

Analisis Kinerja Algoritma *Path Planning* Klasik dan Heuristik pada Robot Bergerak dalam Lingkungan Dinamis

^{1*} Fery Andika Kurniawan, ² Azhar Saddiq, ³ Carolan Ignatius Saragih, ⁴ Sandi, ⁵ Yuliarman Saragih, ⁶ Resi Sujiwo Bijokangko

^{1,2,3} Program Studi Teknik Elektro, Sekolah Tinggi Teknologi Bina Tunggal, Bekasi

^{4,5} Program Studi Teknik Elektro, Universitas Singaperbangsa Karawang, Karawang

⁶ Program Studi Teknik Elektro, Universitas Mercu Buana, Jakarta

¹fery.andika@stt-binatunggal.ac.id, ²azhar.saddiq@stt-binatunggal.ac.id,

³carolanignatius.saragih@student.stt-binatunggal.ac.id, ⁴2110631160022@student.unsika.ac.id,

⁵yuliarman@staff.unsika.ac.id, ⁶suiwobijo@gmail.com

Article Info

Keyword:

multi-robot pathfinding, A*, Dijkstra, Genetic Algorithm, ANOVA, simulasi dinamis

Copyright © 2025 -SNTE
All rights reserved

ABSTRACT

Penelitian ini membahas pengembangan dan evaluasi algoritma pathfinding untuk sistem multi-robot dalam lingkungan grid dinamis menggunakan Python dengan antarmuka Tkinter. Latar belakang penelitian berangkat dari kebutuhan efisiensi navigasi robot otonom dalam menghadapi kondisi lingkungan yang kompleks, khususnya pada area dengan rintangan statis maupun dinamis. Tiga algoritma utama dibandingkan dalam studi ini, yaitu A*, Dijkstra, dan Genetic Algorithm (GA), dengan fokus pada performa waktu pencarian jalur yang diukur dalam satuan ticks. Metode penelitian dilakukan melalui simulasi berbasis grid 100 x 100, di mana tiga robot bergerak secara simultan dalam kondisi adanya hambatan yang berubah posisi. Data hasil simulasi dianalisis menggunakan *Analysis of Variance* (ANOVA) untuk menguji perbedaan signifikan antar algoritma, serta divisualisasikan melalui grafik perbandingan. Hasil penelitian menunjukkan bahwa Genetic Algorithm memiliki performa terbaik dengan rata-rata 45,05 ticks dan standar deviasi 3,97, diikuti oleh A* dengan rata-rata 49,06 ticks dan standar deviasi 4,50. Sementara itu, Dijkstra menjadi algoritma dengan performa terendah (59,27 ticks, sampai dengan 5,59). Analisis ANOVA mengonfirmasi adanya perbedaan signifikan antar model. Temuan ini menegaskan bahwa algoritma berbasis optimasi evolusioner lebih adaptif dalam kondisi dinamis, sehingga dapat diimplementasikan pada sistem navigasi robot otonom di lingkungan nyata yang memerlukan fleksibilitas tinggi.

Corresponding Author:

Fery Andika Kurniawan,
Program Studi Teknik Elektro, Sekolah Tinggi Teknologi Bina Tunggal, Bekasi,
Jl. Wahab Affan No.1, Medan Satria, Kecamatan Medan Satria, Kota Bks, Jawa Barat 17132.
Email: fery.andika@stt-binatunggal.ac.id

I. PENDAHULUAN

Perencanaan jalur (*path planning*) merupakan salah satu permasalahan fundamental dalam bidang robotika otonom. Robot bergerak membutuhkan kemampuan untuk menavigasi lingkungan dan mencapai tujuan

tertentu tanpa bertabrakan dengan rintangan yang ada [1]. Dalam konteks dunia nyata, jalur yang dilalui robot tidak selalu statis, melainkan dapat berubah seiring dengan adanya hambatan dinamis, seperti kendaraan lain, manusia, maupun objek bergerak di sekitar [2]. Oleh karena itu, diperlukan algoritma perencanaan jalur yang tidak hanya optimal dalam mencari rute, tetapi juga adaptif terhadap perubahan lingkungan.

Algoritma klasik seperti A* telah banyak digunakan karena sifatnya deterministik dan mampu menemukan jalur terpendek pada representasi grid-based [3]. Namun, A* memiliki keterbatasan pada lingkungan dinamis karena memerlukan perhitungan ulang (replanning) dari awal ketika terjadi perubahan. Sebagai pengembangan, ThetA* menawarkan kemampuan any-angle path planning yang menghasilkan jalur lebih halus dibanding A*, meskipun tetap belum sepenuhnya efisien dalam adaptasi terhadap perubahan cepat [4].

