

Perencanaan Jalur Autonomous Guided Vehicle (AGV) menggunakan Rapidly Exploring Random Trees (RRT) dan Dynamic Window Approach (DWA) pada ROS

Annisa Izaty^{1*}, Dini Hariani Fitri Lubis²

^{1,2} Politeknik Negeri Bandung; Bandung Barat; polban@polban.ac.id

Keywords:

Path Planning, Obstacle Avoidance, Rapidly Random Trees (RRT), Dynamic Window Approach (DWA), Robot Operating System (ROS).

Correspondent Email:

annisa.izaty@polban.ac.id

Abstrak. Kendaraan material atau robot pemindah barang (AGV) digunakan di lingkungan industri seperti pabrik atau gudang dan memiliki kemampuan navigasi. Sistem navigasi terdiri dari penghindaran rintangan dan perencanaan jalur. Metode perencanaan jalur berbasis sampling telah berkembang dari perencanaan robotik dasar ke aplikasi yang lebih kompleks dan beragam untuk mendapatkan solusi yang efektif. Penelitian ini mengusulkan algoritma terintegrasi baru untuk membuat jalur navigasi dalam lingkungan dinamis. Solusi untuk masalah ini biasanya terbatas pada perencanaan jalur tetapi tidak mencakup perintah kecepatan atau perubahan lingkungan di sekitar AGV. Untuk perencanaan jalur global, algoritma Rapidly-Exploring Random Trees (RRT) mengambil sampel acak dari ruang konfigurasi untuk menghasilkan jalur bebas tabrakan. Algoritma Dynamic Window Approach (DWA) digunakan untuk mengambil sampel ruang kecepatan forklift untuk perencanaan jalur lokal. Ini dilakukan dengan menghitung kecepatan translasi dan rotasi AGV, dan simulasi menunjukkan bahwa algoritma ini dapat membuat jalur AGV yang efisien dalam lingkungan dinamis.



Copyright © [JITET](http://www.jitet.org) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. *Material handling vehicles or automated guided vehicles (AGVs) are used in industrial environments such as factories or warehouses and have navigation capabilities. The navigation system consists of obstacle avoidance and path planning. Sampling-based path planning methods have evolved from basic robotic planning to more complex and diverse applications to obtain effective solutions. This article proposes a new integrated algorithm for creating navigation paths in dynamic environments. Solutions to this problem are usually limited to path planning but do not include speed commands or changes in the environment around the AGV. For global path planning, the Rapidly-Exploring Random Trees (RRT) algorithm takes random samples from the configuration space to generate collision-free paths. The Dynamic Window Approach (DWA) algorithm is used to sample the forklift's velocity space for local path planning. This is done by calculating the AGV's translational and rotational velocities, and simulations show that this algorithm can create efficient AGV paths in dynamic environments.*

1. PENDAHULUAN

Revolusi Industri Keempat (IR 4.0) adalah istilah yang diusulkan pertama kali di Jerman pada tahun 2011 [1] dan mengacu pada penggunaan kecerdasan buatan di semua bidang, termasuk desain, manufaktur, produksi, distribusi, dan pembayaran barang dan jasa.

Revolusi Industri 4.0 melibatkan kemajuan teknologi yang pesat. Dibandingkan dengan gudang tradisional, proses otomatisasi meningkatkan fungsionalitas gudang pintar. Tujuannya adalah untuk mengurangi biaya operasional gudang sambil meningkatkan fungsi yang rumit dan aksesibilitas kualitas [2].

Gudang kontemporer memiliki alat yang digunakan untuk mengangkut kebutuhan logistik. Untuk menangani barang-barang di gudang, transportasi cepat (AGV) adalah cara terbaik.

