

**LAPORAN**  
**PENERAPAN FDK PADA *DIFFERENTIAL DRIVE MOBILE***  
**ROBOT (DDMR)**



**Disusun Oleh:**

**Kelompok VI**

M Fat Hiy Ilman N (210491100003)

Astri Nur P (210491100004)

Siti Nurhalisa (210491100017)

Devan Yusfa S (210491100027)

Firman Aliyansyah S (210491100030)

**PROGRAM STUDI TEKNIK MEKATRONIKA**  
**JURUSAN TEKNIK ELEKTRO**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS TRUNOJOYO MADURA**  
**2024**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

*Differential Drive Mobile Robot* (DDMR) adalah jenis robot yang bergerak menggunakan dua roda yang dikendalikan secara independen. Sistem ini memungkinkan robot untuk bermanuver dengan fleksibilitas tinggi, membuatnya cocok untuk berbagai aplikasi seperti eksplorasi, transportasi otomatis, dan operasi penyelamatan. Dalam menghadapi lingkungan yang dinamis dan kompleks, diperlukan sistem kendali yang mampu mengatur pergerakan robot dengan akurat dan efisien. Salah satu metode yang menjanjikan adalah *Forward Differential Kinematic* (FDK). FDK adalah pendekatan kendali yang memungkinkan penyesuaian real-time pada kecepatan roda untuk mencapai lintasan yang diinginkan. Dengan menggunakan metode ini, DDMR dapat mengatasi tantangan seperti perubahan medan, rintangan yang tidak terduga, dan navigasi yang presisi di lingkungan yang tidak terstruktur. FDK memberikan kemampuan adaptasi yang lebih baik

dibandingkan metode kendali konvensional, yang sering kali tidak mampu menangani ketidakpastian dan variasi dalam lingkungan operasi.

Implementasi FDK pada DDMR membutuhkan pemahaman yang mendalam tentang kinematika robot dan pengembangan algoritma kendali yang efektif. Dalam konteks ini, FDK menawarkan solusi yang lebih robust dan fleksibel, memungkinkan DDMR untuk beroperasi dengan lebih efisien dan akurat. Penerapan FDK tidak hanya meningkatkan kinerja robot dalam navigasi, tetapi juga dalam stabilitas dan respons terhadap situasi yang berubah-ubah. Penelitian ini berfokus pada penerapan FDK pada DDMR, dengan tujuan untuk merancang, mengimplementasikan, serta menguji dan mengevaluasi sistem kendali ini. Diharapkan penelitian ini dapat memberikan kontribusi signifikan dalam pengembangan teknologi kendali robotika, khususnya dalam meningkatkan efektivitas dan efisiensi DDMR.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas, rumusan masalah yang akan dibahas dalam penelitian ini adalah

sebagai berikut:

1. Bagaimana cara mengimplementasikan sistem kendali FDK pada *differential drive mobile robot* (DDMR)?
2. Bagaimana menguji dan mengevaluasi kinerja sistem kendali FDK pada DDMR?

### **1.3 Tujuan Masalah**

Berdasarkan rumusan masalah yang dipaparkan diatas, dibawah ini merupakan tujuan dari penelitian ini sebagai berikut:

1. Dapat mengimplementasikan sistem kendali FDK pada *differential drive mobile robot* (DDMR).
2. Menguji dan mengevaluasi kinerja sistem kendali FDK pada DDMR.

### **1.4 Batasan Masalah**

Agar penelitian ini dapat terfokus dan berjalan dengan efektif, maka ditetapkan beberapa batasan masalah sebagai berikut:

1. Pengujian dilakukan dengan variasi kecepatan dan arah yang telah ditentukan untuk mengukur

performa sistem kendali.

2. Pengujian dilakukan di lingkungan yang terkontrol dan didesain untuk merepresentasikan skenario nyata namun terbatas dalam skala dan kompleksitas.