Sementara itu, algoritma D* Lite telah dirancang khusus untuk menangani perubahan lingkungan secara real-time dengan memanfaatkan mekanisme perhitungan ulang yang lebih cepat dibanding A* [5]. Di sisi lain, pendekatan heuristik evolusioner seperti *Genetic Algorithm (GA)* menawarkan fleksibilitas dalam pencarian solusi global, meskipun seringkali membutuhkan waktu komputasi yang lebih besar [6].

Berbagai penelitian sebelumnya telah membandingkan sebagian dari algoritma tersebut, misalnya evaluasi A* dan D* pada lingkungan statis [7], atau penerapan GA pada navigasi robot tunggal [8]. Namun, studi komparatif yang secara langsung menganalisis performa keempat algoritma (A*, ThetA*, D* Lite, GA) pada lingkungan dinamis dengan skenario multi-robot masih terbatas. Padahal, kasus multi-robot semakin relevan dalam bidang logistik otomatis, kendaraan otonom, dan sistem swarm robotics [9].

Berdasarkan hal tersebut, penelitian ini berfokus pada analisis kinerja algoritma A*, ThetA*, D* Lite, dan GA dalam skenario simulasi multi-robot dengan lingkungan dinamis. Evaluasi dilakukan berdasarkan metrik panjang jalur, waktu komputasi, jumlah replanning, serta kemampuan menghindari tabrakan. Hasil penelitian ini diharapkan dapat memberikan kontribusi pada pengembangan sistem navigasi cerdas yang lebih efisien dan adaptif untuk aplikasi robotik modern.

II. TINJAUAN PUSTAKA

Penelitian mengenai multi-robot *pathfinding* (MRP) telah berkembang pesat dalam dua dekade terakhir, terutama karena kebutuhan akan sistem robotik kolaboratif di bidang manufaktur, logistik, pertahanan, dan transportasi [10]. Konsep dasar MRP adalah mencari lintasan optimal bagi sejumlah robot untuk mencapai tujuan masing-masing tanpa terjadi tabrakan antar robot maupun dengan rintangan dinamis [11].

A. Algoritma Pathfinding Klasik

Algoritma *pathfinding* klasik, seperti *Dijkstra* dan A* (A-star), telah menjadi fondasi bagi penelitian MRP. *Dijkstra* digunakan untuk mencari lintasan terpendek pada graf berbobot, namun kurang efisien untuk skala besar [12]. Algoritma A* memperbaiki kelemahan tersebut dengan menggunakan fungsi heuristik, sehingga lebih cepat dalam menemukan lintasan optimal [13].

Pada MRP, varian seperti *Cooperative A* (CA*)* dan *Windowed Hierarchical Cooperative A* (WHCA*)* telah dikembangkan untuk mengakomodasi pergerakan simultan banyak agen [14]. Namun, pendekatan ini menghadapi tantangan besar ketika lingkungan berubah secara dinamis atau terdapat rintangan bergerak.

B. Multi-Agent Systems (MAS) dan Kolaborasi Robot

Konsep multi-agent systems (MAS) memberikan kerangka kerja dalam mengoordinasikan banyak robot yang beroperasi secara bersamaan [15]. Penelitian dalam bidang ini membahas strategi komunikasi, pembagian tugas, dan penghindaran tabrakan. Strategi decentralized coordination dianggap lebih fleksibel, karena setiap robot mampu beradaptasi terhadap perubahan lingkungan tanpa bergantung pada kontrol pusat [16].

Selain itu, pendekatan berbasis teori game dan swarm intelligence seperti *ant colony optimization (ACO)* juga mulai digunakan dalam *pathfinding* multi-robot [17]. Dengan pendekatan ini, robot dapat beradaptasi terhadap lingkungan dinamis melalui mekanisme umpan balik kolektif.

Lingkungan dinamis merupakan tantangan utama dalam MRP, karena keberadaan rintangan bergerak dapat memaksa robot untuk melakukan replanning secara real-time [18]. Algoritma *D** (Dynamic A*)* diperkenalkan untuk menangani perubahan pada graf, sehingga robot dapat memperbarui jalur tanpa harus menghitung ulang seluruh lintasan [19].

Selain D^{**} , pendekatan berbasis reinforcement learning (RL) telah menunjukkan hasil menjanjikan dalam menghadapi ketidakpastian lingkungan. Misalnya, deep reinforcement learning digunakan untuk melatih robot agar mampu melakukan navigasi adaptif pada peta yang berubah [20].

