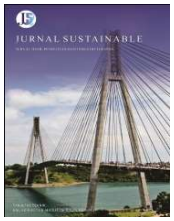


Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan
Vol. 06, No. 01, hal. 16-23, Mei 2017
Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan
ISSN 2087-5347

Simulasi Kendali Pergerakan *Mobile Robot* Menggunakan Algoritma A-star dalam Menentukan Jarak Terpendek

Tonny Suhendra[1,*], Sapta Nugraha[2], Anton H Yunianto[3], Alena Uperiati[4]
[1,2,3,4]Jurusan Teknik Elektro, Fakultas Teknik, Universitas Maritim Raja Ali Haji
[1,2,3]Jl. Politeknik Senggarang, Tanjungpinang 29100
[*]Corresponding Author: tonny@umrah.ac.id

Abstract— Planning the path is one of the most important things in the world of robotics, especially autonomous robots, to be able to move the autonomous robot requires a path that will guide the movement, or steps to be taken next, can also be spelled out as the determination of the point of coordinates to be addressed so that the robot can move to destination by taking the nearest lane and guiding the robot not to take unnecessary steps. This research uses adaptive A-star algorithm as the shortest path finding algorithm, the algorithm used is the development of A-star algorithm so that it can perform path search gradually and done repeatedly to determine every step that must be taken robot in the future and this algorithm belongs to a simple algorithm in a family heuristic algorithm. The test environment is built using Netlogo 5.3.1 Application, an agent-based application developed by Uri Wilensky at the center for Connected Learning and Compute-based Modeling at Northwestern University. The results of the tests have shown that the adaptive A-star algorithm can perform the optimal shortest path search and not trapped in the optimal local conditions with a standard deviation of 0.422%.
Keyword— Path planning, searching algorithm, heuristic algorithm, A-star, mobile robot

Intisari— Perencanaan jalur merupakan salah satu hal yang sangat penting pada dunia robotika terutama *autonomous robot*, untuk dapat bergerak maka robot *autonomous* membutuhkan jalur yang akan menuntuk pergerakan, atau langkah yang harus diambil selanjutnya, bisa juga dibilang sebagai penentuan titik koordinat yang akan dituju sehingga robot dapat bergerak sampai ketujuan dengan menempuh jalur terdekat dan menuntun robot untuk tidak mengambil langkah yang tidak perlu. Penelitian ini menggunakan algoritma *adaptive A-star* sebagai algoritma pencari jalur terpendek, algoritma yang digunakan merupakan pengembangan dari algoritma A-star sehingga dapat melakukan pencarian jalur secara bertahap dan dilakukan secar berulang untuk menentukan setiap langkah yang harus diambil robot kedepannya dan algoritma ini termasuk kedalam algoritma sederhana dalam keluarga *heuristic algorithm.* Lingkungan pengujian dibangun dengan menggunakan Aplikasi Netlogo 5.3.1, merupakan aplikasi berbasis agen yang dikembangkan oleh Uri Wilensky di *center for Connected Learning and Compute-based Modeling Northwestern University.* Hasil pengujian yang telah dilakukan menunjukkan bahwa algoritma *adaptive A-star* dapat melakukan pencarian jalur terpendek dengan optimal dan tidak terjebak pada kondisi local optimal dengan nilai standar deviasi 0.422%.
Kata kunci— Perencanaan jalur, algoritma pencari bertahap, algoritma heuristik, algoritma A-star, robot bergerak

I. PENDAHULUAN

Algoritma perencanaan jalur pada *mobile robot* berkembang sangat cepat, dengan tujuan untuk mencapai optimasi pergerakan *mobile robot*, sehingga robot (*mobile robot*) dapat bergerak secara *autonomous* dengan tujuan yang jelas dan tidak mengambil langkah-langkah yang tidak perlu. Robot yang memiliki kemampuan mandiri atau bersifat *autonomous* atau robot pintar (cerdas) memiliki perbedaan sifat dengan robot yang telah diprogram terlebih dahulu atau *embedded* [1] (program ditaman pada mikrokontrol).

