

Implementasi Algoritma *Cuckoo Search* untuk Optimasi Lintasan dan Penghindaran Rintangan Robot PoleBot

Adhitya Sumardi Sunarya¹, Wahyu Adhie Candra², Rifqiaqil Mufahir Erwansyah*³, Pipit Anggraeni⁴, Noval Lilansa⁵, Nur Jamiludin Ramadhan⁶

^{1,3,5}Teknologi Rekayasa Mekatronika, Politeknik Manufaktur Bandung, Bandung, Indonesia

²Teknologi Rekayasa Sistem Aerial Nirawak, Politeknik Manufaktur Bandung, Bandung, Indonesia

⁴S2 SFS, Politeknik Manufaktur Bandung, Bandung, Indonesia

⁶Teknologi Rekayasa Otomasi, Politeknik Manufaktur Bandung, Bandung, Indonesia

e-mail: ¹adhitya@polman-bandung.ac.id, ²wahyu@polman-bandung.ac.id, *³222442020@mhs.polman-bandung.ac.id, ⁴pipit_anggraeni@polman-bandung.ac.id, ⁵noval_l@polman-bandung.ac.id, ⁶nj_ramadhan@polman-bandung.ac.id

Abstrak

Penelitian ini mengimplementasikan *Cuckoo Search Algorithm (CSA)* untuk optimasi lintasan dan penghindaran rintangan pada robot PoleBot AMR berpenggerak diferensial berbasis ROS2 Jazzy. CSA diterapkan pada dua tingkat: CSA global yang mengoptimasi waypoint untuk tiga kandidat lintasan, dan CSA lokal yang memilih pasangan kecepatan linear dan sudut pada horizon prediksi 1,15 s sebagai lapisan penghindaran rintangan. Keduanya berjalan sebagai satu simpul ROS2 yang terintegrasi dengan driver motor servo Tongyi (CANopen), LiDAR 2D Autonics LSC 270°, dan lokalisasi AMCL. Perencana global CSA dibandingkan dengan Genetic Algorithm (GA) dan RE-RRT pada navigator yang identik dengan lapisan lokal CSA yang tetap, melalui 30 percobaan simulasi 2D (10 per konfigurasi). Seluruh konfigurasi mencapai target tanpa tabrakan. GA paling efisien (17,4 m; 177 s) dengan jarak bebas minimum terkecil (0,59 m); RE-RRT menghasilkan lintasan terpanjang dan paling bervariasi ($19,4 \pm 1,8$ m) dengan jarak bebas minimum terbesar (1,69 m); CSA berada di antaranya (17,6 m; 181 s; 0,67 m). Pada robot fisik dengan rintangan statis, 10 dari 10 percobaan berhasil tanpa tabrakan (242 ± 32 s; efisiensi lintasan 0,84). Implementasi pada robot fisik mengungkap enam masalah integrasi, antara lain pose TF yang basi dan batas pengereman motor, beserta perbaikannya.

Kata kunci— Autonomous Mobile Robot, Cuckoo Search Algorithm, Penghindaran Rintangan, PoleBot, ROS2

Abstract

This study implements the *Cuckoo Search Algorithm (CSA)* for path optimization and obstacle avoidance on the differential-drive PoleBot AMR running ROS2 Jazzy. CSA is applied at two levels: a global CSA that optimizes waypoints for three candidate paths, and a local CSA that selects linear and angular velocity pairs over a 1.15 s prediction horizon as the obstacle-avoidance layer. Both run as a single ROS2 node integrated with the Tongyi servo motor driver (CANopen), the Autonics LSC 270° 2D LiDAR, and AMCL localization. The global CSA planner was compared with the Genetic Algorithm (GA) and RE-RRT on an identical navigator with the CSA local layer fixed, in 30 2D-simulation runs (10 per configuration). All configurations reached the goal without collision. GA was the most efficient (17.4 m; 177 s) with the smallest

minimum clearance (0.59 m); RE-RRT produced the longest and most variable paths (19.4 ± 1.8 m) with the largest minimum clearance (1.69 m); CSA lay in between (17.6 m; 181 s; 0.67 m). On the physical robot with static obstacles, 10 of 10 runs succeeded without collision (242 ± 32 s; path efficiency 0.84). The physical implementation exposed six integration problems, including stale TF poses and the motor braking limit, together with their fixes.

Keywords— *Autonomous Mobile Robot, Cuckoo Search Algorithm, Obstacle Avoidance, PoleBot, ROS2*

1. PENDAHULUAN

Autonomous Mobile Robot (AMR) semakin banyak digunakan pada aplikasi industri, logistik, dan layanan karena mampu bergerak mandiri dan membawa muatan dengan intervensi manusia yang minimal [1]. Kebutuhan utamanya adalah bernavigasi secara aman dan efisien di lingkungan yang memuat rintangan statis, seperti dinding dan perabot, maupun rintangan dinamis, seperti manusia yang berjalan [2], [3].

