ANALISIS KINERJA ALGORITMA A-STAR DALAM PENENTUAN LINTASAN OPTIMAL AMR PADA TATA LETAK GUDANG DENGAN VARIASI DENSITAS HAMBATAN

Abdul Rahman[1], Firmansyah[2], Marlaily Idris[3]
[1]Program Studi Teknik Mekatronika, Fakultas Teknik, Universitas Lancang Kuning
[2]Program Studi Manajemen, Fakultas Ekonomi dan Bisnis, Universitas Lancang Kuning
[3]Program Studi Teknik Sipil, Fakultas Teknik, Universitas Lancang Kuning
Jl. Yos Sudarso km. 8 Rumbai, Pekanbaru, Telp. (0761) 52324
Email: abdulrahman@unilak.ac.id, firmansyah@unilak.ac.id, marlailyidris@unilak.ac.id

ABSTRAK

Keberhasilan navigasi *Autonomous Mobile Robot* (AMR) pada gudang pintar bergantung pada kemampuan algoritma pencarian jalur menghadapi kerapatan rak penyimpanan. Penelitian ini menganalisis kinerja algoritma A* berheuristik *Manhattan Distance* pada grid 10×10 dengan titik awal (0,0) dan titik tujuan (9,9). Analisis memuat tiga studi kasus densitas hambatan (16%, 20%, dan 35%) yang diperluas dengan eksperimen *Monte Carlo* (1.000 grid acak untuk setiap densitas 5%–50%, total 46.000 eksekusi) dan analisis konektivitas *breadth-first search*. Pada studi kasus, A* menemukan lintasan optimal 18 langkah pada densitas 16% dan 20% (61 dan 52 node dieksplorasi), tetapi gagal pada 35% (*path not found*, 48 node) karena dinding hambatan memutus konektivitas titik awal dan tujuan. Eksperimen *Monte Carlo* menunjukkan bahwa ambang kegagalan berupa zona transisi: *success rate* turun di bawah 95% pada densitas sekitar 13%, mencapai 50% pada sekitar 30,5%, dan di bawah 5% pada sekitar 43%. Panjang lintasan rata-rata hanya naik dari 18,00 menjadi 18,65 langkah pada densitas 35%, sehingga densitas lebih memengaruhi ada-tidaknya lintasan daripada panjangnya. Hasil dirangkum menjadi zona densitas indikatif dan protokol verifikasi tata letak untuk model yang diuji.

Kata Kunci: Algoritma A*, *Autonomous Mobile Robot*, densitas hambatan, *Monte Carlo*, tata letak gudang

ABSTRACT

The success of Autonomous Mobile Robot (AMR) navigation in smart warehouses depends on how well a pathfinding algorithm copes with the density of storage racks. This study analyzes the performance of the A* algorithm with the Manhattan distance heuristic on a 10×10 grid with the start at (0,0) and the goal at (9,9). Three case studies at obstacle densities of 16%, 20%, and 35% are extended with a Monte Carlo experiment (1,000 random grids for each density from 5% to 50%, 46,000 runs in total) and a breadth-first-search connectivity analysis. In the case studies, A* found an optimal 18-step path at 16% and 20% density (61 and 52 nodes explored) but failed at 35% (path not found, 48 nodes) because a wall of obstacles severed the connectivity between start and goal. The Monte Carlo experiment shows that the failure threshold is a transition zone rather than a single value: the success rate falls below 95% at about 13% density, reaches 50% at about 30.5%, and drops below 5% at about 43%. The mean path length rises only from 18.00 to 18.65 steps at 35% density, so density affects whether a path exists more than how long it is. The results are summarized as indicative density zones and a layout verification protocol for the tested model.

Keywords: A* algorithm, Autonomous Mobile Robot, obstacle density, Monte Carlo, warehouse layout

1. PENDAHULUAN

Efisiensi operasional gudang menjadi faktor penentu daya saing dalam logistik modern, dan *order picking* dikenal sebagai aktivitas yang paling padat tenaga kerja dan paling mahal dalam operasional gudang [1]. Kondisi ini mendorong penggunaan *Autonomous Mobile Robot* (AMR) untuk pemindahan barang. Tinjauan pustaka menunjukkan bahwa keberhasilan AMR pada sistem intralogistik sangat bergantung pada kemampuan sistem navigasi merencanakan lintasan yang efisien di lingkungan kerja yang padat [3], [7].

