

**IMPLEMENTASI ALGORITMA A*(A-STAR) UNTUK MENENTUKAN
RUTE TERPENDEK PADA NPC GAME HEALTHY FOOD**

SKRIPSI

oleh :

**NUR KARTIKA OKTAFIANI
NIM. 16650033**



JURUSAN TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

**UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG**

2020

LEMBAR PENGAJUAN

**IMPLEMENTASI ALGORITMA A*(A-STAR) UNTUK MENENTUKAN RUTE
TERPENDEK PADA GAME HEALTHY FOOD**

SKRIPSI

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

**Oleh:
NUR KARTIKA OKTAFIANI
NIM. 16650033**

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

LEMBAR PERSETUJUAN
IMPLEMENTASI ALGORITMA A*(A-STAR) UNTUK MENENTUKAN RUTE
TERPENDEK PADA GAME HEALTHY FOOD

SKRIPSI

oleh:
NUR KARTIKA OKTAFIANI
NIM 16650033

Telah Diperiksa dan Disetujui
untuk Diuji Tanggal : 15
Januari 2021

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Muhammad. Faisal, M.T
NIP 19740510 200501 1 007

Hani Nurhayati, M.T
NIP 19780625 200801 2 006

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

Dr. CahyoCrysdian
NIP. 19740424 200901 1008

LEMBAR PENGESAHAN
IMPLEMENTASI ALGORITMA A*(A-STAR) UNTUK MENENTUKAN RUTE
TERPENDEK PADA GAME HEALTHY FOOD

SKRIPSI

Oleh:
NUR KARTIKA OKTAFIANI
NIM 16650033

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima sebagai Salah Satu Persyaratan untuk
Memperoleh Gelar Sarjana Komputer (S.Kom) Tanggal: 15
Januari 2021

Susunan DewanPenguji:

Tanda Tangan

PengujiUtama	:	<u>Yunifa Miftachul Arif, M. T</u> NIP. 198306162011011004	()
KetuaPenguji	:	<u>Khadijah Fahmi Hayati Holle, M.Kom</u> NIP. 19900626201608012077	()
SekretarisPenguji	:	<u>Dr. Muhammad Faisal, M.T</u> NIP. 197405102005011007	()
AnggotaPenguji	:	<u>Hani Nurhayati, M.T</u> NIP. 197806252008012006	()

Mengetahui,

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

Dr. CahyoCrysdian
NIP. 19740424 200901 1008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Nur Kartika Oktafiani

Nim 16650033

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul Skripsi : Sistem Implementasi Algoritma A*(A-Star) untuk Menentukan Rute
Terpendek pada *Game Healthy Food*

Menyatakan dengan sebenarnya bahwa penelitian dan penulisan skripsi ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan data, tulisan atau pikiran orang lain yang saya akui menjadi hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan atau referensi pada daftar pustaka. Apabila pada kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan saya.

Malang, 16 Januari 2021
Yang membuat pernyataan


Nur Kartika Oktafiani
Nim 16650033

KATA PENGANTAR

Assamu'alaikum Warohmatullaahi Wabarakaatuh

Segala puji bagi *Ilahi Rabbi* Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya kepada kita semua. Shalawat serta salam semoga tercurah kepada Rasulullah SAW beserta keluarganya. Sehingga penulis mampu menyelesaikan skripsi dengan judul “Implementasi Algoritma A*(A-Star) untuk Menentukan Rute Terpendek pada *Game Healthy Food*”.

Selama proses pengerjaan skripsi ini, penulis mendapatkan banyak dukungan dan bantuan serta bimbingan dari banyak pihak. Oleh karena itu, dengan besar hati penulis ingin mengucapkan terima kasih serta doa yang tulus kepada:

1. Bapak Dr. Muhammad Faisal, M.T, selaku dosen pembimbing I yang telah membimbing, serta memberikan masukan, saran dan juga arahan hingga akhir.
2. Ibu Hani Nurhayati, M.T selaku dosen pembimbing II yang telah dengan teliti membimbing penulis untuk menyelesaikan skripsi.
3. Bapak Yunifa Miftachul Arif, M.T dan Ibu Khadijah Fahmi Hayati Holle, M.T selaku dosen penguji dengan sikap profesional telah menguji seluruh proses ujian sidang skripsi penulis mulai dari seminar proposal hingga sidang skripsi yang berjalan dengan lancar.
4. Seluruh jajaran staf dan dosen jurusan Teknik Informatika secara langsung maupun tidak langsung terlibat dalam proses pengerjaan skripsi.
5. Bapak Dodik Soeprayogi dan Ibu Susiati selaku ayah dan ibu serta keluarga yang selalu memberikan semangat, doa dan dukungan kepada penulis agar selalu berusaha menyelesaikan skripsi tepat waktu.
6. Seluruh sahabat, teman se-angkatan, kakak tingkat, adik tingkat dan seluruh teman-teman seperjuangan baik secara langsung maupun tidak langsung terlibat dalam proses pengerjaan skripsi.

Penulis sangat menyadari bahwa penelitian ini masih jauh dari kata sempurna. Maka dari itu penulis membuka kesempatan selebar-lebarnya untuk setiap saran dan kritik yang membangun. Terlepas dari kekurangan skripsi ini, peneliti berharap ada manfaat yang dapat diambil oleh kita semua. *Aamiin ya rabbal'aalamin.*

Wassalamu'alaikum WarohmatullahiWabarokaatu

Malang, 16 Januari 2021

Penulis



DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGAJUAN.....	ii
LEMBAR PERSETUJUAN.....	iii
LEMBAR PENGESAHAN.....	iv
PERNYATAAN KEASLIAN TULISAN.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
ABSTRAK.....	xi
ABSTRACT	xii
ملخص البحث.....	xiii
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.....	4
BAB II TINJAUAN PUSTAKA	5
2.1 Penelitian Terkait	5
2.2 Landasan Teori.....	8
2.2.1 Algoritma A*	8
2.2.2 Game	10
2.2.3 Unity.....	12
2.2.4 NPC	13
2.2.5 Makanan Sehat	14
2.2.6 Junk Food	14
BAB II ANALISA DAN PERANCANGAN GAME.....	15
3.1 Analisis dan Perancangan Game	15
3.1.1 <i>Storyboard</i>	15
3.1.2 <i>Finite State Machine</i>	19
3.1.3 Blok Diagram	21
3.1.4 Deskripsi Karakter dan Item.....	22
3.1.5 Desain Interface dalam Game	24
3.2 Perancangan Metode	26

BAB IV PENDAHULUAN	31
4.1 Implementasi dan Pengujian	31
4.1.1 Script.....	31
4.1.2 Implementasi Skenario Game	38
4.2 Pengujian	42
4.3 Pembahasan	49
BAB V KESIMPULAN.....	52
5.1 Kesimpulan	52
5.2 Saran	53
DAFTAR PUSTAKA.....	54



DAFTAR GAMBAR

Gambar 3.1 FSM NPC level 1.....	19
Gambar 3.2 FSM NPC level 2.....	20
Gambar 3.3 FSM NPC level 3.....	20
Gambar 3.4 blok diagram dalam game.....	21
Gambar 3.5 NPC dalam <i>game</i>	22
Gambar 3.6 pemain dalam <i>game</i>	22
Gambar 3.7 item dalam <i>game</i>	23
Gambar 3.8 <i>icon</i> dari <i>game</i>	24
Gambar 3.9 tampilan awal dari <i>game</i>	24
Gambar 3.10 tampilan pemilihan karakter.....	25
Gambar 3.11 tampilan lingkungan dalam <i>game</i>	25
Gambar 3.12 Contoh graf Algoritma A*.....	26
Gambar 3.13 <i>open list</i> pertama.....	27
Gambar 3.14 <i>closed list</i>	27
Gambar 3.15 <i>open list</i> dan <i>closed list</i>	28
Gambar 3.16 <i>open list</i> setelah A diperbaharui.....	28
Gambar 3.17 <i>open list</i> setelah dimasukkan simpul H dan D.....	29
Gambar 3.18 pemindahan simpul H ke <i>close list</i>	29
Gambar 3.19 simpul F dan G dimasukkan ke dalam <i>open list</i>	30
Gambar 3.20 jalur terbaik dari simpul S ke E.....	30
Gambar 4.1 Tampilan Menu Utama.....	38
Gambar 4.2 Gambar <i>Scene Story</i>	39
Gambar 4.3 Gambar <i>Scene</i> Pemilihan karakter.....	39
Gambar 4.4 <i>Scene</i> saat <i>game</i> dimulai.....	40
Gambar 4.5 <i>Scene</i> saat tombol menu ditekan.....	40
Gambar 4.6 <i>Scene</i> posisi NPC awal.....	41
Gambar 4.7 <i>Scene</i> saat memenangkan pertandingan.....	41
Gambar 4.8 <i>Scene</i> saat pemain kalah.....	42
Gambar 4.9 Gambar percobaan dengan kondisi <i>map</i> normal.....	42
Gambar 4.10 Gambar percobaan dengan kondisi <i>map</i> 1.25x.....	43
Gambar 4.11 Gambar percobaan dengan kondisi <i>map</i> 1.5x.....	44
Gambar 4.12 Gambar percobaan dengan kondisi <i>map</i> 1.75x.....	44
Gambar 4.13 Gambar percobaan dengan kondisi <i>map</i> 2x.....	45
Gambar 4.14 Gambar percobaan dengan kondisi <i>map</i> normal.....	46
Gambar 4.15 Gambar percobaan dengan kondisi <i>map</i> 1.25x.....	46
Gambar 4.16 Gambar percobaan dengan kondisi <i>map</i> 1.5x.....	47
Gambar 4.17 Gambar percobaan dengan kondisi <i>map</i> 1.75x.....	48
Gambar 4.18 Gambar percobaan dengan kondisi <i>map</i> 2x.....	48
Gambar 4.19 Gambar grafik hasil selisih waktu.....	50

DAFTAR TABEL

Tabel 3.1 <i>Storyboard</i> pada <i>game</i>	16
Tabel 4.1 Tabel <i>Script</i>	32
Tabel 4.2 Tabel Hasil Pengujian.....	49



ABSTRAK

Oktafiani, Nur Kartika. 2021. **IMPLEMENTASI ALGORITMA A*(A-STAR) UNTUK MENENTUKAN RUTE TERPENDEK PADA GAME HEALTHY FOOD**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang

Pembimbing: (I) Dr. Muhammad. Faisal, M.T (II) Hani Nurhayati, M.T

Kata Kunci : Algoritma A*, NPC, Rute, Graf, *Pathfinding*

Dalam upaya mencegah generasi muda dari resiko penyakit akibat terlalu banyak memakan junk food banyak cara yang dapat dilakukan, salah satunya dengan pembuatan game dengan tema makanan sehat yang bernama Healthy Food. Dalam game Healthy Food sendiri terdapat seorang player yang diharuskan untuk memperoleh semua makanan sehat yang berada dalam game. Player tersebut juga harus berhadapan dengan NPC yang menjaga makanan-makanan yang berada dalam game. Pergerakan dari NPC sendiri cukup lambat, maka dari itu dibutuhkan suatu rute terpendek dari NPC sendiri untuk dapat mencapai posisi player. Maka dari itu digunakannya Algoritma A* untuk menentukan rute terpendek dari NPC menuju posisi player. Pemilihan Algoritma A* dikarenakan proses pencarian jalur terpendek untuk membuat musuh mengejar karakter pada permainan ini dapat berjalan lebih baik.