Pada dasarnya sebuah robot cerdas terdiri dari tiga sistem utama yaitu sistem mekanik robot, sistem elektronik robot dan sistem kontrol robot yang saling bekerja sama untuk mencapai tujuan tertentu. Dari ketiga komponen tersebut, kontrol robot merupakan hal yang paling utama menjadi perhatian selama ini, bisa dibilang sebagai otak sebuah robot. Dibutuhkan sebuah sebuah kontrol (otak) yang baik sehingga robot dapat bekerja dengan baik dan kontrol dari robot dikatakan cerdas jika sistem yang diberikan merupakan sistem yang memiliki kemampuan mandiri atau memiliki kepintaran dalam mengambil keputusan berdasarkan informasi yang diterima.

Smart system memiliki tiga komponen penting : (i) interaktif, kolektif, koordinasi dan operasi paralel (ii) kemampuan mengorganisasi diri (iii) kemampuan beradaptasi dan memiliki sifat fleksibel [2], kemampuan tersebut tidak dimiliki oleh *embedded system*, walaupun demikian tanpa adanya *embedded system* maka *smart system* tidak akan terbangun. Perencanaan jalur merupakan salah satu dari beberapa proses yang digunakan untuk membangun sistem robot cerdas (*smart robot system*), penggunaan algoritma sangat dianjurkan dalam membangun sistem cerdas, baik itu penggunaan algoritma tunggal maupun kombinasi beberapa algoritma untuk mencapai suatu tujuan, dan salah satu algoritma yang dapat diimplementasikan pada perencanaan jalur adalah algoritma A-star.

Banyak penelitian telah dilakukan untuk menguji keamampuan Algoritma yang digunakan pada Perencanaan Jalur (*path planning*) robot bergerak, baik itu pengembangan sebuah algortima dan atau kombinasi dua bahkan tiga algoritma sebagai perbandingan algoritma, mencari yang paling optimal. Pergerakan robot pada dasarnya dikendalikan oleh algoritma yang digunakan, semakin baik kemampuan algoritma maka pergerakan robot akan semakin baik. Seperti yang dilakukan oleh [3] menggunakan algoritma D-star (D*) yang telah dimodifikasi, digunakan sebagai navigasi *autonomous* pada lingkungan yang tidak dikenal.

Selain sebagai navigasi algoritma juga digunakan untuk efisiensi waktu [4] menggunakan algoritma A-star (A*), yang dimaksud efisiensi waktu adalah waktu ketika algoritma ketika melakukan eksekusi (*running*), semakin cepat algoritma melakukan ekseskusi maka semakin cepat juga robot mengambil keputusan ketika bergerak. Proses modifikasi algoritma dilakukan untuk mengurangi waktu proses, dimana hasilnya lebih baik jika dibandingkan sebelum dimodifikasi, mencapai 65% dan jalur yang didapat lebih pendek meningkat sebesr 3.4%. Melakukan perbandingan algoritma ACO dan A-star untuk mencari jalur terpendek pada lingkungan dinamis [5] sehingga diketahui algoritma yang lebih optimal dalam mencari jalur terpendek, hasil penelitian mendapatkan bahwa algoritma A-star lebih baik pada lingkungan statis, sementara itu untuk lingkungan dinamis ACO lebih baik dengan mendapatkan jalur terpendek.