PoleBot AMR adalah platform robot berbasis ROS2 yang dikembangkan untuk penelitian navigasi otonom [4]. Platform ini menyediakan infrastruktur mekanik, elektrik, dan sensor, tetapi kinerja navigasinya bergantung pada perencana lintasan dan pengambil keputusan penghindaran rintangan yang dipasang di atasnya, serta pada seberapa baik perangkat lunak tersebut terintegrasi dengan driver motor, sensor, dan lokalisasi.

Berbagai metode telah dipakai untuk perencanaan lintasan dan penghindaran rintangan, antara lain *Artificial Potential Field* (APF) [5], *Genetic Algorithm* (GA) [6], [7], *Particle Swarm Optimization* (PSO) [8], *Rapidly-exploring Random Tree* (RRT) [9], dan *Dynamic Window Approach* (DWA) [10]. APF konvensional diketahui dapat terjebak pada minimum lokal sehingga varian yang diperbaiki dikembangkan [5]; metode berbasis metaheuristik memperluas ruang pencarian solusi, tetapi biaya komputasi dan kemampuan reaktifnya bergantung pada representasi solusi dan anggaran evaluasi yang dipakai.

Cuckoo Search Algorithm (CSA) adalah algoritma optimasi bio-inspirasi yang memadukan perilaku parasitisme sarang burung *cuckoo* dengan langkah *Lévy flight* [11]–[13]. Pada robot, CSA telah dipakai untuk perencanaan lintasan waktu-optimal [14] dan untuk perencanaan jalur dengan seleksi turnamen guna mengurangi jebakan minimum lokal [15]. Dua penelitian tersebut menempatkan CSA terutama sebagai perencana. Pada kedua penelitian tersebut, CSA tidak diuji sebagai pemilih kecepatan lokal yang dipasangkan dengan perencana global lain. Hal yang kami tuju adalah implementasi CSA pada dua tingkat (perencana global dan pemilih kecepatan lokal) sebagai satu simpul ROS2 pada robot fisik beserta dokumentasi integrasinya, dan perbandingannya dengan GA dan RE-RRT pada navigator yang identik. Posisi penelitian ini terhadap studi terdahulu dirangkum pada Tabel 1.

Tabel 1 Posisi penelitian terhadap studi terdahulu

Studi	Algoritma	Peran	Fokus
[14]	CSA yang diperbaiki	Lintasan waktu-optimal	Waktu gerak minimum robot
[15]	CSA + seleksi turnamen	Perencanaan jalur	Seleksi turnamen menggantikan seleksi acak untuk mengurangi jebakan minimum lokal

Studi	Algoritma	Peran	Fokus
[7]	GA + <i>velocity obstacle</i> (GAVO)	Penghindaran rintangan lokal	GA pada pengambilan keputusan <i>real-time</i> dengan LiDAR untuk rintangan statis dan bergerak
[6]	GA	Perencanaan jalur	Lintasan pada lingkungan dengan banyak rintangan
[9]	RE-RRT	Perencanaan lintasan	Lintasan lebih cepat dan mulus
[10]	A* + DWA (IA-DWA)	Perencana global dan lokal	Waktu perencanaan turun 23,3% dan panjang jalur 1,8% dibanding A*-DWA; diuji pada peta raster dan robot ROS
Penelitian ini	CSA global dan lokal; GA; RE-RRT	Implementasi CSA dua tingkat pada PoleBot dengan ROS2; CSA lokal tetap pada tiga perencanaan global	Integrasi perangkat keras–perangkat lunak; panjang lintasan, waktu, jarak bebas minimum; simulasi 2D dan validasi fisik

Dalam artikel ini, “optimasi lintasan” berarti optimasi waypoint oleh CSA global untuk tiga kandidat lintasan, dan “penghindaran rintangan” berarti pemilihan pasangan kecepatan oleh CSA lokal. Kriteria keberhasilan ditetapkan bertingkat: (i) target tercapai dalam toleransi 0,45 m tanpa tabrakan pada seluruh percobaan; (ii) jarak bebas minimum sebagai ukuran keselamatan, karena robot diarahkan beroperasi di dekat manusia; (iii) panjang lintasan; dan (iv) waktu menuju target. Metode dibandingkan per metrik, dan tidak ada klaim “terbaik” tanpa menyebut metriknya.

Kontribusi penelitian ini adalah: (1) implementasi CSA dua tingkat, yaitu CSA waypoint untuk tiga profil kandidat lintasan (langsung, belok kiri, belok kanan) dan CSA lokal untuk memilih pasangan kecepatan, sebagai satu simpul ROS2 pada PoleBot; seleksi turnamen diadopsi dari [15] sehingga kebaruan terletak pada implementasi dan konfigurasinya, bukan pada algoritma dasarnya; (2) dokumentasi integrasi dengan driver motor CANopen, LiDAR, TF, dan AMCL beserta masalah dan perbaikan yang ditemukan pada robot fisik; (3) lapisan lokal CSA yang dipakai ulang oleh GA dan RE-RRT dengan fungsi *fitness* identik, dievaluasi pada 30 percobaan simulasi dengan analisis statistik sederhana; dan (4) validasi terbatas pada robot fisik dengan rintangan statis.