## **BAB II**

### **STUDI LITERATUR**

#### **2.1. Robot Beroda**

Robot beroda adalah jenis robot yang menggunakan roda sebagai alat utama untuk bergerak. Dalam beberapa tahun terakhir, definisi dan pemahaman mengenai robot beroda telah berkembang seiring dengan kemajuan teknologi. Menurut Choset et al. (2019) Robot beroda adalah robot yang menggunakan satu atau lebih roda untuk bergerak. Mereka menekankan bahwa roda menawarkan stabilitas dan penanganan yang lebih baik dibandingkan moda transportasi lainnya, sehingga ideal untuk aplikasi yang memerlukan pergerakan cepat dan efisien pada permukaan datar. Thrun et al. (2020), Robot beroda dirancang untuk beroperasi di lingkungan terstruktur di mana roda digunakan untuk navigasi. Mereka menekankan bahwa robot beroda memiliki kemampuan navigasi presisi tinggi berkat an sensor dan algoritma kontrol canggih. Sedangkan menurut Kim et al. (2021), Robot beroda merupakan robot yang

menggunakan radar untuk mobilitas yang optimal. Mereka menunjukkan bahwa roda memungkinkan pergerakan cepat dengan stabilitas yang baik dan lebih mudah dikendalikan dibandingkan robot berkaki. Siciliano dan Khatib (2022), robot beroda menggunakan satu atau lebih roda untuk bergerak dan bermanuver. Mereka menekankan bahwa robot beroda sangat cocok untuk aplikasi di industri manufaktur dan layanan, di mana kecepatan dan efisiensi energi adalah faktor penting.

Ada beberapa jenis robot beroda, yang pertama, ada differential drive robots menggunakan dua roda yang dikontrol secara independen yang memungkinkan robot berputar dengan mengatur kecepatan masing-masing roda. Tipe ini populer karena desain dan kemudahan pengoperasiannya. Sistem roda lebih mudah dirancang dan diimplementasikan dibandingkan dengan sistem kaki atau rantai (Choset et al. 2019). Kedua, ackerman steering robots menggunakan sistem kemudi mirip mobil, di mana roda depan digunakan untuk mengarahkan robot. Konfigurasi ini memberikan stabilitas baik pada kecepatan tinggi. Roda memungkinkan pergerakan yang cepat dan efisien di atas permukaan datar. Integrasi sensor canggih

seperti LIDAR dan kamera stereoskopik telah meningkatkan kemampuan navigasi dan deteksi rintangan pada robot beroda. (Thrun et al. 2020). Ketiga, omnidirectional robots menggunakan roda omni atau mecanum yang memungkinkan Anda bergerak ke segala arah tanpa harus berbelok. Konfigurasi ini meningkatkan kemampuan manuver di ruang sempit atau lingkungan dinamis. an algoritma kendali adaptif memungkinkan robot beroda untuk menyesuaikan kecepatan dan arah berdasarkan kondisi lingkungan secara real-time. robot beroda sangat efektif digunakan untuk navigasi di lingkungan dalam ruangan yang terstruktur, seperti rumah sakit dan pusat perbelanjaan. (Kim et al. 2021).

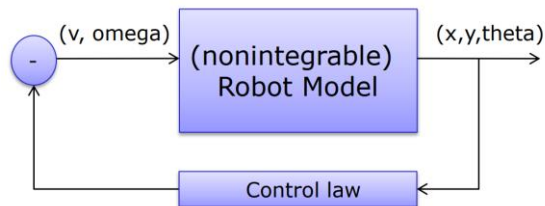


# BAB III

## PERANCANGAN SISTEM KONTROL GERAK DENGAN FDK

### 3.1 Sistem Kontrol Gerak dengan FDK

Berikut merupakan sistem kontrol gerak dari FDK yang ditunjukkan dibawah ini pada Gambar 3.1.



**Gambar 3.1.** Sistem Kontrol

#### 3.1.1 Kecepatan linear ( $v$ )

Besaran yang menunjukkan seberapa cepat sebuah objek bergerak sepanjang lintasan lurus. Dalam konteks robotik, kecepatan linear mengacu pada kecepatan translasi pusat massa robot dalam arah tertentu, yang biasanya diukur dalam meter per detik (m/s). Kecepatan dihitung sebagai rata-rata kecepatan roda kiri dan kanan pada sistem penggerak diferensial. Kecepatan linear

penting untuk menentukan seberapa cepat robot dapat bergerak maju atau mundur, dan merupakan komponen utama dalam navigasi robot untuk bergerak dari satu titik ke titik lain dengan efisien. Berikut adalah persamaan yang digunakan kecepatan linear untuk forward differential kinematics

$$v = \frac{v_{left} + v_{right}}{2}$$

Cara untuk menghitung kecepatan linear ( $v$ ) sebuah robot dengan forward diferensial kinematics:

- ( $v$ ) adalah kecepatan linear dari robot, yaitu seberapa cepat robot bergerak maju atau mundur.
- $v_{left}$  adalah kecepatan roda kiri.
- $v_{right}$  adalah kecepatan roda kanan.

