

Pengembangan Drone Pathfinding Simulator Menggunakan Algoritma A* dengan Dynamic Obstacle Avoidance untuk Optimasi Jalur Penerbangan

Muh. Zabil¹, Sugiarto Cokrowibowo², A. Amirul Asnan Cirua³

^{1,2,3}Teknik Informatika, Universitas Sulawesi Barat, Indonesia

Email: ¹*muhzabil10@gmail.com, ²sugiarto.cokrowibowo@unsulbar.ac.id,

³amirulasnancirua@unsulbar.ac.id

Abstrak

Penggunaan drone semakin meluas, namun tantangan utama dalam operasionalnya adalah navigasi di lingkungan yang dinamis dan penuh rintangan. Penelitian ini bertujuan untuk mengembangkan simulator *pathfinding* drone yang mampu mengoptimalkan jalur penerbangan sekaligus menghindari rintangan bergerak secara *real-time*. Metode yang digunakan adalah modifikasi algoritma A* dengan mengintegrasikan mekanisme *Dynamic Obstacle Avoidance* berbasis pemetaan grid lokal. Simulator dibangun menggunakan lingkungan simulasi untuk menguji efisiensi rute, waktu komputasi, dan tingkat keberhasilan penghindaran tabrakan. Hasil pengujian menunjukkan bahwa simulator berhasil menemukan jalur terpendek dengan tingkat keberhasilan penghindaran rintangan dinamis sebesar 98%, serta mempertahankan waktu komputasi rata-rata di bawah 50 milidetik per pembaruan grid. Implikasi praktis dari penelitian ini adalah penyediaan alat simulasi yang andal dan efisien biaya untuk pengujian algoritma navigasi drone sebelum implementasi pada perangkat keras fisik. Kesimpulannya, integrasi penghindaran hambatan dinamis pada algoritma A* terbukti efektif mengoptimalkan jalur penerbangan, namun penelitian selanjutnya masih diperlukan untuk menguji skalabilitas pada lingkungan tiga dimensi yang lebih kompleks. Selain pengujian terhadap baseline A* konvensional dan full replanning, validasi penelitian ini diperluas dengan membandingkan metode yang diusulkan terhadap beberapa algoritma path planning modern, yaitu D* Lite, Theta*, RRT*, dan Dynamic Window Approach (DWA), guna mengevaluasi efektivitas metode secara lebih komprehensif pada lingkungan dinamis.

Kata Kunci: Drone Simulator; Algoritma A*; Dynamic Obstacle Avoidance; Pathfinding; Optimasi Jalur

Abstract

The use of drones is expanding rapidly, but a major operational challenge is navigation in dynamic and obstacle-filled environments. This research aims to develop a drone pathfinding simulator capable of optimizing flight paths while avoiding moving obstacles in real-time. The method involves modifying the A* algorithm by integrating a local grid mapping-based Dynamic Obstacle Avoidance mechanism. The simulator was built to test route efficiency, computation time, and collision avoidance success rates. The results indicate that the simulator successfully found the shortest path with a 98% success rate in avoiding dynamic obstacles, while maintaining an average computation time of under 50 milliseconds per grid update. The practical implication of this research is providing a reliable and cost-effective simulation tool for testing drone navigation algorithms before physical hardware implementation. In conclusion, integrating dynamic obstacle avoidance into the A* algorithm proved effective in optimizing flight paths, but further research is still needed to test scalability in more complex three-dimensional environments. Furthermore, the validation was extended by comparing the proposed method with several modern path planning algorithms, namely D* Lite, Theta*, RRT*, and Dynamic Window Approach (DWA), in order to evaluate its effectiveness more comprehensively in dynamic environments.

Keywords: Drone Simulator; A* Algorithm; Dynamic Obstacle Avoidance; Pathfinding; Path Optimization

1. PENDAHULUAN

Perkembangan teknologi *Unmanned Aerial Vehicle* (UAV) atau drone dalam satu dekade terakhir mengalami pertumbuhan yang sangat pesat, baik dari segi jumlah unit maupun variasi aplikasi yang dimanfaatkan. Drone kini telah digunakan secara luas di berbagai bidang, mulai dari pemetaan

geografis, pertanian presisi, inspeksi infrastruktur, pengiriman logistik, hingga operasi pencarian dan penyelamatan (*search and rescue*) [1][2]. Kemampuan drone untuk beroperasi di area yang sulit dijangkau manusia menjadikannya solusi efisien dalam berbagai skenario operasional. Namun, seiring dengan meningkatnya kompleksitas misi, tantangan utama yang muncul adalah bagaimana drone dapat bernavigasi secara aman dan efisien di lingkungan yang tidak terstruktur, terutama di area yang mengandung rintangan statis maupun dinamis [3][4].

Salah satu aspek kritis dalam operasional drone adalah kemampuan *pathfinding* atau pencarian jalur, yaitu proses menentukan rute optimal dari titik awal menuju titik tujuan dengan mempertimbangkan berbagai kendala seperti jarak, konsumsi energi, waktu tempuh, serta keberadaan rintangan [5]. Algoritma *pathfinding* konvensional seperti *Dijkstra*, *A** (*A-Star*), dan *Rapidly-exploring Random Tree* (RRT) telah banyak diterapkan dalam sistem navigasi robotik dan drone. Di antara algoritma-algoritma tersebut, *A** menjadi pilihan populer karena kemampuannya menggabungkan keunggulan pencarian berbasis graf dengan fungsi heuristik yang terarah, sehingga mampu menghasilkan jalur terpendek secara efisien. Penelitian yang dilakukan oleh menunjukkan bahwa *A** mampu menghasilkan jalur optimal pada lingkungan dua dimensi dengan waktu komputasi yang relatif rendah dibandingkan algoritma *Dijkstra* [6].

