

Implementasi Algoritma A^* (*A Star*) Dalam Penentuan Rute Terpendek Yang Dapat Dilalui *Non Player Character* Pada *Game Good Thief*

Rifky Fajar Oktanugraha¹, Salamun Rohman Nudin²

^{1,2} Program Studi S1 Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya
rifyoktanugraha16051204020@mhs.unesa.ac.id
salamunrohman@unesa.ac.id

Abstrak—*Game* merupakan salah satu media dimana didalamnya terdapat sebuah makna dan seiring berjalannya waktu, pengguna akan mengetahui makna yang disampaikan oleh si pembuat *game* dari *game* yang dimainkan. Algoritma diperlukan untuk membuat *game* lebih menarik. Hal ini berbanding lurus dengan jumlah pengguna, semakin menarik *game* yang dimainkan, semakin banyak orang yang akan memainkan *game* tersebut. Algoritma A^* (*A Star*) merupakan algoritma yang digunakan untuk menentukan rute terpendek objek menuju ke tujuan, dengan menghitung harga yang harus dipakai dan mencari harga terkecil yang harus dibayarkan. Pada penelitian ini akan dibuat sebuah *game* dengan nama *Good Thief* dimana *game* ini bercerita tentang seorang pencuri baik yang dikejar oleh penjaga. Algoritma A^* (*A Star*) akan digunakan untuk menentukan rute terpendek yang dapat dilalui oleh NPC. Algoritma A^* (*A Star*) akan diterapkan di setiap NPC dari Tingkat kesulitan mudah (LV 1) hingga tingkat kesulitan sulit (LV 3) dilakukan beta testing untuk mengetahui apakah *game* dapat berjalan dengan baik di setiap levelnya, hasil yang didapatkan dari 6 responden yang ikut dalam pengujian *game* ini cukup memuaskan yaitu sebesar 83,22%

Kata Kunci— *Game*, Media, Algoritma, A^* (*A Star*), NPC, Rute Terpendek.

I. PENDAHULUAN

Game adalah permainan yang menggunakan media elektronik, merupakan sebuah hiburan berbentuk multimedia yang di buat semenarik mungkin agar pemain bisa mendapatkan sesuatu sehingga adanya kepuasan batin [1]. Jumlah *gamer mobile* di Tanah Air mencapai 60 juta. Jumlah tersebut diperkirakan akan meningkat menjadi 100 juta pada 2020. Hasil studi bahkan menunjukkan, mayoritas aktivitas yang dilakukan masyarakat melalui *smartphone* yaitu bermain *game* (25%). Mereka rata-rata bermain *game mobile* dengan durasi 53 menit. Aktivitas lainnya yang dilakukan pengguna dengan *smartphone* yaitu bersosial media (17%), *streaming video* (12%), *browsing* (10%), hingga *berbelanja online* (7%) [2].

Algoritma diperlukan untuk membuat *game* lebih menarik. Hal ini berbanding lurus dengan jumlah pengguna, semakin menarik *game* yang dimainkan, semakin banyak orang yang akan memainkan *game* tersebut. Di dalam sebuah *game* algoritma biasa digunakan untuk menentukan level, tingkat kesulitan, skor/nilai, hingga sebagai pengambil keputusan

dalam *game*. Algoritma A^* (*A Star*) merupakan algoritma yang digunakan untuk menentukan rute terpendek objek menuju ke tujuan, dengan menghitung harga yang harus dipakai dan mencari harga terkecil yang harus dibayarkan.

Penelitian terdahulu tentang pengimplementasian A^* (*A Star*) yaitu oleh Dedy & Sepri pada tahun 2018 dengan judul penelitian Penerapan Algoritma A^* (*A Star*) Sebagai Pencari Rute Terpendek pada Robot Hexapod. Pada penelitian tersebut, algoritma A^* (*A Star*) digunakan untuk pencarian rute terpendek sebuah robot. akan disediakan titik tujuan yang akan menjadi tujuan robot tersebut. Penelitian dilakukan dengan dua skenario, skenario pertama tanpa sebuah halangan dan langsung ke tujuan, sedangkan skenario kedua adalah dengan menerapkan sebuah halangan. Karena terpengaruh bentuk robot, maka hasilnya sedikit berbeda, dimana hasil tanpa penghalang lebih akurat daripada dengan penghalang [3].

Penelitian lainnya oleh Kiki, Supriyadin, Imam, & Roy pada tahun 2018 dengan judul menghitung rute terpendek menggunakan algoritma A^* (*A Star*) dengan fungsi *euclidean distance*, pada penelitian ini algoritma A^* (*A Star*) digunakan untuk pencarian rute terpendek, akan ada dua tempat yang menjadi titik awal dan tujuan, A^* (*A Star*) digunakan untuk pencarian rute terpendek dan terefisien [4].

Pada tahun 2017 dilakukan penelitian oleh Imam & Wahyu dengan judul Penerapan Algoritma A^* (*A Star*) pada *Game* Petualangan Labirin Berbasis Android. Di penelitian ini *game* yang dibuat adalah labirin, dan memiliki misi untuk membawa kelinci menuju ke makanannya. 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 [5].

Penelitian lainnya yang terkait dengan implementasi A^* (*A Star*) adalah penelitian yang dilakukan oleh Aditya, Paulus, & Sri pada tahun 2019 dengan judul penelitian Penerapan *Pathfinding* Menggunakan Algoritma A^* (*A Star*) Pada *Non Player Character* (NPC) Di *Game*. Pada *game* yang dibuat terdapat tokoh *enemy* yang akan menjadi *Non Player Character* (NPC) dengan menggunakan area hutan yang luas. Dimana pada nantinya NPC dirancang agar dapat mencari

target utama / pemain dengan rute terpendek dan dapat menghindari penghalang yang ada di area *game* [6].

