

**IMPLEMENTASI ALGORITMA A* (A STAR) PADA GAME
VIRTUAL REALITY PENGENALAN TANAMAN
DI KEBUN RAYA PURWODADI**

SKRIPSI

Oleh:
DESTYAN PROBO SASONO
NIM. 13650093



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

2019

**IMPLEMENTASI ALGORITMA A* (A STAR) PADA GAME VIRTUAL
REALITY PENGENALAN TANAMAN DI KEBUN
RAYA PURWODADI**

SKRIPSI

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

**Oleh:
DESTYAN PROBO SASONO
NIM. 13650093**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2019**

LEMBAR PERSETUJUAN

IMPLEMENTASI ALGORITMA A* (A STAR) PADA GAME VIRTUAL
REALITY PENGENALAN TANAMAN DI KEBUN
RAYA PURWODADI

SKRIPSI


Oleh :
DESTYAN PROBO SASONO
NIM. 13650093

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal 26 Desember 2018

Dosen Pembimbing I

Dosen Pembimbing II


Dr. M. Faisal, M.T
NIP. 197405102005011007


Khadijah F.H. Holle, M.Kom
NIDT. 19900626 20160801 2 077

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




Dr. Cahyo Crysdiyan
NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN

**IMPLEMENTASI ALGORITMA A* (A STAR) PADA GAME VIRTUAL
REALITY PENGENALAN TANAMAN DI KEBUN
RAYA PURWODADI**

SKRIPSI

Oleh :
DESTYAN PROBO SASONO
NIM. 13650093

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk
Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 8 Januari 2019

Susunan Dewan Penguji

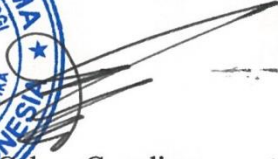
- | | |
|-----------------------|---|
| 1. Penguji Utama | : <u>Yunifa Miftachul Arif, M.T</u>
NIP. 19830616 201101 1 004 |
| 2. Ketua Penguji | : <u>Fressy Nugroho, M.T</u>
NIP. 19710722 201101 1 001 |
| 3. Sekertaris Penguji | : <u>Dr. Muhammad Faisal, M.T</u>
NIP. 19740510 200501 1 007 |
| 4. Anggota Penguji | : <u>Khadijah F.H.Holle, M.Kom</u>
NIDT. 19900626 20160801 2 077 |

Tanda Tangan

()
()
()
()

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




D. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini :

Nama : Destyan Probo Sasono
NIM : 13650093
Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : **IMPLEMENTASI ALGORITMA A* (A STAR)
PADA GAME VIRTUAL REALITY PENGENALAN
TANAMAN DI KEBUN RAYA PURWODADI**

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar – benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan 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, 26 Desember 2018
Yang membuat pernyataan,



Destyan Probo Sasono
NIM. 13650093

MOTTO

***“BERBAKTI LAH KEPADA
ORANG TUA”***



HALAMAN PERSEMBAHAN

Sujud syukur ku persembahkan pada ALLAH yang maha kuasa, berkat dan rahmat detak jantung, denyut nadi, nafas dan putaran roda kehidupan yang diberikan-Nya hingga saat ini saya dapat mempersembahkan skripsi ku pada orang-orang tersayang.

Orang tua ku merupakan orang terhebat di dunia, papa (Tri Yulianto) dan mama (Siti Utari) selalu memberikan semangat yang luar biasa. Ketika mulai berputus asa, muncul perasaan semangat membara saat kulihat perjuangan orang tuaku. Adik-adikku (Aldi dan Peppy) juga memberikan semangat ekstra agar kakak nya mencapai hasil yang maksimal dalam meraih kesuksesan dalam pengerjaan skripsi ini. Dan tak lupa semua keluarga besar yang lain seperti tante tini, om joko, keluarga kandang dan keluarga tuban.

Kepada dosen pembimbing Pak Faisal yang membantu dalam pengerjaan skripsi dan menuntun dengan sabar serta dipermudah dalam mengikuti sidang skripsi. Selain itu Ibu Khadijah yang juga membantu dalam penulisan skripsi agar benar. Dan juga Ibu Roro selaku dosen wali dari semester 1 hingga akhir ini selalu memberikan nasihat-nasihat baik.

Tak lupa semua teman-teman ku 1 Kontrakan 70an yang sering mengingatkan untuk mengerjakan skripsi terutama mas Ahmad Dzulfikri yang selalu membantu saat pengerjaan skripsi mulai dari mengingatkan untuk mengerjakan, dibantu pembetulan dalam penulisan serta hal-hal lain. Dan juga Ahmad Fathan Hunaifi yang sedia memberikan file-file sebagai acuan penulisan skripsi dan teman konsultasi bersama.

Kepada semua pihak yang tidak bisa saya sebutkan satu-persatu saya mengucapkan terima kasih yang sebesar-besarnya, semoga Allah SWT senantiasa memberikan rezekinya kepada kita semua. Amin

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamua'alaikum Wr. Wb.

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 **“IMPLEMENTASI ALGORITMA A* (A STAR) PADA GAME VIRTUAL REALITY PENGENALAN TANAMAN DI KEBUN RAYA PURWODADI”** ini ditulis untuk memenuhi salah satu syarat guna memperoleh gelar Sarjana Strata Satu (S1) Fakultas Sains dan Teknologi Universitas Islam Negeri 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. Abdul Haris, M.Ag., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Bapak Dr. Muhammad Faisal, M.T., selaku Dosen Pembimbing 1 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. Ibu Khadijah F.H. Hole, M.Kom., selaku Dosen Pembimbing 2 yang telah memberi masukan, serta pengarahan dalam penyusunan laporan skripsi ini.

4. Ibu Roro Inda Melani, M.T., M.Sc selaku dosen wali yang juga selalu memberi pengarahan terkait akademik selama masa studi.
5. Bapak Dr. Cahyo Crysdiyan selaku ketua jurusan Teknik Informatika yang mendukung dan mengarahkan skripsi ini.
6. Segenap civitas akademika Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang terutama seluruh dosen, terimakasih atas segala ilmu dan bimbingannya.
7. Kepada kedua orang tua tercinta yang selalu memberikan doa, kasih sayang, semangat, dukungan moril, serta motivasi sampai saat ini.

Malang, Desember 2018

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	Error! Bookmark not defined.
LEMBAR PENGESAHAN	iii
HALAMAN PERNYATAAN KEASLIAN TULISAN	iv
MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
ملخص.....	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	4
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	4
1.5 Batasan Penelitian	5
BAB II TINJAUAN PUSTAKA	6
2.1 Penelitian Terkait	6
2.2 Landasan Teori.....	9
2.2.1 A*(AStar)	9
2.2.2 <i>Game</i>	14
2.2.3 Tumbuhan	18
2.2.4 Kebun Raya Purwodadi.....	19
2.2.5 <i>Virtual Reality</i>	24
2.2.6 <i>Unity</i>	26
BAB III ANALISA DAN PERANCANGAN GAME	32
3.1 Analisis dan Perancangan <i>Game</i>	32
3.1.1 <i>Storyboard</i>	32

3.1.2 <i>Finite State Machine (FSM)</i>	34
3.1.3 Blok Diagram pada Game	39
3.1.3 Deskripsi Karakter dan Item	40
3.1.4 Deskripsi Objek Tanaman.....	41
3.1.5 Desain <i>Interface Game</i>	45
3.2. Perancangan Metode	48
3.2.1 Rancangan Metode <i>A* (AStar)</i>	48
3.2.2 Simulasi <i>A* (AStar)</i>	48
BAB IV HASIL DAN PEMBAHASAN	59
4.1 Implementasi	59
4.1.1 Implementasi Sistem	59
4.2 Implementasi Algoritma <i>A*</i> pada Perilaku <i>NPC</i>	60
4.2.1 <i>A*</i> Sebagai Algoritma Perilaku Pencarian	60
4.2.2 Script Komponen <i>Pathfinding A*</i>	61
4.3 Implementasi Aplikasi <i>Game</i>	66
4.3.1 Pengaturan Menu Awal.....	66
4.3.2 Pembentukan Kotak <i>Grid</i>	66
4.3.3 Pengaturan <i>NPC</i> berupa <i>Enemy</i> Musuh.....	67
4.3.4 Pengaturan Pemain.....	68
4.4 Uji Coba	69
4.4.1 Uji Coba Algoritma <i>A*</i>	69
4.4.2 Uji Coba <i>Game</i>	76
4.5 Integrasi Islam.....	80
BAB V PENUTUP.....	84
5.1 Kesimpulan	84
5.2 Saran.....	84
DAFTAR PUSTAKA	86

DAFTAR GAMBAR

Gambar 2.1 Contoh algoritma A*	9
Gambar 2.2 Tahap awal pencarian algoritma A*	11
Gambar 2.3 Ilustrasi <i>open list</i> dan <i>closed list</i>	12
Gambar 2.4 Perhitungan menentukan nilai F	13
Gambar 2.5 <i>Game</i> First Person Shooter.	15
Gambar 2.7 Pintu masuk Kebun Raya Purwodadi	21
Gambar 2.8 Tampilan Unity	27
Gambar 3.1 <i>Finite State Machine (FSM)</i> pada <i>menu game</i>	35
Gambar 3.2 <i>Finite State Machine (FSM)</i> pada <i>NPC level 1 game</i>	36
Gambar 3.3 <i>Finite State Machine (FSM)</i> pada <i>level 2 game</i>	36
Gambar 3.4 <i>Finite State Machine (FSM)</i> pada <i>level 3 game</i>	37
Gambar 3.5 <i>Finite State Machine (FSM)</i> pada <i>level 4 game</i>	38
Gambar 3.6 <i>Finite State Machine (FSM)</i> pada <i>level 5 game</i>	38
Gambar 3.7 Blok Diagram penempatan A* pada NPC	39
Gambar 3.8 Penampilan NPC pada <i>game</i> pengenalan tumbuhan <i>virtual reality</i>	40
Gambar 3.9 Tampilan dari kotak SEHAT	41
Gambar 3.10 Anggrek Hitam	42
Gambar 3.11 Bambu Petung	42
Gambar 3.12 <i>Selaginella plana</i>	43
Gambar 3.13 <i>Musa acuminata Colla var. Rutilifera</i>	44
Gambar 3.14 Palm	45
Gambar 3.15 Desain <i>Interface Icon</i>	45
Gambar 3.16 Desain <i>Interface Menu Utama</i>	46
Gambar 3.17 Desain <i>Interface Permainan</i>	46
Gambar 3.18 Desain <i>Interface menu Petunjuk</i>	47
Gambar 3.19 Desain <i>Interface Tentang</i>	47
Gambar 3.20 Tahap awal pencarian algoritma A*	49
Gambar 3.21 Ilustrasi <i>open list</i> dan <i>closed list</i>	50
Gambar 3.22 Perhitunag menentukan nilai F	51
Gambar 3.23 Penjelasan A* menggunakan <i>tree</i>	52

Gambar 3.24 Penjelasan A* pada h kedua	53
Gambar 3.25 Penjelasan A* menggunakan <i>tree</i> pada h kedua.....	54
Gambar 3.26 Penjelasan A* pada h ketiga	54
Gambar 3.27 Penjelasan A* dengan <i>tree</i> pada h ketiga	55
Gambar 3.28 Penjelasan A* pada h keempat.....	56
Gambar 3.29 Penjelasan A* dengan <i>tree</i> pada h keempat	56
Gambar 3.30 Penjelasan A* pada h kelima	57
Gambar 3.31 Penjelasan A* dengan <i>tree</i> pada h kelima	58
Gambar 3.32 Penjelasan A* pada h keenam	58
Gambar 4.1 Lokasi target dan lokasi <i>NPC</i>	64
Gambar 4.2 Menampilkan <i>grid</i> pada panel.....	64
Gambar 4.3 Meletakkan script A* pada <i>GameObject</i> yang sama dengan <i>grid</i>	65
Gambar 4.4 Tampilan jalur path yang akan dilalui <i>NPC</i> menuju Target	65
Gambar 4.5 Membuat menu awal <i>game</i>	66
Gambar 4.6 Membentuk <i>grid</i> dan memasukan script A*	66
Gambar 4.7 Membuat <i>NPC</i> dan memasukan A*.....	67
Gambar 4.8 Pemain yang berupa <i>First Person Shooter</i>	68
Gambar 4.9 Memasukan komponen pendukung <i>Virtual Reality</i>	68
Gambar 4.10 Letak percobaan kedua <i>NPC</i> dan Target.....	72
Gambar 4.11 Letak percobaan kedua <i>NPC</i> dan Target.....	75
Gambar 4.12 Tampilan Menu.....	76
Gambar 4.13 Tampilan pemain menembak <i>NPC</i>	76
Gambar 4.13 Tampilan <i>NPC</i> menyerang tanaman	77
Gambar 4.14 Tampilan informasi mengenai tumbuhan terkait	77

DAFTAR TABEL

Tabel 3.1 <i>Storyboard</i> dari permainan.....	33
Tabel 3.2 Perhitungan h pertama.....	52
Tabel 3.3 Perhitungan h kedua.....	53
Tabel 3.4 Perhitungan h ketiga.....	55
Tabel 3.5 Perhitungan h keempat.....	56
Tabel 3.6 Perhitungan h kelima.....	57
Tabel 4.1 Tabel komponen A* pada Script Pathfinding.....	61
Tabel 4.2 Hasil pengujian A* pertama.....	69
Tabel 4.3 Hasil pengujian A* kedua.....	72
Tabel 4.4 Tabel percobaan pada <i>device smartphone</i>	77
Tabel 4.5 Persentase Uji Coba <i>Game</i>	79



ABSTRAK

Sasono, Destyan Probo. 2018 **Implementasi Algoritma A* (A STAR) Pada Game Virtual Reality Pengenalan Tanaman Di Kebun Raya Purwodadi**. Skripsi, Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) M.Faisal, M.T , (II) Khadijah F.H. Holle, M.Kom

Kata Kunci : *Algoritma A**, *Game*, *Virtual Reality*, *NPC*

Tanaman merupakan salah satu makhluk hidup yang ada di bumi, Negara Indonesia merupakan salah satu negara yang kaya akan keanekaragaman flora. Banyak tempat digunakan untuk melakukan penelitian tentang tanaman salah satunya adalah di Kebun Raya Purwodadi di Kabupaten Pasuruan. Banyak sekali tanaman yang tumbuh disana, untuk mengenal lebih dekat tentang tanaman tersebut, perlu dibuat sebuah *game virtual* untuk menambah informasi melalui hiburan. Untuk itu penulis tergerak untuk membuat sebuah *game* bertema *First Person Shooter* untuk mengenalkan tanaman yang ada di Kebun Raya Purwodadi.

Pada *game* ini akan ada *NPC* berwujud *NPC* yang akan merusak dan menghancurkan tanaman tersebut. Untuk menuju tanaman, *NPC* perlu diterapkan kecerdasan. Penulis menggunakan algoritma *A** untuk diterapkan sebagai kecerdasan pada *NPC* untuk menemukan jalur terpendek menuju tanaman. *Game* ini juga menerapkan teknologi *virtual reality* sebagai media pengenalan tanaman yang interaktif dan menarik.

ABSTRACT

Sasono, Destyan Probo. 2018 **Implementation of A*(A Star) Algorithm in Virtual Reality Game to Introduction of Plants at Purwodadi Botanical Garden.** Undergraduate Thesis, Informatics Engineering Department of Science and Technology Faculty Islamic State University Maulana Malik Ibrahim Malang. Supervisor : (I) M.Faisal, M.T , (II) Khadijah F.H. Holle, M.Kom

Kata Kunci : *Algoritma A**, *Game*, *Virtual Reality*, *NPC*

Plants are one of the living things that exist on earth, Indonesia is a country rich in flora diversity. Many places are used to conduct research on plants, one of which is at the Purwodadi Botanical Garden in Pasuruan Regency. There are so many plants that grow there, to get to know more about these plants, it needs to be made a virtual game to add information through entertainment. For this reason, the author was moved to make a game with the theme First Person Shooter to introduce plants in the Purwodadi Botanical Gardens.

In this game there will be NPCs in the form of NPCs that will damage and destroy the plants. To get to plants, NPC needs to be applied to intelligence. The author uses the A * algorithm to be applied as intelligence to the NPC to find the shortest path to the plant. This game also applies virtual reality technology as an interactive and interesting medium for introducing plants.

ملخص

ساسونو، ديستيان فربو. 2018. تنفيذ خوارزمية (A Star) *A في لعبة الواقع الافتراضي لإدخال النباتات في حديقة Purwodadi النباتية. أطروحة الجامعية. قسم هندسة المعلوماتية لكلية العلوم والتكنولوجيا في جامعة الدولة الإسلامية مولانا مالك إبراهيم مالانج. المشرف: (الأحد) الدكتور فيسال، ماجستير في التكنولوجيا، (الإثنان) خادجة فهمي حياتي هولي

الكلمات الدالة: . خوارزمية *A ، لعبة ، الواقع الافتراضي ، NPC

النباتات هي واحدة من الكائنات الحية الموجودة على الأرض ، وإندونيسيا بلد غني بتنوع النباتات. تستخدم العديد من الأماكن لإجراء البحوث على النباتات ، واحدة منها في حديقة Purwodadi النباتية في باسوروان ريجنسي. هناك العديد من النباتات التي لديها لعبة افتراضية لإضافة المعلومات من خلال وسائل الترفيه. لهذا السبب ، تم نقل المؤلف لجعل لعبة مع موضوع أول شخص مطلق النار لإدخال النباتات في الحدائق النباتية Purwodadi .

في هذه اللعبة سيكون هناك NPCs في شكل NPCs من شأنها أن تضر وتدمر النباتات . للوصول إلى النباتات ، يحتاج NPC إلى تطبيقه على الذكاء. يستخدم المؤلف خوارزمية A * ليتم تطبيقها كمعلومات استخباراتية للمجلس الوطني لنواب الشعب لإيجاد الطريق الأقصر إلى المصنع . كما تطبق هذه اللعبة تكنولوجيا الواقع الافتراضي كوسط تفاعلي ومفيد لإدخال النباتات.

BAB I

PENDAHULUAN

Pada bab ini akan dijelaskan mengenai latar belakang penelitian, identifikasi masalah, tujuan penelitian, manfaat penelitian dan batasan penelitian. Latar belakang penelitian berisi penjelasan mengenai alasan peneliti mengangkat dan melakukan penelitian ini. Identifikasi masalah berisi sebuah pertanyaan yang didasarkan pada alasan dilakukan penelitian ini dan Tujuan Penelitian berisi tujuan dilakukannya penelitian ini. Batasan Penelitian berisi batasan pada penelitian agar penelitian tidak meluas dari fokus dilakukannya penelitian ini.

1.1 Latar Belakang

Negara Indonesia merupakan salah satu negara yang kaya akan sumber daya alam khususnya keanekaragaman flora. Banyaknya keanekaragaman flora di Indonesia dikarenakan Indonesia merupakan daerah tropis yang memungkinkan banyak tumbuh-tumbuhan dapat berkembang dengan baik. Diperkirakan di seluruh dunia terdapat 2 jutaan spesies tumbuhan yang telah dikenali dan 60 % dari jumlah tersebut terdapat di Indonesia. Menurut Mustaid Siregar, Kepala Pusat Konservasi Tumbuhan LIPI, meskipun Indonesia kaya akan keragaman flora, namun saat ini baru ada 8000 jenis yang sudah teridentifikasi, jumlah tersebut diperkirakan baru 20% dari jumlah yang ada di Indonesia (Utomo, 2011).

Observasi merupakan kegiatan penting yang dilakukan oleh saintis untuk memperoleh data dan gambaran lapangan. Observasi ada dua macam yakni observasi partisipan dan non partisipan. Observasi partisipan yakni peneliti terlibat dalam objek pengamatan, sedangkan observasi *non* partisipan merupakan

peneliti tidak terlibat secara langsung, hanya sekedar pengamatan dan pencatatan pengamatan. Salah satu kegiatan observasi *non* partisipan adalah kegiatan pengamatan tanaman–tanaman yang terdapat di Kebun Raya Purwodadi–Pasuruan, Jawa Timur. Observasi ini dilakukan dalam rangka berproses dalam membentuk jiwa saintis, peduli terhadap tanaman, lebih memperkaya memori dalam menghafal jenis – jenis tanaman dan refreasing otak.

Kebun Raya Purwodadi merupakan Kebun botani (atau taman botani) atau suatu lahan yang ditanami berbagai jenis tumbuhan yang ditujukan untuk keperluan koleksi, penelitian, dan konservasi *exsitu* (di luar habitat). Selain untuk penelitian, kebun botani dapat berfungsi sebagai sarana wisata dan pendidikan bagi pengunjung. *Arboretum* adalah semacam kebun botani yang mengkoleksi pepohonan. Sehingga sangat cocok dalam menambah wawasan keilmuan kita dalam keanekaragaman tanaman (<http://www.krpurwodadi.lipi.go.id/>).

Menurut penulis, sejak kecil anak-anak harus diingatkan akan pentingnya tumbuhan, mengenalkan mereka dengan tumbuhan merupakan hal paling dasar untuk mencintai tumbuhan dan akan muncul rasa untuk menjaga dan melestarikan tumbuhan. Banyak cara mengenalkan tumbuhan kepada anak-anak mulai dari yang diajarkan di sekolah formal hingga melalui permainan, permainan dinilai salah satu cara yang menyenangkan dalam mengenalkan tumbuhan kepada anak-anak. *Game* sangat digemari oleh banyak orang terutama anak-anak, saat ini banyak kita jumpai anak-anak lebih senang bermain *game* pada *smartphone* daripada bermain di lapangan. Selain digunakan untuk kepuasan bermain, *game* harus dimanfaatkan dalam hal positif salah satunya sebagai media edukatif untuk mengenalkan tumbuhan kepada anak-anak sehingga anak-anak memiliki

keinginan untuk menjaga kelestarian alam khususnya tumbuhan. Pentingnya edukasi juga merupakan anjuran dalam Agama Islam sesuai firman Allah SWT dalam Al-Quran dalam surat Thaha ayat 114 :

فَتَعَالَى اللَّهُ الْمَلِكُ الْحَقُّ وَلَا تَعْجَلْ بِالْقُرْآنِ مِنْ قَبْلِ أَنْ يُقْضَى إِلَيْكَ وَحْيُهُ وَقُلْ رَبِّ زِدْنِي عِلْمًا

Artinya “Maka Maha Tinggi Allah Raja Yang sebenar-benarnya, dan janganlah kamu tergesa-gesa membaca Al qur'an sebelum disempurnakan mewahyukannya kepadamu, dan katakanlah: "Ya Tuhanku, tambahkanlah kepadaku ilmu pengetahuan”.

Game sudah berkembang sangat pesat, salah satu perkembangan game yang paling pesat adalah dengan memanfaatkan teknologi *virtual reality*. *Virtual reality* merupakan salah satu sarana hiburan yang dimiliki oleh sistem *android*, *Virtual reality* atau biasa disebut realitas maya adalah sebuah teknologi yang dibuat agar pengguna dapat melakukan interaksi dengan lingkungan yang disimulasikan oleh komputer dan dengan bantuan alat ketiga yang berfungsi untuk memfokuskan pandangan dengan lingkungan yang telah disimulasikan tadi. Konsep dasar *virtual reality* adalah mencoba membuat suatu dunia didalam komputer, kebanyakan lingkungan utamanya merupakan pengalaman visual, ditampilkan pada layar melalui kaca mata stereoskopik khusus, tetapi beberapa simulasi termasuk sensor informasi tambahan.