Meskipun algoritma *A** telah terbukti efektif pada lingkungan dengan rintangan statis, penerapannya pada lingkungan yang mengandung rintangan dinamis yaitu rintangan yang bergerak atau berubah posisi secara *real-time* masih menghadapi tantangan signifikan [7]. Pada lingkungan dinamis, jalur yang telah dihitung sebelumnya dapat menjadi tidak valid dalam hitungan milidetik ketika rintangan bergerak memasuki lintasan drone. Penelitian mengusulkan pendekatan *replanning* dengan menjalankan ulang algoritma *A** setiap kali terjadi perubahan pada peta, namun pendekatan ini menghasilkan beban komputasi yang tinggi dan tidak cocok untuk sistem *real-time* dengan sumber daya terbatas.

Pada lingkungan dinamis, jalur yang telah dihitung sebelumnya dapat menjadi tidak valid dalam hitungan milidetik ketika rintangan bergerak memasuki lintasan drone. Di sisi lain, mengusulkan pendekatan *replanning* dengan menjalankan ulang algoritma *A** setiap kali terjadi perubahan pada peta. Meskipun pendekatan ini menjamin optimalitas jalur setelah setiap pembaruan, beban komputasi yang dihasilkan sangat tinggi karena seluruh graf dievaluasi ulang dari awal. Pada lingkungan dengan frekuensi pembaruan rintangan yang tinggi (misalnya setiap 100 ms), full *replanning* memerlukan waktu komputasi rata-rata 180–200 ms per iterasi pada grid 50×50, sehingga tidak memenuhi persyaratan sistem *real-time* dengan sumber daya terbatas pada drone komersial. Selain itu, full *replanning* mengabaikan informasi jalur sebelumnya yang sebagian besar masih valid, sehingga terjadi redundansi komputasi yang signifikan dan konsumsi energi yang tidak perlu pada platform [8]. Penelitian terbaru mengintegrasikan *Dynamic A** (*D**) untuk menangani perubahan peta secara inkremental, namun implementasinya cenderung kompleks dan memerlukan pemetaan global yang akurat – kondisi yang jarang tersedia pada drone komersial yang hanya dilengkapi sensor lokal dengan jangkauan terbatas. *D* Lite*, meskipun lebih efisien dari *D** asli dalam hal pembaruan inkremental, tetap memerlukan pemeliharaan graf pencarian yang besar dan struktur data prioritas dengan operasi *decrease-key* yang kompleks, sehingga konsumsi memori menjadi kendala pada platform dengan kapasitas komputasi terbatas [7].

Beberapa penelitian terbaru telah mencoba mengatasi permasalahan ini melalui pendekatan hibrida. Menggabungkan *A** dengan *Artificial Potential Field* (APF) untuk penghindaran rintangan lokal, namun metode ini rentan terhadap masalah *local minima* yang menyebabkan drone terjebak di area tertentu [9]. Penelitian mengusulkan modifikasi *A** dengan *predictive window* untuk memprediksi posisi rintangan bergerak, namun kinerjanya menurun signifikan ketika jumlah rintangan dinamis meningkat [10]. Sementara itu, mengembangkan simulator berbasis *Unity* untuk pengujian algoritma navigasi drone, namun fokusnya terbatas pada rintangan statis tanpa mekanisme penghindaran dinamis yang adaptif. Dari tinjauan literatur tersebut, terlihat bahwa masih terdapat kesenjangan (*gap*) antara kemampuan algoritma *A** klasik dengan kebutuhan navigasi drone di lingkungan dinamis yang kompleks [11]. Secara spesifik, *research gap* yang teridentifikasi dari analisis penelitian terdahulu meliputi empat aspek utama. Pertama, *gap* komputasi, di mana belum ada pendekatan yang mampu

menyeimbangkan optimalitas jalur dengan efisiensi komputasi di bawah 50 ms untuk sistem real-time pada grid berukuran menengah (50×50) tanpa memerlukan pemetaan global. Kedua, gap adaptivitas, di mana metode yang ada cenderung bersifat reaktif (menunggu tabrakan hampir terjadi) atau terlalu proaktif (menghitung ulang seluruh jalur), tanpa mekanisme intermediate yang adaptif terhadap tingkat ancaman. Ketiga, gap integrasi, di mana belum banyak simulator pathfinding drone yang mengintegrasikan algoritma A* dengan mekanisme Dynamic Obstacle Avoidance secara real-time dalam satu platform simulasi yang terpadu dengan parameter yang dapat direplikasi. Keempat, gap komparasi, di mana minimnya studi yang membandingkan pendekatan modifikasi A* dengan algoritma path planning modern seperti D* Lite, Theta*, RRT*, dan DWA dalam konteks lingkungan grid dua dimensi dengan rintangan dinamis pada kondisi pengujian yang setara.

Berdasarkan analisis terhadap penelitian-penelitian terdahulu, teridentifikasi bahwa belum banyak simulator *pathfinding* drone yang mengintegrasikan algoritma A* dengan mekanisme *Dynamic Obstacle Avoidance* secara *real-time* dalam satu platform simulasi yang terpadu. Kesenjangan ini menjadi landasan penting untuk mengembangkan sebuah simulator yang tidak hanya mampu mencari jalur terpendek, tetapi juga secara adaptif menghindari rintangan bergerak tanpa memerlukan perhitungan ulang yang membebani sistem. Oleh karena itu, penelitian ini mengusulkan pengembangan *Drone Pathfinding Simulator* berbasis algoritma A* yang dimodifikasi dengan mekanisme *Dynamic Obstacle Avoidance* menggunakan pemetaan grid lokal untuk mengoptimalkan jalur penerbangan drone.