ABSTRACT

Oktafiani, Nur Kartika. 2021. **IMPLEMENTATION OF A*(A-STAR) ALGORITHM TO DETERMINE THE SHORTEST ROUTES IN HEALTHY FOOD GAMES**. Thesis. Informatics Engineering Department of Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang

Supervisor: (I) Dr. Muhammad. Faisal, M.T (II) Hani Nurhayati, M.T

Keyword : *Algorithm A**, *NPC*, *Route*, *Pathfinding*, *Graph*

In an effort to prevent the younger generation from the risk of disease due to eating too much junk food, there are many ways you can do, one of which is by making a game with a healthy food theme called Healthy Food. In the Healthy Food game itself, there is a player who is required to get all the healthy foods in the game. These players also have to deal with NPCs who guard the food in the game. The movement of the NPC itself was quite slow, so it needed the shortest route for the NPC to reach the player's position. Therefore, the A * Algorithm was used to determine the shortest route from the NPC to the player's position. The selection of A * Algorithm is due to the process of finding the shortest path to make the enemy chase the characters in this game to run better.

ملخص البحث

قاسم .اطروجه .الصحي الطعام ألعاب في طريق أقصر ل تحديد مالك مولانا وال تكنولوجيا المعلومات كالمية وماتية الم عمل هندسة الإسلامية مالانغ جامعة إبراهيم

المشرف: (1) د. محمد. فيصل، م. ت (2). زوره يتي هاني

البياني ، الرسم الطريق ، NPC ، A* خوارزمية :الرؤية الكلمات ال
المسارات

بالأمراض الإصابة خطر من الشباب جيل لمنع محاولة في من العديد هناك السرعة الوجدات من الكثير تناول بسب جعل طريق عن هي منها واحدة بها ، القيام يمكن التي الطرق لعبة في .الصحي الغذاء يسمى الصحي الغذاء موضوع مع لعبة علمي للحصول مطلوب هو الذي لاعب هناك نفسها ، الصحي الغذاء NPCs اجهتوا أيضا اللاعب .اللعبة في الصحي الطعام جميع اشخصيات حركة .اللعبة في الطعام علمي يحافظون الذين من طريق أقصر يأخذ فإذنه وباليتالي ، جدا بطة نفسها لاعب موقف إلى الوصول علمي قادرة ل تكون أنفسهم NPCs NPCs من مسار أقصر ل تحديد A* خوارزمية استخدم يتم لذلك ، عملية إلى ذلك جمعوي ر A* الخوارزمية اختيار .لاعب مواقف إلى

اللمعة هذه في الأحرف مطاردة أعداء الجمل مسار أقصر اللمعة
أف ضللت تشغيل يمكن



BAB I

PENDAHULUAN

1.1 Latar Belakang

Makanan merupakan sumber kebutuhan pokok makhluk hidup khususnya manusia. Makanan sebagai sumber energi utama manusia untuk menjalankan aktivitas dalam kehidupan sehari-hari. Dalam sebuah makanan, terdapat kandungan-kandungan yang bermacam-macam seperti karbohidrat, protein, mineral, lemak, vitamin, dan air. Tentunya setiap kandungan dari makanan tersebut memiliki manfaat tersendiri untuk tubuh manusia.

Makanan sendiri dapat dikelompokkan menjadi dua jenis yaitu, makanan sehat dan *junk food*. Makanan sehat merupakan berbagai jenis makanan yang seimbang yang mengandung zat gizi baik mikro maupun makro yang dibutuhkan bagi tubuh manusia. Sedangkan *junk food* merupakan kebalikan dari makanan sehat sendiri. *Junk food* merupakan makanan yang memiliki kandungan gizi rendah tetapi kaya dengan zat yang tidak dibutuhkan oleh tubuh.

Banyak masyarakat terutama remaja lebih memilih *junk food* dibandingkan dengan makanan sehat. Menurut Barasi (2007, hal. 40) dalam

tahun-tahun terakhir ini, anak-anak telah memilih makanan kentang goreng, burger dan hidangan utama tinggi lemak lainnya, kue, serta minuman ringan, hanya sedikit yang memilih buah, sayuran, atau salad. Hal ini tentunya membahayakan kesehatan anak-anak sendiri. Jika anak-anak terlalu sering mengonsumsi *junk food*, maka resiko terserang penyakit juga akan lebih besar. Salah satu dari berbagai macam penyakit yang menghantui anak-anak tersebut adalah obesitas. Menurut Proverawati (2010, hal. 72) secara ilmiah, obesitas terjadi akibat mengonsumsi kalori lebih banyak dari yang diperlukan tubuh. Meskipun penyebab utamanya belum diketahui, namun obesitas pada remaja terlihat cenderung kompleks, multifaktorial dan berperan sebagai pencetus terjadinya penyakit kronis dan degeneratif.

Untuk mencegah generasi muda dari resiko penyakit akibat terlalu banyak memakan *junk food* banyak cara yang dapat dilakukan untuk mempromosikan hidup sehat melalui makanan sehat seperti kampanye kesehatan dalam TV nasional, menggelar acara makan makanan sehat bersama dan juga dapat membuat sebuah permainan dengan tema makanan sehat. Pembuatan *game* dengan tema makanan sehat pada era ini dapat dikatakan merupakan suatu cara yang efektif terutama sasaran dari program ini merupakan anak-anak. Jadi anak-anak tidak hanya bermain *game* saja akan tetapi mereka juga dapat belajar untuk memilih mana makanan yang baik dan makanan yang tidak baik seperti firman Allah Ta'ala :

تَعْبُدُونَ إِيَّاهُ كُنْتُمْ إِنْ لَبَّيْتُمْ وَأَشْكُرُوا رَزَقَكُمْ مَا طَيَّبْتُمْ مِنْ كُلُوا ءَامِنُوا الَّذِينَ يَأْتِيهَا

"Hai orang-orang yang beriman, makanlah di antara rezeki yang baik-baik yang

Kami berikan kepadamu dan bersyukurlah kepada Allah, jika benar-benar kepada-Nya kamu menyembah." (QS Al Baqarah : 172).

Di dalam ayat ini, Allah mengulangi kembali agar memakan makanan yang baik, sebagaimana yang ditegaskan dalam ayat 172. Dengan adanya *game* bertemakan makanan sehat seperti *Healthy Food*, membuat generasi muda dapat lebih menyukai makanan sehat daripada *junk food* agar kesehatannya lebih terjaga dan lebih mengenal makanan mana yang perlu dikurangi agar terhindar dari berbagai macam penyakit.

Dalam *game Healthy Food* sendiri terdapat seorang *player* yang diharuskan untuk memperoleh semua makanan sehat yang berada dalam *game*. *Player* tersebut juga harus berhadapan dengan NPC yang menjaga makanan-makanan yang berada dalam *game*. Pergerakan dari NPC sendiri cukup lambat, maka dari itu dibutuhkan suatu rute terpendek dari NPC sendiri untuk dapat mencapai posisi *player*. Maka dari itu digunakannya Algoritma A* untuk menentukan rute terpendek dari NPC menuju posisi *player*. Selain Algoritma A*, juga dapat menggunakan algoritma lain yaitu Algoritma Dijkstra. Pemilihan Algoritma A* dikarenakan proses pencarian jalur terpendek untuk membuat musuh mengejar karakter pada permainan ini dapat berjalan lebih baik (Hermawan dan Hendi, 2019). Selain itu menurut Prasetyo, et.al (2019), "Algoritma A* dapat menentukan waktu proses pencarian rute terdekat dengan 5 lebih cepat dengan rata-rata waktu 0.37 ms dan langkah 200 daripada Algoritma Dijkstra yaitu 0.41 ms dan langkah 497".

Berdasarkan latar belakang diatas peneliti mengajukan penelitian berupa pembuatan *game* bernama *Healthy Food* yang diharapkan dapat sebagai salah satu

usaha untuk mensosialisasikan makanan sehat khususnya bagi generasi muda lewat media edukatif dan menyenangkan.

1.2 Identifikasi Masalah

Bagaimana pengaruh dari penerapan Algoritma A* pada waktu tempuh NPC ke posisi pemain yang diterapkan *game Healthy Food*?

1.3 Tujuan Penelitian

Untuk mengetahui pengaruh dari penerapan Algoritma A* pada waktu tempuh NPC ke posisi pemain yang diterapkan dalam *game Healthy Food*.

1.4 Manfaat Penelitian

Adapun manfaat yang dapat diambil dari penelitian ini yaitu mengetahui pengaruh dari penerapan Algoritma A* pada waktu tempuh NPC ke posisi pemain dalam *game Healthy Food*.

1.5 Batasan Penelitian

Supaya pembahasan pada penelitian ini tidak melebar, maka diperlukan adanya batasan dalam penelitian. Batasan pada penelitian ini adalah:

1. *Game* yang dibuat berupa *singleplayer*.
2. Lingkungan dalam *game* ini bertemakan hutan dan pegunungan.
3. *Game* dapat dimainkan menggunakan laptop ataupun telepon seluler..
4. Pengaruh algoritma A* berfokus pada waktu tempuh dari posisi NPC ke pemain.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

A. Penerapan Algoritma A* (Star) menggunakan *graph* untuk menghitung jarak terpendek

Menurut Bagus (2018), menyebutkan bahwa (A Star) dalam sains komputer, A* (dibaca “A star”) adalah algoritma komputer yang digunakan secara luas dalam mencari jalur (*path finding*) dan grafik melintang (*graph traversal*), proses *plotting* sebuah jalur melintang secara efisien antara titik-titik, disebut *node*. Terkenal karena penampilannya dan akurasi, algoritma ini diperluas untuk berbagai bidang. Pada *game Healthy Food* sendiri algoritma A* yang digunakan akan lebih cepat menuju kepada *player* dikarenakan terdapat penambahan *script* berupa *character movement* disaat NPC sudah menemukan *player*.