Dengan memanfaatkan teknologi *virtual reality*, penulis tergerak untuk membangun sebuah game yang bertujuan untuk mengenalkan tumbuhan-tumbuhan khususnya yang dimiliki oleh Kebun Raya Purwodadi. Game ini menceritakan ada *monster* yang serakah dimana mereka merusak tumbuhan hanya untuk kesenangan mereka saja. Peran kita sebagai pemain adalah mencegah

monster itu untuk menghancurkan tumbuhan tersebut. Untuk menuju ke tumbuhan, *NPC* memerlukan algoritma yang akan membangkitkan perilakunya untuk mencari jalur terpendek menuju targetnya. Dari beberapa algoritma *pathfinding* yang digunakan untuk mencari rute terpendek, algoritma A^* (*A-Star*) merupakan salah satu metode *pathfinding* yang saya mengerti dan pahami.

Dalam penelitian ini, penulis berharap dengan seiring berkembangnya era modern ini serta majunya teknologi, anak-anak tidak lupa kepada alam dan semakin peduli akan pentingnya tumbuhan serta mampu memanfaatkan teknologi untuk menambah wawasan keilmuan mereka.

1.2 Identifikasi Masalah

1. Bagaimana membuat *game* pengenalan koleksi tumbuhan Kebun Raya Purwodadi dengan teknologi *virtual reality*?
2. Bagaimana menerapkan algoritma A^* untuk mencari jalur terpendek sebagai perilaku *NPC* pada *game* pengenalan tumbuhan di Kebun Raya Purwodadi dengan teknologi *virtual reality*?

1.3 Tujuan Penelitian

1. Untuk membuat *game* pengenalan koleksi tumbuhan Kebun Raya Purwodadi dengan teknologi *virtual reality*.
2. mengimplementasikan algoritma A^* untuk mencari jalur terpendek sebagai perilaku *NPC* pada *game* tersebut.

1.4 Manfaat Penelitian

Manfaat yang didapat dari penelitian ini dapat dipandang dari tiga aspek yaitu peneliti, keilmuan dan manfaat dari sisi pembaca. Adapun manfaat tersebut adalah sebagai berikut :

1. Mengetahui cara kerja algoritma A^* untuk menemukan jalur terpendek pada *NPC* untuk menemukan *target*.
2. Mengembangkan *game virtual reality* untuk permainan edukasi yang lebih menarik.
3. Mengenalkan tanaman-tanaman khususnya yang berada di Indonesia yang ditanam di kebun raya purwodadi.

1.5 Batasan Penelitian

Agar pembahasan penelitian ini tidak menyimpang, maka diperlukan batasan-batasan. Batasan-batasan dalam penelitian ini adalah

1. *Game* ini berupa single player.
2. Objek tanaman yakni 5 jenis tumbuhan .
3. Objek tanaman berada di wilayah kebun raya purwodadi.
4. Dimainkan oleh anak-anak dengan usia 5 tahun hingga 15 tahun.

BAB II

TINJAUAN PUSTAKA

Pada bagian ini membahas tentang penelitian yang terkait, referensi-referensi penelitian dan konsep tentang teori yang digunakan dalam melakukan penelitian ini.

2.1 Penelitian Terkait

Penelitian-penelitian yang telah dilakukan sebelumnya menjadi bahan referensi dalam membangun sebuah aplikasi atau media. Terutama penelitian-penelitian yang berhubungan dengan *virtual reality*.

1. Implementasi Pathfinding Algoritma A* Pada Game Petualangan Labirin Rahasia

Penelitian yang berjudul *Implementasi Pathfinding Algoritma A* Pada Game Petualangan Labirin Rahasia* yang disusun oleh Rozalia Deviana bahwa dengan penggunaan algoritma *AStar* dapat diimplementasikan pada *game* labirin yang diimplementasikan ke *NPC* untuk menemukan player, namun dalam penelitian ini tidak menggunakan teknologi *virtual reality* yang mana akan membuat *game* ini terlihat biasa saja (Deviana, 2015).

Penulis akan mengambil ilmu yang terkait tentang metode *A** pada penelitian ini dan akan mengembangkan dengan menggunakan teknologi *virtual reality*.

2. Penerapan Algoritma A* Pada Game Edukasi The Maze Island Berbasis Android

Kemudian Aplikasi *game* yang dibuat pada penelitian ini berhasil menerapkan algoritma *A**, algoritma tersebut telah berjalan dengan baik karena

musuh dapat mendeteksi posisi pemain. Kemudian musuh dapat mendeteksi posisi pemain. Kemudian musuh juga dapat mengejar pemain dengan memilih rute terpendek melalui nilai heuristik terkecil dari setiap simpul yang dipilih (Agung Pamungkas, 2015).

Penulis akan mengambil ilmu yang terkait tentang metode A* pada penelitian ini dan akan mengembangkan dengan menggunakan teknologi *virtual reality*.

3. Implementasi Pathfinding dengan Algoritma A* pada Game Funny English Menggunakan 3D Berbasis Graf

Game ini bertujuan untuk mempermudah pengenalan kosakata bahasa Inggris. Algoritma A* diimplementasikan pada NPC untuk mencari rute menuju ke suatu tempat. Dari seluruh pengujian yang telah dilakukan, dapat diambil kesimpulan bahwa seluruh rute yang diambil oleh NPC telah sesuai dengan rute optimal yang harus ditempuh secara logika manusia (Riwinoto & Alfian, 2015).

Penulis akan mengambil ilmu yang terkait tentang metode A* pada penelitian ini dan akan mengembangkan dengan menggunakan teknologi *virtual reality*.

4. Penerapan Algoritma A* pada Aplikasi Puzzle

Lalu penelitian yang berkaitan mengenai algoritma A* yakni *Penerapan Algoritma A* pada Aplikasi Puzzle* yang ditulis oleh Iatus Hermawan dan rekannya R. Kristoforus Jawa Bendi. Mengemukakan bahwa untuk menyelesaikan *puzzle* 3x3 dapat digunakan dengan menggunakan algoritma A*. Bahkan dengan menggunakan algoritma A* lebih efektif daripada menggunakan metode *pathfinding* yang lainnya (Hermawan & Jawa Bendi, 2013).

Penulis akan mengambil ilmu yang terkait tentang metode A* pada penelitian ini dan akan mengembangkan dengan menggunakan teknologi *virtual reality*.

5. Pembuatan Kebun Binatang Virtual Reality Untuk Pembelajaran Mengenai Binatang Liar

Teknologi *virtual reality* pada aplikasi ini mampu berjalan dengan baik pada perangkat *android* dengan adanya penurunan kualitas grafik apabila dibandingkan pada saat aplikasi berjalan pada *dekstop* (Gunawan, Liliana, & Budhi, 2016).

Penulis akan mengambil ilmu tentang manfaat teknologi *virtual reality* sebagai sarana informasi pada *game* yang akan dibangun.

6. Pengembangan Museum Virtual Interaktif Menggunakan Teknologi Desktop Virtual Reality Pada Museum Ranggawarsita

Pada Penelitian ini, data yang digunakan sangat valid dengan persentasi 94,853%, dan menurut pengelola museum dan pengunjung museum dinyatakan sangat layak (Vidiardi, 2015).

Penulis akan mengambil ilmu tentang manfaat teknologi *virtual reality* sebagai sarana informasi pada *game* yang akan dibangun.

7. Faktor-Faktor Yang Mempengaruhi Pencapaian Target Penerimaan Negara Bukan Pajak (PNBP) Pada UPT Balai Konservasi Tumbuhan Kebun Raya Purwodadi

Pada penelitian ini, peneliti meneliti kebun raya purwodadi dengan keuangan, hasil penelitian menunjukkan bahwa PNPB tercapai sesuai target di Kebun Raya Purwodadi (Al'amin, Faktor-Faktor Yang Mempengaruhi Pencapaian Target Penerimaan Negara Bukan Pajak (PNBP) Pada UPT Balai Konservasi Tumbuhan Kebun Raya Purwodadi, 2015).

Penulis mengambil data-data yang diperlukan yang berkaitan dengan Kebun Raya Purwodadi.

2.2 Landasan Teori

Pada sub bab ini dijelaskan mengenai teori-teori yang berkaitan dengan penelitian ini.

2.2.1 A*(AStar)

Dalam buku yang berjudul *A* Pathfinding for Beginner*, Patrick Lester menceritakan sejarah dan pengertian tentang algoritma A*. Algoritma ini pertama kali diperkenalkan pada 1968 oleh Peter Hart, Nils Nilsson, dan Bertram Raphael. Dalam ilmu computer, A* (yang diucapkan dengan "A Star") merupakan salah satu algoritma pencarian *graph* terbaik yang mampu menemukan jalur dengan biaya pengeluaran paling sedikit dari titik permulaan yang diberikan sampai ke titik tujuan yang diharapkan (dari satu atau lebih mungkin tujuan) (Hart, Raphael, & Nilsson, 1972).



n	S	A	B	C	D	E	F	G	H	J	K	L	M
h(n)	80	80	60	70	85	74	70	0	40	10	30	20	70

Gambar 2.1 Contoh algoritma A*

Pada gambar 2.1, masalah pencarian rute dalam suatu daerah yang direpresentasikan dalam suatu graph 2 arah. Setiap simpul menyatakan suatu kota. Busur menyatakan jarak sebenarnya antara satu kota dengan kota lainnya. Dan $h(n)$ menyatakan biaya perkiraan dari simpul n menuju G . Kemudian mencari rute terpendek dari S menuju G .

Harga (F) adalah nilai yang diperbolehkan dari penjumlahan nilai G , jumlah nilai tiap simpul dalam jalur terpendek dari *starting point* ke *currentnode*, dan H , jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan. Simpul tujuan adalah simpul yang dituju. Rintangan adalah sebuah atribut yang menyatakan bahwa sebuah simpul tidak dapat dilalui oleh A .

AStar memiliki 2 fungsi utama dalam menentukan solusi terbaik. Fungsi pertama disebut sebagai $g(n)$ merupakan fungsi yang digunakan untuk menghitung total *cost* yang dibutuhkan dari *starting point* menuju *node* tertentu. Fungsi kedua yang biasa disebut sebagai $h(n)$ merupakan fungsi perkiraan total *cost* yang diperkirakan dari suatu *node* ke *node* akhir.

Pada *AStar*, setiap *node* dari *node* awal ditelusuri kemudian dihitung *cost* dari tiap-tiap *node* dan dimasukkan ke tabel prioritas. *Node* dengan *cost* paling rendah akan diberikan tingkat prioritas paling tinggi. Kemudian pencarian dilanjutkan pada *node* dengan nilai prioritas tertinggi pada tabel.

$$F(n) = G(n) + H(n) \quad (2.1)$$

Dengan :

n = posisi koordinat *node*

$F(n)$ = fungsi evaluasi

$G(n)$ = biaya (*cost*) yang sudah dikeluarkan dari keadaan sampai keadaan n

$H(n)$ = estimasi biaya untuk sampai pada suatu tujuan mulai dari n .

Untuk perhitungan nilai H digunakan fungsi heuristic, metode yang digunakan di dalam contoh ini adalah metode Manhattan dimana perhitungan jumlah *node* hanya yang bergerak secara *vertical* dan *horizontal* menuju tujuannya serta mengabaikan penghalang, yang kemudian nilainya dikalikan dengan 10. Atau dirumuskan dengan:

$$H = 10 * (abs(currentX - targetX) + abs(currentY - targetY)) \quad (2.2)$$

Node dengan nilai terendah merupakan solusi terbaik untuk diperiksa pertama kali pada $G(n) + H(n)$. Dengan fungsi *heuristic* yang memenuhi kondisi tersebut, maka pencarian dengan algoritma *AStar* dapat optimal.

A				
			B	

Gambar 2.2 Tahap awal pencarian algoritma A*

Pencarian wilayah dibagi ke dalam *node-node* yang terdiri dari *node* awal (*node* A) dan *node* akhir (*node* B). Selanjutnya, melakukan pencarian untuk menemukan jalan terpendek dengan cara memeriksa *node-node* yang berdekatan dengan *node* awal untuk bisa sampa ke *node* B.

Berikut adalah h pencarian yang dilakukan oleh *node* awal:

1. h pertama, dimulai dari *node* awal yang ditambahkan ke dalam *Open List* yang merupakan *list* untuk *node-node* yang harus diperiksa.
2. Periksa semua *node* yang berdekatan dengan *node* awal apakah bisa dilalui atau tidak, *node* penghalang yang berwarna biru diabaikan lalu tambahkan

semua *node* tersebut ke dalam *Open List*, untuk setiap *node* yang baru ditambahkan, simpan *node* awal sebagai “*parent*” yang nantinya akan digunakan untuk menelusuri jalan.

3. Hapus *node* awal dari *Open List*, dan tambahkan ke dalam *closed list*. *Close list* merupakan *list* untuk *node* yang perlu diperiksa atau dalam artian merupakan bagian dari jalur terpendek yang sudah didapatkan.

<i>Close list</i> A	<i>Open list</i>			
<i>Open list</i>	<i>Open list</i>		B	

Gambar 2.3 Ilustrasi *open list* dan *closed list*

h selanjutnya adalah memilih *node* yang memiliki nilai F terendah di dalam *open list* dengan menggunakan rumus ($F=G+H$), G adalah nilai yang dikeluarkan untuk *node* A ke *node* yang ada disekitarnya. Sedangkan H adalah fungsi *heuristic* yang digunakan untuk menghitung perkiraan biaya yang dikeluarkan dari *node* awal menuju *node* tujuan.

0	8	1	6		
	1		2	3	4
8	NPC	8			5
2	6	3	4	8	9
	6		7		10
8		7		Player	
11		12	13	14	15

16	17	18	19	20
----	----	----	----	----

Gambar 2.4 Perhitungan menentukan nilai F

Proses ini dilakukan secara berulang-ulang, dimulai dari *open list* dan menemukan *node* dengan nilai F paling rendah. Perhitungan persamaan untuk mendapatkan nilai F dijelaskan pada gambar dibawah.

Perhitungan nilai H yaitu nilai estimasi jalur terpendek dari *node* awal ke *node* tujuan atau target dirumuskan dengan:

$$H = 2 * (abs(n.x - tujuan.x) + abs(n.y - tujuan.y)) \quad (2.3)$$

Sebagai contoh *node* yang terletak disebelah kanan dari *node* awal diberikan nilai H=6 yang didapatkan dari perhitungan 6 *node* menuju tujuan dan dikalikan 2, $2*2=6$

Setelah didapatkan nilai F dari setiap *node* yang ada di *open list*, maka akan dipilih *node*, h selanjutnya adalah sebagai berikut:

1. Hapus *current node* dari *open list* dan masukan ke dalam *closed list*.
2. Periksa semua *node* yang berdekatan dengan *current node*, abaikan *node* penghalang. Jika tidak ada *open list*, tambahkan kedalam *open list*, lalu tambahkan “parent” untuk *node* tersebut.

3. Jika *node* suda ada pada *open list*, lalu bandingkan dengan *node* awal apakah jalur yang di ambil lebih baik dari *node* awal, dengan memeriksa nilai G-nya apakah lebih rendah jika kita menggunakan *node* ini, jika lebih rendah maka hitung ulang nilai F dan G serta merubah arah *pointer*, jika tidak maka jangan lakukan apa-apa (Wafiqurahman, 2015).

Dalam penerapannya, Algoritma A* memiliki beberapa terminologi dasar diantaranya *startnode*, *targetnode*, simpul (*nodes*), *open list*, *closed list*, harga (*cost*), halangan (*unwalkable*). Berikut ringkasannya:

1. *Startnode* adalah sebuah terminologi untuk posisi awal sebuah benda.
2. *Targetnode* adalah posisi akhir yang dituju oleh NPC.
3. *Currentnode* adalah posisi sekarang yang dilalui oleh NPC
4. Simpul adalah petak-petak kecil sebagai representasi dari area pathfinding. Bentuknya dapat berupa persegi, lingkaran, maupun segitiga.
5. *Open list* adalah tempat menyimpan data simpul yang mungkin diakses dari starting point maupun simpul yang sedang dijalankan.
6. *Closed list* adalah tempat menyimpan data simpul sebelum *currentnode* yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan.
7. Harga adalah nilai yang diperoleh dari penjumlahan, jumlah nilai tiap simpul dalam jalur terpendek dari *startnode* ke *currentnode*, dan jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan.
8. Simpul tujuan yaitu simpul yang dituju.
9. Halangan adalah sebuah atribut yang menyatakan bahwa sebuah simpul tidak dapat dilalui oleh *currentnode*.

2.2.2 Game

Agustinus Nilwan (1998) dalam bukunya yang berjudul “*Pemrograman Animasi dan Game Profesional 4*” mengatakan bahwa *game* sebagai permainan komputer yang dibuat dengan teknik dan metode animasi. Jika ingin mendalami

penggunaan animasi haruslah memahami pembuatan *game*, sebaliknya juga berlaku demikian sebab keduanya saling berkaitan (Nalwan, 1998).



Gambar 2.5 *Game* First Person Shooter.

Game merupakan salah satu hiburan yang dapat digunakan sebagai penyegar pikiran saat pengguna sedang penat. Tidak hanya itu jika *game* digunakan dengan baik maka hal itu dapat merangsang kemampuan berfikir dan kecerdasan berfikir pada anak. Hal itu disebabkan karena didalam *game* itu sendiri terdapat masalah-masalah yang harus diselesaikan oleh pemain yang dimana dibutuhkan kemampuan berfikir yg teliti dan kreatif.

Game sendiri menurut John C Beck dan Mitchell Wade yang dikutip dalam buku Pengaruh Bermain *Game* Terhadap Perkembangan Remaja adalah penarik perhatian yang telah terbukti. *Game* merupakan lingkungan pelatihan yang baik bagi dunia nyata dalam organisasi yang menurut pemecahan masalah secara kolaborasi (Beck & Mitchell, 2016).

2.2.2.1 Jenis jenis game

Game dikategorikan dalam beberapa jenis dalam buku yang berjudul *Andrew Rollings and Ernest Adam on Game Design*, yaitu sebagai berikut:

1. *Fun Games*

Fun games adalah *game* seperti : *Skateboard, Puzzle, Tetris, Bilyard, Catur, Puzzle, Golf, Windows Entertainment Pack Games* dan semua semua *game* yang memiliki sedikit animasi. *Game* seperti ini jika dilihat dari grafiknya terlihat mudah namun dari segi algoritma permainannya biasanya lebih susah. *Game* ini sangat baik untuk melatih otak karena permainannya yang sederhana tapi membutuhkan cara berfikir yang ekstra.

2. *Arcade Games*

Berbeda dengan *fun games*, *arcade games* merupakan *games* yang lebih mudah dimengerti dan memiliki grafik yang lebih bagus jika dibandingkan dengan *fun games*. *Game* arcade ini lebih mudah dimengerti karena merupakan *game* yang mengandung unsur yang disenangi, seperti menembak, memukul, menendak, menusuk, balapan.

3. *Strategic Games*

Sesuai dengan namanya, *game* strategi merupakan permainan yang membutuhkan strategi untuk memenangkannya. *Game* strategi ini banyak terdapat pada *game – game* seperti *game* perang.

4. *Adventure Games*

Game *adventure* sendiri terbagi menjadi beberapa yaitu petualangan biasa (*Multi Layered Adventure*), *Dungeon-Underworld Adventure* (*3D Adventure*),

dan *Roll Playing Game Adventure*. Untuk segi algoritma *game* ini sedang-sedang saja sampai dengan sulit. Namun grafik dari *game* jenis ini dapat dikatakan susah.

5. *Simulation Games*

Dari semua jenis *game* yang ada, masing-masing memiliki tingkat kesulitan dan kemudahannya masing-masing. Jika algoritma permainannya susah maka dalam animasinya akan lebih mudah. Namun *game* simulasi ini merupakan *game* yang paling susah untuk dimainkan, karena algoritma maupun animasi dari *game* ini tergolong susah. Pemain harus dapat memperhitungkan semua kejadian yang akan terjadi dalam kondisi sebenarnya.

6. *First Person Shooter(FPS)*

Genre permainan video yang ciri utamanya adalah penggunaan sudut pandang orang pertama dengan tampilan layar yang mensimulasikan apa yang dilihat melalui mata karakter yang dimainkan. Ciri utama lain adalah penggunaan senjata genggam jarak jauh (Rollings & Ernest, 2003).

Dari semua jenis *game* diatas, jenis *game FPS* yang paling cocok digunakan dalam *game* yang akan penulis bangun karena *virtual reality* sendiri merupakan teknologi *first person*.

2.2.2.2 *Non Player Character (NPC)*

Salah satu unsur terpenting dalam *game* adalah *NPC*, menurut Yunifa Miftahul Arif (2010) dalam penelitiannya yang berjudul *Strategi Menyerang pada Game FPS Menggunakan Hierarchy Finite State Machine dan Logika Fuzzy* menjelaskan bahwa *NPC* adalah Jenis *Otonomous agent* yang ditujukan untuk penggunaan komputer animasi dan media interaktif seperti *games* dan *virtual reality*. Agen ini mewakili tokoh dalam cerita atau permainan dan memiliki

kemampuan untuk improvisasi tindakan mereka. Ini adalah kebalikan dari seorang tokoh dalam sebuah film animasi yang tindakannya ditulis di muka dan untuk “*avatar*” dalam sebuah permainan atau *virtual reality*, tindakan yang diarahkan secara real time oleh pemain.

Secara garis besar maka *NPC* dapat diartikan sebagai karakter pada *game* yang sepenuhnya dikendalikan komputer dan tidak dapat dimainkan oleh pemain. Pengendalian *NPC* umumnya menggunakan bidang ilmu kecerdasan buatan yang dimasukan dengan tujuan agar *NPC* dapat melakukan pekerjaan seperti apa yang dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer, logika fuzzy, jaringan syaraf tiruan dan robotika (Arif, 2010).

2.2.3 Tumbuhan

Cavalier-Smith Dalam biologi, tumbuhan adalah organisme *eukariota multiseluler* yang tergolong ke dalam kerajaan *Plantae*. Di dalamnya terdiri atas beberapa klad yakni, tanaman berbunga, *Gymnospermae* atau Tumbuhan berbiji terbuka, *Lycopodiopsida*, paku-pakuan, lumut, serta sejumlah alga hijau (Cavalier-Smith, 1981).

Tanaman hijau memiliki dinding sel yang kokoh mengandung *selulosa*. Hampir semua anggota tumbuhan bersifat *autotrof*, yakni memproduksi energi sendiri dengan mengubah energi cahaya matahari melalui proses yang disebut fotosintesis dalam organel sel bernama *kloroplas*. Karena warna hijau yang dominan pada anggota kerajaan ini, nama lain yang dipakai adalah *Viridiplantae* ("tumbuhan hijau"). Nama lainnya adalah *Metaphyta*. Namun ada juga tumbuhan yang bersifat parasit dan beberapa sudah tidak memiliki kemampuan

fotosintesis dengan sedikit atau bahkan tanpa *klorofil*. Tanaman juga bisa dikarakterisasi dari cara mereka berkembang biak, kemampuan pertumbuhan, dan pergiliran keturunan.

Tercatat sekitar 350.000 spesies organisme termasuk di dalamnya, tidak termasuk alga hijau. Dari jumlah itu, 258.650 jenis merupakan tumbuhan berbunga dan 18.000 jenis tumbuhan lumut. Tumbuhan hijau menghasilkan hampir seluruh molekul oksigen di muka bumi ini dan merupakan bagian terpenting dalam sistem ekologi bumi. Tumbuhan-tumbuhan yang sudah di domestikasi bisa menghasilkan biji, buah-buahan dan sayuran yang berguna sebagai bahan dasar pangan manusia. Selain itu tumbuhan juga digunakan sebagai tanaman hiasan dan banyak yang berkhasiat obat serta digunakan dalam ilmu medis. Ilmu mengenai studi tanaman disebut botani, yakni salah satu cabang ilmu biologi (Lewis & McCourt, 2004).