Algoritma A-Star (A*) dirumuskan oleh Hart dkk. [5] dan menjamin lintasan berbiaya minimum selama fungsi heuristiknya *admissible*. Algoritma ini banyak dikaji pada peta grid [12], [13], dan pengembangannya mencakup pemangkasan simpul seperti *Jump Point Search* [4] serta perbandingan dengan metode lain pada peta grid [2], [6]. Kinerja A* di lapangan, meskipun demikian, dipengaruhi oleh karakteristik lingkungan. Densitas hambatan

p-ISSN: 1858-4217, e-ISSN: 2622-710X, DOI: https://doi.org/10.31849/e4tx4312

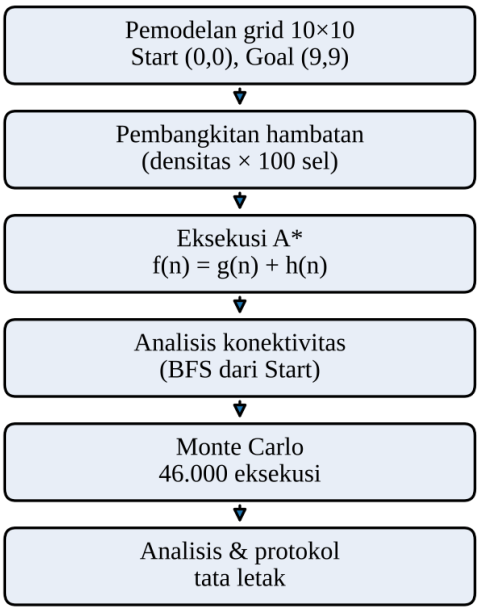
lazim dipakai sebagai parameter untuk mengevaluasi keberhasilan pencarian jalur pada peta grid [13]. Pada densitas tinggi, hambatan yang saling berdekatan dapat menutup akses menuju tujuan sehingga lintasan tidak ditemukan (*path not found*) atau pencarian membebani komputasi.

Kegagalan pembentukan lintasan satu robot berbeda dari *deadlock* antar-robot. *Deadlock* merupakan masalah multi-robot yang ditangani melalui kontrol zona dan perencanaan multi-agen [8], [11], dan tidak diteliti di sini. Penelitian ini membatasi diri pada pertanyaan yang lebih dasar: pada kerapatan hambatan berapa A* masih menemukan lintasan, dan bagaimana kerapatan itu memengaruhi panjang lintasan dan beban komputasi.

Satu konfigurasi hambatan untuk setiap densitas belum menunjukkan apakah hasilnya khas atau kebetulan, padahal evaluasi algoritma pada grid umumnya memakai banyak instansi uji [12]. Penelitian ini karena itu memadukan (i) tiga studi kasus densitas 16%, 20%, dan 35% yang diuraikan langkah demi langkah, (ii) eksperimen *Monte Carlo* pada densitas 5%–50% untuk menunjukkan perilaku umum dan batas kegagalan, serta (iii) analisis konektivitas untuk menjelaskan penyebab kegagalan. Hasilnya dirangkum menjadi zona densitas indikatif dan protokol verifikasi tata letak gudang.

2. METODE PENELITIAN

Penelitian menggunakan pendekatan kuantitatif berbasis simulasi pada ruang diskret. Alur penelitian diperlihatkan pada Gambar 1.



Gambar 1. Alur proses penelitian

Pemodelan Grid

Denah lantai gudang dimodelkan sebagai grid dua dimensi 10×10 (100 sel) dengan S(x,y) = 0 untuk sel bebas dan S(x,y) = 1 untuk hambatan statis (rak). Titik awal (*Start*) berada di (0,0) sebagai area *receiving* dan titik tujuan (*Goal*) di (9,9) sebagai area *picking/shipping*. Koordinat ditulis (x,y) dengan x sebagai kolom dan y sebagai baris. Pergerakan AMR dibatasi empat arah ortogonal (atas, bawah, kiri, kanan).