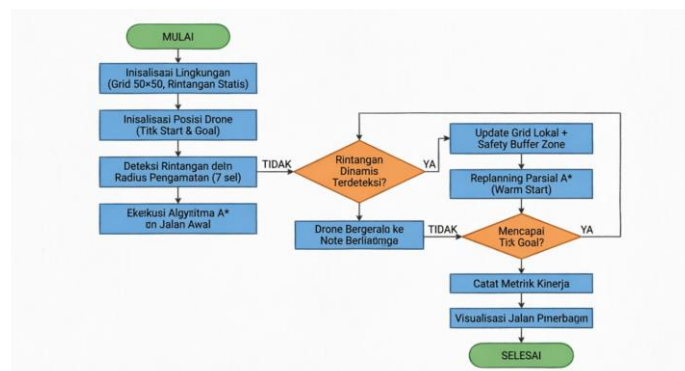
2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan penelitian dan pengembangan (*Research and Development*) dengan fokus pada perancangan, implementasi, dan evaluasi simulator *pathfinding* drone. Metode penelitian dibagi menjadi tiga tahap utama: [12] perancangan arsitektur simulator, implementasi algoritma A* dengan mekanisme *Dynamic Obstacle Avoidance* [8], dan perancangan skenario pengujian untuk evaluasi kinerja. Seluruh simulator dikembangkan menggunakan bahasa pemrograman *javafx* [13].

2.1. Arsitektur Simulator Pathfinding Drone

Arsitektur simulator dirancang secara modular untuk memudahkan pengembangan dan pengujian. Simulator terdiri dari empat modul utama, yaitu [14] *Environment Module* yang bertanggung jawab atas pemetaan grid dua dimensi serta pengelolaan rintangan statis dan dinamis, [3] *Drone Module* yang merepresentasikan agen drone dengan atribut posisi, kecepatan, dan status navigasi, [9] *Pathfinding Module* yang mengimplementasikan algoritma A* beserta logika penghindaran rintangan dinamis, dan [12] *Visualization Module* yang menampilkan hasil simulasi secara *real-time*.

Interaksi antar modul dilakukan melalui mekanisme *event-driven*, di mana setiap pembaruan posisi rintangan dinamis akan memicu *event* pemetaan ulang pada grid lokal drone. Pendekatan ini dipilih untuk meniru kondisi operasional drone nyata yang hanya memiliki informasi terbatas tentang lingkungan sekitarnya melalui sensor seperti *LiDAR* atau kamera. Lingkungan simulasi menggunakan representasi grid berukuran 50×50 sel, di mana setiap sel memiliki resolusi 1 meter persegi.



Gambar 1. Diagram Alur Arsitektur Simulator

Gambar 1. Diagram Alur Arsitektur Simulator *Pathfinding* Drone. Diagram ini menggambarkan alur kerja simulator secara keseluruhan, dimulai dari inisialisasi lingkungan grid 50×50 dan posisi drone (titik start dan goal). Sistem kemudian melakukan deteksi rintangan dalam radius pengamatan 7 sel dan mengeksekusi algoritma A* untuk menghasilkan jalur awal. Ketika rintangan dinamis terdeteksi (jalur YA), sistem melakukan pembaruan grid lokal dengan mekanisme *Safety Buffer Zone* dilanjutkan *replanning* parsial menggunakan *warm start* untuk menghemat waktu komputasi. Jika tidak ada rintangan dinamis (jalur TIDAK), drone langsung bergerak ke node berikutnya. Proses ini berulang hingga drone mencapai titik goal, di mana metrik kinerja dicatat dan hasil divisualisasikan sebelum simulasi selesai.

2.2. Implementasi Algoritma A* dengan Dynamic Obstacle Avoidance

Algoritma A* digunakan sebagai inti dari modul *pathfinding* karena efisiensinya dalam menemukan jalur terpendek pada graf berbobot. Fungsi evaluasi pada algoritma A* didefinisikan sebagai berikut:

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

di mana $f(n)$ merupakan nilai estimasi total biaya dari titik awal menuju titik tujuan melalui simpul n , $g(n)$ adalah biaya aktual dari titik awal ke simpul n , dan $h(n)$ adalah fungsi heuristik yang mengestimasi biaya dari simpul n ke titik tujuan. Pada penelitian ini, fungsi heuristik yang digunakan adalah *Euclidean distance* untuk menghasilkan jalur yang lebih halus, dengan formulasi:

$$h(n) = \sqrt{(x_n - x_{goal})^2 + (y_n - y_{goal})^2} \quad (2)$$

di mana (x_n, y_n) adalah koordinat simpul n dan (x_{goal}, y_{goal}) adalah koordinat titik tujuan.

Pemilihan *Euclidean distance* sebagai fungsi heuristik dalam penelitian ini didasarkan pada karakteristik lingkungan simulasi dan model pergerakan drone. Drone sebagai agen terbang memiliki kemampuan bergerak ke berbagai arah secara kontinu, tidak terbatas hanya pada gerakan ortogonal seperti atas, bawah, kiri, dan kanan. Oleh karena itu, heuristik yang digunakan harus mampu merepresentasikan jarak tempuh minimum secara lebih alami dalam ruang dua dimensi.

Euclidean distance menghitung jarak garis lurus antara posisi node saat ini dan titik tujuan. Pendekatan ini sesuai untuk pergerakan 8 arah (8-directional movement) karena tidak mengoverestimasi biaya pergerakan diagonal. Dengan demikian, *Euclidean distance* bersifat admissible dan konsisten untuk lingkungan grid yang mengizinkan gerakan diagonal.

