View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Jurnal Online Universitas Katolik Parahyangan / Parahyangan Catholic University Journal

Prosiding Seminar Nasional
Riset dan Teknologi Terapan (RITEKTRA) 2021
Menuju *Society* 5.0: Teknologi Cerdas yang Berpusat pada Manusia
Bandung, 12 Agustus 2021

ISSN: 2807-999X

IMPLEMENTASI KECERDASAN BUATAN MENGGUNAKAN ALGORITMA A-STAR DAN *REPULSIVE FIELD* PADA SIMULASI GAME 3D

Akuwan Saleh[1*], Dayan Wisnu P[2]
[1,2]Program Studi Teknik Telekomunikasi, Departemen Teknik Elektro
Politeknik Elektronika Negeri Surabaya

[*]E-mail: akuwan@pens.ac.id

ABSTRAK
Kecerdasan buatan mengalami perkembangan yang pesat pada dunia komputer atau robot. Kecerdasan buatan pada komputer digunakan untuk menentukan suatu keputusan seperti manusia dan dapat berjalan otomatis guna membantu kehidupan manusia. Salah satu algoritma yang digunakan adalah A-Star untuk pencarian jalur terdekat dari titik permulaan yang diberikan sampai ke titik tujuan terdiri dari satu atau beberapa tujuan. Pada paper ini, algoritma A-Star digunakan untuk menentukan urutan pencarian jalan terdekat dengan fungsi jarak dan diterapkan di dalam obyek 3D manusia yang berjalan dalam sebuah environment game dengan 1600 lintasan kotak persegi. Agar obyek 3D bergerak otomatis dan bisa menghidari penghalang digunakan metode repulsive field. Hasil dari pengujian diperoleh nilai cost terpendek sebesar 250 dengan panjang lintasan sejumlah 23, waktu komputasi tersingkat 8,00 ms dengan lintasan sejumlah 61, obyek 3D manusia dapat mencari jalan terdekat menuju tujuan dan mampu menghindari penghalang yang ada.

Kata kunci: *kecerdasan buatan, A-Star, repulsive field, game 3D*

1. PENDAHULUAN

Perkembangan kecerdasan buatan memberi arah bahwa komputer masa depan yang dijalankan dengan program kecerdasan buatan dapat menirukan kemampuan manusia seperti menalar, mengenali pola, melakukan generalisasi, mengingat dan dapat membantu dalam perhitungan, mempermudah hidup manusia, hingga menggantikan tugas yang dilakukan manusia dalam suatu tempat berbahaya seperti pencarian korban bencana di pegunungan.

Dalam perkembangannya area tersebut dapat diaplikasikan pada simulasi animasi 3D dalam game yang dapat mewakili kondisi lingkungan pegunungan yang tidak dapat ditebak. Menurut (M Alibahtiar, dkk, 2011) animasi 3D ini juga dapat dikontrol menggunakan keyboard. Penggunaan kecerdasan buatan membuat simulasi game yang diaplikasikan pada obyek 3D dapat bergerak secara otomatis pada environment yang berbentuk pegunungan sebagai penghalang dan obyek 3D tersebut berjalan menuju ke titik tujuan dengan pemilihan jalan yang terpendek dan mampu menghindari penghalang.

Dalam pencarian jalan terpendek, proses penghitungan dapat menggunakan bermacam-macam logika seperti alogaritma djikstra, AStar dan sebagainya (Sofwan, dkk, 2009). Algoritma djikstra adalah algoritma greedy yang dipakai untuk menyelesaikan permasalahan jalan terpendek dengan graf berarah atau tak-berarah dengan nilai yang tidak negative sedangkan A-Star adalah perkembangan lebih lanjut dari alogaritma Djikstra yang diberi nilai heuristik dalam perhitungannya (Pugas, dkk, 2013).

Metode A-Star dipilih sebagai metode pencarian dalam aplikasi ini karena dinilai sebagai metode yang lebih cepat dalam menghasilkan rute terpendek dan *cost* yang efisien dengan model *obstacle* dalam aplikasi tersebut (Tilawah, 2011).

1.1. Pekerjaan Terkait

Penelitian yang dilakukan oleh (Wafiqurrahman N, 2014) tentang "Penerapan Algoritma A*(A-Star) Untuk Penentuan Rute Terpendek Pada Game Pramuka". Metode untuk path finding menggunakan A-Star yang diterapkan untuk kecerdasan *Non Player Character* dalam pencarian jalur terpendek. Game yang dibuat menganut tipe permainan adventure yaitu sebuah game petualangan dimana musuh yang datang selalu menggunakan lintasan terpendek untuk mencapai player. Jika *Non Player Character* sudah mencapai posisi target maka permainan berakhir.

Penelitian yang dilakukan oleh (Andrey Simonov, dkk, 2019) mengenai kecerdasan buatan dengan judul "Applying Behavior characteristics to decision-making process to create believable game AI", dalam makalahnya mengusulkan model pengambilan keputusan berbasis utilitas yang memberikan kemungkinan untuk menghasilkan karakter dengan perilaku yang dapat dipercaya. Model telah dirancang berdasarkan beberapa

konsep inti: siklus keputusan Observe-Orient-Decide-Act (OODA), Utility AI, pemodelan multi-agen, dan pendekatan industri game untuk menciptakan karakter dengan perilaku yang dapat dipercaya dan beragam.

