RANCANG BANGUN GAME MAZE 2D "RETURN TO EARTH"

Ridwan Rismanto[1], Dyah Ayu Irawati[2], Ari Mahardika Ahmad Nafis[3]

[1,2,3]Jurusan Teknologi Informasi, Program Studi Teknik Informatika, Politeknik Negeri Malang
rismanto@polinema.ac.id[1], dyah.ayu@polinema.ac.id[2], arimahardika.an@gmail.com[3]

Abstrak

Salah satu unsur dari *game* yang membuat suatu *game* menjadi menarik adalah keberadaan NPC (*non-playable character*) pada *game* yang bisa menjadi lawan main player. Untuk bisa membuat NPC bergerak dan mengambil keputusan seperti manusia diperlukan kecerdasan buatan yang diimplementasikan pada NPC tersebut.Penelitian ini akan dilakukan dengan menggunakan metode A* yang diimplementasikan pada tingkah laku NPC untuk menemukan jalur menuju karakter player. Metode A* (A-Star) merupakan metode untuk melakukan pencarian jalur terdekat dari titik awal ke titik tujuan. Selain itu dalam penelitian ini dikembangkan sebuah *game* 2D berjudul "*Return To Earth*" sebagai media untuk mengimplementasikan metode A* dalam mengendalikan pergerakan NPC. Game ini menceritakan seorang astronot yang berusaha kembali ke Bumi setelah diculik oleh alien. Astronot harus berusaha menuju titik tujuan dengan menghindari alien yang mendekati karakter astronot.Berdasarkan hasil uji coba yang dilakukan, diperoleh kesimpulan bahwa jalur yang diambil oleh algoritma A* berhasil menemukan jalur terpendek untuk setiap level. Dengan demikian algoritma A* ini dapat digunakan untuk mengambil jalur terdekat dari NPC menuju player.

Kata kunci : *Video Game, A* Algorithm, Maze*

1. Pendahuluan

Sampai saat ini peminat industri *game* tidak pernah surut, bahkan semakin meningkat dari waktu ke waktu. Anak-anak, remaja, dewasa, bahkan orangtua juga banyak yang gemar bermain *game* di *handphone, tablet,* atau laptop. Salah satu unsur dari *game* yang membuat suatu *game* menjadi menarik adalah keberadaan AI atau *artificial inteligence* (kecerdasan buatan) pada NPC atau *non-playable character* pada *game* yang bisa menjadi lawan main *player*. Kecerdasan buatan adalah salah satu bidang dalam ilmu komputer yang membuat komputer agar bertindak dan sebaik seperti manusia (menirukan kerja otak manusia).

Game yang akan dikembangkan adalah *game maze* 2D, *single-player*, dengan *map* berbentuk *maze*. Pada *game* yang akan dikembangkan, *player* akan berusaha untuk keluar dari *maze* dengan terus menghindari NPC yang mengejar *character player*, karena NPC pada *game* ini harus bisa mengejar *player*, maka diperlukan algoritma yang bisa membuat NPC bisa mengejar *character player.* Pada *game* yang akan dikembangkan ini, penulis menggunakan algoritma A* yang akan diimplementasikan pada NPC.

Metode A* dikembangkan oleh Peter Hart, Nils Nilsson dan Bertram Raphael. Metode A* adalah sebuah *graph* atau metode pohon pencarian yang digunakan untuk mencari jalan dari sebuah *node* awal ke *node* tujuan (*goal node)* yang telah ditentukan. Metode ini menggunakan "estimasi heuristic" h(n) pada setiap *node* untuk mengurutkan setiap *node* n berdasarkan estimasi rute terbaik yang melalui *node* tersebut. Algoritma A* mencari jalur dengan *cost* terkecil dari *node* awal ke *node tujuan (*Harianja, 2013).