Persamaan ini mengambil rata-rata kecepatan dari kedua roda untuk menentukan kecepatan linear keseluruhan dari robot. Dengan kata lain, kecepatan angular robot adalah rata-rata dari kecepatan kedua roda.

### 3.1.2 Kecepatan Angular ( $\omega$ )

Kecepatan angular merupakan ukuran seberapa cepat sebuah objek berputar atau mengubah orientasinya,

dan dalam konteks robot berpengerak diferensial, ini merujuk pada seberapa cepat robot dapat berbelok. Kecepatan angular ( $\omega$ ) dihitung dengan menggunakan perbedaan kecepatan antara roda kiri ( $v_{left}$ ) dan roda kanan ( $v_{right}$ ) serta jarak antara kedua roda tersebut ( $b$ ). Besaran ini diukur dalam radian per detik (rad/s). Kecepatan angular sangat penting untuk navigasi robot, karena memungkinkan robot untuk berbelok dan menghindari hambatan, sehingga membuat robot lebih responsif dan adaptif terhadap perubahan lingkungan di sekitarnya. Berikut adalah persamaan yang digunakan kecepatan angular untuk forward differential kinematics

$$\omega = \frac{v_{left} - v_{right}}{b}$$

Untuk menghitung kecepatan angular ( $\omega$ ) sebuah robot dengan sistem pengerak diferensial. Berikut adalah pengertian dari komponen-komponen dalam persamaan tersebut:

- ( $\omega$ ) adalah kecepatan angular, yang mengukur seberapa cepat robot berputar atau mengubah orientasinya, dinyatakan dalam radian per detik

(rad/s).

- $v_{left}$  adalah kecepatan roda kiri.
- $v_{right}$  adalah kecepatan roda kanan.
- $(b)$  adalah jarak antara roda kiri dan roda kanan, yang dikenal juga sebagai track width atau wheelbase, dinyatakan dalam meter.

Persamaan ini mengukur seberapa cepat robot dapat berbelok berdasarkan perbedaan kecepatan antara kedua rodanya dan jarak di antara mereka. Kecepatan angular penting untuk mengontrol gerakan berbelok robot, memastikan robot dapat menavigasi dengan tepat dan efisien melalui berbagai lintasan dan lingkungan. Kecepatan angular ( $\omega$ ) menggambarkan tingkat perubahan orientasi robot. Jika kecepatan roda kanan ( $v_{right}$ ) lebih besar dari kecepatan roda kiri ( $v_{left}$ ), robot akan berputar ke kiri, menghasilkan kecepatan angular positif. Sebaliknya, jika ( $v_{left}$ ) lebih besar dari ( $v_{right}$ ), robot akan berputar ke kanan, menghasilkan kecepatan angular negatif.

### **3.2 Prinsip kerja**

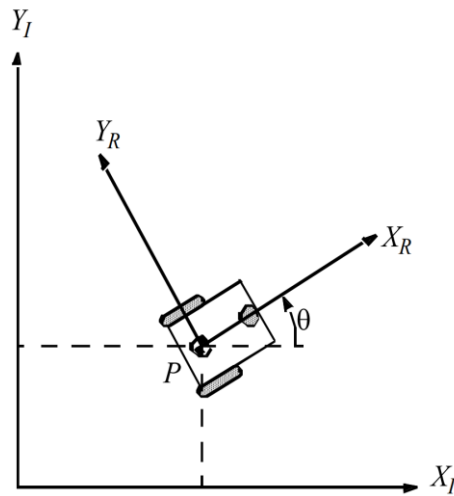
Penerapan Forward Differential Kinematics (FDK) pada DDMR merupakan suatu metode yang digunakan untuk mengendalikan pergerakan robot berdasarkan kinematika diferensial. Kinematika diferensial menggambarkan hubungan antara kecepatan roda dan kecepatan linier dan sudut robot. Pada aplikasi FDK terdapat algoritma kontrol yang mengubah kecepatan yang diinginkan menjadi kecepatan roda yang sesuai sehingga robot dapat bergerak sesuai keinginan. Penerapan FDK pada DDMR memungkinkan kontrol yang tepat terhadap pergerakan robot, menjadikannya sangat berguna dalam situasi yang membutuhkan navigasi yang akurat dan adaptif. Dengan desain sistem kendali yang baik, pemrograman yang efektif, dan kalibrasi yang tepat, DDMR dapat menjadi alat yang sangat efektif dalam berbagai tugas otomatisasi dan layanan.

