

PENENTUAN PERGERAKAN *NON-PLAYER CHARACTER* MENGGUNAKAN ALGORITMA A* PADA GAME ACTION- *ROLE-PLAYING GAME*

Atthariq¹, Dimas Ariandy Putra²

^{1,3} Jurusan Tekniknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

dimas4rie@gmail.com

Abstrak— *Game* adalah salah satu bentuk dari animasi interaktif dimana *player* dapat berinteraksi dengan dunia *game*. Dalam sebuah *game*, salah satu unsur yang dapat dianggap penting untuk mendukung jalannya *game* dan realitas dari dunia *game* adalah *NPC* (Non-Player Character). *NPC* dapat membuat sebuah *game* menjadi lebih nyata dari segi cara Perpindahannya, maka dibutuhkan suatu algoritma *pathfinding* yang mampu membuat *NPC* tersebut melakukan perpindahan layaknya suatu makhluk hidup berpindah di dunia nyata. A*(A-Star) adalah algoritma pencarian yang dapat digunakan untuk melakukan *pathfinding*, dalam hal ini A Star akan digunakan untuk mencari suatu jarak terpendek antara *NPC* dan karakter *player*. Penelitian ini dilakukan untuk melakukan percobaan terhadap implementasi algoritma A* pada lingkungan *game*.

Kata kunci—*NPC (Non Player Character), pathfinding, Algoritma A Star, Game*

Abstract— *Game* is one form of interactive animation where the player can interact with the gaming world. In a game, one element that can be considered important to support the game and the reality of the game world is *NPC* (Non-Player Character). *NPC* can make a game becomes more real in terms of way of movement, then it takes a *pathfinding algorithm* that is able to make the *NPC* move like a living creature. A * (A-Star) is a search algorithm that can be used to perform *pathfinding*, in which case A Star will be used to find a shortest Path between *NPC* and player character. This research was conducted to experiment on the implementation of A* algorithm in game environment.

Keywords— *Game, A* Algorithm, NPC, pathfinding Algorithm.*

I. PENDAHULUAN

Game merupakan permainan yang terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun dengan untuk memaksimalkan kemampuan sendiri ataupun meminimalkan kemenangan lawan^[1]

Dari sekian banyaknya genre *game* yang berkembang, salah satu jenis genre *game* adalah *RPG (Role-Playing Game)*. Jenis Genre ini telah tumbuh dan berkembang menjadi jenis genre yang di perhitungkan. Tidak seperti *game* pada jaman sekarang yang mengandalkan Grafis, *RPG* lebih condong terhadap teka-teki, alur dan plot cerita.

Autonomous character adalah jenis otonomous agent yang ditunjukkan untuk penggunaan komputer animasi dan media interaktif seperti *game* dan virtual reality. Agen ini mewakili tokoh dalam cerita atau pemain dan memiliki kemampuan untuk improvisasi tindakan mereka. Ini adalah kebalikan dari seorang tokoh dalam sebuah film animasi, yang tindakannya ditulis dimuka, untuk “avatar” dalam sebuah permainan atau virtual reality, tindakan yang diarahkan secara real time oleh pemain. Dalam permainan, karakter otonom biasanya disebut *NPC (Non-Player Character)*.^[2]

Role Playing Games adalah sebuah permainan dimana seseorang masuk ke dalam sebuah peran dan dunia kemudian diberikan kesempatan untuk berpartisipasi dan berinteraksi dengan isi dari dunia tersebut.^[3] Non Player Character memainkan peranan penting dalam banyak permainan, seperti

menyajikan story-line, menjadi musuh ataupun teman dan memberikan informasi kepada pengguna yang berhubungan dengan permainan^[4]

Dibutuhkan adanya AI (artificial intelligence) agar *game* lebih menarik. AI didalam *Game* dapat diterapkan pada *NPC* (non-playable character). *NPC* adalah karakter yang mampu berinteraksi dengan objek didalam dunia *Game* dan tindakannya tidak dapat dikendalikan oleh pemain. Tindakan *NPC* yang tidak dapat diprediksi salah satunya adalah pergerakannya dari suatu tempat ketempat lain. Dengan diterapkannya AI pada *NPC* sehingga membuat *NPC* mampu mengotomasikan rute pergerakannya didalam *game*.^{[5][6]}

Artificial Intelligence (AI) untuk karakter lawan yang dibutuhkan untuk menemukan ada jalan (*pathfinding*) yang dirasa penting untuk banyak permainan komputer, khususnya secara *Role Playing Game*. Algoritma *pathfinding* yang diimplementasikan pada sebuah *game* adalah algoritma A*.^[7]

