

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi robotika telah mendorong terciptanya *mobile robot* atau robot bergerak yang mampu beroperasi secara otonom untuk menyelesaikan berbagai tugas di lingkungan nyata [1],[2]. Robot otonom dirancang agar dapat mendeteksi kondisi sekitar, merencanakan lintasan, dan bergerak menuju tujuan tanpa intervensi manusia secara langsung, sehingga banyak diterapkan di bidang logistik, manufaktur, dan layanan publik [3],[4]. Salah satu kemampuan mendasar yang harus dimiliki robot otonom adalah sistem navigasi yang andal, yaitu kemampuan bergerak dari titik awal ke titik tujuan secara efisien sambil menghindari rintangan [5].

Namun, efisiensi pergerakan robot seringkali terkendala oleh batasan mekanis sistem penggerak. Robot beroda konvensional seperti *differential drive* memiliki keterbatasan non-holonomik yang membuatnya tidak dapat bergerak langsung ke arah lateral (samping) [6]. Akibatnya, manuver seperti berbelok atau menghindari rintangan memerlukan ruang lebih luas dan pola gerak memutar yang kompleks [7]. Untuk mengatasi keterbatasan ini, penggunaan roda mekanum menjadi solusi yang efektif [8].

Roda mekanum memungkinkan robot bergerak secara omnidireksional, yaitu dapat bergerak maju, mundur, ke samping, serta berputar di tempat tanpa mengubah orientasi bodi secara signifikan [9]. Desain unik berupa rol miring 45° di sekeliling roda memberi robot tiga derajat kebebasan (DoF) pada bidang datar: translasi pada sumbu-X, sumbu-Y, dan rotasi sumbu-Z [10]. Kemampuan gerak holonomik ini menjadikan robot beroda mekanum sangat cocok digunakan di area terbatas dan sempit seperti gudang atau laboratorium [11]-[13]. Fleksibilitas tersebut juga mempermudah robot menghindari rintangan tak terduga secara cepat dan efisien [14].

Dari segi distribusi beban, roda mekanum memiliki permukaan kontak yang lebih lebar, membuatnya lebih stabil saat membawa muatan atau melintasi permukaan lantai yang tidak

rata. Sementara itu, omniwheel cenderung lebih licin karena rol-rolnya hanya mendukung arah gerak tertentu secara dominan, yang meningkatkan risiko slip saat robot harus melakukan manuver mendadak. Berdasarkan hasil tersebut, dipilih roda mecanum untuk memastikan robot dapat bergerak fleksibel dan responsif, tetapi tetap mempertahankan kestabilan dan akurasi posisi saat menghadapi kondisi di lingkungan nyata seperti gudang atau laboratorium.

Selain keunggulan mekanis, sistem navigasi robot juga harus didukung oleh perencanaan lintasan (*path planning*). Secara umum, lingkungan operasi robot dapat bersifat statis dimana posisi rintangan tetap [15],[16]. Pada penelitian ini, robot dirancang untuk menghadapi kondisi tersebut melalui integrasi algoritma perencanaan lintasan lokal dan *obstacle avoidance* lokal [17]. Lingkungan uji berbentuk peta *grid* yang didesain manual dan diobservasi dengan sensor LIDAR (*light detection and ranging*) [18]. Ketika robot bergerak, sensor ultrasonik dimanfaatkan untuk mendeteksi rintangan mendadak yang tidak ada pada peta awal [19].

Pada prinsipnya, algoritma perencanaan jalur (*path planning*) dapat dibagi ke dalam beberapa kategori utama. Yang pertama adalah algoritma berbasis pencarian graf (*graph search*), yang mencakup metode klasik seperti Dijkstra dan A*. Algoritma jenis ini bekerja dengan menelusuri ruang pencarian dari titik awal hingga titik tujuan. Dijkstra melakukan pencarian secara menyeluruh tanpa menggunakan informasi tambahan, sedangkan A* menggabungkan biaya aktual dan nilai heuristik untuk mempercepat pencarian jalur optimal. Heuristik ini membantu algoritma lebih fokus ke arah tujuan dengan mempertimbangkan faktor-faktor seperti jarak, waktu tempuh, dan efisiensi energi, sehingga proses pencarian menjadi lebih efisien. terdapat juga pendekatan berbasis sampling, seperti *Rapidly-Exploring Random Tree* (RRT), yang banyak digunakan pada robotika untuk menemukan jalur di ruang berdimensi tinggi yang membutuhkan jumlah parameter atau variable yang perlu dipertimbangkan sangat banyak seperti robot manipulator, dan humanoid.