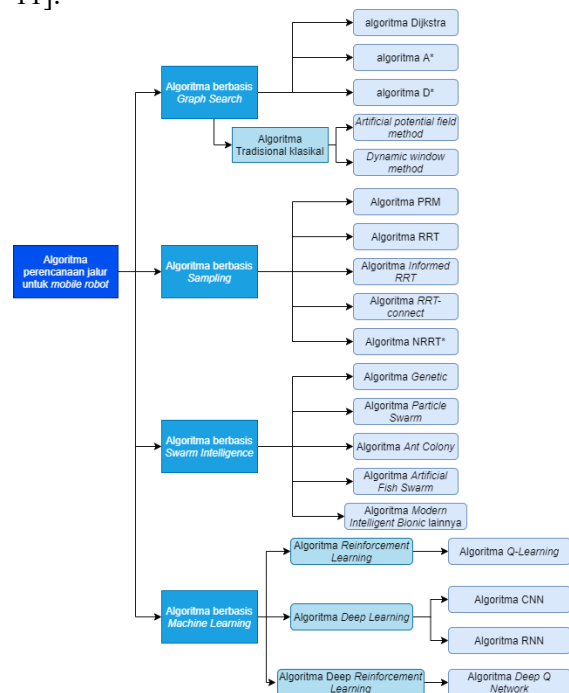
AGV otonom membutuhkan dua kemampuan utama untuk bernavigasi: perencanaan jalur dan penghindaran rintangan [3, 4]. Sensor yang membaca lingkungan sekitarnya membantu AGV otonom bernavigasi dan mencapai tujuan dalam lingkungan yang tidak terstruktur secara real-time [5, 6]. Perencanaan jalur berbasis optimasi memiliki tujuan utama untuk menemukan jalur yang lebih efisien dari awal ke tujuan tanpa menghadapi hambatan dan dalam waktu yang lebih singkat [7, 8]. Ada dua jenis perencanaan jalur: perencanaan jalur global yang didasarkan pada peta lingkungan dan perencanaan jalur lokal yang didasarkan pada pembacaan sensor informasi lingkungan secara real-time yang memungkinkan robot untuk menghindari hambatan [9].

Penelitian ini mengusulkan teknik perencanaan jalur yang menggabungkan perencanaan jalur lokal dan global untuk mencapai hasil yang lebih baik. Ini berbeda dengan sejumlah studi literatur lainnya. Algoritma Rapidly Exploring Random Trees (RRT) akan digunakan untuk menghasilkan jalur bebas tabrakan dengan mengambil sampel acak dari ruang konfigurasi. Di sisi lain, Dynamic Window Approaches (DWA) akan digunakan untuk perencanaan jalur lokal dengan menghitung kecepatan translasi dan rotasi AGV dan mengambil sampel dari ruang kecepatan AGV [10]. Robot Operating System (ROS) digunakan untuk membangun sistem navigasi yang akan digunakan pada AGV. Sistem operasi Linux Ubuntu 18.04 LTS akan menjadi basis ROS.

2. TINJAUAN PUSTAKA

Berdasarkan literatur-literatur yang telah dikaji terdapat beberapa jenis algoritma perencanaan jalur dan penghindaran rintangan pada mobile robot (gambar 1) yaitu Graph Search, A*, Dijkstra, D* Lite, Artificial Potential Field (APF), Rapidly-Exploring Random Trees (RRT), Bug Algorithm, Vector Field Histogram (VFH+), Follow the Gap Method (FGM), Obstacle Restriction Method (ORM), Velocity Obstacles Method (VO),

Dynamic Window Approaches (DWA), dan Curvature Velocity Method (CVM), dll [6, 10, 11].



Gambar 1. Diagram struktur algoritma perencanaan jalur untuk mobile robot.

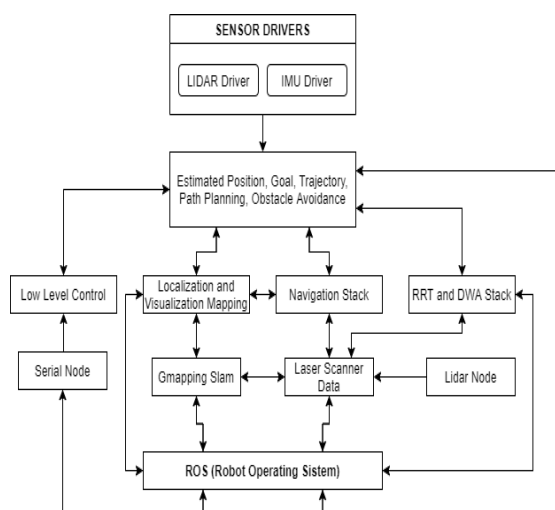
Beberapa literatur tentang perencanaan jalur untuk robot mobil atau AGV otonom dapat dijelaskan sebagai berikut: Algoritma D* Lite yang efektif meningkatkan efisiensi komputasi dalam perencanaan ulang jalur dalam lingkungan yang sangat dinamis [12]; Metode perencanaan jalur global dinamis yang menggabungkan algoritma A* dan Approach Dynamic Window (DWA) berbasis MATLAB [9]; Algoritma D* Lite yang efektif meningkatkan efisiensi komputasi dalam perencanaan ulang jalur dalam lingkungan dinamis yang besar [12]; Sistem perangkat keras dan perangkat lunak terintegrasi yang digunakan untuk menemukan robot otonom yang bergerak di ruang tidak terstruktur dan tidak rata; Metode adaptasi Monte Carlo untuk lokalisasi diri, metode elastic band sebagai perencanaan lokal, dan langkah-langkah variabel dalam perencanaan global diurutkan oleh robot menggunakan pendekatan Rapidly Exploring Random Trees (RRT) [13]; Perencanaan jalur menggunakan algoritma pembelajaran heuristik arah potensial, yang menggunakan arsitektur jaringan saraf yang biasa digunakan dalam segmentasi semantik [14]; Algoritma Self-Adaptive Learning