Algoritma A* sering digunakan dalam *game,* seperti dalam penelitian yang dilakukan oleh Agung Pamungkas, Eka Puji Widiyanto, dan Renni Anggreni dengan judul "Penerapan Algoritma A* pada *Game* Edukasi *The Maze Island* Berbasis *Android*" pada tahun 2011.Algoritma A* juga diimplementasikan pada bidang lain selain game, seperti pada penelitian sebelumnya yang telah dilakukan oleh Muhammad Irsyad dan Endang Rasilla dengan judul "Aplikasi Pencarian

Lokasi Gedung dan Ruangan Universitas Negeri Sultan Syarif Kasim Riau pada *Platform* Android Menggunakan Algoritma A-Star (A*)"pada tahun 2015, dapat diketahui bahwa metode A* digunakan untuk pencarian jalur terpendek. Dan didapatkan hasil bahwa Algoritma A-Star memberikan informasi rute terdekat yang akurat dengan 10 kali percobaan dan menghasilkan rute yang sama antara pencarian rute secara manual dengan pencarian rute pada aplikasi (Irsyad).

Penelitian juga dilakukan oleh Mulia Purwati, Okti Firnawati, dan Willy, dengan judul "Penerapan Algoritme A* (*A STAR)* Dalam Optimasi Penentuan Halte Transmusi di Palembang Berbasis Android", dan didapatkan hasil yang sesuai, yang berarti hasilnya sesuai dengan perhitungan manual berdasarkan parameter yang digunakan.

Berdasarkan latar belakang diatas, dilakukan pembangunan game ini sebagai tempat untuk

melakukan penelitian dalam mengimplementasikan AI menggunakan algoritma A* pada NPC game.

2. Kajian Pustaka

2.1 Algoritma A*

Algoritma ini merupakan agoritma *Best First Search* yang menggabungkan *Uniform Cost Search* dan *Greedy Best-First Search*. Biaya yang diperhitungkan didapat dari biaya sebenarnya ditambah dengan biaya perkiraan. Dalam notasi matematika dituliskan sebagai :

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

Dengan $g(n)$ adalah *cost of the path* dari keadaan awal ke *node* n dan $h(n)$ adalah perkiraan heuristik atau *cost* atau *path* dari *node* n ke tujuan. Jadi, $f(n)$ adalah perkiraan total *cost* terendah dari setiap *path* yang akan dilalui *node* n ke *node* tujuan. Dengan kata lain, *cost* adalah jarak yang telah ditempuh, dan panjang garis lurus antara *node* n dengan *node* akhir adalah perkiraan heuristiknya (Purwati). Dengan perhitungan biaya seperti ini algoritma A* adalah *complete* dan *optimal*.

Sama dengan algoritma dasar *Best First Search,* algoritma A* ini juga menggunakan dua senarai: *OPEN* dan *CLOSED*. Terdapat tiga kondisi bagi setiap suksesor yang dibangkitkan, yaitu : sudah berada di *OPEN*, sudah berada di *CLOSED*, dan tidak berada di *OPEN* maupun *CLOSED*. Pada ketiga kondisi tersebut diberikan penanganan yang berbeda-beda.

Jika suksesor sudah pernah berada di *OPEN*, maka dilakukan pengecekan apakah perlu pengubahan *parent* atau tidak tergantung pada nilai *g*-nya melalui *parent* lama atau *parent* baru. Jika melalui *parent* baru memberikan nilai *g* yang lebih kecil, maka dilakukan pengubahan *parent.* Jika pengubahan parent dilakukan, maka dilakukan pula perbaruan (*update*) nilai *g* dan nilai *f* pada suksesor tersebut. Dengan perbaruan ini, suksesor tersebut memiliki kesempatan yang lebih besar untuk terpilih sebagai simpul terbaik (*best node*).