Dalam kinematika diferensial maju dua dimensi untuk robot bergerak (mobile robot), kecepatan linier merujuk pada laju perubahan posisi dari pusat massa robot

dalam bidang dua dimensi. Kecepatan ini dinyatakan dalam vektor  $\mathbf{v}$ , yang komponennya merepresentasikan kecepatan dalam arah sumbu x dan y. Vektor kecepatan linier ( $\mathbf{v}$ ) dihitung berdasarkan turunan posisi pusat massa robot terhadap waktu. Untuk robot bergerak, kecepatan linier sering kali diukur dalam arah maju (kecepatan translasi) dan dinyatakan sebagai  $\mathbf{v} = (v_x, v_y)$  dimana  $v_x$  dan  $v_y$  adalah kecepatan dalam arah x dan y.

Kecepatan angular dalam konteks robot bergerak merujuk pada laju perubahan orientasi robot dalam bidang dua dimensi, biasanya dinyatakan sebagai laju rotasi ( $\omega$ ) sekitar titik pivot atau pusat rotasi robot. Kecepatan angular adalah skalar yang menunjukkan seberapa cepat robot berputar dalam arah searah jarum jam atau berlawanan arah jarum jam. Untuk robot bergerak, kecepatan angular ( $\omega$ ) sangat penting untuk mengendalikan arah gerak robot, memungkinkan robot untuk berbelok atau mengubah orientasi dalam bidang dua dimensi. Kombinasi kecepatan linier dan angular memungkinkan robot bergerak untuk mengikuti lintasan

yang diinginkan dengan akurasi tinggi, melakukan navigasi di lingkungan yang kompleks, dan menghindari rintangan. Prinsip kerja gerak forward differential kinematics yang di tunjukan pada gambar 3.2.



**Gambar 3.2.** Prinsip Kerja Gerak Forward Differential Kinematics

## **BAB IV**

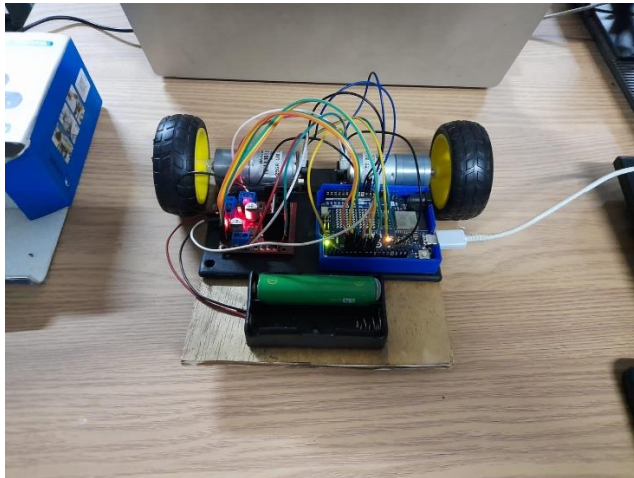
### **HASIL DAN PEMBAHASAN**

#### **4.1 Uji Coba**

Berikut merupakan beberapa hasil uji coba pada mobile robot menggunakan gui Tkinter sebagai Kecepatan angular serta translasi sebagai input nilai

##### **A. Desain mobile robot**

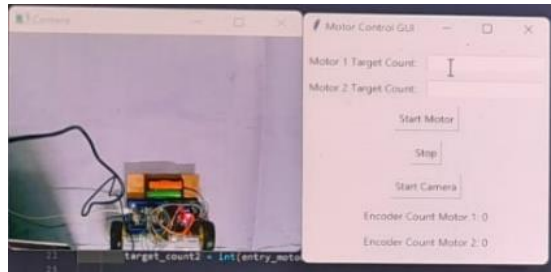
Berikut merupakan desain mobile robot yang digunakan pada percobaan real-time pada Gambar 4..



##### **B. Uji coba Forward differensial drive**



Berikut merupakan hasil uji coba forward differential drive menggunakan gui Tkinter yang ditunjukkan pada Gambar 4.2.