Pembangkitan Hambatan

Jumlah hambatan sama dengan densitas dikalikan 100 sel, dengan *Start* dan *Goal* dikecualikan. Tiga studi kasus (16%, 20%, 35%) memakai konfigurasi ilustratif dengan koordinat hambatan tetap agar dapat direproduksi. Generalisasi dilakukan melalui eksperimen *Monte Carlo* dengan hambatan acak seragam tanpa pengembalian pada setiap grid.

Formulasi Algoritma A*

A* mengevaluasi setiap simpul n dengan fungsi biaya total

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

dengan $g(n)$ biaya riil dari *Start* (setiap langkah bernilai 1) dan h(n) estimasi biaya sisa. Karena gerak dibatasi empat arah, digunakan *Manhattan Distance*

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

p-ISSN: 1858-4217, e-ISSN: 2622-710X, DOI: https://doi.org/10.31849/e4tx4312

yang *admissible* pada grid empat arah sehingga lintasan yang ditemukan optimal [5]. Proses pencarian diringkas pada Algoritma 1.

```
ALGORITMA 1. A* (Manhattan, 4 arah)
1 OPEN ← min-heap {Start}; CLOSED ← ∅
2 selama OPEN tidak kosong:
3   n ← node f terkecil di OPEN; n → CLOSED
4   jika n = Goal: kembalikan lintasan, |CLOSED|
5   untuk tiap tetangga m dari n (4 arah):
6     lewati jika di luar grid/hambatan/CLOSED
7     g(m) ← g(n)+1; h(m) ← Manhattan(m,Goal)
8     f(m) ← g(m)+h(m)
9     lewati jika di OPEN ada posisi sama
       dengan g ≤ g(m)
10    masukkan m ke OPEN
11 kembalikan Path Not Found, |CLOSED|
```

Implementasi memakai Python 3.10 pada Google Colab. *Open list* berupa *min-heap* (heapq), *closed set* berupa himpunan posisi, dan waktu diukur dengan time.perf_counter(). Perbandingan node pada *heap* hanya berdasarkan f(n); bila f(n) sama, urutan mengikuti perilaku internal heapq tanpa kriteria tambahan h(n). Node *Start* diinisialisasi dengan g = h = f = 0 karena selalu diambil pertama sehingga tidak memengaruhi hasil. Kinerja diukur dengan status pencarian, panjang lintasan (langkah), jumlah node dieksplorasi (ukuran *closed set*), dan waktu komputasi.

Analisis Konektivitas

Penelusuran *breadth-first search* (BFS) dari *Start* menentukan wilayah sel yang terjangkau. Jika *Goal* berada di luar wilayah itu, status *Path Not Found* dijelaskan secara topologi sebagai pemutusan konektivitas graf (*grid isolation*).

Eksperimen Monte Carlo

Untuk setiap densitas 5%–50% (langkah 1%) dibangkitkan 1.000 grid acak (*seed* 2026) lalu dijalankan A* yang identik dengan studi kasus, sehingga total 46.000 eksekusi. *Success rate* adalah proporsi grid dengan lintasan ditemukan, SR = 100% × k/N, dengan k jumlah grid berhasil dan N = 1.000. Interval kepercayaan 95% dihitung dengan metode Wilson [14]. Jumlah node dan panjang lintasan dilaporkan sebagai rata-rata ± simpangan baku (SD) dari grid yang berhasil. Pada 5.000 grid tambahan untuk densitas 10%, 20%, 30%, 35%, dan 40%, kegagalan diklasifikasikan menjadi *Start* atau *Goal* terkurung langsung (kedua tetangganya hambatan) dan pemutusan oleh dinding hambatan.

3. HASIL DAN PEMBAHASAN

*Penelusuran Algoritma A**

Tabel 1 memperlihatkan delapan iterasi awal A* pada grid densitas 20%. Pada setiap iterasi, node ber-f terkecil diambil dari *open list* dan dimasukkan ke *closed set*, lalu tetangga yang valid (di dalam grid, bukan hambatan, belum di *closed set*) ditambahkan ke *open list*.