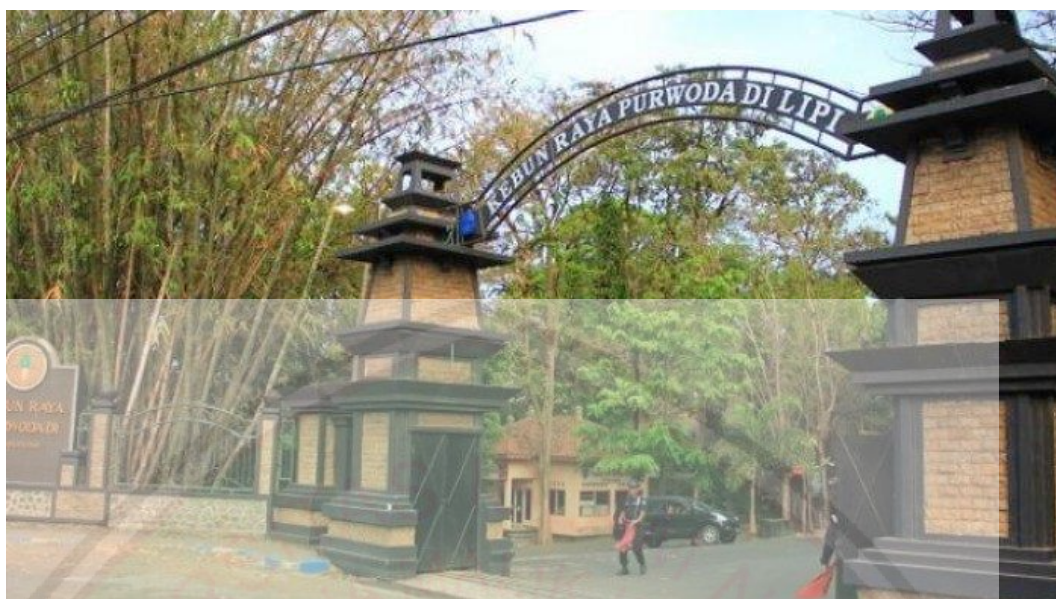
2.2.4 Kebun Raya Purwodadi

2.2.4.1 Sejarah Kebun Raya Purwodadi

Menurut informasi yang dimuat dihalaman resminya, Kebun Raya Purwodadi yang juga dikenal dengan nama Hortus Ilkim Kering Purwodadi didirikan pada tanggal 30 Januari 1941 oleh Dr. L.G.M. Baas Becking. Kebun ini merupakan salah satu dari 3 cabang Kebun Raya Indonesia (Kebun Raya Bogor) yang memiliki tugas dan fungsi mengkoleksi tumbuhan yang hidup di dataran rendah kering. Kebun Raya Purwodadi merupakan Balai Konservasi Tumbuhan yang bernaung dibawah dan bertanggung jawab kepada Pusat Konservasi

Tumbuhan Kebun Raya, Kedeputan Bidang Ilmu Pengetahuan Hayati-LIPI (Lembaga Ilmu Pengetahuan Indonesia) (Tentang Kebun Raya Purwodadi, 2008).

Johannes Viets adalah orang Belanda pertama yang memimpin Kebun Raya Purwodadi yaitu pada tahun 1941-1942, dengan pengangkatannya sebagai pimpinan pada tanggal 30 Januari 1941, Johannes Viets meletakkan pola-pola dasar pengembangan kebun raya. Hal penting yang dilakukannya yaitu areal Kebun Raya Purwodadi yang baru diperoleh dari masyarakat berupa tanah sawah dan pekarangan yang ditanami dengan tanaman penutup tanah serta lamtoro untuk menambah kesuburan tanah secara alami. Pada Tahun 1943, Kebun Raya Purwodadi dipimpin oleh bangsa Jepang, yaitu Tanaka. Pada masa kepemimpinannya, Tanaka membangun jalan utama yang membelah kebun menjadi dua, serta jalan-jalan lain dari arah utara ke selatan. Setelah masa kemerdekaan sampai saat ini, Kebun Raya Purwodadi dipimpin oleh bangsa Indonesia sendiri kecuali pada tahun 1949-1954, yaitu oleh seorang Belanda H.O. van Leusen. Moestopo (1945-1949) adalah orang Indonesia pertama yang memimpin Kebun Raya Purwodadi (Sejarah Kebun Raya Purwodadi, 2008).



Gambar 2.7 Pintu masuk Kebun Raya Purwodadi

Pada mulanya, kebun ini dipergunakan untuk kegiatan penelitian tanaman perkebunan kemudian pada tahun 1954 mulai diterapkan dasar-dasar perkebunrayaan yaitu dengan dimulainya pembuatan petak-petak tanaman koleksi. Kebun Raya Purwodadi untuk pertama kalinya dibuka untuk umum pada masa kepemimpinan Sarwana. Peresmian pembukaan tersebut dilakukan pada tanggal 10 Maret 1963. Setelah pembukaan Kebun Raya untuk umum, pembangunan sarana fisik dan pembangunan sistem pengelolaan kebun semakin digalakkan. Sejak tahun 1980 sebagian tanaman ditata kembali menurut kelompok suku yang menganut sistem klasifikasi Engler dan Prantl. Penyempurnaan vak koleksi, pembangunan gedung kantor, penambahan koleksi melalui eksplorasi, pertukaran biji menjadi program pimpinan Kebun Raya Purwodadi selanjutnya. Dalam perkembangannya, UPT Balai Konservasi Tumbuhan Kebun Raya Purwodadi diharapkan akan menjadi pusat konservasi dan penelitian tumbuhan dataran rendah kering Indonesia.

2.2.4.2 Tugas dan fungsi Kebun Raya Purwodadi

Tugas utama adalah melakukan inventarisasi, eksplorasi dan konservasi tumbuh-tumbuhan dataran rendah kering yang mempunyai nilai ilmu pengetahuan dan ekonomi.

Fungsi Kebun Raya Purwodadi :

1. Melakukan inventarisasi, eksplorasi dan konservasi tumbuh-tumbuhan yang mempunyai nilai ilmu pengetahuan dan ekonomi, dan endemik. Terutama untuk *flora* Indonesia dari dataran rendah kering.
2. Menyediakan fasilitas penelitian, pendidikan dan pemanduan, khususnya di bidang botani.
3. Menyediakan fasilitas rekreasi di alam terbuka.

2.2.4.3 Koleksi di Kebun Raya Purwodadi

Di kebun raya purwodadi memiliki banyak koleksi tanaman-tanaman yang terdiri dari beberapa jenis tanaman, antara lain:

1. Koleksi Anggrek

Ditempatkan di rumah kaca yang kondisinya disesuaikan dengan habitat alaminya. Ada sekitar 2.344 spesimen anggrek alam yang terdiri atas 319 jenis, 69 marga, dan 277 masih sp.. Sekitar 7 jenis merupakan anggrek endemik Jawa Timur seperti *Appendicula imbricata*, *Dendrobium arcuatum*, *Paphiopedilum glaucophyllum* (anggrek selop), dan lain-lain.

2. Koleksi Paku-Pakuan

Koleksi ini ditata di bawah pepohonan besar dan rindang, karena tumbuhan ini membutuhkan tempat rindang dan lembab. Koleksinya mencapai 60 jenis, 36 marga dan 21 suku. Di antaranya Paku Sarang Burung (*Asplenium nidus*), Suplir (*Adiantum trapeziforme*), Paku Hata (*Lygodium circinnatum*), dan Paku Tanduk Rusa/ Simbar Menjangan (*Platyserium coronarium*).

3. Koleksi Polong-Polongan

Digolongkan menjadi 3 suku yaitu *Mimosaceae*, *Caesalpiniaceae*, dan *Papilionaceae*. Ada 157 jenis dari 70 marga yang termasuk dalam suku-suku tersebut. Berbagai jenis polong-polongan dimanfaatkan sebagai tanaman hias seperti jenis-jenis dari marga *Amherstia*, *Brownea*, *Cassia*, *Senna*, dan *Saraca*.

4. Koleksi Pisang

Koleksi pisang di Kebun Raya Purwodadi terdapat sekitar 316 spesimen, 114 kultivar dari 4 jenis induk silangan. Diantara koleksinya adalah *Musa acuminata*, *M. balbisiana* (Pisang Klutuk Wulung/Pisang Hitam) yang merupakan jenis liar yang berperan sebagai induk silangan pisang-pisang kultivar di Indonesia. Salah satu pisang kultivar kebanggaan Kebun Raya Purwodadi adalah Pisang Kates.

5. Koleksi Tanaman Taman

Koleksi tanaman taman Kebun Raya Purwodadi terdiri dari beberapa tanaman hias diantaranya adalah palem, bougenvile, kaktus, dan tanaman hias lain yang ditata secara cantik menjadi sebuah taman yang akan membuat anda betah menikmati keanggunan dan keindahannya. Saat pertama memasuki wilayah kebun, anda akan disugahi sebuah taman palm yang cantik diatas karpet hijau rumput cantik.

6. Koleksi Bambu

Sekitar 30 jenis bambu telah dikoleksi Kebun Raya Purwodadi, 16 jenis berasal dari Jawa, 2 jenis dari Maluku, 2 jenis dari Sulawesi, dan 10 jenis dari beberapa negara Asia (Cina, Jepang, Thailand, India, dan Birma). Beberapa dari koleksi adalah bambu yang bisa dimanfaatkan untuk industri dan ada pula yang rebungnya dimakan.

7. Koleksi Tanaman Obat

Terletak di petak XIV G dan V A, ditata sedemikian rupa hingga berfungsi sebagai taman yang menarik untuk dinikmati. Diantara koleksinya adalah Pace (*Morinda citrifolia*), buahnya untuk obat batuk dan tekanan darah tinggi, Daun ungu (*Graptophyllum pictum*), daunnya untuk obat ambeien (Koleksi Kebun Raya Purwodadi, 2008).

2.2.5 Virtual Reality

Virtual reality adalah sebuah teknologi dimana dapat membuat pengguna atau user dapat berinteraksi dengan lingkungan yang ada dalam dunia maya yang disimulasikan oleh komputer, sehingga pengguna merasa berada didalam

lingkungan tersebut. Di dalam bahasa Indonesia *virtual reality* dikenal dengan istilah realitas maya.

Kelebihan utama dari *virtual reality* adalah pengalaman yang membuat user merasakan sensasi dunia nyata dalam dunia maya. Bahkan perkembangan teknologi *virtual reality* saat ini memungkinkan tidak hanya indra penglihatan dan pendengaran saja yang bisa merasakan sensasi nyata dari dunia maya dari *virtual reality*, namun juga indra yang lainnya. Istilah realitas maya atau *virtual reality* sendiri belum diketahui asalnya. Pengembangnya yaitu Jaron Lanier mengakui bahwa ia menggunakan istilah itu pertama kali dan ada istilah yang terkait digunakan oleh Myron Krueger yaitu “kenyataan tiruan” yang dimana sudah digunakan sejak 1970 (Herlangga, 2016).

Syarat-syarat yang harus ada pada *virtual reality* yaitu :

1. Gambar/grafis/ penglihatan 3D yang nyata menurut perspektif penglihatan pengguna.
2. Kemampuan untuk dapat mendeteksi gerakan-gerakan yang dilakukan oleh pengguna seperti gerakan tangan, kepala, arah gerakan bola mata untuk menyesuaikan grafis yang dihasilkannya supaya dapat menyesuaikan dengan perubahan dunia 3Dnya.

Ada dua komponen dalam perasaan melebur saat menggunakan *virtual reality* ini.

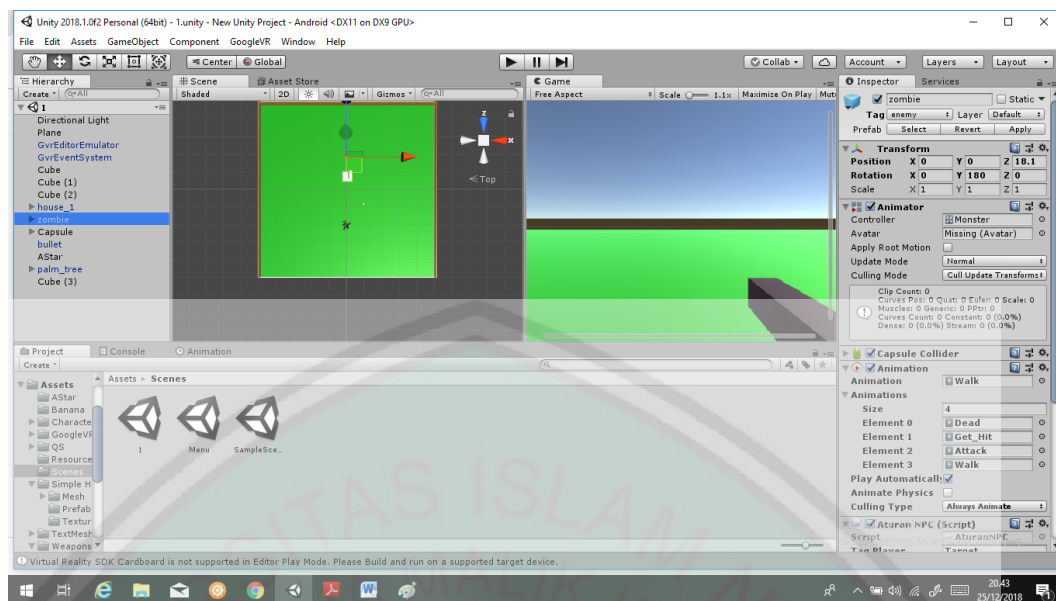
1. *Depth of information* merupakan banyak dan kualitas dari data yang ditransfer demi menciptakan lingkungan *virtual reality*, seperti resolusi, ketajaman gambar, dll.

2. *Breadth of information*, yaitu sampai sejauh mana indera dari pengguna dimanipulasi saat sedang menggunakan *virtual reality*. Yang biasanya terbatas pada penglihatan dan pendengaran. Namun saat ini sedang dikembangkan *virtual reality* yang dapat memanipulasi sentuhan dan bau (Armbruster, Kuhlen, Wolter, & Fimm, 2008)

2.2.6 Unity

2.2.6.1 Pengertian Unity

Unity merupakan suatu aplikasi yang digunakan untuk mengembangkan *Game* multi platform yang didesain untuk mudah digunakan. *Unity* itu bagus dan penuh perpaduan dengan aplikasi yang profesional. Editor pada *Unity* dibuat dengan *user interface* yang sederhana. Editor ini dibuat setelah ribuan jam yang mana telah dihabiskan untuk membuatnya menjadi nomor satu dalam urutan ranking teratas untuk editor *Game*. Grafis pada *Unity* dibuat dengan grafis tingkat tinggi untuk *OpenGL* dan *directX*. *Unity* mendukung semua *format file*, terutamanya format umum seperti semua format dari *art applications*. *Unity* cocok dengan versi 64-bit dan dapat beroperasi pada Mac OSx dan *Windows* dan dapat menghasilkan *Game* untuk Mac, *Windows*, *Wii*, *iPhone*, *iPad* dan *Android*. *Unity* secara rinci dapat digunakan untuk membuat video *Game* 3D, real time animasi 3D dan visualisasi arsitektur dan isi serupa yang interaktif lainnya.



Gambar 2.8 Tampilan Unity

Editor *Unity* dapat menggunakan plugin untuk web player dan menghasilkan *Game browser* yang didukung oleh *Windows* dan *Mac*. *Plugin web player* dapat juga dipakai untuk *widgets Mac*. *Unity* juga akan mendukung *console* terbaru seperti *PlayStation 3* dan *Xbox 360*. Pada tahun 2010, telah memperoleh *Technology Innovation Award* yang diberikan oleh *Wall Street Journal* dan tahun 2009, *Unity Technology* menjadi 5 perusahaan *Game* terbesar. Tahun 2006, menjadi juara dua pada *Apple Design Awards*.

Server aset dari *Unity* dapat digunakan semua *scripts* dan aset *Game* sebagai solusi dari versi kontrol dan dapat mendukung proyek yang terdiri atas banyak *gigabytes* dan ribuan dari file multi-megabyte. Editor *Unity* dapat menyimpan metadata dan versi mereka, itu dapat berjalan, pembaharuan dan didalam perbandingan versi grafis. Editor *Unity* dapat diperbaharui dengan sesegera mungkin seperti file yang telah dimodifikasi. *Server aset Unity* juga

cocok pada *Mac*, *Windows* dan *Linux* dan juga berjalan pada *PostgreSQL*, *database server opensource*.

Perizinan atau *license* dari *Unity* ada dua bentuk. Ada *Unity* dan *Unity Pro*. Versi *Unity* tersedia dalam bentuk gratis, sedang versi *Unity Pro* hanya dapat dibeli. Versi *Unity Pro* ada dengan fitur bawaan seperti efek *post processing* dan *render efek texture*. Versi *Unity* merupakan yang gratis memperlihatkan aliran untuk *Game web* dan *layar splash* untuk *Game* yang berdiri sendiri. *Unity* dan *Unity Pro* menyediakan *tutorial*, isi, contoh *project*, *wiki*, dukungan melalui forum dan perbaruan kedepannya. *Unity* digunakan pada *iPhone*, *iPod* dan *iPad* *operating system* yang mana *iOS* ada sebagai *add-ons* pada *Unity* editor yang telah ada lisensinya, dengan cara yang sama juga pada *Android*.

2.2.6.2 *Fitur-Fitur Unity*

Unity memiliki beberapa fitur-fitur untuk membantu para pengembang dalam melakukan pembuatan aplikasi dan *game*, berikut fitur-fitur dalam editor *unity*:

a) *Rendering*

Graphics Engine yang digunakan adalah *Direct3D* (*Windows*, *Xbox 360*), *OpenGL* (*Mac*, *Windows*, *Linux*, *PS3*), *OpenGL ES* (*Android*, *iOS*), dan *proprietary APIs* (*Wii*). Ada pula kemampuan untuk *bump mapping*, *reflection mapping*, *parallax mapping*, *screen space ambient occlusion* (*SSAO*), *dynamic shadows using shadow maps*, *render-to-texture* and *full-screen post-processing effects*.

Unity dapat mengambil format desain dari *3ds Max*, *Maya*, *Softimage*, *Blender*, *modo*, *ZBrush*, *Cinema 4D*, *Cheetah3D*, *Adobe Photoshop*, *Adobe*

Fireworks and Allegorithmic Substance. Asset tersebut dapat ditambahkan ke *Game project* dan diatur melalui *graphical user interface Unity*.

ShaderLab adalah bahasa yang digunakan untuk *shaders*, dimana mampu memberikan deklaratif “*programming*” dari *fixed-function pipeline* dan program *shader* ditulis dalam GLSL atau Cg. Sebuah *shader* dapat menyertakan banyak varian dan sebuah spesifikasi *fallback declarative*, dimana membuat *Unity* dapat mendeteksi berbagai macam *video card* terbaik saat ini, dan jika tidak ada yang kompatibel, maka akan dilempar menggunakan *shader* alternatif yang mungkin dapat menurunkan fitur dan performa.

Pada 3 Agustus 2013, seiring dengan diluncurkannya versi 4.2, *Unity* mengizinkan *developer indie* menggunakan *Realtime shadows* hanya untuk *Directional lights*, dan juga menambahkan kemampuan dari DirectX11 yang memberikan *shadows* dengan resolusi *pixel* yang lebih sempurna, tekstur untuk membuat objek 3d dari *grayscale* dengan lebih grafik *facial*, animasi yang lebih halus dan mempercepat FPS.

b) *Scripting*

Script Game engine dibuat dengan Mono 2.6, sebuah implementasi *open-source* dari *.NET Framework*. Programmer dapat menggunakan *UnityScript* (bahasa terkustomisasi yang terinspirasi dari *syntax ECMAScript, dalam bentuk JavaScript*), C#, atau Boo (terinspirasi dari *syntax* bahasa pemrograman *python*). Dimulai dengan dirilisnya versi 3.0, *Unity* menyertakan versi *Monodevelop* yang terkustomisasi untuk *debug script*.

c) *Platform*

Unity support pengembangan ke berbagai *platform*. Didalam, *developer* memiliki kontrol untuk mengirim perangkat *mobile*, *web browser*, *desktop*, dan *console*. *Unity* juga mengijinkan spesifikasi kompresi tekstur dan pengaturan resolusi di setiap *platform* yang didukung.

Saat ini *platform* yang didukung adalah *BlackBerry 10*, *Windows 8*, *Windows Phone 8*, *Windows*, *Mac*, *Linux*, *Android*, *iOS*, *Unity Web Player*, *Adobe Flash*, *PlayStation 3*, *Xbox 360*, *Wii U* and *Wii*. Meskipun tidak semua terkonfirmasi secara resmi, *Unity* juga mendukung *PlayStation Vita* yang dapat dilihat pada *Game Escape Plan* dan *Oddworld: New 'n' Tasty*.

Rencana *platform* berikutnya adalah *PlayStation 4* dan *Xbox One*. Dan juga rumor untuk kedepanya mengatakan *HTML* akan menjadi platformnya, dan *plug-in Adobe* baru dimana akan disubtitusikan ke *Flash Player*, juga akan menjadi *platform* berikutnya.

d) *Asset Tracking*

Unity juga menyertakan *Server Unity Asset* – sebuah solusi terkontrol untuk *developer Game asset* dan *script*. Server tersebut menggunakan *PostgreSQL* sebagai *backend*, sistem audio dibuat menggunakan *FMOD library* (dengan kemampuan untuk memutar *Ogg Vorbis compressed audio*), *video playback* menggunakan *Theora codec*, *engine* daratan dan vegetasi (dimana mensupport *tree billboarding*, *Occlusion Culling* dengan *Umbra*), *built-in lightmapping* dan *global illumination* dengan *Beast*, *multiplayer networking* menggunakan *RakNet*, dan navigasi *mesh* pencari jalur *built-in*.

e) *Asset Store*

Diluncurkan November 2010, *Unity Asset Store* adalah sebuah *resource* yang hadir di *Unity editor*. *Asset store* terdiri dari koleksi lebih dari 4,400 *asset packages*, beserta *3D models*, *textures* dan *materials*, sistem *particle*, musik dan efek suara, tutorial dan *project*, *scripting package*, *editor extensions* dan *servis online*.

f) *Physics*

Unity juga memiliki *support built-in* untuk *PhysX physics engine* (sejak *Unity 3.0*) dari *Nvidia* (sebelumnya *Ageia*) dengan penambahan kemampuan untuk simulasi *real-time cloth* pada *arbitrary* dan *skinned meshes*, *thick ray cast*, dan *collision layers* (Rosikhana M, Aristiawan, 2013).

BAB III

ANALISA DAN PERANCANGAN GAME

Bab ini akan menjelaskan tentang analisa dan perancangan sistem dari implementasi metode A* pada *game* labirin dengan menggunakan teknologi *virtual reality* untuk menemukan jalur terpendek. Analisa dan perancangan sistem dilakukan sesuai kebutuhan dengan tujuan agar mempermudah dalam proses pembuatan aplikasi.