Penelitian terdahulu lainnya adalah oleh Nawaf, Sinan, & Mustafa pada tahun 2016 dengan judul penelitian *Pathfinding in Strategy Games and Maze Solving Using A* Search Algorithm*. Dalam penelitian ini, algoritma pencarian A^* (*A Star*) digunakan untuk menemukan jalur terpendek antara titik awal dan tujuan pada gambar yang mewakili peta atau labirin. Menemukan jalan melalui labirin adalah masalah ilmu komputer dasar yang dapat mengambil banyak bentuk. Algoritma A^* (*A Star*) banyak digunakan dalam *pathfinding* dan *graph traversal*. Gambar peta dan labirin yang berbeda digunakan untuk menguji kinerja sistem (100 gambar untuk setiap peta dan labirin). Kinerja keseluruhan sistem dapat diterima dan dapat menemukan jalur terpendek antara dua titik pada gambar. Lebih dari 85% gambar dapat menemukan jalur terpendek antara dua titik yang dipilih [7].

Penelitian lain dilakukan oleh Sharma & B.L.Pal pada tahun 2015 Dengan judul penelitian *Shortest Path Searching for Road Network using A* Algorithm* dengan tujuan mengetahui lalu lintas dan jalan tercepat kemudian peneliti membandingkan Dijkstra dengan algoritma A^* (*A Star*) untuk menunjukkan kinerjanya dalam berbagai perbedaan berhenti berdasarkan parameter yang berbeda - berbeda [8].

Pada penelitian ini akan dikembangkan sebuah *game* dengan judul *game* yaitu *Good Thief*, dimana *game* ini bercerita tentang seorang pencuri baik yang mencoba membantu memenuhi kebutuhan panti asuhan tempat dia berada dan algoritma A^* (*A Star*) akan diterapkan ke Penjaga (NPC). Di dalam *game* yang dikembangkan, penjaga (NPC) memiliki peran untuk mengikuti pencuri (Pemain) secara terus menerus hingga pencuri (Pemain) berhasil ditangkap. Algoritma A^* (*A Star*) akan membantu penjaga (NPC) dalam memilih rute terpendek yang bisa dilalui dalam menangkap pencuri (Pemain).

II. METODOLOGI PENELITIAN

A. Skenario Permainan

Algoritma A^* (*A Star*) pada pembuatan *game* ini, diterapkan pada NPC bertujuan untuk menemukan rute terpendek menuju ke tujuan. *Game* yang akan dibuat berbasis desktop dan bersifat *single player*. Genre yang diambil adalah petualangan dikarenakan *game* menceritakan tentang petualangan seorang pencuri untuk menghidupi adik – adiknya di panti asuhan. Arena pada *game* akan dibagi menjadi 3, yaitu mudah, sedang dan sulit. Pada penelitian ini, *game* yang dibuat menceritakan perjalanan si pencuri dalam melakukan pencurian. Dikarenakan kebutuhan panti asuhan semakin lama semakin meningkat, maka si pencuri diharuskan untuk mencuri demi memenuhi kebutuhan panti asuhan milik pencuri.

Tugas utama pemain adalah berperan sebagai si pencuri yang bertugas untuk mencuri uang di setiap rumah yang ada, si pencuri akan melakukan pencurian sesuai dengan kebutuhan dari panti asuhannya. Semakin banyak

kebutuhannya, semakin sering pula si pencuri melakukan pencurian.

Alur permainan tiap tingkat kesulitan dalam *game Good Thief* adalah sebagai berikut :

1. Arena Mudah

Dalam kategori mudah, pemain diberi tugas untuk memenuhi kebutuhan panti asuhan sebesar 1000 *gold*. Dimana bisa dipenuhi dengan mencuri *gold* dari rumah – rumah yang disediakan. Pada arena mudah akan disediakan tiga rumah dengan masing – masing *gold* yang dimiliki adalah 500. Pemain akan berkeliling kota bersamaan dengan menghindari penjaga yang berkeliling. Pemain akan memasuki rumah yang disediakan dan keluar dari pintu lain yang tersedia, dikarenakan objek rumah pada *game* memiliki 2 pintu, masing – masing pintu masuk dan keluar. pintu masuk dan pintu keluar tidak ditentukan, jadi pemain bebas memilih untuk masuk melewati pintu mana sesuai dengan strategi pemain. Pemain tidak akan bisa memasuki rumah ketika dalam kondisi membawa *gold*, pemain diharuskan membawa *gold* yang didapatkan kembali ke panti asuhan terlebih dahulu dan mulai pergi mencuri lagi hingga kebutuhan panti asuhan terpenuhi. Saat arena mudah dimulai, pemain akan memiliki 3 nyawa dan akan berkurang jika tertangkap oleh NPC yang bertugas menjadi seorang penjaga. Saat tertangkap, pemain akan dipindahkan ke titik *spawn* di depan objek penjara sebagai tempat awal memulai permainan kembali. Saat tertangkap, apabila pemain membawa *gold*, maka *gold* akan dihilangkan kembali menjadi 0. Algoritma A^* (*A Star*) akan diterapkan pada NPC yang terdapat di dalam *game Good Thief*, tujuan yang di *setting* pada algoritma A^* (*A Star*) adalah pemain, sehingga NPC akan berusaha menuju lokasi pemain dengan mencari rute terpendek yang seolah olah akan menangkap pemain yang berperan sebagai pencuri. Sepanjang permainan, pemain akan memiliki berbagai macam kemampuan, dimana kemampuan tersebut akan mempengaruhi *gameplay* dari *game Good Thief*. Ada 3 objek bantuan yang dapat dimanfaatkan pemain selama permainan berlangsung, yaitu roti, tali, dan kantung harta. Dalam cerita, dikatakan bahwa si pencuri menyukai roti, sehingga apabila pemain mengambil roti yang terdapat di arena, maka kecepatan pencuri dapat meningkat 100% selama 5 detik. Objek bantuan tali yang terdapat di arena bisa digunakan pemain untuk melumpuhkan penjaga, setelah pemain mengambil tali, maka penjaga akan terdiam selama 5 detik. Objek bantuan kantung uang akan memberikan pencuri 100 *gold* secara instan, dengan kapasitas kantung si pencuri adalah 200. Permainan dikatakan selesai apabila kebutuhan panti asuhan 1000 *gold* terpenuhi, dan akan berakhir saat nyawa si pencuri menjadi 0. Setelah

pemain memenuhi kebutuhan panti asuhan, maka pemain akan dipindahkan ke arena permainan sedang.