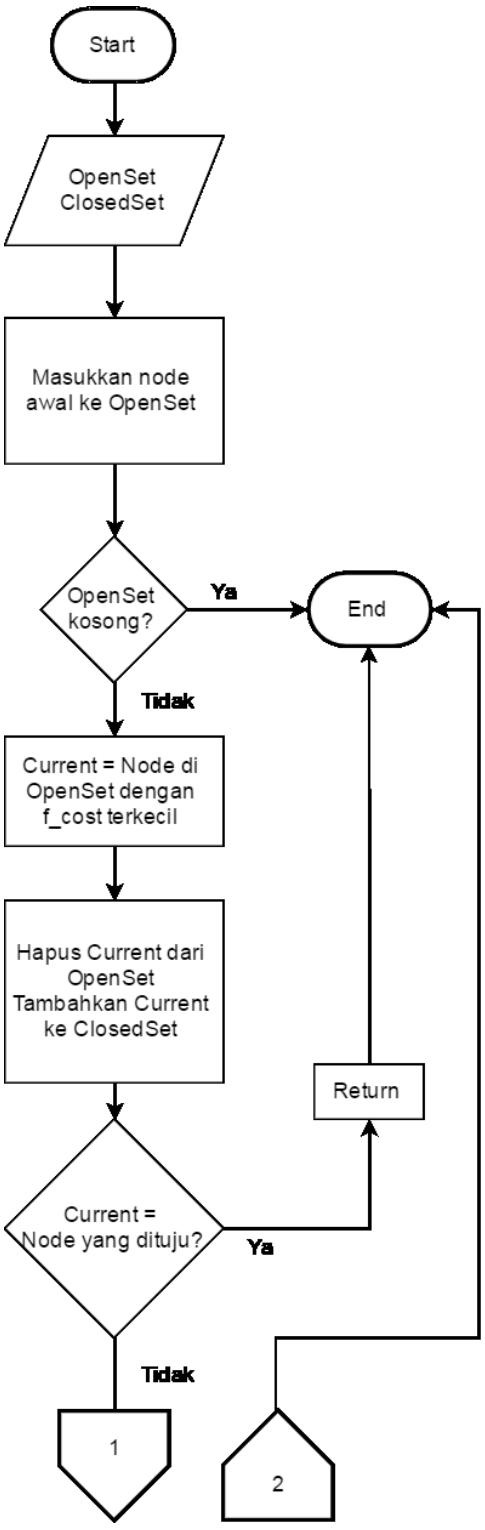
Jika suksesor sudah pernah berada di *CLOSED*, maka dilakukan pengecekan apakah perlu pengubahan *parent* atau tidak. Jika ya, maka dilakukan perbaruan nilai g dan f pada suksesor tersebut serta semua "anak cucunya" yang sudah berada di *OPEN*. Dengan perbaruan ini, maka semua anak cucunya tersebut memiliki kesempatan yang lebih besar untuk terpilih sebagai simpul terbaik (*best node*).

Jika suksesor tidak berada di *OPEN* maupun *CLOSED*, maka suksesor tersebut dimasukkan ke dalam *OPEN*. Tambahkan suksesor tersebut sebagai suksesornya *best node.* Hitung biaya suksesor tersebut dengan rumus $f = g + h$ (Suyanto, 2014)

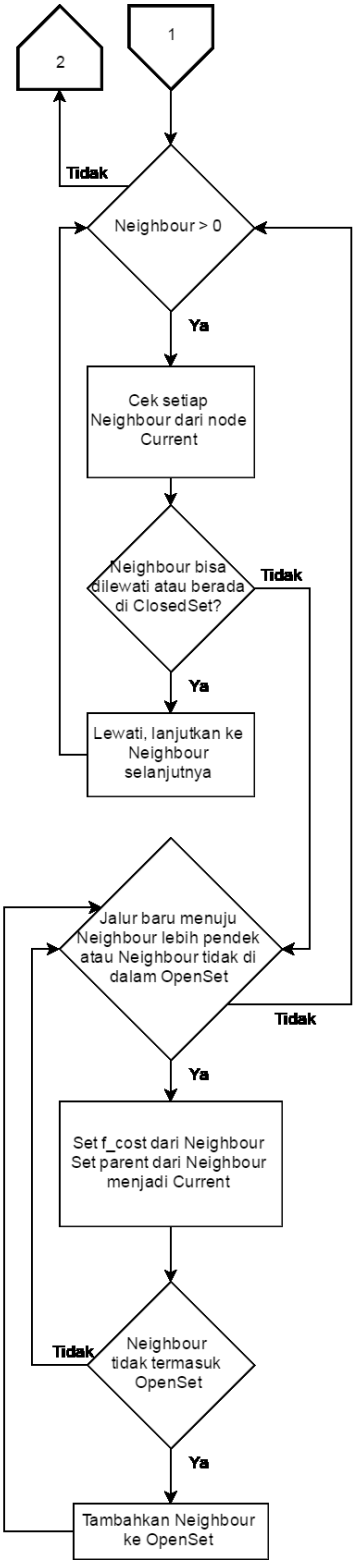
3. Metode Penelitian

3.1 A* Pada NPC

Karena pergreakan NPC tergantung pada jalur yang digunakan oleh pemain itu sendiri, maka algoritma A* diimplementasikan pada NPC. Adapun flowchart algoritma A* adalah sebagai berikut.