3.1 Analisis dan Perancangan Game

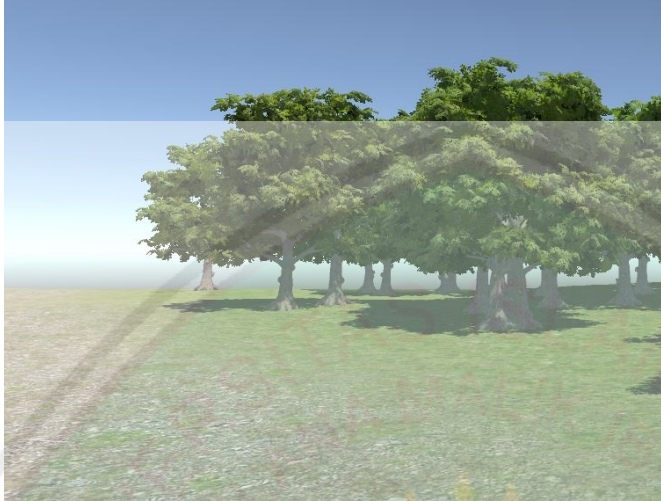

Analisa aplikasi ini bertujuan untuk mengidentifikasi masalah – masalah yang ada pada aplikasi kemudian menentukan kebutuhan-kebutuhan yang dapat membantu dalam mempermudah dalam pembangunan aplikasi. Pada penelitian ini kami mencoba untuk membangun sebuah aplikasi *game* labirin *virtual reality* dengan menggunakan metode A* untuk menemukan jalur terpendek yang mana metode ini diimplementasikan pada NPC yakni NPC.



3.1.1 Storyboard

Permainan ini bernama PENGENALAN TANAMAN KEBUN RAYA PURWODADI pada permainan ini, menceritakan seseorang yang sedang berjalan-jalan didalam kawasan kebun raya purwodadi. Dia harus menjaga tanaman yang berada dalam hutan buatan kebun raya purwodadi. Pemain akan dihadapkan oleh NPC, NPC akan berjalan dari arah yang tidak diketahui untuk mencari tumbuhan dan menghancurkannya. Namun, pemain akan diberi 1 jenis item bantuan yakni kotak SEHAT agar membantu pemain menemukan dan mencapai titik-titik tanaman. kotak sehat digunakan menambah nyawa pemain yang berkurang setelah diserang oleh NPC.

Tabel dibawah akan menjelaskan mengenai *Storyboard* dari permainan ini.

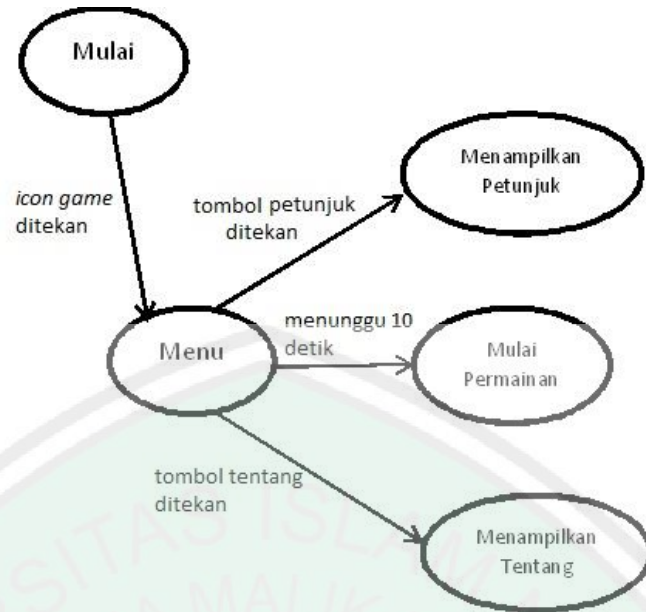
Tabel 3.1 *Storyboard* dari permainan

Scene	Keterangan
	<p>Ketika pemain memilih menu main, pemain akan <i>spawn</i> di dalam hutan. Pemain harus berjalan menyusuri hutan untuk mencari tanaman .</p>
	<p><i>NPC</i> akan muncul di tempat yang acak, setelah itu dengan adanya algoritma A^* maka <i>NPC</i> akan melakukan perjalanan berdasarkan jalur terpendek yang telah ditemukan oleh algoritma A^*</p>
	<p>Item yang membantu yakni Kotak SEHAT akan <i>spawn</i> ditempat labirin secara acak, item kotak SEHAT akan menambah nyawa yang terluka.</p>

	<p>Tanaman akan diserang oleh <i>NPC</i> untuk dihancurkan</p>
	<p>Pemain melakukan tembakan pada <i>NPC</i> dan akan menghilang sehingga <i>NPC</i> tidak bisa melakukan serangan kepada tanaman.</p>

3.1.2 Finite State Machine (FSM)

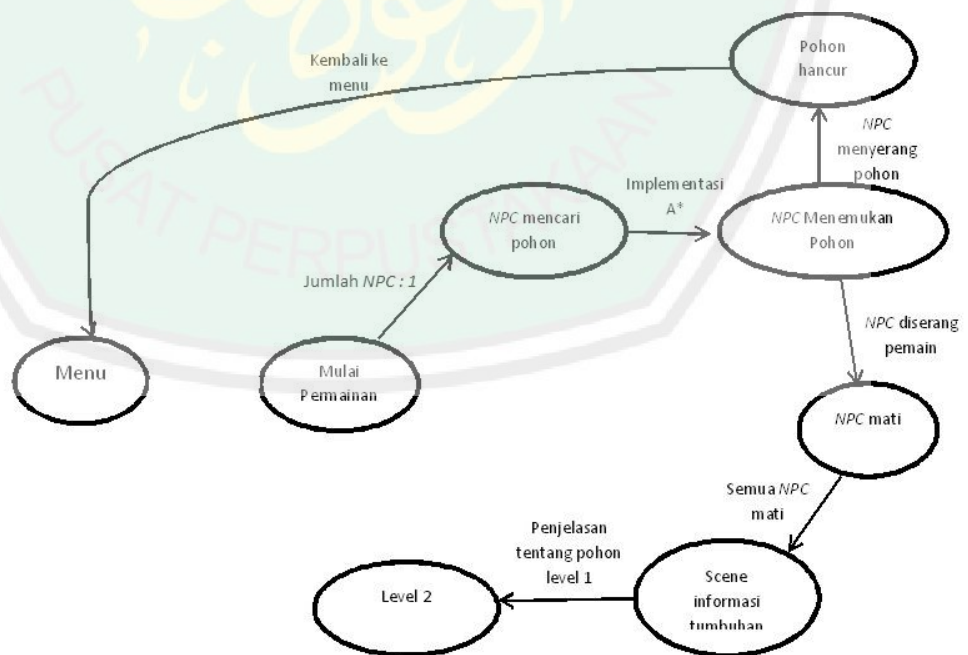
3.1.2.1 Finite State Machine (FSM) pada Menu Game



Gambar 3.1 *Finite State Machine (FSM)* pada menu game

Pada *FSM* diatas menunjukan 3 submenu pilihan yakni menu main, menu tentang dan menu petunjuk.

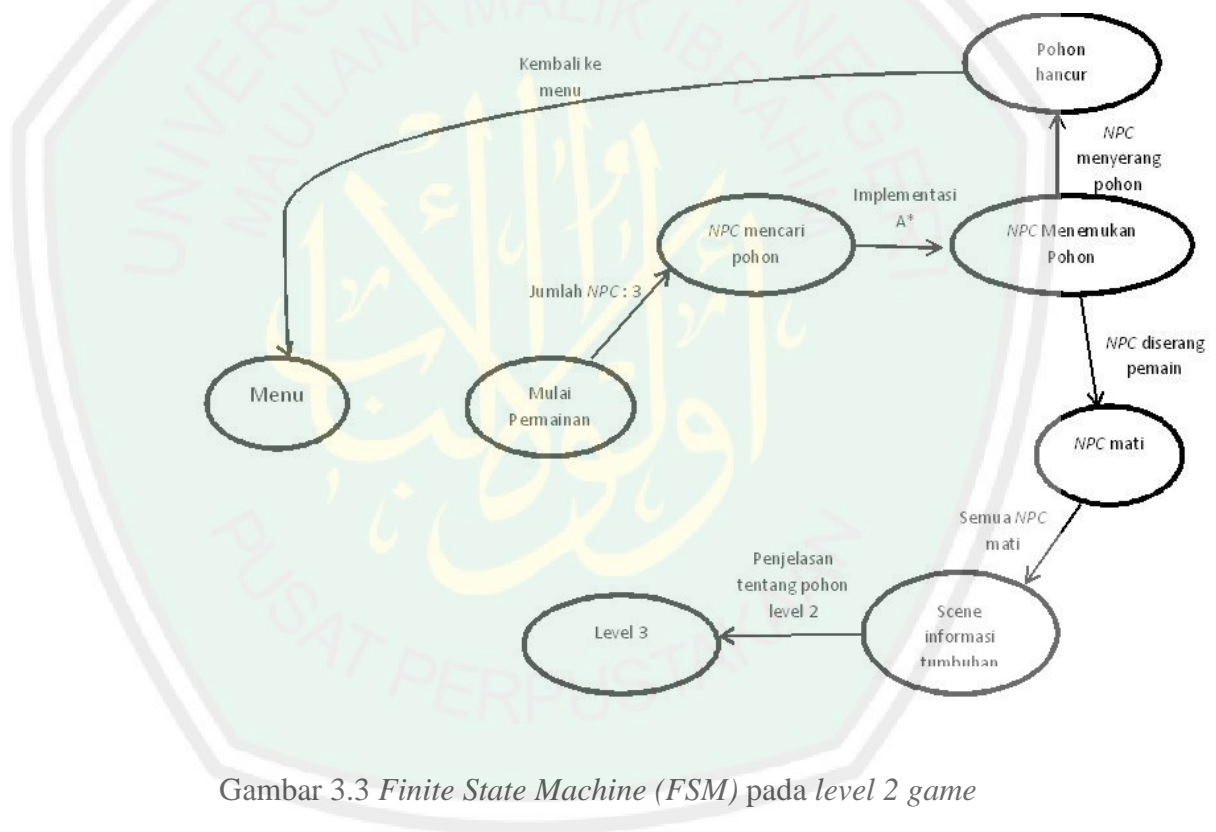
3.1.2.2 Finite State Machine (FSM) pada NPC Level 1



Gambar 3.2 *Finite State Machine (FSM)* pada *NPC level 1 game*

Pada *FSM* di atas menjelaskan bahwa *NPC* akan mencari target berupa tumbuhan dengan menerapkan algoritma A^* , *NPC* akan melakukan serangan untuk mengurangi nyawa tumbuhan dan jika tumbuhan telah kehabisan nyawa permainan akan berakhir. Pada *level* ini jumlah *NPC* yaitu 1 dengan keadaan normal.

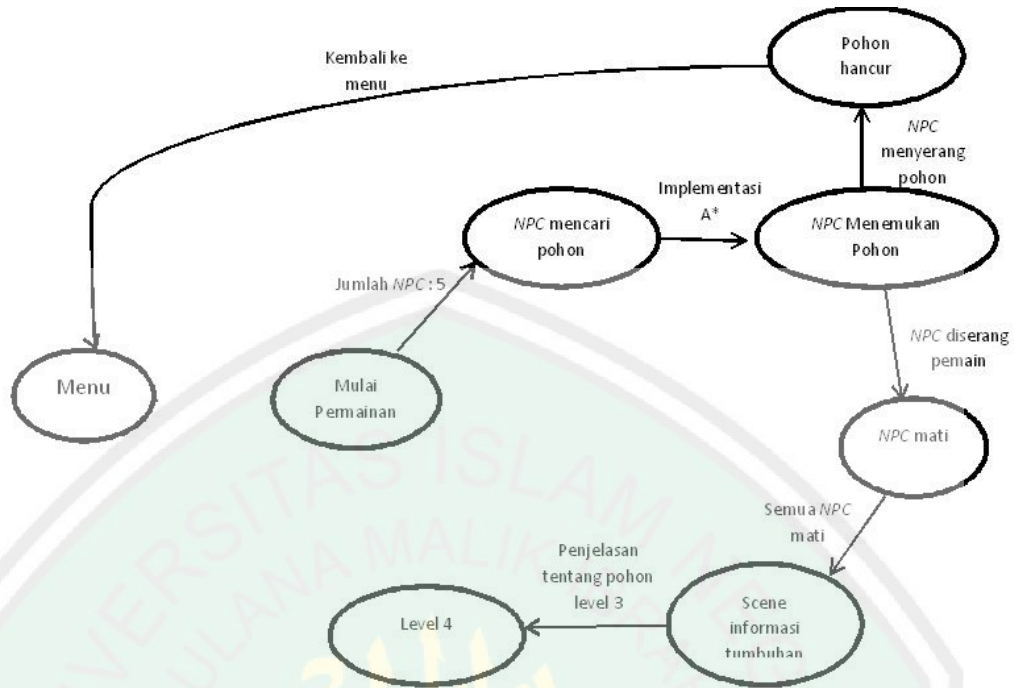
3.1.2.3 *Finite State Machine (FSM)* pada *NPC Level 2*



Gambar 3.3 *Finite State Machine (FSM)* pada *level 2 game*

Pada *FSM* di atas menjelaskan bahwa terdapat 3 *NPC* yang akan mencari target berupa tumbuhan dengan menerapkan algoritma A^* , *NPC* akan melakukan serangan untuk mengurangi nyawa tumbuhan dan jika tumbuhan telah kehabisan nyawa permainan akan berakhir. Pada *level* ini jumlah *NPC* yaitu 3 dengan keadaan normal.

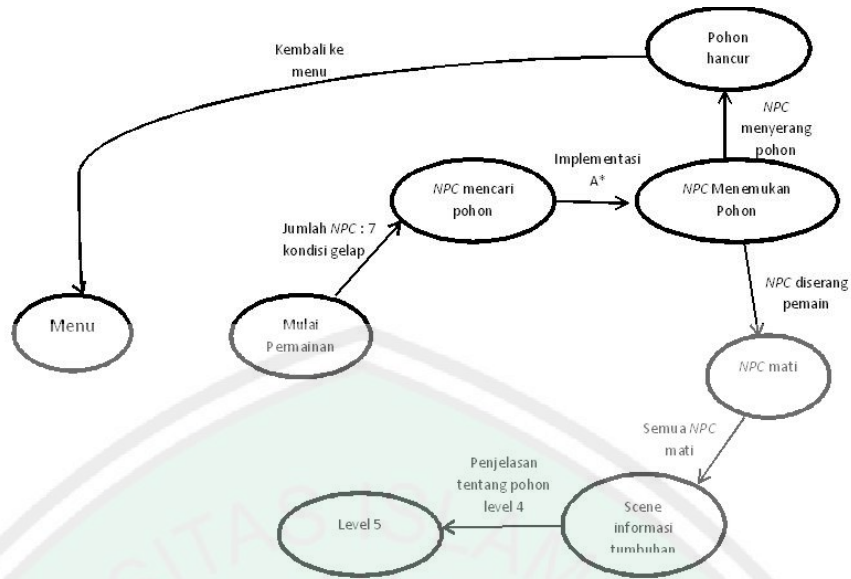
3.1.2.4 Finite State Machine (FSM) pada NPC Level 3



Gambar 3.4 Finite State Machine (FSM) pada level 3 game

Pada FSM di atas menjelaskan bahwa terdapat 5 NPC yang akan mencari target berupa tumbuhan dengan menerapkan algoritma A*, NPC akan melakukan serangan untuk mengurangi nyawa tumbuhan dan jika tumbuhan telah kehabisan nyawa permainan akan berakhir. Pada level ini jumlah NPC yaitu 5 dengan keadaan normal.

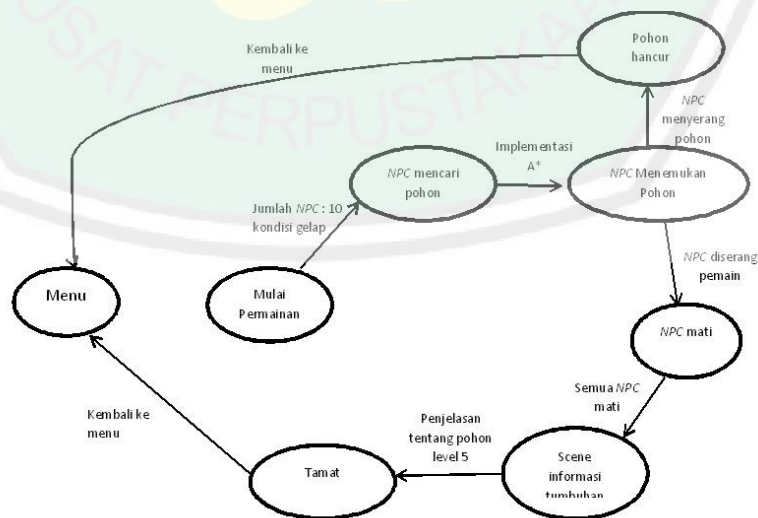
3.1.2.5 Finite State Machine (FSM) pada NPC Level 4



Gambar 3.5 *Finite State Machine (FSM)* pada level 4 game

Pada *FSM* di atas menjelaskan bahwa terdapat 7 *NPC* yang akan mencari target berupa tumbuhan dengan menerapkan algoritma A^* , *NPC* akan melakukan serangan untuk mengurangi nyawa tumbuhan dan jika tumbuhan telah kehabisan nyawa permainan akan berakhir. Pada *level* ini jumlah *NPC* yaitu 7 dengan keadaan gelap.

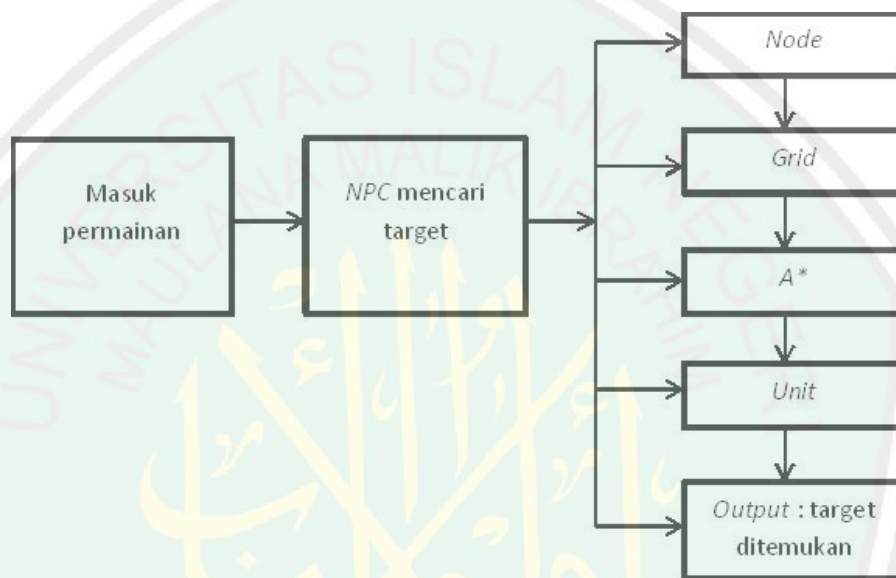
3.1.2.5 *Finite State Machine (FSM)* pada NPC Level 5



Gambar 3.6 *Finite State Machine (FSM)* pada level 5 game

Pada *FSM* di atas menjelaskan bahwa terdapat 10 *NPC* yang akan mencari target berupa tumbuhan dengan menerapkan algoritma *A**, *NPC* akan melakukan serangan untuk mengurangi nyawa tumbuhan dan jika tumbuhan telah kehabisan nyawa permainan akan berakhir. Pada *level* ini jumlah *NPC* yaitu 10 dengan keadaan gelap.

3.1.3 Blok Diagram pada Game



Gambar 3.7 Blok Diagram penempatan *A** pada *NPC*

Pada blok diagram seperti pada gambar 3.7 menjelaskan alur dari algoritma *A** pada *NPC* dimulai dari input hingga proses yang dilalui algoritma *A** untuk mencari targetnya. Berikut merupakan keterangan mengenai proses-proses untuk mencari target pada *NPC* berdasarkan blok diagram diatas:

1. *Masuk permainan* : merupakan awal sistem berjalan.
2. *NPC mencari target* : memasukkan algoritma *A** untuk mencari rute terpendek menuju target.
3. *Node* : menampung nilai dalam konstraktor untuk pembuatan *node*.
4. *Grid* : pemetaan tempat untuk menandakan jalur *walkable* (yang dapat dilewati) dan *unwalkable* (halangan) berdasarkan jumlah grid yang telah diatur.

5. A^* : proses perhitungan *pathfinding* untuk menemukan jalur terpendek menuju target.
6. *Unit* : proses untuk menjalankan *NPC* menuju target nya berdasarkan proses perhitungan A^* .
7. *Output* : hasil yang diperoleh pada perhitungan A^* adalah *NPC* menemukan target dan melakukan serangan.

3.1.3 Deskripsi Karakter dan Item

Analisis mengenai deskripsi karakter pada *game* labirin *virtual reality*.

a) *NPC*



Gambar 3.8 Penampilan *NPC* pada *game* pengenalan tumbuhan *virtual reality*

NPC merupakan makhluk yang ditugaskan untuk mengganggu pemain agar tidak mencapai titik akhir. Agar tidak mencapai titik akhir, *NPC* akan mencari pemain dan setelah pemain telah terlihat oleh *NPC*, maka *NPC* akan mendekat dan akan melakukan penyerangan terhadap pemain. *NPC* tidak akan mati karena pemain harus menghindari dengan cara berlari.

b) Pemain

Pemain adalah tokoh utama yang dimainkan oleh user dimana memiliki misi untuk mencapai titik akhir sebuah labirin. Karakter

pemain tidak terlihat karena kita akan mengambil sistem permainan dari sudut pandang mata pemain, namun pemain akan di siapkan dengan senjata pedang untuk melakukan serangan kepada *NPC*.

c) Kotak Sehat



Gambar 3.9 Tampilan dari kotak SEHAT

Kotak SEHAT merupakan item yang dibutuhkan pemain ketika nyawa mereka berkurang karena serangan dari *NPC*, ketika pemain mengambil item ini, maka nyawa mereka akan bertambah. Kotak ini akan muncul secara acak.

3.1.4 Deskripsi Objek Tanaman

Untuk menunjang permainan ini, maka dipilih 5 jenis tanaman koleksi kebun raya purwodadi, ke 5 jenis tumbuhan tersebut antara lain:

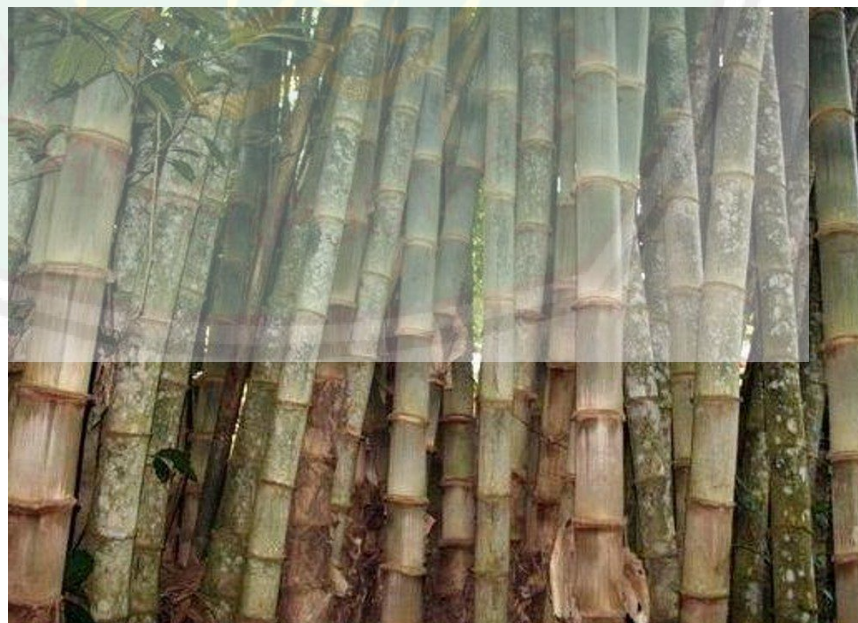
1. Anggrek



Gambar 3.10 Anggrek Hitam

Anggrek hitam yang eksotis ini memiliki nama ilmiah *Coelogyne pandurata Lindl.* Dinamakan anggrek hitam karena labelumnya berwarna hitam. Jenis anggrek dengan *pseudobulb* yang sangat pipih ini memiliki sepal dan petal yang berwarna hijau. Ciri khas lainnya yaitu anggrek ini mengeluarkan bau semerbak yang wangi (Anggrek Hitam, 2008).

