

Penerapan Navmesh Dengan Algoritma A Star Pathfinding Pada Game Edukasi 3d Go Green

Linda Safira¹⁾; Paulus Harsadi²⁾; Sri Harjanto³⁾

^{1) 2) 3)} Program Studi Informatika, STMIK Sinar Nusantara

¹⁾ 17510022.linda@sinus.ac.id; ²⁾ paulusharsadi@sinus.ac.id ; ³⁾ sriharjanto@sinus.ac.id

ABSTRACT

Cleanliness is one of the requirements to create a healthy living environment. But in reality, not everyone have a concern for environmental hygiene and health. For this reason, learning about the importance of caring for environmental cleanliness must be instilled early on. One interesting method to provide learning about environmental cleanliness can be through games. With the 3D visualization of this environmental cleanliness game, it is an effort to raise awareness of the importance of environmental hygiene and health in the community. The location used in making this game is Campus 1 Poltekkes Surakarta. The purpose of this study is implementation of Navmesh with the A Star (A*) Pathfinding Algorithm on the characters so that players can find the trash location point and the location point of the tree with the shortest route. Making this game uses the GDLC (Game Development Life Cycle) method with stages of Initiation, Pre Production, Production, Testing, Beta and Release. This game can be played on desktop and mobile devices. The results of the research that has been done are seen from the algorithm simulation testing is the nodes that are passed by as many as 50 nodes, where there are duplicate data as many as 7 nodes due to real time systems that read nodes over and over. Thus, the application accuracy rate reaches 86%.

Keywords: Educational Games, A * Pathfinding, Navmesh

I. PENDAHULUAN

Politeknik Kesehatan Surakarta adalah perguruan tinggi kesehatan yang penyelenggaraan pendidikan tenaga kesehatan program vokasi Diploma III, Diploma IV dan Profesi yang berlokasi di Surakarta. Saat ini, Poltekkes Surakarta telah berkembang menjadi 9 jurusan dan 20 program studi.

Kebersihan merupakan salah satu syarat untuk mewujudkan lingkungan hidup yang sehat. Untuk itu, pembelajaran tentang pentingnya kepedulian terhadap kebersihan lingkungan harus ditanamkan sejak dini. Salah satu metode menarik untuk memberikan pembelajaran tentang kebersihan lingkungan dapat melalui game. Dengan adanya visualisasi 3D dari game kebersihan lingkungan ini menjadi upaya untuk meningkatkan kesadaran pentingnya kebersihan dan kesehatan lingkungan pada masyarakat. Lokasi yang digunakan dalam pembuatan game ini adalah Kampus 1 Poltekkes Surakarta.

Go Green adalah upaya menyelamatkan bumi dengan tindakan nyata menghilangkan kebiasaan buruk masyarakat yang mengakibatkan kerusakan dengan fokus pada penghijauan lingkungan.

Pada pembuatan game edukasi 3D Go Green ini menerapkan penggunaan Navmesh dengan Algoritma A Star (A*) Pathfinding pada

karakter agar pemain dapat mencari titik lokasi sampah dan titik lokasi pohon dengan rute terpendek. Metode Algoritma A Star (A*) ini memberikan solusi yang tepat dalam pencarian jalur terbaik.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Berikut delapan jurnal penelitian terdahulu yang terkait dengan penentuan menggunakan Algoritma A Star (A*).

Penggunaan A Star (A*) pada Game *Funny English* dimana game ini ditujukan untuk memudahkan belajar bahasa inggris dengan objek 3D dengan menambahkan fitur *artificial intelligence* (AI) di *non-playable character* (NPC). Dari 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. Rata-rata waktu yang dibutuhkan untuk pengambilan rute yaitu 0.2 detik. [1]

Penelitian berikutnya menelusuri lebih jauh bagaimana sebuah NPC bergerak ke sebuah titik dalam peta dengan efektif tanpa melalui jalur yang tidak diperlukan dengan menggunakan algoritma A Star (A*) dalam lingkungan 3 dimensi. Peta dalam game ini dibantu dengan navmesh, terbukti membantu dalam game sehingga terlihat realistis dan

masuk akal [2].

Dalam kehidupan sehari-hari perpindahan dari suatu tempat ke tempat lain sangat mempertimbangkan efisiensi waktu dan biaya, maka diperlukan pengetahuan untuk menentukan suatu jalur terpendek antar tempat yang ingin dituju. Algoritma pencarian jalur terpendek atau lebih dikenal sebagai *shortest-path* dipakai dalam menentukan rute dalam sebuah *graph*. Simulasi ini dapat menentukan rute (jalur) terbaik dari titik awal (*start*) menuju titik akhir (*finish*) dengan hambatan-hambatan yang diberikan di setiap rute. Dari hasil pengujian, rute yang ditemukan merupakan rute yang terbaik dengan nilai $f(n)$ terkecil dibandingkan dengan rute-rute (jalur-jalur) lainnya [3].

Permasalahan *path finding* merupakan masalah yang sangat penting dalam berbagai game. *Path finding* digunakan untuk menentukan arah pergerakan suatu objek dari satu titik ke titik lain dalam game berdasarkan keadaan peta dan objek lainnya. *Navigation Mesh* sendiri merupakan suatu bentuk struktur graf yang dapat menyederhakan perhitungan *path finding*. Metode *Path Finding* diperlukan sebab setiap pergerakan pada game tidak hanya bergerak menuju tujuan tertentu, tetapi juga melakukan hal lainnya bersamaan dengan bergerak. Maka dari itu, algoritma A^* dan *Navigation Mesh* dipakai untuk melakukan *pathfinding* lebih cepat menuju ke tujuan [4].