2. Arena sedang

Skenario di arena sedang hampir sama dengan arena mudah. Perbedaan yang terdapat di arena sedang adalah jumlah kebutuhan panti asuhan yang pada arena mudah sebesar 1000 *gold* dan pada arena sedang akan meningkat menjadi 2000 *gold*. Terdapat 5 objek pada arena sedang yang dikategorikan sebagai rumah dengan masing – masing rumah memiliki *gold* sebesar 500. Pemain bisa memenuhi kebutuhan hanya dengan mencuri pada 4 rumah saja, rumah mana saja yang akan diambil akan diserahkan kepada pemain dalam pengambilan keputusan. Satu NPC penjaga akan ditambahkan, namun dengan karakteristik yang berbeda, yaitu kecepatan penjaga akan satu tingkat di bawah pemain, jika kecepatan pemain 10, maka kecepatan penjaga ini adalah 9.

3. Arena sulit

Pada arena sulit, kebutuhan panti asuhan yang sebelumnya 2000 *gold* akan ditingkatkan menjadi 3000 *gold*. Semua *gold* akan tersebar pada rumah – rumah yang terletak di arena. Rumah yang disediakan pada arena sulit adalah 6 rumah, dengan masing – masing rumah memiliki 600 *gold* yang siap untuk dicuri. Jumlah objek bantuan juga akan ditambahkan sesuai kebutuhan pemain. Total penjaga pada arena sulit adalah 3 penjaga dengan karakteristik 1 penjaga normal, 1 penjaga kecepatan ditambah dan 1 penjaga yang memiliki kemampuan menghilang.

B. Algoritma A* (A Star)

Algoritma A* (A Star) pada penelitian ini digunakan untuk penentuan rute terpendek yang dapat dilalui, Rumus yang digunakan pada algoritma A* (A Star) adalah sebagai berikut:

$$F(n) = G(n) + H(n) \tag{1}$$

Dimana,

$F(n)$ = biaya yang dibutuhkan untuk berpindah

$G(n)$ = biaya yang ditempuh dari *node* asal

$H(n)$ = nilai perkiraan dari *node* saat ini ke *node* tujuan.

Data input berupa nilai ditentukan dari *node* vertikal horizontal, maupun *node* diagonal.

1. Nilai G Cost

G Cost didapatkan dari jarak *node* ke *node* awal, nilai *node* vertikal horizontal dibedakan dengan *node* diagonal. *Node* vertikal dan horizontal diberi nilai 10 dan *node* diagonal diberi nilai 14.

Tujuan				
		14	10	14
		10	NPC	10
		14	10	14

Gbr. 1 Nilai G Cost untuk *node* vertikal horizontal diberi nilai 10 dan *node* diagonal diberi nilai 14

2. Nilai H Cost

H Cost didapatkan dari jarak *node* ke *node* tujuan, dilakukan penjumlahan baik vertikal horizontal maupun diagonal untuk mendapatkan nilai sesuai dengan jarak *node* ke *node* tujuan.

Tujuan							
		14	28	10	38	14	48
		10	38	NPC	10	52	
		14	48	10	52	14	56

Gbr. 2 Nilai H Cost, didapatkan dari penjumlahan *node* ke *node* tujuan

3. Menentukan F Cost

F Cost didapatkan dari penjumlahan *G Cost* dan *H Cost*

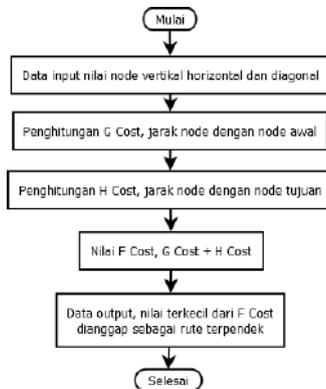
Tujuan				
		42	48	62
		48	NPC	62
		62	62	70

Gbr. 3 Nilai F Cost, didapatkan dari penjumlahan G Cost dan H Cost.

Pada penelitian ini Algoritma A*(A Star) akan diimplementasikan dengan alur sebagai berikut :

1. Mulai permainan.
2. Data *input* berupa nilai Horizontal Vertikal dan Diagonal.
3. Penghitungan *G Cost*, jarak *node* dengan *node* awal.

4. Penghitungan H Cost, jarak $node$ dengan $node$ tujuan.
5. Nilai F Cost, dengan menjumlahkan G Cost dan H Cost.
6. F Cost terkecil akan dianggap sebagai rute terpendek. Implementasi $A^*(A$ Star) Pada Penelitian ini akan digambarkan seperti pada gambar di bawah ini



Gbr. 4 Alur pengembangan algoritma A^* (A Star) pada game *Good Thief*

C. Tool Development

Tool yang digunakan dalam pengembangan *game* ini adalah *Unity*, *Unity* merupakan *game engine* yang sangat umum digunakan oleh pengembang – pengembang *game* pada umumnya dikarenakan UI yang sederhana dan mudah dimengerti, selain itu *unity* juga dikenal sebagai *multiplatform* sehingga *game* yang dikembangkan didalamnya saat selesai dan akan di-*publish*, pilihan format yang diberikan ke pengembang cukup banyak. Dalam pengembangan *game* ini, bahasa yang digunakan oleh peneliti adalah *C#*.

Selama pengembangan *game*, peneliti menggunakan *CorelDraw* dan *Photoshop* sebagai media desain yang utama dalam hal desain aset pada *game Good Thief*

III. HASIL DAN PEMBAHASAN

A. Spesifikasi Sistem

Dalam penelitian ini, komputer digunakan sebagai media untuk membangun *game Good Thief*. Komputer yang digunakan memiliki spesifikasi sebagai berikut.

1. Processor Intel(R) Core(TM) i58250U CPU @ 1.60GHz (8 CPUs)
2. Memori RAM 8,00 GB
3. Layar ukuran 1366 x 768
4. Keyboard beserta Mouse.