2. Bambu



Gambar 3.11 Bambu Petung

Jenis bambu ini dicirikan dengan rumpunnya yang simpodial, tegak dan rapat. Rebung berwarna hitam keunguan, tertutup bulu berwarna coklat hingga kehitaman. Tinggi buluh mencapai 20 m, tegak dengan ujung melengkung. Buluh berwarna hijau, hijau tua atau hijau keunguan, atau agak hijau keputih-putihan dan bertotol putih karena ada lumut ketika tua, selain itu buku-bukunya dikelilingi oleh akar udara (Bambu Petung, 2008).

3. Paku



Gambar 3.12 *Selaginella plana*

Daun kecil, tunggal, melingkari batang, dan pada cabang sampingnya tersusun dalam 4 baris (2 daun samping terdiri dari daun besar yang mudah rontok, dan 2 baris depan berdaun kecil yang duduknya menempel). Sporangia berada di ketiak daun yang fertil. Sporofil lebih besar daripada sporangia, terkumpul menjadi bulir terminal berbentuk persegi 4 dan terkadang agak pipih (Selaginella Plana, 2008).

4. Pisang



Gambar 3.13 *Musa acuminata Colla var. Rutilifes*

Varietas ini berukuran sedang dengan tangkai daun kekuningan. Perbungaan horisontal kemudian menjurai. Jantung berbentuk bulat telur berujung lancip. Buah berbentuk silindris, ramping dengan ujung lancip, panjang hingga 10 cm. Kulit luar tipis, daging buah kuning pada saat matang, lembut dan manis. Biji bundar-pipih-mampat, berwarna coklat kehitaman dan berdiameter sekitar 3-4 mm (Musa acuminata , 2008)

5. Palem



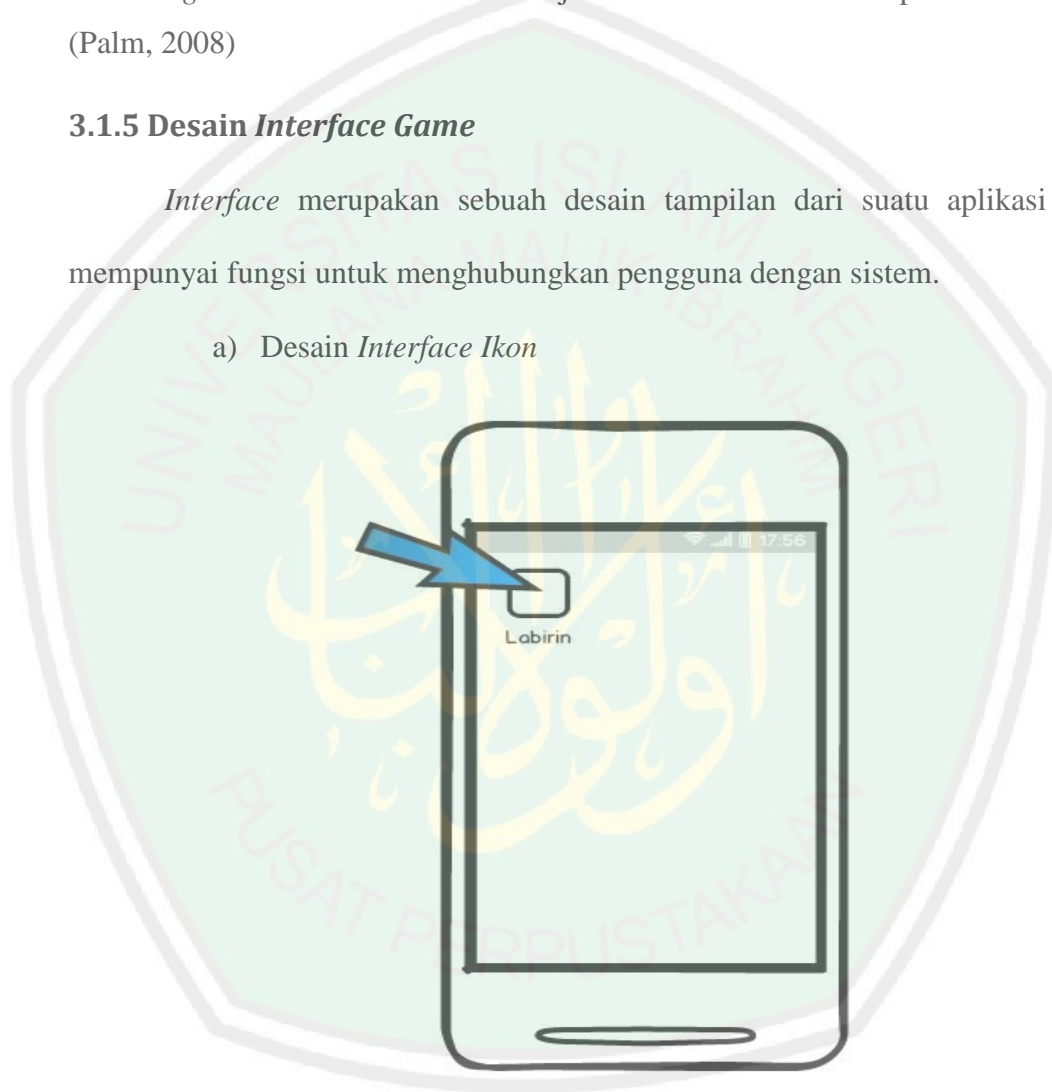
Gambar 3.14 Palm

Palem termasuk dalam famili *Arecaceae* dan merupakan jenis-jenis tertua yang telah dijumpai sejak jaman Cretaceus, kurang lebih 120 juta tahun yang lalu. *Arecaceae* sangat menarik dari segi botani, keindahan bentuknya, keanekaragaman jenis dan kegunaannya. Famili *Arecaceae* di dunia diperkirakan 200-300 *genus* dan sekitar 2000-3000 jenis tersebar di daerah tropis dan *sub tropis* (Palm, 2008)

3.1.5 Desain *Interface Game*

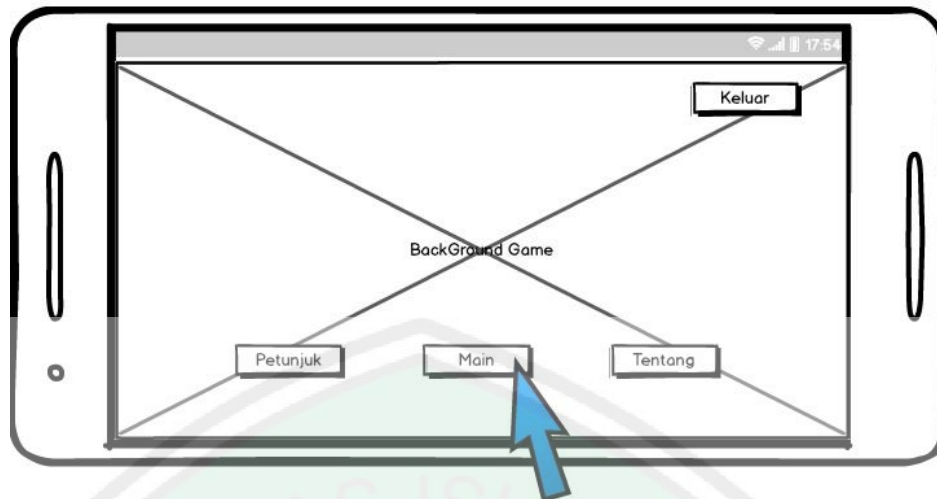
Interface merupakan sebuah desain tampilan dari suatu aplikasi yang mempunyai fungsi untuk menghubungkan pengguna dengan sistem.

a) Desain *Interface Ikon*

Gambar 3.15 Desain *Interface Ikon*

Desain *interface icon* merupakan desain untuk mengetahui ikon game pengenalan tumbuhan dengan ikon aplikasi pada *smartphone*.

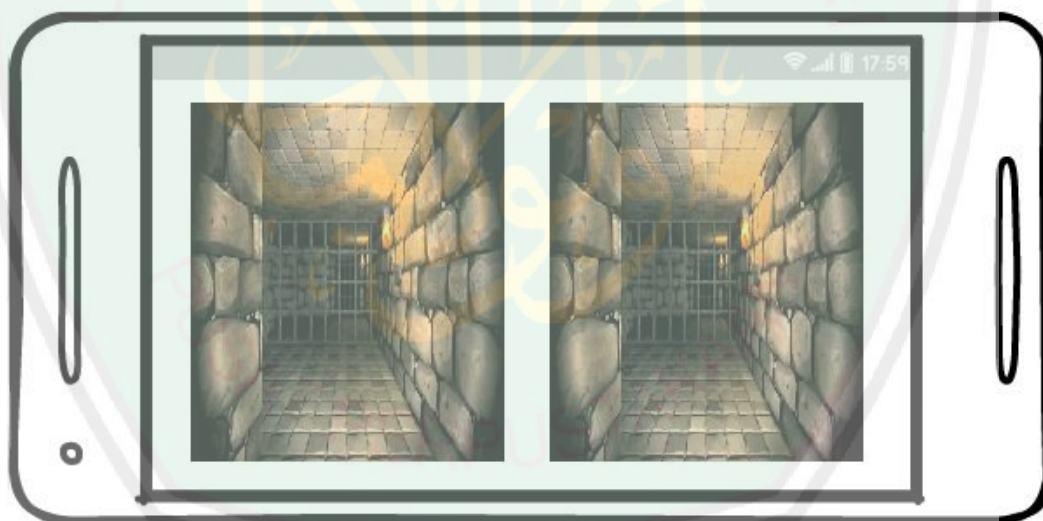
b) Desain *Interface Menu Utama*



Gambar 3.16 Desain *Interface* Menu Utama

Desain menu utama terdiri dari 3 menu yakni menu main, menu petunjuk dan menu tentang.

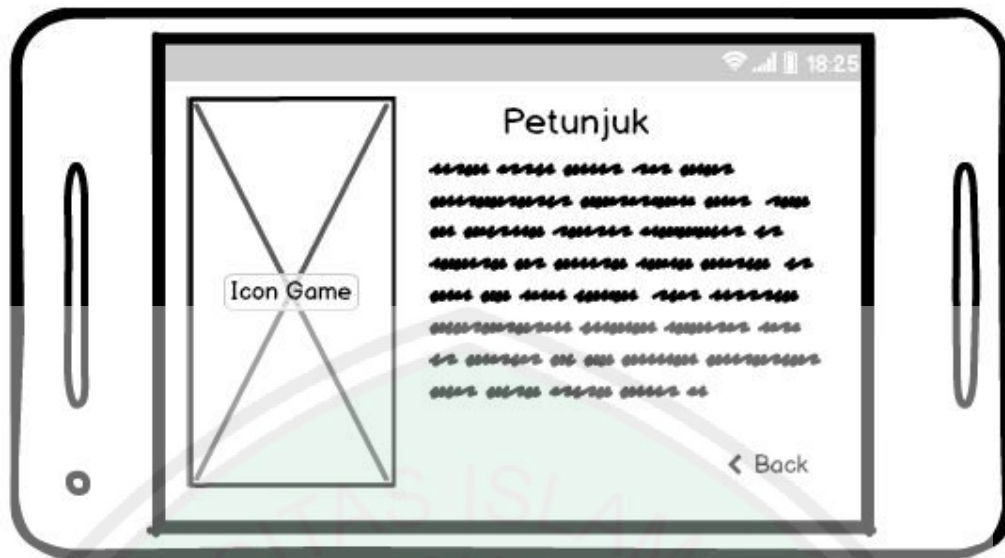
c) Desain *Interface* Permainan



Gambar 3.17 Desain *Interface* Permainan

Desain permainan menjadi *split* atau 2 bagian dikarenakan teknologi *virtual reality* dengan sendirinya akan membagi 2 tampilan permainan sehingga ketika dipakai pada *google cardboard* akan terasa kita berada didalam *game*.

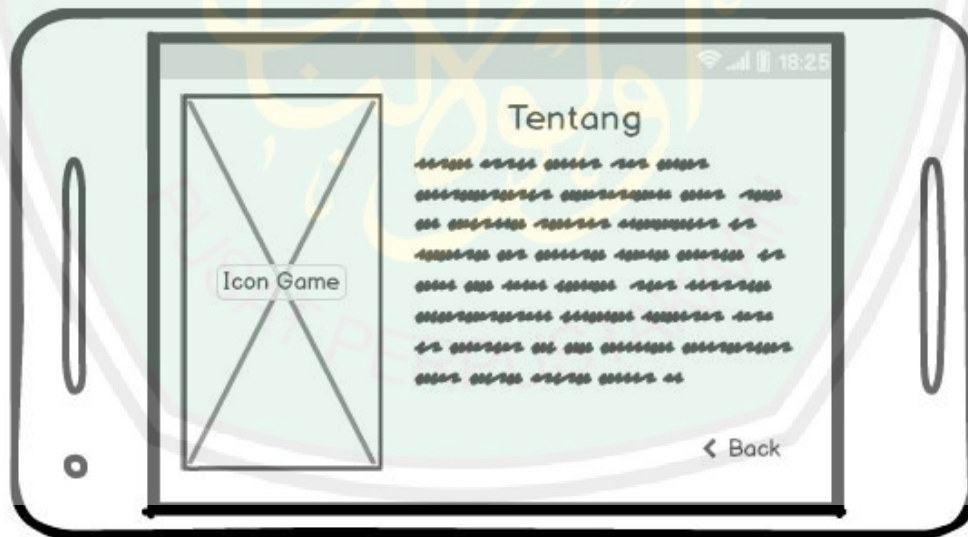
d) Desain *Interface* Menu Petunjuk



Gambar 3.18 Desain *Interface* menu Petunjuk

Desain menu petunjuk akan menampilkan bagaimana kita bermain game pengenalan tumbuhan jika dirasa pemain belum mengerti cara bermainnya.

e) Desain *Interface* Menu Tentang



Gambar 3.19 Desain *Interface* Tentang

Desain menu tentang akan menampilkan tentang game ini mulai dari biodata penulis hingga manfaat *game* ini.

3.2. Perancangan Metode

Rancangan metode merupakan sebuah rancangan yang berfungsi untuk mengimplementasikan sebuah rumus matematika ke dalam sebuah sistem. Pada aplikasi *game* labirin *virtual reality* ini menggunakan metode A^* (*AStar*) yang diimplementasikan pada perilaku *NPC* nya.

3.2.1 Rancangan Metode A^* (*AStar*)

Dalam *game* ini A^* digunakan untuk menentukan perilaku yang dilakukan oleh *Non Player Character (NPC)*. Dengan adanya metode ini yang diterapkan pada *NPC* musuh dapat menentukan perilaku musuh pada *game* ini yang diterapkan pada *NPC* sehingga *NPC* akan mencari karakter utama dengan mencari jalan terpendek untuk menuju karakter tersebut.

3.2.2 Simulasi A^* (*AStar*)

Berikut ini simulasi dari perhitungan manual pencarian jalur terpendek menggunakan algoritma A^* (*AStar*).

$$F(n) = G(n) + H(n) \quad (3.1)$$

Dimana :

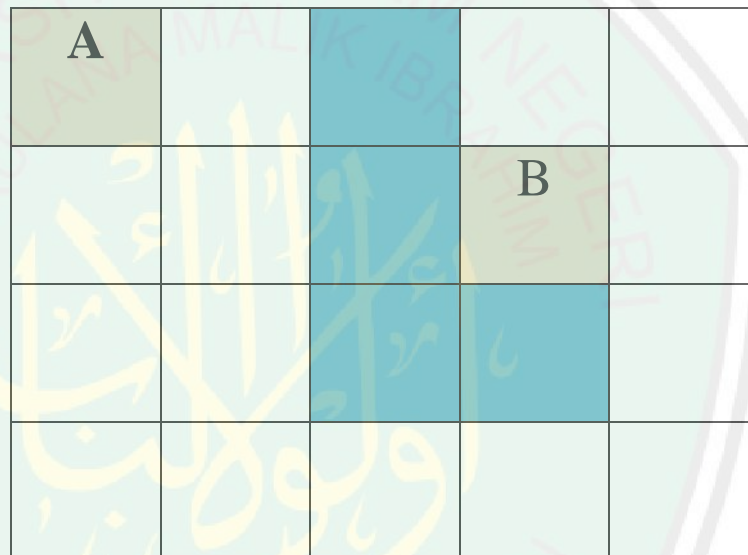
$F(n)$ = harga untuk simpul n

$G(n)$ = harga untuk mencapai simpul n dari akar. G digunakan nilai 10 untuk gerakan vertikal atau horisontal. Angka 10 digunakan untuk memudahkan perhitungan. Sedangkan pada gerakan diagonal digunakan perhitungan $10 \times \sqrt{2} = 14,14$. Demi memudahkan perhitungan angka dibulatkan menjadi 14.

$H(n)$ = jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan.

Untuk perhitungan nilai H digunakan fungsi heuristic, metode yang digunakan di dalam contoh ini adalah metode Manhattan dimana perhitungan jumlah *node* hanya yang bergerak secara vertikal dan horizontal menuju tujuannya serta mengabaikan penghalang, yang kemudian nilainya dikalikan dengan 10. Atau dirumuskan dengan:

$$H = 10 * (abs(currentX - targetX) + abs(currentY - targetY)) \quad (3.2)$$



Gambar 3.20 Tahap awal pencarian algoritma A*

Pencarian wilayah dibagi ke dalam *node-node* yang terdiri dari *node* awal (*node* A) dan *node* akhir (*node* B). Selanjutnya, melakukan pencarian untuk menemukan jalan terpendek dengan cara memeriksa *node-node* yang berdekatan dengan *node* awal untuk bisa sampa ke *node* B.

Berikut adalah h pencarian yang dilakukan oleh *node* awal:

4. h pertama, dimulai dari *node* awal yang ditambahkan ke dalam *Open List* yang merupakan *list* untuk *node-node* yang harus diperiksa.

5. Periksa semua *node* yang berdekatan dengan *node* awal apakah bisa dilalui atau tidak, *node* penghalang yang berwarna biru diabaikan lalu tambahkan semua *node* tersebut ke dalam *Open List*, untuk setiap *node* yang baru ditambahkan, simpan *node* awal sebagai “*parent*” yang nantinya akan digunakan untuk menelusuri jalan.
6. Hapus *node* awal dari *Open List*, dan tambahkan ke dalam *closed list*. *Close list* merupakan *list* untuk *node* yang perlu diperiksa atau dalam artian merupakan bagian dari jalur terpendek yang sudah didapatkan.

<i>Close list</i> A	<i>Open list</i>			
<i>Open list</i>	<i>Open list</i>		B	

Gambar 3.21 Ilustrasi *open list* dan *closed list*

h selanjutnya adalah memilih *node* yang memiliki nilai F terendah di dalam *open list* dengan menggunakan rumus ($F=G+H$), G adalah nilai yang dikeluarkan untuk *node* A ke *node* yang ada disekitarnya. Sedangkan H adalah fungsi *heuristic* yang digunakan untuk menghitung perkiraan biaya yang dikeluarkan dari *node* awal menuju *node* tujuan.

0	8	1	6			
	1		2	3	4	5
8	NPC	8				

2	6	3	4	8	9	10
8	6	7	7	Player		
11	12	13	14	15		
16	17	18	19	20		

Gambar 3.22 Perhitunag menentukan nilai F

Proses ini dilakukan secara berulang-ulang, dimulai dari *open list* dan menemukan *node* dengan nilai F paling rendah. Perhitungan persamaan untuk mendapatkan nilai F dijelaskan pada gambar dibawah.

Perhitungan nilai H yaitu nilai estimasi jalur terpendek dari *node* awal ke *node* tujuan atau target dirumuskan dengan:

$$H = 2 * (abs(n.x - tujuan.x) + abs(n.y - tujuan.y)) \quad (3.3)$$

Sebagai contoh *node* yang terletak disebelah kanan dari *node* awal diberikan nilai H=6 yang didapatkan dari perhitungan 6 *node* menuju tujuan dan dikalikan 2, $2*2=6$

Setelah didapatkan nilai F dari setiap *node* yang ada di *open list*, maka akan dipilih *node*, h selanjutnya adalah sebagai berikut:

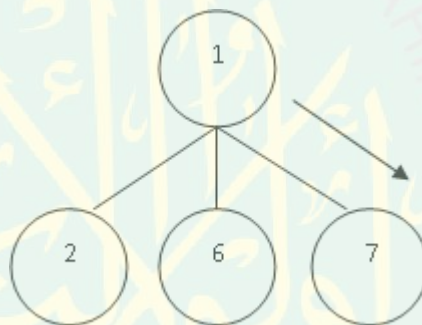
4. Hapus *current node* dari *open list* dan masukan ke dalam *closed list*.
5. Periksa semua *node* yang berdekatan dengan *current node*, abaikan *node* penghalang. Jika tidak ada *open list*, tambahkan kedalam *open list*, lalu tambahkan “*parent*” untuk *node* tersebut.
6. Jika *node* suda ada pada *open list*, lalu bandingkan dengan *node* awal apakah jalur yang di ambil lebih baik dari *node* awal, dengan memeriksa

nilai G-nya apakah lebih rendah jika kita menggunakan *node* ini, jika lebih rendah maka hitung ulang nilai F dan G serta merubah arah *pointer*, jika tidak maka jangan lakukan apa-apa.

Tabel 3.2 Perhitungan h pertama

<i>Open List</i>	2,6,7
<i>Closed List</i>	1
<i>Parent</i>	1
<i>Current Node</i>	7

Dibawah ini adalah penjelasan h pertama algoritma A* dengan menggunakan *tree*.



Gambar 3.23 Penjelasan A* menggunakan *tree*

Proses yang dijelaskan di atas, dilakukan berulang kali atau *looping*, dimulai dari *open list*, dimana *open list* juga berperan sebagai *parent* dan menemukan *node* dengan F yang paling rendah yang selanjutnya akan dijadikan *current* proses yang dijelaskan di atas, dilakukan berulang kali atau *looping*, dimulai dari *open list*, dimana *open list* juga berperan sebagai *parent* dan menemukan *node* dengan nilai F yang paling rendah yang selanjutnya akan dijadikan *current node* (1,1), h-h sebagai berikut:

1. Hapus *current node* dari *open list* atau *parent* lalu masukkan *current node* tersebut ke dalam *closed list*.

2. Periksa semua *node* yang berdekatan dengan *current node*, abaikan penghalang atau unwalked *node*. Jika tidak ada pada *open list*, tambahkan ke dalam *open list* dan tambahkan parent untuk *node* tersebut.
3. Jika *node* sudah ada pada *open list*, periksa *open list* yang mana memiliki nilai F terendah dengan rumus $F=G+H$, *node* dengan nilai F terendah akan Perhitungan pada h kedua yang berada pada *node* dengan koordinat dapat dilihat pada gambar dibawah.

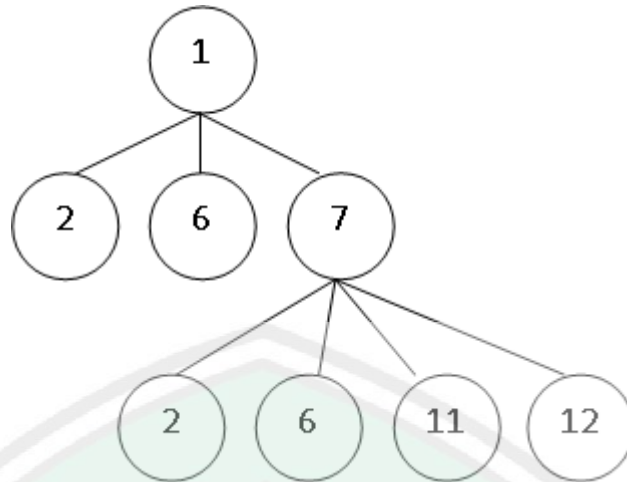
0 8 1 8 NPC	5 6 2 11	3	4	5
5 6 6 11	3 4 7 7 closed list	8	9 Player	10
6 8 11 14	5 6 12 11	13	14	15
8 10 16 18	7 8 17 15	8 6 18 14	19	20

Gambar 3.24 Penjelasan A* pada h kedua

Tabel 3.3 Perhitungan h kedua

<i>Open List</i>	2,6,11,12
<i>Closed List</i>	1,3,7,8,13
<i>Parent</i>	7
<i>Current Node</i>	12

Dibawah ini adalah penjelasan h pertama algoritma AStar dengan menggunakan *tree*.



