa dari proses belajar adalah diistimewakan derajatnya oleh Allah Swt. Sebagaimana yang termaktub pada *Al-Quran* surat *Al-Mujaadilah* ayat 11:

يَأْتِيهَا الَّذِينَ ءَامَنُوا إِذَا قِيلَ لَكُمْ تَفَسَّحُوا فِي الْمَجَالِسِ فَافْسَحُوا يَفْسَحِ اللَّهُ لَكُمْ وَإِذَا قِيلَ أَدْنُوا فَأَنْشُرُوا يَرْفَعِ اللَّهُ الَّذِينَ ءَامَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ

Terjemahan ayat : *Hai orang-orang beriman apabila kamu dikatakan kepadamu:*

"Berlapang-lapanglah dalam majlis", maka lapangkanlah niscaya Allah akan

memberi kelapangan untukmu. Dan apabila dikatakan: "Berdirilah kamu", maka berdirilah, niscaya Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat. Dan Allah Maha Mengetahui apa yang kamu kerjakan. (QS. al-Mujaadilah ayat 11)

Tuntunan untuk senantiasa berusaha terus mengkaji dan meneliti juga terangkum dalam Qs Thaaha ayat 114.

وَقُلْ رَبِّ زِدْنِي عِلْمًا

Artinya :

Dan katakanlah: "Ya Rabbku, tambahkanlah kepadaku ilmu pengetahuan".

[Thaha :114]

Berkata Ibnu `Uyainah rahimahullah.

"Rasulullah Shallallahu 'alaihi wa sallam senantiasa bertambah ilmunya sampai Allah Subhanahu wa Ta'ala mewafatkan beliau" [Tafsir Ibnu Katsir 3/167]

Melalui permainan ini, pengguna diharapkan mampu mendapatkan hikmah dan tauladan dari masing-masing karakter wayang bersama segala nilai sejarah yang melekat. Dasar tentang informasi karakter wayang disarikan dari berbagai sumber pewayangan yang sudah melekat di masyarakat. Semoga dengan hadirnya permainan ini, menjadikan pemainnya mampu mengambil banyak kebaikan yang membaikkan, sebagai pemicu untuk terus mengupayakan kebaikan.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil dari implementasi dan pengujian yang dilakukan peneliti, maka ditarik kesimpulan bahwa penelitian ini berhasil menerapkan algoritma *firefly* sebagai pembangkit perilaku pencarian pada NPC (Non Player Character), dalam hal ini dilakukan 20 kali pengujian dengan persentase keberhasilan rata – rata 90.36% dimana persentase keberhasilan terendah adalah 75.71% dan persentase keberhasilan tertinggi adalah 98.57%.

5.2. Saran

Masih terdapat kekurangan dalam pembuatan *Game* ini dan hal tersebut disadari oleh penulis yang nantinya akan menjadi bahan pertimbangan untuk dikembangkan, diantaranya:

1. Kurang beragamnya karakter dalam *Game* baik NPC ataupun karakter *player*.
2. Visualisasi dalam *Game* perlu dikembangkan seperti tema *game* dan *map* lebih diperjelas, sehingga karakter *player*, NPC dan setiap *map* lebih seragam.
3. *Genre* dari *game* merupakan *Adventure Game* dengan visual yang 3D, diharapkan akan banyak *game* dengan *genre* tersebut yang membawa tema yang sama yaitu edukasi khususnya kepada budaya, seperti seni musik atau tari-tarian lokal Indonesia.

DAFTAR PUSTAKA

- Alif Ifa. 2015. *3D Wayang Adventure Game Untuk Pengenalan Budaya Wayang Nusantara Menggunakan A* Pathfinding Algorithm Sebagai Pembangkit Perilaku Pencarian Pada NPC*. Malang : UIN Maulana Malik Ibrahim Malang.
- Grace Lindsay. 2005. *Game Type and Game Genre* Washington D.C: American University
- Haryadi Fitra. 2012. *Metode Pathfinding pada map 2D Menggunakan Algoritma Diagonal & Bidirectional BFS*. Bandung : Institut Teknologi Bandung.
- Indah Turita S. 2008. *Ragam Wayang Di Nusantara*. Jakarta : Universitas Indonesia
- Husni Imam. 2009. *Artificial Intelligence dalam Proses Industri Manufaktur*. Semarang : Universitas Stikubank Semarang
- Jati G. K. Dan Suyanto. 2011. *Analisis Dan Implementasi Firefly Algorithm (Fa) Pada Masalah Travelling Salesman Problem (Tsp)*. Bandung : Universitas Telkom.
- Luqmanul Alvi. 2010. *Implementasi metode forward chaining untuk penentuan kasus karakter pada game petualangan*. Bandung : UNIKOM
- Pramono Andy. 2005. *Algoritma Pathfinding A* pada Game RPG Tanaman Higienis*. Malang : Universitas Negeri Malang
- Riyadi, Puanta Della Maharani. 2009. *Algoritma Pencarian A* dengan Fungsi Heuristik Jarak Manhattan*. Bandung : Intitut Teknologi Bandung.

- S. Deepthi Dan Ravikumar Aswanthy. 2015. *A Study from the Perspective of Nature-Inspired Metaheuristic Optimization Algorithms*. Thiruvananthapuram, India : Dept. of Computer Science and Engineering Mar Baselios College of Engineering and Technology.
- S. Haryanto. Pratiwimba Adhiluhung: Sejarah dan Perkembangan Wayang. Jakarta: Djambatan. 1988.
- Setiawan Hendry. dkk 2015. *Implementasi Algoritma Kunang-Kunang Untuk Penjadwalan Mata Kuliah di Universitas Ma Chung*. Malang : Universitas Ma Chung.
- Setiawan Willy. 2010. *Metode Pathfinding pada map 2D Menggunakan Algoritma Diagonal & Bidsirectional BFS*. Bandung : Institut Teknologi Bandung.
- Sri Mulyono. Wayang, Asal-usul, Filsafat dan Masa Depan. Jakarta: Gunung Agung. 1982.
- Wijanarko F. W. dkk 2011. *Pencarian Solusi Optimal Cutting Stock Problem Dengan Menggunakan Firefly Algorithm*. Bandung : Universitas Telkom.
- Yang X. S. 2010. *Nature-Inspired Metaheuristic Algorithms Second Edition*. United Kingdom : Luniver Press.
- Yang X. S. 2010. *Firefly Algorithms for Multimodal Optimization*. United Kingdom: University of Cambridge.
- Yang, X. S. 2010. *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley, Chichester.
- Yang X. S. dkk 2013. *A comprehensive review of Firefly algorithms*. Maribor, Slovenia : University of Maribor.

Yunanto, Andhik Ampuh. Herumurti Darlis, Kuswardaya Imam. 2017.
*Kecerdasan Buatan Pada Game Edukasi Untuk Pembelajaran Bahasa
Inggris Berbasis Pendekatan Heuristik Similaritas*. Surabaya : Institut
Teknologi Sepuluh Nopember