Sebagai gambaran lebih jelas, berikut adalah tabel perbandingan dari tiga algoritma yang sudah di jelaskan:

Tabel 1.1 - Perbandingan Antara Algoritma Perencanaan Lintasan

Algoritma	Prinsip Kerja	Kelebihan	Kekurangan
A*	Fungsi pencarian seperti dijkstra tapi menggunakan perkiraan tujuan sehingga lebih efisien	Pencarian lebih efisien dan cepat serta mengurangi perhitungan berulang	Jalur cenderung dekat dengan rintangan tidak memperhatikan jarak aman
Dijkstra	Melakukan pencarian secara menyeluruh tanpa menggunakan perkiraan tujuan	Menemukan jalur terpendek	Terlalu banyak node yang diperiksa, sehingga lambat
RRT	Membangun pohon-pohon(tree) dari titik awal yang menyebar ke segala arah, sehingga node baru diperoleh dari sampling acak	Cepat di ruang kompleks karena acak	Kurang efisien karena menambahkan node yang tidak penting dan biaya jalur lebih tinggi dibandingkan algoritma A* dan dijkstra

Dengan Tabel 1.1 perbandingan perbandingan algoritma perencanaan lintasan, penelitian ini mengimplementasikan algoritma A* (*A-Star*). Algoritma A* dikenal luas dalam perencanaan lintasan robot karena kemampuannya mencari jalur terpendek di peta *grid* dengan efisien karena menggunakan fungsi heuristiknya dalam menemukan dan menentukan jalur yang optimal, menggunakan metode evaluasi berbasis jarak tempuh dan perkiraan sisa jarak ke tujuan [20]. Dengan mempertimbangkan biaya perjalanan sejauh ini dan estimasi ke titik akhir, A* memilih jalur yang paling menjanjikan hingga tujuan tercapai [21],[22]. Implementasi algoritma ini dilakukan pada peta *grid* manual sehingga jalur yang dihasilkan dapat dikonversi menjadi perintah gerakan dasar robot seperti maju, mundur, atau belok [23].

Agar implementasi algoritma dapat dikembangkan dan diuji lebih cepat serta aman, penelitian ini memanfaatkan pustaka *Pygame* sebagai alat bantu visualisasi berbasis Python [24]. Visualisasi ini tidak sekadar berfungsi sebagai simulasi, tetapi juga membantu

memvalidasi jalur hasil perhitungan A* secara kualitatif sebelum dijalankan oleh robot fisik. Dengan menampilkan peta *grid*, posisi robot, letak rintangan, dan jalur lintasan secara *real-time*, pengembangan dan *debugging* algoritma menjadi lebih efisien [25].

Visualisasi berbasis library *Pygame* ini meminimalkan risiko kesalahan pergerakan robot saat pengujian di arena sebenarnya, serta mempermudah dokumentasi lintasan yang telah ditempuh. Dengan pendekatan ini, algoritma A* yang telah terverifikasi secara visual dapat diterapkan langsung pada robot beroda mekanum untuk bergerak dari start menuju goal di lingkungan sesungguhnya.

1.2 Identifikasi Masalah

Berdasarkan informasi yang telah dinyatakan pada latar belakang, maka 3 masalah yang dapat diidentifikasi:

1. Keterbatasan manuver robot dengan roda konvensional pada umumnya menyebabkan kesulitan dalam navigasi terutama di lingkungan sempit. Bagaimana mengatasi keterbatasan tersebut sehingga robot dapat bergerak lincah di ruang kerja
2. Rintangan di tempat kerja robot dapat bergerak atau muncul tiba-tiba. Bagaimana membuat robot yang dapat membaca lingkungan dan merespon secara adaptif secara *real-time*
3. Robot harus dapat bergerak sesuai jalur yang paling layak dan aman. Bagaimana merencanakan navigasi dan mengoreksi pergerakan robot selama bernavigasi akibat ketidakidealan robot maupun tempat bergerak robot sehingga robot tetap dapat sampai titik tujuan sesuai misi navigasi

1.3 Konsep Sistem

Berdasarkan latar belakang dan 3 masalah yang telah diidentifikasi, maka sistem yang dirancang dalam skripsi ini secara umum:

1. Lingkungan kerja robot adalah sebuah arena sebagaimana dijumpai di ruang pergudangan dengan lantai datar dan ada berbagai benda di antara jalur pergerakan orang maupun robot.

2. Robot harus dilengkapi sistem penggerak sehingga mampu bermanuver lincah di ruang yang terbatas, dengan kemampuan gerak dasar maju, mundur, kiri, kanan, serong, dan memutar di tempat.
3. Robot harus dilengkapi sistem sensor yang memudahkan membaca lingkungan sehingga dapat menghindari kerusakan akibat menabrak benda di ruang kerja.
4. Robot harus mampu merencanakan dan mengeksekusi rute pergerakan secara efisien dan terus melakukan pemetaan diri dan lingkungan secara statis dengan memanfaatkan data dari sensor-sensor yang terpasang