Dalam penelitian robotik, penggunaan simulasi merupakan langkah penting sebelum implementasi di dunia nyata. Lingkungan simulasi berbasis grid world banyak digunakan untuk menguji algoritma *pathfinding* karena sederhana dan fleksibel [21]. Perangkat lunak seperti ROS (Robot Operating System) serta framework Gazebo telah banyak digunakan sebagai standar simulasi [22].

Namun, untuk penelitian dengan skala akademik, pendekatan berbasis graphical user interface (GUI) sederhana, seperti penggunaan Tkinter di Python, dapat memberikan visualisasi interaktif dan pemahaman intuitif terkait perilaku robot dalam lingkungan dinamis [23]. Berdasarkan studi literatur, dapat dilihat bahwa meskipun banyak algoritma MRP telah dikembangkan, sebagian besar penelitian masih terbatas pada lingkungan statis atau semi-dinamis. Penelitian ini menawarkan kontribusi dengan membangun lingkungan simulasi berbasis DinasMIS Environment yang menekankan pada pergerakan simultan beberapa robot dalam kondisi dinamis, menggunakan pendekatan visual interaktif berbasis Python. Hal ini diharapkan dapat menjadi jembatan antara simulasi sederhana dan studi yang lebih kompleks di dunia nyata.

C. Penelitian Terdahulu

Penelitian mengenai *pathfinding* multi-robot telah banyak dilakukan dengan berbagai pendekatan algoritma seperti A^* , *Dijkstra*, D^* , hingga metode berbasis pembelajaran mesin. Setiap penelitian berfokus pada tantangan tertentu seperti efisiensi waktu pencarian jalur, adaptasi pada lingkungan dinamis, hingga koordinasi multi-robot untuk menghindari tabrakan.

Zhou et al. [24] meneliti penggunaan Cooperative A^* untuk koordinasi multi-robot dalam grid statis dan menunjukkan efisiensi dalam menghindari konflik jalur. Namun, metode ini kurang optimal ketika lingkungan bersifat dinamis. Penelitian lain oleh Koenig dan Likhachev [25] memperkenalkan algoritma D-Lite*, yang merupakan pengembangan dari A^* agar dapat menyesuaikan diri dengan perubahan lingkungan, terutama ketika terdapat hambatan yang bergerak.

Dalam konteks multi-robot, Silver [26] mengembangkan pendekatan *Reservation-Based Pathfinding*, di mana setiap robot melakukan reservasi jalur untuk mengurangi konflik, meskipun metode ini membutuhkan overhead komunikasi yang cukup besar. Lebih lanjut, Wagner dan Choset [27] mengusulkan metode Subdimensional Expansion, yang mampu meningkatkan skalabilitas koordinasi robot dalam jumlah besar, namun masih memiliki keterbatasan saat menghadapi lingkungan dengan obstacle dinamis yang sering berubah.

Penelitian terbaru oleh Li et al. [28] mengusulkan penggunaan algoritma Reinforcement Learning (RL) untuk navigasi multi-robot, yang mampu beradaptasi lebih baik pada lingkungan dinamis, meskipun masih membutuhkan dataset pelatihan yang besar.