Penelitian ini membahas cara efektif untuk mengoptimalkan masalah penelusuran jalur terpendek, yaitu *Navgasi Mesh (NavMesh)*. Dimana, *NavMesh* diimplementasikan dengan menggunakan algoritma A^* (*A Star*) dan diperiksa dalam mesin game Unity 3D. Game Pac-Man dipilih sebagai contoh *pathfinding* terpendek dengan menggunakan *NavMesh* di Unity 3D. Algoritma A^* diimplementasikan pada musuh Pac-Man (tiga hantu), yang jalurnya dirancang dengan menggunakan konsep *NavMesh*. Dengan demikian, pergerakan hantu dalam menangkap Pac-Man adalah hasil dari tinjauan efektivitas konsep ini [5].

Contoh game yang populer adalah game labirin yang merupakan sebuah *puzzle* dalam bentuk percabangan jalan yang kompleks. Penelitian ini menyajikan hasil pengembangan game petualangan labirin yang menceritakan mengenai petualangan kelinci melewati sebuah labirin untuk mencari makanannya. Al yang digunakan dalam pembangunan game petualangan labirin ini adalah algoritma A^* (*A Star*) dengan *Euclidean distance*. Hasil uji dari

game petualangan labirin ini adalah jika pemain dalam kesusahan menemukan jalur menuju makanan kelinci, maka pemain dapat menggunakan tombol bantuan yang akan dicarikan jalur terpendek oleh algoritma A^* (*A Star*) dengan *euclidean distance* untuk menuju lokasi tempat makanan kelinci berada. [6]

Salah satu *genre game* yang banyak dimainkan adalah *turn-based strategy*, yaitu *game* (yang biasanya permainan perang terutama strategi perang) dimana pemain secara bergantian mengatur strateginya untuk mendekati pihak lawan. Biasanya *game* ini juga menggunakan NPC. Dimana NPC juga harus terlihat nyata dalam pergerakan atau teknik *pathfinding* sehingga memerlukan metode algoritma. Algoritma A^* (*A Star*) sering digunakan pada *game* untuk kasus pencarian jalur. Penerapan algoritma A^* (*Star*) dengan menggunakan nilai heuristik yang diperoleh dengan mengkombinasikan jarak garis lurus antar tujuan mampu menyelesaikan kasus *multiple-goal* dengan hasil yang *complete* dan optimal [7].

2.2 Graf

Graf adalah kumpulan simpul (*node*) yang dihubungkan satu sama lain melalui sisi (*edges*). Suatu graf adalah himpunan suatu benda-benda yang disebut dengan *verteks* atau *node* yang terhubung oleh *edge-edge*. Biasanya graf digambarkan sebagai kumpulan titik-titik (melambangkan *verteks*) yang dihubungkan oleh garis[8].

Suatu Graf G terdiri dari dua himpunan yaitu himpunan V dan himpunan E . Terdapat beberapa istilah penting yang berkaitan dengan graf. Berikut ini didefinisikan beberapa terminologi yang sering digunakan [9]:

1. *Graf* tak berarah didefinisikan sebagai suatu pasangan berurutan, (V, E) , yang mempunyai sifat-sifat sebagai berikut :
 - a. V adalah terbatas (*finite*), himpunan yang tidak kosong. Elemen V ini disebut *verteks* dari G .
 - b. E adalah himpunan terbatas dari suatu himpunan. Setiap elemen E , adalah suatu himpunan yang terdiri dari tepat dua *verteks* yang berbeda. Elemen E disebut *edge* dari G .
2. Graf G dikatakan berbobot (*weighted*) bila terdapat bilangan riil yang berasosiasi dengan masing-masing *edge* G . Bobot-bobot itu sendiri biasanya berfungsi pada saat dibuat suatu lintasan antar *verteks-verteks* pada graf tersebut.

2.2 Algoritma A Star (A*)

Algoritma A* adalah algoritma yang dikemukakan oleh Hart, Nilsson, dan Raphael (1968). Algoritma A* merupakan salah satu algoritma untuk melakukan pencarian solusi dengan menggunakan informasi tambahan (heuristik) dalam menghasikan solusi yang optimal. Heuristik adalah penilai yang memberi harga pada tiap simpul yang memandu mendapatkan solusi yang diinginkan. Nilai pada setiap simpul n menyatakan taksiran nilai termurah lintasan dari simpul n ke simpul target menggunakan persamaan 1 yaitu sebagai berikut. [10]

$$f(n) = \text{nilai taksiran lintasan terdekat dari simpul status } n \text{ ke status tujuan} \quad (1)$$

Dengan kata lain, $f(n)$ menyatakan batas bawah (*lower bound*) dari ongkos pencarian solusi dari status n . Fungsi heuristik yang terdapat pada Algoritma A* untuk menghitung taksiran nilai dari suatu simpul dengan simpul yang telah dilalui menggunakan persamaan 2 sebagai berikut.

$$f(n) = g(n) + h(n) \quad (2)$$

Keterangan:

(n) = Estimasi total biaya yang melalui n sampai node tujuan

(n) = Biaya yang diperlukan untuk mencapai goal melalui node n

$h(n)$ = Estimasi biaya dari node ke *goal state*

2.3. NavigationMesh(Navmesh)

Navigation Mesh adalah *Pathfinding* yang dilakukan diantara poligon dalam *mesh* dapat diselesaikan dengan algoritma pencarian salah satu grafik dalam jumlah besar, misalnya A Star. *Navigation mesh (Navmesh)* telah menjadi konsep populer yang digunakan dalam masalah penelusuran jalan terpendek dari game 3D, karena lingkungan 3D sebagian besar menggunakan struktur poligon[11].