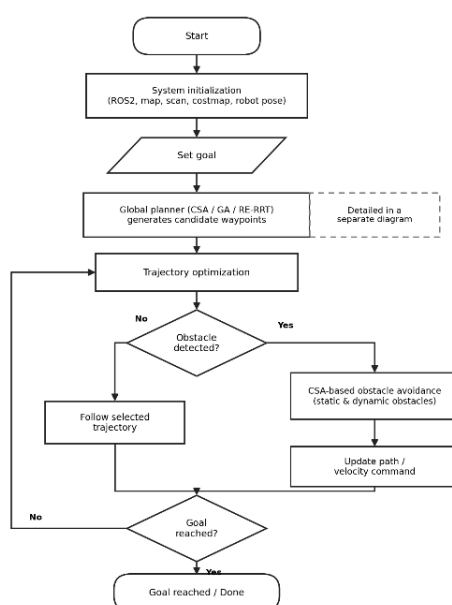
2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan implementasi eksperimental untuk mengevaluasi CSA dua tingkat pada robot PoleBot AMR. Pengujian dilakukan pada simulasi 2D ringan berbasis ROS2 Jazzy sebanyak 30 percobaan (10 per konfigurasi perencanaan global: CSA, GA, dan RE-RRT) pada navigator yang identik, dilanjutkan validasi terbatas pada robot fisik dengan rintangan statis.

2.1 Arsitektur Sistem

Perangkat lunak terdiri atas simpul driver, simpul lokalisasi, dan satu simpul navigator CSA yang bukan *plugin* Nav2, berjalan pada ROS2 Jazzy [16]. Driver motor Tongyi menerbitkan /odom (sekitar 50 Hz) dan TF odom→base_link serta menerima /cmd_vel. Driver LiDAR menerbitkan /scan pada bingkai *laser*, sedangkan TF base_link→laser diterbitkan oleh

robot_state_publisher dari URDF dengan pemasangan LiDAR (0,450; 0,005; 0,267) m dan yaw -90° . AMCL dan *map_server* (Nav2) menyediakan TF *map*→*odom*. Simpul navigator berlangganan /*map*, /*scan*, /*goal_pose*, dan TF, lalu menerbitkan perintah kecepatan langsung ke /*cmd_vel* serta lintasan terpilih, tiga lintasan kandidat, jejak yang ditempuh, dan status perencanaan untuk RViz. Pencatat data menulis keadaan ke berkas CSV pada 10 Hz. Alur kerja navigator ditunjukkan pada Gambar 1: setelah target diterima, perencanaan global membangkitkan tiga kandidat lintasan dan lintasan pertama dipilih; lintasan diikuti dengan *pure pursuit* dan CSA lokal memilih kecepatan pada 10 Hz. Bila robot macet selama 5 s, robot berpindah ke kandidat berikutnya, dan bila ketiga kandidat terblokir, set kandidat baru dibangkitkan; saat pemulihan robot mundur hingga jarak bebas pada peta mencapai 0,5 m (maksimum 5 s). Pada profil yang dipakai, titik pindai yang jatuh dekat sel peta terisi tidak diteruskan ke lapisan reaktif (filter peta); rintangan terpetakan dihindari melalui lintasan global, jarak bebas pada peta, dan pengecekan *footprint* persegi saat evaluasi kandidat, sedangkan titik pindai tingkat penuh dipakai untuk pengereman darurat. Perlu ditekankan bahwa pemilih kecepatan lokal selalu CSA, terlepas dari perencanaan global yang aktif; GA dan RE-RRT hanya menggantikan perencanaan global.



Gambar 1 Diagram alir optimasi lintasan dan penghindaran rintangan berbasis CSA pada PoleBot AMR (perencana global dapat berupa CSA, GA, atau RE-RRT, sedangkan pemilih kecepatan lokal selalu CSA).

PoleBot AMR berpengerak diferensial dengan badan $0,928 \times 0,638$ m, jarak antar roda 0,59 m, dan jari-jari roda 0,079 m (Gambar 2). Komputasi memakai mini PC Intel NUC i3 dengan Ubuntu 24.04 dan ROS2 Jazzy. Aktuator berupa motor servo AC Tongyi ± 400 W (3000 rpm nominal) dengan driver servo digital berumpan balik *encoder* dan antarmuka CANopen melalui SocketCAN *can0* 500 kbps. LiDAR 2D Autonic LSC 270° [17] terhubung melalui Ethernet TCP/IP dan menerbitkan /*scan* pada 5 Hz dengan jangkauan 0,30–25 m; sudut pindai dipotong -28° hingga 200° untuk membuang pantulan badan robot sendiri. Sensor jarak dekat berupa ultrasonik HC-SR04 melalui mikrokontroler Teensy, dengan catu daya 48 V dan *DC-DC converter* ke 12 V untuk mikrokontroler dan kendali. Pemasangan LiDAR 0,45 m di depan poros roda dan bumper depan 0,49 m dari poros menjadi dasar parameter *footprint* dan jarak bebas pada Subbab 2.3.



Gambar 2 Platform PoleBot AMR yang digunakan.