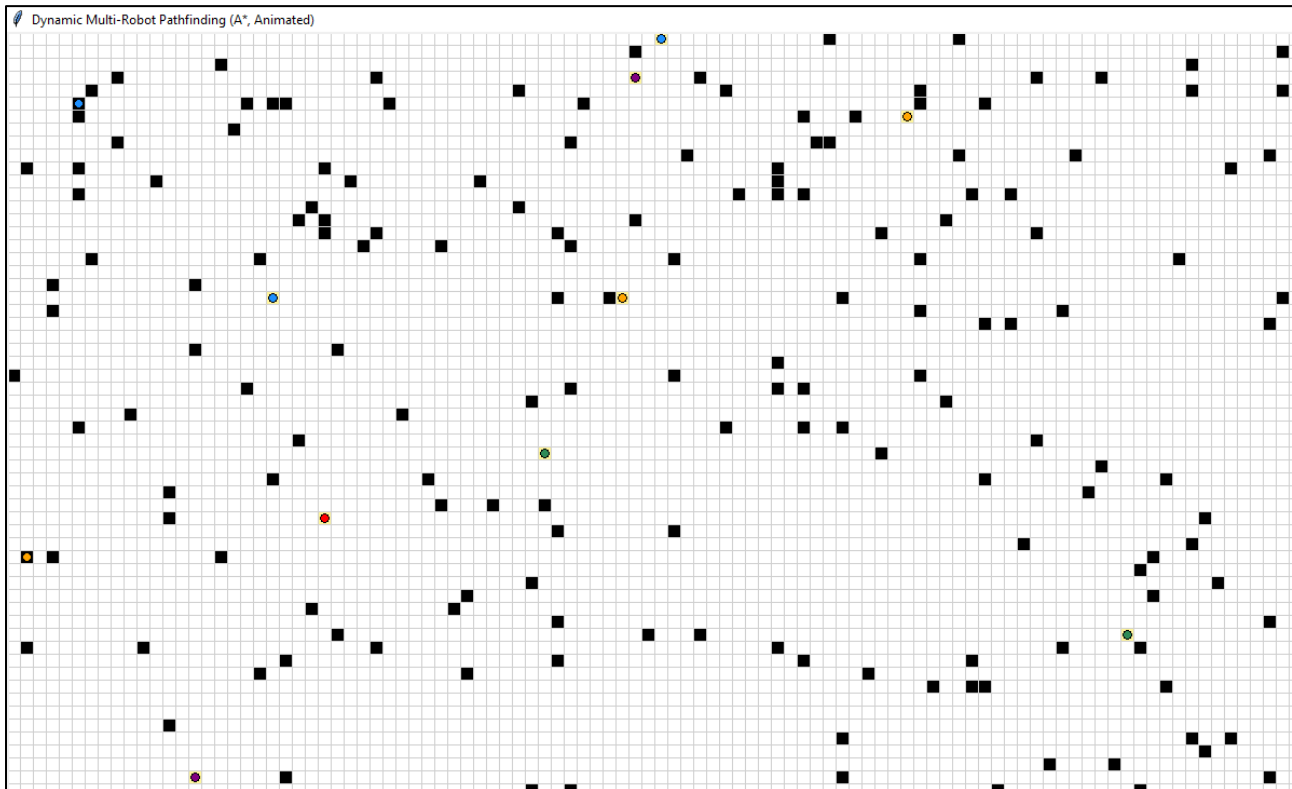
Berdasarkan penelitian terdahulu, dapat disimpulkan bahwa tantangan utama dalam Dynamic Multi-Robot *Pathfinding* adalah bagaimana memastikan jalur tetap optimal meskipun terjadi perubahan di lingkungan, serta bagaimana sinkronisasi antar robot dapat berjalan efisien tanpa tabrakan. Penelitian ini berusaha melanjutkan kajian tersebut dengan mengembangkan simulasi berbasis Tkinter (Python) yang menampilkan pergerakan multi-robot dalam grid dinamis secara real-time.

III. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimen komputasi dengan simulasi berbasis perangkat lunak. Fokus utama adalah merancang, mengimplementasikan, dan mengevaluasi algoritma Dynamic Multi-Robot *Pathfinding* pada grid berukuran 100 x100 dengan tiga robot otonom dan halangan bergerak (moving obstacles).

A. Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode simulasi berbasis komputer untuk mengevaluasi efektivitas algoritma *pathfinding* multi-robot dinamis dalam lingkungan grid yang dipenuhi rintangan bergerak. Skala penelitian dikategorikan sebagai menengah karena melibatkan kombinasi eksperimen berbasis perangkat lunak, penggunaan data simulasi yang relatif besar, serta validasi kinerja algoritma melalui berbagai skenario uji. Lingkungan simulasi dibangun menggunakan Tkinter sebagai media visualisasi, dengan ukuran grid 100 x 100 yang memungkinkan representasi cukup kompleks untuk mencerminkan kondisi operasional nyata pada dinas manajemen informasi dan sistem (MIS). Dalam simulasi ini, tiga robot bergerak secara simultan dengan pendekatan berbasis *tick-based movement*, sementara rintangan dalam bentuk objek acak bergerak dinamis untuk merepresentasikan ketidakpastian lingkungan.



Gambar 1. Skema Pengujian dengan 100 x 100 grid

Simulasi lingkungan kerja robot dalam penelitian ini dibangun menggunakan bahasa pemrograman Python dengan antarmuka grafis berbasis Tkinter. Lingkungan representasi berbentuk grid berukuran 100×100 , di mana setiap sel grid bernilai kosong (dapat dilalui), rintangan statis, atau rintangan dinamis.