Pada penelitian yang dilakukan oleh (Smolka J, dkk, 2019) tentang "A* Pathfinding Algorithm Modification for a 3D Engine" menyajikan algoritma baru, berdasarkan A*, yang lebih cocok untuk digunakan dalam mesin game 3D. Modifikasi dievaluasi dengan serangkaian tes komparatif. Algoritma A* standar digunakan sebagai patokan dalam perbandingan. Perubahan pada algoritma terdiri dari penggunaan heuristik yang berbeda, penambahan hukuman titik, dan pasca-pemrosesan jalur.

Selanjutnya makalah yang ditulis oleh (Akuwan Saleh, dkk, 2011) tentang "Model Penghindaran Tabrakan Multi Obyek Menggunakan Repulsive Field" menyajikan teknik yang mampu menampilkan gerakan penghindaran tabrakan multi obyek statis dan dinamis secara alami. Metode yang digunakan Sphere-Plane Detection (SPD) dan Sphere-Sphere Detection (SSD) untuk mendeteksi tabrakan dengan referensi jarak dan metode potential field jenis repulsive untuk penghindaran tabrakan.

1.2. Alogaritma A-STAR (A*)

A-Star adalah algoritma pencarian jalan yang paling popular dan memiliki kinerja yang terbaik (Smolka, dkk, 2019) dan algoritma paling popular dalam game *kecerdasan buatan* (Xiao, dkk, 2011). Algoritma ini menggunakan fungsi *cost* ditambah biaya biasanya di notasikan dengan f(x) untuk menentukan urutan pencarian jalan terdekat. *Cost* selalu dinotasikan dengan g(x), dimungkinkan bernilai heuristic ataupun tidak, dan sebuah kemungkinan penerimaan atas perkiraan heuristis *cost* ke titik tujuan dinotasikan dengan h(x).

$$f(x) = h(x) + g(x) \quad (1)$$

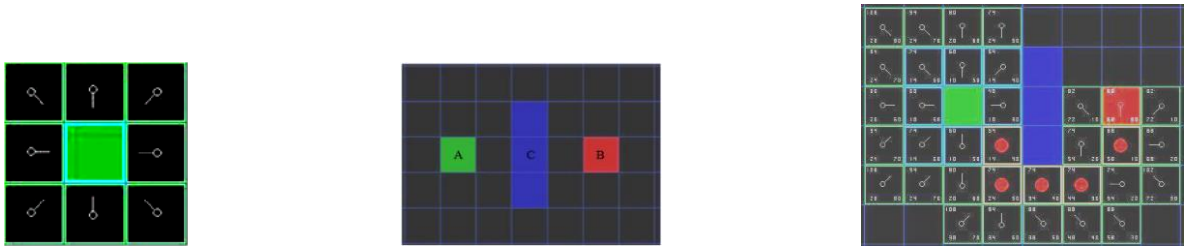
Fungsi lintasan - *cost* g(x) adalah jumlah biaya yang harus dikeluarkan dari lintasan awal menuju lintasan yang akan dituju. Dengan h(x) bagian dari fungsi f(x) yang harus dapat memperkirakan nilai heuristik, yang mana tidak diperbolehkan untuk terlalu jauh memperkirakan *cost* ke arah tujuan. Oleh karena itu untuk aplikasi seperti routing, h(x) mungkin mewakili garis lurus *cost* ke titik tujuan, karena hal ini secara nyata dimungkinkan adanya *cost* terpendek diantara dua titik yang dapat dirumuskan sebagai :

$$h(x) = \sqrt{(x - x1)^2 + (y - y1)^2} \quad (2)$$

Dimana :
x = Koordinat x dari lintasan awal
y = Koordinat y dari lintasan awal
x1 = Koordinat x dari lintasan lokasi ke n
y1 = Koordinat y dari lintasan lokasi ke n

1.3. Cara Kerja Metode A-STAR

Panjang lintasan ditentukan dari seberapa banyak lintasan yang dilalui oleh obyek 3D untuk berjalan dari titik awal menuju ke titik tujuan. Jalan dapat ditemukan dengan mencari tahu mana kotak yang harus diambil untuk mendapatkan dari satu titik ke titik yang lain. Setelah jalan ditemukan, obyek bergerak dari pusat satu persegi ke pusat berikutnya sampai target tercapai. Titik pusat/tengah ini disebut "lintasan" seperti terlihat pada Gambar 1a. Lintasan terpendek diperoleh dari nilai *cost* total. *Cost* total dapat diperolah dari panjang lintasan (*pathlength*), waktu yang dibutuhkan dan *cost* yang paling sedikit untuk menghubungkan 2 titik atau bergerak dari titik awal menuju ke titik tujuan. Ada beberapa macam persoalan pencarian lintasan terpendek yaitu lintasan terpendek antara dua buah simpul tertentu, lintasan terpendek antara semua pasangan simpul, lintasan terpendek dari simpul tertentu ke semua simpul yang lain dan lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu.