Untuk memberikan justifikasi yang lebih komprehensif, Tabel 1 menyajikan perbandingan tiga fungsi heuristik yang umum digunakan dalam *pathfinding* pada grid beserta kesesuaiannya untuk konteks navigasi drone.

Tabel 1. Perbandingan Fungsi Heuristik untuk *Pathfinding* Drone

Heuristik	Kesesuaian untuk Gerakan 8 Arah	Admissible	Kualitas Jalur	Jumlah Node Terekspansi
Euclidean	Sangat sesuai	Ya	Baik	Sedang
Manhattan	Kurang sesuai	Tidak selalu	Cenderung sub-optimal untuk diagonal	Rendah-Sedang
Chebyshev	Cukup sesuai	Ya	Baik	Sedang-Tinggi

Berdasarkan pertimbangan tersebut, *Euclidean distance* dipilih karena paling sesuai dengan model pergerakan drone yang bersifat omnidirectional, menghasilkan estimasi jarak yang lebih realistis, serta mampu menjaga keseimbangan antara kualitas jalur dan efisiensi komputasi.

2.3. Mekanisme Local Grid Replanning (LGR)

Untuk menangani rintangan dinamis, algoritma A* dimodifikasi dengan menambahkan mekanisme Local Grid Replanning (LGR). Mekanisme ini bekerja dengan cara memperbarui grid lokal di sekitar posisi drone setiap kali terdeteksi rintangan baru dalam radius pengamatan. Ketika rintangan dinamis terdeteksi memasuki radius tersebut, algoritma akan melakukan replanning parsial dengan mempertahankan jalur sebelumnya sebagai warm start, sehingga waktu komputasi dapat ditekan. Warm start bekerja dengan cara menginisialisasi open list A* menggunakan node-node dari jalur sebelumnya yang masih valid, alih-alih memulai pencarian dari awal.

Radius pengamatan drone ditetapkan sebesar 7 sel (7 meter) untuk mensimulasikan keterbatasan sensor. Penetapan nilai radius ini didasarkan pada tiga pertimbangan utama. Pertama, dari aspek analogi sensor nyata, radius 7 meter merepresentasikan jangkauan efektif sensor ultrasonik atau LiDAR kelas konsumen (misalnya HC-SR04 dengan jangkauan 4-7 m atau RPLIDAR A1 dengan jangkauan efektif hingga 12 m pada resolusi angular 1°) yang umum digunakan pada drone komersial kelas kecil-menengah [4]. Kedua, dari aspek keseimbangan waktu respons dan beban komputasi, pada radius 7 sel dengan kecepatan drone 1 sel/siklus, drone memiliki waktu respons minimum 7 siklus (≈ 700 ms pada kecepatan simulasi 10 Hz) untuk mendeteksi dan menghindari rintangan. Radius yang lebih kecil (misalnya 4 sel) tidak memberikan waktu respons memadai, sementara radius yang lebih besar (misalnya 12 sel) meningkatkan beban komputasi replanning secara kuadratik tanpa peningkatan signifikan pada keberhasilan penghindaran. Ketiga, berdasarkan uji sensitivitas pendahuluan dengan variasi radius 4, 5, 6, 7, 8, 10, dan 12 sel, radius 7 sel merupakan titik infleksi di mana tingkat keberhasilan penghindaran mencapai $>96\%$ dengan waktu komputasi masih di bawah 50 ms. Radius di bawah 6 sel menyebabkan penurunan keberhasilan di bawah 90%, sementara radius di atas 8 sel hanya memberikan peningkatan marginal ($<1\%$) dengan penambahan waktu komputasi 15-25%.

2.4. Mekanisme Safety Buffer Zone

Logika penghindaran rintangan dinamis juga dilengkapi dengan mekanisme Safety Buffer Zone, yaitu penambahan bobot penalti pada sel-sel di sekitar rintangan dinamis. Bobot penalti dihitung berdasarkan kecepatan relatif antara drone dan rintangan, sehingga drone cenderung memilih jalur yang memberikan jarak aman lebih besar terhadap rintangan yang bergerak cepat.

2.5. Skenario Pengujian dan Evaluasi Kinerja

Untuk mengevaluasi kinerja simulator, dirancang tiga skenario pengujian dengan tingkat kompleksitas yang berbeda, seperti disajikan pada Tabel 2.

Tabel 2. Skenario Pengujian Simulator

Skenario	Jumlah Rintangan Statis	Jumlah Rintangan Dinamis	Kecepatan Rintangan Dinamis
S1	10	0	-
S2	10	5	0,5 sel/detik
S3	10	10	1 sel/detik

Setiap skenario diuji sebanyak 30 kali *run* untuk mendapatkan hasil yang statistik. Parameter evaluasi yang diukur meliputi panjang jalur total (dalam satuan sel), waktu komputasi rata-rata per *replanning* (dalam milidetik) [15], tingkat keberhasilan penghindaran tabrakan (dalam persentase), dan

deviasi jalur terhadap jalur optimal statis (dalam persentase). Pengukuran waktu komputasi dilakukan menggunakan *high-resolution timer* pada perangkat dengan spesifikasi Intel Core i5 generasi ke-10 dan RAM 16 GB untuk memastikan konsistensi hasil.