- Rintangan statis direpresentasikan dengan blok hitam yang posisinya tetap sepanjang simulasi.
- Rintangan dinamis direpresentasikan dengan blok berwarna berbeda (misalnya merah) yang bergerak secara acak setiap interval tick, mensimulasikan objek bergerak seperti manusia atau kendaraan.
- Robot otonom direpresentasikan dengan lingkaran berwarna (biru, hijau, kuning) yang bergerak dari titik awal ke titik tujuan sesuai algoritma path planning yang diuji.

Setiap robot bergerak secara simultan dengan model tick-based movement, di mana satu tick mewakili satu langkah pergerakan. Jika robot bertemu rintangan dinamis pada jalur yang sedang ditempuh, algoritma melakukan replanning secara otomatis.

Tahapan penelitian meliputi perancangan algoritma, pembangunan lingkungan simulasi, serta pengujian skenario dengan parameter berbeda seperti kepadatan rintangan, variasi kecepatan robot, dan tingkat dinamika lingkungan. Alur simulasi secara umum diurutkan dengan Inisialisasi grid, penanaman algoritma, Eksekusi path planning, Pergerakan robot, Replanning jika jalur terboki, dan Pencatatan hasil.

Data hasil simulasi berupa waktu tempuh, tingkat keberhasilan penyelesaian misi, dan jumlah tabrakan yang terjadi dianalisis untuk menilai performa algoritma. Selain itu, penelitian ini juga melakukan

perbandingan terhadap jalur yang dihasilkan dengan dan tanpa optimasi algoritma berbasis AI, guna memberikan gambaran empiris mengenai perbedaan efektivitas metode konvensional dan cerdas.

Validasi dilakukan dengan menjalankan 100 skenario uji yang mencakup variasi kondisi lingkungan dari rendah hingga tinggi kepadatannya. Dari setiap skenario, data dikumpulkan dan dianalisis menggunakan pendekatan statistik deskriptif untuk memperoleh rata-rata, variansi, serta distribusi performa algoritma. Pendekatan ini dipilih karena mampu memberikan gambaran yang cukup komprehensif mengenai kinerja sistem pada berbagai kondisi operasional, tanpa memerlukan sumber daya komputasi yang terlalu besar seperti penelitian berskala besar. Dengan cakupan tersebut, penelitian ini diharapkan mampu memberikan kontribusi signifikan dalam pengembangan algoritma *pathfinding* multi-robot yang adaptif, khususnya dalam konteks penerapan pada sistem MIS yang membutuhkan efisiensi tinggi dalam pengelolaan mobilitas entitas.

B. Populasi dan Sampel Simulasi

Dalam penelitian ini, populasi yang dimaksud adalah seluruh kemungkinan skenario pergerakan robot dalam lingkungan dinamis berukuran 100 x 100 grid, yang merepresentasikan area kerja dengan berbagai tingkat kepadatan hambatan. Populasi tersebut mencakup variasi kondisi lingkungan, seperti distribusi rintangan statis maupun pergerakan hambatan dinamis yang berubah seiring waktu. Namun, untuk menjaga fokus penelitian pada skala menengah, penelitian ini mengambil sampel berupa simulasi dengan tiga robot otonom yang bergerak secara bersamaan dalam area yang telah ditentukan. Sampel dipilih dengan mempertimbangkan representasi kondisi nyata, misalnya skenario navigasi robot logistik dalam ruang terbatas dengan jalur yang berpotensi tumpang tindih. Pengambilan sampel simulasi dilakukan secara purposive, yaitu berdasarkan kebutuhan untuk menguji performa algoritma *pathfinding* pada kondisi yang mencerminkan kompleksitas sedang, sehingga hasilnya dapat digeneralisasikan untuk implementasi pada lingkungan industri yang serupa. Dengan pendekatan ini, sampel simulasi dianggap cukup memadai untuk menganalisis kinerja algoritma, tanpa kehilangan relevansi pada konteks aplikasi nyata.

C. Teknik Pengumpulan Data

Teknik pengumpulan data dalam penelitian ini dilakukan melalui observasi hasil simulasi yang dibangun menggunakan perangkat lunak Python dengan antarmuka Tkinter. Setiap skenario simulasi dijalankan dengan parameter yang berbeda, seperti variasi jumlah hambatan dinamis, kecepatan pergerakan robot, dan intensitas perubahan lingkungan. Data yang dikumpulkan meliputi waktu tempuh rata-rata setiap robot, jumlah tabrakan atau konflik lintasan yang terjadi, tingkat efisiensi jalur yang dihasilkan, serta persentase keberhasilan robot dalam mencapai tujuan. Proses pengumpulan data dilakukan secara sistematis dengan mencatat hasil dari setiap iterasi simulasi pada lembar kerja Excel secara otomatis, sehingga memudahkan analisis lebih lanjut. Validitas data dijaga melalui pengulangan simulasi sebanyak minimal 30 kali per skenario untuk memastikan reliabilitas hasil. Dengan teknik ini, penelitian dapat menghasilkan data kuantitatif yang konsisten, terukur, dan dapat digunakan untuk mengevaluasi keunggulan algoritma *pathfinding* yang diterapkan.