Tabel 1. Penelusuran delapan iterasi awal A* pada grid densitas 20%

Iterasi	Node diambil (x,y)	g	h	f	Ukuran *open list*
1	(0,0) Start	0	0	0	2
2	(0,1)	1	17	18	3
3	(1,0)	1	17	18	3
4	(0,2)	2	16	18	4
5	(1,1)	2	16	18	3
6	(0,3)	3	15	18	4
7	(2,0)	2	16	18	4
8	(0,4)	4	14	18	5

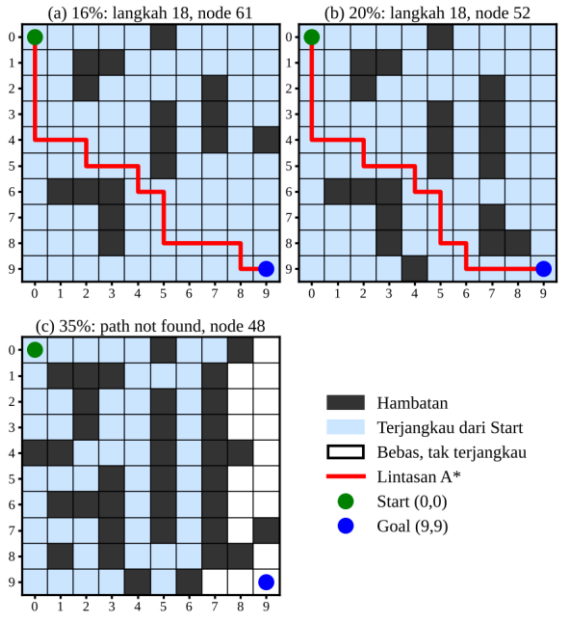
Setelah node *Start*, selama lintasan belum terhalang setiap node yang diperluas memiliki f(n) = 18, yaitu sama dengan jarak *Manhattan Start–Goal*. Banyak node berada pada nilai f yang sama (*tie*), sehingga A* memperluas sejumlah node ber-f = 18 sebelum mencapai *Goal*. Inilah sebabnya node dieksplorasi (52–61) jauh lebih banyak daripada panjang lintasan akhir (18 langkah). Urutan penanganan *tie* memengaruhi jumlah node yang diperluas, tetapi tidak memengaruhi keoptimalan lintasan.

Hasil Tiga Studi Kasus

Tabel 2. Kinerja A* pada tiga studi kasus

Parameter	16%	20%	35%
Jumlah hambatan (sel)	16	20	35
Sel bebas terjangkau dari *Start*	84 dari 84	80 dari 80	48 dari 65
Status pencarian	Path Found	Path Found	Path Not Found
Panjang lintasan (langkah)	18	18	–
Node dieksplorasi	61	52	48
Waktu komputasi (ms)	0,4087	0,6203	0,4527

p-ISSN: 1858-4217, e-ISSN: 2622-710X, DOI: https://doi.org/10.31849/e4tx4312



Gambar 2. Hasil simulasi pada densitas 16% (a), 20% (b), dan 35% (c)

Pada densitas 16% dan 20%, seluruh sel bebas terhubung dengan *Start* dan A* menemukan lintasan 18 langkah, sama dengan jarak *Manhattan* sehingga tidak diperlukan jalur memutar (Gambar 2a, 2b). Node yang dieksplorasi (61 dan 52) berasal dari perluasan pada nilai f yang sama, bukan dari penjelajahan seluruh ruang.

Pada densitas 35%, hanya 48 dari 65 sel bebas yang terjangkau dari *Start*; 17 sel bebas, termasuk *Goal*, terisolasi (Gambar 2c). A* berhenti setelah *closed set* berisi 48 node, tepat sama dengan ukuran wilayah terjangkau, sehingga status *Path Not Found* sahih secara topologi. Pemutus konektivitas adalah dinding Kolom 7 (y = 1–8) yang ditutup di ujung atas oleh hambatan (8,0) dan di ujung bawah oleh hambatan (6,9). Kolom 5 tidak membentuk dinding penuh karena sel (5,1) bebas. Waktu komputasi berasal dari satu kali eksekusi dan hanya menggambarkan orde sub-milidetik, bukan dasar perbandingan antar-studi kasus.