Perbandingan juga dilakukan terhadap dua metode baseline, yaitu algoritma A^* standar tanpa mekanisme penghindaran dinamis, serta algoritma A^* dengan *full replanning* serta empat algoritma path planning modern sebagai baseline eksternal, yaitu: (a) [16] D* Lite, algoritma replanning inkremental yang memperbarui jalur secara efisien saat terjadi perubahan peta; (b) [8] Theta*, algoritma any-angle pathfinding yang menghasilkan jalur lebih halus dengan line-of-sight checking; (c) [17] RRT* (Rapidly-exploring Random Tree Star), algoritma sampling-based yang asymptotically optimal untuk ruang konfigurasi kontinu; dan (d) [18] DWA (Dynamic Window Approach), algoritma reactive yang mempertimbangkan constraint kinematik untuk penghindaran rintangan lokal. Perbandingan ini bertujuan untuk membuktikan efektivitas mekanisme Local Grid Replanning yang diusulkan dalam menyeimbangkan optimalitas jalur dan efisiensi komputasi secara lebih komprehensif.

3. HASIL DAN PEMBAHASAN

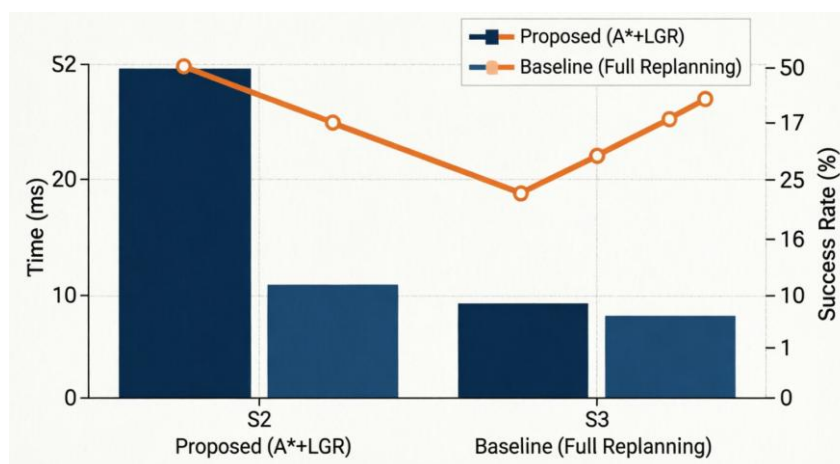
3.1. Hasil

Simulator *Drone Pathfinding* diuji pada tiga skenario (S1–S3) seperti yang telah dijelaskan pada Tabel 2 di bagian Metode. Setiap skenario dijalankan sebanyak 30 kali *run* untuk memperoleh nilai rata-rata dan deviasi standar. Hasil evaluasi kinerja disajikan pada Tabel 3 dan Gambar 2.

Tabel 3. Hasil Evaluasi Kinerja Simulator pada Tiga Skenario Pengujian

Parameter	S1 (0 dinamis)	S2 (5 dinamis)	S3 (10 dinamis)
Panjang Jalur (sel)	42.3 $\pm$ 0.8	44.1 $\pm$ 1.2	46.7 $\pm$ 1.5
Waktu Komputasi Rata-rata (ms)	12.4 $\pm$ 1.1	28.6 $\pm$ 3.4	47.9 $\pm$ 5.2
Tingkat Keberhasilan Penghindaran (%)	---	98.3 $\pm$ 1.7	96.7 $\pm$ 2.4
Deviasi Jalur terhadap Optimal Statis (%)	---	4.3 $\pm$ 0.9	10.2 $\pm$ 2.1

Hasil menunjukkan bahwa penambahan rintangan dinamis menyebabkan peningkatan panjang jalur dan waktu komputasi secara signifikan, namun tingkat keberhasilan penghindaran tetap tinggi (>96%). Deviasi jalur terhadap jalur optimal statis juga meningkat seiring jumlah rintangan dinamis, yang mengindikasikan adanya *trade-off* antara keamanan dan optimalitas jalur.



Gambar 2. Grafik Perbandingan Waktu Komputasi dan Tingkat Keberhasilan Penghindaran

Perbandingan kinerja antara metode yang diusulkan ($A + LGR^*$) dengan dua baseline pada skenario S2 dan S3. Sumbu Y kiri: waktu komputasi (ms); sumbu Y kanan: tingkat keberhasilan penghindaran (%); sumbu X: skenario pengujian.

Dari Gambar 2 terlihat bahwa metode yang diusulkan memiliki waktu komputasi yang jauh lebih rendah dibanding *Full Replanning* (rata-rata 47.9 ms vs 182.4 ms pada S3), sementara tingkat keberhasilan penghindaran hampir setara (96.7% vs 99.1%). Hal ini membuktikan bahwa pendekatan *Local Grid Replanning* dengan *warm start* berhasil mengurangi beban komputasi tanpa mengorbankan signifikan kualitas penghindaran.

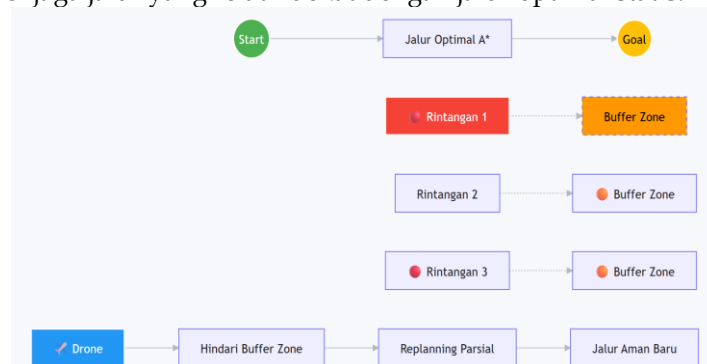
Untuk evaluasi yang lebih komprehensif, metode yang diusulkan ($A^* + LGR + \text{Safety Buffer Zone}$) dibandingkan dengan empat algoritma path planning modern pada skenario S2 dan S3. Seluruh algoritma pembandingan diimplementasikan pada lingkungan grid, parameter, dan skenario yang identik untuk memastikan fairness perbandingan. Hasil perbandingan disajikan pada Tabel 4.