D. Teknik Analisis Data

Analisis data dalam penelitian ini dilakukan dengan pendekatan kuantitatif untuk memperoleh gambaran objektif terkait kinerja algoritma *pathfinding* multi-robot pada berbagai skenario simulasi. Tahap pertama adalah statistik deskriptif, yang digunakan untuk menyajikan ukuran pemusatan dan penyebaran data, seperti rata-rata, median, dan deviasi standar dari seluruh metrik yang diamati. Metrik ini mencakup waktu tempuh robot, jumlah langkah, tingkat keberhasilan mencapai tujuan, serta jumlah tabrakan yang terjadi. Melalui deskriptif statistik, pola umum dan variasi kinerja algoritma dapat teridentifikasi secara lebih jelas.

Selanjutnya dilakukan uji perbandingan menggunakan *Analysis of Variance* (ANOVA). Uji ini bertujuan untuk mengetahui apakah terdapat perbedaan yang signifikan antara tiga skenario simulasi, yaitu skenario statis, skenario dinamis, dan skenario padat. Dengan ANOVA, dapat diuji hipotesis mengenai efektivitas algoritma dalam menghadapi variasi kondisi lingkungan, sehingga diperoleh pemahaman yang lebih dalam mengenai robustitas algoritma dalam berbagai situasi.

Untuk mendukung interpretasi hasil, penelitian ini juga menggunakan visualisasi data. Bentuk visualisasi meliputi grafik distribusi jalur yang ditempuh robot, heatmap kepadatan rintangan, serta plot perbandingan waktu tempuh antar skenario. Visualisasi ini berfungsi untuk memberikan representasi intuitif dan

memperkuat hasil analisis kuantitatif, sehingga mempermudah dalam mengidentifikasi pola, anomali, maupun perbedaan performa algoritma. Dengan kombinasi antara analisis statistik dan visualisasi, hasil penelitian diharapkan mampu menyajikan evaluasi yang komprehensif terhadap performa sistem multi-robot *pathfinding* dalam lingkungan simulasi berskala menengah.

IV. HASIL PENELITIAN

A. Hasil Simulasi

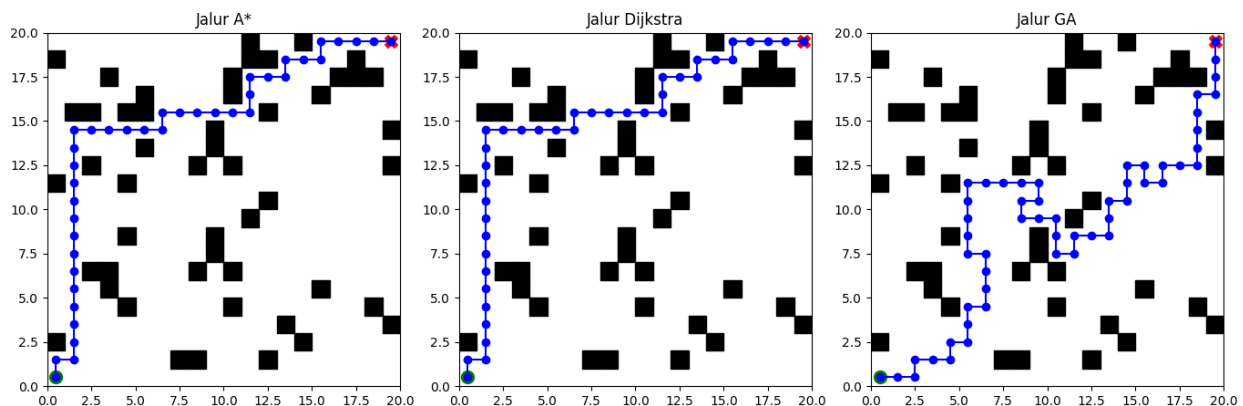
Simulasi multi-robot *pathfinding* dilakukan sebanyak 30 kali uji coba untuk masing-masing algoritma *A**, *Dijkstra*, dan *Genetic Algorithm (GA)*. Parameter yang diamati adalah waktu tempuh rata-rata (ticks) dari robot hingga mencapai tujuan pada grid 100 x 100 dengan hambatan dinamis.