Gambar 3.25 Penjelasan A* menggunakan tree pada h kedua

Pada h kedua terdapat *node* yang memiliki nilai F yang sama pada kasus seperti ini kita dapat memilih jalur yang mana saja tinggal kita menyesuaikan dengan *node* mana yang mendekat menuju *node* tujuan, pada h kedua ini jalur yang dipilih ialah pada indeks no 7.

Perhitungan h ketiga yang berada pada *node* dengan indeks nomor 12 dapat dilihat pada gambar dibawah.

0	8	5	6			
	1		2	3	4	5
8	NPC	11				
5	6	3	4			
	6		7	8	9	10
11		7 closed list			Player	
6	8	5	6			
	11		12	13	14	15
14		11 closed list				
8	10	7	8	8	4	
	16		17	18	19	20
18		15		12		

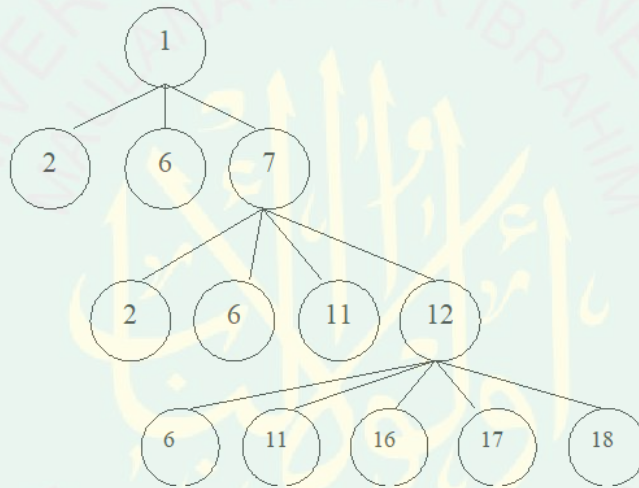
Gambar 3.26 Penjelasan A* pada h ketiga

Tabel dibawah menunjukkan penjelasan tentang pencarian jalur terpendek untuk h ketiga.

Tabel 3.4 Perhitungan h ketiga

Open List	6, 11, 16, 17, 18
Closed List	12,13,8, 7
Parent	12
Current Node	18

Penjelasan melalui *tree* untuk pencarian jalan terpendek pada h ketiga.



Gambar 3.27 Penjelasan A* dengan *tree* pada h ketiga

Perhitungan h keempat yang berada pada *node* dengan indeks 18 dapat dilihat pada gambar dibawah

0	8	5	6		
	1		2	3	5
8	NPC	11			
5	6	3	4		
	6		7	8	10
11		7 closed list		9	Player

↓

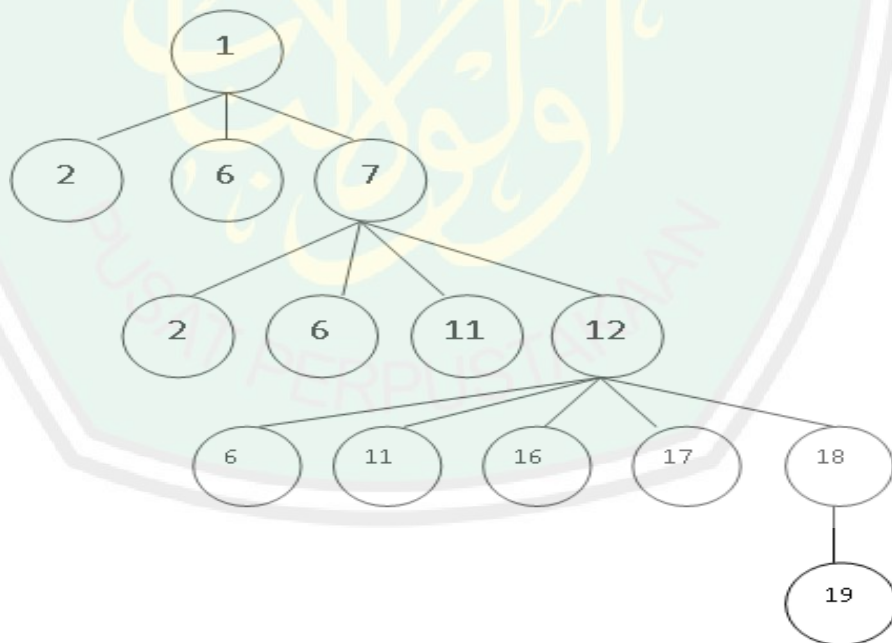
6 11 14	8 5 6 12	13	14	15
8 16 18	10 7 8 17 18	14closed list	8 4 18	10 4 19 14

Gambar 3.28 Penjelasan A* pada h keempat

Tabel 3.5 Perhitungan h keempat

Open List	17, 19
Closed List	12, 13, 14, 18
Parent	18
Current Node	19

Gambar menunjukkan penjelasan melalui *tree* untuk pencarian jalan terpendek pada h keempat.



Gambar 3.29 Penjelasan A* dengan *tree* pada h keempat

Perhitungan pada h kelima yang berada pada *node* dengan indeks 15 yang memiliki F=16 bisa dilihat pada gambar dibawah.



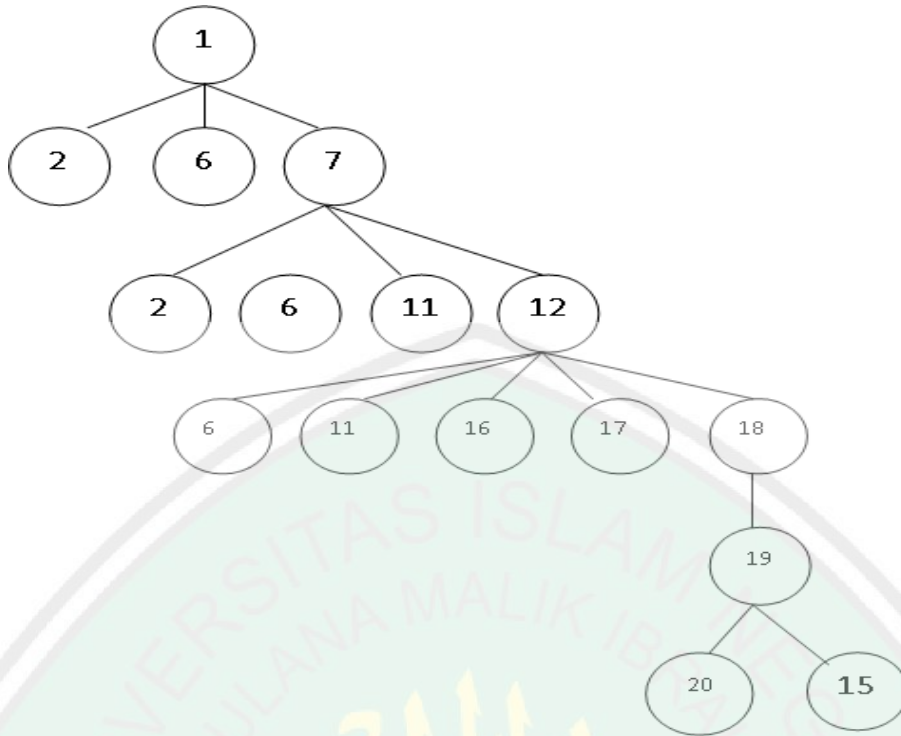
Gambar 3.30 Penjelasan A* pada h kelima

Tabel menjelaskan melalui tabel untuk pencarian jalur terpendek pada h keenam.

Tabel 3.6 Perhitungan h kelima

Open List	15, 20
Closed List	13, 14, 18
Parent	19
Current Node	15

Gambar menunjukkan penjelasan tree untuk pencarian jalan terpendek pada h kelima.



Gambar 3.31 Penjelasan A* dengan tree pada h kelima

Perhitungan pada h keenam yang berada pada node dengan indeks ke 15 yang memiliki nilai F=16 bisa dilihat pada gambar dibawah

0	8	5	6		19	2	19	3
	1		2			4		5
8	NPC	11			21		22	
5	6	3	4		17	0	16	2
	6		7			8		9
11		7 closed list			17	Player	18	10
6	8	5	6				14	2
	11		12			13		14
14		11 closed list			Closed List		Closed List	
8	10	7	8		8	4	11	4
	16		17			18		19
18		15			15 closed list		15	20
							13	4
								17

Gambar 3.32 Penjelasan A* pada h keenam

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi

Bab ini membahas mengenai implementasi dari perencanaan yang telah diajukan. Selain itu pada bab ini dilakukan pengujian terhadap *game* untuk mengetahui apakah *game* tersebut telah berjalan sesuai dengan tujuan penelitian yang ingin dicapai.

4.1.1 Implementasi Sistem

Analisis kebutuhan non fungsional meliputi analisis perangkat keras/*hardware*, analisis perangkat lunak/*software*, dan analisis pengguna/*user*.

a. Analisis Perangkat Keras/*Hardware*

Analisi perangkat keras/ *hardware* yaitu perangkat keras yang mendukung aplikasi agar aplikasi yang dibangun dapat berjalan di *smartphone*. Minimal spesifikasi dari *smartphone* yang dibutuhkan guna mendukung fitur *game virtual reality* yang berjalan adalah sebagai berikut :

1. Memiliki Sensor *Gyroscope*
2. Memiliki Sensor *Accelerometer*
3. Memiliki Sensor *magnetometer* atau sensor kompas
4. Layar *smartphone* minimal 5.0 *inchi*
5. Minimal *android* versi 5.0 (*Lollipop*)

b. Analisis Perangkat Lunak/ *Software*

Perangkat lunak yang dibutuhkan untuk dapat menjalankan *game virtual reality* adalah sistem operasi berbasis *android*. Spesifikasi

minimal adalah *android* versi 5.0 (*Lollipop*) dan maksimal *android* 8.0 (*Oreo*).

c. Analisis Pengguna/ *User*

Aplikasi ini dapat berjalan dengan semestinya jika pengguna memiliki kemampuan dan pengetahuan untuk menjalankannya. Terlebih aplikasi yang dibangun merupakan sebuah aplikasi berupa *game virtual reality* dimana pengguna atau pemain harus mampu mengoperasikan *smartphone*. tidak hanya mampu tetapi juga mengerti tentang bagaimana penggunaan *virtual reality* pada *smartphone* sehingga nantinya dapat memahami maksud dan tujuan dari aplikasi ini. Berikut kriteria yang dibutuhkan untuk dapat menjalankan aplikasi adalah sebagai berikut :

1. Usia : anak-anak usia 5 tahun hingga 15 tahun
2. Keterampilan : Dapat menjalankan *smartphone* dengan sistem operasi *android* dan memahami cara kerja *virtual reality*.

4.2 Implementasi Algoritma A* pada Perilaku NPC

Proses implementasi adalah proses penerapan komponen sistem utama yang dibangun berdasarkan rancangan yang telah dibuat dan diajukan sebelumnya. Implementasi kecerdasan buatan pada penelitian ini diterapkan pada perilaku pencarian pada NPC dengan menggunakan algoritma A* posisi NPC merupakan *starting point* dan target diinisiasi sebagai tujuan.

4.2.1 A* Sebagai Algoritma Perilaku Pencarian

Pada bagian ini membahas tentang penerapan algoritma A* sebagai metode perilaku pencarian NPC untuk menemukan jalur tercepat menuju target dan ketika

NPC telah menemukan target maka selanjutnya akan melakukan serangan terhadap target untuk mengurangi nyawa dari target yang diserang. Pemain bertugas untuk menemukan target untuk diselamatkan dan menjaga target dari serangan *NPC* dengan waktu yang telah ditentukan. Pada penerapannya, algoritma membutuhkan 3 komponen utama untuk menemukan jalur terpendek menuju sasaran. 3 komponen tersebut di implementasikan menggunakan bahasa pemrograman C# dengan masing-masing menggunakan *public class* sendiri. 3 komponen tersebut yakni:

- a. Komponen *Node*
- b. Komponen *Grid*
- c. Komponen *Pathfinding A**

4.2.2 Script Komponen *Pathfinding A**

Komponen *Pathfinding A** adalah komponen utama yang penulis gunakan sebagai *code* untuk membangun perilaku *NPC* untuk mencari jalur terpendek dengan menggunakan algoritma *A**.

Tabel 4.1 Tabel komponen *A** pada Script *Pathfinding*

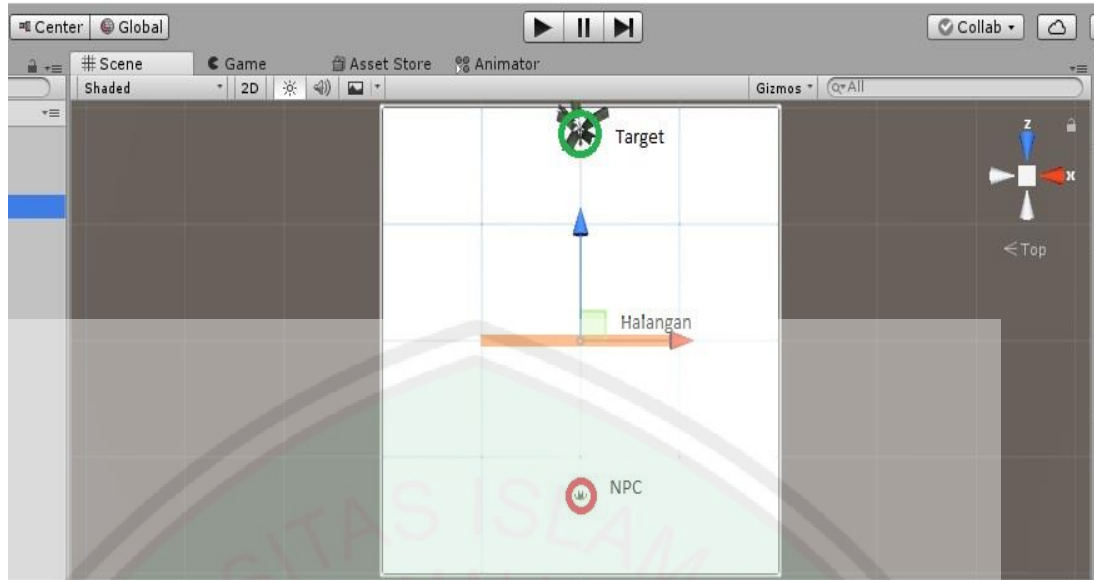
NO	Method	Keterangan
1	<pre>PathRequestManager requestManager; Grid grid; void Awake() { requestManager = GetComponent<PathRequest Manager>(); grid = GetComponent<Grid>(); }</pre>	Memanggil komponen yang dibutuhkan dalam perhitungan algoritma <i>A*</i> .
2	<pre>public void StartFindPath(Vector3 startPos, Vector</pre>	Mulai menghitung

	<pre> 3 targetPos) { StartCoroutine(FindPath(startPos,targetPos)); } </pre>	<p>bobot untuk mencari rute dan menentukan <i>node</i> awal hingga <i>node</i> target.</p>
3	<pre> IEnumerator FindPath(Vector3 startPos, Vector3 targetPos) { Vector3[] waypoints = new Vector3[0]; bool pathSuccess = false; Node startNode = grid.NodeFromWorldPoint(startPos); Node targetNode = grid.NodeFromWorldPoint(targetPos); </pre>	<p>Method yang digunakan untuk mencari rute dan menentukan <i>node</i> awal hingga <i>node</i> tujuan.</p>
4	<pre> if (startNode.walkable && targetNode.walkable) { Heap<Node> openSet = new Heap<Node>(grid.MaxSize); HashSet<Node> closedSet = new HashSet<Node>(); openSet.Add(startNode); </pre>	<p>Cek apakah lokasi awal dan lokasi tujuan dapat dilalui.</p>
5	<pre> Node startNode = grid.NodeFromWorldPoint(startPos); Node targetNode = grid.NodeFromWorldPoint(targetPos); </pre>	<p>Memanggl koordinat <i>node</i> awal dan <i>node</i> tujuan kemudian memasukan ke variabel <i>startNode</i> dan <i>targetNode</i></p>
6	<pre> foreach (Node neighbour in grid.GetNeighbours(currentNode)) { if (!neighbour.walkable closedSet.Contains(neighbour)) { </pre>	<p>Method yang digunakan ketika <i>NPC</i> menemukan <i>targetNode</i></p>

	<pre> continue; } </pre>	
10	<pre> int newMovementCostToNeighbour = currentNode. Gcost + GetDistance(currentNode, neighbour); if (newMovementCostToNeighbour < neighbour.Gcost !openSet.Contains(neighbour)) { neighbour.Gcost = newMovementCo stToNeighbour; neighbour.Hcost = GetDistance(neig hbour, targetNode); neighbour.parent = currentNode; </pre>	Perhitungan untuk menentukan nilai <i>node</i> dan menentukan nilai <i>Gcost</i> dan <i>Hcost</i> dari <i>node</i> yang dihitung.

Setelah ke tiga komponen utama untuk menjalankan algoritma A* yang telah dibuat dalam bentuk code tersebut selesai dilakukan, maka dilakukan implementasi ketiga komponen tersebut ke dalam *game*. berikut urutan implementasi algoritma A* pada pembuatan *game*:

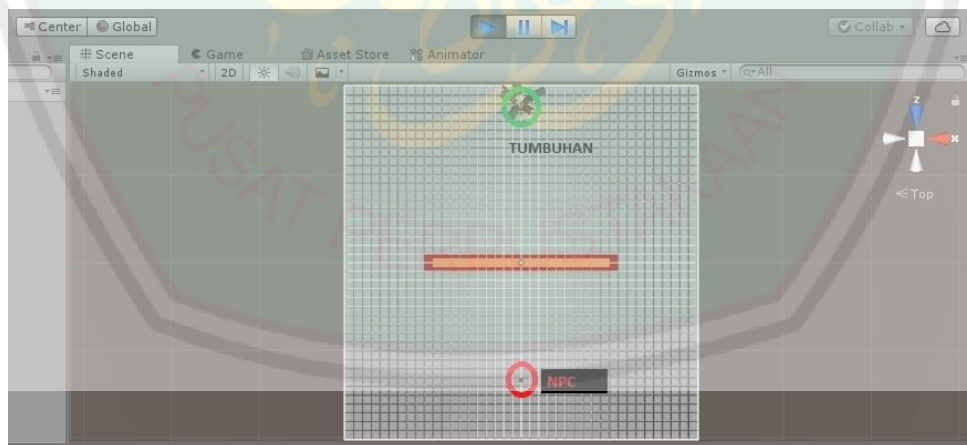
1. Memunculkan objek berupa *NPC* sebagai musuh, kubus sebagai halangan dan tumbuhan sebagai target.



Gambar 4.1 Lokasi target dan lokasi *NPC*

Terdapat 3 objek yakni target, halangan dan *NPC*. Ini adalah tampilan sebelum percobaan dijalankan, yakni tidak terlihat kotak-kotak *grid* serta warna dari penandaan *grizmos*.

2. Meletakkan komponen *grid* untuk mengetahui jalur-jalur yang dapat dilalui oleh *NPC*.



Gambar 4.2 Menampilkan *grid* pada panel

Ketika sudah memasukan *script grid* pada *gameobject* yang dipilih. Lalu diatur ingin berapa sumbu x dan sumbu y beserta ketebalannya. Disini saya menginput masing-masing sumbu x dan y dengan nilai 100 dengan ketebalan 0.5f.

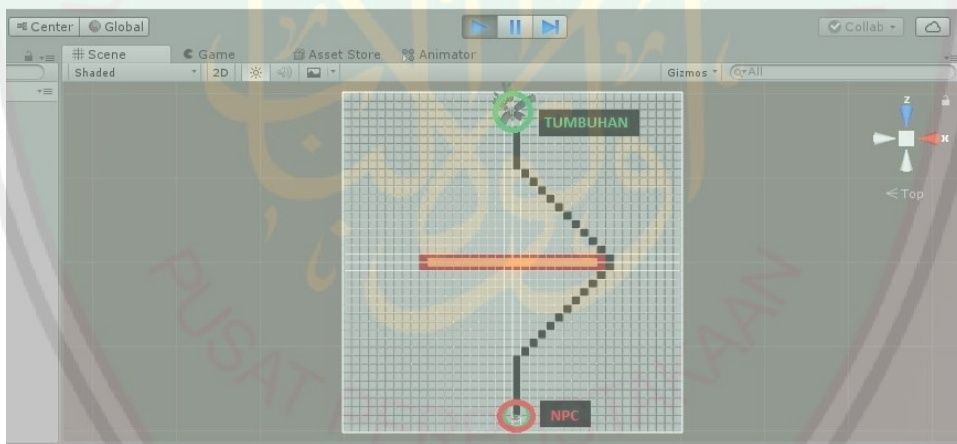
- Meletakkan komponen pathfinding A* pada *gameObject* yang sama dengan *script grid*.



Gambar 4.3 Meletakkan script A* pada *GameObject* yang sama dengan *grid*

Lalu kita masukan *script pathfinding* pada *gameObject* yang sama dengan *script grid*.

- Mengetahui hasil dari algoritma A* dengan jalur-jalur yang harus dilalui oleh *NPC* menuju target dengan jalur terpendek.



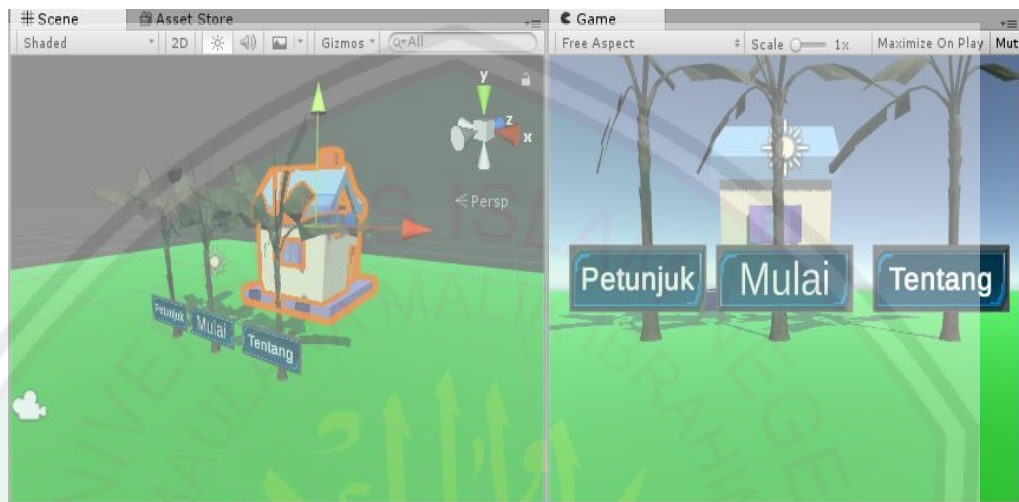
Gambar 4.4 Tampilan jalur path yang akan dilalui *NPC* menuju Target

Setelah itu kita masukan *script* unit untuk menandai target dan *npc* dimana kita dapat mengetahui nilai *currentNode* pada setiap langkah yang dilalui *NPC* menuju target. *CurrentNode* ini berpengaruh untuk menampilkan posisi *node*, *grid* dan nilai *Fcost*, *Hcost* serta *Gcost*.

