Random Obstacle Advoidance oleh *Non Player Character* Menggunakan Algoritma A*

Latius Hermawan[1], Maria Bellaniar Ismiati[2]
[1,2] Program Studi Informatika
Fakultas Sains dan Teknologi UNIKA
Musi Charitas Palembang

***Abstract*—Belum ditemukannya solusi yang optimal untuk menyelesaikan masalah pathfinding. Biasanya solusinya tergantung pada spesifik dari ketentuan untuk dapat menemukan jalan keluar untuk setiap permainan yang diberikan. Pathfinding yang optimal merupakan suatu hal yang sulit untuk ditemukan, dimana terdapat perbedaan antara istilah path dan shortest path. Sehingga pathfinding yang ada bertujuan untuk menemukan jalur terpendek yang optimal. Salah satu algoritma yang digunakan pada pathfinding adalah A* digunakan dalam melakukan pencarian jalur yang optimal yang menghubungkan dua titik pada peta (grafik) dari permainan yang ada. Algoritma A* dapat membantu NPC untuk menemukan rute dalam mencari keberadaan target dengan berbagai halangan yang disediakan secara acak dalam satu waktu. Sehingga algoritma ini dapat digunakan untuk mencari rute walaupun ada banyak obstacle yang bermunculan secara acak.**

Kata kunci: Algoritma A*, Pathfinding, Obstacle Advoidance, Kecerdasan Buatan, AI.

I. PENDAHULUAN

Belum ditemukannya solusi yang optimal untuk menyelesaikan masalah *pathfinding*. Biasanya solusinya tergantung pada spesifik dari ketentuan untuk dapat menemukan jalan keluar untuk setiap permainan yang diberikan [1]. *Pathfinding* yang optimal merupakan suatu hal yang sulit untuk ditemukan, dimana terdapat perbedaan antara istilah *path* dan *shortest path*. Sehingga *pathfinding* yang ada bertujuan untuk menemukan jalur terpendek yang optimal [3]. *Pathfinding* memiliki fungsi menemukan jalan terpendek antara dua titik yang ada. Proses dimulai dari mulai *node* awal hingga mampu menemukan jalan keluar yang terpendek agar masalah pencarian rute yang ada dapat diselesaikan dengan baik [2]. Seperti halnya *Non Player Character* yang sering ditemui pada permainan yang ada. Salah satu kemampuan *Non Player Character* yaitu membutuhkan pengetahuan mengenai pencarian rute pada suatu permainan.

Non Player Character memiliki peranan penting dalam permainan, seperti menyajikan *story-line*, menjadi musuh, teman ataupun sebagai objek tambahan pada suatu permainan agar terlihat lebih nyata [4]. Banyak permainan memerlukan desain *Artificial Intelligence* yang baik untuk mengontrol karakter. Sebagai contoh, *Non Player Character* harus bisa berperilaku alami dan dapat bergerak untuk mengepung dan menyerang musuh. merupakan kunci dari permainan yang telah didiskusikan pada literature [5][6][7].

Salah satu algoritma yang digunakan pada pathfinding adalah A* [8][9], digunakan dalam melakukan pencarian jalur yang optimal yang menghubungkan dua titik dari permainan yang ada. A* lebih cocok untuk sebuah lingkungan di mana ada beberapa alternatif rute yang ada. Sebuah tempat dapat dibuat dengan cara membuat beberapa jalan keluar untuk mencapai suatu tempat serta diberikan beberapa halangan yang ada. Pada kasus multi *Non Player Character*, setiap *Non Player Character* harus mencari jalurnya masing-masing untuk mencapai sebuah target yang telah ditentukan sebelumnya dengan berbagai *obstacle* yang telah ada dan dapat diubah sesuai kebutuhan. Sehingga permainan yang sudah dibuat menjadi lebih menarik, disinilah kesulitan yang harus diselesaikan dengan menggunakan metode yang berbasis *pathfinding*.