Particle Swarm Optimization (SLPSO) untuk perencanaan jalur global robot mobil [15]; Sistem navigasi untuk robot mobil otonom yang bergerak di dalam ruangan yang menggunakan Robot Operating System (ROS) [16]; dan Metode Monte Carlo Adaptif pada AGV dengan data LIDAR untuk lokalisasi di dalam ruangan [17].S

3. METODE PENELITIAN

3.1. Algoritma Rapidly-Exploring Random Trees (RRT)

Kinerja kombinasi algoritma RRT dan DWA akan diuji dan diverifikasi menggunakan Robot Operating System (ROS). Algoritma perencanaan jalur dibangun di atas tumpukan RRT dan DWA, yang dapat dilihat pada Gambar 3. Secara umum, perencanaan jalur dianggap sebagai pencarian di ruang konfigurasi T , di mana setiap $X \in T$ menentukan posisi satu atau lebih robot dalam lingkungan. Status robot seperti jarak, waktu, dan energi di T ditentukan oleh metrik ρ . T_{free} dilambangkan sebagai himpunan konfigurasi di mana robot tidak bertabrakan dengan hambatan di ruang bebas, sedangkan T_{obs} dilambangkan jika terjadi tabrakan. Untuk menemukan jalur kontinu dari titik awal ($X_{init} \in T_{free}$) ke titik tujuan ($X_{goal} \in T_{free}$), digunakan algoritma perencanaan jalur dengan satu kali kueri. Sementara algoritma deteksi tabrakan digunakan untuk menentukan apakah X berada di T_{free} atau T_{obs} .



Gambar 2. Diagram Blok ROS navigation stack

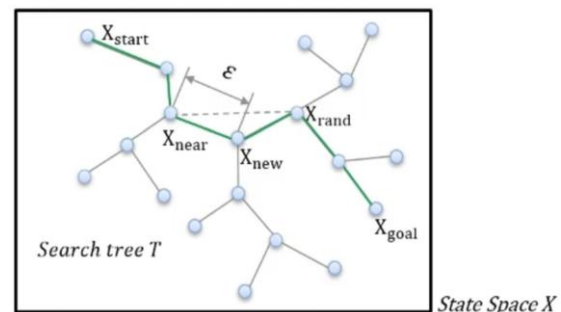
Untuk menangani derajat kebebasan yang tinggi dan batasan nonholonomik, RRT dirancang sebagai struktur data acak dengan kecepatan pencarian yang tinggi. Keunggulan utama RRT adalah ekspansi bias ke komponen konfigurasi ruang yang belum dieksplorasi. Ketika $X \in T_{free}$ dan setiap ujung noda sesuai dengan jalur di T_{free} , sebuah RRT akan terbentuk (gambar 3).

Untuk konfigurasi awal dan tujuan spesifik X_{start} , X (ϵ di sekitar X_{goal}) $\in X_{free}$, sebuah RRT, T , dengan K node yang dibangun seperti yang ditunjukkan dalam Tabel 1:

Tabel 1. Algoritma Rapidly-Exploring Random Trees (RRT)

```

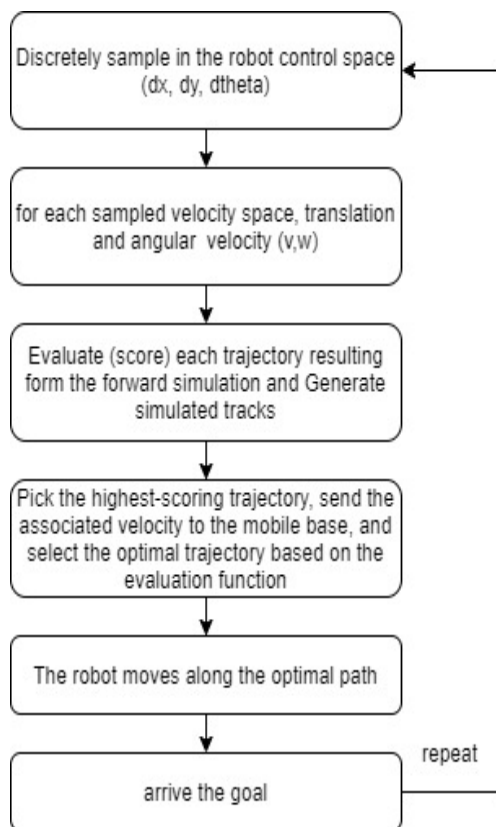
GENERATE_RRT( $X_{start}$ ,  $K$ ,  $\Delta t$ )
1.  $T$ .init( $X_{start}$ );
2. for k=1 to K do
3.    $X_{rand} \leftarrow$  RANDOM_POSITION();
4.    $X_{near} \leftarrow$  NEAREST_NEIGHBOR( $X_{rand}$ ,  $T$ );
5.    $u \leftarrow$  SELECT_INPUT( $X_{rand}$ ,  $X_{near}$ );
6.    $X_{new} \leftarrow$  NEW_POSITION( $X_{near}$ ,  $u$ ,  $\Delta t$ );
7.    $T$ .add_vertex( $X_{new}$ );
8.    $T$ .add_edge( $X_{near}$ ,  $X_{new}$ ,  $u$ );
9.   if  $X_{new} = X$ 
10.    Return Reached
11.  else
12.    Return Reached
13.  Return Trapped
14. Return  $T$ 
    
```



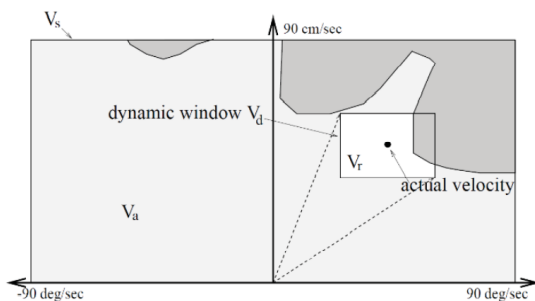
Gambar 3. Pencarian jalur dengan algoritma RRT

3.2. Algoritma Dynamic Window Approach (DWA)

Algoritma perencanaan jalur lokal DWA dapat digunakan untuk kemampuan robot dalam menghindari rintangan secara otonom. Ketika sensor mendeteksi rintangan yang tidak tercantum pada peta, DWA akan diterapkan sebagai metode pencarian kecepatan, di mana pencarian perintah yang mengatur kecepatan rotasi dan translasi robot dilakukan langsung di ruang kecepatan. Ide dasar algoritma DWA ditunjukkan pada Gambar 4.



Gambar 4. Diagram Alir Algoritma DWA Dasar



Gambar 5. Ruang kecepatan robot yang dihasilkan

Kinematika robot dihitung pada algoritma DWA lokal dengan menemukan ruang kecepatan yang tepat. Pada jalur melingkar terdapat pemilihan ruang pencarian kecepatan, yang ditentukan oleh kecepatan translasi dan rotasi (v, ω) (Gambar 4). Area ruang pencarian V_r tumpang tindih dengan rintangan di ruang konfigurasi, seperti yang ditunjukkan pada Gambar 5. Persamaan 1 menunjukkan V_r .

$$V_r = V_s \cap V_a \cap V_d \quad (1)$$

Rentang sampel harus sesuai dengan batasan bingkai robot dan lingkungan berikut: (1) batas kecepatan maksimum dan minimum (V_s), (2) batas percepatan/perlambatan maksimum (V_d),

dan (3) rentang kecepatan robot pada percepatan maksimum dan perlambatan maksimum (V_a) sebelum proses evaluasi (skor) ditunjukkan pada Gambar 4.