4.3 Implementasi Aplikasi *Game*

Proses imlementasi adalah proses penerapna komponen sistem utama yang dibangun berdasarkan rancangan yang telah dibuat dan diajukan sebelumnya.

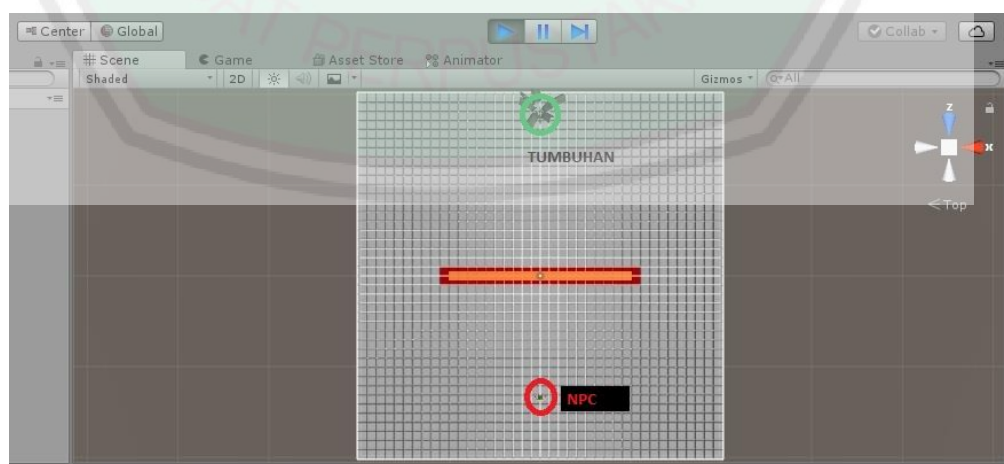
4.3.1 Pengaturan Menu Awal



Gambar 4.5 Membuat menu awal *game*

Pada tampilan menu, terdiri dari 3 *submenu* yakni menu mulai, menu tentang dan menu petunjuk. Untuk memulai kita hanya menunggu waktu 10 detik hingga masuk kedalam permainan, untuk masuk ke dalam menu tentang dan petunjuk harus menekan tombol terlebih dahulu.

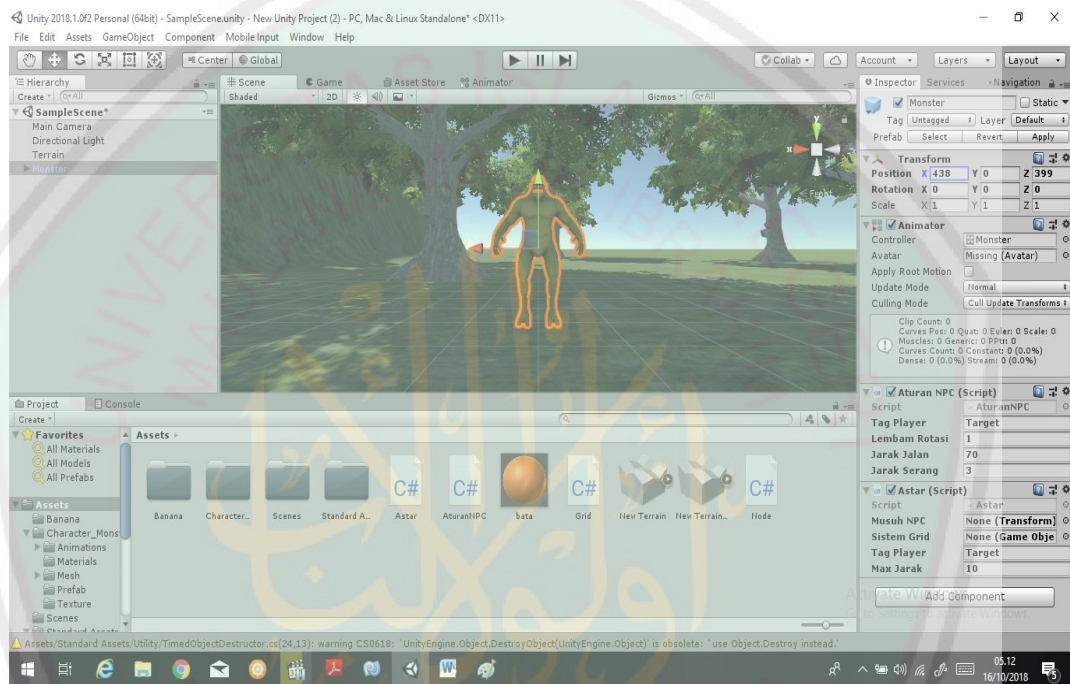
4.3.2 Pembentukan Kotak *Grid*



Gambar 4.6 Membentuk *grid* dan memasukan script A*

Agar algoritma A^* mampu bekerja, maka perlu dibentuk *grid* yakni kotak-kotak yang akan dilalui oleh *NPC* berdasarkan *path* yang telah terbentuk. Disini *grid* bernilai 100 kotak sumbu x dan 100 kotak sumbu y dengan ketebalan 0.5f. Kotak-kotak *grid* yang berwarna putih adalah yang mampu dilalui, lalu kotak-kotak berwarna hitam adalah jalur *path* yang dilalui dan kotak-kotak berwarna merah adalah *unwalkable* atau halangan.

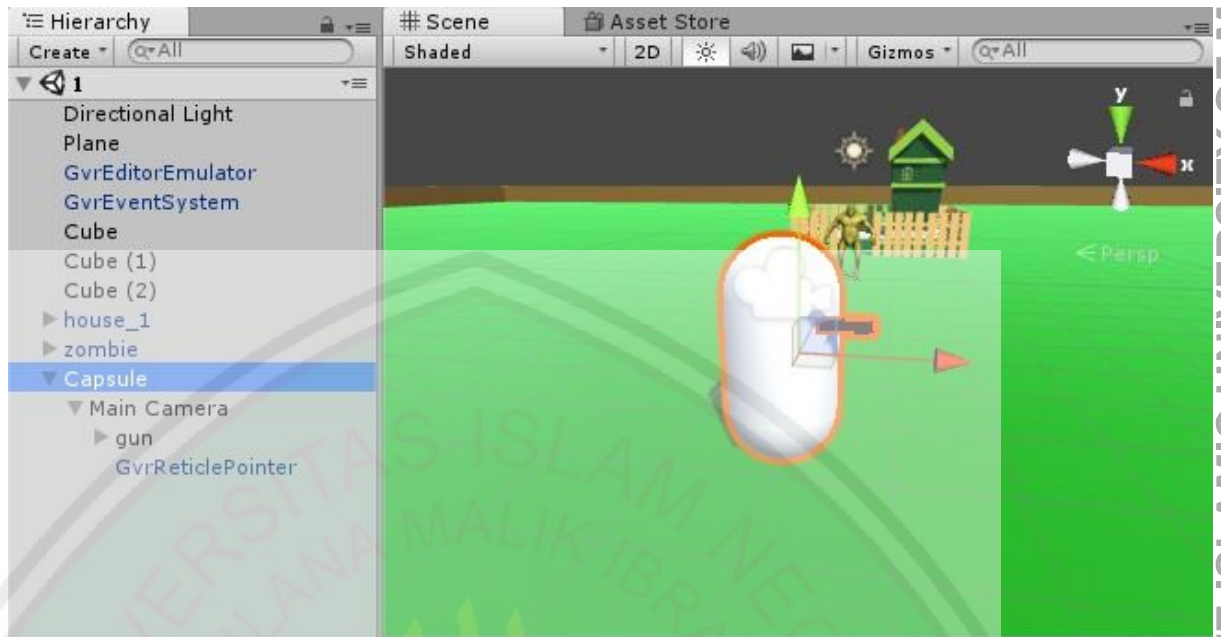
4.3.3 Pengaturan NPC berupa *Enemy* Musuh



Gambar 4.7 Membuat NPC dan memasukan A^*

Bentuk *NPC* berupa *zombie* yang berwarna hijau untuk tipe *NPC* ringan dan berwarna merah untuk tipe *NPC* berat. Pada *NPC*, diberi kemampuan untuk menemukan jalur terpendek menggunakan algoritma A^* serta menyerang tumbuhan. Pada *NPC* juga diberikan beberapa animasi seperti *walk* (berjalan), *attack* (menyerang), *get hit* (diserang) dan *dead* (mati).

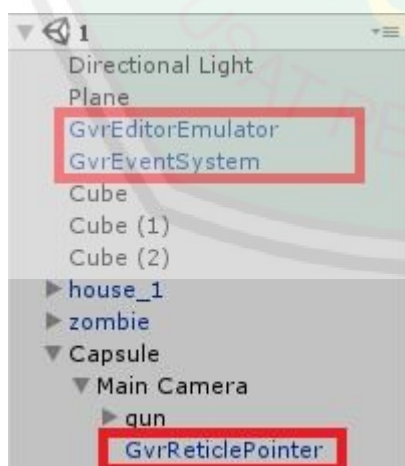
4.3.4 Pengaturan Pemain



Gambar 4.8 Pemain yang berupa *First Person Shooter*

Bentuk pemain sendiri hanya lah sebuah objek berbentuk *capsule* yang diberi kemampuan untuk menembak *NPC* secara otomatis ketika *NPC* berjalan lewat ditengah layar *smartphone*. Pemain ini dibekali oleh senjata berupa senjata jarak jauh

4.3.5 Pengaturan Penggunaan *Virtual Reality* pada *Game*



Gambar 4.9 Memasukan komponen pendukung *Virtual Reality*

Beberapa komponen yang diperlukan agar *virtual reality* dapat berjalan pada smartphone yakni *GvrEditorEmulato*, *GvrEventSystem*, *GvrReticlePointer*.

4.4 Uji Coba

Pada bagian ini membahas tentang uji coba yang telah dilakukan. Terdapat dua uji coba yakni uji coba penggunaan algoritma A* untuk menentukan jalur terdekat terhadap target dan uji coba pada *game*. Berikut pembahasan uji coba tersebut.

4.4.1 Uji Coba Algoritma A*

Uji coba algoritma A* dilakukan untuk melihat hasil dari implementasi algoritma tersebut ke dalam *game* yang telah dibuat menggunakan *Unity*. Gambar adalah tampilan *tab console* pada *unity* yang menampilkan hasil dari perhitungan algoritma A* yang memperlihatkan posisi *node* dan *grid* yang dilalui musuh untuk menuju target.

1. Pengujian pertama :

Posisi Target berdasarkan *Node*: X=-16.6 Z=0.0 Y=-17.4

Posisi Awal Musuh berdasarkan *Node*: X=27.1 Z=0.0 Y=26.0

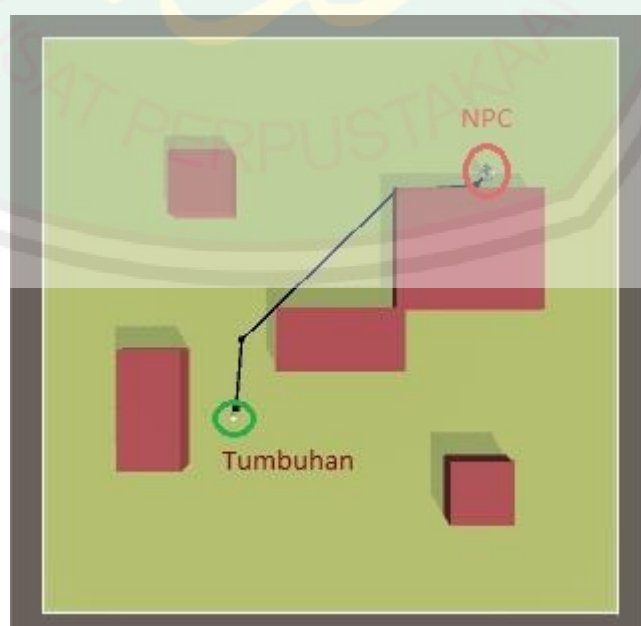
Jumlah Path yang dilalui Musuh = 57

Tabel 4.2 Hasil pengujian A* pertama

No	Grid X	Grid Y	Node X	Node Y	Gcost	Hcost	FCost
1	76	75	26.5	25.5	0	0	0
2	75	74	25.5	24.5	14	588	602
3	74	73	24.5	23.5	28	574	602
4	73	72	23.5	22.5	42	560	602
5	72	72	22.5	22.5	52	556	608
6	71	72	21.5	22.5	62	552	614
7	70	72	20.5	22.5	72	548	620

8	69	72	19.5	22.5	82	544	626
9	68	72	18.5	22.5	92	540	632
10	67	72	17.5	22.5	102	536	638
11	66	72	16.5	22.5	112	532	644
12	65	72	15.5	22.5	122	528	650
13	64	72	14.5	22.5	132	524	656
14	63	72	13.5	22.5	142	520	662
15	62	72	12.5	22.5	152	516	668
16	61	72	11.5	22.5	162	512	674
17	60	72	10.5	22.5	172	508	680
18	59	71	9.5	21.5	186	494	680
19	58	70	8.5	20.5	200	480	680
20	57	69	7.5	19.5	214	466	680
21	56	68	6.5	18.5	228	452	680
22	55	67	5.5	17.5	242	438	680
23	54	66	4.5	16.5	256	424	680
24	53	65	3.5	15.5	270	410	680
25	52	64	2.5	14.5	284	396	680
26	51	63	1.5	13.5	298	382	680
27	50	62	0.5	12.5	312	368	680
28	49	61	-0.5	11.5	326	354	680
29	48	60	-1.5	10.5	340	340	680
30	47	59	-2.5	9.5	354	326	680
31	46	58	-3.5	8.5	368	312	680
32	45	57	-4.5	7.5	382	298	680
33	44	56	-5.5	6.5	396	284	680
34	43	55	-6.5	5.5	410	270	680
35	42	54	-7.5	4.5	424	252	680
36	41	53	-8.5	3.5	438	242	680
37	40	53	-9.5	3.5	448	238	686
38	39	52	-10.5	2.5	462	224	686

39	38	51	-11.5	1.5	476	210	686
40	37	50	-12.5	0.5	490	196	686
41	36	49	-13.5	-0.5	504	182	686
42	35	48	-14.5	-1.5	518	168	686
43	34	47	-15.5	-2.5	532	254	686
44	33	46	-16.5	-3.5	546	140	686
45	33	45	-16.5	-4.5	556	130	686
46	33	44	-16.5	-5.5	566	120	686
47	33	43	-16.5	-6.5	576	110	686
48	33	42	-16.5	-7.5	586	100	686
49	33	41	-16.5	-8.5	596	90	686
50	33	40	-16.5	-9.5	606	80	686
51	33	39	-16.5	-10.5	616	70	686
52	33	38	-16.5	-11.5	626	60	686
53	33	37	-16.5	-12.5	636	50	686
54	33	36	-16.5	-13.5	646	40	686
55	33	35	-16.5	-14.5	656	30	686
56	33	34	-16.5	-15.5	666	20	686
57	33	33	-16.5	-16.5	676	10	686



Gambar 4.10 Letak percobaan kedua NPC dan Target

2. Pengujian kedua :

Posisi Target berdasarkan *Node*: $X=-36.6$ $Z=0.0$ $Y=-27.4$

Posisi Awal Musuh berdasarkan *Node*: $X=36.1$ $Z=0.0$ $Y=40.7$

Jumlah Path yang dilalui Musuh = 86

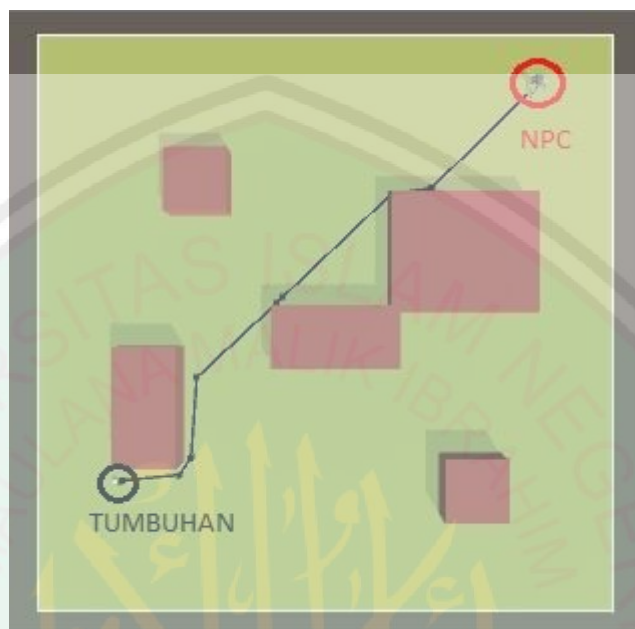
Tabel 4.3 Hasil pengujian A* kedua

No	Grid X	Grid Y	Node X	Node Y	Gcost	Hcost	FCost
1	85.5	90	35.5	40.5	0	0	0
2	84	89	34.5	39.5	14	978	992
3	83	88	33.5	38.5	28	964	992
4	82	87	32.5	37.5	42	950	992
5	81	86	31.5	36.5	56	936	992
6	80	85	30.5	35.5	70	922	992
7	79	84	29.5	34.5	84	908	992
8	78	83	28.5	27.5	98	894	992
9	77	82	27.5	28.5	112	880	992
10	76	81	26.5	29.5	126	866	992
11	75	80	25.5	30.5	140	852	992
12	74	79	24.5	29.5	154	838	992
13	73	78	23.5	28.5	168	824	992
14	72	77	22.5	27.5	182	810	992
15	71	76	21.5	26.5	196	796	992
16	70	75	20.5	25.5	210	782	992
17	69	74	19.5	24.5	224	768	992
18	68	73	18.5	23.5	238	754	992
19	67	72	17.5	22.5	252	740	992
20	66	72	16.5	22.5	262	730	992
21	65	72	15.5	22.5	272	720	992

22	64	72	14.5	22.5	282	710	992
23	63	72	13.5	22.5	292	700	992
24	62	72	12.5	22.5	302	696	998
25	61	72	11.5	22.5	312	692	1004
26	60	72	10.5	22.5	322	688	1010
27	59	71	9.5	21.5	336	674	1010
28	58	70	8.5	20.5	350	660	1010
29	57	69	7.5	19.5	364	646	1010
30	56	68	6.5	18.5	378	632	1010
31	55	67	5.5	17.5	392	618	1010
32	54	66	4.5	16.5	406	604	1010
33	53	65	3.5	15.5	420	590	1010
34	52	64	2.5	14.5	434	576	1010
35	51	63	1.5	13.5	448	562	1010
36	50	62	0.5	12.5	462	548	1010
37	49	61	-0.5	11.5	476	534	1010
38	48	60	-1.5	10.5	490	520	1010
39	47	59	-2.5	9.5	504	506	1010
40	46	58	-3.5	8.5	518	492	1010
41	45	57	-4.5	7.5	532	478	1010
42	44	56	-5.5	6.5	546	464	1010
43	43	55	-6.5	5.5	560	450	1010
44	42	54	-7.5	4.5	574	436	1010
45	41	53	-8.5	3.5	588	422	1010
46	40	53	-9.5	3.5	598	418	1016
47	39	52	-10.5	2.5	612	404	1016
48	38	51	-11.5	1.5	626	390	1016
49	37	50	-12.5	0.5	640	376	1016
50	36	49	-13.5	-0.5	654	362	1016
51	35	48	-14.5	-1.5	668	348	1016
52	34	47	-15.5	-2.5	682	334	1016

53	33	46	-16.5	-3.5	696	320	1016
54	32	45	-17.5	-4.5	710	306	1016
55	31	44	-18.5	-5.5	724	292	1016
56	30	43	-19.5	-6.5	738	278	1016
57	29	42	-20.5	-7.5	752	264	1016
58	28	41	-21.5	-8.5	766	250	1016
59	27	40	-22.5	-9.5	780	236	1016
60	26	39	-23.5	-10.5	794	222	1016
61	26	38	-23.5	-11.5	804	212	1016
62	26	37	-23.5	-12.5	814	202	1016
63	26	36	-23.5	-13.5	824	192	1016
64	26	35	-23.5	-14.5	834	182	1016
65	26	34	-23.5	-15.5	844	178	1022
66	26	33	-23.5	-16.5	854	174	1028
67	26	32	-23.5	-17.5	864	170	1034
68	26	31	-23.5	-18.5	874	166	1040
69	26	30	-23.5	-19.5	884	162	1046
70	26	29	-23.5	-20.5	894	158	1052
71	26	28	-23.5	-21.5	904	154	1058
72	26	27	-23.5	-22.5	914	150	1064
73	26	26	-23.5	-23.5	924	146	1070
74	26	25	-23.5	-24.5	934	142	1076
75	25	24	-24.5	-25.5	948	128	1076
76	24	23	-25.5	-26.5	962	114	1076
77	23	22	-26.5	-27.5	976	100	1076
78	22	22	-27.5	-27.5	986	90	1076
79	21	22	-28.5	-27.5	996	80	1076
80	20	22	-29.5	-27.5	1006	70	1076
81	19	22	-30.5	-27.5	1016	60	1076
82	18	22	-31.5	-27.5	1026	50	1076
83	17	22	-32.5	-27.5	1036	40	1076

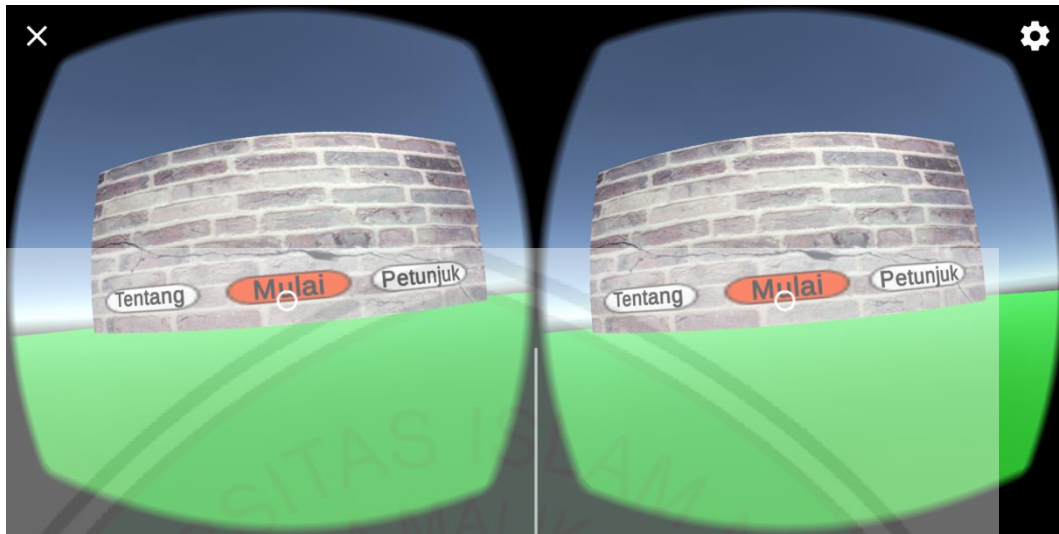
84	16	22	-33.5	-27.5	1046	30	1076
85	15	22	-34.5	-27.5	1056	20	1076
86	14	22	-35.5	-27.5	1066	10	1076



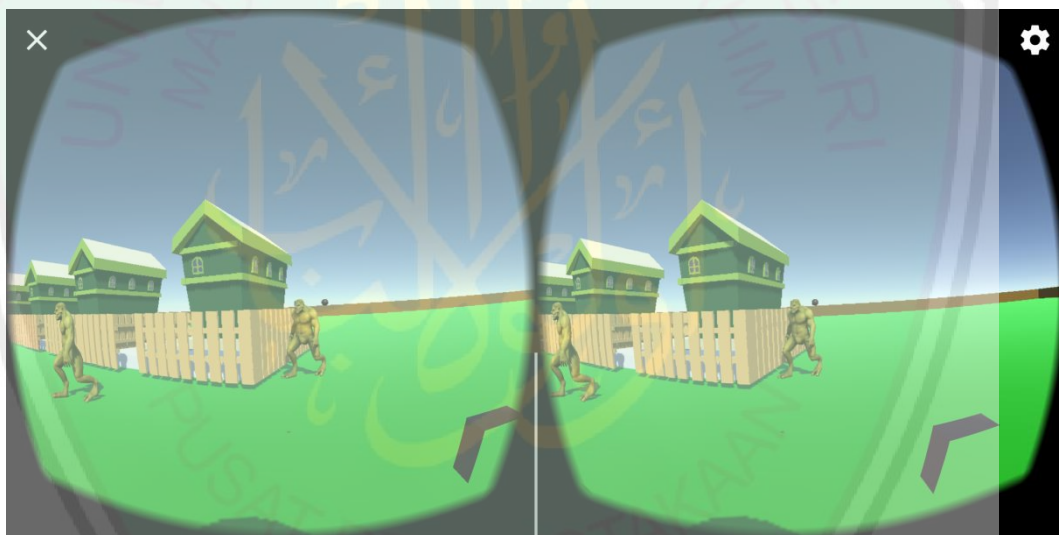
Gambar 4.11 Letak percobaan kedua NPC dan Target

Pada hasil 2 percobaan diatas dapat diambil kesimpulan bahwa nilai $Gcost$ akan mengalami kenaikan nilai, sedangkan nilai pada $Hcost$ akan mengalami penurunan. Dimana perkiraan beban cenderung mengalami kenaikan dari $startnode$ menuju ke $targetnode$, pada estimasi jarak mengalami penurunan nilai. Namun nilai bertambahnya $Gcost$ sama dengan nilai berkurangnya $Hcost$. Sedangkan nilai $Fcost$ mengalami kenaikan walaupun cenderung lebih banyak mengalami nilai yang tetap. Nilai $Fcost$ paling minimum diantara $neighbour$ lah yang diambil oleh NPC untuk dilalui sebagai rute pencarian jalur terpendek.

