

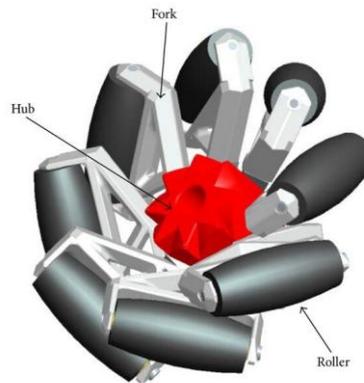
BAB I

PENDAHULUAN

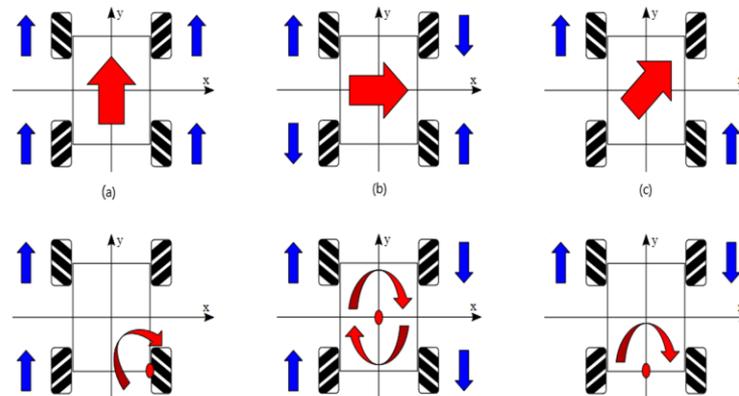
1.1 Latar Belakang

Teknologi robotika memegang peranan penting dan telah diterapkan secara luas di berbagai bidang, mulai dari industri manufaktur, logistik, hingga aplikasi di bidang medis. Perkembangan teknologi di bidang ini telah memberikan dampak yang signifikan terhadap sektor industri, khususnya dalam peningkatan produktivitas dan efisiensi kerja. Penggunaan robot tidak hanya memungkinkan penyelesaian tugas secara lebih cepat dan akurat, tetapi juga memungkinkan operasi di lingkungan yang berisiko tinggi bagi manusia, seperti pabrik dengan suhu ekstrem, area dengan paparan radiasi tinggi, atau lokasi berbahaya lainnya. Salah satu jenis robot yang sering digunakan dalam kondisi tersebut adalah *mobile robot*. Implementasi sistem penggerak pada *mobile robot* memungkinkan robot untuk bergerak secara otonom, tanpa intervensi manusia secara langsung [1, 2, 3, 4]. Kemampuan untuk dapat bergerak secara otonom, fleksibel, dan adaptif pada lingkungan yang dinamis menjadikan *mobile robot* sebagai solusi ideal pada berbagai pengaplikasian, seperti transportasi barang, inspeksi hingga penyelamatan [5, 6, 7].

Pada skala industri, *mobile robot* banyak digunakan untuk menggantikan manusia dalam hal mengangkut barang agar dapat mengurangi terjadinya kecelakaan, akibat kesalahan manusia. Salah satu jenis *mobile robot* yang digunakan adalah *Mecanum Wheel Mobile robot (MWMR)*. Pada roda mecanum terdiri dari 3 bagian yaitu; roda primer (*fork* dan *hub*), dan *roller* yang ditempatkan pada sudut sebesar 45° terhadap roda primer [4, 8, 9], ilustrasi komponen roda *mecanum* dapat dilihat pada Gambar 1.1.