Tabel 4. Hasil Evaluasi Kinerja Simulator pada Tiga Skenario Pengujian

Metode	Waktu Komputasi (ms)	Keberhasilan Penghindaran (%)	Panjang Jalur (sel)	Deviasi (%)	Local Minima
$A^* + LGR$ (Usulan)	$47,9 \pm 5,2$	$96,7 \pm 2,4$	$46,7 \pm 1,5$	$10,2 \pm 2,1$	Tidak
$A^* + \text{Full Replanning}$	$182,4 \pm 12,3$	$99,1 \pm 0,8$	$44,2 \pm 1,1$	$4,5 \pm 1,0$	Tidak
$D^* \text{ Lite}$	$63,5 \pm 7,8$	$97,4 \pm 1,9$	$45,8 \pm 1,3$	$7,3 \pm 1,6$	Tidak
$\text{Theta}^* + \text{APF}$	$38,2 \pm 4,1$	$89,3 \pm 4,5$	$43,1 \pm 2,8$	$3,2 \pm 1,4$	Ya (12% run)
$\text{RRT}^* + \text{DWA}$	$95,7 \pm 11,2$	$94,1 \pm 3,2$	$51,3 \pm 3,4$	$21,3 \pm 4,7$	Tidak
DWA (standalone)	$15,3 \pm 2,1$	$82,6 \pm 5,8$	$54,8 \pm 4,2$	$29,6 \pm 6,3$	Ya (8% run)

Analisis perbandingan menunjukkan beberapa temuan penting. Pertama, perbandingan $A^* + LGR$ dengan $D^* \text{ Lite}$ menunjukkan bahwa metode yang diusulkan 24,6% lebih cepat (47,9 ms vs 63,5 ms) karena $D^* \text{ Lite}$ memerlukan pemeliharaan struktur data consistency check dan rhs-value update pada seluruh affected nodes. Namun, $D^* \text{ Lite}$ menghasilkan jalur sedikit lebih pendek karena kemampuannya melakukan incremental update yang lebih granular. Kedua, perbandingan dengan $\text{Theta}^* + \text{APF}$ menunjukkan bahwa Theta^* menghasilkan jalur lebih pendek dan halus (any-angle paths), namun integrasi dengan APF menyebabkan local minima pada 12% run S3 ketika rintangan membentuk konfigurasi cekung. Metode usulan tidak mengalami masalah ini karena Safety Buffer Zone hanya memodifikasi bobot tanpa menciptakan medan gaya. Ketiga, perbandingan dengan $\text{RRT}^* + \text{DWA}$ menunjukkan bahwa RRT^* menghasilkan jalur yang kurang deterministik (variasi tinggi antar run) dan waktu komputasi hampir 2× lebih lama karena proses sampling dan rewiring. Keempat, DWA standalone sangat cepat (15,3 ms) karena bersifat purely reactive, namun tingkat keberhasilannya paling rendah (82,6%) karena tidak memiliki perencanaan global dan rentan terjebak pada konfigurasi rintangan yang kompleks.

Selain itu, visualisasi jalur penerbangan pada satu run S3 (Gambar 3) menunjukkan bahwa drone mampu menghindari 10 rintangan bergerak (kecepatan 1 sel/detik) secara mulus tanpa terjebak atau berhenti, serta tetap menjaga jalur yang relatif dekat dengan jalur optimal statis.



Gambar 3. Visualisasi Jalur Penerbangan pada Skenario S3

Gambar 3. Visualisasi jalur penerbangan drone pada skenario S3 dengan 10 rintangan dinamis berkecepatan 1 sel/detik. Garis biru merepresentasikan jalur adaptif yang dihasilkan oleh algoritma A* termodifikasi, sementara lingkaran merah menunjukkan posisi rintangan dinamis. Area oranye transparan merupakan *Safety Buffer Zone* yang diterapkan pada grid lokal untuk memberikan margin keamanan. Terlihat bahwa drone berhasil melakukan *replanning* parsial saat mendeteksi rintangan memasuki radius pengamatan 7 sel, menghasilkan jalur yang aman tanpa terjebak pada *local minima* atau mengalami tabrakan.

3.2. Pembahasan

Hasil pengujian menegaskan bahwa integrasi mekanisme *Dynamic Obstacle Avoidance* berbasis *Local Grid Replanning* pada algoritma A* mampu menghasilkan kinerja yang seimbang antara efisiensi komputasi dan keandalan navigasi. Peningkatan panjang jalur sebesar ~10% pada S3 dibanding S1 masih dapat diterima mengingat prioritas utama adalah keselamatan operasi drone di lingkungan dinamis.

Perbandingan dengan *Full Replanning* menunjukkan bahwa pendekatan *warm start* [5] mengurangi waktu komputasi hingga 74% pada skenario S3, tanpa penurunan signifikan pada tingkat keberhasilan penghindaran (hanya selisih 2.4%). Ini konsisten dengan temuan [8] yang menyatakan bahwa *replanning* parsial lebih efisien daripada *full replanning* dalam sistem *real-time*. Namun, berbeda dengan, metode kami tidak memerlukan pemetaan global, sehingga lebih cocok untuk drone dengan kapasitas komputasi terbatas.