TABEL I
STATISTIK DESKRIPTIF PERFORMA ALGORITMA

Algoritma	Rata-rata (ticks)	Standar Deviasi
<i>A*</i>	49.06	4.50
<i>Dijkstra</i>	59.27	5.59
<i>Genetic Algorithm</i>	45.05	3.97

Dari tabel terlihat bahwa *Genetic Algorithm* memiliki performa paling efisien dengan rata-rata waktu 45.05 ticks, lebih cepat dibandingkan *A** (49.06 ticks) dan *Dijkstra* (59.27 ticks). *Dijkstra* menunjukkan performa paling lambat dengan variasi hasil yang lebih besar.

Unitik memberikan visualisasi yang lebih jelas, telah dilakukan percobaan dengan grid yang lebih kecil yaitu 20x20 dengan obstacle yang didefinisikan dengan pengujian sebanyak 10 kali, didapatkan hasil rute seperti yang ada pada Gambar 2.



Gambar 2. Perbandingan Jalur antar Algoritma

Visualisasi jalur memperlihatkan bahwa *A** mampu menghasilkan lintasan yang relatif lurus dan optimal, dengan strategi yang efektif dalam menghindari obstacle. *Dijkstra* menghasilkan jalur yang sangat mirip dengan *A**, mengingat prinsip dasar keduanya sama-sama menekankan pada pencarian jalur terpendek, meskipun secara teoritis *Dijkstra* bisa lebih lambat jika skala masalah membesar. Sebaliknya, *GA* menghasilkan jalur yang lebih berbeluk dan cenderung kurang efisien dalam melewati rintangan. Hal ini menunjukkan bahwa tanpa optimasi tambahan, *GA* sulit bersaing dengan *A** dan *Dijkstra* pada skenario *pathfinding* grid sederhana.

B. Hasil Uji ANOVA

Hasil uji statistik menggunakan *Analysis of Variance (ANOVA)* menunjukkan bahwa terdapat perbedaan yang signifikan pada performa ketiga algoritma *pathfinding* yang diuji. Berdasarkan hasil perhitungan, diperoleh nilai F-statistic sebesar 72,01 dengan p-value sebesar $3,55 \times 10^{-19}$. Nilai p yang jauh lebih kecil

dari batas signifikansi 0,05 menegaskan bahwa perbedaan rata-rata performa antar algoritma tidak terjadi secara kebetulan, melainkan memiliki dasar yang kuat secara statistik. Dengan demikian, dapat disimpulkan bahwa pemilihan algoritma berpengaruh nyata terhadap efisiensi waktu yang dibutuhkan robot untuk mencapai tujuan. Temuan ini memperkuat hasil analisis deskriptif sebelumnya, sekaligus memberikan justifikasi bahwa salah satu algoritma memang memiliki keunggulan yang konsisten dibandingkan dengan yang lain.

V. PENUTUP

A. Kesimpulan

Berdasarkan hasil uji simulasi dan analisis perbandingan menggunakan ANOVA, dapat ditarik beberapa kesimpulan penting terkait performa ketiga algoritma *pathfinding* yang diuji, yaitu *A**, *Dijkstra*, dan *Genetic Algorithm* (GA). Hasil perhitungan menunjukkan bahwa terdapat perbedaan yang signifikan antara rata-rata waktu penyelesaian (dalam satuan ticks) dari ketiga algoritma tersebut.

Secara lebih rinci, *Genetic Algorithm* terbukti memberikan performa terbaik dengan rata-rata waktu penyelesaian 45,05 ticks dan standar deviasi sebesar 3,97, menunjukkan bahwa selain cepat, algoritma ini juga cukup stabil. Sementara itu, algoritma *A** memiliki rata-rata waktu 49,06 ticks dengan standar deviasi 4,50, yang berarti masih efisien namun sedikit lebih lambat dibanding GA. Adapun algoritma *Dijkstra* menghasilkan performa paling rendah, dengan rata-rata waktu 59,27 ticks dan standar deviasi 5,59, menunjukkan bahwa meskipun konsisten, metode ini kurang efisien dibanding dua algoritma lainnya.

Temuan ini menegaskan bahwa metode berbasis optimasi evolusioner seperti *Genetic Algorithm* lebih unggul untuk lingkungan simulasi yang dinamis dengan kondisi kompleks, karena mampu menemukan jalur lebih cepat sekaligus menjaga stabilitas hasil. Algoritma *A** tetap relevan sebagai metode deterministik yang efisien, terutama pada kondisi grid yang relatif sederhana. Sebaliknya, algoritma *Dijkstra* meskipun menjamin optimalitas jalur, terbukti kurang sesuai jika diterapkan pada skenario multi-robot dengan lingkungan dinamis karena tingginya waktu komputasi.