Pada dasarnya terdapat dua teknik pencarian dan pelacakan yang digunakan, yaitu pencarian buta (*blind search*) dan pencarian terbimbing (*heuristic search*). Dalam pencarian terbimbing ada beberapa algoritma yang dapat digunakan, salah satunya adalah A*. Algoritma ini adalah salah satu algoritma yang begitu populer didunia dan juga memberika solusi yang cukup bagus bagi proses *pathfinding*, sehingga begitu sering digunakan dalam pembuatan *game*. Algoritma A* merupakan pengembangan dari *best-first search (BFS)* yang digunakan untuk mencari jalan terpendek (*shortest path*)

yang sering dipakai dalam game programming. Pada penelitian ini diimplementasikan algoritma A* untuk penerapan perilaku NPC.

Berdasarkan latar belakang tersebut yang dijadikan masalah dalam penelitian ini adalah Bagaimana membangun proses perancangan *game RPG The Fallen Kingdom* dengan penerapan *Health Point NPC* dan Bagaimana sebuah *game* dapat menerapkan nilai kesehatan dalam bertahan dan menyerang

II. METODOLOGI PENELITIAN

Dalam pembuatan sistem, tahapan pertama yang dilakukan adalah analisis sistem. Kemudian perancangan sistem untuk menganalisis kebutuhan yang diperlukan untuk pembuatan system.

A. Block Diagram

Pada penelitian ini, akan dibahas tentang perilaku NPC (Non Player Character) berupa penentuan posisi perilaku yang dilakukan oleh NPC.

a. Proses Perilaku Pergerakan

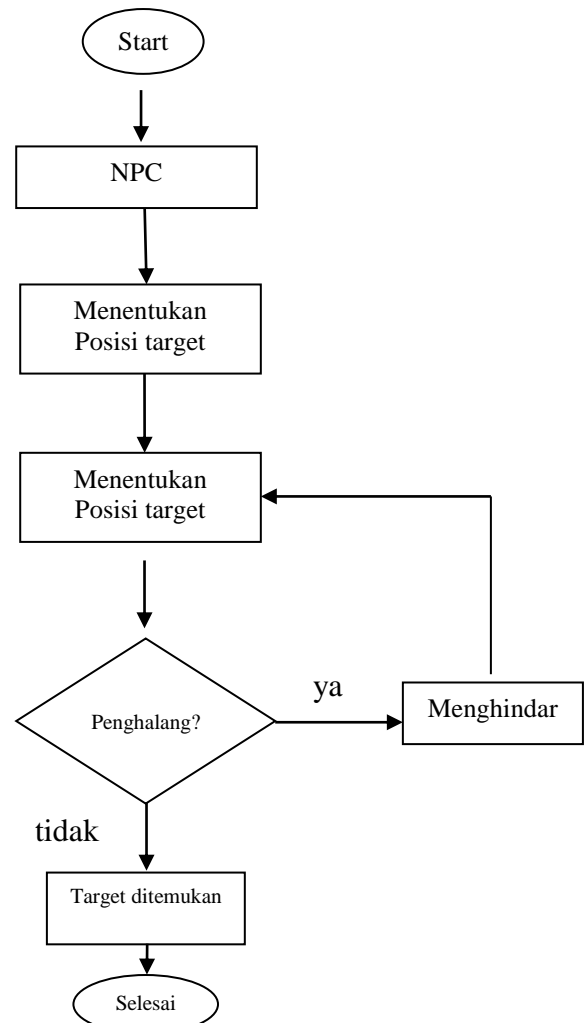
Bagian ini akan menunjukkan proses dari penentu perilaku menyerang NPC dapat dilihat pada gambar berikut ini.



Gambar 1 Alur Proses Menyerang

Dimana proses input perancangan strategy penyerang, character akan mencari monster enemy untuk melakukan battle. Pada proses logika fuzzynya akan dilakukan pencarian damage yang ditimbulkan pada saat battle. Pada keluaran aksi penyerangan NPC, kita bisa mengetahui nilai-nilai yang akan diterima pada character.

b. Diagram Alir perilaku pergerakan NPC



Gambar 2. Diagram A Star

Gambar 2 menjelaskan ketika NPC menentukan target, NPC akan menghitung jarak dari NPC ke target. Ketika jumlah sudah didapat maka NPC akan mengejar target,. Jika NPC menemukan penghalang maka NPC akan menghindari dan kembali mengejar target lagi, dan ketika NPC tidak menemukan penghalang, maka target ditemukan.