Gambar 1.1. Komponen pada *mecanum wheel*.¹



Gambar 1.2. Ilustrasi arah putar roda terhadap pergerakan MWMR[10]

Penggunaan roda *mecanum* pada *mobile robot* memungkinkan robot untuk dapat melakukan pergerakan *omnidirectional*. Gerakan tersebut mengacu pada kemampuan robot untuk bergerak ke segala arah tanpa perlu mengubah orientasinya terlebih dahulu [6, 11]. Gambar 1.2 menunjukkan ilustrasi arah putar masing-masing roda yang menghasilkan beragam jenis pergerakan pada MWMR, seperti gerakan diagonal, tegak lurus (maju dan mundur), menyamping (kanan dan kiri), serta rotasi (berputar di tempat). Kemampuan tersebut menjadikan MWMR sangat ideal untuk digunakan di lingkungan yang dinamis, sempit, dan kompleks, seperti gudang, ruang operasi, dan restoran, di mana manuver yang presisi dan

¹ <https://onlinelibrary.wiley.com/doi/10.1155/2016/5892784>

fleksibel termasuk gerakan vertikal-horizontal, rotasi, dan diagonal sangat dibutuhkan [12].

Salah satu aspek yang paling menantang dalam pengembangan *mobile robot* agar dapat bergerak secara otonom adalah kemampuannya dalam melakukan navigasi dan lokalisasi. Untuk mendukung kemampuan tersebut, robot memerlukan sistem perencanaan jalur (*path planning*) yang efektif dan sensor untuk mendeteksi objek di sekitar robot. Jika terdapat rintangan robot akan menuju rute baru yang dapat dilewatinya.

Sistem *path planning* merupakan proses menentukan jalur atau rute yang aman dilalui dan optimal untuk robot bergerak dari titik awal (*start point*) menuju titik tujuan (*end point*) sambil menghindari rintangan [2]. Salah satu tantangan utama dalam *path planning* adalah menemukan jalur yang optimal dari segi waktu, jarak dan biaya (*cost*), dengan memperhitungkan apakah jalur tersebut aman dan dapat dilalui oleh robot. Umumnya, metode *path planning* terbagi menjadi dua, yaitu: *global path planning* dan *local path planning* [13]. *Global path planning* menggunakan informasi pemetaan lingkungan telah diketahui sebelumnya (*statis*), sedangkan *local path planning* menggunakan informasi yang diterima robot secara *real time*. Dengan algoritma *local path planning*, yaitu metode perencanaan jalur yang memungkinkan robot menyesuaikan pergerakannya secara *real-time* berdasarkan kondisi lingkungan sekitar, robot dapat menentukan jalurnya saat mendeteksi adanya rintangan [13, 14, 15, 16].

Pemilihan algoritma yang tepat untuk digunakan memiliki pengaruh terhadap kecepatan dan efisiensi robot dalam menemukan rute tercepat dan terpendek. Untuk mencari rute/ jalur terpendek, terdapat beberapa metode algoritma yang dapat digunakan, contohnya adalah algoritma *A** (*A-star*). Algoritma *A** menggabungkan keunggulan dari algoritma Dijkstra dengan memanfaatkan fungsi heuristik untuk mempercepat pencarian jalur yang optimal². Pemilihan fungsi

² <https://ieeexplore.ieee.org/document/7979125>

heuristik yang tepat memiliki pengaruh terhadap hasil *path planning*, termasuk jarak tempuh dan jalur yang dihasilkan[17].

Pada dasarnya, algoritma *path planning* dapat diklasifikasikan menjadi beberapa kelompok utama. Pertama, terdapat algoritma *path planning* berbasis *graph search*, yang mencakup metode klasik seperti Dijkstra dan A*. Metode ini melakukan pencarian jalur dari titik awal hingga titik tujuan dengan menelusuri seluruh ruang pencarian. Sementara algoritma Dijkstra menerapkan pendekatan pencarian secara menyeluruh tanpa menggunakan heuristik, algoritma A* memanfaatkan fungsi heuristik guna mencari jalur yang optimal. Pendekatan heuristik sendiri bertujuan untuk menemukan jalur yang lebih optimal dari titik awal menuju titik tujuan dengan mempertimbangkan berbagai parameter, seperti jarak tempuh, kecepatan pergerakan, dan efisiensi penggunaan energi, sehingga proses pencarian jalur menjadi lebih efisien. Kedua, terdapat algoritma perencanaan jalur cerdas yang terinspirasi dari bionik, termasuk algoritma PSO, algoritma genetika, dan pembelajaran penguatan. Ketiga, algoritma perencanaan jalur berbasis *sampling*, seperti RRT [18].