**Gambar 4.2.** FDD Menggunakan gui Tkinter

Berikut merupakan hasil uji coba forward differential drive menggunakan gui Tkinter yang ditunjukkan pada Gambar 4.3.



**Gambar 4.3.** Forward Differential Drive gui Tkinter

Motor DC 25GA370 12V 100RPM dengan encoder memiliki putaran 100 RPM pada tegangan 12V.

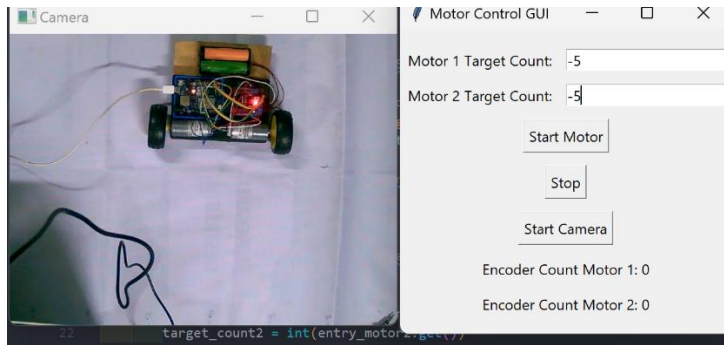
Motor tersebut dilengkapi dengan encoder, yang memungkinkan untuk pengukuran dan kontrol putaran dengan presisi tinggi. Encoder biasanya digunakan untuk mendapatkan umpan balik posisi atau kecepatan motor secara akurat.

Spesifikasi umum dari motor ini mencakup Tegangan operasi: 12V DC Kecepatan: 100 RPM (Rotations Per Minute) Torsi: Tinggi, disesuaikan dengan aplikasi torsi tinggi Encoder: Digunakan untuk mengukur posisi atau kecepatan dengan tepat. Motor digunakan dalam kontrol akurat terhadap putaran motor.

## **4.2 Pembahasan**

Menggunakan Tkinter untuk mengontrol dua motor melalui Arduino. GUI ini memungkinkan untuk memasukkan target hitungan untuk masing-masing motor dan mengirimkan perintah melalui koneksi serial ke Arduino untuk menggerakkan motor sesuai dengan input yang diberikan. Terdapat juga fitur untuk memulai dan menghentikan motor, serta menampilkan gambar dari kamera yang diambil menggunakan OpenCV. Selain itu,

program juga terus-menerus membaca nilai hitungan encoder dari Arduino dan memperbarui tampilan GUI secara real-time. Tampilan Gui ditunjukkan pada gambar 4.4.



**Gambar 4.4.** Tampilan GUI Tkinter

memasukkan nilai input untuk kedua motor pada input yang disediakan. Setelah itu, program akan mengirimkan perintah ke Arduino melalui koneksi serial untuk mengatur motor dengan target hitungan input. Selain itu, program juga memungkinkan untuk memulai dan menghentikan motor, serta untuk melihat feed kamera langsung menggunakan OpenCV. Selama operasi, program secara terus-menerus memperbarui tampilan GUI dengan nilai hitungan encoder yang dikirimkan oleh

Arduino, sehingga dapat memantau status motor secara real-time.

```
void loop() {  
  
    if (Serial.available() > 0) {  
  
        String input = Serial.readStringUntil('\n');  
  
        int commaIndex = input.indexOf(',');  
  
        if (commaIndex > 0) {  
  
            targetCount1          =          input.substring(0,  
commaIndex).toInt();  
  
            targetCount2  =  input.substring(commaIndex  +  
1).toInt();  
  
            encoderCount1 = 0;  
  
            encoderCount2 = 0;  
  
        }  
  
    }  
  
    float    targetSpeedLeft    =    (targetCount1    -  
encoderCount1) * Kp;
```

```

float targetSpeedRight = (targetCount2 -
encoderCount2) * Kp;

float V = (targetSpeedLeft + targetSpeedRight) / 2.0;

float omega = (targetSpeedRight - targetSpeedLeft) /
L;

int motorSpeedLeft = V - omega * L / 2;

int motorSpeedRight = V + omega * L / 2;

if (motorSpeedLeft > 0) {

    digitalWrite(in1, HIGH);

    digitalWrite(in2, LOW);

    analogWrite(enA, abs(motorSpeedLeft) * 2);

} else if (motorSpeedLeft < 0) {

    digitalWrite(in1, LOW);

    digitalWrite(in2, HIGH);

    analogWrite(enA, abs(motorSpeedLeft) * 2);

} else {