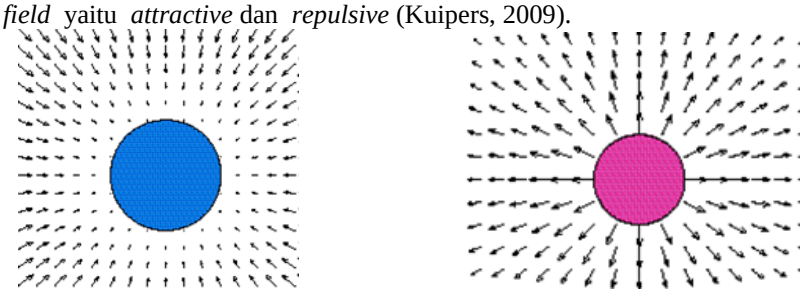
(a) Lintasan (b) Titik awal A dan tujuan B dengan penghalang C (c) Lintasan yang terpilih

Gambar 1. Penentuan lintasan

Diasumsikan obyek yang ingin menuju target dari titik A ke titik B dan dinding C memisahkan dua titik tersebut seperti pada Gambar 1b, sedangkan Gambar 1c menunjukkan proses metode A-Star dalam memilih lintasan untuk menentukan jalur terpendek (Tilawah, 2011). Jadi cara menentukan adalah hanya mulai dari target persegi merah, dan bekerja mundur bergerak dari satu persegi ke predesesornya, mengikuti panah. Hal ini pada akhirnya akan membawa obyek kembali ke pusat awal, dan itulah jalannya. Pindah dari starting persegi A ke tujuan persegi B adalah hanya masalah bergerak dari pusat setiap persegi (lintasan) ke pusat berikutnya di jalan, sampai obyek mencapai target/tujuan.

1.4. *Potential Field*

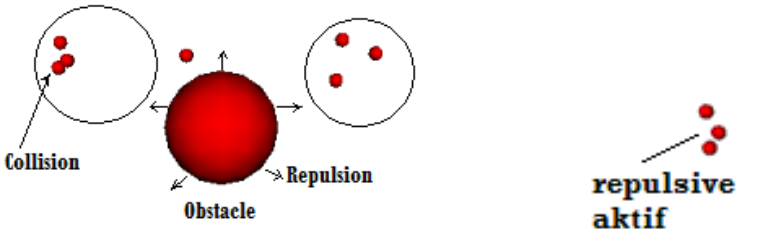
Proses penghindaran tabrakan antar obyek didisain menggunakan metode *repulsion*/tolakan. Ada dua model dari *potensial field* yaitu *attractive* dan *repulsive* (Kuipers, 2009).



(a) Model *potensial field attractive* (b) Model *potensial field repulsive*

Gambar 2. Model *potensial field*

Implementasi dari kedua model *potensial field* ini adalah: Tujuan: menggunakan model *attractive field*, Rintangan: menggunakan model *repulsive fields*, Agen: menggunakan model *repulsive fields*. Adapun contoh implementasinya diperlihatkan pada Gambar 3 (A Saleh, dkk, 2011).



(a) *Collision* antar agen (b) Tolakan antar agen

Gambar 3. *Repulsive field* pada obyek 3D

Pada Gambar 3a terjadi tabrakan antar obyek. Pemberian *repulsive field* pada tiap agen/obyek 3D seperti ditunjukan pada Gambar 3b, maka terjadi tolakan dari tiap agen yang mengakibatkan terjadi saling tolak atau tidak terjadi tabrakan karena fungsi *repulsive field* aktif. Pemberian nilai tolakan pada *repulsive field* ada di persamaan 3.

$$V_{direction} = 180°$$

$$V_{magnitude} = \{(D-d)/D \; for \; d \leq D$$

$$V_{magnitude} = \begin{cases} \frac{(D-d)}{D} & for \; d \leq D \\ 0 & for \; d > D \end{cases} \quad (3)$$

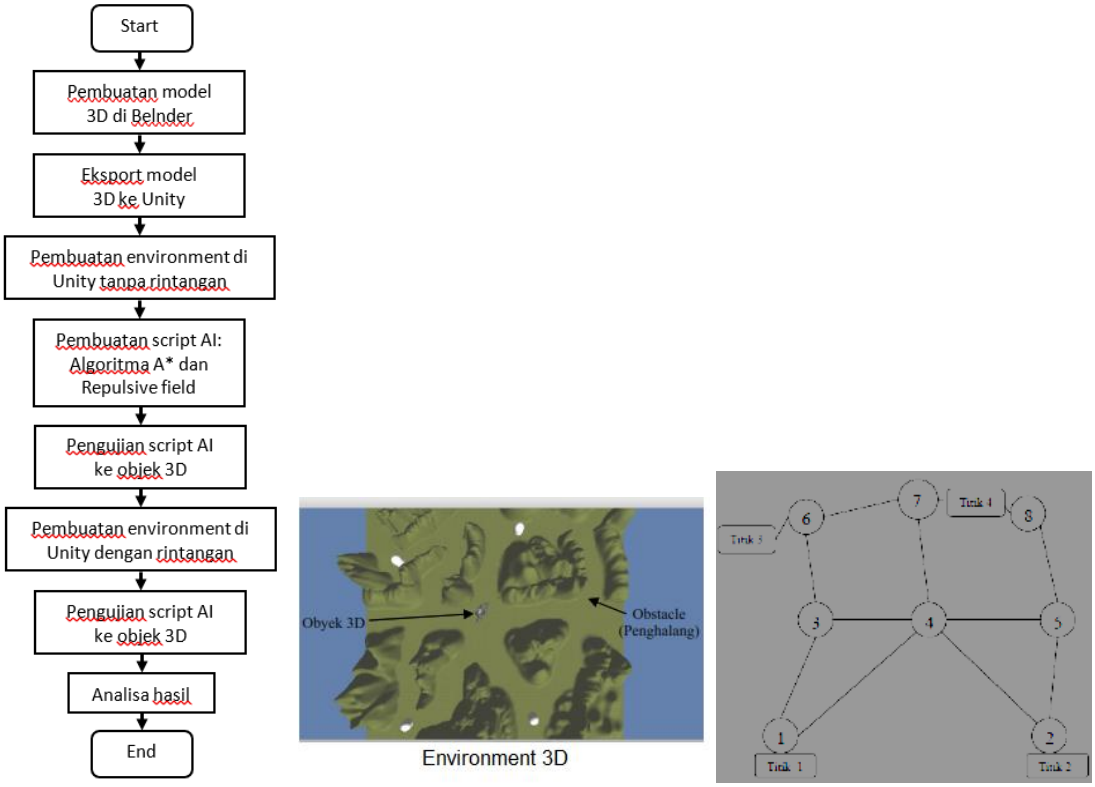
Dimana : D adalah range maksimum dari efek *field*
d adalah jarak obyek ke titik pusat $V_{direction} = 180°$