2.4 Pathfinding

Pathfinding dalam game adalah pergerakan pemain melalui suatu jalan terpendek tanpa menabrak halangan. Penggunaan metode *pathfinding* adalah pada game *real-time* dimana seseorang pemain pada titik awal menentukan titik akhir/tujuan[12].

III. METODE PENELITIAN

3.1 Teknik Pengumpulan Data

a. Observasi

Pengumpulan data dengan melihat langsung data bangunan yang berhubungan dengan pembuatan game edukasi 3D *Go Green* Kampus 1 Poltekkes Surakarta.

b. Wawancara

Wawancara dilakukan dengan pimpinan Poltekkes Surakarta beserta karyawan IT yang berperan secara langsung dalam proses pembuatan game edukasi 3D *Go Green* Kampus 1 Poltekkes Surakarta.

c. Studi Pustaka

Studi pustaka dilakukan dengan cara mencari referensi atau teori mengenai metode yang digunakan dalam sistem aplikasi melalui buku-buku acuan dan jurnal-jurnal ilmiah guna membangun sistem aplikasi game edukasi 3D *Go Green* Kampus 1 Poltekkes Surakarta Menggunakan *Navmesh* dengan Algoritma A Star (A*) *Pathfinding*.

3.2 Pengembangan Game

Pengembangan game ini menggunakan prosedur pengembangan *Game Development Life Cycle* (GDLC), berikut tahapan-tahapan dari prosedur pengembangannya [13].

a. Initiation

Pada tahap ini dilakukan pembuatan konsep game yang ingin dibangun.

b. Pre-Production

Tahapan ini merupakan tahapan sebelum *production* yaitu dimana pada tahapan ini dilakukan pembuatan desain dan perancangan game.

c. Production

Pada *pre-production* terdapat *prototype* dan *game design* yang disempurnakan pada *production*.

d. Testing

Merupakan pengujian terhadap *prototype build*. Setelah sistem aplikasi game selesai dibuat yaitu dilakukan pengujian game berupa uji black box

e. Beta

Tahap ini dilakukan untuk menguji sistem aplikasi game, juga untuk mendeteksi berbagai *error* serta keluhan dari *third party tester*/pihak ketiga.

f. Release

Setelah game sudah selesai dibuat

dan lulus pengujian tahap *beta testing*, maka menandakan bahwa game siap untuk dirilis ke publik.

3.3 Implementasi

Implementasi sistem aplikasi game edukasi 3D *Go Green* Kampus 1 Poltekkes Surakarta menggunakan *Navmesh* dengan Algoritma *A Star (A*) Pathfinding* dapat digunakan pada perangkat *desktop* atau PC dan android.

3.4 Pengujian

Pada tahap ini dilakukan pengujian dengan menggunakan metode *blackbox testing*. Dalam pengujian game ini dilakukan dengan 2 cara yaitu pengujian berdasarkan fungsional game dan pengujian komparabilitas *device* [14].

a. Pengujian Fungsional Game

Uji fungsional dilakukan dengan metode *black box testing* yaitu untuk mencari kesalahan dan kekurangan dari sistem aplikasi yang telah dibuat untuk kemudian dilakukan perbaikan dan penyempurnaan sistem aplikasi.

b. Pengujian Komparabilitas Device

Dalam pengujian komparabilitas dilakukan dengan pengujian *device* pada beberapa perangkat android yang bertujuan untuk apakah game berjalan baik pada setiap perangkat maupun versi android.

IV. HASIL DAN PEMBAHASAN

4.1 Initation

Tahap *initiation* berfokus pada ide dan konsep dasar game. Game edukasi 3D *Go Green* merupakan game edukasi yang memiliki genre *Adventure*. Game ini dibuat dengan menggunakan Unity 3D dan diterapkan dalam *platform* android.

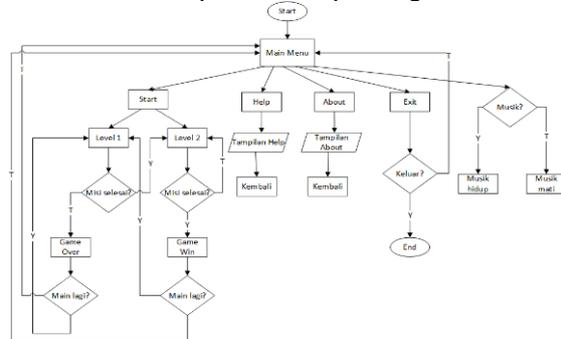
4.2 Pre-Production

Tahap *pre-production* bertujuan memperoleh gambaran jelas tentang konsep pengembangan game yang dibuat dan membantu dalam menentukan target serta deadline dalam proses pembuatan game. Pada tahap desain game mencakup penjelasan berupa narasi, misi, *storyboard*, target pemain, *gameplay*, *playability*, *genre* dan konsep seni yang digunakan pada game edukasi 3D *Go Green*, sedangkan pada tahap perancangan berupa pembuatan rancangan

dari karakter dan tampilan lingkungan pada game edukasi 3D *Go Green*.