B. Tampilan Pada Permainan

Di bawah ini merupakan tampilan dari *game Good Thief*



Gbr. 5 Tampilan Halaman Awal pada game *Good Thief*

Pada Gbr. 5 terdapat tampilan Halaman Awal Pada *game*, dimana pada halaman awal terdapat tombol Mulai, Tentang dan Keluar. Tombol mulai akan mengarahkan pemain ke Halaman Tingkat Kesulitan, tombol Tentang akan mengarahkan pemain ke Halaman Tentang yang berisi sekilas tujuan *game* dibuat, sedangkan tombol Keluar digunakan untuk keluar dari permainan.



Gbr. 6 Tampilan Halaman Tentang pada game *Good Thief*

Pada Gbr. 6 terdapat tampilan Halaman Tentang, dimana halaman ini berisi informasi tentang tujuan pembuatan *game Good Thief*



Gbr. 7 Tampilan Halaman Tingkat Kesulitan pada game *Good Thief*

Pada Gbr. 7 terdapat tampilan Halaman Tingkat Kesulitan, pada halaman ini pemain bisa memilih tingkat kesulitan yang akan dipilih sesuai dengan keinginan pemain.



Gbr. 8 Tampilan Halaman Pengantar Kesulitan Mudah pada game Good Thief

Pada Gbr. 8 terdapat Tampilan Halaman Pengantar Kesulitan Mudah dimana pada halaman tersebut terdapat informasi dan juga misi yang harus dilakukan pemain untuk menyelesaikan game dalam tingkat kesulitan mudah.



Gbr. 9 Tampilan Halaman Tutorial 1 pada game Good Thief

Pada Gbr. 9 terdapat Tampilan Halaman Tutorial 1, dimana pada halaman ini terdapat informasi bagaimana pemain dapat menggerakkan pencuri yang terdapat di dalam game, dan juga terdapat informasi tentang bentuk penjaga yang berada didalamnya.



Gbr. 10 Tampilan Halaman Tutorial 2 pada game Good Thief

Pada Gbr. 10 terdapat Tampilan Halaman Tutorial 2, dimana pada halaman ini terdapat informasi mengenai item – item yang dapat diambil dan digunakan oleh pemain berupa Roti, Tali dan Kantung Uang.



Gbr. 11 Tampilan Halaman Tutorial 3 pada game Good Thief

Pada Gbr.11 terdapat Tampilan Halaman Tutorial 3, dimana pada halaman ini terdapat informasi bagaimana cara pemain menyelesaikan misi yang diberikan dan kapasitas kantung yang dimiliki oleh pemain.



Gbr. 12 Tampilan In Game 1 pada game Good Thief (LV 1)

Pada Gbr. 12 terdapat Tampilan In Game 1, dimana pada In Game 1 ini, pemain mulai bermain dan menyelesaikan misi yang diberikan dengan tingkat kesulitan mudah, di pojok kiri atas terdapat Nyawa, pojok kanan atas terdapat Mini map, pojok kiri bawah terdapat kapasitas kantung, pojok kanan bawah terdapat progres misi yang harus diselesaikan.



Gbr. 13 Tampilan In Game 2 pada game Good Thief (LV 2)

Pada Gbr. 13 terdapat Tampilan In Game 2, dimana pada In Game 2 ini, pemain mulai bermain dan menyelesaikan misi yang diberikan dengan tingkat kesulitan Sedang, pada tingkat ini, penjaga ada 2 jenis yaitu jenis normal dengan kecepatan 7 dan jenis penjaga dengan kecepatan 9, satu tingkat di bawah pemain dengan kecepatan 10.



Gbr. 14 Tampilan *In Game 3* pada game *Good Thief* (LV 3)

Pada Gbr. 14 terdapat Tampilan *In Game 3*, dimana pada *In Game 3* ini, pemain mulai bermain dan menyelesaikan misi yang diberikan dengan tingkat kesulitan Sulit, pada tingkat ini, penjaga ada 3 jenis yaitu jenis normal dengan kecepatan 7 dan jenis penjaga dengan kecepatan 9, satu tingkat di bawah pemain dengan kecepatan 10. Ada 1 penjaga lagi dengan kecepatan normal yaitu 7 akan tetapi bisa menghilang selama 3 detik, dan setelah 3 detik maka akan muncul kembali.



Gbr. 15 Tampilan Misi Berhasil Mudah pada game *Good Thief*

Pada Gbr. 15 terdapat Tampilan Misi Berhasil Mudah, tampilan ini akan muncul ketika pemain bisa menyelesaikan misi yang diberikan. Pada tampilan ini, pemain akan mendapatkan akses untuk kembali ke Halaman Awal, Mengulang, atau melanjutkan ke tingkat sedang.



Gbr. 16 Tampilan Misi Gagal Mudah pada game *Good Thief*

Pada Gbr. 16 terdapat Tampilan Misi Gagal Mudah, tampilan ini akan muncul ketika pemain gagal menyelesaikan misi yang diberikan. Hal ini bisa terjadi dikarenakan pemain sering tertangkap oleh penjaga hingga nyawa yang diberikan kepada pemain habis. Pada halaman ini pemain tidak diberi akses menuju ke tingkat sedang, dan diharuskan mengulang atau

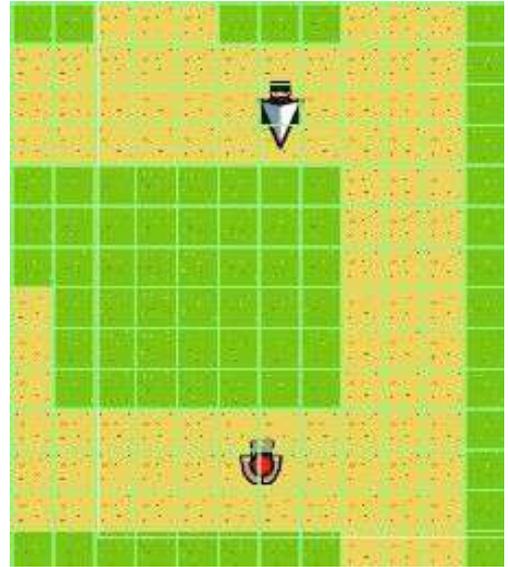
kembali ke halaman awal untuk memilih tingkat sedang dari luar.

Objek – objek apa saja yang terdapat di dalam *game* dan pemain dapat berinteraksi ditampilkan pada Tabel I.