2.2 Model Kinematika PoleBot AMR

Robot berpengerak diferensial. Kecepatan linear v dan sudut ω dinyatakan dari kecepatan roda kanan v_R dan kiri v_L serta jarak antar roda b :

$$v = \frac{v_R + v_L}{2} \quad (1)$$

$$\omega = \frac{v_R - v_L}{b} \quad (2)$$

Pose prediksi pada langkah diskret Δt dihitung dengan model *unicycle*:

$$\begin{aligned} x(k+1) &= x(k) + v \cos \theta(k) \Delta t \\ y(k+1) &= y(k) + v \sin \theta(k) \Delta t \\ \theta(k+1) &= \theta(k) + \omega \Delta t \end{aligned} \quad (3)$$

2.3 Formulasi Algoritma *Cuckoo Search* (CSA)

CSA diimplementasikan pada dua tingkat. CSA lokal memilih pasangan kecepatan linear dan sudut pada tiap siklus kendali sebagai lapisan penghindaran rintangan, sedangkan CSA global mengoptimasi waypoint untuk tiga kandidat lintasan. Keduanya memakai langkah Lévy [11]–[13], evaluasi *fitness* atau biaya, dan penggantian sebagian kandidat terburuk.

CSA lokal untuk pemilihan kecepatan. Setiap kandidat adalah pasangan (v, ω) . Populasi awal berisi 10 kandidat *seed* yang diturunkan dari referensi *pure pursuit* dan bias penghindaran, dilengkapi kandidat acak hingga $n = 28$. Pada tiap iterasi, kandidat diperbarui dengan langkah Lévy yang dibangkitkan dengan algoritma Mantegna:

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta) \sin(\pi\beta/2)}{\Gamma\left(\frac{1+\beta}{2}\right) \beta 2^{(\beta-1)/2}} \right\}^{1/\beta} \quad (4)$$

$$s = \frac{u}{|z|^{1/\beta}}, \quad u \sim N(0, \sigma_u^2), \quad z \sim N(0, 1) \quad (5)$$

$$X_i(t+1) = X_i(t) + \alpha s [X_i(t) - X_{\text{best}}(t)] \quad (6)$$

Hasil pembaruan dibatasi pada rentang kecepatan yang diizinkan. Sebagian p_a dari kandidat terburuk diganti kandidat acak, dan proses berhenti setelah iterasi tetap (3 iterasi). Tiap kandidat dievaluasi dengan memproyeksikan lintasan selama 1,15 s (11 langkah, $\Delta t = 0,10$ s). Kandidat dinyatakan tidak valid (biaya 10^6) bila pada salah satu langkah jarak ke titik pindai dikurangi radius robot 0,23 m kurang dari 0,20 m, bila jarak ke rintangan pada peta kurang dari 0,33 m, atau bila *footprint* persegi menyentuh sel terisi. Untuk kandidat valid, biaya adalah:

$$J = w_p d_p + w_h |e_\theta| + \frac{w_o}{(c_d + \varepsilon)^2} + \frac{w_m}{(c_m + \varepsilon)^2} + w_v (|v - v_{\text{ref}}| + 0,28 |\omega - \omega_{\text{ref}}|) + w_s (0,5 |\Delta v| + 1,35 |\Delta \omega|) + J_{\text{prog}} + 0,25 |\omega| \quad (7)$$

dengan d_p jarak titik akhir ke lintasan, e_θ galat arah ke titik *lookahead*, c_d dan c_m jarak bebas dinamis (dari pindai LiDAR) dan jarak bebas pada peta, Δv dan $\Delta \omega$ perubahan terhadap perintah sebelumnya, dan J_{prog} suku kemajuan menuju target ($-1,30 \times$ kemajuan bila positif, $4,0 \times$ |kemajuan| bila negatif). Bobot w_p , w_o , dan w_h bergantung pada zona jarak depan: bahaya ($< 1,05$ m), hati-hati ($1,05$ – $1,35$ m), dan bebas. Nilai seluruh parameter dirangkum pada Tabel 2.

Tabel 2 Parameter CSA lokal dan pengendali

Parameter	Symbol	Nilai
Ukuran populasi; iterasi	n ; –	28; 3
Probabilitas pengabaian	p_a	0,25
Parameter Lévy; skala langkah	β ; α	1,5; 0,055
Horizon; langkah prediksi	–; Δt	1,15 s; 0,10 s
Probabilitas kandidat Gaussian; σ_v ; σ_ω	–	0,90; 0,05; 0,10
Kecepatan linear maksimum	v_{max}	0,12 m/s
Kecepatan sudut maks. (bebas; dekat rintangan)	ω_{max}	0,35; 0,95 rad/s
Batas perubahan perintah per siklus	Δv ; $\Delta \omega$	0,05 m/s; 0,05 rad/s
Zona bahaya; hati-hati (jarak depan)	–	1,05; 1,35 m
Ambang darurat; jangkauan rintangan	–	0,85; 1,80 m
Bobot (w_p ; w_o ; w_h) zona bahaya	–	3,0; 14,0; 1,8
Bobot (w_p ; w_o ; w_h) zona hati-hati	–	2,5; 11,0; 2,0
Bobot (w_p ; w_o ; w_h) zona bebas	–	7,0; 0,6; 2,4
Bobot peta w_m ; bobot kehalusan w_s	–	2,4; 2,5
Footprint persegi (depan/belakang/samping)	–	0,49; 0,49; 0,31 m
Frekuensi kendali	–	10 Hz