Jenis *repulsive field* pada penelitian ini diterapkan di obyek 3D berupa manusia sebagai agen dan obyek rintangan berbentuk gunung. Sehingga nilai d merupakan jarak obyek 3D manusia terhadap obyek rintangan berbentuk gunung yang diukur dari titik pusatnya. Sedangkan nilai D adalah efek tolakan maksimum yang ditimbulkan ketika kedua obyek tersebut berdekatan.

2. METODE

2.1 Rancangan Sistem

Pada perancangan sistem *scripting* kecerdasan buatan dan *environment* dibuat langsung di Unity. Langkah-langkah proses pembuatan seperti ditunjukkan didalam Gambar 4a.



(a) Proses pembuatam sistem (b) *Environment* 3D (c) Pembagian lintasan dalam *environment*

Gambar 4. Pembuatan sistem

Dari *environment* pada Gambar 4b dibuat beberapa lintasan sebagai titik acuan, dimana lintasan tersebut adalah belokan atau persimpangan seperti tersaji pada Gambar 4c. Dari lintasan-lintasan tesebut selanjutnya diukur nilai efisiensi dari metode yang digunakan. Pembuatan Obyek 3D berbentuk manusia pada sistem dibuat menggunakan software Blender. Obyek 3D berjalan dalam sebuah *environment* game yang memiliki 1600 lintasan (kotak-kotak persegi) dengan 40 lintasan horisontal dan 40 lintasan vertikal dengan penghalang statis berupa pegunungan. Untuk perkiraan jarak dibuat skala 1:100 dengan 1 lintasan (kotak persegi) yang bergerak horisontal atau vertikal mewakili 1meter sedangkan yang diagonal 1,4 meter, sehingga diperoleh luas *environment* adalah 1600 meter persegi. Jarak diagonal dengan mengambil asumsi bahwa mekanisme game memungkinkan agen/objek 3D untuk bergerak secara horisontal, vertikal, maupun diagonal (Julio, dkk, 2019).

2.2. Penerapan Alogaritma A-Star

Penemuan jalur yang cerdas untuk rute yang baik dapat diterapkan dalam game apa pun (Yee, dkk,2004). Seperti pada kebanyakan algoritma pencarian informasi, terlebih dahulu dicari rute yang tampaknya mempunyai kemungkinan besar untuk menuju ke arah tujuan. Algoritma ini mengambil jarak perjalanan ke arah tujuan (dimana g (x) bagian dari heuristis adalah *cost* dari awal, dan tidak sekedar menjadi *cost* lokal sebelum lintasan diperluas). Beberapa terminologi dasar yang terdapat pada algoritma ini adalah starting point, lintasan, open list, closed list, bobot (*cost*), halangan (*unwalkable*) dengan penjelasan sebagai berikut :

1. Starting point adalah sebuah terminologi untuk posisi awal sebuah benda.
2. A adalah lintasan yang sedang dijalankan dalam algortima pencarian jalan terpendek.
3. Lintasan adalah petak-petak kecil sebagai representasi dari area *pathfinding*.
4. Open list adalah tempat menyimpan data lintasan yang mungkin diakses dari starting point maupun lintasan yang sedang dijalankan.
5. Closed list adalah tempat menyimpan data lintasan sebelum A yang juga merupakan bagian dari jalan terpendek yang telah berhasil didapatkan.

6. Bobot (F) adalah nilai yang diperoleh dari penjumlahan, nilai G merupakan jumlah nilai tiap lintasan dalam jalan terpendek dari starting point ke A, dan H adalah jumlah nilai perkiraan dari sebuah lintasan ke lintasan tujuan. Sehingga dapat diformulasikan f(x)= h(x) + g(x).
7. Lintasan tujuan yaitu lintasan yang dituju.
8. Halangan adalah sebuah atribut yang menyatakan bahwa sebuah lintasan tidak dapat dilalui oleh A.

Alogaritma A-Star menggunakan gabungan jarak dan bobot dimana harus memperkirakan nilai heuristik, yang mana tidak diperbolehkan untuk terlalu jauh memperkirakan jarak ke arah tujuan.
Penentuan rute terpendek ini didasarkan pada panjang lintasan yaitu jumlah kotak persegi yang dilalui dari titik awal obyek berada menuju target dan *cost* yaitu jumlah bobot yang diperlukan dari titik awal obyek berada menuju target berdasarkan panjang lintasan yang telah didapat. Untuk itu ditetapkan bobot 10 sampai masing-masing persegi horizontal atau vertikal bergerak, dan bobot 14 untuk bergerak diagonal. Jarak yang sebenarnya untuk bergerak diagonal adalah 14,14 yang diperoleh dari :