```

```
digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

analogWrite(enA, 0);

}

if (motorSpeedRight > 0) {

    digitalWrite(in3, HIGH);

    digitalWrite(in4, LOW);

    analogWrite(enB, abs(motorSpeedRight) * 2);

} else if (motorSpeedRight < 0) {

    digitalWrite(in3, LOW);

    digitalWrite(in4, HIGH);

    analogWrite(enB, abs(motorSpeedRight) * 2);

} else {

    digitalWrite(in3, LOW);

    digitalWrite(in4, LOW);

}
```

```
    analogWrite(enB, 0);  
  
}  
  
Serial.print("Encoder Count Motor 1: ");  
  
Serial.println(encoderCount1);  
  
Serial.print("Encoder Count Motor 2: ");  
  
Serial.println(encoderCount2);  
  
delay(100);  
  
}
```

Ketika memasukkan nilai target untuk motor melalui gui Python Tkinter, prosesnya dimulai dengan membaca dan memisahkan nilai tersebut untuk masing-masing motor. Nilai-nilai ini kemudian diinterpretasikan sebagai target hitungan encoder yang ingin dicapai oleh motor 1 dan motor 2. Perhitungan kecepatan translasi dan rotasi pada robot diferensial memberikan pemahaman tentang bagaimana kecepatan roda kiri dan kanan berkontribusi terhadap gerakan keseluruhan robot.

Kecepatan translasi  $v$  adalah hasil dari rata-rata kecepatan roda kiri ( $v_{left}$ ) dan kecepatan roda kanan ( $v_{right}$ )



## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Penerapan FDK pada DDMR memungkinkan kontrol yang akurat dan efisien terhadap pergerakan robot. Melalui an model kinematika dan algoritma kontrol yang tepat, DDMR dapat bergerak dengan presisi tinggi, menjadikannya sangat berguna dalam berbagai aplikasi industri dan layanan. Implementasi yang tepat dari sistem kendali, pemrograman, dan kalibrasi sangat penting untuk memastikan kinerja optimal dari DDMR.

Pemahaman tentang perhitungan kecepatan translasi dan rotasi adalah kunci untuk menguji performa dan responsivitas robot. Kecepatan translasi  $V$ , yang merupakan rata-rata dari kecepatan roda kiri dan kanan, memungkinkan pengujian untuk mengevaluasi seberapa baik robot dapat bergerak maju dengan kecepatan yang diinginkan. Sementara itu, kecepatan rotasi  $\omega$ , yang dihitung dari perbedaan kecepatan roda kanan dan kiri, serta lebar sumbu roda  $L$ , penting dalam menguji

kemampuan robot untuk melakukan putaran dan manuver di tempat dengan akurat. Uji coba ini memastikan bahwa robot dapat mengontrol orientasi dan posisi dengan tepat sesuai perintah yang diberikan.

Dengan mengintegrasikan kedua perhitungan ini secara efektif dalam uji coba, dapat mengevaluasi performa robot dalam berbagai skenario navigasi dan manuver, serta memperbaiki atau menyesuaikan parameter kontrol untuk meningkatkan responsivitas dan presisi gerakan robot dalam kondisi uji coba yang berbeda-beda.

## **5.2. Saran**

Implementasikan sistem pemantauan kondisi sensor secara real-time untuk mendeteksi dan mengoreksi kesalahan dengan cepat. Pilih motor dengan spesifikasi yang sesuai, seperti torsi yang cukup dan respons yang cepat, untuk memastikan pergerakan yang presisi dan sesuai kebutuhan.

## **DAFTAR PUSTAKA**

Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., & Thrun, S. (2019). Principles of Robot Motion: Theory, Algorithms, and Implementations. MIT Press.

Thrun, S., Burgard, W., & Fox, D. (2020). Probabilistic Robotics. MIT Press.

Kim, J., Kim, D., & Park, J. (2021). A Survey on Wheeled Mobile Robots: Mechanisms and Control. Journal of Robotics and Autonomous Systems, 132, 103641.

Siciliano, B., & Khatib, O. (Eds.). (2022). Springer Handbook of Robotics (2nd ed.). Springer.