B. Implementasi Algoritma A-Star pada permainan komputer *Roguelike* berbasis *Unity*

Menurut Hermawan dan Hendi (2019) menyebutkan dengan digunakannya algoritma A*, proses pencarian jalur terpendek untuk membuat musuh mengejar karakter pada permainan ini dapat berjalan lebih baik.

C. Penerapan Algoritma A Star (A*) pada *Game* petualangan labirin berbasis Android

Menurut Ahmad dan Widodo (2017), penerapan algoritma A* (A Star) untuk menu bantuan pada *game* petualangan labirin sudah berjalan menggunakan *euclidean heuristic*. Kegunaan algoritma A*(A Star) ini adalah untuk mencari jalur terpendek bagi kelinci (karakter) menuju tempat makanannya.

D. Pencarian jalur terpendek pada *Snake Game* menggunakan Algoritma A*

Menurut Masri dan Ari (2014), menyebutkan bahwa algoritma A* adalah algoritma pencarian terbaik dalam mencari jalur terpendek dengan perhitungan terkecil pada jalur dengan simpul awal menuju simpul akhir. Algoritma A* pada permainan ini teruji sangat efektif dalam mendapatkan makanannya dengan jalur paling pendek.

E. *Implementation of Finding Shortest Route for Pharmacy Location Using A-star Algorithm*

Menurut Juniawan et.al (2019), telah berhasil mengembangkan sistem pencarian khususnya pada pencarian rute terpendek di apotek terdekat yang menggunakan algoritma A*. Dari hasil pengujian pengujian *alpha* menggunakan pengujian *black-box* didapatkan hasil dari sistem ini telah mencapai 100% seperti yang diharapkan. Sedangkan dari pengujian beta dengan penyebaran kuisisioner yang melibatkan 10 responden sudah mendapatkan hasil yang sempurna. Misalnya pada pertanyaan pertama mengenai tampilan sistem dengan desain yang

user friendly dan menarik diperoleh nilai sekitar 84% pada kategori sangat setuju. Selanjutnya hasil pertanyaan ketiga mengenai sistem yang dapat memenuhi dan memperkaya informasi apotek di kota memperoleh nilai sekitar 86% dalam kategori sangat setuju. Dari pertanyaan keempat yang memperoleh 82% kategori “Sangat setuju” terhadap pertanyaan tentang sistem yang memberikan kemudahan dalam proses pencarian apotek terdekat dari lokasi pengguna.

F. Penerapan Metode A* pada *game Mobile Learning* pemilihan sampah organik dan anorganik berbasis Android

Menurut Fuad dan Ahsan (2015), menyebutkan bahwa algoritma A* merupakan algoritma *best first search* yang melakukan traversal satu per satu pada tiap simpul untuk memperoleh lintasan terpendek pada suatu graf. Ketika seluruh lintasan telah selesai dihitung, algoritma A* akan memilih lintasan yang paling pendek. Biaya yang diperhitungkan didapat dari biaya sebenarnya ditambah dengan biaya perkiraan. Dalam notasi seperti ini, algoritma A* adalah *complete* dan optimal.

G. Perbandingan Algoritma A-star dan Dijkstra dalam menentukan rute terdekat

Menurut Prasetyo et.al (2019), menyebutkan bahwa bahwa algoritma A* dapat menentukan waktu proses pencarian rute terdekat dengan lebih cepat dengan rata-rata waktu 0.37 ms dan langkah 200 daripada algoritma Dijkstra yaitu 0.41 ms dan langkah 497.

2.2 Landasan Teori

2.2.1 Algoritma A*

Menurut Arhami et.al (2006), algoritma A* merupakan algoritma *best first search* dengan modifikasi dari fungsi heuristik. Algoritma ini membuat total biaya lintasan lebih sedikit, dan pada kondisi yang tepat akan memberikan solusi yang terbaik dalam waktu yang optimal.

Algoritma A* membutuhkan dua antrian, yaitu *open* dan *closed*. *open* adalah senarai (*list*) yang digunakan untuk menyimpan simpul-simpul yang pernah dipakai dan nilai heuristiknya telah dihitung tetapi belum dipilih sebagai simpul terbaik (*best node*). Dengan kata lain, *open* berisi simpul-simpul yang masih memiliki peluang untuk terpilih sebagai simpul terbaik, sedangkan *closed* adalah senarai untuk menyimpan simpul-simpul yang sudah pernah dibangkitkan dan sudah pernah dipilih sebagai simpul terbaik. Artinya, *closed* berisi simpul-simpul yang tidak mungkin dipilih sebagai simpul terbaik (peluang untuk terpilih sudah tertutup). Selain antrian tersebut, ada juga fungsi heuristik yang memprediksi keuntungan setiap *node* yang dibuat. Hal ini akan memungkinkan algoritma untuk melakukan pencarian-pencarian lintasan yang lebih dapat diharapkan. Fungsi tersebut disebut sebagai pendekatan dari fungsi yang merupakan fungsi evaluasi yang sebenarnya terhadap node n. Dalam banyak penerapan, akan lebih baik jika fungsi ini didefinisikan sebagai kombinasi. Fungsi merupakan ukuran biaya yang dikeluarkan dari keadaan awal sampai ke node n. Nilai yang diperoleh merupakan jumlah biaya penerapan setiap aturan yang dilakukan pada sepanjang lintasan terbaik menuju suatu simpul dan bukan merupakan hasil estimasi. Adapun fungsi merupakan pengukur biaya tambahan

yang harus dikeluarkan dari node n sampai mendapatkan tujuan. Perlu diketahui bahwa nilainya tidak negatif karena bila negatif, maka lintasan yang membalik siklus pada graf akan tampak lebih baik dengan semakin panjangnya lintasan. Secara matematis, fungsi sebagai estimasi fungsi evaluasi terhadap node n dapat dituliskan seperti di bawah ini.

$$f(n) = g(n) + h(n) \quad (\text{Persamaan 2.1})$$

Keterangan:

$f(n)$ = biaya estimasi terendah

$g(n)$ = biaya dari node awal ke node n

$h(n)$ = perkiraan biaya dari node n ke node akhir.

Dalam penerapannya, algoritma A* memiliki beberapa terminologi dasar diantaranya starting point, simpul (*nodes*), A, *open list*, *closed list*, harga (*cost*), halangan (*unwalkable*). Berikut penjelasannya:

- a. Simpul awal adalah sebuah terminologi untuk posisi awal sebuah benda.
- b. A merupakan simpul yang dijalankan dalam algoritma pencarian jalan terpendek
- c. Simpul merupakan petak kecil yang digunakan sebagai representasi dari area yang dicari. Bentuk dari simpul ini bermacam-macam dapat berupa lingkaran, segi panjang, dan juga segitiga.
- d. *Open list* merupakan suatu tempat penyimpanan data simpul yang sedang diproses dari simpul awal .

- e. *Closed List* merupakan tempat penyimpanan data simpul yang sudah diproses.
- f. Harga merupakan nilai yang diperoleh dari penjumlahan jumlah nilai tiap simpul dalam jalur terpendek dari simpul awal ke A dan jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan.
- g. Simpul tujuan merupakan sebuah simpul yang dituju.
- h. Halangan adalah sebuah atribut yang menyatakan bahwa sebuah simpul tidak dapat dilalui oleh posisi sekarang .

2.2.2 Game

Game berasal dari bahasa Inggris yang berarti permainan. *Game* (permainan) merupakan sebuah aktivitas yang bertujuan untuk bersenang-senang, mengisi waktu luang, dan berolahraga ringan. *Game* dapat dilakukan sendiri atau bersama-sama. *Game* adalah sebuah aktifitas yang terstruktur dengan tujuan sebagai hiburan dan juga dapat digunakan sebagai sarana pendidikan. Ciri khas dari sebuah *game* yang membuat hati lebih senang, memberikan motivasi, membuat kecanduan dan membuat banyak orang menggemari aktifitas ini. Peraturan dari setiap game berbeda-beda sehingga membuat jenis game semakin banyak. Hal ini mengacu salah satu fungsi *game* juga sebagai penghilang stress atau rasa jenuh untuk itu setiap orang senang bermain *game* baik anak kecil, remaja maupun dewasa.

Menurut Expro (2010) terdapat genre *game* sebagai berikut:

1. *Action Games*. Jenis *game* yang satu ini adalah *game* yang menggunakan refleks, akurasi, dan waktu yang tepat untuk menyelesaikan sebuah tantangan

dan merupakan genre dasar dari sebuah permainan, dan memiliki mode permainan terbanyak.

2. *Fighting game* adalah jenis *game* bertarung. Seperti dalam *arcade*, pada *game* jenis *fighting* dapat memilih karakter dengan kemampuan yang berbeda-beda dan dapat mengeluarkan kemampuan khusus dalam pertarungannya

3. FPS (*First Person Shooter*) adalah jenis *game* tembak-tembakan dengan tampilan dimana tokoh yang sedang dimainkan cara pandangnya hanya memperlihatkan tangan dan senjatanya saja.

4. TPS (*Third Person Shooter*) adalah *game* yang memiliki ciri khas mirip dengan FPS yaitu memiliki *gameplay* tembak menembak hanya saja dengan sudut pandang yang digunakan dalam *game* ini adalah orang ketiga yang terlihat setengah badan.

5. *Strategy* adalah genre *game* yang memiliki *gameplay* untuk mengatur suatu unit atau pasukan untuk menyerang markas musuh dalam rangka memenangkan permainan.

6. *Tycoon* adalah *game* yang menjadikan pemain sebagai seorang *bussinesman* yang akan mengembangkan sesuatu *property* untuk dikembangkan hingga laku di pasaran.

7. *Racing Game* adalah *game* yang memungkinkan untuk mengendalikan sebuah kendaraan untuk memenangkan sebuah balapan.

8. *Action Adventure* adalah *game* petualangan dimana salah seorang karakter yang penuh dengan aksi yang akan terus ada hingga *game* tersebut tamat.

9. *Arcade game* adalah genre *game* yang tidak terfokus pada cerita, melainkan hanya dimainkan “*just for fun*” atau untuk kejar-mengejar *point/ highscore*.

10. *Sports* adalah genre bertema permainan olahraga. Sistem permainan akan berbeda-beda tergantung jenis olahraga yang menjadi tema *game* tersebut.

12. *Casual* adalah sebuah tipe *game* yang menyuguhkan *gameplay* sederhana sehingga mampu dinikmati oleh banyak kalangan. *Gameplay* yang relatif sederhana juga berimbas pada biaya produksi yang kemudian juga menjadi relatif lebih murah.

2.2.3 Unity

Unity 3D diluncurkan untuk pertama kalinya pada saat acara *Apple's Worldwide Developers Conference* pada tahun 2005. Pada awalnya *Unity* hanya dapat digunakan di *Mac* saja. Namun seiring dengan berkembangnya waktu *Unity 3D* menjadi software multi *platform* yang dapat digunakan pada OS lain seperti *Windows* dan *Linux*.