Perencana lintasan global. Ketiga perencana membangkitkan tiga kandidat lintasan dengan profil yang sama: langsung (offset 0), belok kiri (+1,2 m), dan belok kanan (–1,2 m). Kandidat memakai 5 waypoint antara posisi awal dan target, kecuali RE-RRT yang menghasilkan jumlah titik bervariasi. Seluruh perencana dinilai dengan fungsi *fitness* yang identik:

$$F = w_L L + w_S S + w_O \frac{1}{N} \sum_k \frac{1}{\max(c_k, 0,04)} + w_H H + w_Y |\bar{y} - y^*| + D + 500 N_{\text{tab}} + 800 N_{\text{luar}} \quad (8)$$

dengan L panjang lintasan, S jumlah sudut belok, c_k jarak bebas pada sel ke- k , H penalti riwayat lintasan terblokir, $\bar{y} - y^*$ simpangan lateral terhadap profil, D penalti kemiripan antar kandidat, serta N_{tab} dan N_{luar} jumlah sel bertabrakan dan di luar peta. Bobot $w_L = 1,00$, $w_S = 0,35$, $w_O = 2,80$, $w_H = 1,00$, $w_Y = 0,45$. Lintasan pertama (langsung) dipilih sebagai awal; bila terblokir, robot berpindah ke kandidat berikutnya (2, lalu 3).

CSA waypoint. Satu sarang adalah himpunan waypoint. Sarang baru diperoleh dengan menarik induk (dipilih lewat turnamen $K = 4$) ke sarang terbaik dan ke profil dasar, ditambah langkah Lévy (skala 0,85 m) dan gangguan Gaussian tegak lurus lintasan ($\sigma = 0,30$ m). Sebagian $p_a = 0,25$ sarang terburuk diganti sarang acak; sarang terbaik selalu dipertahankan. **GA.**

Kromosom adalah deretan waypoint; operator berupa seleksi turnamen ($K = 4$), *crossover* (peluang 0,85; campuran aritmetik dan satu titik potong), mutasi Gaussian (peluang 0,25 per gen, $\sigma = 0,30$ m), dan elitisme dua individu. **RE-RRT**. Implementasi berupa RRT* dengan *informed sampling*, *rewiring*, dan pemangkasan *shortcut*; ini adalah adaptasi dari [9], bukan reproduksi utuh. Parameter ketiganya pada Tabel 3.

Tabel 3 Parameter perencanaan global

Parameter	CSA	GA	RE-RRT
Populasi / ukuran	48 sarang	48 individu	pohon maks. 2500 simpul
Iterasi	70	70	maks. 6000 sampel
Seleksi	turnamen $K = 4$	turnamen $K = 4$; elitisme 2	–
Operator	Lévy $\beta = 1,5$ (0,85 m); Gaussian 0,30 m; $p_a = 0,25$	<i>crossover</i> 0,85; mutasi 0,25 (σ 0,30 m)	langkah 0,40 m; <i>goal bias</i> 0,10; radius <i>rewire</i> 0,90 m; 150 iterasi <i>shortcut</i>
Kriteria berhenti	iterasi tetap	iterasi tetap	batas simpul/iterasi, atau 600 iterasi tanpa perbaikan
Percobaan per profil	maks. 6	maks. 6	maks. 6
Fungsi <i>fitness</i>	identik (persamaan 8)	identik	identik (dipakai menilai lintasan akhir)

Anggaran evaluasi CSA dan GA setara (48×70 evaluasi per profil), sedangkan RE-RRT dibatasi oleh jumlah simpul dan sampel pada Tabel 3.

2.4 Akuisisi Data dan Metrik Evaluasi

Keberhasilan berarti robot masuk dalam radius 0,45 m dari target tanpa tabrakan. Waktu menuju target adalah selang dari penerbitan target hingga pertama kali berada di dalam toleransi (waktu tempuh; *bukan* waktu perencanaan). Panjang lintasan adalah jumlah jarak antar-sampel pose yang tercatat. Jarak bebas minimum adalah nilai minimum jarak pindai LiDAR pada sektor depan ($\pm 23^\circ$) selama percobaan, tanpa filter peta, diukur dari pusat sensor (0,45 m di depan poros roda, sekitar 0,04 m di belakang muka depan badan). Efisiensi lintasan adalah jarak lurus awal–target dibagi panjang lintasan aktual. Hasil dilaporkan sebagai rerata $\pm$ simpangan baku, interval kepercayaan 95% (distribusi t, $n = 10$), dan selisih antar metode diuji dengan uji Welch; karena sifatnya eksploratif, nilai p tidak dikoreksi untuk perbandingan ganda.

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil implementasi CSA dua tingkat pada simulasi 2D (30 percobaan) dan validasi terbatas pada robot fisik. Hasil simulasi diperoleh dari simulator 2D ringan, sedangkan hasil robot fisik hanya mencakup konfigurasi CSA dengan rintangan statis.