Ketika bergerak secara *autonomous,* robot membutuhkan pemandu biasanya disebut sebagai navigasi, yang akan memandu robot untuk mencapai tujuan tanpa mengalami hambatan dan dapat mencapai tujuan dengan selamat. Navigasi terdiri dari empat bagian penting yaitu *perception, localization, cognition* dan *path planning* [6]. *Path planning* sendiri terdiri dari metode klasik (*cell decomposition, potntial field method, subgoal network,* dan *road map*) serta metode heuristik (*heuristic method*), algoritma A*, D* dan ACO diatas termasuk kedalam metode *heuristic*, ada juga *neural network, fuzzy logic,* dan *hybrid algorithm.*

 ISSN 2087-5347 (Print)

Penelitian ini akan menganalisa kinerja algoritma perencanaan jalur dalam memprediksi pergerakan target dengan menggunakan algoritma A*, dan menggunakan aplikasi *Netlogo* sebagai proses simulasi. Lingkungan yang akan digunakan bersifat statis dan dinamis, dan pola pergerakan target yang akan digunakan adalah pola gerak linier, horizontal dan diagonal.

II. METODE PENELITIAN

Seperti telah disebutkan sebelumnya, penelitian ini akan menggunakan Netlogo sebagai media simulasi. Simulasi dipilih karena merupakan salah satu cara yang paling cepat untuk mengetahui kemampuan algoritma yang akan diuji dan juga dapat menghemat biaya karena tanpa melakukan pengujian langsung. Netlogo, adalah aplikasi bebas (*free tools*) berbasis agen, dikembangkan oleh Uri Wilensky di *center for Connected Learning and Compute-based Modeling Northwestern University.*

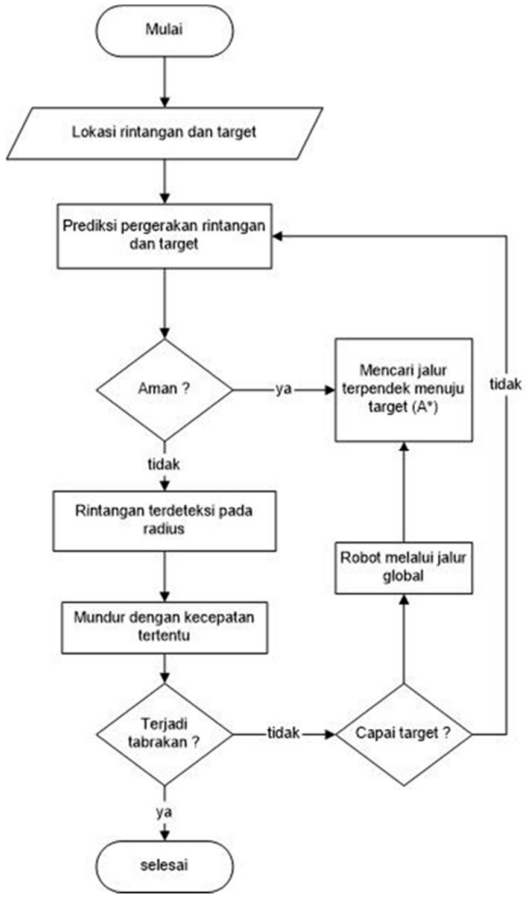
Metode penelitian akan menggunakan metode yang telah dilakukan oleh [5] pada penelitian-nya yang berjudul "Analisis Perbandingan Algoritma Perencanaan Jalur Robot Bergerak Pada Lingkungan Dinamis", metode yang digunakan adalah sebagai berikut :

A. Perencanaan Jalur Robot Bergerak

Perencanaan jalur (*path planning*) dikelompokan menjadi empat bagian utama, yaitu :

1) *Rintangan dan target* : Kelebihan menggunakan Netlogo adalah kemampuan untuk membuat sebuah agen untuk setiap objek, pada penelitian ini terdapat tiga objek yang dianggap penting (robot, rintangan dan target). Ketiga objek tersebut harus didifinisikan dan harus bisa berdiri sendiri, setiap informasi dari setiap agen harus selalu terpantau dan disimpan yang nantinya akan digunakan oleh algoritma ketika melakukan perhitungan (*running process*).