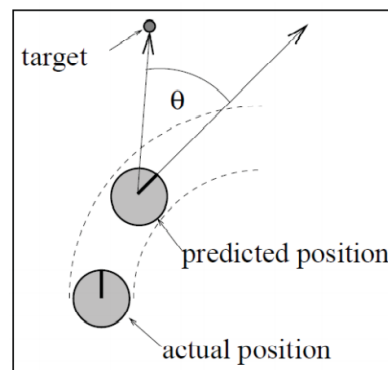
$$V_s = \{(v, \omega) \mid v_{min} \leq v \leq v_{max}, \omega_{min} \leq \omega \leq \omega_{max}\} \quad (2)$$

$$V_a = \{(v, \omega) \mid v \leq \sqrt{2 \cdot dist(v, \omega) \cdot \ddot{v}_b}, \omega \leq \sqrt{2 \cdot dist(v, \omega) \cdot \ddot{\omega}_b}\} \quad (3)$$

$$V_d = \{(v, \omega) \mid v \in [V_a - \dot{v} \cdot t, V_a + \dot{v} \cdot t], \omega \in [\omega_a - \dot{\omega} \cdot t, \omega_a + \dot{\omega} \cdot t]\} \quad (4)$$

Setelah rentang pengambilan sampel untuk kecepatan robot ditentukan, evaluasi atau optimasi trajektori akan dilakukan dengan fungsi objektif yang ditunjukkan dalam persamaan 5.

$$O(v, \omega) = a \cdot heading(v, \omega) + b \cdot velocity(v, \omega) + c \cdot dist(v, \omega) \quad (5)$$



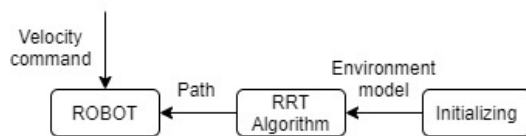
Gambar 6. Sudut θ dari target

Pada Persamaan 5, terdapat fungsi heading, kecepatan, dan jarak. Fungsi $heading(v, \omega)$ mengevaluasi perbedaan antara sudut posisi target dan sudut posisi yang diprediksi. Metrik dari fungsi tersebut adalah θ (Gambar 6). Fungsi $velocity(v, \omega)$ menghitung nilai kecepatan linier dalam kecepatan translasi dan rotasi. Sementara fungsi $dist(v, \omega)$ menunjukkan jarak robot ke rintangan terdekat yang memotong jalur robot. Nilai dari fungsi ini diatur ke konstanta terbesar jika tidak ada persimpangan dengan jalur robot. Konstanta a, b, c diperoleh dengan melakukan operasi pemulusan yang berfungsi untuk menormalkan tiga fungsi objektif ke $[0, 1]$. Ini dilakukan untuk mengatasi situasi di mana ada bagian yang terputus dari beberapa komponen fungsi objektif, sehingga membuat beberapa

komponen mengambil peran utama.

3.3. Metode yang Diusulkan

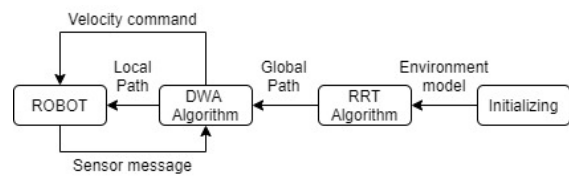
Menggabungkan algoritma RRT dengan algoritma DWA dilakukan dengan membagi dimana algoritma RRT digunakan untuk perencanaan jalur global dan algoritma DWA digunakan untuk perencanaan jalur lokal. RRT dilakukan dengan memilih wilayah target dari robot berbasis sampel yang dapat mencari ruang berdimensi tinggi dengan cepat. Pencarian ini didasarkan pada kendala aljabar dan diferensial. Kemampuan RRT untuk mengeksplorasi seluruh ruang konfigurasi dapat dimanfaatkan untuk perencanaan jalur global. Gambar 7 menunjukkan konfigurasi sistem yang disederhanakan. Blok inisialisasi menyediakan informasi seperti: posisi rintangan, posisi awal dan tujuan robot, bersama dengan jejak robot. Informasi tersebut diperlukan untuk perencanaan rute RRT. Setelah inisialisasi, rencana jalur RRT akan menawarkan robot jalur tanpa melewati rintangan. Perintah kecepatan diberikan agar robot dapat mengikuti trayektori dengan lancar. Ada beberapa masalah yang harus diselesaikan oleh algoritma RRT seperti memastikan nilai kecepatan, memastikan robot mengikuti jalur dengan benar, memastikan koneksi antarmuka dengan pesan sensor robot, dll. Ini adalah poin kunci bahwa pada saat yang sama memberikan perintah kecepatan yang ditentukan oleh rintangan dan jalur, perencana RRT tidak dapat memberikan jalur bebas tabrakan.