II. LANDASAN TEORI

Algoritma A* (A star) adalah algoritma pathfinding pengembangan dari Algoritma BFS (*Best First Search*). Seperti halnya pada BFS, untuk menemukan solusi, A* juga 'dituntun' oleh fungsi heuristik[1,2,3]. Notasi yang dipakai oleh Algoritma A* adalah sebagai berikut:

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

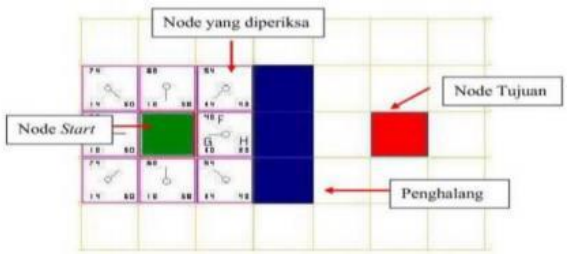
Pada notasi standar yang dipakai untuk algoritma A* tersebut, digunakan g(n) untuk mewakili

cost rute dari *node* awal ke *node* n, lalu h(n) mewakili perkiraan *cost* dari *node* awal ke *node*goal, yang dihitung dengan fungsi heuristik [12].

Autonomous character mewakili tokoh dalam cerita atau permainan dan memiliki kemampuan untuk improvisasi tindakan mereka. Ini adalah kebalikan dari seorang tokoh dalam sebuah film animasi, yang tindakannya ditulis di muka, dan dalam sebuah permainan, tindakan yang diarahkan oleh pemain. Karakter otonom dalam permainan biasanya disebut NON PLAYER CHARACTER (Non-Player Character) [15].

III. METODE

A-Star Terdapat beberapa hal yang perlu didefinisikan terlebih dahulu dalam kasus game pathfinding dengan penerapan algoritma A* (A Star). Beberapa terminologi dasar yang terdapat pada algoritma ini adalah *starting point*, simpul (*node*s), A, *Open List*, *Closed List*, harga (*cost*), halangan. Starting point adalah posisi awal sebuah benda. A* adalah simpul yang sedang dijalankan algortima pencarian jalan terpendek. Sedangkan simpul adalah petak-petak kecil sebagai representasi dari area pathfinding. *Open List* adalah tempat menyimpan data simpul yang mungkin diakses dari starting point maupun simpul yang sedang dijalankan. *Closed List* adalah tempat menyimpan data simpul sebelum A* yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan. Harga (F) adalah nilai yang diperoleh dari penjumlahan nilai G. Simpul tujuan yaitu simpul yang dituju. Rintangan adalah sebuah atribut yang menyatakan bahwa sebuah simpul tidak dapat dilalui oleh A*.

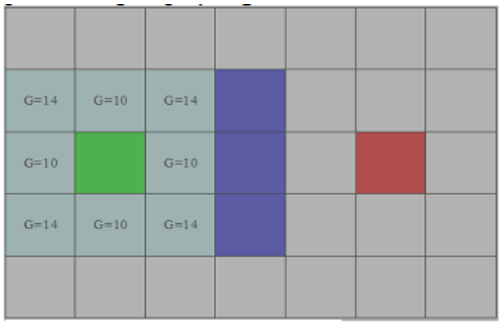


Gambar 1. Agoritma A Star

[13] Fungsi f sebagai estimasi fungsi evaluasi terhadap *node* n, dapat dituliskan :

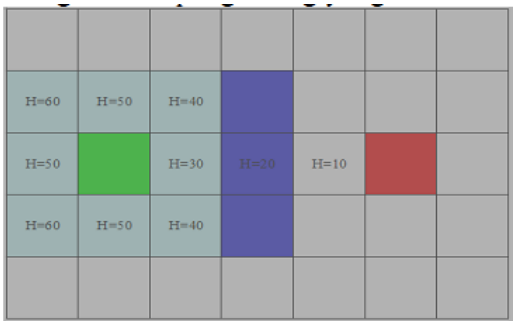
f(n) = g(n) + h(n) (2)

f(n) adalah fungsi evaluasi (jumlah g(n) dengan h(n)), g(n) adalah biaya (*cost*) yang dikeluarkan dari keadaan awal sampai keadaan n, h(n) adalah estimasi biaya untuk sampai pada suatu tujuan mulai dari n. [14] Pergerakan diagonal pada map diperbolehkan, maka digunakan fungsi heuristic Non-Manhattan Distance. Tahapan yang dilakukan yaitu, Asumsikan setiap langkah dari hijau adalah legal baik vertikal, horizontal, maupun diagonal dengan catatan tidak membentur tembok. G adalah biaya dalam setiap langkah. Dalam kasus ini kita akan berikan nilai 10 untuk setiap langkah vertikal maupun horizontal, dan 14 untuk diagonal. Nilai 14 kita dapatkan dari perhitungan phytagoras.