4.2.1 Flowchart

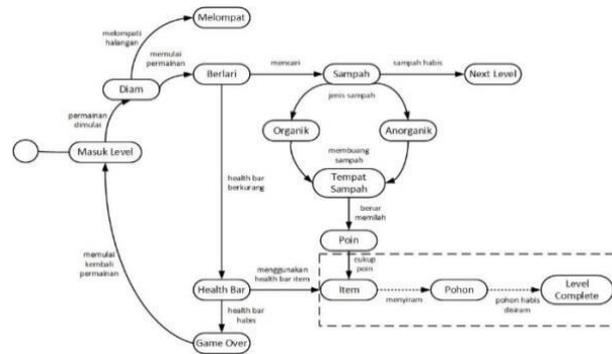
Berikut adalah *flowchart* game edukasi 3D *Go Green* dapat dilihat pada gambar 1.



Gambar 1 *Flowchart* Permainan

4.2.2 Finite State Machine

Berikut adalah *finite state machine* game edukasi 3D *Go Green* dapat dilihat pada Gambar 2.



Gambar 2 *Finite State Machine* Permainan

4.2.3 Cara Kerja Algoritma A*

Berikut adalah *pseudocode* algoritma A*:

1. Simpan simpul awal atau *current state* pada list dengan nama *OPEN* dan list *CLOSED* = {}.
2. Loop :
 - a. Cari node (n) dengan nilai f (n) yang memiliki nilai minimum dalam *OPEN list*. Node ini sekarang menjadi *current state*.
 - b. Keluarkan *current state* dari dalam
 - *OPEN list* lalu simpan ke dalam
 - *CLOSED list*.
 - c. Untuk setiap objek yang dituju dari node *current state* lakukan pengecekan seperti berikut :
 - Jika tidak dapat dilalui atau sudah berada di dalam *CLOSED list*, maka tidak akan dihitung lagi.

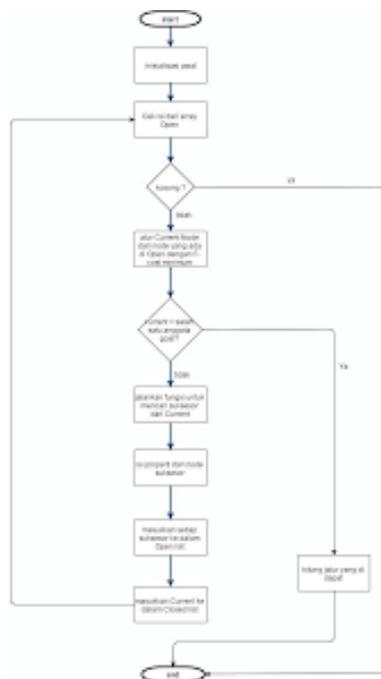
- Jika belum pernah tersimpan di dalam *OPEN list* maka simpan di dalam *OPEN list*. Kemudian buat *parent* baru dari node ini dan simpan nilai f, g, dan h dari node ini.
- Jika sudah berada di dalam *OPEN list*, cek bila node saat ini memiliki nilai f(n) lebih kecil dari node sebelumnya. Jika node $g'(n) < g(n)$ maka ganti *parent* dari node $g'(n)$ yang ada di dalam list *OPEN* menjadi *current node*. Kemudian lakukan perhitungan terbaru dari nilai f(n) yang sekarang.

d. Loop dihentikan apabila :

- Node tujuan telah terdapat di dalam *OPEN list*.
- Belum menemukan node tujuan, sementara list dalam *OPEN* sudah tidak ada lagi yang artinya tidak ditemukan solusi.

3. Simpan rute.

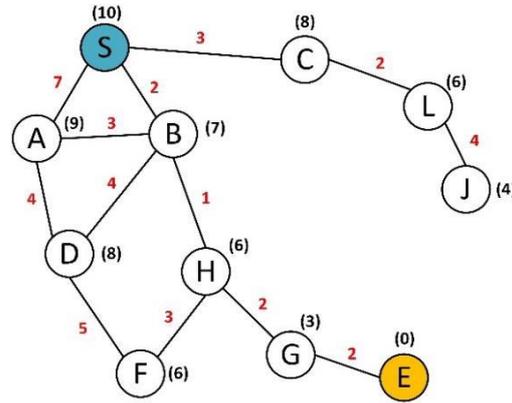
Lakukan *backtrack* urutan dari node tujuan menuju node awal, maka solusi telah ditemukan.



Gambar 3 Flowchart Algoritma A Star

4.2.4 Studi Kasus Algoritma A*

Berikut adalah studi kasus pada Algoritma A Star (A*) dapat dilihat pada Gambar 4.



Gambar 4 Studi Kasus Algoritma A Star

Diketahui :

- Angka tanpa tanda kurung adalah jarak antar titik atau g(n) yang dapat dilihat pada Tabel 1.

Tabel 1 Tabel Jarak Antar Titik

Lintasan	g(n)	Lintasan	g(n)
S-A	7	B-D	4
S-B	2	B-H	1
S-C	3	D-F	5
C-L	2	H-F	3
L-J	4	H-G	2
A-B	3	G-E	2
A-D	4		

- Angka dengan tanda kurung adalah nilai heuristik atau h(n) yang dapat dilihat pada Tabel 2.