III. HASIL DAN PEMBAHASAN

A. Perancangan Unsur Permainan

Permainan komputer tidak lepas dari beberapa unsur atau jenis game. Setiap game atau permainan memiliki rating tersendiri. Unsur-unsur tersebut bila disusun dengan cermat secara terencana tentunya akan menghasilkan permainan yang menarik bagi pemainnya.

a) Tampilan Layar Utama

Pada gambar berikut merupakan hasil akhir gambar tampilan layar utama saat permainan dijalankan.



Gambar 3. Tampilan Layar Utama

Pada tampilan layar utama terlihat beberapa bagian awal game yaitu, New Game, Continue, Shutdown.

b) Tampilan Layar Menu

Pada gambar berikut merupakan hasil gambar tampilan menu dengan menekan tombol Esc pada keyboard maka menu tampilan akan tertampil dipermainan. Dalam menu tampilan terdapat bagian-bagian menu yaitu, Item, Skill, Equipment, Status, Quest, Formation, Save dan Game End.



Gambar 5. Tampilan Layar Menu

c) Tampilan Layar Battel Permainan

Pada gambar berikut merupakan hasil gambar saat sistem pertarungan dalam permainan.



Gambar 6. Tampilan Layar Battel Permainan

d) Tampilan Gambar Area Dungeon

Pada Area Dungeon dalam permainan yang saya gunakan

• Area Maze Entrance Dungeon.

Pada gambar berikut merupakan Area Dungeon yang harus kita taklukan. Pada daerah ini monster yang sangat mendominasi



Gambar 7. Tampilan Area Maze

• Area Goblin Cave Dungeon

Pada gambar berikut merupakan Area Dungeon yang harus kita taklukan. Disini semua pasukan siluman kerajaan berkumpul di daerah ini.



Gambar 8. Tampilan Area Goblin

• Script ATB Stamina System

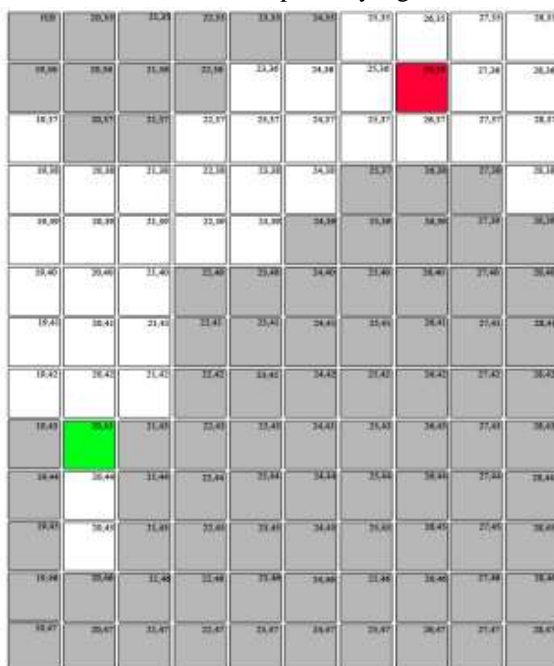
ATB Stamina System adalah sistem battle stamina yang akan menjadi patokan character untuk melakukan Action kepada monster. Jika bar ATB Stamina belum penuh maka character tidak bisa melakukan Action. Tampilan ATB system tertera dibawah ini



Gambar 9. Tampilan Script ATB Stamina

B. Pengujian Sistem

Sebelum diimplementasikan pada game yang akan dibuat, algoritma A-Star akan diuji terlebih dahulu. Pada dasarnya Algoritma A-Star adalah algoritma Pathfinding dengan cara melakukan pencarian nilai heuristik terkecil dari setiap node yang ada disekitar



Gambar 10. Simulasi A star

Keterangan :