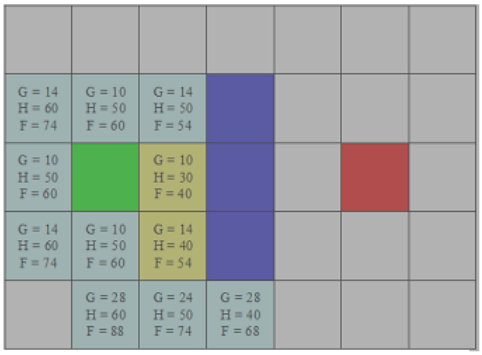
Gambar 2. *Cost* Pada A-Star

Selanjutnya hitung biaya estimasi pergerakan yang disimbolkan dengan H. Nilai H adalah jarak / estimasi biaya dari pergerakan dari suatu titik ke tujuan dengan mengabaikan penghalang yang ada.

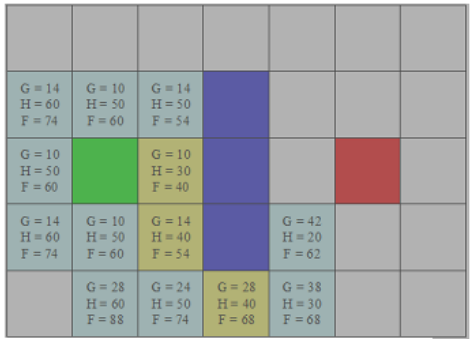


Gambar 3. Perhitungan Jawak Awal A-Star

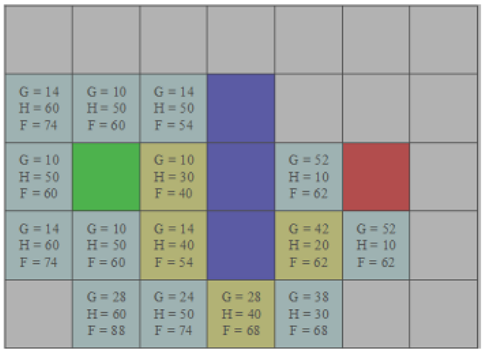
Setelah nilai G dan H didapatkan, maka beri skor dari masing-masing titik yang akan dilalui. Skor kita lambangkan misalnya dengan F dimana nilai F = G + H. Setelah pergerakan pertama selesai selanjutnya lakukan perulangan dari dari langkah 1 sampai 4.