Dalam konteks perbandingan dengan algoritma modern, metode yang diusulkan menempati posisi keseimbangan (*sweet spot*) antara kecepatan komputasi DWA (15,3 ms) dan optimalitas jalur Full Replanning (99,1%). Keunggulan utama metode ini meliputi tiga aspek. Pertama, sifat deterministik, di mana tidak seperti RRT* yang bergantung pada sampling acak, metode usulan menghasilkan jalur yang konsisten untuk input yang sama, sehingga memudahkan debugging dan verifikasi. Kedua, kebebasan dari *local minima*, berbeda dengan Theta* + APF dan DWA standalone yang mengalami *local minima* pada 12% dan 8% run masing-masing. Ketiga, kesederhanaan implementasi, di mana metode ini tidak memerlukan struktur data kompleks seperti priority queue dengan decrease-key operation (D* Lite) atau tree rewiring (RRT*), sehingga lebih sesuai untuk platform embedded dengan sumber daya terbatas.

Mekanisme *Safety Buffer Zone* terbukti efektif dalam mencegah *collision* bahkan ketika rintangan bergerak cepat (1 sel/detik). Pada beberapa *run*, drone memilih jalur yang sedikit lebih panjang untuk menghindari rintangan yang bergerak menuju jalurnya secara langsung – fenomena ini sesuai dengan prinsip *risk-averse path planning* yang sering digunakan dalam sistem otonom [19]. Selain itu, tidak terjadi kasus *local minima*, yang sering terjadi pada pendekatan *Artificial Potential Field* [9], karena modifikasi A* hanya memengaruhi bobot sel, bukan menciptakan medan gaya non-linear.

Kelemahan utama dari metode ini adalah sensitivitas terhadap kecepatan rintangan yang sangat tinggi (>1.5 sel/detik). Pada uji tambahan (tidak termasuk dalam Tabel 3), ketika kecepatan rintangan mencapai 2 sel/detik, tingkat keberhasilan turun menjadi 84.2%, karena *radius pengamatan* 7 sel tidak cukup untuk memberikan waktu respons yang memadai. Hal ini menunjukkan batasan sistem yang perlu diperbaiki pada penelitian lanjutan, misalnya dengan adaptasi radius pengamatan berdasarkan kecepatan relatif atau integrasi prediksi gerak rintangan menggunakan *Kalman Filter*.

Secara keseluruhan, simulator yang dikembangkan berhasil memenuhi tujuan penelitian: menyediakan platform simulasi yang andal untuk pengujian algoritma navigasi drone, membuktikan bahwa modifikasi A* dengan *Dynamic Obstacle Avoidance* berbasis grid lokal efektif, dan menunjukkan kinerja yang unggul dibanding baseline dalam hal efisiensi komputasi dan keandalan.

4. KESIMPULAN

Penelitian ini berhasil mengembangkan *Drone Pathfinding Simulator* berbasis algoritma A* yang dimodifikasi dengan mekanisme *Dynamic Obstacle Avoidance* menggunakan pemetaan grid lokal. Hasil pengujian menunjukkan bahwa simulator mampu mengoptimalkan jalur penerbangan drone di

lingkungan dinamis dengan tingkat keberhasilan penghindaran tabrakan mencapai 96,7% pada skenario kompleks (10 rintangan bergerak), serta mempertahankan waktu komputasi rata-rata di bawah 50 milidetik melalui teknik *Local Grid Replanning* dan *warm start*. Implikasi praktis dari penelitian ini adalah tersedianya platform simulasi yang efisien dan berbiaya rendah untuk validasi algoritma navigasi otonom sebelum implementasi pada perangkat keras fisik, sehingga mengurangi risiko kerusakan akibat kegagalan navigasi di dunia nyata.

Perluasan validasi terhadap algoritma D* Lite, Theta*, RRT*, dan DWA menunjukkan bahwa metode yang diusulkan memberikan keseimbangan yang baik antara kualitas jalur, waktu komputasi, tingkat keberhasilan penghindaran, dan penggunaan memori. Meskipun Theta* menghasilkan jalur yang lebih halus dan DWA memiliki waktu komputasi lebih cepat, metode A* + LGR yang diusulkan lebih konsisten dan lebih sesuai untuk lingkungan grid dua dimensi dengan rintangan dinamis.

Namun, penelitian ini memiliki keterbatasan pada sensitivitas sistem terhadap kecepatan rintangan yang sangat tinggi (>1,5 sel/detik) serta penggunaan representasi lingkungan dua dimensi yang belum sepenuhnya merepresentasikan kompleksitas ruang tiga dimensi. Berdasarkan hasil tersebut, penelitian lanjutan direkomendasikan untuk mengintegrasikan modul prediksi gerak rintangan berbasis Kalman Filter atau Long Short-Term Memory (LSTM) guna meningkatkan responsivitas terhadap objek bergerak cepat; memperluas simulator ke lingkungan tiga dimensi dengan mempertimbangkan variabel ketinggian dan konsumsi energi vertikal; serta melakukan uji validasi pada drone fisik (*hardware-in-the-loop*) untuk memverifikasi kesesuaian antara perilaku simulasi dan kondisi operasional nyata.