Tabel 2 Tabel Nilai Heuristik

n	S	A	B	C	D	E	F	G	H	J	L
h(n)	10	9	7	8	8	0	6	3	6	4	6

- Kemudian mengerjakan titik E, karena pada *open list* titik B memiliki titik terendah.
- Titik S merupakan titik asal atau titik lokasi karakter pemain
- Titik E merupakan titik tujuan atau titik lokasi tempat sampah. Sehingga, nilai heuristik titik tujuan E adalah 0
- Titik A, B, C, D, F, G, H, J dan L merupakan titik sampah yang dapat diambil oleh karakter pemain
- Lintasan S, A, B adalah lintasan dengan baris sisi (A,B), (B,S), (S,A) dst
- Lintasan A dan B memiliki panjang 3, dst.

Cara menghitung harga S-B-H adalah sebagai berikut.

$$\text{Jarak S-H} = 2 + 1 = 3$$

Kemudian jumlahkan jarak dengan

nilai heuristik titik H yaitu 6.

Maka, harga S-B-H = $3+6 = 9$

Ditanya : Berangkat dari titik S ke titik E?

Jawab :

Siapkan dua tabel terlebih dahulu yaitu *open list* dan *closed list*. Dimana, *Open list* akan berisi daftar titik yang sedang diproses sedangkan *Closed list* akan berisi daftar titik yang sudah selesai diproses dan merupakan jalur terbaik yang didapatkan. Lihat titik sumber, dari titik S maka dapat menuju 3 titik yaitu titik A, titik B dan titik C. Tampilan *open list* bagian pertama dapat dilihat pada Tabel 3.

Tabel 3 Tabel *Open List* Bagian Pertama

OPEN LIST			Keterangan
Via	Harga	Tujuan	
S	$2 + 7 = 9$	B	Isi <i>open list</i> harus berurutan berdasarkan harganya
S	$3 + 8 = 11$	C	
S	$9 + 7 = 16$	A	

Karena titik S sudah selesai dikerjakan maka masukkan titik S ke dalam *closed list*, sehingga titik S hanya dihitung dari nilai heuristik nya saja. Tampilan *closed list* bagian pertama dapat dilihat pada Tabel 4.

Tabel 4 Tabel *Closed List* Bagian Pertama

CLOSED LIST		
Via	Harga	Tujuan
S	10	S

Pindahkan terlebih dahulu titik B ke *closed list*. Tampilan *open list* bagian kedua dapat dilihat pada Tabel 5.

Tabel 5 Tabel *Open List* Bagian Kedua

OPEN LIST		
Via	Harga	Tujuan
S	$3 + 8 = 11$	C
S	$9 + 7 = 16$	A

Tampilan *closed list* bagian kedua dapat dilihat pada Tabel 6.

Tabel 6 Tabel *Closed list* Bagian Kedua

CLOSED LIST		
Via	Harga	Tujuan
S	10	S
S	$2 + 7 = 9$	B

Titik B dapat menuju ke titik A, titik D dan titik

H. Harga titik S-B-A = $2+3+9 = 14$.

Harga ini lebih murah dibandingkan S ke A, maka *update* titik A yang ada pada *open list*. Tampilan *open list* bagian ketiga dapat dilihat pada Tabel 7.

Tabel 7 Tabel *Open List* Bagian Ketiga

OPEN LIST		
Via	Harga	Tujuan
S	$3 + 8 = 11$	C
B	$2+3+9 = 14$	A

Harga titik S-B-D = $2+4+8 = 14$. Masukkan ke dalam *open list*. Tampilan *open list* bagian keempat dapat dilihat pada Tabel 8.

Tabel 8 Tabel *Open List* Bagian Keempat

OPEN LIST		
Via	Harga	Tujuan
S	$3 + 8 = 11$	C
B	$2+3+9 = 14$	A
B	$2+4+8 = 14$	D

Harga titik S-B-H = $2+1+6 = 9$. Masukkan ke dalam *open list*. Tampilan *open list* bagian kelima dapat dilihat pada Tabel 9.

Tabel 9 Tabel *Open List* Bagian Kelima

OPEN LIST		
Via	Harga	Tujuan
B	$2+1+6 = 9$	H
S	$3 + 8 = 11$	C
B	$2+3+9 = 14$	A
B	$2+4+8 = 14$	D

Titik B sudah selesai dikerjakan. Kemudian akan mengerjakan titik H, karena pada *open list* titik H memiliki harga terendah. Pindahkan terlebih dahulu titik H ke *closed list*. Tampilan *open list* bagian keenam dapat dilihat pada Tabel 10.

Tabel 10 Tabel *Open List* Bagian Keenam

OPEN LIST		
Via	Harga	Tujuan
S	$3 + 8 = 11$	C
B	$2+3+9 = 14$	A
B	$2+4+8 = 14$	D

Tampilan *closed list* bagian ketiga dapat dilihat pada Tabel 11.

Tabel 11 Tabel *Closed list* Bagian Ketiga

CLOSED LIST		
Via	Harga	Tujuan
S	10	S
S	$2 + 7 = 9$	B
B	$2+1+6 = 9$	H

Harga titik S-B-H-F = $2+1+3+6 = 12$. Masukkan ke dalam *open list*. Tampilan *open list* bagian ketujuh dapat dilihat pada Tabel 12.