Secara singkat, sistem robot beroda mekanum yang dilengkapi dengan dua jenis yaitu sensor LIDAR yang akan mengukur jarak di sekitar robot dalam bidang 2D kemudian data dari pembacaan jarak tersebut akan dikirimkan dari mikrokomputer raspberry pi 4 ke laptop pengguna (*user*) untuk di proses menjadi peta 2D. Hasil pemetaan 2D tersebut akan meverifikasi pergerakan robot dan sensor ultrasonik didesain agar mampu mendeteksi objek sekitar robot sebelum bergerak. Navigasi robot dilakukan dengan pendekatan *local path planning* algoritma A* untuk menghitung lintasan optimal berdasarkan peta lingkungan yang dihasilkan oleh sensor LIDAR. Sistem ini dilengkapi dengan fitur *obstacle avoidance* menggunakan sensor ultrasonik, agar robot setiap saat bisa mendeteksi rintangan mendadak dan menyesuaikan lintasan yang lebih aman saat robot bergerak.

Seluruh proses perencanaan, eksekusi gerak, dan simulasi dilakukan menggunakan *library Pygame* sebagai media visualisasi peta jalur lintasan, posisi robot relatif dalam lingkungannya dan jarak aman selama pengujian algoritma. Agar pembagian tugas robot yang begitu banyak dapat dilakukan secara terstruktur, sistem operasi ROS akan dimanfaatkan memungkinkan sehingga komunikasi antar pc/laptop pengguna dan bagian robot dapat lebih efektif.

1.4 Batasan Sistem

Batasan sistem ditetapkan untuk menjaga fokus cakupan pengembangan dan pengujian dalam ruang lingkup yang terukur. Adapun batasan-batasan sistem pada produk ini meliputi:

1. Area operasional robot sudah ditetapkan sebelumnya yaitu 2 meter x 2 meter dan posisi rintangan juga sudah ditentukan.
2. Sensor ultrasonik yang digunakan hanya memiliki jangkauan sekitar 10 cm untuk mendeteksi rintangan dekat robot.
3. Robot mobile memiliki keterbatasan akurasi dan presisi gerakan akibat potensi slip roda mekanum terhadap lantai, perbedaan kecepatan motor, atau permukaan lintasan. Namun, analisis deviasi lintasan, evaluasi presisi jarak, dan perhitungan error posisi tidak dibahas dalam penelitian ini. Desain dan implementasi robot difokuskan untuk dapat menyelesaikan masalah yang telah diidentifikasi sebelumnya.
4. Peta lingkungan yang digunakan berbentuk *grid* 10x10, setiap *grid* dibuat ukuran 20cm x 20cm dimana 20cm adalah jarak linier satu putaran roda robot. *Grid* disusun secara manual berdasarkan hasil observasi menggunakan sensor LIDAR. Penelitian ini tidak membahas pemetaan otomatis (SLAM) maupun pembentukan peta dinamis, meskipun data LIDAR tetap akan divisualisasikan untuk verifikasi posisi aktual di arena dan posisi virtual robot relatif di layar komputer.

1.5 Fungsi dan Manfaat Sistem

Sistem yang dikembangkan dalam skripsi ini memiliki fungsi utama sebagai sistem navigasi robot beroda mekanum, yang mampu bergerak secara omnidireksional. Kemampuan ini memungkinkan robot untuk bermanuver dengan lebih fleksibel dan efisien, terutama dalam ruang terbatas atau lingkungan yang dinamis.

Fungsi navigasi tersebut didukung oleh algoritma perencanaan lintasan global berbasis A*, yang bekerja dengan memanfaatkan data peta hasil pembacaan sensor LiDAR untuk menentukan jalur optimal dari titik awal ke tujuan. Selain itu, sistem ini dilengkapi dengan fitur pendeteksian dan penghindaran rintangan secara *real-time* menggunakan sensor

ultrasonik, yang memungkinkan robot untuk beradaptasi terhadap perubahan lingkungan secara langsung saat bergerak.

Untuk mendukung pengembangan dan pengujian, sistem juga menyediakan lingkungan simulasi berbasis *Pygame*, yang memungkinkan visualisasi dan evaluasi lintasan secara interaktif tanpa harus langsung diterapkan pada perangkat keras. Dengan fungsi-fungsi tersebut, sistem ini memberikan berbagai manfaat yang signifikan. Di antaranya adalah kelincihan navigasi robot mobile, terutama saat beroperasi di lingkungan indoor yang sering berubah.

Selain itu, sistem ini juga menjadi dasar yang kuat untuk pengembangan lebih lanjut pada sistem robot otonom yang membutuhkan ketepatan dalam navigasi dan kecepatan dalam menghindari rintangan. Keberadaan simulasi berbasis perangkat lunak juga memberi nilai tambah dalam proses *debugging* dan uji coba, karena memungkinkan pengembang untuk menguji berbagai skenario navigasi tanpa risiko merusak komponen robot. Secara keseluruhan, sistem ini dapat digunakan tidak hanya dalam konteks penelitian, tetapi juga untuk pengembangan robot edukatif maupun prototipe aplikasi indoor