3.1 Lingkungan Simulasi dan Skenario Pengujian

Simulasi memakai simulator 2D ringan yang memuat peta hunian, menggerakkan model diferensial dari perintah kecepatan, dan membangkitkan pindai LiDAR dengan *raycasting* (270 sampel pada sudut $\approx 228^\circ$; 0,30–25 m; 5 Hz; tanpa derau). Peta hunian area uji fisik (hingga 35×30 m, resolusi 0,05 m/sel) dipakai pada simulasi dan uji fisik, dengan rintangan statis pada peta dan tanpa rintangan bergerak. Posisi awal (−1,72; 3,66) m dan target (13,20; 11,65) m memberi jarak lurus $\approx 16,9$ m, dengan toleransi target 0,45 m dan batas waktu 250 s. Ketiga konfigurasi dijalankan pada navigator terpadu yang identik (*follower*, lapisan keselamatan, pemulihan, dan

seluruh parameter sama), sehingga hanya perencana global yang berbeda. Pada bagian hasil, nama CSA, GA, dan RE-RRT merujuk pada perencana global yang dipakai; CSA lokal selalu aktif pada ketiganya. Tiap konfigurasi diuji 10 kali dengan *seed* acak 1–10 (total 30 percobaan). Batas kecepatan roda diatur sama pada ketiga konfigurasi sehingga waktu tempuh dapat dibandingkan. Deteksi tabrakan memakai lingkaran $r = 0,28$ m terhadap sel terisi pada peta. Uji fisik memakai profil parameter yang sama dengan konfigurasi CSA.

3.2 Perbandingan Metode pada Navigasi Statis Multi-Lintasan

Ketiga konfigurasi berhasil pada 30 dari 30 percobaan tanpa tabrakan (Tabel 4). Selisih rerata antar metode beserta hasil uji Welch ditunjukkan pada Tabel 5.

Tabel 4 Navigator terpadu, 10 percobaan per konfigurasi (rerata $\pm$ SD; [IK 95%])

Perencana	Berhasil	Waktu ke target (s)	Waktu teoretis (s)	Panjang lintasan (m)	Kehalusan (std ω , rad/s)	Jarak bebas min. (m)
CSA	10/10	181 ± 2 [179,6–182,4]	146,7	$17,6 \pm 0,1$ [17,53–17,67]	0,035	0,67
GA	10/10	177 ± 2 [175,6–178,4]	145,0	$17,4 \pm 0,0$ (SD < 0,05)	0,032	0,59
RE-RRT	10/10	184 ± 14 [174,0–194,0]	161,7	$19,4 \pm 1,8$ [18,11–20,69]	0,032	1,69

(Jarak bebas minimum dilaporkan sebagai satu nilai ringkasan per konfigurasi; sebaran antar percobaan tidak tersedia sehingga tidak diuji secara statistik. Waktu teoretis dihitung sebagai panjang lintasan rerata dibagi kecepatan linear maksimum 0,12 m/s ($t = L/v_{\text{max}}$), yaitu waktu yang diharapkan bila robot selalu melaju pada kecepatan maksimum).

Tabel 5 Selisih rerata (CSA dikurangi pembandingan), IK 95%, dan p (uji Welch, tanpa koreksi)

Metrik	Pembandingan	Selisih	IK 95%	p
Panjang (m)	GA*	0,20	0,12 – 0,28	< 0,001
	RE-RRT	–1,80	–3,09 – –0,51	0,012
Waktu (s)	GA	4,0	2,1 – 5,9	< 0,001
	RE-RRT	–3,0	–13,1 – 7,1	0,52

*SD panjang lintasan GA dilaporkan 0,0 (dibulatkan); dihitung dengan 0,05 m. Angka dihitung dari rerata dan SD ringkasan ($n = 10$ per konfigurasi), bukan dari data per percobaan.

Lapisan lokal CSA bekerja dengan ketiga perencana global dan perbedaan terletak pada karakter lintasan, bukan keberhasilan. GA paling efisien (17,4 m; 177 s) tetapi memiliki jarak bebas minimum terkecil (0,59 m). RE-RRT menghasilkan lintasan terpanjang dan paling bervariasi ($19,4 \pm 1,8$ m) namun jarak bebas minimum terbesar (1,69 m), karena cenderung memutar menjauhi rintangan. CSA mendekati GA (selisih lintasan 0,2 m, sekitar 1%; waktu 4 s) dengan jarak bebas minimum di antara keduanya (0,67 m). Uji Welch (Tabel 5) menunjukkan GA lebih pendek dan lebih cepat daripada CSA secara nyata ($p < 0,001$), CSA lebih pendek daripada RE-RRT (selisih 1,8 m; $p = 0,012$), dan waktu CSA tidak berbeda nyata dari RE-RRT ($p = 0,52$). Variansi lintasan CSA kecil (SD 0,1 m) tetapi setara GA, sehingga CSA tidak dapat disebut paling konsisten. CSA tidak unggul pada satu metrik pun; keunggulannya hanya bahwa ia tidak buruk pada ketiga metrik.