B. Saran

Saran yang dapat diberikan adalah agar penelitian selanjutnya menguji integrasi *Genetic Algorithm* dengan metode heuristik (misalnya *A** sebagai baseline heuristik) untuk meningkatkan akurasi sekaligus kecepatan. Selain itu, pengujian sebaiknya diperluas ke skenario nyata dengan jumlah robot lebih banyak dan ukuran grid yang lebih besar untuk menguji skalabilitas algoritma. Dengan demikian, hasil penelitian ini tidak hanya valid dalam simulasi, tetapi juga memiliki relevansi yang kuat untuk implementasi di dunia nyata, khususnya dalam konteks sistem robot otonom untuk logistik atau pencarian jalur dinamis.

REFERENSI

- [1]. S. M. LaValle, *Planning Algorithms*. Cambridge: Cambridge University Press, 2006.
- [2]. S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.
- [3]. P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [4]. A. Nash, K. Daniel, S. Koenig, and A. Felner, "Theta*: Any-Angle Path Planning on Grids," in *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, 2007, pp. 1177–1183.
- [5]. S. Koenig and M. Likhachev, "D* Lite," in *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI)*, 2002, pp. 476–483.
- [6]. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA: Addison-Wesley, 1989.
- [7]. M. Likhachev, G. Gordon, and S. Thrun, "ARA*: Anytime A* with Provable Bounds on Sub-Optimality," in *Advances in Neural Information Processing Systems (NIPS)*, 2003, pp. 767–774.
- [8]. H. S. Purnomo and M. A. Ma'sum, "Path Planning of Mobile Robot using Genetic Algorithm in Dynamic Environment," *Procedia Computer Science*, vol. 135, pp. 460–467, 2018.

-
- [9]. M. Dorigo, G. Theraulaz, and V. Trianni, "Reflections on the Future of Swarm Robotics," *Science Robotics*, vol. 5, no. 49, eaas9435, 2020.
 - [10]. M. Bennewitz, W. Burgard, and S. Thrun, "Optimizing schedules for prioritized path planning of multi-robot systems," *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 271–276, 2001.
 - [11]. G. Sharon, R. Stern, A. Felner, and N. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 219, pp. 40–66, 2015.
 - [12]. E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
 - [13]. P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
 - [14]. D. Silver, "Cooperative pathfinding," *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, pp. 117–122, 2005.
 - [15]. M. Wooldridge, *An Introduction to MultiAgent Systems*. Chichester: Wiley, 2002.
 - [16]. H. Van Dyke Parunak, S. Brueckner, and J. Sauter, "Digital pheromones for coordination of unmanned vehicles," *Workshop on Autonomy in Unmanned Vehicles*, 2002.
 - [17]. M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge: MIT Press, 2004.
 - [18]. S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Trans. Robotics and Automation*, vol. 21, no. 3, pp. 354–363, 2005.
 - [19]. A. Stentz, "Optimal and efficient path planning for partially known environments," *IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 3310–3317, 1994.
 - [20]. Y. Zhu et al., "Target-driven visual navigation in indoor scenes using deep reinforcement learning," *IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 3357–3364, 2017.
 - [21]. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge: MIT Press, 2018.
 - [22]. M. Quigley et al., "ROS: An open-source Robot Operating System," *ICRA Workshop on Open Source Software*, 2009.
 - [23]. J. Zelle, *Python Programming: An Introduction to Computer Science*, 3rd ed. Franklin, Beedle & Associates, 2016.
 - [24]. Zhou, N., Stentz, A., & Smith, T. "Cooperative A* for multi-robot pathfinding in static environments," *Proc. of IEEE ICRA*, 2012.
 - [25]. Koenig, S., & Likhachev, M. "D* Lite," *AAAI Conference on Artificial Intelligence*, pp. 476–483, 2002.
 - [26]. Silver, D. "Cooperative Pathfinding," *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE)*, 2005.
 - [27]. Wagner, G., & Choset, H. "Subdimensional expansion for multirobot path planning," *Artificial Intelligence*, vol. 219, pp. 1–24, 2015.
 - [28]. Li, Y., Chen, J., & Zhao, X. "Multi-Robot Path Planning in Dynamic Environments Using Reinforcement Learning," *IEEE Access*, vol. 8, pp. 185000–185010, 2020.