Unity 3D sendiri dasarnya adalah *game engine* yang basisnya 3D. Tetapi *Unity* juga dapat membuat *game 2D*. *Unity* menggunakan sistem navigasi bebas dalam pembuatan *game*, hal itu memudahkan pengguna untuk melihat setiap sisi 3D dalam objek yang dibuat. Hal ini sama seperti menggunakan *Blender 3D*.

Dalam proses pembuatan, *game developer* dapat menggunakan script-script, seperti: *JavaScript*, *C#*, dan *Boo Script* melalui *panel coding* yang di sediakan dan dapat di *compile* untuk dijalankan pada *console* yang dipilih.

Dalam *Unity*, *developer* bebas berkarya sesuai yang diinginkannya. Tidak hanya terbatas pada satu genre saja, tetapi berbagai genre yang disukai *developer*. Kedepannya *Unity 3D* diharapkan akan menimbulkan banyak *developer-developer game* baru yang mandiri dan berkualitas yang dapat menghasilkan karya-karya yang bisa dinikmati publik tidak hanya sebagai hiburan saja, bahkan bisa memberikan edukasi kepada masyarakat khususnya masyarakat dalam negeri.

2.2.4 NPC

Menurut Kurniawan Teguh Prawono (2015) dalam penelitiannya berjudul Pengembangan *game* dengan menggunakan *Game Engine Game Maker*, NPC (*Non Playable Character*), merupakan karakter dalam *game* yang tidak dapat dimainkan oleh pemain. Karakter ini memiliki kemampuan dengan tambahan AI atau kecerdasan buatan.

Secara garis besar NPC dapat disebut sebagai karakter pada *game* yang dikendalikan oleh komputer dan tidak dapat dimainkan oleh *user*. NPC umumnya dikendalikan menggunakan kecerdasan buatan yang ditujukan 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).

2.2.5 Makanan Sehat

Menurut Prasetyono (2009), makanan sehat adalah dengan meramu berbagai jenis makanan yang seimbang, sehingga terpenuhi seluruh kebutuhan gizi bagi tubuh dan mampu dirasakan secara fisik dan mental. Sedangkan menurut Oetoro (2012), makanan sehat adalah makanan yang kaya nutrisi mengandung zat gizi makro (karbohidrat, protein, dan lemak sehat) serta zat gizi mikro (vitamin dan mineral), tetapi tidak terlalu padat kalori alias tidak melebihi kebutuhan tubuh akan kalori harian.

2.2.6 Junk food

Berbicara perihal gizi, *junk food* merupakan kebalikan dari makanan sehat. Menurut Griffindors (2013), *junk food* merupakan makanan yang lebih mengutamakan cita rasa daripada kandungan gizi. Misalnya keripik kentang yang mengandung garam. Beberapa junk food juga mengandung banyak gula misalnya, minuman bersoda, permen dan kue tar. Gula, terutama gula buatan sangat tidak baik bagi kesehatan tubuh kita karena dapat menyebabkan penyakit diabetes, kerusakan pada gigi kita dan menyebabkan obesitas. Minuman bersoda mengandung paling banyak gula, sementara kebutuhan gula dalam tubuh tidak boleh lebih dari 4 gram atau satu sendok teh sehari.

BAB III

ANALISA DAN PERANCANGAN GAME

3.1 Analisis dan Perancangan Game

Analisa aplikasi ini ditujukan untuk mengetahui masalah yang terjadi pada aplikasi dan nantinya akan menentukan apa saja yang dibutuhkan untuk membuat pembangunan aplikasi lebih mudah. Pada percobaan ini menerapkan algoritma A* dalam *game* untuk mencari rute terpendek.

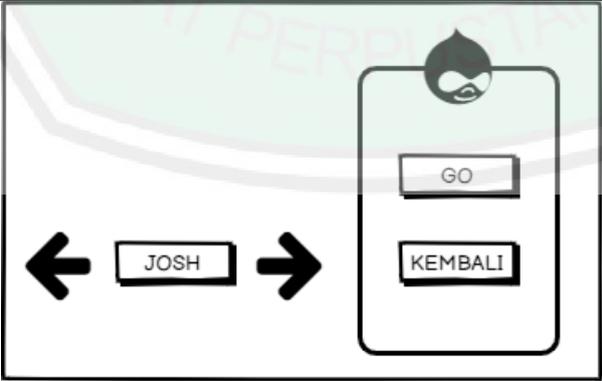
3.1.1 Storyboard

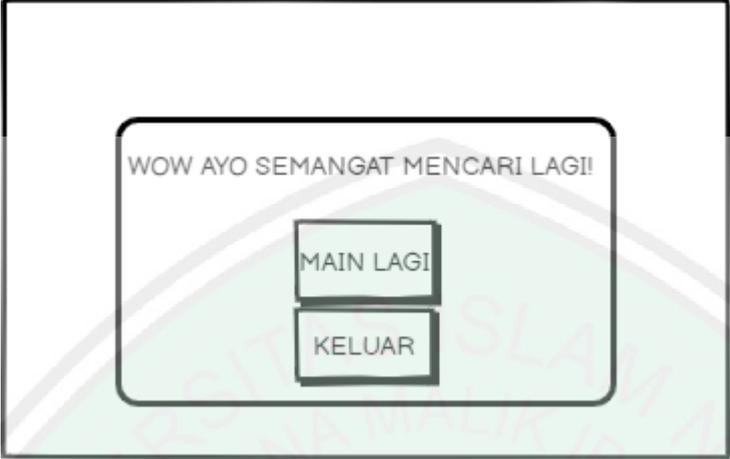
Game ini bernama *Healthy Food* yang terdiri dari 3 level. Level 1 terdapat makanan sehat dan *junk food* beserta sebuah NPC, kemudian *player* harus mendapatkan poin dengan mencari makanan sehat. Level 2 terdapat tiga NPC yang mengganggu *player* dalam petualangan. Level 3 NPC terdapat 5 NPC dan jumlah makanan sehat lebih sedikit.

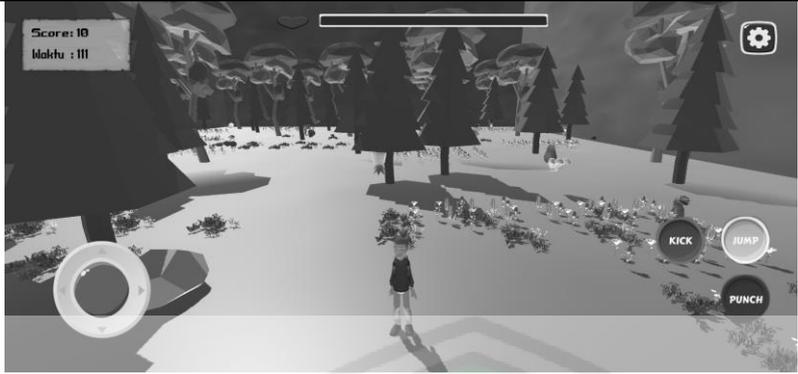
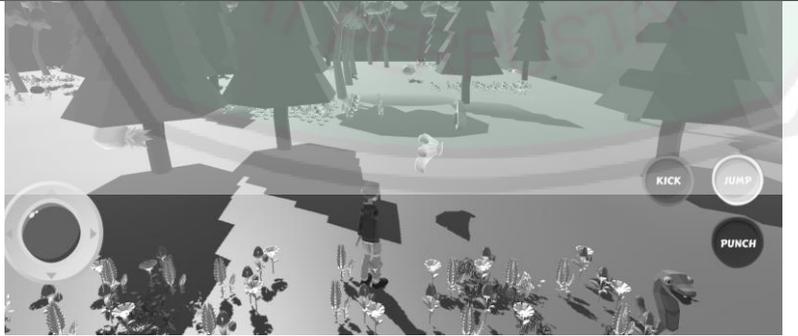
Pada *game* ini, *player* diharuskan mengumpulkan poin sesuai dengan ketentuan agar naik level. Dalam pengumpulan poin *player* diberikan batas waktu beserta monster yang menyerang. *Player* akan kalah ketika monster menyerang terus menerus dan jika *player* kehabisan waktu. *Player* dikatakan menang apabila ketika mampu mencari makanan sesuai dengan batas poin yang ditentukan. Ketika

mendapat makanan sehat poin menambah 20 dan ketika *junk food* poin berkurang 20. *Player* menyerang monster dengan pukulan dan tendangan. Monster mendekati pemain dalam radius yang ditentukan menggunakan algoritma A*.

Tabel 3.1 *Storyboard* pada *game*

Scene	Keterangan
	<p><i>Scene</i> disamping merupakan tampilan dari menu awal pada saat <i>game</i> dimulai. Terdapat 4 tombol yaitu mulai, <i>story</i> dan keluar.</p>
	<p><i>Scene</i> disamping merupakan sebuah <i>scene</i> dimana pemain dapat memilih karakter mana yang akan</p>

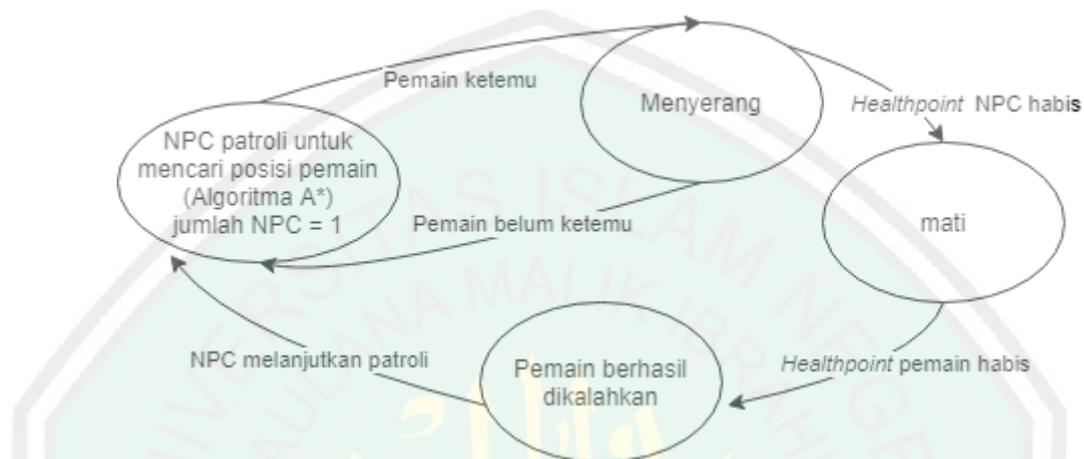
	<p>dipakai dalam permainan.</p>
 <p>WOW AYO SEMANGAT Mencari Lagi!</p> <p>MAIN LAGI</p> <p>KELUAR</p>	<p><i>Scene</i> disamping merupakan sebuah <i>scene</i> pada saat pemain kehabisan waktu atau pemain telah kalah dikarenakan diserang NPC terus menerus.</p>
 <p>WOW SELAMAT! MARI LANJUTKAN!!</p> <p>LANJUT</p> <p>KELUAR</p>	<p><i>Scene</i> disamping merupakan sebuah <i>scene</i> dimana pemain dapat menyelesaikan misi yang ada di dalam <i>game</i></p>