Salah satu pendekatan yang banyak digunakan dalam penerapan algoritma *path planning* adalah metode *grid-based*. Pada metode ini, lingkungan dibagi menjadi sel-sel berbentuk *grid* yang merepresentasikan area yang dapat atau tidak dapat dilalui oleh robot. Kelebihan dari pendekatan *grid-based* adalah kesederhanaan implementasi, kemudahan integrasi dengan algoritma pencarian jalur seperti A*, serta efisiensinya dalam lingkungan terstruktur. Selain itu, representasi *grid* sangat cocok untuk digunakan dalam simulasi komputer dan pemetaan lingkungan statis. *Grid-based* peta memungkinkan penyesuaian resolusi peta sesuai dengan kebutuhan, sehingga dapat meningkatkan akurasi perencanaan jalur. Meskipun metode ini memerlukan penyimpanan memori yang lebih besar untuk *grid* beresolusi tinggi, kejelasan struktur dan kestabilan algoritma menjadikannya pilihan yang sangat sesuai untuk keperluan simulasi *path planning* [13, 17].

Pada skripsi ini, algoritma yang diimplementasikan pada MWMM adalah Algoritma *path planning A-star (A*)*. Algoritma *A** telah banyak digunakan dalam penelitian karena kekuatan fungsi heuristiknya dalam menemukan dan menentukan jalur yang optimal dibandingkan dengan metode *classical path planning* [19, 20]. Algoritma *A** menggunakan *cost function* untuk mengevaluasi *potential path* agar dapat menentukan jalur yang optimal berdasarkan bobot dari keseluruhan *path* [21].

Untuk mempermudah pengujian dan analisis hasil implementasi *path planning* algoritma *A** pada MWMM, simulasi menjadi alat yang penting. Perangkat lunak CoppeliaSim merupakan aplikasi simulasi robotika yang banyak digunakan karena fleksibilitasnya serta kemampuannya dalam mensimulasikan berbagai jenis robot dan lingkungan [22, 23].

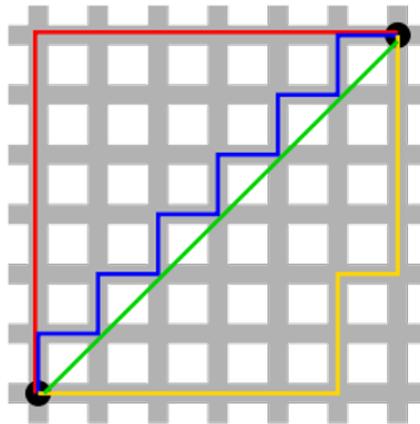
Implementasi algoritma *path planning* pada simulasi MWMM yang dilakukan menggunakan 2 pendekatan heuristik untuk perhitungan jarak, yaitu:

1. *Manhattan distance*: digunakan pada lingkungan *grid based*, di mana hasil pergerakan menggunakan perhitungan tersebut hanya dapat dilakukan secara vertikal dan horizontal.
2. *Euclidean distance*: digunakan untuk memperkirakan jarak lurus antar titik dalam lingkungan yang memperbolehkan pergerakan diagonal

Pemilihan kedua jenis heuristik ini bertujuan untuk mengevaluasi pengaruh fungsi heuristik terhadap kinerja algoritma *A**, khususnya dalam hal efisiensi pencarian jalur dan kualitas lintasan yang dihasilkan. *Manhattan distance* umumnya digunakan pada lingkungan *grid* di mana robot hanya dapat bergerak dalam arah tegak lurus, sehingga cocok untuk simulasi dengan rintangan yang tersusun secara ortogonal. Sedangkan, *Euclidean distance* memberikan estimasi jarak yang lebih realistis dan presisi dalam lingkungan yang memungkinkan gerakan diagonal, yang lebih sesuai dengan karakteristik pergerakan robot beroda *mecanum* yang memiliki kemampuan *omnidirectional*. Dengan membandingkan hasil dari kedua heuristik ini, diharapkan dapat diperoleh pemahaman yang lebih baik mengenai efektivitas dan efisiensi masing-masing pendekatan dalam mendukung navigasi robot secara otonom pada berbagai skenario lingkungan.