Tabel 12 Tabel *Open List* Bagian Ketujuh

OPEN LIST		
Via	Harga	Tujuan
S	$3 + 8 = 11$	C
H	$2+1+3+6 = 12$	F
B	$2+3+9 = 14$	A
B	$2+4+8 = 14$	D

Harga titik S-B-H-G = $2+1+2+3 = 8$. Masukkan ke dalam *open list*. Tampilan *open list* bagian kedelapan dapat dilihat pada Tabel 13.

Tabel 13 Tabel *Open List* Bagian Kedelapan

OPEN LIST		
Via	Harga	Tujuan
H	$2+1+2+3 = 8$	G
S	$3 + 8 = 11$	C
H	$2+1+3+6 = 12$	F
B	$2+3+9 = 14$	A
B	$2+4+8 = 14$	D

Titik dengan harga terendah adalah titik G, maka proseskan titik G. Namun dapat dilihat bahwa titik G ini hanya memiliki tujuan titik E saja dan merupakan titik tujuan. Maka sudah didapatkan bahwa jalur terbaik dari graph ini adalah S-B-H-G-E dengan total harganya adalah $2+1+2+2 = 7$.

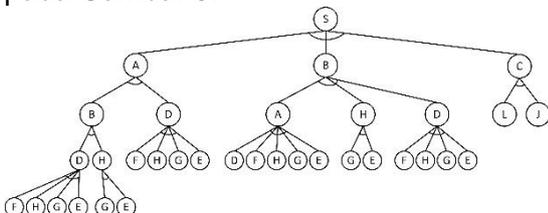
Representasi ruang keadaan dalam graf dapat didetailkan sebagai berikut.

a. Graf keadaan dengan titik S menunjukkan titik awal dan titik E adalah titik tujuan. Ada 6 lintasan dari titik S ke titik E, antara lain :

- S-A-D-F-H-G-E
- S-A-B-H-G-E
- S-A-B-D-F-H-G-E
- S-B-H-G-E
- S-B-A-D-F-H-G-E
- S-B-D-F-H-G-E

b. Lintasan buntu atau lintasan yang tidak sampai ke tujuan yaitu S-C-L-J

Tampilan Pohon AND OR dapat dilihat pada Gambar 5.



Gambar 5 Tampilan Pohon AND OR

4.3 Production

Tahap *production* mencakup pembuatan *asset*, *programming* dan *integration* Antara *asset* serta *source code* dan implementasi *Navmesh* dengan Algoritma A Star (A*)

pathfinding. Pada tahap *Asset* dan seni game berupa desain, *modelling* dan material, *lighting*, *animation*, *game system*, musik dan suara serta *asset* dan item tambahan yang digunakan pada game edukasi 3D Go Green.



Gambar 6 Tampilan Menu



Gambar 7 Tampilan Game

4.4 Testing

Pada tahap ini dilakukan pengujian dengan menggunakan metode *blackbox testing*. Dalam pengujian game ini dilakukan dengan 2 cara yaitu pengujian berdasarkan fungsional game dan pengujian kompatibilitas *device*.

4.4.1 Pengujian Fungsional Game

Pada pengujian ini menampilkan hasil uji dari setiap komponen yang telah dibuat dan telah diimplementasikan. Pengujian yang diuji berupa fungsi pada *scene*, *button* dan *gameplay*. Untuk hasil pengujian dapat dilihat pada tabel 14 berikut ini.

Tabel 14 Tabel Pengujian Fungsional Game

No	Fungsi	Jenis	Pengujian	Keterangan
	Main Menu		Tampil setelah <i>scene loading</i> dari Unity saat memuat game	Berhasil
	Loading		Tampil setelah <i>scene</i> pilih level	Berhasil

No	Fungsi	Jenis	Pengujian	Keterangan
1	Scene	Menu Level 1	Tampil setelah <i>scene</i> pilih level dan <i>loading</i>	Berhasil
		Menu Level 2	Tampil setelah <i>scene</i> pilih level dan <i>loading</i>	Berhasil
2	Button	Start	Berpindah ke <i>scene</i> pilih level ketika tombol ditekan	Berhasil
		How To Play	Tampil tampilan cara bermain dari game	Berhasil
		About	Tampil tampilan about dari pengembang game	Berhasil
		Quit	Keluar dari game ketika tombol ditekan	Berhasil
		Music	Mematikan dan menghidupkan musik dari game	Berhasil
		Pause	Tampil tampilan pause dan menghentikan waktu game ketika tombol ditekan	Berhasil
		Resume	Kembali ke dalam game ketika tombol ditekan dan mengembalikan waktu	Berhasil
		Hint	Tampil tampilan misi pada level game yang sedang berjalan	Berhasil
		Quit	Keluar dari game menuju ke <i>scene main</i>	Berhasil
		Items	Berhasil mengaktifkan item ketika <i>button item</i> di dalam <i>Inventory Tabs</i> ditekan	Berhasil
		Shop	Menampilkan tombol item ketika poin mencukupi dan tombol beli ditekan	Berhasil
		Level 1	Untuk perpindahan <i>scene</i> ke level 1	Berhasil
		Level 2	Untuk perpindahan <i>scene</i> ke level 2	Berhasil