	<p><i>Scene</i> disamping merupakan sebuah <i>scene</i> dimana permainan baru saja dimulai</p>
	<p><i>Scene</i> disamping merupakan sebuah <i>scene</i> dimana pemain akan menghadapi seekor NPC berupa harimau yang dapat menyerang pemain.</p>
	<p><i>Scene</i> disamping merupakan sebuah <i>scene</i> dimana pemain bertugas untuk mencari makanan sehat di</p>

	sekitarnya.
--	-------------

3.1.2 *Finite State Machine*

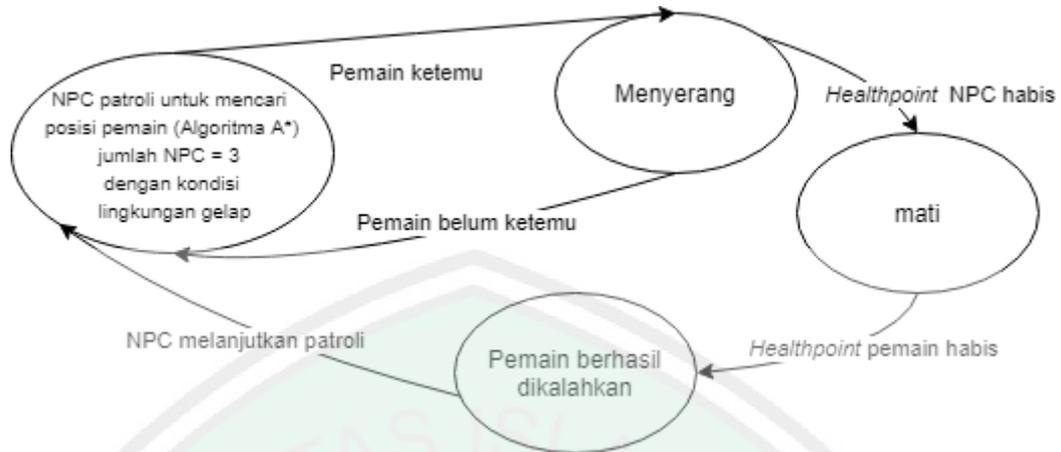
a. *Finite State Machine* pada NPC level 1



Gambar 3.1 FSM NPC level 1

Pada gambar 3.1 (FSM NPC level 1), menjelaskan bahwa NPC akan mencari sebuah target dimana dalam *game* ini targetnya adalah *player*. Pencarian target NPC ini menggunakan algoritma A*. Jika NPC telah menemukan atau berada dekat dengan pemain, NPC akan menyerang pemain dan serangan tersebut dapat mengurangi *health bar* dari pemain hingga menyebabkan pemain kalah dan *game* berakhir. Pada level 1 mempunyai satu NPC saja.

b. *Finite State Machine* pada NPC level 2



Gambar 3.2 FSM NPC level 2

Pada gambar 3.2 (FSM NPC level 2), menjelaskan bahwa NPC akan mencari sebuah target dimana dalam *game* ini targetnya adalah *player*. Pencarian target NPC ini menggunakan algoritma A*. Jika NPC telah menemukan atau berada dekat dengan pemain, NPC akan menyerang pemain dan serangan tersebut dapat mengurangi *health bar* dari pemain hingga menyebabkan pemain kalah dan *game* berakhir. Pada level 2 mempunyai 3 NPC dengan kondisi lingkungan lebih gelap.

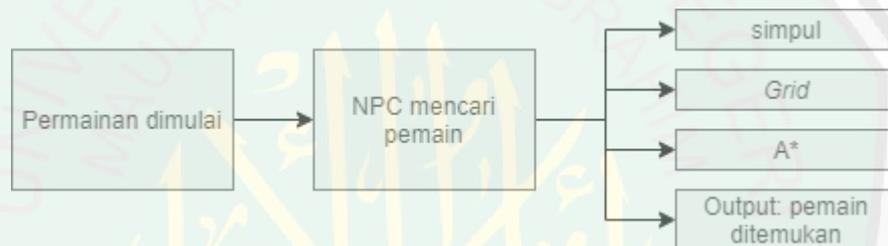
c. *Finite State Machine* pada NPC level 3



Gambar 3.3 FSM NPC level 3

Pada gambar 3.3 (FSM NPC level 3), menjelaskan bahwa NPC akan mencari sebuah target dimana dalam *game* ini targetnya adalah *player*. Pencarian target NPC ini menggunakan algoritma A*. Jika NPC telah menemukan atau berada dekat dengan pemain, NPC akan menyerang pemain dan serangan tersebut dapat mengurangi *health bar* dari pemain hingga menyebabkan pemain kalah dan *game* berakhir. Pada level 3 mempunyai 5 NPC dengan kondisi lingkungan gelap dan jumlah makanan sehat lebih sedikit.

3.1.3 Blok diagram



Gambar 3.4 blok diagram dalam *game*

Pada gambar 3.4 (blok diagram dalam *game*), menjelaskan bagaimana jalan cerita dari algoritma A* yang diterapkan pada NPC. Berikut penjelasan setiap proses dari blok diagram pada gambar 3.4 diatas:

- a. Permainan dimulai : merupakan bagian awal dimana sistem akan berjalan nantinya.
- b. NPC mencari pemain : merupakan sebuah proses dimana diterapkannya algoritma A* untuk mencari rute terpendek bagi NPC menemukan pemain.
- c. Simpul : merupakan sebuah petak atau titik sebagai representasi area dari NPC ke pemain.
- d. *Grid* : sebuah area yang sudah ditandai sebagai mana jalur yang dapat dilewati dan mana jalur yang lebih sulit dilewati.

- e. A* : sebuah proses yang ditujukan untuk menghitung *pathfinding* yang bertujuan untuk menemukan jalur terpendek menuju ke pemain.
- f. *Output* : tujuan dari perhitungan algoritma A* itu sendiri berupa NPC yang menemukan posisi target (pemain) dan menyerang.

3.1.4 Deskripsi karakter dan item

a. NPC



Gambar 3.5 NPC dalam game

Dalam game *Healthy Food*, NPC yang digunakan berupa harimau. Harimau dalam game ini memiliki sebuah *bar* yang menunjukkan nyawa dari binatang tersebut. Pemain dapat mengalahkan harimau ini dengan terus melakukan *hit* pada NPC.

b. Pemain



Gambar 3.6 pemain dalam game

Untuk aset pemain, dalam *game* ini menggunakan *free asset* dari *Unity Asset Store* sendiri. Aset pemain kali ini ada 2, yaitu Josh dan Vyn. Keduanya berkarakter sebagai seorang anak-anak dengan memakai jaket hijau.

c. Item



Gambar 3.7 item dalam *game*

Dalam *game* ini terdapat dua macam item yaitu berupa makanan sehat dan *junk food*. Makanan sehat dalam *game* ini berupa buah dan sayur-sayuran. Sedangkan *junk food* dalam *game* ini berupa *burger*, *hot dog* dan *french fries*.

3.1.5 Desain *Interface* dalam *game*

A. Desain icon



Gambar 3.8 *icon* dari *game*

Dalam *game* ini, mempunyai logo berupa gabungan antara wortel dan labu diikuti teks “*Healthy Food*”. Hal ini merepresentasikan makanan sehat itu sendiri jika orang sekilas melihat logonya saja.

B. Desain menu utama



Gambar 3.9 tampilan awal dari *game*

Gambar 3.9 (tampilan awal dari *game*), terdapat 3 tombol yaitu tombol mulai, *story* dan keluar. Selain itu juga terdapat logo dan karakter awal beserta sebagian lingkungan.

C. Desain pemilihan karakter



Gambar 3.10 tampilan pemilihan karakter

Gambar 3.10 (tampilan pemilihan karakter), terdapat dua tombol yaitu tombol *Go* dan *kembali*. Selain itu juga terdapat logo dan dua karakter yang dapat dipilih yaitu Josh dan Vyn.

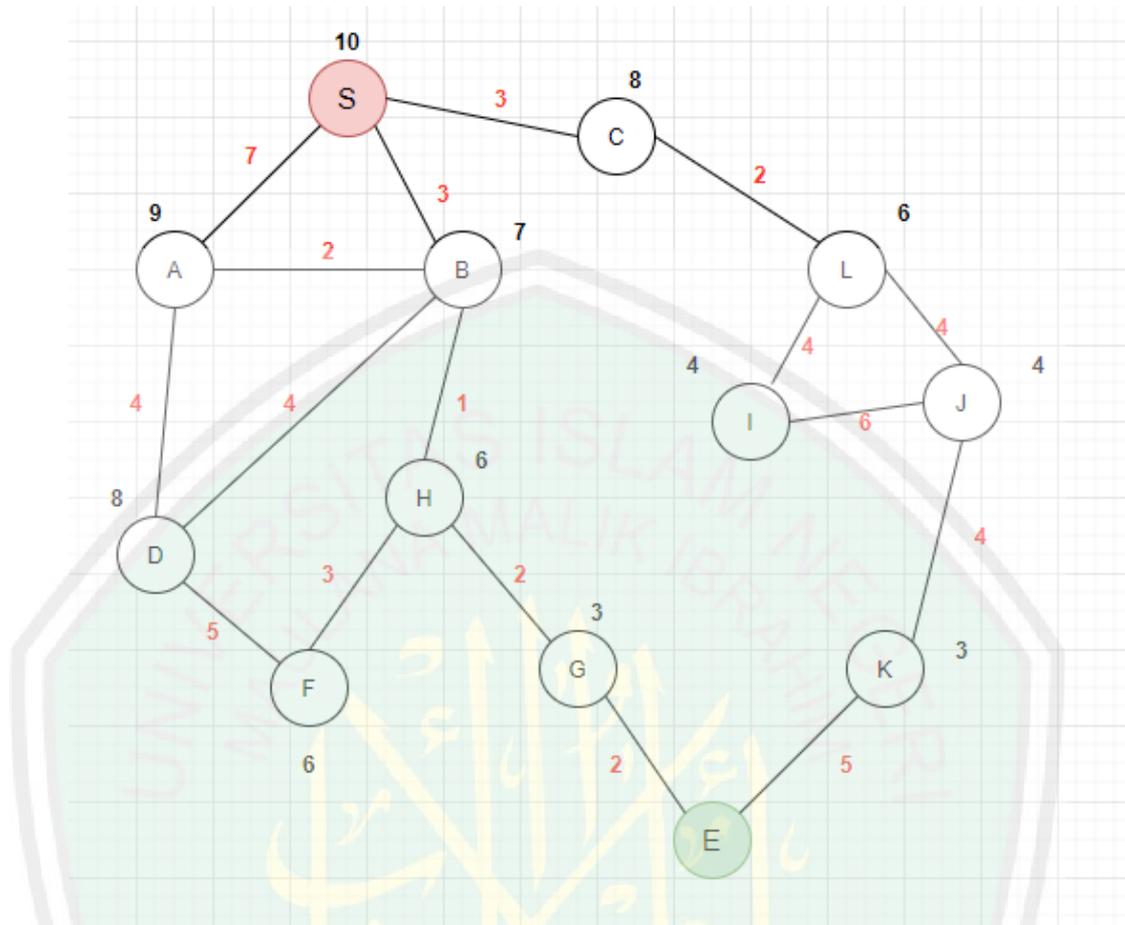
D. Desain latar permainan



Gambar 3.11 tampilan lingkungan dalam game

Gambar 3.11 (tampilan pemilihan karakter), terdapat banyak aset seperti bunga, pohon, rumput, item dan juga NPC.