Kotak merah = node awal

Kotak hijau = node tujuan

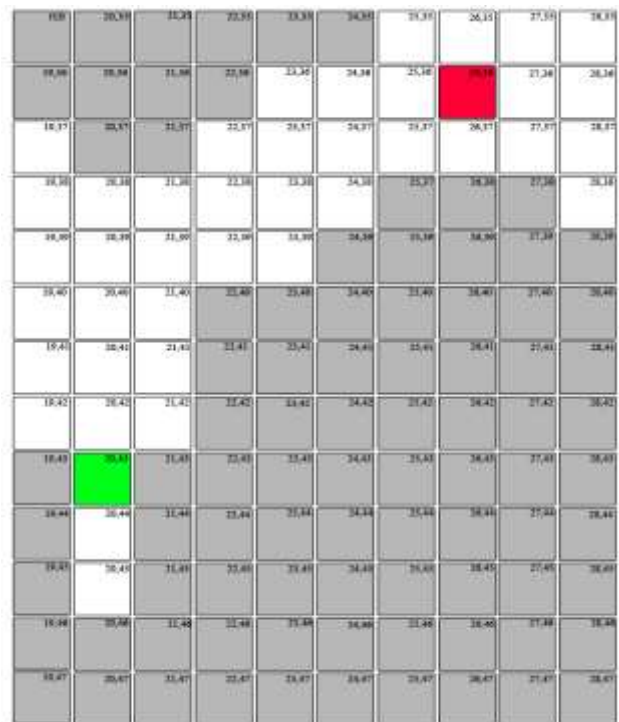
Kotak abu-abu = penghalang

Langkah penyelesaian A*

Pertama-tama algoritma A* menginisiasi lokasi musuh sebagai node awal dan lokasi pemain sebagai node tujuan.

Berikut penjelasan langkah-langkah pencarian rute pada algoritma A*:

- Memasukkan node A (node awal) ke dalam open list
- Memeriksa node-node yang berdekatan dengan node A. Abaikan node dengan penghalang lalu tambahkan semua node tersebut dalam open list. Node A sendiri nantinya akan menjadi "parent" dan digunakan untuk menelusuri jalan.
- Menghapus node A dari open list dan dimasukkan kedalam closed list



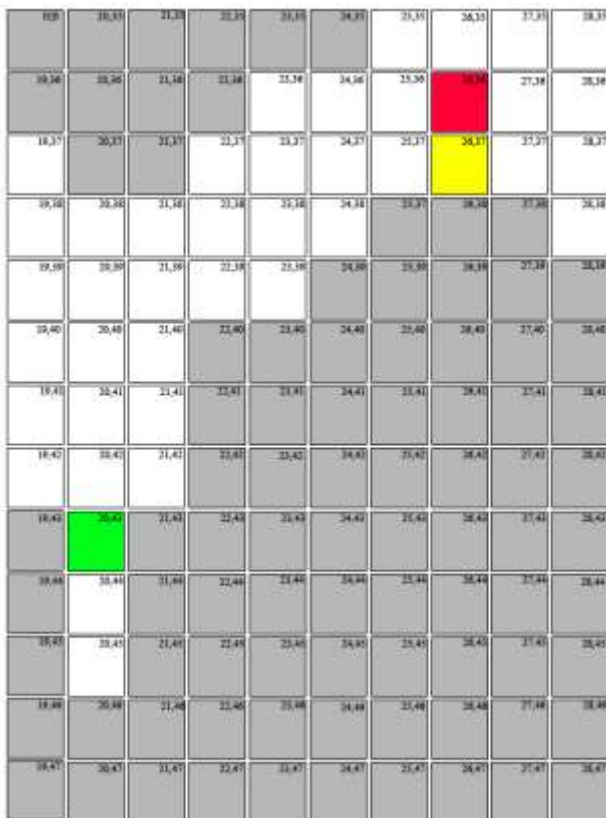
Gambar 11. Simulasi A star

Pada gambar 4.18, node (26,32) pada awalnya dimasukkan ke dalam open list dan selanjutnya memeriksa node yang berdekatan dengan node (26,32) yaitu, node (25,35), (26,35), (27,35), (27,36), (27,37), (26,37), (25,37), (25,36). Setelah diperiksa node node tersebut dimasukkan kedalam openlist, hapus node (26,32) dari open list dan masukkan node tersebut ke dalam closed list.

Pemilihan node dengan nilai F terendah:

Node (25,35)	Node (26,35)	Node (27,35)	Node (25,36)	node (25,37)	node (27,37)	node(27,36)
$F = G + H$	$F = G + H$	$F = G + H$	$F = G + H$	$F = G + H$	$F = G + H$	$F = G + H$
$F = 14 + 130$	$F = 10 + 140$	$F = 14 + 150$	$F = 14 + 120$	$F = 10 + 110$	$F = 10 + 130$	$F = 14 + 140$
$F = 144$	$F = 150$	$F = 164$	$F = 134$	$F = 120$	$F = 140$	$F = 154$
Node (27,36)	node (27,37)	Node (26,37)				
$F = G + H$	$F = G + H$	$F = G + H$				
$F = 10 + 140$	$F = 14 + 130$	$F = 10 + 120$				
$F = 150$	$F = 144$	$F = 130$				
Node (25,37)	node (25,36)					
$F = G + H$	$F = G + H$					
$F = 14 + 110$	$F = 10 + 120$					
$F = 124$	$F = 130$					

Lalu dari node node yang berada di dalam open list, pilih node dengan nilai F yang paling rendah, yaitu node (25,37) dengan nilai $F = 124$. Tetapi, pada game engine RPG Maker ini hanya bisa berjalan horizontal dan vertikal, maka nilai pada jalan diagonal tidak bisa dipakai, yang di ambil hanya nilai pada horizontal dan vertikal, yaitu node (26,37).