Informasi pertama yang diperlukan adalah posisi target dan rintangan, posisi rintangan digunakan untuk menentukan apakah robot dalam posisi aman selama melakukan pergerakan, sementara itu posisi target digunakan untuk menentukan jarak antara robot dan target, jarak terpendek akan dipilih robot ketika menentukan pergerakan. Pada simulasi robot, rintangan dan target akan menempati koordinat tertentu (*patch*), yang memiliki posisi spesifik terhadap koordinat *y* dan koordinat *x* [7], kedaan ini dikarenakan metode lingkungan yang dibangun pada simulasi berbasis *grid map.*



Gambar 1. Diagram Alir : Perencanaan Jalur *Mobile Robot*

Diagram alir perencanaa jalur *mobile robot* seperti yang terlihat pada Gambar 1, penentuan posisi target dan rintangan harus diketahui terlebih dahulu, baru algoritma pencarian jalur bisa bekerja. Rintangan dan target selain bersifat statis, juga dapat bergerak secara dinamis. Rintangan akan diatur untuk dapat bergerak secara horizontal dan vertikal. Sementara itu untuk target selain horizontal dan vertikal, dapat juga bergerak secara sinusoidal. Pengaturan pergerakan sinusoidal menggunakan persamaan matematis, yaitu $y = A \sin \omega T$, dimana A adalah amplitudo, ω adalah kecepatan sudut dan T merupakan periode getaran (waktu).

Setiap kali objek bergerak (robot, rintangan dan target) maka titik koordinat keberaan-nya akan disimpan didalam sebuah variabel yang nantinya akan dipanggil oleh algoritma ketika dilakukan proses pencarian.

2) *Pergerakan Target (prediksi)* : Pada penelitian ini, objek yang akan diprediksi pola pergerakannya adalah pola pergerakan target, prediksi pergerakan untuk satu langkah berikutnya, sementara itu untuk pergerakan rintangan yang dihitung adalah jarak onjek rintangan dengan robot, jika jaraknya telah mencapai radius tertentu maka algoritma akan memerintahkan robot untuk menagbil langkah yang diperlukan sehingga dapat terhindar dari terjadinya tabarakan.

Memprediksi posisi target akan digunakan konsep yang dikembangkan oleh [7] pada penelitiannya, yaitu prediksi posisi untuk satu langkah didepannya. Algoritma bisa berjalan jika informasi posisi target diketahui untuk posisi saat ini dan posisi target sebelumnya (satu langkah sebelumnya). Persamaan matematis yang digunakan adalah sebagai berikut, persamaan (1) dan persamaan (2) :

$$P_{(n+1)x} = 2P_{nx} - P_{(n-1)x} \quad (1)$$

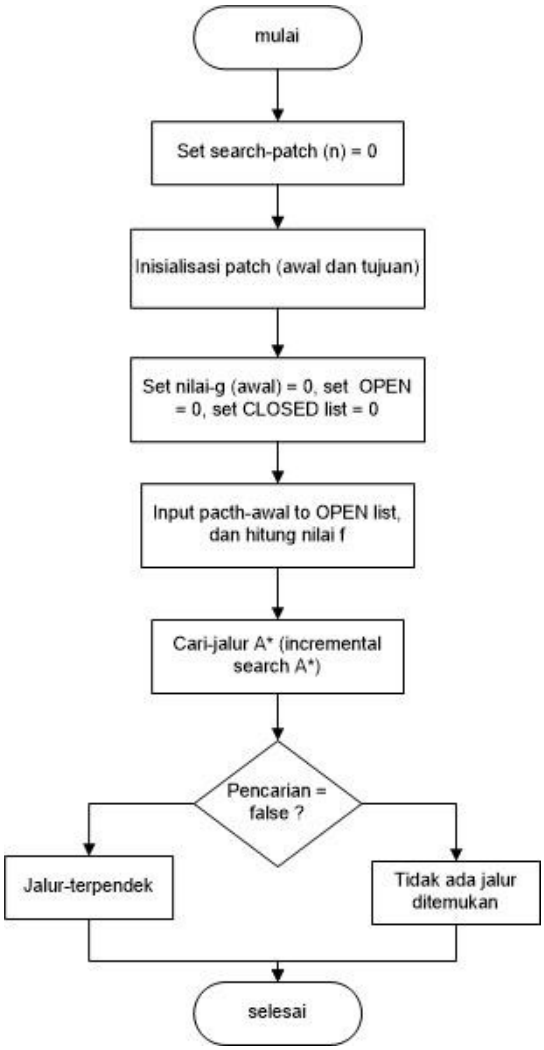
$$P_{(n+1)y} = 2P_{ny} - P_{(n-1)y} \quad (2)$$