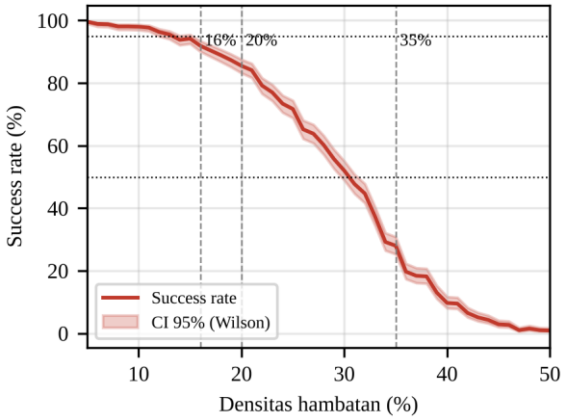
Hasil Eksperimen Monte Carlo

Tabel 3 dan Gambar 3 meringkas hasil *Monte Carlo*. Dengan interpolasi linear antar-titik, *success rate* turun di bawah 95% pada densitas sekitar 13,3%, mencapai 50% pada sekitar 30,5%, dan turun di bawah 5% pada sekitar 43,1%.

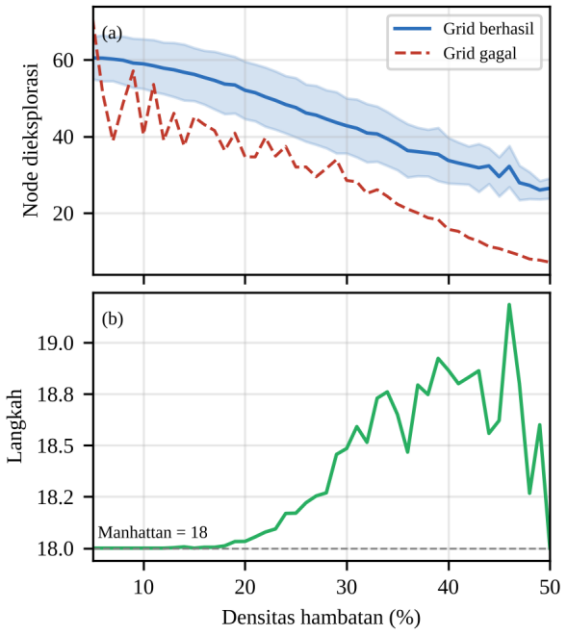
Tabel 3. Hasil *Monte Carlo* pada densitas terpilih (1.000 grid per densitas)

Densitas (%)	SR (%)	CI 95% (%)	Node (rata-rata ± SD)	Panjang rata-rata	*Detour* (%)
10	98,0	96,9–98,7	58,9 ± 6,5	18,00	0,0
16	91,9	90,0–93,4	55,4 ± 7,0	18,00	0,2
20	85,5	83,2–87,5	52,0 ± 7,5	18,03	1,5
25	71,7	68,8–74,4	47,5 ± 7,8	18,17	6,0
30	52,0	48,9–55,1	42,8 ± 7,2	18,48	16,0
35	28,0	25,3–30,9	38,0 ± 6,8	18,65	20,4
40	9,7	8,0–11,7	33,7 ± 6,0	18,87	25,8
45	2,9	2,0–4,1	29,5 ± 5,1	18,62	24,1
50	0,9	0,5–1,7	26,4 ± 2,7	18,00	0,0

Detour = persentase grid berhasil dengan lintasan lebih dari 18 langkah. Pada 45% dan 50% hanya 29 dan 9 grid yang berhasil sehingga rata-rata kurang stabil.



Gambar 3. *Success rate* A* terhadap densitas hambatan (garis putus-putus vertikal: densitas studi kasus)



Gambar 4. Node dieksplorasi (a) dan panjang lintasan rata-rata (b) terhadap densitas hambatan