TABEL I
 OBJEK YANG TERDAPAT DI DALAM *GAME GOOD THIEF*

No	Nama	Gambar
1	Pencuri (Pemain)	
2	Penjaga (NPC)	
3	Roti (Item)	
4	Tali (Item)	
5	Kantong Uang (Item)	
6	Panti Asuhan (Bangunan)	
7	Penjara (Bangunan)	

Pengujian Algoritma A^* (A^* Star) dilakukan dengan cara memberikan skenario ke NPC agar NPC mencari sendiri rute terpendek yang dapat dilalui.



Gbr. 17 Pengujian Algoritma A^* (A^* Star) di tingkat kesulitan mudah (LV 1)

Pada Gbr. 17 peneliti melakukan uji coba pada NPC di tingkat kesulitan mudah (LV 1), dimana NPC diharuskan menemukan rute terpendek yang dapat dilalui untuk menuju ke pemain.



Gbr. 18 Pengujian Algoritma A^* (A^* Star) di tingkat kesulitan mudah (LV 1) NPC menemukan rute terpendek yang bisa dilalui

Pada Gbr. 18 ditingkat kesulitan mudah (LV 1), NPC berhasil menemukan rute terpendek sehingga dapat dikatakan bahwa pengujian algoritma A^* pada tingkat kesulitan mudah (LV 1) berhasil. Pengujian sendiri menampilkan G Cost, H Cost dan

- 8

Rumah 1
(Bangunan)


- 9

Rumah 2
(Bangunan)


- 10

Rumah 3
(Bangunan)


- 11

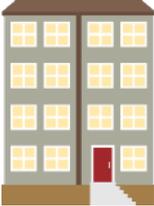
Rumah 4
(Bangunan)


- 12

Rumah 5
(Bangunan)

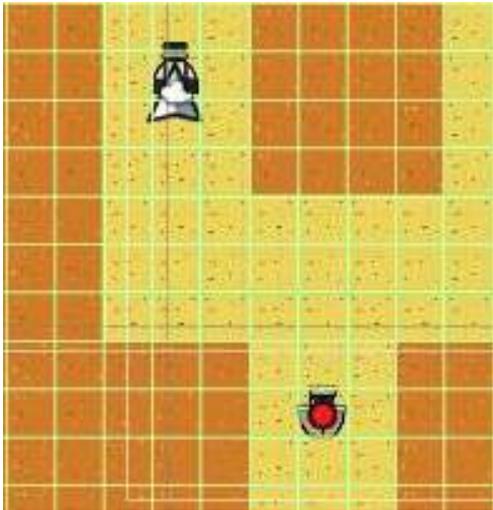

- 13

Rumah 6
(Bangunan)



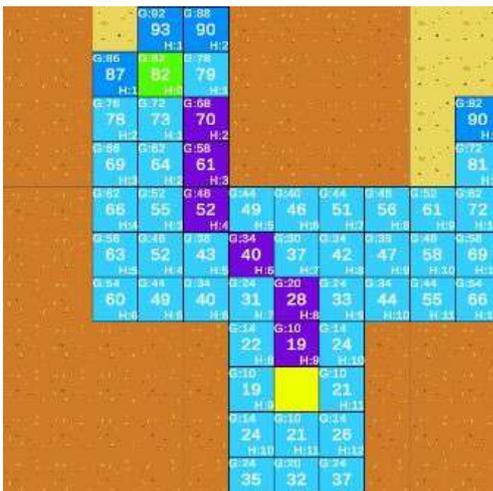
C. Pengujian Algoritma A^* (A^* Star)

$F Cost$, dimana NPC akan memilih $F Cost$ terendah sebagai rute terpendek untuk menuju pemain.



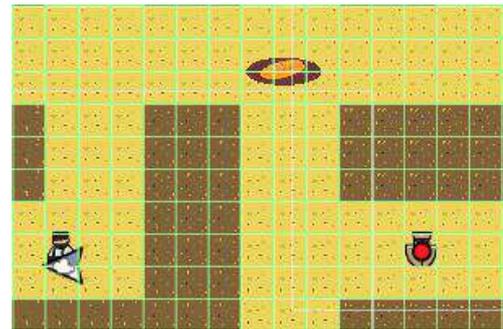
Gbr. 19 Pengujian Algoritma A^* ($A Star$) di tingkat kesulitan sedang (LV 2)

Pada Gbr. 19 peneliti melakukan uji coba pada NPC di tingkat kesulitan sedang (LV 2), dimana NPC diharuskan menemukan rute terpendek yang dapat dilalui untuk menuju ke pemain.



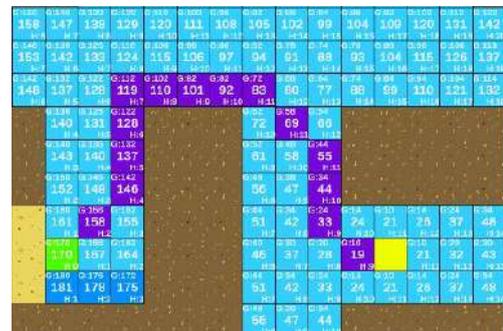
Gbr. 20 Pengujian Algoritma A^* ($A Star$) di tingkat kesulitan sedang (LV 2) NPC menemukan rute terpendek yang bisa dilalui

Pada Gbr. 20 ditingkat kesulitan sedang (LV 2), NPC berhasil menemukan rute terpendek menuju pemain dengan memilih $F Cost$ terendah, detail yang terdapat pada petak yaitu dipojok kiri atas merupakan $G cost$, pojok kanan bawah merupakan $H Cost$, dan di tengah – tengah petak terdapat $F Cost$.



Gbr. 21 Pengujian Algoritma A^* ($A Star$) di tingkat kesulitan sulit (LV 3)

Pada Gbr. 21 peneliti melakukan uji coba pada NPC di tingkat kesulitan sulit (LV 3), dimana NPC diharuskan menemukan rute terpendek yang dapat dilalui untuk menuju ke pemain.