$$P_{(n+1)} = [\, P_{(n+1)x} \;\; P_{(n+1)y} \,] \quad (3)$$

Perhitungan yang digunakan adalah dengan mengambil informasi posisi target, yaitu koordinat x dan y pada lingkungan pengujian yang dibangun. $P_{(n+1)x}$ and $P_{(n+1)y}$ merupakan posisi x dan y pada langkah (n+1), satu langkah berikutnya. $P_{(n-1)x}$ dan $P_{(n-1)y}$ merupakan posisi x dan y pada langkah (n-1), satu langkah sebelumnya. $P_{(n+1)}$ pada persamaan (3) merupakaan koordinat hasil perhitungan algoritma pada satu langkah selanjutnya, ketiga persamaan diatas (1), (2) dan (3) dapat digunakan untuk memprediksi pola pergerakan vertical, horizontal dan juga pergerakan diagonal.

3) *Menghindari Rintangan* : Kelebihan dari simulasi adalah mudahnya mengambil posisi spesifik dari masing-masing objek (robot, rintangan, target), karena pada simulasi setiap objek anak dianggap sebagai agen. Ketika posisi rintangan diketahui maka algoritma akan mulai menghitung apakah masih dalam batas aman atau tidak.

Ukuran yang digunakan menggunakan jarak (radius), jika robot masuk dalam radius tertentu (sudah ditetapkan) maka algoritma akan mengkondisikan keadaan aman atau tidak aman sehingga robot dapat melakukan pergerakan menghindar jika diperlukan.



Gambar 2. Diagram Alir : Algoritm Pencari A-star

4) *Algoritma Pencarian Jalur :* Penggunaan algoritma pada path planning menjadi hal yang sangat penting, algoritma dapat meningkatkan kemampuan robot untuk mencapai target dengan cepat (jalur terpendek) dan aman (terhindar dari tabrakan). Algortima tersebut biasanya disebut

 ISSN 2087-5347 (Print)

sebagai algoritma pencari (searching algorithm) yang merupakan satu kesatuan algoritma yang digunakan pada perencanaan jalur.

Algoritma pencari pada penelitian ini akan menggunakan konsep yang dikembangkan oleh [8], *incremental search Adaptive A-star*. *Incremental search* dikembangkan berdasarkan pada metode pencariansebelumnya, proses pencarian kembali (replan) dimulai dari awal lagi (permulaan). Algoritma ini terbagi menjadi 2 kelas utama berdasarkan informasi heuristik :

a. Kelas pertama, memperbaharui nilai heuristic (*h-value*) pada pencarian saat ini dengan menggunakan informasi (*h-value*) dari pencarian sebelumnya, sehingga menyebabkan pencarian saat ini memiliki lebih banyak informasi dan pencarian selanjutnya akan lebih focus.
b. Kelas kedua, proses memperbaharui menggunakan konsep *previous search tree* dari pencarian sebelumnya, sehingga proses pencarian saat ini tidak perlu dilakukan dari awal.

B. Algoritma Pencari A-star

Algoritma yang digunakan sebagai algoritma pencari jalur adalah algoritma berbasis A-star (Gambar 2) yang telah dikembangkan atau dimodifikasi sehingga memiliki sifat *incremental search* atau memiliki sifat yang menggunakan kembali informasi dari pencarian sebelumnya yang digunakan untuk meningkatkan pencarian saat ini dan dengan sehingga dapat menemukan jalur terpendek lebih cepat dari pada algoritma konvensional biasa.