No	Fungsi	Jenis	Pengujian	Keterangan
		Next Level	Untuk perpindahan <i>scene</i> dari level 1 ke level 2	Berhasil
		Play Again	Untuk perpindahan <i>scene</i> ke level 1	Berhasil
3	Gameplay	Animasi <i>player</i> mengambil sampah	Tombol <i>pick up</i> ditekan ketika <i>player</i> mendekati sampah	Berhasil
		Pergerakan <i>player</i>	<i>Player</i> digerakkan dengan <i>virtual analog</i>	Berhasil
		Animasi <i>player</i> menyiram pohon	Ketika <i>player</i> telah mengaktifkan item <i>watering</i>	Berhasil
		Shop system	Membeli item	Berhasil
		Use item	Saat <i>player</i> menggunakan <i>item</i> yang dibeli	Berhasil
		Minimap	Menampilkan kamera secara vertikal dalam game	Berhasil

4.4.2 Pengujian Kompabilitas Device

Dalam pengujian kompabilitas dilakukan dengan pengujian *device* pada beberapa perangkat android yang bertujuan untuk apakah game berjalan baik pada setiap perangkat maupun versi android. Pengujian yang diuji berupa jenis test pada animasi, audio, tombol, grafis, gameplay dan memory. Untuk melihat hasil dari pengujian kompabilitas dapat dilihat pada Tabel 15 berikut ini.

Tabel 15 Tabel Pengujian Kompabilitas Device

No	Jenis Device	Spesifikasi Device	Jenis Test	Keterangan
1	Xiaomi Redmi 5A	Android 7.1.2 (Nougat)	Animasi	Animasi berjalan lancar
		5 inches 1280 x 1080 pixel	Audio	Audio berjalan dengan lancar
		2 GB	Tombol	Seluruh tombol berfungsi dengan baik
			Grafis	Tampilan grafis penuh dan sesuai tidak ada sisi kosong
			Gameplay	Keseluruhan Game berjalan dengan baik

No	Jenis Device	Spesifikasi Device	Jenis Test	Keterangan
			Memory	Ketika menginstall game memakan

4.4.3 Pengujian Simulasi Algoritma A*

Pada pengujian simulasi algoritma ini akan dilakukan pengujian apakah *Navmesh* dengan Algoritma A* *Pathfinding* yang telah diterapkan pada game yang telah dibangun telah berjalan sesuai dengan yang diharapkan. Jika dicari menggunakan *Navmesh* dengan Algoritma A* *Pathfinding* maka didapat solusi optimal pada *console game* seperti pada Gambar 8.



Gambar 8 Tampilan Solusi Optimal pada Console

Dari gambar 5 didapat hasil node yang dilewati sebanyak 50 node, dimana terdapat duplikat data sebanyak 7 node dikarenakan *real time* sistem yang membaca node berulang. Sehingga, tingkat akurasi aplikasi mencapai 86%. Rute yang ditemukan merupakan rute yang terbaik dengan nilai $f(n)$ terkecil dibandingkan dengan rute-rute (jalur-jalur) lainnya. Apabila *Navmesh* dengan Algoritma A Star (A*) *Pathfinding* diterapkan pada game yang telah dibangun maka didapat rute dimana karakter utama menuju titik lokasi sampah dan titik lokasi pohon.

4.5 Beta

Tahap ini menguji sistem aplikasi game edukasi 3D *Go Green*, juga untuk mendeteksi berbagai *error* serta keluhan yang diperoleh dari *third party tester*/pihak ketiga.

Setelah semua tahap berhasil dilewati, maka pengembang mengedarkan kuesioner yang diedarkan kepada pengguna khususnya kepada 10 orang pegawai dan 20 orang calon mahasiswa baru Poltekkes Kemenkes Surakarta, dengan karakteristik

responden antara lain jenis kelamin, asal sekolah, jabatan dan usia. Kuesioner kepuasan pengguna diolah dengan menggunakan Skala Likert dengan rentang skala 1 sampai 4.

Sangat Baik = 4, Baik = 3, Kurang Baik = 2, Tidak Baik = 1

Tabel 16 Tabel Rentang Nilai Skala Likert

Rentang Nilai	Keterangan
90% - 100%	Sangat Baik
80% - 89%	Baik
70% - 79%	Kurang Baik
0% - 69%	Tidak Baik

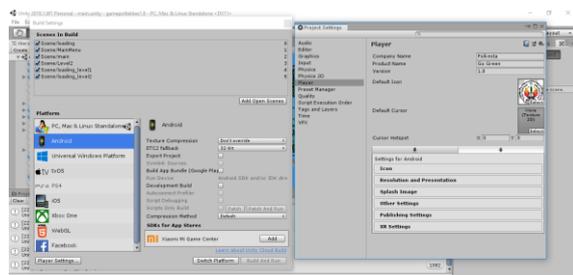
Berikut adalah indikator pertanyaan dan hasil untuk kuesioner game edukasi 3D *Go Green*.