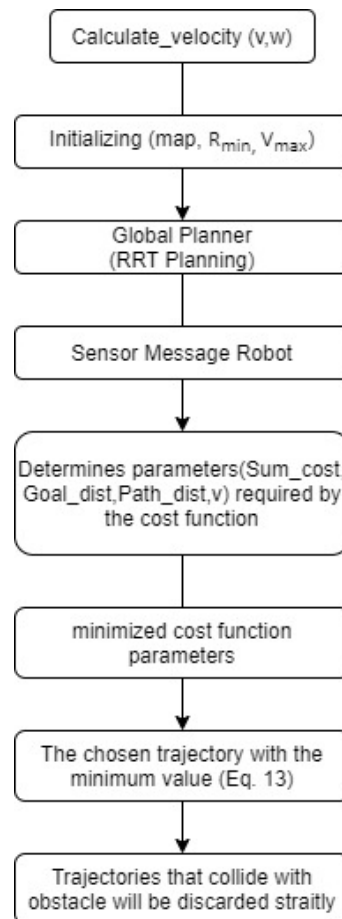
Gambar 7. Perencanaan Jalur RRT Dasar

Penambahan algoritma DWA setelah algoritma RRT (Gambar 8) menunjukkan bahwa inisialisasi tidak identik dengan parameter dinamis robot seperti menggabungkan radius rotasi minimum roda robot dengan kecepatan maksimum robot. Perencanaan RRT akan mencari jalur fleksibel dari posisi awal robot ke tujuannya, kemudian jalur tersebut dikirim ke perencanaan DWA. Secara bersamaan, pesan sensor robot dikirimkan dan perencanaan DWA akan menghitung kecepatan translasi dan rotasi

terbaik untuk robot.



Gambar 8. Perencanaan jalur dengan RRT dan DWA



Gambar 9. Diagram Alir Algoritma RRT dan DWA

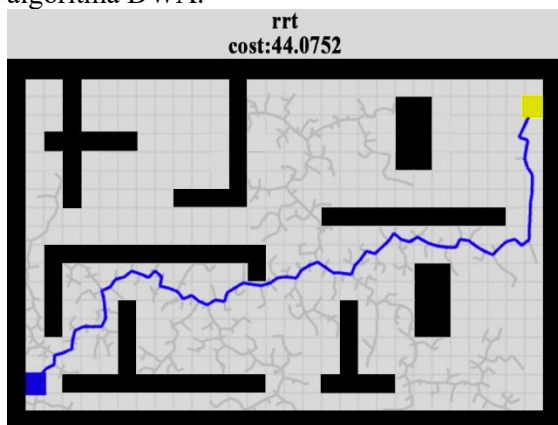
DWA mengambil sampel ruang kecepatan secara langsung dan mempertimbangkan kendala dinamis (Gambar 9). Langkah pertama untuk robot adalah menginisialisasi perencanaan dengan parameter dinamika dan peta. Setelah itu, terima jalur global yang telah disediakan oleh perencana RRT dan pesan sensor robot. Untuk setiap kecepatan sampel, simulasi maju dari kondisi robot saat ini dilakukan untuk memprediksi apa yang akan terjadi jika laju sampel diterapkan dalam waktu singkat. Jalur sampel yang berpotongan dengan rintangan akan dihilangkan secara langsung. Jadi, fungsi objektif diganti dengan fungsi biaya:

$$C(t) = a.sum_cost + b.goal_dist + c.path_dist + \sigma \cdot \frac{1}{v^2} \quad (6)$$