Gambar 4. Perhitungan bobot akhir A-Star



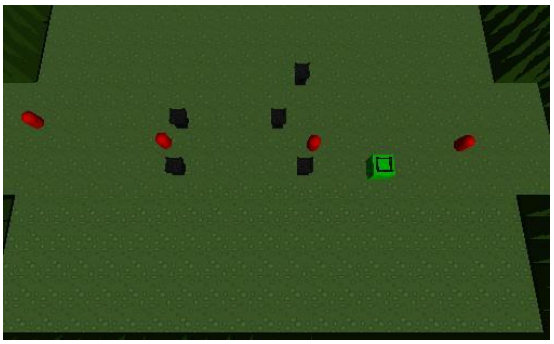
Gambar 5. Rute yang akan dilalui



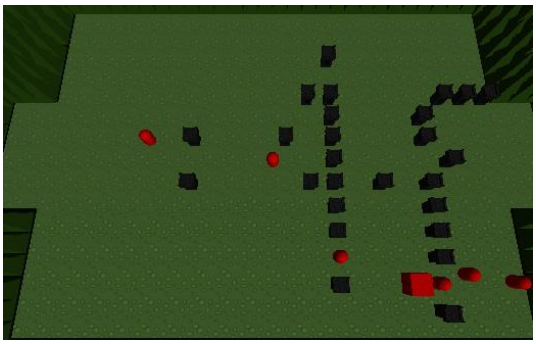
Gambar 6. Rute yang telah didapat

IV. HASIL DAN PEMBAHASAN

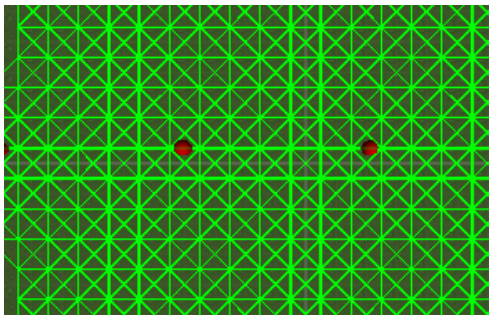
Dari perhitungan metode yang telah dilakukan dengan A-Star, terlihat bahwa NPC dapat menemukan rute dan jalur yang akan dilewati dengan *obstacle* (penghalang) yang timbuk secara random dengan cara klik pada sembarang posisi pada percobaan yang sudah dilakukan. Langkah pertama yang dilakukan NPC dalam menemukan target (sisi kanan) adalah dengan melihat layout tempat dimana *obstacle* dan target berada. Selanjutnya NPC melihat rute mana yang dapat dilalui dengan menghitung bobot terendah dari masing-masing rute. Setelah mendapatkan bobot terendah, NPC dapat melewati rute tersebut dan menemukan keberadaan target yang dimaksud. Jika *obstacle* yang ada ditambah lagi, NPC harus menghitung ulang bobot yang ada agar dapat menemukan rute terbaru untuk menuju ke tujuan. Pada penelitian ini, *obstacle* dapat dibuat ke berbagai tempat sesuai keinginan peneliti. Setiap *obstacle* acak ditambah atau berpindah tempat, NPC dapat menentukan beberapa rute yang harus dilewati. Posisi target dan NPC dipaparkan pada Gambar 7, sedangkan untuk rute yang dapat dilewati oleh NPC untuk sampai ke target dipaparkan pada Gambar 8.



Gambar 7. Posisi Random *Obstacle* (Kotak kecil hijau) dan NPC (Merah)



Gambar 8. NPC Berhasil melewati *obstacle* acak yang dibuat



Gambar 9. Bentuk dari alternatif rute yang ada

V. KESIMPULAN

Dari percobaan yang dilakukan, algoritma A* dapat membantu NPC untuk menemukan rute dalam mencari keberadaan target dengan berbagai halangan yang disediakan secara acak dalam satu waktu. Sehingga algoritma ini dapat digunakan untuk mencari rute walaupun ada banyak *obstacle* yang bermunculan secara acak.

DAFTAR PUSTAKA

[1] Jung Y. An Effective Method of Pathfinding in a Car Racing Game. IEEE Computer and Automation Engineering (ICCAE) The 2nd International Conference : 544-547 .2010.

[2] Geethu E, Mathew. Direction Based Heuristic For Pathfinding In Video Games, IEEE Sponsored Second International Conference On Electronics And Communication Systems.2015.

[3] Björnsson, Y. TBA*: Time-Bounded A*. IEEE Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-09); 431-436.2009.

[4] Jafar A. Portable Non-Player Character Tutors with Quest Activities, IEEE Virtual Reality : 253-354, 2010.

[5] S. Priesterjahn, O. Kramer, A. Evolution of human competitive agents in modern computer games. IEEE Congress on Evolutionary Computation (CEC2006): 777–784. 2006.

[6] J. Hagelbäck and S. J. Johansson, The rise of potential fields in realtime strategy bots. Fourth Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE) : 42–47. 2008.

[7] P. Avery, S. Louis. Evolving Coordinated Spatial Tactics For Autonomous Entities Using Influence Maps. IEEE Symposium on Computational Intelligence and Games (CIG2009) : 341–348. 2009.

[8] R. Dechter and J. Pearl, Generalized Best-First Search Strategies And The Optimality Of A*. Journal of the ACM, vol. 32 : 505-536. 1985.

[9] D. M. Bourg, and G. Seemann, AI for Game Developers 1st ed., O'Reilly Media : 51-75. 2004.

[10] Hartley, T. In-Game Adaptation Of A Navigation Mesh Cell Path. IEEE 17th International Conference on Computer Games: AI, Animation, Mobile,

[11] Jie Hu. A Pathfinding Algorithm in Real-time Strategy Game based on Unity3D . IEEE ICALIP : 1159-1162. 2012.

[12] Riftadi, M. Variasi Penggunaan Fungsi Heuristik dalam Pengaplikasian Algoritma A*, Makalah IF2251, Teknik Informatika ITB, Bandung. 2007.

[13] Nelly I. Membangun Game Edukasi Sejarah Walisongo. Jurnal Ilmiah Komputer dan Informatika (KOMPUTA). 2012.

[14] Zou, Huilai., Qu, Zening., Qu, Youtian. Optimized Application and Practice of A* Algorithm in Game Map Path- Finding.

[15] Yunifa, Mochamad Hariadi dan Supeno Mardi. Strategi Menyerang pada Game FPS Menggunakan Hierarchy Finite Machine dan Logika Fuzzy, Institut Teknologi Sepuluh November, 2012.