3.2 Perancangan Metode



Gambar 3.12 Graf Rute dalam permainan

Pada gambar 3.12 (graf rute dalam permainan), dimana terdapat 13 simpul dengan nilai dan jarak yang sudah diberikan. Untuk menghitung jarak terpendek yang harus diambil dari titik S (NPC) ke E (pemain) menggunakan cara berikut.

- a) Dari simpul S dapat menuju simpul A, B dan C. Maka dari itu simpul A dan B dimasukkan ke dalam *open list*. Jarak dari S ke A adalah 7, sedangkan nilai A 9 jadi harganya adalah 16. Jarak dari S ke B adalah 3, sedangkan nilai B adalah 7, jadi harganya adalah 10. Total harga dari simpul C adalah 11.

OPEN LIST

TUJUAN	HARGA	VIA
B	9	S
C	11	S
A	16	S

Gambar 3.13 *open list* pertama

- b) Karena simpul S sudah dihitung, maka simpul S dimasukkan ke dalam *closed list*.

CLOSED LIST

TUJUAN	HARGA	VIA
S	10	10

Gambar 3.14 *closed list*

- c) Simpul B merupakan simpul lanjutan yang harus dikerjakan karena pada *open list* simpul B memiliki harga terendah yaitu 9. Titik B dipindah terlebih dahulu ke dalam *closed list*.

OPEN LIST

TUJUAN	HARGA	VIA
C	11	S
A	16	S

CLOSED LIST

TUJUAN	HARGA	VIA
S	10	10
B	9	S

Gambar 3.15 *Open list* dan *closed list*

- d) Simpul B dapat menuju ke titik A, D dan H. Simpul S, B dan A memiliki harga 14. Harga ini lebih murah dari harga simpul S ke A. Jadi dalam *open list* harus memperbaharui harga tujuan A menjadi 14.

OPEN LIST

TUJUAN	HARGA	VIA
C	11	S
A	14	S

Gambar 3.16 *Open list* setelah A diperbaharui

- e) Simpul S, B dan D memiliki harga 14. Sedangkan simpul S, B dan H harganya 9. Keduanya dimasukkan ke dalam *open list*.

OPEN LIST

TUJUAN	HARGA	VIA
H	9	B
C	11	S
A	14	S
D	14	B

Gambar 3.17 *Open list* setelah dimasukkan simpul H dan D

- f) Simpul H memiliki harga terendah, maka dari itu simpul H dipindahkan ke dalam *close list*.

OPEN LIST

TUJUAN	HARGA	VIA
C	11	S
A	14	S
D	14	B

CLOSED LIST

TUJUAN	HARGA	VIA
S	10	10
B	9	S
H	9	B

Gambar 3.18 Pemindahan simpul H ke *close list*

- g) Simpul S, B, H dan F memiliki harga 12. Sedangkan simpul S, B, H dan G memiliki harga 8. Keduanya dimasukkan ke dalam *open list*.

OPEN LIST

TUJUAN	HARGA	VIA
G	8	H
C	11	S
F	12	H
A	14	S
D	14	B

Gambar 3.19 simpul F dan G dimasukkan ke dalam *open list*

- h) Karena simpul G merupakan simpul terkecil, maka simpul G menjadi simpul yang akan diproses selanjutnya. Karena simpul G hanya memiliki satu simpul tujuan yaitu simpul E yang merupakan tujuan utama, maka jalur terbaik dari graf ini sudah didapatkan yaitu S-B-H-G-E dengan total harga 7.

CLOSED LIST

TUJUAN	HARGA	VIA
S	10	10
B	9	S
H	9	B
G	8	H

Gambar 3.20 Jalur terbaik dari simpul S ke E

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi algoritma A*(A-Star)

Implementasi merupakan sebuah proses dimana dilakukannya penerapan komponen dari sistem yang akan dibangun serta mengacu pada rancangan yang telah dibuat sebelumnya. Pada penelitian ini mengimplementasikan algoritma A*(A-Star) kepada NPC untuk menentukan rute terpendek.

4.1.1 Script

Script merupakan salah satu komponen utama dari beberapa komponen yang digunakan untuk mengimplementasikan suatu program. Dalam *script* sendiri terdiri dari baris-baris kode yang terhubung satu sama lain. Dalam pembuatan *game* ini penulis menggunakan *Visual Studio Code* sebagai *code editor*. Beberapa *script* yang diimplementasikan pada *game Healthy Food* seperti pada tabel 4.1 (tabel *script*).

Tabel 4.1 Tabel Script

NO	Script	Keterangan
1	<pre> Private void OnTriggerEnter(Collider other){ switch(other.gameObject.tag){ Case "cointt"; TimeLeftScript.timeleft +=5f; other.gameObject.SetActive(false); break; SetScoreText();}} Void SetScoreText(){ ScoreText.text="Score: "+ points;} </pre>	<p><i>Script</i></p> <p>disamping digunakan disaat pemain sudah makan makanan dan <i>healthpoint</i> pemain akan bertambah.</p>
2	<pre> public Text countdownText; void Start () { StartCoroutine("LoseTime");} void Update () { countdownText.text =("Waktu : "+timeLeft); if(timeLeft <= 0){ StopCoroutine("LoseTime"); countdownText.text="Waktu Habis"; gameOver(); } else { Resume(); } void gameOver(){ menuContainer1.SetActive(true); Pause(); } </pre>	<p><i>Script</i></p> <p>disamping digunakan untuk mengatur waktu jika waktu yang ditentukan selesai</p>

	<pre>IEnumerator LoseTime() { while(true){ yield return new WaitForSeconds(1); timeLeft--; } }</pre>	
3	<pre>private void OnTriggerEnter (Collider other){ if(other.name =="Player"){ other.GetComponent <playerpoint>().points+=- 20; switch (other.gameObject.tag){ case "coinnt": TimeLeftScript.timeLeft +=15f; other.gameObject.SetActive(false); break; } Destroy(gameObject); } }</pre>	<p><i>Script</i></p> <p>disamping</p> <p>digunakan</p> <p>disaat pemain</p> <p>memakan</p> <p>makanan <i>junk</i></p> <p><i>food</i> dan akan</p> <p>mengurangi</p> <p><i>healthpoint</i></p> <p>pemain.</p>
4	<pre>private void OnTriggerEnter (Collider other){ if (other.name =="Player"){ other.GetComponent <playerpoint>().points+= 10; switch (other.gameObject.tag){ case "coinnt": TimeLeftScript.timeLeft +=15f; other.gameObject.SetActive(false);</pre>	<p><i>Script</i></p> <p>disamping</p> <p>digunakan</p> <p>disaat pemain</p> <p>memakan</p> <p>makanan bagus</p> <p>dan akan</p>

	<pre>break; } Destroy(gameObject); } }</pre>	menambah poin dari pemain.
5	<pre>void Update () { Vector3 direction = player.position- this. transform.position; float angle = Vector3.Angle(direction, this. transform.forward); if (Vector3.Distance (player.position,this. transform.position) <10 && angle < 70) { direction.y = 0; this.transform.rotation = Quaternion.Slerp (this.transform. rotation, Quaternion. LookRotation (direction), 0.05f); anim.SetBool ("isIdle", false); } }</pre>	<p><i>Script</i></p> <p>disamping digunakan saat NPC menuju ke posisi pemain tanpa menggunakan algoritma A*</p>
6	<pre>public class floortrigger : MonoBehaviour { public bool isDamaging; public float damage = 10; private void OnTriggerStay(Collider col){ if(col.tag == "Player") col.SendMessage ((isDamaging)?"TakeDamage": "HealDamage",Time.deltaTime * damage); } }</pre>	<p><i>Script</i></p> <p>disamping digunakan saat pemain diserang NPC dan <i>healthpoint</i> berkurang.</p>
7	<pre>private void UpdateHealthbar(){</pre>	<i>Script</i> yang

	<pre>float ratio= hitpoint/maxhitpoint; healthbar.rectTransform.localScale = new Vector3(ratio,1,1); ratiotext.text = (ratio*100).ToString("0") + ' '; } private void TakeDamage(float damage){ hitpoint -= damage; if(hitpoint<0){ hitpoint=0; Debug.Log ("Dead!"); gameOver(); } UpdateHealthbar(); }</pre>	<p>digunakan jika NPC diserang pemain dan <i>healthpoint</i> NPC akan berkurang.</p>
8	<pre>public void waktuberhenti(){ if(PanelTujuan==true){ if(GameIsPaused){ Resume(); } else { Pause(); } } } public void Resume(){ PanelTujuan.SetActive (false); Time.timeScale=1f; GameIsPaused=false;</pre>	<p><i>Script</i> disamping digunakan saat menu <i>pause</i> ditekan saat <i>game</i> dimulai.</p>