Seperti disebutkan sebelumnya bahwa metode *incremental search* memiliki dua kelas pencarian, pada penelitian ini akan menggunakan kelas pertama pada proses pencarian jalur, salah satu metode yang dikembangkan adalah metode *adaptive A-star* [8]. A* memiliki empat nilai untuk setiap *node s* ϵ *S.*

1. Heuristic *value* (*h-value*) h(s) dari *s* memperkirakan *dist**(*s*,s_{goal}), biaya minimal pergerakan dari *s* ke s_{goal} dari setiap pencarian.
2. Biaya minimal pergerakan dari *node* posisi awaal ke *s, g-value g(s).*
3. Biaya minimal pergerakan dari *node* posisi awal melalui *s* menuju *node* tujuan (target), *f(s)=g(s) + h(s).*
4. *Parent pointer, parent(s) point* ke salah satu dari *node* pendahulunya *s* dari *s, s* merupakan *parent* dari *s* pada *search tree*. *Parent pointer* digunakan untuk menghasilkan jalur setelah pencarian selesai.

Robot akan selalu mengikuti jalur terpendek yang dihasilkan oleh algoritma pencari, dari posisi saat ini ke posisi target, sampai robot mencapai posisi target. Dua data struktur yang dimiliki oleh algoritma A*:

1. OPEN, merupakan daftar yang berisi *node* yang akan dikunjungi, dimulai dari *node* awal dimana nilai *g-value* masih kosong dan juga pada *parent* NULL. Proses yang dilakukan oleh algoritma A* adalah dengan cara berulang akan memindahkan *node s* yang memiliki nilai terkecil pada *g-value* dan *h-value* dari daftar OPEN dan dikirim ke daftar CLOSED.
2. Daftar CLOSED pada dasarnya merupakan *node* yang tadinya ada pada daftar OPEN.

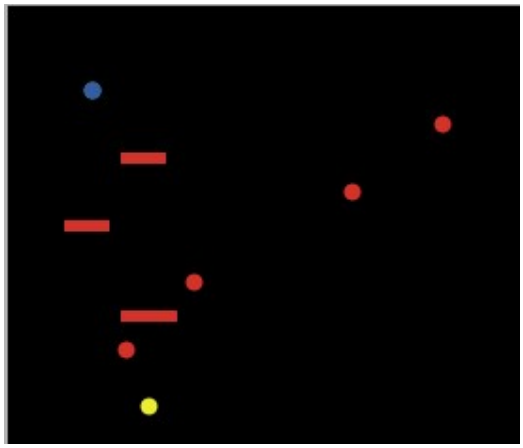
Algoritma pencarian bertahap (*incremental search*) yang digunakan pada penelitian ini adalah algoritma *Adaptive A-star,* berbasis pembelajaran heuristik, untuk setiap pencarian memerlukan nilai *h-value* yang konsisten menuju posisi target, sehingga didapat jalur dengan biaya minimal (*cost-minimal*), dan nilai *h* (*h-value*) akan selalu diperbaharui.

C. Lingkungan Pengujian

Pada penelitian ini akan dibangun sebuah lingkungan pengujian, terdiri dari tiga buah rintangan statis dan 4 buah rintangan dinamis. Statis, menandakan bahwa rintangan tersebut diam tidak bisa bergerak bertujuan menghalangi jalur yang dibangun oleh algoritma. Rintangan dinamis, menandakan bahwa rintangan tersebut dapat bergerak dengan bebas, namun tidak bersifat *random*, arah bergerak memotong jalur yang dihasilkan oleh algoritma, dengan tujuan

untuk melihat kemampuan robot ketika ada objek yang mendekati robot atau menghalangi jalur yang dibangun oleh algoritma.