1. Ketertarikan penggunaan game dengan hasil nilai **87,5% (Baik)**
2. Tampilan menarik pada game dengan hasil nilai **90,8% (Sangat Baik)**
3. Kemudahan penggunaan game dengan hasil nilai **86,7% (Baik)**
4. Kelengkapan fitur game dengan hasil nilai **82,5% (Baik)**
5. Kesesuaian topik penyajian informasi game dengan hasil nilai **84,2% (Baik)**
6. Pemahaman bahasa game dengan hasil nilai **85% (Baik)**
7. Pemahaman isi game dengan hasil nilai **83,3% (Baik)**
8. Game termasuk interaktif dengan hasil nilai **91,7% (Sangat Baik)**
9. Game dapat mendorong kesadaran tentang pentingnya kebersihan dan kesehatan dengan hasil nilai **86,7% (Baik)**
10. Pengguna merasa tertantang menyelesaikan game dengan hasil nilai **88,3% (Baik)**

Sehingga, hasil nilai rata-rata indikator pertanyaan pada kuesioner game edukasi 3D *Go Green* adalah **86.67%** dan termasuk dalam kategori **Baik**.

4.6 Release

Dalam tahap ini merupakan tahap akhir dalam pengembangan game *Go Green* berjalan sesuai dengan alur sistem secara keseluruhan. Ketika game sudah selesai dikembangkan dan diuji game siap untuk dirilis. Untuk proses perilis game dapat dilihat pada Gambar 9.



Gambar 9 Tampilan Proses Building Game

V. PENUTUP

5.1 Kesimpulan

1. Telah berhasil dibuat Game Edukasi 3D Go Green Kampus 1 Poltekkes Surakarta menggunakan Navmesh dengan Algoritma A Star (A*) Pathfinding pada karakter agar pemain dapat mencari titik lokasi sampah dan titik lokasi. Game ini juga digunakan sebagai media untuk menyampaikan pembelajaran kepada pengguna game tentang pentingnya sikap peduli terhadap kebersihan lingkungan.
2. Dari hasil pengujian simulasi algoritma adalah node yang dilewati sebanyak 50 node, dimana terdapat duplikat data sebanyak 7 node dikarenakan real time sistem yang membaca node berulang. Sehingga, didapat tingkat akurasi aplikasi yaitu 86%.

5.2 Saran

Dapat dilakukan penambahan aturan pada game dan lingkungan wilayah pada area game agar mencakup gedung-gedung dalam wilayah sehingga dapat dikembangkan dengan mode multiplayer dan metode yang berbeda untuk menentukan rute.

DAFTAR PUSTAKA

- [1] Riwinoto and Alfian, "Implementasi Pathfinding dengan Algoritma A * pada Game Funny English Menggunakan Unity 3D Berbasis Graf Navmesh," *Batam: Skripsi, Teknik Multimedia Jaringan, Fakultas Teknik, Potiteknik Negeri Batam*, vol. 4, 2015.
- [2] E. K. Tandiono, "Algoritma A* Memanfaatkan Navigation Meshes dalam 3 Dimensional Pathfinding," *J. Strateg. Algoritm.*, vol. 7, no. 1, 2017.
- [3] I. B. G. W. A. Dalem, "Penerapan Algoritma A* (Star) Menggunakan Graph Untuk Menghitung Jarak Terpendek," *J. Resist.*, vol. 1, no. 1, pp. 41–47, 2018
- [4] F. Yonathan, "Metode Path Finding pada

Game 3D Menggunakan Algoritma A * dengan Navigation Mesh," *Strateg. Algoritm.*, vol. 1, no. 1, 2011.

- [5] M. Zikky, "Review of A * (A Star) Navigation Mesh Pathfinding as the Alternative of Artificial Intelligent for Ghosts Agent on the Pacman Game," *Emit. Int. J. Eng. Technol.*, vol. 4, no. 1, pp. 141–149, 2016.
- [6] W. Widodo and I. Ahmad, "Penerapan Algoritma A Star (A*) pada Game Petualangan Labirin Berbasis Android," *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 3, no. 2, pp. 57–63, 2018.
- [7] P. J. Taufiq, A. T. Wibowo, and G. Septiana, "Implementasi dan Analisis Algoritma A*(Star) untuk Menentukan Jalur dengan Multiple Goal pada Pegerakan NPC (Non-Playable Character)," *eProceedings of Engineering.*, vol. 2, no. 3, pp. 7799–7810, 2015.
- [8] Zakaria, *Teknologi Informasi dan Komunikasi*. Jakarta: Arya Duta, 2006.
- [9] P. Kadek Ayu Yanti and J. Purwadi, "Implementasi Algoritma Physical-A Star (PHA*) Untuk Menemukan Lintasan Terpendek," *Semin. Nas. Teknol.*, vol. 2, no. 1, 2007.
- [10] J. A. Aslami, "Implementasi Algoritma A* (A Star) sebagai Dasar Pergerakan NPC (Non Player Character) Mendekati Player untuk Meningkatkan Realitas Game Pembelajaran Kosakata Bahasa Arab," *Diss. Universitas Islam Negeri Maulana Malik Ibrahim.*, 2015.
- [11] G. Snook, *Simplified 3D Movement and Pathfinding Using Navigation Mesh in Game Programming Gem I*. Mark DeLoura (ed.): Charles River Media, 2000.
- [12] F. Haryadi, "Metode Pathfinding pada Map 2D Menggunakan Algoritma Diagonal & Bidirectional BFS," *J.Strateg. Algoritm.*, vol. 2, no. 1, 2012.
- [13] A. Ernest, *Fundamentals of Game Design (2nd Edition)*. Berkeley: New Riders, 2009.
- [14] M. S. Mustaqbal, R. Firdaus, and H.Rahmadi, "Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis," *J. Ilm. Teknol. Inf. Terap.*, vol. 2, 2015.