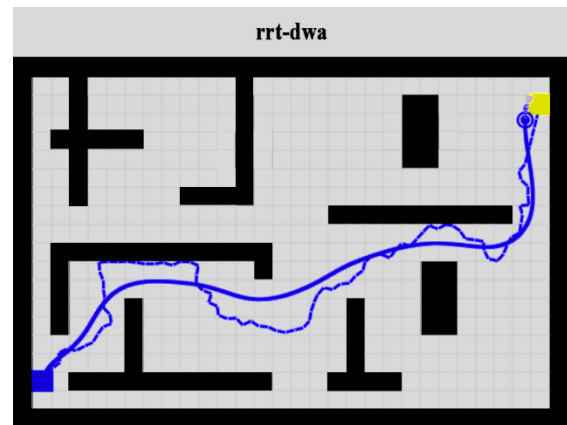
Di mana *sum_cost* adalah jumlah biaya yang dilalui oleh jalur robot, *goal_dist* adalah jarak terpendek dari titik akhir jalur ke tujuan, *path_dist* adalah jarak terpendek dari titik akhir jalur ke jalur global, dan *v* adalah komponen translasi dari perintah kecepatan yang menghasilkan jalur. Persamaan 6 menghasilkan beberapa sifat, yaitu: rute yang aman dan jauh dari rintangan (a), bias ke tujuan (b), tetap dekat dengan jalur optimal (c) dan memiliki kecepatan tinggi (d). Setelah semua parameter terpenuhi, jalur yang bertabrakan dengan rintangan akan segera dibuang.

4. HASIL DAN PEMBAHASAN

Algoritma RRT dan DWA disimulasikan terpisah terlebih dahulu pada aplikasi MATLAB. Gambar 10 merupakan algoritma perencanaan jalur global RRT yang disimulasikan dengan jumlah maksimum titik sampel sebanyak 10000 dan jarak ekspansi maksimum satu Langkah sebesar 0.5. Dari simulasi yang dilakukan didapatkan besar nilai fungsi biaya sebesar 44.0752. Sedangkan algoritma perencanaan jalur lokal DWA (gambar 11) menghasilkan jalur yang lebih efisien dan smooth dikarenakan adanya pengendalian kinematic dari robot oleh algoritma DWA.

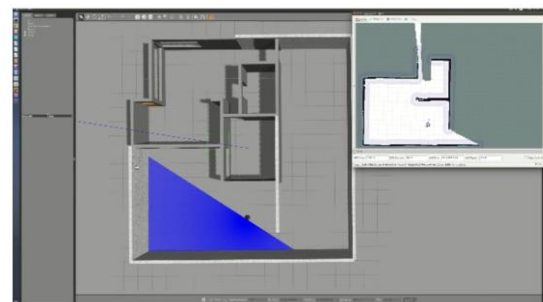


Gambar 10. Algoritma perencanaan jalur global RRT

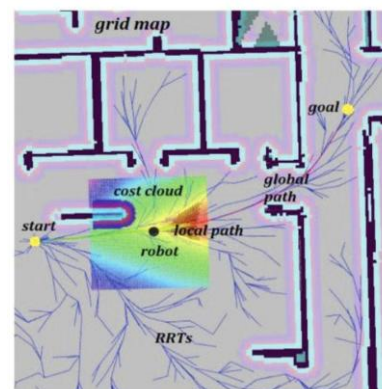


Gambar 11. Algoritma perencanaan jalur global RRT dan local DWA

Algoritma yang telah dirancang akan disimulasikan di Rviz dan Gazebo (Gambar 12). Perencanaan jalur dilakukan dalam paket *move_base*. ROS Navigation Stack dibagi menjadi modul perencanaan jalur global dan modul perencanaan jalur lokal. Paket *move_base* juga memelihara dua *costmap* yaitu *global_costmap* dan *local_costmap*. *Costmap* digunakan untuk menyimpan dan memelihara informasi dalam bentuk grid okupansi tentang rintangan di lingkungan dan di mana robot harus bernavigasi. *Costmap* diinisialisasi dengan peta statis sebelumnya, kemudian akan diperbarui menggunakan data sensor tentang informasi rintangan yang ada di lingkungan.



Gambar 12. Gazebo dan Rviz



Gambar 13. Perencanaan jalur RRT-DWA

Pada Gambar 13, setelah menentukan konfigurasi awal dan tujuan (titik kuning), algoritma RRT mencari ruang konfigurasi (garis biru), sehingga jalur konfigurasi dapat ditemukan (garis merah), kemudian DWA menghitung jalur fleksibel (garis hijau) untuk mengikuti garis merah menghindari rintangan. Tugas berakhir ketika robot mencapai titik tujuan. Berdasarkan analisis, RRT dapat menyediakan jalur yang efisien dalam skala besar dan DWA dapat langsung memberikan perintah kecepatan yang tepat untuk robot.