Gbr. 22 Pengujian Algoritma A^* ($A Star$) di tingkat kesulitan sulit (LV 3) NPC menemukan rute terpendek yang bisa dilalui

Pada Gbr. 22 ditingkat kesulitan sulit (LV 3) NPC berhasil menemukan rute terpendek menuju pemain dengan memilih $F Cost$ terendah.

D. Alpha dan Beta testing

Alpha testing merupakan pengujian yang dilakukan oleh pengembang untuk mengetahui apakah *game* berjalan dengan baik atau tidak, apabila terjadi masalah akan segera diperbaiki hingga masalah tidak ditemukan. Pada tahap *alpha testing* ini, metode *blackbox* digunakan dalam pengujian *game Good Thief*.

TABEL II
 ALPHA TESTING PADA GAME GOOD THIEF

Skenario pengujian	Hasil yang diharapkan	Hasil Pengujian
Pencuri (Pemain) berjalan memasuki range Penjaga (NPC)	Penjaga (NPC) mulai mengejar Pencuri (Pemain) setelah Penjaga (Pemain) memasuki range	Berhasil
Pencuri (Pemain) mengambil Roti (item)	Kecepatan Pencuri (Pemain) akan meningkat 2x selama 5 detik	Berhasil

Pencuri (Pemain) mengambil Tali (item)	Penjaga (NPC) akan terdiam dan tidak mengejar Pencuri (pemain) selama 5 detik	Berhasil			kecepatan bergerak 7
Pencuri (Pemain) mengambil Kantung Uang (item)	Kapasitas kantung akan otomatis terisi 100 <i>gold</i>	Berhasil			Pencuri, Panti asuhan, Penjara serta Rumah dimunculkan di arena. Teks misi mengalami penyesuaian sesuai dengan tingkat kesulitan sedang. Rumah yang dimunculkan berjumlah 5 dengan masing – masing rumah memiliki 500 <i>gold</i> , jumlah Penjaga(NPC) yang dimunculkan yaitu 2, dengan kecepatan bergerak 7 dan kecepatan bergerak 9
Pencuri (Pemain) masuki rumah 1 – 6 dengan kapasitas kantung 0 (kosong)	Kapasitas <i>gold</i> di dalam rumah berkurang sesuai dengan maksimal kapasitas kantung Pencuri (Pemain), dan kapasitas kantung Pencuri (Pemain) terisi penuh yaitu sebanyak 200 <i>gold</i>	Berhasil	Pemain memulai tingkat kesulitan sedang (LV 2) dengan menekan tombol <i>play</i> pada halaman tutorial 3		Berhasil
Pencuri (Pemain) masuki rumah 1 – 6 dengan kapasitas kantung 100 <i>gold</i> (setengahnya)	Kapasitas <i>gold</i> di dalam rumah berkurang, sesuai dengan sisa kapasitas kantung yang dimiliki oleh Pencuri (Pemain)	Berhasil			Pencuri, Panti asuhan, Penjara serta Rumah dimunculkan di arena. Teks misi mengalami penyesuaian sesuai dengan tingkat kesulitan sulit. Rumah yang dimunculkan berjumlah 6 dengan masing – masing rumah memiliki 600 <i>gold</i> , jumlah Penjaga(NPC) yang dimunculkan 3 yaitu Penjaga 1 dengan kecepatan bergerak 7, penjaga 2 dengan kecepatan bergerak 9 dan Penjaga 3 dengan kecepatan 7 serta memiliki kemampuan menghilang dengan durasi menghilang 3 detik (berulang)
Pencuri (Pemain) dengan kapasitas kantung 200 <i>gold</i> (penuh) berjalan masuk ke dalam rumah	Pemain tidak bisa memasuki rumah	Berhasil			Pencuri (Pemain) akan dimunculkan kembali di sekitar penjara, nyawa yang dimiliki pemain akan berkurang, dan apabila sebelum tertangkap kapasitas kantung
Pencuri (Pemain) dengan kapasitas kantung 200 <i>gold</i> (penuh) & kapasitas kantung 100 <i>gold</i> (setengahnya) berjalan masuk ke dalam panti asuhan	Kapasitas kantung menjadi 0, <i>gold</i> di panti asuhan bertambah sejumlah <i>gold</i> yang terdapat dikantung dan teks misi ter- <i>update</i> sesuai perkembangan <i>gold</i> yang didapat panti asuhan	Berhasil	Pemain memulai tingkat kesulitan sulit (LV 3) dengan menekan tombol <i>play</i> pada halaman tutorial 3		Berhasil
Pemain memulai tingkat kesulitan mudah (LV 1) dengan menekan tombol <i>play</i> pada halaman tutorial 3	Pencuri, Panti asuhan, Penjara serta Rumah dimunculkan di arena, Teks misi mengalami penyesuaian sesuai dengan tingkat kesulitan mudah. Rumah yang dimunculkan berjumlah 3 dengan masing – masing rumah memiliki 500 <i>gold</i> , jumlah Penjaga(NPC) yang dimunculkan yaitu 1, dengan	Berhasil			

pemain dalam
kondisi terisi maka
akan dikosongkan

Beta testing merupakan tahap lanjutan setelah *alpha testing*, pada tahap *beta testing* dipilih 6 orang sebagai responden untuk memainkan *game Good Thief*, kemudian setelah itu 6 responden tersebut akan diberikan pernyataan dan diharuskan untuk memberi respons terhadap pernyataan yang diberikan untuk mengetahui tingkat kepuasan terhadap *game* yang dikembangkan, dan hasil yang dikeluarkan dapat membantu peneliti untuk memahami apakah *game* dapat berjalan dengan baik berdasarkan hasil kepuasan pengguna atau tidak.