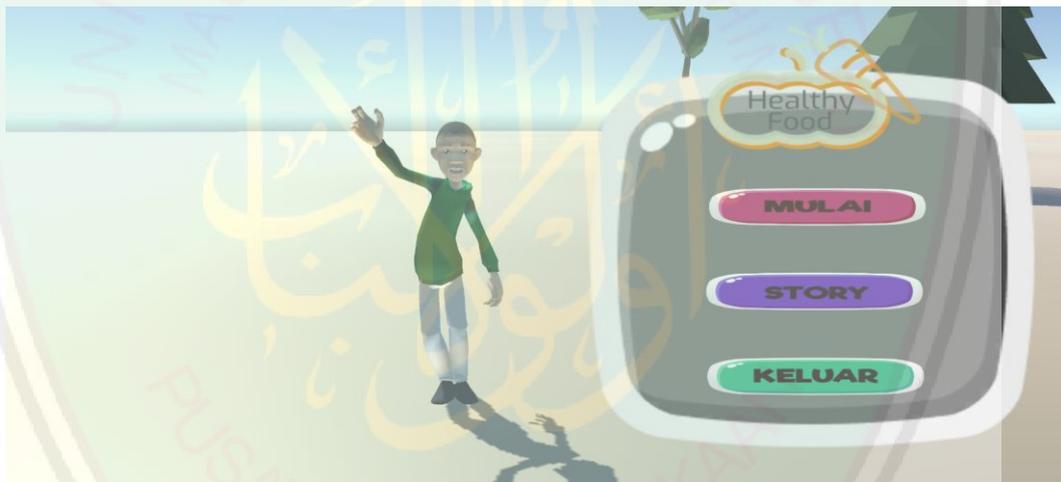
	<pre> } void Pause(){ PanelTujuan.SetActive (true); Time.timeScale=0f; GameIsPaused=true; } } </pre>	
9	<pre> public Node(bool _walkable, Vector3 _worldPos, int _gridX, int _gridY) { walkable = _walkable; worldPosition = _worldPos; gridX = _gridX; gridY = _gridY; } public int fCost { get { return gCost + hCost; } } } </pre>	<p><i>Script</i></p> <p>disamping digunakan untuk mengatur <i>node</i> saat NPC bergerak menuju ke posisi pemain.</p>
10	<pre> public List<Node> GetNeighbours(Node node) { List<Node> neighbours = new List<Node>(); for (int x = -1; x <= 1; x++) { for (int y = -1; y <= 1; y++) { if (x == 0 && y == 0) continue; int checkX = node.gridX + x; int checkY = node.gridY + y; </pre>	<p><i>Script</i></p> <p>disamping digunakan untuk mengatur <i>grid</i> disaat NPC berjalan ke posisi pemain.</p>

	<pre> if (checkX >= 0 && checkX < gridSizeX && checkY >= 0 && checkY < gridSizeY) {neighbours.Add (grid[checkX,checkY]); } } } return neighbours; } </pre>	
11	<pre> void FindPath(Vector3 startPos, Vector3 targetPos) { timer.Reset(); timer.Start(); Node startNode = grid.NodeFromWorldPoint (startPos); Node targetNode= grid.NodeFromWorldPoint (targetPos); UnityEngine.Debug.Log ("Posisi Awal NPC: " + musuhNPC.transform.position); UnityEngine.Debug.Log ("Posisi Player: " + dicari.transform.position); List<Node> openSet = new List<Node> (); HashSet<Node> closedSet = new HashSet<Node> (); openSet.Add(startNode) ; while (openSet.Count > 0) { </pre>	<p><i>Script</i></p> <p>disamping merupakan inti dari algoritma A* dimana NPC akan mencari posisi pemain dari posisi awal NPC.</p>

<pre> Node currentNode = openSet[0]; for (int i = 1; i < openSet.Count; i++) { if (openSet[i].fCost < currentNode.fCost openSet[i].fCost == currentNode.fCost && openSet[i].hCost < currentNode.hCost) { currentNode = openSet[i] ; } } </pre>	
--	--

4.1.2 Implementasi Skenario Game

A. Menu Utama



Gambar 4.1 Tampilan menu utama

Gambar 4.1 (tampilan menu utama), terdapat 3 tombol yakni tombol mulai, tombol *story* dan juga tombol *out*. Tombol mulai jika ditekan akan menuju ke dalam *scene* pemilihan karakter. Tombol *story* jika ditekan akan menuju ke *panel story*. Sedangkan tombol keluar jika ditekan akan keluar dari *game Healthy Food*.

B. Scene Story



Gambar 4.2 Scene story

Gambar 4.2 (*Scene Story*), terdapat karakter tokoh beserta rangkuman cerita singkat tentang *game* ini beserta tombol kembali yang apabila ditekan akan kembali ke menu utama.

C. Scene Pemilihan Karakter



Gambar 4.3 Scene pemilihan karakter

Gambar 4.3 (*scene* pemilihan karakter), terdapat karakter tokoh yang berjumlah 2 yaitu Josh dan Vyn. Pemain dapat mengganti karakter mana yang

akan dimainkan dengan menekan tanda panah samping. Selain itu juga terdapat tombol *Go* dan juga kembali.

D. *Scene* saat Game dimulai



Gambar 4.4 *Scene* saat *game* dimulai

Gambar 4.4 (*scene* saat *game* dimulai), terdapat karakter pemain yang sudah dipilih dan berada pada lingkungan dari *game* tersebut (hutan). Pada *scene* ini terdapat tombol pengaturan, tombol tendang dan juga pukul. Selain itu juga terdapat *bar* yang memberi informasi seberapa HP (*healthpoint*) dari pemain.

E. *Scene* saat tombol pengaturan ditekan



Gambar 4.5 *Scene* saat tombol pengaturan ditekan

Gambar 4.5 (*Scene* saat tombol pengaturan ditekan), terdapat tiga tombol yaitu tombol *resume*, tombol menu dan tombol keluar dari *game*.

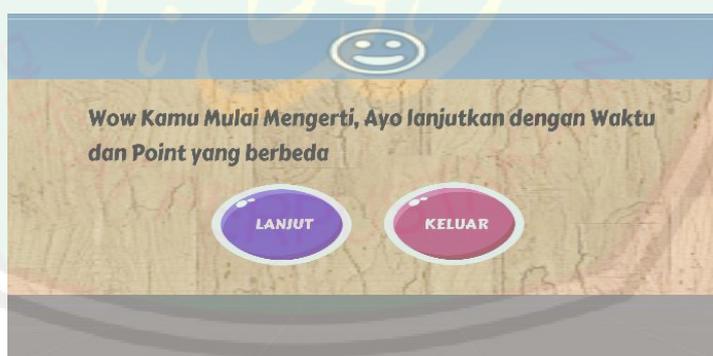
F. *Scene* posisi awal NPC



Gambar 4.6 *Scene* posisi harimau awal

Gambar 4.6 (*scene* posisi harimau awal), saat *game* dimulai dengan bantuan Algoritma A* maka akan membuat NPC menuju ke posisi dimana pemain berada.

G. *Scene* saat memenangkan pertandingan



Gambar 4.7 *Scene* saat memenangkan pertandingan

Gambar 4.7(*Scene* saat memenangkan pertandingan), kondisi ini terjadi apabila pemain dapat mengumpulkan poin sesuai target dan juga tidak mati dalam *game*.

H. Scene saat pemain kalah



Gambar 4.8 Scene saat pemain kalah

Gambar 4.8 (scene saat pemain kalah), kondisi ini terjadi apabila pemain mati atau poin yang didapatkan tidak sesuai target.

4.2 Pengujian

A. Pengujian waktu tempuh NPC ke posisi player dengan algoritma A*

- Pengujian 1 *map* normal



Gambar 4.9 Gambar percobaan waktu tempuh dengan kondisi *map* normal

Pada gambar 4.9 (percobaan waktu tempuh dengan kondisi *map* normal), dilakukan ujicoba dengan menggunakan algoritma A*. Pada pengujian kali ini diperoleh hasil dimana untuk pergi ke posisi pemain NPC memerlukan waktu sebanyak 32 detik.

- Pengujian 2 *map* diperbesar 1.25 kali



Gambar 4.10 Gambar percobaan waktu tempuh dengan kondisi *map* diperbesar 1.25 kali

Pada gambar 4.10 (percobaan waktu tempuh dengan kondisi *map* diperbesar 1.25 kali), dilakukan ujicoba dengan menggunakan algoritma A*. Pada pengujian kali ini diperoleh hasil dimana untuk pergi ke posisi pemain NPC memerlukan waktu sebanyak 41 detik.

- Pengujian 3 map diperbesar 1.5x



Gambar 4.11 Gambar percobaan waktu tempuh dengan kondisi *map* diperbesar 1.5 kali

Pada gambar 4.11(percobaan waktu tempuh dengan kondisi *map* diperbesar 1.5 kali), dilakukan ujicoba dengan menggunakan algoritma A*. Pada pengujian kali ini diperoleh hasil dimana untuk pergi ke posisi pemain NPC memerlukan waktu sebanyak 48 detik.

- Pengujian 4 map diperbesar 1.75x



Gambar 4.12 Gambar percobaan waktu tempuh dengan kondisi *map* diperbesar 1.75 kali

Pada gambar 4.12 (percobaan waktu tempuh dengan kondisi *map* diperbesar 1.75 kali), dilakukan ujicoba dengan menggunakan algoritma A*. Pada pengujian kali ini diperoleh hasil dimana untuk pergi ke posisi pemain NPC memerlukan waktu sebanyak 56 detik.

- Pengujian 5 map diperbesar 2x



Gambar 4.13 Gambar percobaan waktu tempuh dengan kondisi *map* diperbesar 2 kali

Pada gambar 4.13 (percobaan waktu tempuh dengan kondisi *map* diperbesar 2 kali), dilakukan ujicoba dengan menggunakan algoritma A*. Pada pengujian kali ini diperoleh hasil dimana untuk pergi ke posisi pemain NPC memerlukan waktu sebanyak 65 detik.

B. Pengujian waktu tempuh NPC ke posisi player tanpa Algoritma A*

- Pengujian 1 *map* normal



Gambar 4.14 Gambar percobaan waktu tempuh dengan kondisi *map* normal

Pada gambar 4.14 (percobaan waktu tempuh dengan kondisi *map* normal), dilakukan ujicoba tanpa menggunakan algoritma A*. Pada pengujian kali ini diperoleh hasil dimana untuk pergi ke posisi pemain NPC memerlukan waktu sebanyak 66 detik.

- Pengujian 2 *map* diperbesar 1.25x



Gambar 4.15 Gambar percobaan waktu tempuh dengan kondisi *map* diperbeesar

1.25 kali

Pada gambar 4.15 (percobaan waktu tempuh dengan kondisi *map* diperbesar 1.25 kali), dilakukan ujicoba tanpa menggunakan algoritma A*. Pada pengujian kali ini diperoleh hasil dimana untuk pergi ke posisi pemain NPC memerlukan waktu sebanyak 81 detik.

- Pengujian 3 *map* diperbesar 1.5x



Gambar 4.16 Gambar percobaan waktu tempuh dengan kondisi *map* diperbesar 1.5 kali

Pada gambar 4.16 (percobaan waktu tempuh dengan kondisi *map* diperbesar 1.5 kali), dilakukan ujicoba tanpa menggunakan algoritma A*. Pada pengujian kali ini diperoleh hasil dimana untuk pergi ke posisi pemain NPC memerlukan waktu sebanyak 96 detik.