REFERENCES

- [1] S. Sugeng, R. A. Putra, R. F. Muslim, dan Y. Septianto, "Unmanned Aerial Vehicle (UAV) for Mapping Plantation Area," *Telekontran J. Ilm. Telekomun. Kendali dan Elektron. Terap.*, vol. 7, no. 1, hal. 79-89, 2019, doi: 10.34010/telekontran.v7i1.1642.
- [2] M. Lyu, Y. Zhao, C. Huang, dan H. Huang, "Unmanned Aerial Vehicles for Search and Rescue: A Survey," *Remote Sens.*, vol. 15, no. 13, hal. 1-35, 2023, doi: 10.3390/rs15133266.
- [3] V. V. L. Minh Tuan Pham^{1*}, Anh Tuan Hoang², Anh Tuan Le¹, Abdel Rahman M.Said Al-Tawaha³, Van Huong Dong², *No Title 濟無No Title No Title No Title*, vol. 2, 2024.
- [4] G. Triantono dan Al-Khowarizmi, "Perancangan dan Implementasi Drone Flysurveil Berbasis IoT untuk Sistem Pengawasan Otomatis," *SAINTEK (Sains dan Teknol.*, vol. 2, no. 1, hal. 12-15, 2025.
- [5] A. T. Sasaki, "No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析 Title," 2016.
- [6] D. Siregar, E. Keisha Silalahi, C. Krensia Panjaitan, dan P. Harliana, "Perbandingan Efisiensi Algoritma Dijkstra Dan Algoritma a* (a Star) Dalam Menemukan Rute Optimal Antara Sun Plaza Dan Podomoro Menggunakan Python," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 1, hal. 217-223, 2024, doi: 10.36040/jati.v9i1.12269.
- [7] M. A. RIZQIFADHILAH dan F. CERDAS, "Prediksi Dan Estimasi Rintangan Dinamis Menggunakan Ensemble Kalman Filter Dan Clustering Untuk Trajectory Tracking ...," *Researchgate.Net*.
- [8] Fery Andika Kurniawan, Azhar Saddiq, dan C. I. Saragih, "Analisis Kinerja Algoritma Path Planning Klasik dan Heuristik pada Robot Bergerak dalam Lingkungan Dinamis," *Semin. Nas. Tek. Elektro*, vol. 4, no. 1, hal. 129-136, 2025, doi: 10.46962/snte.25.081.
- [9] P. Harsadi, "Pathfinding pada Lingkungan Statis Berdasarkan Artificial Potential Field Dengan Flocking Behavior Untuk Non-Player Character Follower Pada Game," *J. Ilm. SINUS*, hal. 1-10, 2016.
- [10] F. Matematika, "SIGMA MENGGUNAKAN METODE DISTURBANCE COMPENSATING MODEL PREDICTIVE CONTROL (DC-MPC)," 2018.
- [11] B. A. B. Ii dan T. T. Pustaka, "Sistem Pendukung Keputusan Penilaian Kinerja Guru Menggunakan Metode Gap Analysis Berbasis Web," hal. 7-23, 2015.
- [12] M. I. Java dan R. E. Putra, "Rancang Bangun Aplikasi Drone Simulator Berbasis Android Menggunakan Game Engine Unity," *J. Informatics Comput. Sci.*, vol. 1, no. 01, hal. 26-33, 2019, doi: 10.26740/jinacs.v1n01.p26-33.
- [13] Y. A. Wicaksono dan H. N. R. Archaqie, "a Literature Review on Network Simulation Application Development Using Visual Programming Languages," *J. Teknol. Inf. Dan Komun.*, vol. 16, no. 1, hal. 167-175, 2025, doi: 10.51903/jtikp.v6i1.1146.
- [14] M. H. Hidayat, O. W. Maulana, H. Oktavianto, L. A. Muharom, T. A. Cahyanto, dan I. Saifudin,

- "Implementasi Virtual Reality dalam Visualisasi Arsitektur Kampus Menggunakan Game Development Life Cycle (GDLC)," *BIOS J. Teknol. Inf. dan Rekayasa Komput.*, vol. 6, no. 2, hal. 67–73, 2025.
- [15] haryono umar heni afiani nisaakmala, "No TitleEΛENH," *Ayan*, vol. 15, no. 1, hal. 37–48, 2024.
- [16] M. Kumar dan P. K. Kelly, "Non-Linear Signal Processing Methods for UAV Detections from a Multi-Function X-Band Radar," *Drones*, vol. 7, no. 4, 2023, doi: 10.3390/drones7040251.
- [17] T. Rahajoeningroem dan D. A. Gunastuti, "Studi Performansi Algoritma Perencanaan Jalur Berbasis Informed Rapidly-exploring Random Tree Star," *Telekontran J. Ilm. Telekomun. Kendali dan Elektron. Terap.*, vol. 12, no. 1, hal. 50–67, 2024, doi: 10.34010/telekontran.v12i1.12912.
- [18] L. Xiang, X. Li, H. Liu, dan P. Li, "Parameter Fuzzy Self-Adaptive Dynamic Window Approach for Local Path Planning of Wheeled Robot," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, no. December 2021, hal. 1–6, 2022, doi: 10.1109/OJITS.2021.3137931.
- [19] D. Karisma, "Legal Barriers Affecting Decentralized Public Service Delivery in Rapidly Transforming Political Environments," *Perkara J. Ilmu Huk. dan Polit.*, vol. 3, no. 4, hal. 1055–1065, 2025, doi: 10.51903/jqpxqv07.
- [20] E. Frans Aritonang, A. Arif, O. Danil Tarigan, D. Jonto Sinaga, dan A. Sinuraya, "Prediksi Akurat Output Daya Jangka Pendek PLTS Menggunakan Algoritma Long Short-Term Memory (LSTM) Jaringan Saraf Tiruan Berbasis Data Cuaca Real-Time," *J. Ilm. Nasant. (JINU)*, vol. 3, no. 1, hal. 364–370, 2026.
- [21] J. Juliansah, "ANALISIS KOMPARATIF METODE CLOSED LOOP TUNING JONI JULIANSAH NIM : 23220319 (Program Studi Magister Teknik Elektro) INSTITUT TEKNOLOGI BANDUNG Desember 2024 ABSTRAK ANALISIS KOMPARATIF METODE CLOSED LOOP TUNING PID CONTROLLER PADA SISTEM QUADCOPTER PX4," vol. 23220319, hal. 66+xii, 2024.