Terdapat 30 pernyataan yang diberikan oleh peneliti pada Tabel III untuk mengetahui respons dari pengguna sehingga peneliti bisa mengetahui persentase *game* dapat berjalan dengan baik. Terdapat 5 kriteria yang dapat diisi oleh responden yaitu Tidak Setuju (TS), Kurang Setuju (KS), Cukup Setuju (CS), Setuju (S) dan Sangat Setuju (SS). Dengan skala 1 – 5 bergantung pada pernyataan yang diberikan

TABEL III
DAFTAR PERNYATAAN KUISIONER YANG DIAJUKAN

No	Pernyataan
1	Halaman Main menu mudah dimengerti
2	Halaman Tingkat Kesulitan dapat dibedakan dengan jelas
3	Tutorial 1 – 3 mudah dipahami
4	Mengumpulkan 1000 <i>gold</i> pada misi mudah, sudah sesuai
5	Mengumpulkan 2000 <i>gold</i> pada misi sedang, sudah sesuai
6	Mengumpulkan 3000 <i>gold</i> pada misi sulit, sudah sesuai
7	Penjaga dapat mengikuti pemain dengan baik
8	Kapasitas kantung maksimal 200 <i>gold</i> sudah sesuai
9	Item yang disebar memudahkan permainan
10	Item yang disebar dapat digunakan tanpa kendala
11	Durasi dari efek item roti (menambah kecepatan pemain) sudah sesuai
12	Item roti sangat membantu selama permainan
13	Item roti tidak diperlukan selama permainan
14	Penambahan 100 <i>gold</i> dari item kantung uang cukup membantu
15	Item kantung uang sangat membantu selama permainan
16	Item kantung uang tidak diperlukan selama permainan
17	Durasi dari efek yang ditimbulkan dari item tali (menghentikan penjaga) sudah sesuai
18	Item tali sangat membantu selama permainan
19	Item tali tidak dibutuhkan selama permainan
20	Tingkat kesulitan mudah berjalan dengan baik
21	Penjaga pada tingkat kesulitan mudah sudah benar berjumlah 1
22	Tingkat kesulitan sedang berjalan dengan baik
23	Penjaga pada tingkat kesulitan sedang sudah benar berjumlah 2
24	Tingkat kesulitan sulit berjalan dengan baik
25	Penjaga pada tingkat kesulitan sulit sudah benar berjumlah 3
26	Kecepatan bergerak pemain sudah sesuai
27	Penjaga 1 (muncul pada tingkat kesulitan mudah) kecepatan

28	sudah sesuai dan mudah diatasi pemain Penjaga 2 (muncul pada tingkat kesulitan sedang) kecepatan sudah sesuai dan mudah diatasi pemain
29	Penjaga 3 (muncul pada tingkat kesulitan sulit) kecepatan sudah sesuai dan efek berupa menghilang dengan interval tertentu mudah diatasi
30	Pesan yang ingin disampaikan peneliti dari <i>game good thief</i> berhasil tersampaikan dengan baik

Setelah mengisi kuisisioner, persentase dari setiap pernyataan akan didapatkan dengan menggunakan rumus berikut :

$$Y = \frac{\sum NR}{Skor Ideal} \times 100 \% \quad (2)$$

Keterangan :

- N = Skala(1-5)
- R = Jumlah responden yang memilih kriteria tersebut
- Σ = Jumlah
- Skor Ideal = Jumlah Responden dikali skala tertinggi

TABEL IV
HASIL YANG DIDAPKANKAN SETELAH RESPONDEN MENGENSI KUISIONER

No	Kriteria	N	R	N.R	Σ(N.R)	Skor Ideal	Y
1	TS	1	0	0	27	30	90%
	KS	2	0	0			
	CS	3	1	3			
	S	4	1	4			
	SS	5	4	20			
2	TS	1	0	0	30	30	100%
	KS	2	0	0			
	CS	3	0	0			
	S	4	0	0			
	SS	5	6	30			
3	TS	1	0	0	27	30	90%
	KS	2	0	0			
	CS	3	1	3			
	S	4	1	4			
	SS	5	4	20			
4	TS	1	0	0	30	30	100%
	KS	2	0	0			
	CS	3	0	0			
	S	4	0	0			
	SS	5	6	30			
5	TS	1	0	0	29	30	96,67%
	KS	2	0	0			
	CS	3	0	0			
	S	4	1	4			
	SS	5	5	25			
6	TS	1	0	0	21	30	70%
	KS	2	0	0			
	CS	3	3	9			
	S	4	3	12			
	SS	5	0	0			
7	TS	1	0	0	23	30	76,67%
	KS	2	0	0			
	CS	3	1	3			
	S	4	5	20			
	SS	5	0	0			

8	TS	1	0	0	25	30	83,33%
	KS	2	0	0			
	CS	3	2	6			
	S	4	1	4			
	SS	5	3	15			
9	TS	1	0	0	25	30	83,33%
	KS	2	0	0			
	CS	3	0	0			
	S	4	5	20			
	SS	5	1	5			
10	TS	1	0	0	24	30	80%
	KS	2	0	0			
	CS	3	0	0			
	S	4	6	24			
	SS	5	0	0			
11	TS	1	0	0	23	30	76,67%
	KS	2	0	0			
	CS	3	1	3			
	S	4	5	20			
	SS	5	0	0			
12	TS	1	0	0	24	30	80%
	KS	2	0	0			
	CS	3	0	0			
	S	4	6	24			
	SS	5	0	0			
13	TS	5	6	30	30	30	100%
	KS	4	0	0			
	CS	3	0	0			
	S	2	0	0			
	SS	1	0	0			
14	TS	1	0	0	25	30	83,33%
	KS	2	0	0			
	CS	3	0	0			
	S	4	5	20			
	SS	5	1	5			
15	TS	1	0	0	24	30	80%
	KS	2	0	0			
	CS	3	0	0			
	S	4	6	24			
	SS	5	0	0			
16	TS	5	6	30	30	30	100%
	KS	4	0	0			
	CS	3	0	0			
	S	2	0	0			
	SS	1	0	0			
17	TS	1	0	0	23	30	76,67%
	KS	2	0	0			
	CS	3	2	6			
	S	4	3	12			
	SS	5	1	5			
18	TS	1	0	0	25	30	83,33%
	KS	2	0	0			
	CS	3	0	0			
	S	4	5	20			
	SS	5	1	5			
19	TS	5	6	30	30	30	100%
	KS	4	0	0			
	CS	3	0	0			
	S	2	0	0			
	SS	1	0	0			