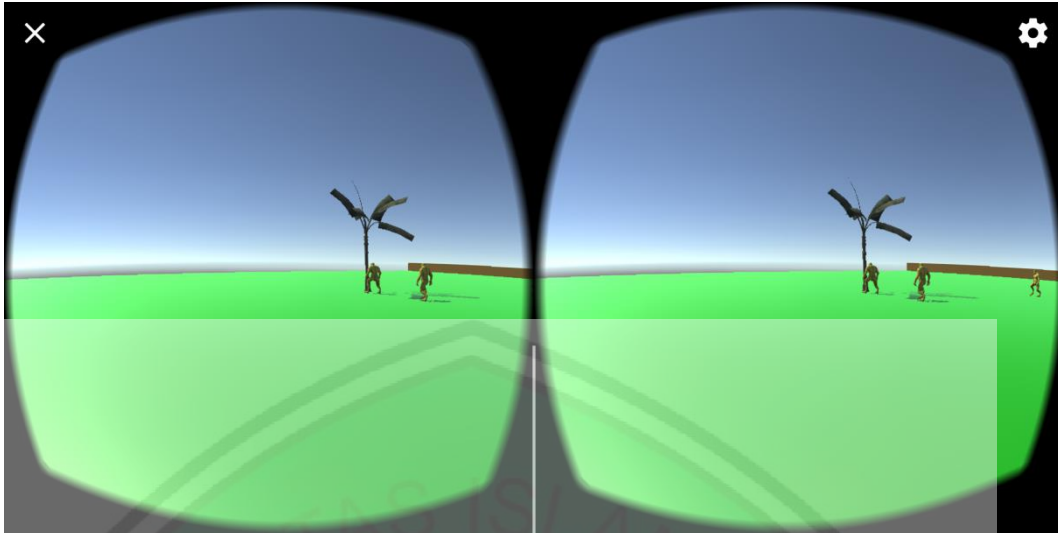
4.4.2 Uji Coba *Game*



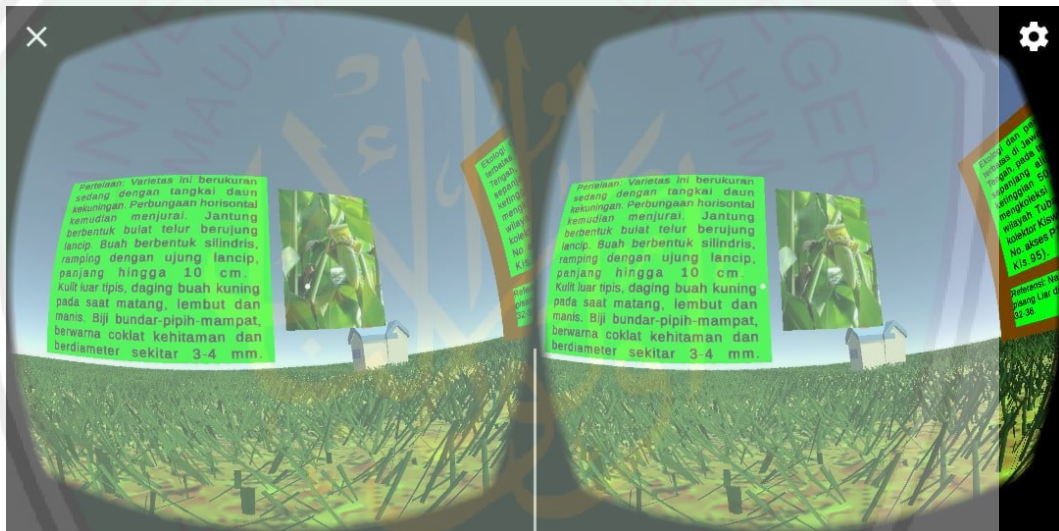
Gambar 4.12 Tampilan Menu



Gambar 4.13 Tampilan pemain menembak NPC



Gambar 4.13 Tampilan *NPC* menyerang tanaman



Gambar 4.14 Tampilan informasi mengenai tumbuhan terkait

Uji coba pada aplikasi game ini dilakukan untuk mengetahui apakah *game* tersebut dapat dimainkan pada *android* dengan spesifikasi yang sudah ditentukan. Berikut adalah hasil dari pengujian yang akan disajikan dalam bentuk tabel.

Tabel 4.4 Tabel percobaan pada *device smartphone*

No	Nama Device	Versi OS	Sensor Gyro	Prosesor	RAM	Layar	Keterangan
1	Oppo	Android		Qualcomm	2 GB	5.0"	Tampilan

	A37	5.1 (Lollipop)		MSM8916 Snapdragon 410 (28 nm)			game berjalan baik, tombol berjalan baik, sistem game berjalan baik
2	Asus Zenfone 4 Selfie	<i>Android</i> 7.0 (Nougat)		Qualcomm MSM8937 Snapdragon 430 (28 nm)	4 GB	5.5"	Tampilan game berjalan baik, tombol berjalan baik, sistem game berjalan baik
3	Asus Zenfone Max Pro (M1)	<i>Android</i> 8.1 (Oreo)		Qualcomm Snapdragon 636 (14 nm)	3 GB	6	Tampilan game berjalan baik, tombol berjalan baik, sistem game berjalan baik
4	Xiaomi Redmi note 5	<i>Android</i> 7.1.1 (Nougat)		Qualcomm Snapdragon 636 (14 nm)	3 GB	6	Tampilan game berjalan

							baik, tombol berjalan baik, sistem game berjalan baik
--	--	--	--	--	--	--	---

Dari pengujian yang dilakukan sebanyak 4 kali, dapat diketahui bahwa game ini telah berhasil berjalan dengan baik pada smarhphone dengan spesifikasi yang ditentukan.

Tabel 4.5 Persentase Uji Coba *Game*

no	Jenis Pengujian	Baik	
		Jumlah	%
1	Sistem	4	$(4/4) \times 100\% = 100\%$
2	Tombol	2	$(2 \times 2) \times 100\% = 100\%$
3	Tampilan	14	$(14 \times 14) \times 100\% = 100\%$

Pada tabel diatas berisi hasil pengujian game dilihat dari 3 aspek yakni sistem game, tombol dan tampilan. Hasil persentase yang didapatkan dari pengujian adalah 100% game berjalan dengan baik pada *device smartphone*.

4.5 Integrasi Islam

Permainan ini terfokus pada pengenalan tanaman yang ada pada kebun raya purwodadi dengan tujuan adanya minat anak-anak untuk mempelajari tanaman dan menanam tanaman.

Para ahli ilmu menyatakan bahwa sedekah jariyah memiliki banyak macam dan jalannya, seperti membuat sumur umum, membangun masjid, membuat jalan atau jembatan, menanam tumbuhan baik berupa pohon, biji-bijian atau tanaman pangan, dan lainnya.

Jadi, menghidupkan lingkungan dengan tanaman yang kita tanam merupakan sedekah dan amal jariyah bagi kita –walau telah meninggal- selama tanaman itu tumbuh atau berketurunan.

Implementasi dari mencintai alam adalah dengan merawatnya bukan merusak alam, seperti yang terjadi akhir-akhir ini. Dalam hal ini didalam islam terdapat penjelasan pada surat Ar-Rum ayat 41.

ظَهَرَ الْفَسَادُ فِي الْبَرِّ وَالْبَحْرِ بِمَا كَسَبَتْ أَيْدِي النَّاسِ لِيُذِيقَهُمْ بَعْضَ الَّذِي عَمِلُوا لَعَلَّهُمْ يَرْجِعُونَ

“Telah nampak kerusakan di darat dan di laut disebabkan karena perbuatan tangan manusia, supaya Allah merasakan kepada mereka sebahagian dari (akibat) perbuatan mereka, agar mereka kembali (ke jalan yang benar)”.

Menurut tafsir Ibnu Katsir adalah telah tampak kerusakan didarat dan dilaut disebabkan oleh tangan manusia. Sesungguhnya kekurangan tanaman pangan dan buah-buahan itu disebabkan oleh kemaksiatan. Abu Aliyah berkata, “barang siapa yang durhaka pada Allah dimuka bumi ini, berarti dia berbuat kerusakan dibumi, hal itu karena kedamaian bumi dan langit adalah dengan ketaatan”. Dan Allah menguji manusia dengan kekurangan kekayaan diri, dan buah-buahan. Ujian ini

merupakan cobaan dan balasan atas perbuatan mereka. Agar mereka kembali kejalan yang lurus.

Dari tafsir diatas kerusakan didarat dan laut terjadi karena ulah tangan-tangan manusia yang tidak bertanggung jawab, eksploitasi secara besar-besaran tanpa memperdulikan efek lingkungan yang terjadi. Eksploitasi hutan dengan cara menebang pohon besar-besaran sehingga banyak jenis tumbuhan yang terancam punah yang juga berdampak terhadap lingkungan itu sendiri.

Rasulullah –Shallallahu alaihi wa sallam– bersabda :

مَا مِنْ مُسْلِمٍ يَغْرِسُ غَرْسًا أَوْ يَزْرَعُ زَرْعًا فَيَأْكُلُ مِنْهُ طَيْرٌ أَوْ إِنْسَانٌ أَوْ بَيْمَةٌ إِلَّا كَانَ لَهُ بِهِ صَدَقَةٌ

“Tak ada seorang muslim yang menanam pohon atau menanam tanaman, lalu burung memakannya atau manusia atau hewan, kecuali ia akan mendapatkan sedekah karenanya”. [HR. Al-Bukhoriy dalam Kitab AL-Muzaro’ah (2320), dan Muslim dalam Kitab Al-Musaqoh (3950)]

Seorang muslim yang menanam tanaman tak akan pernah rugi di sisi Allah –Azza wa Jalla-, sebab tanaman tersebut akan dirasakan manfaatnya oleh manusia dan hewan, bahkan bumi yang kita tempati. Tanaman yang pernah kita tanam lalu diambil oleh siapa saja, baik dengan jalan yang halal, maupun jalan haram, maka kita sebagai penanam tetap mendapatkan pahala, sebab tanaman yang diambil tersebut berubah menjadi sedekah bagi kita.

Al-Qur`an menyebutkan sejumlah buah-buahan yang oleh ilmu pengetahuan modern ditegaskan memiliki khasiat untuk mencegah beberapa jenis penyakit. Buah-buahan yang memberikan manfaat pada tubuh manusia dalam berbagai cara, juga enak rasanya. Di dalam ayat-ayat Al-Qur`an, Allah menyuruh manusia

supaya memperhatikan keberagaman dan keindahan disertai seruan agar merenungkan ciptaan-ciptaan-Nya yang amat menakjubkan.

Tumbuhan merupakan salah satu makhluk hidup ciptaan Allah yang memiliki banyak sekali manfaat. Tumbuh-tumbuhan dapat memunculkan beberapa zat untuk dimanfaatkan oleh makhluk hidup lainnya, misalnya mulai beberapa vitamin-vitamin, minyak dan masih banyak lainnya. Dalam firman-Nya Allah menjelaskan.

وَهُوَ الَّذِي أَنْزَلَ مِنَ السَّمَاءِ مَاءً فَأَخْرَجْنَا بِهِ نَبَاتَ كُلِّ شَيْءٍ فَأَخْرَجْنَا مِنْهُ خَضِرًا مِمَّنْهُ نَخْرُجُ مِنْهُ حَبًّا مَّتَرَاكِبًا وَمِنَ النَّخْلِ مِنْ طَلْعِهَا قِنْوَانٌ دَانِيَةٌ وَجَنَّاتٍ مِنْ أَعْنَابٍ وَالزَّيْتُونَ وَالرُّمَّانَ مُشْتَبِهًا وَغَيْرَ مُنْتَشِبِهِ ۗ أَنْظُرُوا إِلَى ثَمَرِهِ إِذَا أَثْمَرَ وَيَنْعِهِ ۗ إِنَّ فِي ذَلِكَ لَآيَاتٍ لِقَوْمٍ يُؤْمِنُونَ

“Dan Dialah yang menurunkan air hujan dari langit, lalu Kami tumbuhkan dengan air itu segala macam tumbuh-tumbuhan maka Kami keluarkan dari tumbuh-tumbuhan itu tanaman yang menghijau. Kami keluarkan dari tanaman yang menghijau itu butir yang banyak; dan dari mayang korma mengurai tangkai-tangkai yang menjulai, dan kebun-kebun anggur, dan (Kami keluarkan pula) zaitun dan delima yang serupa dan yang tidak serupa. Perhatikanlah buahnya di waktu pohonnya berbuah dan (perhatikan pulalah) kematangannya. Sesungguhnya pada yang demikian itu ada tanda-tanda (kekuasaan Allah) bagi orang-orang yang beriman”.(QS Al-An’am: 99)

Pentingnya mencari ilmu telah dijelaskan pada Al-Quran At-Thaha 114.

فَتَعَالَى اللَّهُ الْمَلِكُ الْحَقُّ وَلَا تَعْجَلْ بِالْقُرْآنِ مِنْ قَبْلِ أَنْ يُقْضَىٰ إِلَيْكَ وَحْيُهُ وَقُلْ رَبِّ زِدْنِي عِلْمًا

Artinya *“Maka Maha Tinggi Allah Raja Yang sebenar-benarnya, dan janganlah kamu tergesa-gesa membaca Al qur'an sebelum disempurnakan mewahyukannya kepadamu, dan katakanlah: "Ya Tuhanku, tambahkanlah kepadaku ilmu pengetahuan”.*

Tafsir menurut Fathul Bari bahwa ilmu lebih berharga daripada emas. Dengan ilmu manusia bisa meraih segalanya. Orang yang berilmu bisa mendapatkan emas, sedang dengan emas manusia belum tentu mendapatkan ilmu. Didalam sebuah hadist, Nabi Muhammad SAW pernah berkata, *“barang siapa yang menginginkan akhirat, maka ia harus memiliki ilmunya, barang siapa yang menginginkan dunia maka ia harus memiliki ilmunya, dan barang siapa yang menginginkan keduanya (akhirat dan dunia), maka ia harus memiliki ilmu keduanya”*. Sebuah hadits dari Imam At-Tirmidzi yang berbunyi *“kelebihan orang yang berilmu dari orang yang beribadah (tanpa ilmu) itu seperti kelebihan saya dari orang yang paling rendah dari para sahabatku.”* Ilmu itu laksana manusia, dia membutuhkan cinta orang yang menuntut ilmu setiap saat selalu ingin bersama yang dicintainya.

Mencari ilmu merupakan hal yang sangat penting bagi umat manusia khususnya umat muslim. Manusia berkewajiban untuk senantiasa belajar segala sesuat terkait hubungan manusia secara vertikal (*hablum minallah*) dan hubungan manusia secara horizontal (*hablum minnas*). Karena dengan menuntut ilmu berguna untuk memperdalam ilmu tentunya akan banyak memberikan kecerahan dalam berfikir, dan semakin tinggi ilmu seseorang dengan begitu dia akan semakin taat kepada Allah SWT. Karena dengan ilmu yang telah dipelajari tersebut memberikan rasa syukur terhadap nikmat dan seluruh kebesaran Allah SWT. Untuk itulah pentingnya menuntut ilmu dan kewajiban untuk mengamalkannya serta menyebarkan ilmu kepada manusia untuk manusia lebih baik.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari implementasi dan pengujian yang dilakukan penulis, maka dapat diambil kesimpulan sebagai berikut :

1. Algoritma A^* dapat diterapkan sebagai perilaku pada *NPC* untuk mencari target dengan rute terpendek. Hal ini ditunjukkan pada tabel . *NPC* mampu melewati segala macam objek yang menghalangi dengan mencari rute lain yang lebih pendek untuk menuju ke target sasaran.
2. Pada Tabel percobaan A^* , nilai *Gcost* mengalami kenaikan sedangkan untuk nilai pada *Hcost* mengalami penurunan. Untuk nilai *Fcost* sendiri cenderung *stagnan* namun juga mengalami kenaikan.
3. Ditinjau dari beberapa percobaan, implementasi A^* pada game tersebut telah didapat tingkat keberhasilan mencapai 100%, dikarenakan *NPC* menemukan jalur menuju target.
4. Ditinjau dari implementasi *device* yang memiliki spesifikasi untuk memenuhi teknologi *virtual reality* yakni diperoleh tingkat keberhasilan 100%.
5. Pembuatan *game* yang menggunakan teknologi *virtual reality* mampu dijalankan di *android* dengan spesifikasi minimal memiliki sensor *gyroscope* dan sistem operasi *Android Lollipop*.

5.2 Saran

Penulis yakin dengan penuh kesadaran bahwa dalam pembuatan permainan ini masih banyak kekurangan yang nantinya sangat perlu untuk

dilakukan pengembangan demi sumbangsih terhadap ilmu pengetahuan, yakni sebagai berikut:

1. Menambah jumlah permainan dan materi yang edukatif sehingga permainan menjadi lebih menarik.
2. Pada *game* ini perlu peningkatan dalam hal gambar, texture, animasi gerakan.
3. Menambah beberapa *NPC* lainnya dengan perilaku yang bervariasi dan menambahkan animasi yang menarik.
4. Permainan ini tidak hanya disajikan di *platform smarthphone android* namun juga dapat digunakan pada *platform smartphone Iphone* dan jenis *platform smarthphone* yang lain.



DAFTAR PUSTAKA

- Koleksi Kebun Raya Purwodadi.* (2008). Dipetik November 21, 2018, dari Kebun Raya Purwodadi: <http://www.krpurwodadi.lipi.go.id/koleksi/>
- Sejarah Kebun Raya Purwodadi.* (2008). Dipetik November 21, 2018, dari Kebun Raya Purwodadi: <http://www.krpurwodadi.lipi.go.id/tentang/sejarah/>
- Tentang Kebun Raya Purwodadi.* (2008). Dipetik November 21, 2018, dari Kebun Raya Purwodadi: <http://www.krpurwodadi.lipi.go.id/tentang/>
- Agung Pamungkas, E. P. (2015). *Penerapan Algoritma A* (A Star) Pada Game Edukasi The Maze Island Berbasis Android.*
- Al'amin, S. G. (2015). *Faktor-Faktor Yang Mempengaruhi Pencapaian Target Penerimaan Negara Bukan Pajak (PNBP) Pada UPT Balai Konservasi Tumbuhan Kebun Raya Purwodadi.* Malang: Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Al'amin, S. G. (2015). *Faktor-Faktor Yang Mempengaruhi Pencapaian Target Penerimaan Negara Bukan Pajak (PNBP) Pada UPT Balai Konservasi Tumbuhan Kebun Raya Purwodadi.* Malang: Fakultas Ekonomi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Arif, Y. M. (2010). *Strategi Menyerang pada Game FPS Menggunakan Hierarchy Finite State Machine dan Logika Fuzzy.* Surabaya: Pasca Sarjana Teknik Elektro ITS.
- Armbruster, C., Kuhlen, T. W., Wolter, M., & Fimm, B. (2008). *Depth Perception in Virtual Reality: Distance Estimations in Peri- and Extrapersonal Space.* CyberPsychology & Behaviour.
- Beck, J. C., & Mitchell, W. (2016). *The Kids are Alright: How the Gamer Generation is Changing the Workplace.* Boston, Massachusetts: Harvard Business School Press.

- Cavalier-Smith, T. (1981). *Eukaryote kingdoms: Seven or nine?* Elsevier Ireland Ltd.
- Deviana, R. (2015). *Implementasi Pathfinding Algoritma A* (A Star) pada Game Petualangan Rahasia*. Batam.
- Gunawan, D. L., Liliana, & Budhi, G. S. (2016). *Pembuatan Kebun Binatang Virtual Untuk Pembelajaran Mengenai Binatang Liar*. Surabaya: Fakultas Teknologi Industri Universitas Kristen Petra Surabaya.
- Hart, P. E., Raphael, B., & Nilsson, N. J. (1972). *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*. ACM SIGART Bulletin 37(37): 28-29.
- Herlangga, K. G. (2016, November). *Virtual Reality dan Perkembangannya*. Dipetik November 21, 2018, dari codepolitan: <https://www.codepolitan.com/virtual-reality-dan-perkembangannya>
- Hermawan, L., & Jawa Bendi, K. (2013). *Penerapan Algoritma A* pada Aplikasi Puzzle*. Palembang: Seminar Nasional Teknologi Informasi dan Komunikasi.
- <http://www.krpurwodadi.lipi.go.id/>. (t.thn.). Dipetik november 21, 2018, dari Kebun Raya Purwodadi: <http://www.krpurwodadi.lipi.go.id/>
- Lewis, L. A., & McCourt, R. M. (2004). *Green Algae and The Origin of Land Plants*. American Journal of Botany.
- Nalwan, A. (1998). *Pemrograman Animasi dan Game Profesional 4*. Jakarta: Elex Media Computindo.
- Riwinoto, & Alfian. (2015). *Implementasi Pathfinding dengan Algoritma A* pada Game Funny English Menggunakan Unity 3D Berbasis Graf*. Batam: Politeknik Negeri Batam.
- Rollings, A., & Ernest, A. (2003). *Andrew Rollings and Ernest Adams on Game Design*. Indianapolis, Indiana: New Riders 201 West 103 Street.

Utomo, Y. W. (2011, 11 30). *Baru 20 Persen Flora yang Teridentifikasi*. Dipetik November 21, 2018, dari Kompas: <https://sains.kompas.com/read/2011/11/30/18112181/Baru.20.Persen.Flora.yang.Teridentifikasi>

Vidiardi, S. (2015). *Pengembangan Museum Virtual Interaktif Menggunakan Teknologi Desktop Virtual Reality Pada Museum Ranggawarsita*. Semarang: Fakultas Teknik Universitas Negeri Semarang.