Pembahasan

Kegagalan navigasi muncul sebagai transisi bertahap, bukan batas tunggal. *Success rate* sekitar 95% atau lebih hingga densitas 13%, turun menjadi 85,5% pada 20%, 52,0% pada 30%, 28,0% pada 35%, dan di bawah 5% pada sekitar 43% (Tabel 3, Gambar 3). Rentang 20%–35% pada studi kasus karena itu lebih tepat dibaca sebagai bagian dari zona transisi (sekitar 25%–40%) daripada ambang tunggal. Kegagalan pada grid 35% adalah salah satu realisasi yang mungkin (peluang berhasil 28%), dan keberhasilan pada grid 20% bukan jaminan (peluang gagal sekitar 14,5%). Pola ini sejalan dengan teori perkolasi pada kisi persegi, yaitu konektivitas jarak jauh hilang ketika fraksi hambatan mendekati nilai kritis sekitar 40,7% (fraksi sel bebas kritis 0,5927) pada kisi tak hingga [9], [10]. Titik 50% pada grid 10×10 (sekitar 30,5%) berada di bawah nilai tersebut. Salah satu penyebabnya adalah *Start* dan *Goal* berada di sudut grid, sedangkan pengaruh ukuran grid yang terbatas tidak diuji terpisah.

Beban komputasi menurun seiring naiknya densitas pada grid yang berhasil: rata-rata node turun dari 58,9 (10%) menjadi 55,4 (16%), 52,0 (20%), dan 33,7 (40%). Penurunan dari 61 node (16%) ke 52 node (20%) pada studi kasus karena itu sejalan dengan tren umum, bukan sekadar kebetulan. Hambatan yang lebih banyak berarti sel bebas yang dapat diperluas lebih sedikit, sedangkan perluasan pada plateau f = 18 tetap berlangsung. Pada grid yang gagal, A* menghabiskan seluruh wilayah terjangkau sehingga jumlah node mengikuti ukuran wilayah itu (48 node pada 35%). Pola ini berlaku untuk grid kecil dengan *Start–Goal* tetap dan perlu dibandingkan dengan studi pada peta yang lebih besar dan beragam [2], [6]. Aturan *tie-breaking* berbasis h(n) atau pemangkasan seperti *Jump Point Search* [4] berpotensi mengurangi node yang diperluas, tetapi belum diuji.

Densitas lebih memengaruhi ada-tidaknya lintasan daripada panjangnya. Rata-rata panjang lintasan hanya naik dari 18,00 pada 10% menjadi 18,48 (30%) dan 18,65 (35%), dan proporsi grid yang memerlukan jalur memutar adalah 1,5% pada 20%, 16,0% pada 30%, dan 20,4% pada 35%. Penambahan hambatan dengan demikian tidak selalu berbanding lurus dengan penambahan panjang lintasan; dampak dominannya adalah pada konektivitas.

Terdapat dua mekanisme kegagalan, yaitu *Start* atau *Goal* terkurung langsung dan pemutusan oleh dinding hambatan seperti pada studi kasus 35%. Mekanisme pertama menyumbang 67,5% kegagalan pada densitas 10%, 55,5% pada 20%, 36,4% pada 30%, dan 32,7% pada 35%; sisanya berasal dari pemutusan oleh dinding atau kombinasi keduanya. Pada densitas rendah hingga sedang kegagalan terutama berasal dari sudut terkurung, sedangkan pada densitas tinggi pemutusan oleh dinding semakin dominan. Hasil ini dirinci pada Tabel 4. Akses di sekitar area *receiving* dan *picking* karena itu sama pentingnya dengan kerapatan rak secara keseluruhan.

Tabel 4. Kegagalan akibat *Start* atau *Goal* terkurung (5.000 grid per densitas)

Densitas (%)	Grid gagal (%)	Karena sudut terkurung	Porsi sudut dari kegagalan (%)
10	2,3	77	67,5
20	15,0	415	55,5
30	49,0	891	36,4
35	72,5	1.187	32,7
40	89,1	1.476	33,1

Rekomendasi Tata Letak

Berdasarkan zona transisi, disusun acuan densitas indikatif pada Tabel 5. Acuan ini berlaku untuk model yang diuji (grid 10×10, hambatan acak, *Start–Goal* di sudut) dan dimaksudkan sebagai panduan awal, bukan batas mutlak untuk gudang nyata.