3.3 Hasil Penghindaran Rintangan Dinamis

Jarak bebas minimum pada Tabel 4 (0,59–1,69 m) berada di bawah ambang zona bahaya (1,05 m) dan ambang darurat (0,85 m) pada CSA dan GA. Hal ini tidak berarti ambang dilanggar tanpa respons: jarak pada Tabel 4 dihitung dari pindai penuh yang memuat dinding terpetakan,

sedangkan ambang pada Tabel 2 berlaku pada titik pindai setelah filter peta. Dinding terpetakan ditangani melalui lintasan global, jarak bebas peta, dan pengecekan *footprint*, sehingga ambang tersebut tidak terpicu oleh dinding. Konsekuensinya, simulasi ini tidak menguji respons lapisan reaktif terhadap rintangan yang tidak ada di peta. Keberhasilan mencapai target juga tidak dengan sendirinya menunjukkan navigasi aman, sehingga jarak bebas minimum dan jumlah tabrakan dilaporkan terpisah.

3.4 Pengujian *Repeatability* (Konsistensi *Playback*)

Implementasi pada robot fisik mengungkap enam masalah integrasi yang tidak muncul pada simulasi. Pertama, pose TF menjadi basi dan robot berputar-putar karena *timeout* pencarian transformasi di dalam *callback* timer memblokir *thread* langganan /tf; *timeout* pencarian TF diatur 0 s. Kedua, robot belum berhenti di target karena rampa pada *velocity smoother* melanjutkan perintah sebelumnya; ini diperbaiki dengan penahan keadaan GOAL_REACHED dan penghentian langsung ke /cmd_vel. Ketiga, antrean AMCL penuh pada LiDAR 15 Hz, sehingga laju dibatasi 5 Hz dan *robot_state_publisher* wajib dijalankan. Keempat, pindai terbalik 180° karena yaw LiDAR pada URDF +90°, yang dikoreksi menjadi -90°. Kelima, pantulan badan robot muncul pada pindai, sehingga sudut dipotong, jarak minimum 0,30 m ditetapkan, dan filter kotak badan $0,62 \times 0,40$ m dipasang. Keenam, robot tidak sempat berhenti pada 0,20 m/s karena deselerasi motor terukur hanya 0,0251 m/s²; jarak henti $v^2/2a$ bernilai 0,29 m pada 0,12 m/s, dibanding 0,80 m pada 0,20 m/s yang melebihi jarak bumper 0,49 m, sehingga kecepatan linear maksimum dibatasi 0,12 m/s. Seluruh perbaikan ini diterapkan sebelum 10 percobaan fisik berikut.

PoleBot fisik dengan konfigurasi CSA diuji 10 kali pada area terbuka hingga 35 × 30 m dengan titik awal dan target identik, *seed* CSA tetap 42, dan rintangan statis (Tabel 6). Seluruh percobaan berhasil tanpa tabrakan, dengan durasi 242 ± 32 s, panjang lintasan $19,6 \pm 0,3$ m terhadap jarak lurus 16,49 m (efisiensi 0,84), dan jarak depan minimum 1,04 m. Perbedaan lingkungan, derau odometri, dan slip roda membuat angka fisik tidak dapat dibandingkan langsung dengan Tabel 4.

Tabel 6 Hasil navigasi pada robot fisik (N = 10)

Parameter	Nilai
Pencapaian target	10/10
Durasi tempuh (s)	242 ± 32
Panjang lintasan aktual (m); jarak lurus (m)	$19,6 \pm 0,3$; 16,49
Efisiensi lintasan	0,84
Kehalusan (std perintah ω , rad/s)	$0,060 \pm 0,004$
Jarak depan minimum (m)	1,04
Tabrakan	0

4. KESIMPULAN

1. CSA berhasil diimplementasikan pada dua tingkat, yaitu CSA waypoint untuk optimasi lintasan dan CSA lokal untuk penghindaran rintangan, sebagai satu simpul ROS2 pada PoleBot yang terintegrasi dengan driver motor Tongyi, LiDAR Autonics LSC, dan AMCL.
2. Pada 30 percobaan simulasi dengan tiga perencanaan global (CSA, GA, RE-RRT), seluruh konfigurasi mencapai target tanpa tabrakan di lingkungan statis, dan pada robot fisik 10 dari 10 percobaan berhasil.
3. Tidak satu pun konfigurasi unggul pada seluruh metrik: GA menghasilkan lintasan terpendek dan tercepat, RE-RRT memiliki jarak bebas minimum terbesar tetapi lintasan terpanjang dan

paling bervariasi, dan CSA berada di antaranya pada ketiga metrik tanpa unggul pada satu pun.

4. Implementasi pada robot fisik mengungkap enam masalah integrasi yang diperbaiki sebelum validasi.

5. SARAN

Perlu evaluasi dengan memperluas lingkungan dinamis yang lebih kompleks dengan jumlah rintangan bergerak yang lebih banyak, serta pengujian pada variasi tata letak ruangan yang berbeda untuk menilai generalisasi metode CSA secara lebih menyeluruh.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada tim riset PoleBot AMR Politeknik Manufaktur Bandung atas izin penggunaan spesifikasi platform serta hasil eksperimen yang disajikan.