- Pengujian 4 map diperbesar 1.75x



Gambar 4.17 Gambar percobaan waktu tempuh dengan kondisi *map* diperbesar 1.75 kali

Pada gambar 4.17 (percobaan waktu tempuh dengan kondisi *map* diperbesar 1.75 kali), dilakukan ujicoba tanpa menggunakan algoritma A*. Pada pengujian kali ini diperoleh hasil dimana untuk pergi ke posisi pemain NPC memerlukan waktu sebanyak 113 detik.

- Pengujian 5 map diperbesar 2x



Gambar 4.18 Gambar percobaan waktu tempuh dengan kondisi *map* diperbesar 2 kali

Pada gambar 4.18 (percobaan waktu tempuh dengan kondisi *map* diperbesar 2 kali), dilakukan ujicoba tanpa menggunakan algoritma A*. Pada pengujian kali ini diperoleh hasil dimana untuk pergi ke posisi pemain NPC memerlukan waktu sebanyak 132 detik.

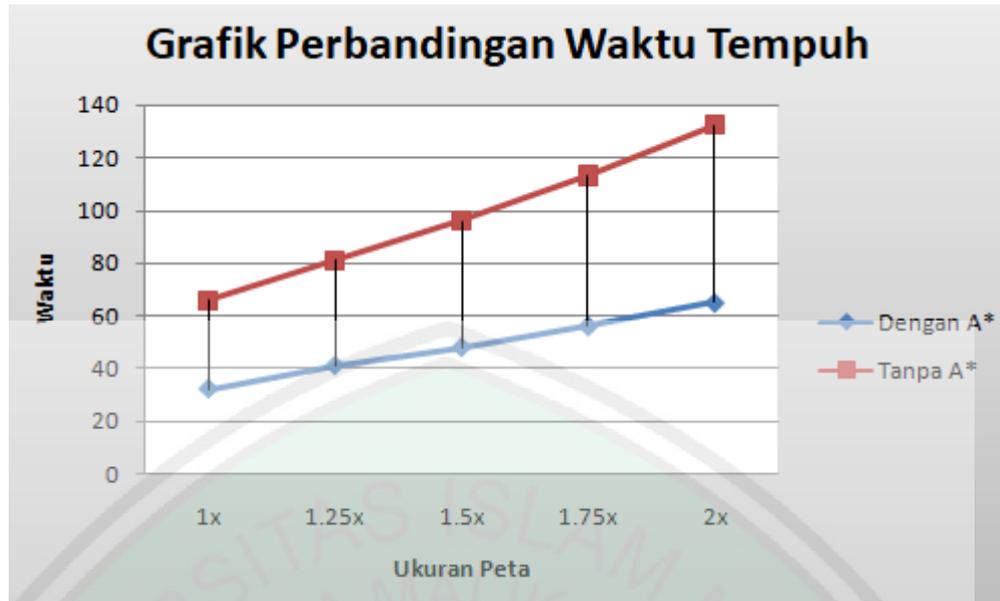
4.3 Pembahasan

Setelah melakukan pengujian waktu tempuh yang digunakan pemain menggunakan algoritma A* dan tanpa menggunakan algoritma A*, diperoleh hasil seperti pada tabel 4.2 (tabel hasil pengujian).

Tabel 4.2 Tabel hasil pengujian

Percobaan Ke-	Waktu tempuh menggunakan algoritma A*	Waktu tempuh tanpa menggunakan algoritma A*	Selisih Waktu tempuh
1	32 detik	66 detik	34 detik
2	41 detik	81 detik	40 detik
3	48 detik	96 detik	48 detik
4	56 detik	113 detik	57 detik
5	65 detik	132 detik	67 detik

Pada percobaan yang dilakukan mendapatkan hasil dimana selisih waktu tempuh yang didapatkan dari percobaan pertama sampai dengan percobaan terakhir naik yaitu dari selisih waktu tempuh awal (percobaan 1 dan 2) selama 6 detik, selisih waktu tempuh kedua (percobaan 2 dan 3) selama 8 detik, selisih waktu tempuh ketiga (percobaan 3 dan 4) selama 9 detik dan selisih waktu tempuh akhir (percobaan 4 dan 5) selama 10 detik.



Gambar 4.19 Gambar grafik perbandingan waktu tempuh

Hasil percobaan perbandingan waktu tempuh menunjukkan bahwa perbedaan waktu tempuh yang dihasilkan tanpa menggunakan algoritma A* dua kali lebih lama. Hal ini disebabkan perbedaan kondisi *map* yang diperbesar dan juga penggunaan algoritma A* yang membuat waktu tempuh berbeda. Jadi, jika tidak menggunakan algoritma A* dan *map* yang digunakan semakin besar maka waktu yang ditempuh akan semakin lama. Hal ini seperti pada gambar 4.19 (grafik perbandingan waktu tempuh).

Dalam *game Healthy Food* mengharuskan pemain harus dapat memanfaatkan waktu sebaik mungkin dalam mencari mana makanan sehat dan mana makanan kurang sehat (*junk food*). Pemain harus dapat mengatur waktunya dikarenakan adanya batasan waktu dalam mencari makanan yang ada. Hal ini sesuai dengan sabda Nabi Muhammad SAW:

نِعْمَتَانِ مَعْبُودُونَ فِيهِمَا كَثِيرٌ مِنَ النَّاسِ: الصِّحَّةُ وَالْفَرَاغُ

"Dua nikmat yang banyak manusia tertipu di dalam keduanya, yaitu nikmat sehat dan waktu luang." (HR. Bukhari, Tirmidzi dan Ibnu Majah).

Sesuai dengan hadits diatas, waktu yang diberikan untuk menyelesaikan *game* ini harus dimanfaatkan dengan sebaik-baiknya oleh pemain dikarenakan terdapat halangan seperti NPC yang akan mengganggu dan menyerang pemain.



BAB V

PENUTUP

5.1 Kesimpulan

Penerapan algoritma A* mempengaruhi waktu tempuh yang dibutuhkan NPC ke posisi pemain pada *game Healthy Food*. Hal ini ditinjau dari sepuluh percobaan yang dilakukan dimana rata-rata waktu tempuh yang dibutuhkan menggunakan algoritma A* sebesar 48.4 detik sedangkan waktu tempuh yang dibutuhkan tanpa menggunakan algoritma A* sebesar 97.6 detik, dapat ditarik kesimpulan bahwa penggunaan algoritma A* dapat menjadi solusi untuk memangkas waktu tempuh yang diperlukan NPC untuk mencari rute menuju ke posisi pemain menjadi lebih cepat.

5.2 Saran

1. Ukuran dari *game Healthy Food* dibuat lebih kecil lagi agar penyimpanan dalam telepon seluler tidak terlalu berat.
2. Dari segi grafik perlu ditingkatkan lagi agar lebih *smooth* dan terlihat lebih menarik.
3. Dapat menambahkan variasi dalam *game* berupa penambahan *quest* yang membuat *game* lebih seru.



DAFTAR PUSTAKA

- Prasetyono, Dwi. 2009. *Buku Pintar Asi Eksklusif*. Diva Press. Yogyakarta.
- Oetoro, S. Parengkuan dan E., Parengkuan, J. 2012. *Smart Eating*. Jakarta : Gramedia Pustaka Utama
- Griffindors, A. 2013. *Mengenal Junk Food (Makanan Sampah)*. Retrieved from <http://artikelampuh.blogspot.com/>
- Barasi, Mary. 2007. *At A Glance Ilmu Gizi: Jakarta* : Penerbit Erlangga
- Proverawati, Atikah. 2010. *Obesitas dan Gangguan Perilaku Makan pada Remaja*. Jakarta: Muha Medika.
- Bagus, Ida. 2018. *Penerapan Algoritma A* (Star) Menggunakan Graph Untuk menghitung Jarak Terpendek* : Jurnal Resistor Vol. 1 No 1 – April 2018.
- Hermawan, Hendi dan Setiyani, Hari. 2019. *Implementasi Algoritma A-Star Pada Permainan Komputer Roguelike Berbasis Unity* : Jurnal Algoritma, Logika dan Komputasi Vol.II (No. 1) : 111 - 120. Th. 2019 .
- Ahmad, Imam dan Widodo, Wahyu. 2017. *Penerapan Algoritma A Star (A*) pada Game Petualangan Labirin Berbasis Android* : Khazanah Informatika Vol. 3 No. 2 Desember 2017.
- Masri dan Tri. 2014. *Pencarian Jalur Terpendek pada Snake Game menggunakan Algoritma A** : Jurnal Imiah SISFOTENIKA Vol. 4, No.1, Januari 2014.
- Fuad, Achmad dan Ahsan, Mohammad. 2015. *Penerapan Metode A* Pada Game Mobile Learning Pemilihan Sampah Organik Dan Anorganik Berbasis Android*: BIMASAKTI: Jurnal Riset Mahasiswa FTI UNIKAMA.
- Expro. Pengenalan Jenis-jenis game. 2010. Internet. <http://forum.indogamers.com/showthread.php?t=414475>. Diakses pada 20 November 2020.
- Teguh, Kurniawan. 2015. *Pengembangan Game dengan menggunakan Game Engine Game Maker*.”Jurnal Sistem Komputer – Vol. 5, No 1, Mei 2015.
- Arif, Yunifa M. (2010). *Strategi Menyerang pada Game FPS menggunakan Hierarchy Finite State Machine dan Logika Fuzzy*. Surabaya: Pasca Sarjana Teknik Elektro ITS
- Arhami, Muhammad dan Desiani, Anita. 2006. *Konsep Kecerdasan Buatan*. Yogyakarta: Penerbit ANDI.

Juniawan, et.al. 2019. *Implementation of Finding Shortest Route for Pharmacy Location Using A-Star Algorithm*. Jurnal Online Informatika Vol 4 No.1 Juni 2019: 46-52

Prasetyo, et.al. 2019. *Perbandingan Algoritma Astar dan Dijkstra Dalam Menentukan Rute Terdekat*. Jurnal Ilmiah SISFOTENIKAJ Vol. 9, No. 1, Januari 2019.

Proverawati, Atikah (2010). *Obesitas dan Gangguan Perilaku Makan Pada Remaja*. Yogyakarta : Nuha Medika.

Oase.id. (2020, 7 Februari). *3 Hadis Ini Bikin Kamu Tak Berani Menyepelkan Waktu*. Diakses pada 28 Desember 2020, dari <https://m.oase.id/read/XW190R-3-hadis-ini-bikin-kamu-tak-berani-menyepelkan-waktu>.

Bayan.id. (2018, 28 Juni). *Surat Al-Baqarah Ayat 172 (2:172)*. Diakses pada 28 Desember 2020, <https://www.bayan.id/quran/2-172/>