Tabel 5. Zona densitas hambatan dan tindakan yang disarankan

Zona	Densitas	Success rate	Tindakan yang disarankan
Aman	≤ 13%	≥ 95%	Verifikasi konektivitas dasar sebelum tata letak diterapkan
Waspada	13%–30%	95% turun ke 50%	Wajib verifikasi konektivitas dan akses titik kerja; perbaiki bila gagal
Berisiko	> 30%	< 50%	Hindari tanpa desain ulang; buka celah atau lorong cadangan hingga lolos verifikasi

Protokol verifikasi tata letak terdiri atas tujuh langkah. Pertama, tetapkan titik kerja dan hitung densitas rak yang direncanakan, lalu bandingkan dengan Tabel 5. Kedua, modelkan tata letak sebagai grid biner. Ketiga, verifikasi konektivitas dengan BFS: setiap titik kerja harus terjangkau dari titik asal robot, dan tata letak yang gagal ditolak atau diperbaiki. Keempat, pastikan setiap titik kerja memiliki sedikitnya satu, dan sebaiknya dua, sel akses bebas; mekanisme sudut terkurung menyumbang sekitar 33%–68% kegagalan pada simulasi. Kelima, hindari barisan rak yang menutup

p-ISSN: 1858-4217, e-ISSN: 2622-710X, DOI: https://doi.org/10.31849/e4tx4312

penuh satu lintasan seperti Kolom 7 pada Gambar 2c dan sisakan sedikitnya satu celah pada setiap barisan. Keenam, jalankan A* pada tata letak dan catat status, panjang lintasan, dan node dieksplorasi; bila *Path Not Found*, buka celah lalu ulangi dari langkah ketiga. Ketujuh, untuk operasi multi-robot atau lingkungan dinamis, lakukan pengujian tambahan.

Jawaban terhadap Rumusan Masalah

Tabel 6. Ringkasan jawaban rumusan masalah

No.	Rumusan masalah	Jawaban berdasarkan hasil
1	Pengaruh densitas (16%, 20%, 35%) terhadap jumlah langkah	Langkah tetap 18 pada 16% dan 20%; rata-rata *Monte Carlo* hanya naik dari 18,00 (≤ 10%) ke 18,65 (35%). Densitas lebih memengaruhi node dan ada-tidaknya lintasan.
2	Efektivitas A* mempertahankan jalur optimal pada ruang gerak terbatas	Selama konektivitas ada, A* selalu menemukan lintasan optimal (*Manhattan* atau lebih panjang bila perlu memutar); *detour* 1,5% pada 20% dan 20,4% pada 35%. Lorong sempit terstruktur belum diuji.
3	Ambang kritis sebelum *path not found* akibat isolasi grid	Berupa zona transisi: ≥ 95% berhasil hingga ± 13%, 50% pada ± 30,5%, < 5% pada ± 43%.
4	Korelasi konfigurasi hambatan dengan stabilitas navigasi dan jarak tempuh	Tidak berbanding lurus. Konfigurasi (sudut terkurung, dinding) menentukan keberhasilan; panjang lintasan hanya naik tipis dan node justru turun.

Keterbatasan

Penelitian ini memakai grid kecil 10×10, *Start–Goal* tetap di sudut, satu robot, hambatan statis acak (bukan rak yang tertata dalam lorong), gerak empat arah, dan waktu komputasi dari eksekusi tunggal. A* hanya melaporkan bahwa lintasan tidak ada; penelitian ini tidak mengklaim A* mencegah *deadlock* antar-robot [8], [11]. Angka-angka di atas khusus untuk model yang diuji dan tidak dapat langsung digeneralisasi ke gudang nyata yang umumnya tertata dalam lorong [1], [3], [7].

4. KESIMPULAN

Pada studi kasus, A* dengan heuristik *Manhattan* menemukan lintasan optimal 18 langkah pada densitas 16% dan 20% dengan 61 dan 52 node dieksplorasi, tetapi gagal pada densitas 35% karena dinding Kolom 7 memutus konektivitas menuju *Goal*. Eksperimen *Monte Carlo* pada 46.000 eksekusi menunjukkan bahwa batas kegagalan berupa zona transisi, bukan satu nilai: *success rate* turun di bawah 95% pada densitas sekitar 13%, mencapai 50% pada sekitar 30,5%, dan di bawah 5% pada sekitar 43%. Densitas hambatan lebih memengaruhi ada-tidaknya lintasan daripada panjangnya, sedangkan jumlah node yang dieksplorasi pada grid yang berhasil justru menurun. Kegagalan banyak berasal dari titik kerja yang terkurung di sudut pada densitas rendah dan dari dinding hambatan pada densitas tinggi.