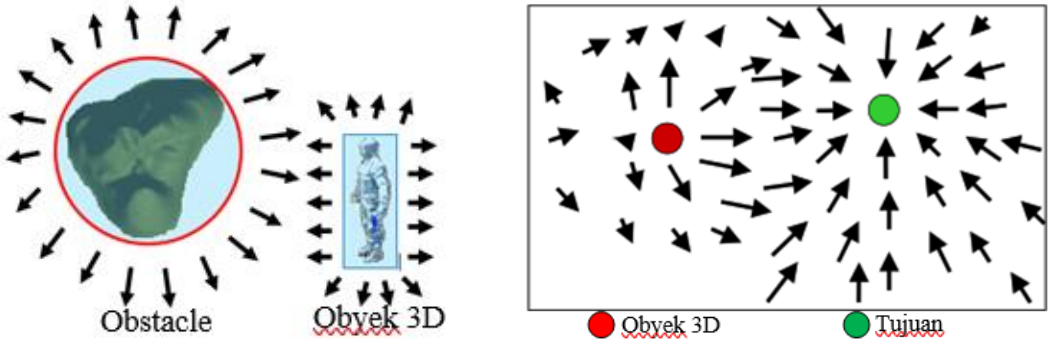
$$cos\ cos\ t\ \_diagonal = \sqrt[2]{cos\ cos\ t\ \_vertical + cos\ cos\ t\ \_horizontal}$$
$$cos\ cos\ t\ \_diagonal = \sqrt[2]{cos\ cos\ t\ \_vertical + cos\ cos\ t\ \_horizontal} \quad (4)$$

Untuk nilai *cost* total didapatkan dari :

$$t\ \_Total = cost\_diagonal + cost\_vertical + cost\_horizontal$$
$$Cost\_Total = cos\ cos\ t\ \_diagonal + cos\ cos\ t\ \_vertical + cost\_horizontal \quad (5)$$

2.3. Penerapan *Repulsive Field*

Penghindaran tabrakan antar obyek menggunakan metode *repulsive field* pada penghalang statis berbentuk pegunungan dan obyek 3D berbentuk manusia diperlihatkan pada Gambar 5.



Gambar 5. Model *repulsive field* obstacle vs obyek 3D dan tujuan vs obyek 3D

Penjelasan Gambar 5, ketika obyek 3D manusia berdekatan dengan penghalang pegunungan akan terjadi saling tolak sehingga tidak terjadi tabrakan antar keduanya. Sedangkan ketika obyek 3D manusia berdekatan dengan titik tujuan dengan jenis *attractive field*, maka terjadi tarikan yang sangat kuat sehingga terjadi kecenderungan gerakan menuju ke tujuan.

3. HASIL DAN PEMBAHASAN

3.1. Pencarian Rute Terpendek

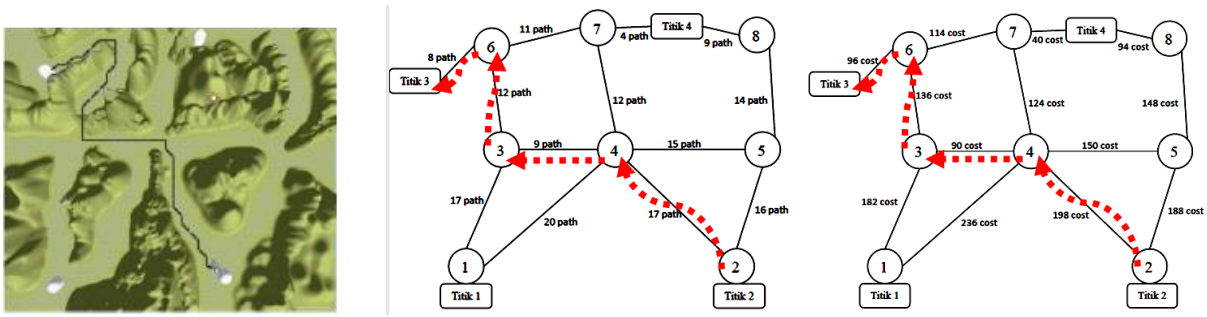
Hasil pengujian pencarian rute terpendek ditujukan pada Tabel 1 dan Gambar 6. Hasil perhitungan nilai *cost_diagonal* dan *cost_total* menggunakan persamaan (4) dan (5). Berdasarkan data ke-5 pada Tabel 1, hasil pergerakan obyek 3D manusia seperti pada Gambar 6a dari titik asal 2 menuju titik target/tujuan 3 dengan menghindari penghalang. Rute perjalanan dapat melewati beberapa rute, diantaranya melalui titik 2-4-7-6-T3 dengan *cost* sebesar 532, titik 2-5-8-T4-7-6-T3 dengan *cost* sebesar 680, titik 2-4-1-3-6-T3 dengan *cost* sebesar 848 dan titik 2-5-4-3-6-T3 dengan *cost* sebesar 660. Terpilih rute perjalanan melalui lintasan 2-4-3-6-T3 dengan *cost* sebesar 520 adalah yang paling dekat dibandingkan dengan rute perjalan yang lainnya.

