

LAPORAN
PENERAPAN IDK 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

DAFTAR ISI

DAFTAR ISI	i
DAFTAR GAMBAR	ii
BAB I : PENDAHULUAN	ii
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Masalah.....	2
1.4. Batasan Masalah.....	3
BAB II : STUDI LITERATUR	4
2.1. Robot Beroda dengan FDK.....	4
2.2. Kelemahan Metode FDK.....	6
BAB III : PERANCANGAN SISTEM KONTROL PATH FOLLOWING DENGAN IDK	8
BAB IV : HASIL DAN PEMBAHASAN	13
BAB V : PENUTUP	20
DAFTAR PUSTAKA	23

DAFTAR GAMBAR

Gambar 4.1. Desain Mobile Robot.....	13
Gambar 4.2. <i>Invers differensial kinematic</i> Menggunakan gui Tkinter.....	14
Gambar 4.3. <i>Invers differensial kinematic</i> gui Tkinter	14

BAB I

PENDAHULUAN

1.1. Latar Belakang

Differential Drive Mobile Robot (DDMR) merupakan salah satu jenis robot bergerak yang banyak digunakan dalam berbagai aplikasi, seperti otomasi industri, eksplorasi, dan pengangkutan material. Keunggulan utama dari DDMR adalah fleksibilitas dan kemampuannya untuk bermanuver di berbagai medan, berkat dua roda penggerak yang dikendalikan secara independen. Namun, untuk mencapai kinerja optimal, diperlukan metode kontrol yang tepat, salah satunya *adalah inverse differential kinematics*. *Inverse differential kinematics* adalah teknik yang digunakan untuk menentukan kecepatan individu dari setiap roda robot berdasarkan kecepatan linier dan angular yang diinginkan dari robot. Teknik ini sangat penting dalam perencanaan jalur dan kontrol gerakan, karena memungkinkan robot untuk bergerak dengan presisi sesuai dengan lintasan yang telah ditentukan. Dalam konteks DDMR, *inverse differential kinematics* berperan penting dalam mengonversi *input* gerakan yang diinginkan menjadi *output* kontrol pada roda.

Penerapan *inverse differential kinematics* pada DDMR melibatkan pengembangan model kinematika yang

akurat, formulasi persamaan kinematika invers, serta implementasi algoritma kontrol yang efisien. Model kinematika yang digunakan harus mencakup parameter-parameter penting seperti jarak antara roda dan radius roda, yang akan mempengaruhi perhitungan kecepatan roda. Selain itu, persamaan kinematika invers harus mampu mengonversi kecepatan linier dan angular robot menjadi kecepatan roda kiri dan kanan dengan presisi tinggi.

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 IDK pada *differential drive mobile robot* (DDMR)?
2. Bagaimana menguji dan mengevaluasi kinerja sistem kendali IDK 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 IDK pada *differential drive mobile robot* (DDMR).

2. Menguji dan mengevaluasi kinerja sistem kendali IDK 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 dengan FDK

Robot beroda adalah jenis robot yang menggunakan roda sebagai mekanisme utama untuk bergerak. Roda memungkinkan robot untuk mencapai kecepatan yang lebih tinggi dan efisiensi energi yang lebih baik dibandingkan dengan mekanisme lainnya. Robot beroda umumnya digunakan dalam berbagai aplikasi, seperti logistik, layanan, dan navigasi otomatis di lingkungan dalam ruangan. Robot beroda adalah pilihan yang populer untuk aplikasi logistik dan layanan karena kemampuan mereka untuk bergerak cepat dan efisien di permukaan yang datar (Kim et al. 2021).

Forward Differential Kinematics (FDK) adalah metode yang digunakan untuk mengendalikan pergerakan robot beroda, khususnya *Differential Drive Mobile Robot* (DDMR). Metode ini menggambarkan hubungan antara kecepatan roda individu dan kecepatan gerak keseluruhan robot, baik dalam hal translasi (gerak maju/mundur) maupun rotasi (putaran). FDK memungkinkan kontrol yang presisi terhadap pergerakan robot beroda dengan menggunakan model kinematika diferensial. Mereka juga menyoroti

pentingnya kalibrasi sensor dan optimasi algoritma kendali untuk mencapai kinerja yang optimal (Kumar et al. 2023).