Hasil ini menjadi dasar zona densitas indikatif dan protokol verifikasi konektivitas tata letak gudang, yang berlaku untuk model yang diuji. Penelitian lanjutan disarankan menguji grid yang lebih besar, rak yang tertata dalam lorong, titik awal dan tujuan yang bervariasi, aturan *tie-breaking* dan *Jump Point Search*, serta skenario multi-robot dan lingkungan dinamis.

5. DAFTAR PUSTAKA

[1] R. de Koster, T. Le-Duc, dan K. J. Roodbergen, 2007, Design and control of warehouse order picking: A literature review, *European Journal of Operational Research*, Vol. 182, No. 2, hal. 481–501, doi: 10.1016/j.ejor.2006.07.009.

[2] A. Elshahed, M. K. B. M. Ali, A. S. A. Mohamed, F. A. B. Abdullah, dan T. L. J. Aun, 2025, Efficient pathfinding on grid maps: Comparative analysis of classical algorithms and incremental line search, *IEEE Access*, Vol. 13, hal. 98473–98484, doi: 10.1109/ACCESS.2025.3575168.

[3] G. Fragapane, R. de Koster, F. Sgarbossa, dan J. O. Strandhagen, 2021, Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda, *European Journal of Operational Research*, Vol. 294, No. 2, hal. 405–426, doi: 10.1016/j.ejor.2021.01.019.

[4] D. Harabor dan A. Grastien, 2011, Online graph pruning for pathfinding on grid maps, *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, hal. 1114–1119.

[5] P. E. Hart, N. J. Nilsson, dan B. Raphael, 1968, A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, hal. 100–107, doi: 10.1109/TSSC.1968.300136.

[6] A. Kherrour, M. Robol, M. Roveri, dkk., 2025, A multi-algorithm pathfinding method: Exploiting performance variations for enhanced efficiency, *Annals of Mathematics and Artificial Intelligence*, Vol. 93, hal. 569–588, doi: 10.1007/s10472-024-09957-3.

[7] T. Lackner, J. Hermann, C. Kuhn, dan D. Palm, 2024, Review of autonomous mobile robots in intralogistics: State-of-the-art, limitations and research gaps, *Procedia CIRP*, Vol. 130, hal. 930–935, doi: 10.1016/j.procir.2024.10.187.

[8] K. Lee dan S. Park, 2023, Geometric zone-control algorithm for collision and deadlock

avoidance in AGV system, *IEEE Access*, Vol. 11, hal. 131289–131301, doi: 10.1109/ACCESS.2023.3335843.

[9] M. E. J. Newman dan R. M. Ziff, 2000, Efficient Monte Carlo algorithm and high-precision results for percolation, *Physical Review Letters*, Vol. 85, hal. 4104–4107, doi: 10.1103/PhysRevLett.85.4104.

[10] D. Stauffer dan A. Aharony, 1994, *Introduction to Percolation Theory*, edisi ke-2, Taylor & Francis, London.

[11] R. Stern dkk., 2019, Multi-agent pathfinding: Definitions, variants, and benchmarks, *Proceedings of the 12th International Symposium on Combinatorial Search*, arXiv:1906.08291.

[12] N. R. Sturtevant, 2012, Benchmarks for grid-based pathfinding, *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 4, No. 2, hal. 144–148, doi: 10.1109/TCIAIG.2012.2197681.

[13] E. G. Tsardoulias, A. Iliakopoulou, A. Kargakos, dan L. Petrou, 2016, A review of global path planning methods for occupancy grid maps regardless of obstacle density, *Journal of Intelligent & Robotic Systems*, Vol. 84, No. 1–4, hal. 829–858, doi: 10.1007/s10846-016-0362-z.

[14] E. B. Wilson, 1927, Probable inference, the law of succession, and statistical inference, *Journal of the American Statistical Association*, Vol. 22, No. 158, hal. 209–212, doi: 10.1080/01621459.1927.10502953.