ISSN: 2807-999X

Tabel 1. Nilai *cost* dan panjang lintasan (*Pathlength)* antar titik

No	Titik Asal	Titik Target	Lintasan Terpendek	*Path Length*	*Cost* Vertikal	*Cost* Horizontal	*Cost* Diagonal	*Cost* Total
1	1	2	1-4-2	37	170	40	224	434
2	1	3	1-3-6-T3	37	220	40	154	414
3	1	4	1-4-7-T4	36	190	70	140	400
4	2	1	2-4-1	37	170	40	224	434
5	2	3	2-4-3-6-T3	46	170	140	210	520
6	2	4	2-4-7-T4	33	200	50	112	362
7	3	1	T3-6-3-1	37	220	40	154	414
8	3	2	T3-6-3-4-2	46	170	140	210	520
9	3	4	T3-6-7-T4	23	130	50	70	250
10	4	1	T4-7-4-1	36	190	70	140	400
11	4	2	T4-7-4-2	33	200	50	112	362
12	4	3	T4-7-6-T3	23	130	50	70	250

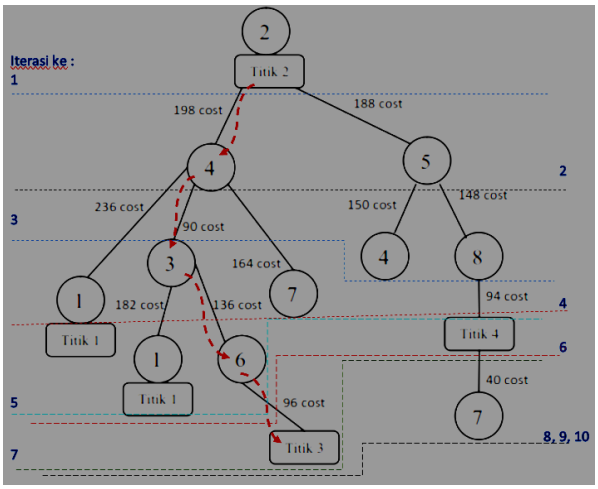
Keterangan: T = Titik



(a). Gerakan obyek 3D, (b). Lintasan dengan nilai *pathlength*, (c). Lintasan dengan nilai *cost*

Gambar 6. Hasil rute titik asal 2 menuju titik tujuan 3

Pada Gambar 6b dan c, menunjukan hasil nilai *cost* yang dibutuhkan untuk mencapai titik tujuan sebesar 520 *cost* dengan nilai *pathlength* 46 melalui rute 2-4-3-6-T3. Perbandingan skala yang digunakan adalah 1:100 dengan 1 lintasan (kotak persegi) yang bergerak horizontal atau vertikal mewakili 1 meter sedangkan yang diagonal 1,4 meter, maka didapatkan jarak tempuh obyek 3D sejauh ((520 x 100) : 10) = 52 meter. Proses iterasi pencarian rutenya adalah sebagai berikut:



Gambar 7. Iterasi pencarian rute terpendek dari titik 2 ke 3

Lintasan terpilih iterasi-1, pada awalnya status lintasan yang berada dalam daftar terbuka / belum terpilih akan diinisialisasikan dengan '0' dan yang sudah terpilih / dalam daftar tertutup diinisialisasikan dengan '1'. *Cost* ditentukan dari lintasan yang berhubungan langsung. Dalam hal ini yaitu dari titik 2 yang berada pada lintasan 2 yang kemudian akan dilanjutkan hingga mencapai titik 3. Ada 2 lintasan tujuan selanjutnya, sementara *cost* terendah adalah melewati lintasan 5. Untuk itu lintasan 5 statusnya menjadi 1, namun lintasan 4 tidak langsung bernilai 1 yang berarti daftar tertutup, karena program belum men-scan semua lintasan hingga menuju tujuan. Untuk predesesor keduanya sama-sama 2 karena langsung berasal / berhubungan dengan lintasan 2.

Lintasan terpilih iterasi-2, Jarak ke lintasan 5 = 188 *cost*, dengan predesesor 2. Ketika lintasan 5 telah tepilih, maka dilanjutkan ke lintasan yang berhubungan yaitu lintasan 4 dan 8.

Lintasan terpilih iterasi-3, Jarak ke lintasan 4 = 198 *cost* dengan predesesor 2, jauh lebih pendek dibanding melalui predesesor 5. Selanjutnya lintasan 4 telah tepilih, maka akan dilanjutkan ke lintasan yang berhubungan dengan lintasan 4 yaitu lintasan 1, 3 dan 7.

Lintasan terpilih iterasi-4, selanjutnya lintasan 3 telah tepilih dengan *cost* yang berubah yaitu :

Cost lintasan 3 = *cost* lintasan 4 + *cost* jalan 3 = 198 + 90 = 288

Dengan predesesor lintasan 3 yaitu lintasan 4. Kemudian akan dilanjutkan ke lintasan yang berhubungan dengan lintasan 3 yaitu lintasan 1 dan 6

Lintasan terpilih iterasi-5, lintasan 8 dipilih karena *cost* nya lebih rendah daripada *cost* lintasan 6 dan lintasan 1 yaitu :

Cost lintasan 8 = *cost* lintasan 5 + *cost* jalan 8 = 188 + 148 = 336

Lintasan terpilih iterasi-6, Titik 4 dimasukkan dalam daftar terbuka, namun karena *cost* nya lebih tinggi daripada *cost* lintasan 6 maka lintasan 6 yang tepilih dengan predesesor lintasan 6 yaitu lintasan 3 dengan *cost* yang berubah yaitu :

Cost lintasan 6 = *cost* lintasan 3 + *cost* jalan 6 = 288 + 136 = 424

Lintasan terpilih iterasi-7, Kemudian akan dilanjutkan ke lintasan yang berhubungan dengan lintasan 6 yaitu lintasan 7 dan target tujuan yaitu titik 3. Selanjutnya dari lintasan 6 yang berhubungan langsung dengan lintasan 7 dan titik 3. Karena tujuan obyek adalah menuju ke target yang berupa titik 3 namun dihitung berdasarkan *cost* terendah dari titik sebelumnya maka titik 4 lah yang terpilih :

Cost titik 4 = *cost* lintasan 8 + *cost* jalan titik 4 = 336 + 94 = 430

Lintasan terpilih iterasi-8, Setelah titik 4 terpilih yang terbuka selanjutnya adalah lintasan 7 yang berhubungan langsung dengan titik 4. Namun lintasan 1(titik 1) memiliki *cost* yang lebih sedikit sehingga yang terpilih menjadi lintasan 1 dengan *cost* lintasan 1 sekarang yaitu :

Cost lintasan 1 = *cost* lintasan 4 + *cost* jalan lintasan 1 = 198 + 236 = 434

Namun saat lintasan 1 terpilih tidak ada daftar terbuka atau lintasan dengan *cost* yang lebih sedikit sehingga kembali titik 4 yang aktif.