20	SS	1	0	0	26	30	86,67%
	TS	1	0	0			
	KS	2	0	0			
	CS	3	0	0			
	S	4	4	16			
21	SS	5	2	10	24	30	80%
	TS	1	0	0			
	KS	2	0	0			
	CS	3	0	0			
	S	4	6	24			
22	SS	5	0	0	24	30	80%
	TS	1	0	0			
	KS	2	0	0			
	CS	3	0	0			
	S	4	6	24			
23	SS	5	0	0	23	30	76,67%
	TS	1	0	0			
	KS	2	0	0			
	CS	3	1	3			
	S	4	5	20			
24	SS	5	0	0	22	30	73,33%
	TS	1	0	0			
	KS	2	1	2			
	CS	3	0	0			
	S	4	5	20			
25	SS	5	0	0	22	30	73,33%
	TS	1	0	0			
	KS	2	1	2			
	CS	3	1	3			
	S	4	4	16			
26	SS	5	0	0	22	30	73,33%
	TS	1	0	0			
	KS	2	0	0			
	CS	3	3	9			
	S	4	2	8			
27	SS	5	1	5	23	30	76,67%
	TS	1	0	0			
	KS	2	0	0			
	CS	3	2	6			
	S	4	3	12			
28	SS	5	1	5	21	30	70%
	TS	1	0	0			
	KS	2	0	0			
	CS	3	3	9			
	S	4	3	12			
29	SS	5	0	0	20	30	66,67%
	TS	1	0	0			
	KS	2	1	2			
	CS	3	2	6			
	S	4	3	12			
30	SS	5	0	0	27	30	90%
	TS	1	0	0			
	KS	2	0	0			
	CS	3	0	0			
	S	4	3	12			

Berdasarkan hasil kuisioner yang diberikan kepada 6 responden dengan total pernyataan sebanyak 30, didapatkan

hasil persentase setiap pernyataan yang ada. Kemudian dilanjutkan dengan mencari rata – rata keseluruhan pernyataan. rata – rata didapatkan dengan menjumlahkan semua hasil persentase setiap pernyataan kemudian dibagi dengan banyaknya pernyataan, didapatkan hasil yang cukup memuaskan yaitu sebesar 83,22% dari 6 responden

IV. KESIMPULAN

Pada penelitian diatas yaitu tentang Implementasi Algoritma A^* (*A Star*) dalam penentuan rute terpendek yang dapat dilalui oleh NPC pada *game Good Thief* berjalan dengan baik, *alpha testing* untuk mengetahui apakah *game* berjalan dengan baik juga berhasil dilakukan dan dapat dilihat pada Gbr. 17 sampai dengan Gbr. 22 mulai dari tingkat kesulitan mudah (LV 1), tingkat kesulitan sedang (LV 2) hingga tingkat kesulitan sulit (LV 3), NPC berhasil menemukan rute terpendek dengan memilih rute berdasarkan *F Cost* terendah. Berdasarkan *beta testing* juga dapat diketahui dari 30 pernyataan yang diberikan kepada 6 responden yang ikut serta dalam pengujian *game Good Thief*, tingkat kepuasan atas *game* yang dimainkan cukup memuaskan yaitu sebesar 83,22%

V. SARAN

Pada penelitian ini terdapat banyak sekali kekurangan, salah satunya adalah dalam masalah target yang dapat diraih oleh NPC, target di sini hanya satu yaitu pemain itu sendiri, untuk ke depannya diharapkan target yang dapat diraih oleh NPC berjumlah lebih dari satu dimana NPC bisa menentukan prioritas target mana yang harus dikejar, sehingga *gameplay* bisa lebih berkembang dan menarik.

UCAPAN TERIMA KASIH

Puji syukur kepada Allah SWT yang telah memberikan penulis kemudahan dalam pengerjaan dan penyelesaian penelitian ini. Serta ucapan terima kasih kepada semua pihak yang membantu penulis baik secara langsung maupun tidak langsung sehingga jurnal ini dapat diselesaikan.

REFERENSI

- [1] Bonifacio, B., Fitri, M., & Indra, D. W. (2018). PEMBUATAN GAME KUIS SIAPA PINTAR . *JIMP - Jurnal Informatika Merdeka Pasuruan* , 44.
- [2] Maulida, L. (2018, October 17). *tek.id*. Retrieved from insight: <https://www.tek.id/insight/jumlah-gamer-di-indonesia-capai-100-juta-di-2020-b1U7v9c4A>
- [3] Dedy, H., & Sepri, D. (2018). Penerapan Algoritma A-Star Sebagai Pencari Rute Terpendek pada Robot Hexapod. *Jurnal Nasional Teknik Elektro*, 123 - 129.
- [4] Kiki, S., Supriyadin, Imam, S., & Roy, B. (2018). MENGHITUNG RUTE TERPENDEK MENGGUNAKAN ALGORITMA A^* DENGAN FUNGSI EUCLIDEAN DISTANCE . *Sentika*, 70 - 79.
- [5] Imam, A., & Wahyu, W. (2017). Penerapan Algoritma A Star (A^*) pada Game Petualangan Labirin Berbasis Android . *Jurnal Ilmu Komputer dan Informatika*, 57 - 63.
- [6] Aditya, S., Paulus, H., & Sri, S. (2019). Penerapan Pathfinding Menggunakan Algoritma A^* Pada Non Player Character (NPC) Di Game. *Jurnal Ilmiah Sinus (JIS)*, 39-50.
- [7] Nawaf, H. B., Sinan, S. M.-D., & Mustafa, A. S. (2016). Pathfinding in Strategy Games and Maze Solving Using A^* Search Algorithm . *Journal of Computer and Communications*, 15-25.
- [8] Sharma, S. K., & B.L.Pal. (2015). Shortest Path Searching for Road Network using A^* Algorithm. *International Journal of Computer Science and Mobile Computing*, 513-522.
- [9] Doni, F., & Fata Nidaul, K. (2019). Pengujian Black Box Dan Kuesioner Pada Game Feed The Animal . *MAHASISWA BINA INSANI*, 193-202.
- [10] Fata, N. K., & Sugeng, M. (2019). PENGUJIAN BETA PADA APLIKASI GAME EDUKASI PENGENALAN DASAR ISLAM MELALUI KUESIONER . *INFOKAM*, 83-89.