5. KESIMPULAN

Dalam penelitian ini, algoritma yang kami rancang adalah kombinasi metode perencanaan jalur global dengan perencanaan jalur lokal untuk menghasilkan jalur bebas tabrakan secara efisien dan pada saat yang sama perintah dikirim ke robot dengan mengasosiasikan pesan sensor pada robot. Kami mendefinisikan algoritma RRT sebagai perencana jalur global yang menghasilkan jalur referensial global untuk memberikan informasi yang lebih spesifik. Kemudian algoritma DWA akan menghitung kecepatan translasi dan kecepatan rotasi robot.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] Mat, N.A.C., N.M.M. Noor, and R. Mohamad, Smart Integrated Partner Selection System (SPISS): A Decision Support System for Partner Selection in the Era of Industry Revolution 4.0, in 2020 IEEE International Conference on Semiconductor Electronics (ICSE). 2020: Kuala Lumpur, Malaysia, Malaysia p. 189-192.
- [2] Geesta, M.v., B. Tekinerdogana, and C. Catalb, Design of a Reference Architecture for Developing Smart Warehouses in Industry 4.0. *Computers in Industry*, 2020. 124: p. 103343.
- [3] Xia, C., Intelligent Mobile Robot Learning in Autonomous Navigation, in *Automatic Control Engineering*. 2015, Beihang University: Ecole Centrale de Lille.
- [4] Siegwart, R., I.R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, ed. S. Edition. 2011, Cambridge, Massachusetts: Massachusetts Institute of Technology.
- [5] Wang, S., K. Wang, and H. He, An autonomous navigation method for mobile robot based on ROS. *Proceedings of the 2nd WRC Symposium on Advanced Robotics and Automation 2019*, 2019: p. 284-290.
- [6] Gómez, E.Z., *Map-Building and Planning For Autonomous Navigation of a Mobile Robot*, in *Department of Automatic Control*. 2015, National Polytechnic Institute: Mexico.
- [7] Yildirim, M.Y. and R. Akay, A Comparative Study of Optimization Algorithms for Global Path Planning of Mobile Robots. *Sakarya University Journal of Science*, 2021. 25(2): p. 417-428.
- [8] Faizal, F.A.F., et al., Rancang Bangun Sistem Pemetaan Dan Lokalisasi Berbasis Algoritma Slam Menggunakan Depth Sensor Kinect Pada Mobile Robot. *Jurnal Informatika dan Teknik Elektro Terapan*, 2025. 13(3S1).
- [9] Zhu, Z., J. Xie†, and Z. Wang, Global Dynamic Path Planning Based on Fusion of A* Algorithm and Dynamic Window Approach. *IEEE Access*, 2019: p. 5572-5576.
- [10] Wang, J., et al., Path Planning Combining Improved Rapidly-Exploring Random Trees with Dynamic Window Approach in ROS. *13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2018: p. 1296-1301. [5] Wang, S., K. Wang, and H. He, an autonomous navigation method for mobile robot based on ROS. *Proceedings of the 2nd WRC Symposium on Advanced Robotics and Automation 2019*, 2019: p. 284-290.
- [11] Baird, R., An Autonomous Forklift Research Platform for Warehouse Operations, in *Department of Electrical Engineering and Computer Science*. 2018, Massachusetts Institute of Technology
- [12] Wang, J., M.A. Garratt, and S.G. Anavatti, Real-time path planning algorithm for autonomous vehicles in unknown environments. *International Journal of Mechatronics and Automation*, 2017. 6(1): p. 1.
- [13] Wang, C., et al., Autonomous Mobile Robot Navigation in Uneven and Unstructured Indoor Environments. 2017.
- [14] Takahashi, T., et al., Learning Heuristic Functions for Mobile Robot Path Planning Using Deep Neural Networks. *Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling 2019*: p. 764-772.
- [15] Li, G. and W. Chou, Path planning for mobile robot using self-adaptive learning particle

- swarm optimization. Science China Information Sciences, 2017. 61(5): p. 1-18.
- [16] Erlangga, D., et al., Sistem Navigasi Mobile Robot dalam Ruang Berbasis Autonomous Navigation. Journal of Mechanical Engineering and Mechatronics 2019, 2019. 4(2): p. 78-86.
- [17] Yilmaz, A. and H. Temeltas, Self-Adaptive Monte Carlo Method for Indoor Localization of Smart AGVs Using LIDAR Data. Robotics and Autonomous Systems, 2019. 122: p. 103285.