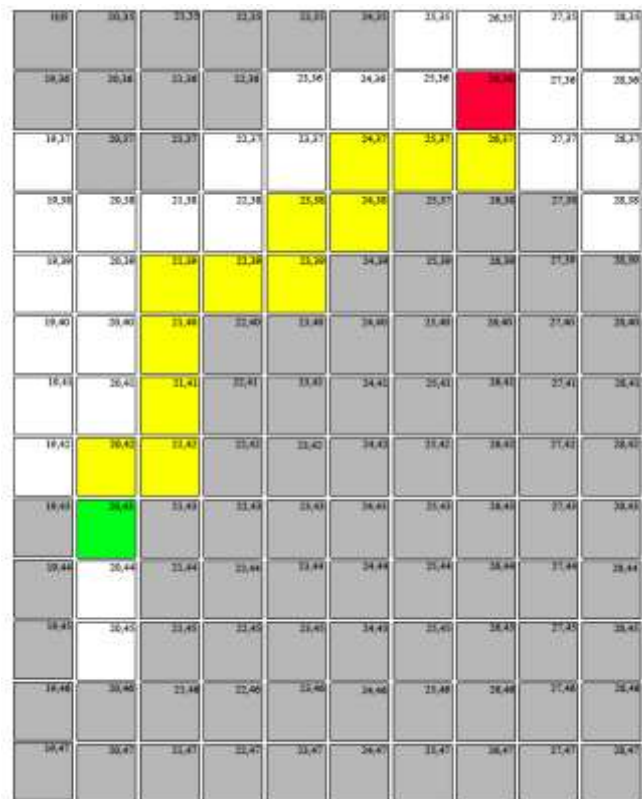


Gambar 12. Simulasi A

Ketika musuh berada di node (26,37) algoritma A* membangkitkan kembali node-node yang berdekatan untuk dicari nilai F nya.

Karena nilai F pada node (25,37) memiliki nilai terendah, maka dipilih node (25,37) sebagai node terpilih untuk dijadikan rute selanjutnya.

Pada gambar 12 merupakan contoh nilai yang didapatkan dari perhitungan pencarian nilai dari beberapa titik terdekat. Mulai dari titik awal (26,36) ke (25,37). Hasil perhitungan rute terdekat pada titik (26,36) sampai dengan titik (20,43) terlihat pada gambar 13



Gambar 13. Simulasi A

Ketika node telah sampai pada node tujuan, maka perhitungan akan dihentikan dan rute akan trace kebelakang, pencarian selesai dan rute tersebut adalah rute terpendek berdasarkan pencarian algoritman A Star. Jika ditengah perjalanan tujuan berubah koordinat, maka NPC akan menuju tempat terakhir dimana player berdiri, dan setelah sampai pada node player, maka posisi tersebut ditetapkan sebagai node awal untuk mencari rute terdekat dari player. Kemudian perhitungan di ulang kembali hingga NPC menemukan koordinat terbaru dari posisi pemain.

IV. KESIMPULAN

Berdasarkan hasil dari implementasi dan pengujian yang dilakukan peneliti, maka dapat ditarik kesimpulan bahwa penggunaan algoritma A* berhasil di terapkan pada game Action-RPG sebagai pembangkit perilaku pencarian pada

NPC, dan NPC mampu berjalan dan melewati halangan yang ada, dan berhasil menemukan keberadaan karakter utama atau player.

REFERENSI

- [1] John Von Neumann and Oskar Morgenstern, 3d ed. 1953. *Theory of games and economic behaviour*
- [2] Miftachul Arif, Yunifa. 2010. Integrasi Hierarchy Finite State Machine dan Logika Fuzzy Untuk Desain Strategi NPC Game.
- [3] Michael Hitchens, Anders Drachen, “The Many Faces of Role-Playing Games”, Macquarie University Australia, 2007.
- [4] Zou, Huilai., Qu, Zening., Qu, Youtian. Optimized Application and Practice of A* Algorithm in Game Map Path-Finding.
- [5] Mark Owen Riedl& AlexanderZook, AI for Game Production,
- [6] ChirstophSalge, Christian Lipski, Tobias Mahlamann, Brigitte mathiak. Using Genetically Optimized Artificial Intelligence to improve Gameplaying Fun for Strategical Games
- [7] Yan Li, Lan-Ming Su, Qiang He, “Case-Based Multi-Task Pathfinding Algorithm”, IEEE Journal, 2012