Lingkungan simulasi yang dibangun dapat dilihat pada Gambar 3, titik biru atas merupakan target, bergerak dari sebelah kiri menuju kekanan dengan pola gerak lurus. Titik kuning merupakan robot yang bergerak mendekati posisi target, robot akan bergerak mengikuti jalur yang dihasilkan. Rintangan statis ditandai dengan kotak persegi panjang berwarna merah (3 kotak), sementara itu rintangan dinamis merupakan titik merah (bulat) berjumlah 4 rintangan yang dapat bergerak.



Gambar 3. Pengujian Algortima Perencanaan Jalur

D. Standar Deviasi

Hasil pengujian akan diukur dengan menggunakan sebuah standar untuk melihat selisih kesalahan (*error*) pada algoritma pencari, untuk melihat sebaran data hasil pengujian, jika sebaran data terlalu besar, maka bisa dikatakan bahwa algoritma tidak memberikan hasil yang konvergen pada solusi, jika semua percobaan memberikan hasil nilai sebaran data nol (0) menunjukkan bahawa algoritma selalu terjebak pada kondisi local optimal. Standar tersebut adalah standar deviasi, persamaan (4) dan persamaan (5) yang diperoleh dari [8].

$$\bar{x} = \frac{\sum_{i=1}^{n} X_i}{n} \quad (4)$$

$$\text{STD} = \sqrt{\frac{\sum_{i=1}^{n} (X_i - \bar{X})^2}{n-1}} \quad (5)$$

Keterangan :

STD = Standar deviasi

X = Jumlah data

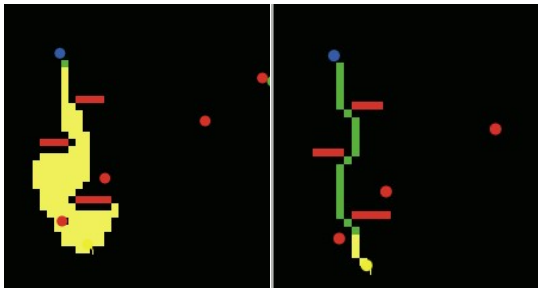
X_i = Nilai data *ke-i*

$\bar{x}$ = Nilai *mean* data

III. HASIL DAN PEMBAHASAN

Aplikasi simulator Netlogo digunakan sebagai media pengujian, dipilih karena aplikasi ini berbasis agen, sehingga setiap objek (robot, rintangan dan target) bisa diberikan sifat-sifat atau kemampuan sesuai dengan keadaan sebenarnya. Seri yang digunakan adalah Netlogo 5.3.1 [9]. Pengujian dilakukan pada lingkungan kompleks dengan target bergerak dengan pola pergerakan linier, dengan dua arah yang berbeda, yang pertama target bergerak mendekati robot dan yang kedua target bergerak menajuhi robot, sehingga nantinya dapat diketahui kemampuan algoritma tersebut dalam mencari jalur terpendek.

Ketika posisi objek (robot, target) diketahui maka algoritma pencari (A*) akan langsung melakukan pencarian, setiap kali melakukan pencarian maka algoritma akan memberikan sebuah titik atau koordinat yang harus dituju oleh robot sebagai langkah selanjutnya. Proses ini akan terus berlanjut sampai robot sampai ketujuan, proses ini lah yang disebut sebagai proses pencarian *incremental search*. Sifat target yang berubah posisi setiap saat, sehingga dibutuhkan sebuah algoritma yang selalu bisa memperbaharui jalur menuju target.



Gambar 4. Pengujian Algortima Perencanaan Jalur

Gambar 4, merupakan proses pencarian algoritma A* sedang berjalan, tanda kuning menandakan banyaknya *patch* yang telah ditelusuri oleh algoritma pencari dan yang

 ISSN 2087-5347 (Print)

berwarna hijau merupakan jalur yang dihasilkan oleh algoritma pencari (A*). Jalur akan selalu diperbaharui dan algoritma akan terus melakukan pencarian, jalur yang terbaik yang akan diberikan ke robot untuk diikuti, robot akan berhenti ketika robot mencapai posisi target.