Lintasan terpilih iterasi-9, dari titik 4 menuju ke lintasan 7 dengan predesesor titik 4. Lintasan 7 berubah menjadi terpilih dengan *cost* sebesar :

Cost lintasan 7 = *cost* titik 4 + *cost* jalan lintasan 7 = 430 + 40 = 470

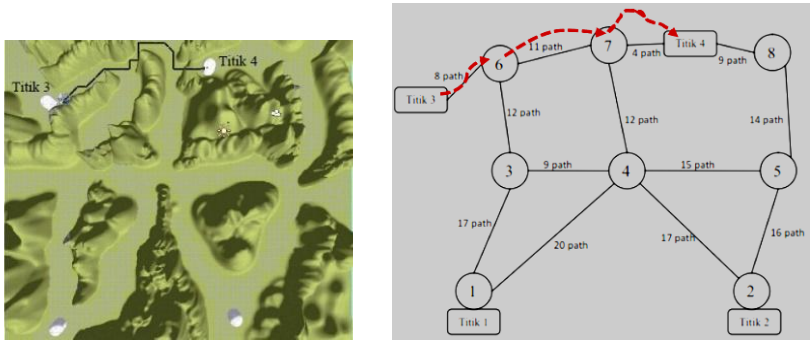
Lintasan terpilih iterasi-10, karena lintasan yang berhubungan langsung dengan lintasan 7 tidak ada yang berada dalam daftar terbuka, maka pencarian langsung mengarah ke titik 3 / target dengan predesesor lintasan 6 dengan *cost* total yaitu : *Cost* Total (titik 3) = *cost* lintasan 6 + *cost* jalan titik 3 = 424 + 96 = 520

Maka *cost* yang dibutuhkan untuk mencapai titik tujuan sebesar 520 *cost* melalui rute 2-4-3-6-T3. Dikarenakan perbandingan skala yang digunakan adalah 1:100 dengan 1 lintasan (kotak persegi) yang bergerak horizontal atau vertikal mewakili 1 meter sedangkan yang diagonal 1,4 meter, maka didapatkan jarak tempuh obyek 3D sejauh ((520 x 100) : 10) = 52 meter. Rute perjalanan dari titik 2 menuju titik 3 sebenarnya dapat melewati beberapa cara,diantaranya adalah melalui :

a) Titik 2-4-7-6-T3 dengan *cost* sebesar 532.
b) Titik 2-5-8-T4-7-6-T3 dengan *cost* sebesar 680.
c) Titik 2-4-1-3-6-T3 dengan *cost* sebesar 848.
d) Titik 2-5-4-3-6-T3 dengan *cost* sebesar 660.

Namun, rute perjalanan melalui lintasan 2-4-3-6-T3 dengan *cost* sebesar 520 adalah yang paling dekat dibandingkan dengan rute perjalan yang lainnya. Rute sama akan dilalui titik 3 ke titik 2 dengan melalui lintasan T3-6-3-4-2 dengan *cost* 520, jarak tempuh 52 meter.

Hasil pengujian yang lain yaitu pergerakan obyek 3D manusia dari titik asal 3 menuju titik target/tujuan 4 seperti pada Gambar 8.



Gambar 8. Gerakan obyek 3D dari titik 3 ke titik 4

Diperoleh rute terpendek dari titik 3 ke titik 4 melalui lintasan T3-6-7-T4 dengan *cost* 250 dan panjang lintasan (*pathlength*) 23. Maka jarak tempuh obyek 3D tersebut adalah ((250 x 100) : 10) = 25 meter. Rute sama akan dilalui titik 4 ke titik 3 dengan melalui lintasan T4-7-6-T3 dengan *cost* 250, jarak tempuh 25 meter.

3.2. Pengujian Lama Waktu Komputasi

Pada bagian ini dilakukan pengamatan terhadap lama komputasi yang dibutuhkan oleh sistem untuk mencari rute terdekat. Lamanya waktu komputasi antar rute tidak terlalu berbeda signifikan, namun waktu komputasi juga tetap bergantung kepada *pathlength*, *cost rute* dan pencarian lintasan (*searched path*) yang dilakukan setiap kali obyek bergerak menuju titik tujuan / target. Hasil pengujiannya menggunakan laptop dengan spesifikasi processor Intel Pentium Dual-Core 2.0GHz, dengan memory DDR2-SDRAM 2GB, dan VGA NVIDIA GEFORCE G102M diperlihatkan pada Tabel 2.