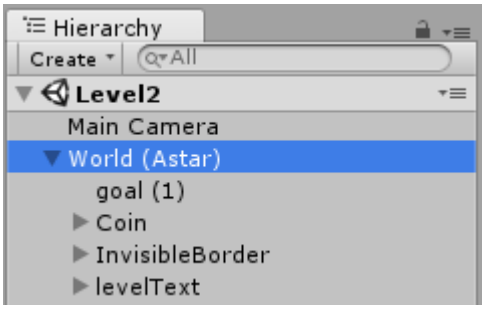
Gambar 3. Flowchart A*



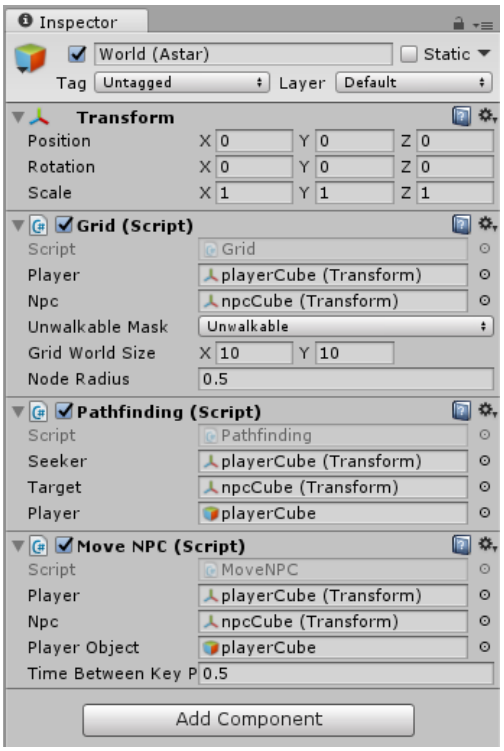
Gambar 4. Flowchart A* (Lanjutan)

4. Hasil dan Pembahasan

Implementasi pada algoritma A* ini diletakkan pada GameObject kosong yang digunakan untuk menampung *script* yang idgunakan untuk mengimplementasikan algoritma A*.



Gambar 5. Tampilan *Hierarchy*



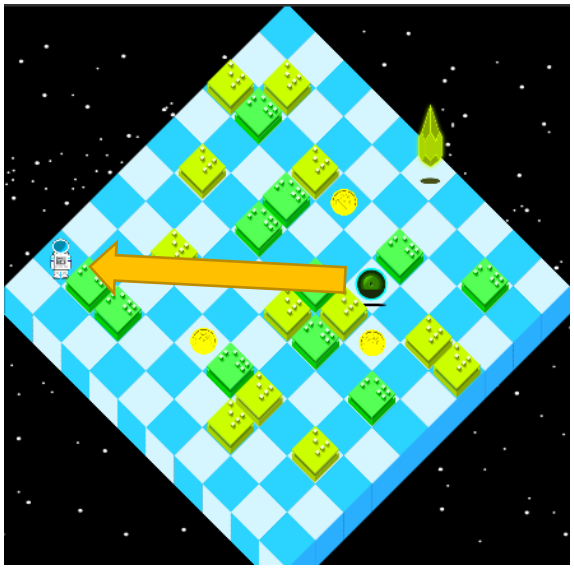
Gambar 6. Tampilan Inspector

Dari gambar diatas diketahui bahwa *GameObject World(Astar)* memiliki 3 buat *script* dimana 2 *script* tersebut merupakan bagian dari algoritma A*, yaitu *Grid* dan *Pathfinding*. *Script Grid* digunakan untuk menentukan posisi *player,* NPC, serta layer apa yang dianggap tidak bisa dilewati. *Grid* juga menentukan ukuran papan yang digunakan dalam level. *Node* radius digunakan untuk mendefinisikan seberapa besar cakupan tiap *node.*

Script Pathfinding digunakan untuk memperhitungkan jalur terdekat dari karakter NPC menuju karakter *player*.