Jarak tempuh dan waktu eksekusi akan dicatat, dan dilihat untuk mengetahui efektifitas algoritma pencari pada perencanaan jalur robot bergerak.

Tabel 1. Hasil Uji Algoritma Pencari A*

No	Waktu	Jarak
1	62.993	1738.4143
2	62.540	1736.223
3	63.321	1738.4143
4	63.835	1736.223
5	63.008	1738.4143
6	63.071	1738.4143
7	63.024	1738.4143
8	63.258	1738.4143
9	62.416	1736.223
10	62.572	1736.223
Total	630.038	17375.3778

Tabel 1 merupakan hasil uji yang telah dilaksanakan, pengujian dilakukan sebanyak 10 kali dan terlihat bahwa tidak terjadi perbedaan yang mencolok untuk setiap pengujian. Dari hasil perhitungan sebaran data hasil pengujian

IV. KESIMPULAN

Dari hasil pengujian yang diperoleh dapat disimpulkan :

1. Simulasi pengujian pencarian jalur terpendek dengan menggunakan algoritma pencari A* dapat dilakukan dengan menggunakan aplikasi berbasis agen Netologo 5.3.1.
2. Algoritma A* dapat menghasilkan jalur optimal dan tidak terjebak pada kondisi local optimal dengan nilai standar deviasi 0.422%.

dengan menggunakan standar deviasi didapat nilai 0.4222 %, sehingga dapat disimpulkan bahwa algoritma pencari (A*) yang digunakan dapat menghasilkan jalur yang optimal dan tidak terjebak pada kondisi lokal optimal.

UCAPAN TERIMA KASIH

Para penulis ingin menyampaikan apresiasi yang besar untuk Universitas Maritim Raja Ali Haji, khususnya Jurusan Teknik Elektro dan untuk semua orang yang membantu tanpa pamrih

REFERENSI

[1] Wikipedia, Sistem Benam, https://id.wikipedia.org/wiki/Sistem_benam, 14 Maret 2018.

[2] Arndt M., Berns K. (2012) Mobile Robots in Smart Environments: The Current Situation. In: Levi P., Zweigle O., Häußermann K., Eckstein B. (eds) Autonomous Mobile Systems 2012. Informatik aktuell. Springer, Berlin, Heidelberg.

[3] Saranya C et all., "Terrain Based D* Algorithm for Path Planning", IFAC-PapersOnline 49-1 (20016) 178-182

[4] Guruji, K. A., Agarwal, H., Parsediya, D.K., "Time-Efficient A* Algorithm for Robott Path Plannig", 3rd International Conference on Innovations in Automation and Mechatonics Engineering, Procedia Technology 23 (2016) 144-149

[5] Suhendra, T and Priyambodo, T., "Analisis Perbandingan Algoritma Perencanaan Jalur Robot Bergerak Pada Lingkungan Dinamis," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 11, no. 1, p. 21, Jan. 2017.

[6] Mac, T.T et all., "Heuristic Approaches in Robot Path Planning : A survey", Robotic and Autonomous Systems 86 (2016) 13-28.

[7] Via, Y. V., "Optimasi Pencapaian Target Pada Simulasi Perencanan Jalur Robot Bergerak di Lingkungan Dinamis." *JUTI: Jurnal Ilmiah Teknologi Informasi* 10.1 (2012): 15. Web. 11 May 2015.

[8] Currel, G., and Dowman, A., "Essential Mathematics and Statistic for Science Second Edition." ISBN 978-0-470-69449-7, ISBN 978-0-470-69448-0, by Jon Wiley & Sons, Ltd., Publication 2009.

[9] Xiaoxun, S., "Incremental Search-based Path Planning for Moving Target Search-Proquest." N.p., 2013.