DAFTAR PUSTAKA

- [1] Cognominal, M., Patronymic, K., dan Wańkiewicz, A., 2021, Evolving Field of Autonomous Mobile Robotics: Technological Advances and Applications, *Fusion of Multidisciplinary Research, An International Journal*, vol. 2, no. 2, hal. 189–200, doi: 10.63995/usas3015.
- [2] Sousa, L. C., dkk., 2025, Obstacle Avoidance Technique for Mobile Robots at Autonomous Human-Robot Collaborative Warehouse Environments, *Sensors*, vol. 25, no. 8, hal. 2387, doi: 10.3390/s25082387.
- [3] Louda, S., Karkar, N., Seghir, F., dan Boutalbi, O., 2025, Fuzzy Dynamic Feedback Linearization for Efficient Mobile Robot Trajectory Tracking and Obstacle Avoidance in Autonomous Navigation, *Int. J. Robot. Control Syst.*, vol. 5, no. 2, hal. 881–901, doi: 10.31763/ijrcs.v5i2.1780.
- [4] Anggraeni, P., Ramdhan, N., dan Khoirunnisa, H., 2021, PoleBot: Polman Education Robotics Open Platform, Politeknik Manufaktur Bandung, Bandung.
- [5] Hao, G., Lv, Q., Huang, Z., Zhao, H., dan Chen, W., 2023, UAV Path Planning Based on Improved Artificial Potential Field Method, *Aerospace*, vol. 10, no. 6, hal. 562, doi: 10.3390/aerospace10060562.
- [6] Rahmaniar, W. dan Rakhmania, A. E., 2022, Mobile Robot Path Planning in a Trajectory with Multiple Obstacles Using Genetic Algorithms, *J. Robot. Control (JRC)*, vol. 3, no. 1, hal. 1–7, doi: 10.18196/jrc.v3i1.11024.
- [7] Gyenes, Z., Bölöni, L., dan Szádeczky-Kardoss, E. G., 2023, Can Genetic Algorithms Be Used for Real-Time Obstacle Avoidance for LiDAR-Equipped Mobile Robots?, *Sensors*, vol. 23, no. 6, hal. 3039, doi: 10.3390/s23063039.
- [8] Ajeil, F. H., Ibraheem, I. K., Sahib, M. A., dan Humaidi, A. J., 2018, Multi-Objective Path Planning of an Autonomous Mobile Robot using Hybrid PSO-MFB Optimization Algorithm, *arXiv:1805.00224*, <https://arxiv.org/abs/1805.00224>.
- [9] Islam, N. U., Gul, K., Faizullah, F., Ullah, S. S., dan Syed, I., 2024, Trajectory optimization and obstacle avoidance of autonomous robot using Robust and Efficient Rapidly Exploring Random Tree, *PLoS One*, vol. 19, no. 10, hal. e0311179, doi: 10.1371/journal.pone.0311179.
- [10] He, Q., Wang, Z., Li, K., Zhang, Y., dan Li, M., 2025, Research on autonomous navigation of mobile robots based on IA-DWA algorithm, *Sci. Rep.*, vol. 15, no. 1, hal. 1–13, doi: 10.1038/s41598-024-84858-3.
- [11] Yang, X.-S. dan Deb, S., 2014, Cuckoo Search: Recent Advances and Applications, *Neural Comput. Appl.*, vol. 24, hal. 169–174, doi: 10.1007/s00521-013-1367-1.

- [12] Yang, X.-S. dan Deb, S., 2010, Engineering Optimisation by Cuckoo Search, *Int. J. Math. Model. Numer. Optim.*, vol. 1, no. 4, hal. 330–343, arXiv:1005.2908.
 - [13] Al-Abaji, M. A., 2021, Cuckoo Search Algorithm: Review and its Application, *Tikrit J. Pure Sci.*, vol. 26, no. 2, doi: 10.25130/tjps.26.2021.039.
 - [14] Wang, W., Tao, Q., Cao, Y., Wang, X., dan Zhang, X., 2020, Robot Time-Optimal Trajectory Planning Based on Improved Cuckoo Search Algorithm, *IEEE Access*, vol. 8, hal. 86923–86933, doi: 10.1109/ACCESS.2020.2992640.
 - [15] Sharma, K., Singh, S., dan Doriya, R., 2021, Optimized cuckoo search algorithm using tournament selection function for robot path planning, *Int. J. Adv. Robot. Syst.*, vol. 18, no. 3, doi: 10.1177/1729881421996136.
 - [16] Wei, Z., Wang, S., Chen, K., dan Wang, F., 2025, ROS-Based Navigation and Obstacle Avoidance: A Study of Architectures, Methods, and Trends, *Sensors*, vol. 25, no. 14, hal. 4306, doi: 10.3390/s25144306.
 - [17] Ran, Y., Xu, X., Tan, Z., dan Luo, M., 2025, A Review of 2D Lidar SLAM Research, *Remote Sens.*, vol. 17, no. 7, hal. 1214, doi: 10.3390/rs17071214.
-