Perbandingan ini juga dapat membantu dalam menentukan pendekatan heuristik yang paling optimal untuk diterapkan dalam implementasi nyata, terutama dalam konteks lingkungan dinamis yang kompleks dan menuntut respons navigasi yang cepat dan adaptif.

Hasil dari penggunaan masing-masing heuristik pada algoritma *path planning* dapat dilihat pada Gambar 1.3.



Gambar 1.3. Representasi perbandingan jalur yang dihasilkan *Manhattan distance* (Merah, Biru, dan Kuning) dan *Euclidean distance* (Hijau).³

1.2 Identifikasi Masalah

Agar dapat bergerak secara otonom, *mobile robot* perlu dapat melakukan navigasi dan lokalisasi. Adapun identifikasi masalah pada latar belakang yang telah dituliskan sebagai berikut:

1. Pada umumnya *mobile robot* masih menggunakan metode *classical* untuk melakukan navigasi
2. Membandingkan metode A^* yang optimal dari segi waktu dan jarak.
3. Penggerakan *mobile robot* yang terbatas jika menggunakan roda konvensional.

³ <https://www.diva-portal.org/smash/get/diva2:918778/FULLTEXT02.pdf>

1.3 Konsep Sistem

Produk *mobile robot* menggunakan roda *mecanum* untuk memungkinkan pergerakan *omnidirectional*, sehingga robot dapat bergerak dengan fleksibel dalam lingkungan yang dinamis dan sempit. Untuk mendukung kemampuan tersebut, *Mecanum Wheel Mobile robot* membutuhkan sistem penggerak dan navigasi seperti sistem *path planning* dan *obstacle avoidance* agar dapat secara otonom. Pengujian pergerakan *mecanum wheel* dan implementasi *path planning* dilakukan pada robot simulator CoppeliaSim. Sistem *local path planning* pada MWMR menggunakan sensor *proximity* untuk *obstacle avoidance*. Sementara itu, sistem *global path planning* yang diimplementasikan pada MWMR adalah algoritma *path planning A** dengan menggunakan dua heuristik yang berbeda yaitu: *Manhattan distance* dan *Euclidean distance* dengan tujuan untuk membandingkan jalur yang dihasilkan dari penggunaan masing-masing heuristik tersebut.

1.4 Batasan Sistem

Terdapat beberapa batasan sistem yang perlu ditetapkan dalam proses simulasi MWMR dengan algoritma *path planning A** pada CoppeliaSim. Pada permukaan yang licin, robot akan lebih mudah tergelincir atau *slip* yang dapat menyebabkan orientasi sudut robot berubah. Kedua robot akan kesulitan saat meliwati permukaan yang tidak rata. Maka dari itu, MWMR hanya dapat di aplikasikan pada permukaan yang rata dan bebas dari rintangan kecil yang tidak mampu dideteksi oleh sensor ultrasonik.

1.5 Fungsi dan Manfaat Sistem

Penggunaan *mecanum wheel* pada *mobile robot* bertujuan agar robot memiliki kemampuan bermanuver yang fleksibel, sehingga robot dapat bergerak bebas di lingkungan dinamis, seperti pada area restoran maupun gudang. Untuk meminimalkan titik buta selama pergerakan, sensor ultrasonik dipasang pada sisi tegak lurus dan sisi diagonal robot. Dengan konfigurasi ini, robot mampu mendeteksi keberadaan rintangan di sekitarnya secara lebih efektif, sehingga dapat melakukan *obstacle avoiding* dan melanjutkan pergerakan menuju koordinat target berikutnya. Sementara itu, algoritma *path planning A** digunakan untuk

menentukan rute atau lintasan optimal yang harus dilalui oleh robot berdasarkan kondisi lingkungan yang telah diketahui sebelumnya.