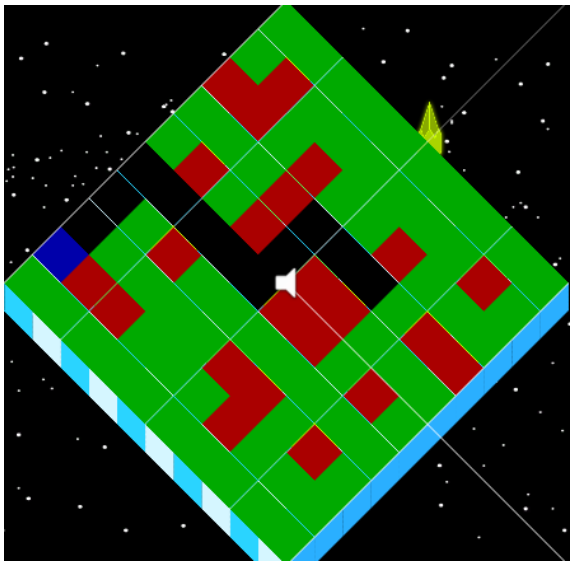
4.1 Pengambilan Jalur dengan Titik Tujuan Diam

Pada gambar dibawah, dilakukan pengujian terhadap A* yang diimplementasikan pada NPC.



Gambar 6. Penentuan Tujuan

Pada Gambar 6. dapat dilihat bahwa untuk NPC bisa mencapai tujuan terdapat benda penghalang yang menutup jalan dari NPC dalam *map*.



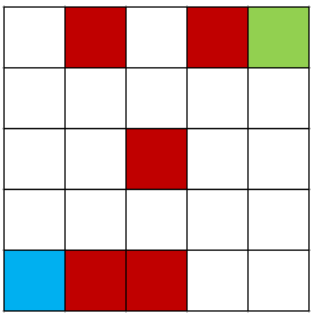
Gambar 7. Pengujian Algoritma A*

Pada Gambar 7, dapat dilihat jalur yang diambil oleh algoritma A*. Garis hitam adalah jalur yang akan diambil oleh NPC, sementara titik merah adalah penghalang, dan titik biru adalah lokasi karakter *player* yang merupakan tujuan dari NPC.

Jalur yang diambil oleh NPC juga akan berubah ketika karakter *player* bergerak.

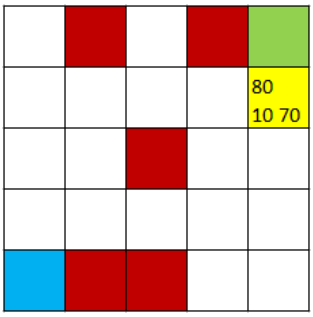
4.2 Proses Perhitungan Dalam Pengambilan Jalur Terdekat

Disini dilakukan pengujian algoritma A* dengan karakter NPC (warna hijau) yang bergerak menuju karakter *player* (warna biru) dengan menghindari penghalang (titik merah).



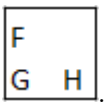
Gambar 8. Layout Map Awal

Pada Gambar 8 dapat dilihat layout awal map dan lokasi awal dari *player* dan NPC.

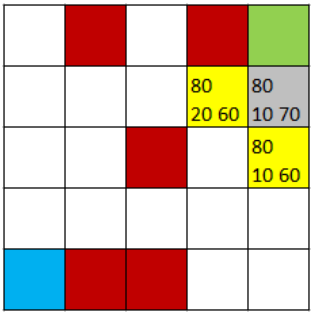


Gambar 9. Proses Pengambilan Jalur Oleh NPC

Pada Gambar 9 proses pengambilan jalur dimulai dengan mengambil jalur ke kotak disekitarnya. Dimana nilai di pojok kanan atas adalah F, pojok kiri bawah adalah G, dan pojok kanan bawah adalah H.

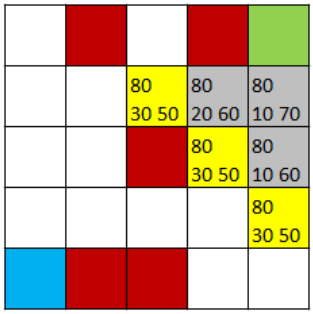


Gambar 10. Format penulisan angka.