Tabel 2. Data Lama Waktu Komputasi

No	Titik Asal	Titik Target	Path Length	*Cost* Total	*Searched Path*	Waktu Komputasi (milisecond)
1	1	2	37	434	246	10,00
2	1	3	37	414	248	10,00
3	1	4	36	400	281	10,00
4	2	1	37	434	205	10,00
5	2	3	46	520	345	12,99
6	2	4	33	362	183	10,00
7	3	1	37	414	133	9,00
8	3	2	46	520	264	10,00
9	3	4	23	250	78	8,00
10	4	1	36	400	128	9,00
11	4	2	33	362	90	8,00
12	4	3	23	250	61	8,00

Pencarian lintasan (*searched path*) adalah banyaknya lintasan yang discan setiap rutenya untuk mengetahui rute terdekat yang dibutuhkan dari titik awal menuju target. Banyaknya *searched path* tergantung dari rintangan dan kemungkinan jalan dalam *environment*. Dari Tabel 2, dapat dilihat bahwa semakin banyak pencarian lintasan maka waktu komputasi yang diperlukan semakin lama walaupun perbedaannya hanya dalam hitungan *milisecond*. Rata-rata waktu yang diperlukan saat komputasi hanya sekitar 10,00 ms.

4. KESIMPULAN

Berdasarkan hasil dan pembahasan di atas dapat diambil kesimpulan sebagai berikut :

1. Nilai *cost* terpendek adalah 250 dengan Panjang lintasan (*pathlength*) sejumlah 23 dari titik 3 menuju titik 4 dengan jarak tempuh 25 meter.
2. Nilai *cost* terpanjang yaitu sebesar 520 dengan *pathlength* sejumlah 46 dari titik 2 ke titik 3.
3. Waktu komputasi terlama yaitu dari titik 2 menuju titik 3 dengan lintasan yang dicari sejumlah 345 lintasan dengan waktu komputasi 12,99 ms.

4. Waktu komputasi tersingkat yaitu ketika obyek 3D bergerak dari titik 4 menuju titik 3 dengan lintasan yang dilalui sejumlah 61 lintasan, waktu komputasi 8,00 ms.
5. Aplikasi ini dapat menghasilkan rute terpendek dengan jumlah pathlength dan *cost* terpendek yang dilalui obyek 3D berupa manusia.
6. Pergerakan obyek 3D manusia dengan AI berhasil mencapai tujuan dengan menempuh jalur terpendek serta dapat menghidari penghalang pada *environment game*.

Keberhasilan dari penelitian ini menjadi dasar penelitian selanjutnya mengenai aplikasi *speech recognition* sebagai sistem pengendali gerakan obyek animasi dan penghindaran tabrakan obyek 3D pada virtual world menggunakan repulsive field berbasis *augmented reality* (AR).

DAFTAR PUSTAKA,

Akuwan Saleh, M Hariadi, S Mardi, 2011, *Model Penghindaran Tabrakan Multi Obyek Menggunakan Repulsive Field*, Industrial Electronics Seminar (IES), pp.388-394, ISBN: 978-979-8689-14-7.

Julio C Y, Alethea S, Richard L, 2019. *Review of Various A* Pathfinding Implementations in Game Autonomous Agent*, International Journal Of New Media Technology (IJNMT), Vol. VI, No. 1. pp.43-49, doi: https://doi.org/10.31937/ijnmt.v6i1.1075.

Kuipers Benjamin, 2009, *Lecture 7: Potential Fields and Model Predictive Control*, CS 344R: Robotics.

M Alibahtiar, Akuwan Saleh, MA Zainudin, 2011, *Sistem Augmented Reality Untuk Animasi Games Menggunakan Camera Pada PC*, EEPIS Final Project, EEPIS, Surabaya.

Pugas D.O, Somantri M, Satoto K.I, 2011. *Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra Dan Astar (A*) Pada SIG Berbasis Web Untuk Pemetaan Pariwisata Kota Sawahlunto*. Jurnal Ilmiah Teknik Elektro (TRANSMISI), Universitas Diponegoro, Semarang, Vol 13, No.1. pp.27-32, doi: 10.12777/transmisi.13.1.27-32.

Smolka J, Miszta K, Paszkowska MS, and Lukasik E, 2019. *A* Pathfinding Algorithm Modification for a 3D Engine*, International Conference of Computational Methods in Engineering Science (CMES'18), MATEC Web Conf. Vol. 252, 03007, pp.1-6, doi: https://doi.org/10.1051/matecconf/201925203007

Sofwan A, Isnanto RR, Maulana P, 2009. *Kecerdasan Buatan Dalam Permainan Snake 3d Menggunakan Visual Basic .Net Dan Directx*. Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro, Semarang

Tilawah H, 2011. *Penerapan Algoritma A-star (A*) Untuk Menyelesaikan Masalah Maze*, Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, Bandung

Xiao Cui and Hao Shi, 2011. *A*-based Pathfinding in Modern Computer Games*, International Journal of Computer Science and Network Security, Vol.11 No.1. pp.125-130.

Yee Chia Hui, Edmond C Prakash and Narendra S C, 2004. *Game AI: Artifical Intelligence For 3d Path Finding*, IEEE Region 10 Conference TENCON 2004, pp.306-309, doi: 10.1109/TENCON.2004.1414592.

Wafiqurrahman, Naufal, 2014. *Penerapan Algoritma A*(A-Star) Untuk Penentuan Rute Terpendek Pada Game Pramuka*, Skripsi Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim, Malang.

Andrey Simonov, Aleksandr Zagarskikh, Victor Fedorov, 2019. *Applying Behavior characteristics to decision-making process to create believable game AI*, 8th International Young Scientist Conference on Computational Science, Procedia Computer Science 156 (2019) 404–413, Published by Elsevier Ltd.