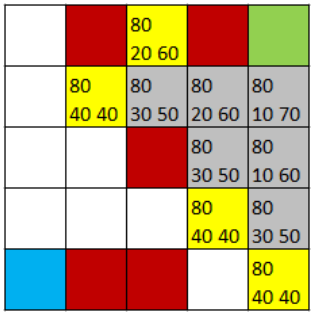


Gambar 11. Proses pengambilan jalur ke-2

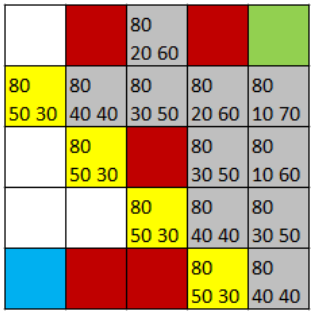
Pada Gambar 11 terdapat kotak berwarna abu - abu, kotak tersebut adalah kotak yang telah terhitung, dan tidak dimasukkan ke dalam perhitungan lagi.



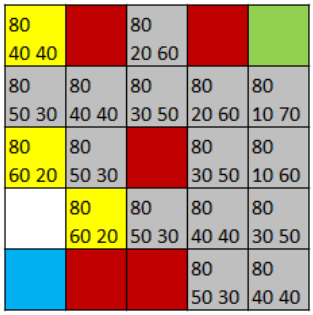
Gambar 12. Proses pengambilan jalur ke-3



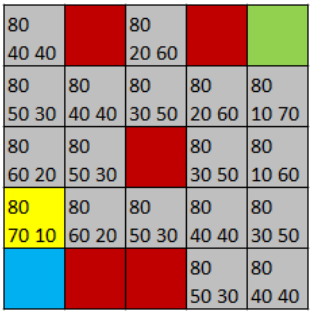
Gambar 13. Proses pengambilan jalur ke-4



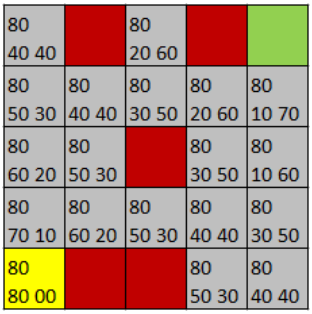
Gambar 14. Proses pengambilan jalur ke-5



Gambar 15. Proses pengambilan jalur ke-6



Gambar 16. Proses pengambilan jalur ke-7



Gambar 17. Proses pengambilan jalur ke-8

Pada gambar 17, pencarian jalur sudah mencapai tujuan dan bisa membuat jalur terpendek dari titik awal menuju titik tujuan.

5. Kesimpulan

Kesimpulan yang didapatkan dari pengujian adalah A* berhasil diimplementasikan dengan baik dan berhasil menemukan jalur terpendek untuk setiap level. Serta hasil yang didapatkan dari kuesioner menunjukkan bahwa pengambilan jalur NPC sangat baik berdasarkan nilai yang diberikan oleh user. Dengan demikian algoritma A* ini dapat digunakan untuk mengambil jalur terdekat dari NPC menuju player.

Daftar Pustaka

Harianja, Firman. 2013. Penerapan Algoritma A* Pada Permasalah Optimalisasi Pencarian Solusi Dynamic Water JUG. Medan: STMIK Budidarma Medan

Irsyad, Muhammad and Rasilla, Endang "Aplikasi Pencarian Lokasi Gedung dan Ruangan Universitas Islam Negeri Sultan Syarif Kasim Riau pada Platform Android Menggunakan Algoritma A-Star (A*)"

Purwati, Mutia and Firnawati, Okti and Willy, Willy "PENERAPAN ALGORITME A* (A STAR) DALAM OPTIMASI PENENTUAN HALTE TRANSMUSI DI PALEMBANG BERBASIS ANDROID." STMIK GI MDP.

Suyanto, 2014, Artificial Intelligence Searching-Reasoning-Planning-Learning : 2.3.2.6 A* (A Bintang), Revisi Kedua, Informatika Bandung :Bandung, 2014, pp33.