Implementasi FDK pada robot beroda melibatkan beberapa langkah penting, termasuk pemodelan kinematika, pengukuran kecepatan roda, dan kontrol gerakan. Pemodelan kinematika diferensial adalah langkah awal dalam menerapkan FDK. Model ini menghubungkan kecepatan roda individu dengan kecepatan translasi dan rotasi robot. Persamaan kinematika digunakan untuk mentransformasikan kecepatan yang diinginkan menjadi kecepatan roda yang sesuai (Thrun et al. 2020). Pengukuran kecepatan roda secara akurat sangat penting untuk kinerja FDK yang baik. Sensor seperti *encoder* roda digunakan untuk mengukur kecepatan aktual roda. Data ini kemudian digunakan dalam algoritma kontrol untuk menyesuaikan kecepatan roda agar sesuai dengan kecepatan yang diinginkan (Siciliano dan Khatib. 2022). Kontrol gerakan melibatkan penggunaan algoritma seperti PID (Proportional-Integral-Derivative) untuk mengurangi kesalahan antara kecepatan yang diinginkan dan kecepatan aktual. Kontrol yang efektif memastikan bahwa robot bergerak sesuai dengan jalur yang direncanakan dan merespons dengan cepat terhadap perubahan kondisi (Choset et al. 2019).

2.2. **Kelemahan Metode FDK**

Meskipun *Forward Differential Kinematics* (FDK) adalah metode yang umum dan efektif, ia memiliki beberapa kelemahan yang dapat mempengaruhi kinerja robot beroda. Kesalahan akumulatif terjadi karena ketidakakuratan dalam pengukuran posisi dan kecepatan. Ini dapat menyebabkan deviasi yang signifikan dari jalur yang diinginkan seiring waktu, terutama dalam operasi jangka panjang (Kim et al. 2021). Kinerja FDK sangat bergantung pada kondisi permukaan tempat robot beroperasi. Permukaan yang tidak rata atau licin dapat menyebabkan roda tergelincir atau kehilangan kontak yang tepat, sehingga mengurangi akurasi pergerakan (Kumar et al. 2023).

Robot beroda dengan FDK tidak dapat dengan mudah mengatasi hambatan fisik di jalurnya. Ini membatasi fleksibilitas dan kemampuan navigasi robot di lingkungan yang kompleks atau dinamis (Siciliano dan Khatib. 2022). Pada kecepatan tinggi, respons sistem kendali FDK dapat menjadi kurang stabil dan kurang akurat. Disebabkan oleh keterbatasan dalam kecepatan respons motor dan sistem kendali itu sendiri (Thrun et al. 2020). Implementasi algoritma FDK membutuhkan pemrosesan yang cukup kompleks, terutama dalam hal kalibrasi dan pemrograman kontrol. Ini memerlukan sumber daya komputasi yang memadai dan

pengetahuan teknis yang tinggi untuk memastikan kinerja yang optimal (Choset et al. 2019).

BAB III

PERANCANGAN SISTEM KONTROL *PATH FOLLOWING* DENGAN IDK

3.1. Definisi

Perancangan sistem kontrol *path following* dengan menggunakan inverse differential kinematics merupakan pendekatan yang penting dalam bidang robotika, terutama untuk aplikasi di mana robot harus mengikuti jalur yang telah ditentukan dengan presisi tinggi. Teknik ini sering diterapkan pada robot bergerak, seperti kendaraan otonom, robot pembersih, dan robot industri yang bergerak di sekitar pabrik. Dalam sistem ini, algoritma kinematika diferensial digunakan untuk menghitung kecepatan sudut dan linear yang diperlukan untuk setiap roda robot agar dapat mengikuti jalur yang diinginkan.

Proses perancangan sistem kontrol *path following* dimulai dengan mendefinisikan jalur yang harus diikuti oleh robot. Jalur ini bisa berupa lintasan lurus, kurva, atau kombinasi keduanya. Setelah jalur ditentukan, algoritma *inverse differential kinematics* digunakan untuk menghitung kecepatan yang diperlukan untuk setiap segmen jalur. Algoritma ini memperhitungkan parameter kinematika robot, seperti panjang sumbu roda dan radius roda, untuk

menghasilkan kontrol yang presisi. Dalam banyak kasus, model kinematika robot dikembangkan terlebih dahulu untuk memahami bagaimana pergerakan robot dipengaruhi oleh *input* kontrol yang diberikan.

Setelah model kinematika dikembangkan, tahap berikutnya adalah merancang pengendali yang akan mengatur kecepatan roda robot berdasarkan *input* dari algoritma kinematika. Pengendali ini dapat berupa PID controller, *fuzzy logic controller*, atau metode kontrol lainnya yang cocok dengan dinamika sistem. Pengendali ini bertugas memastikan bahwa robot dapat mengikuti jalur yang diinginkan dengan kesalahan minimal, meskipun terdapat gangguan atau variasi di lingkungan sekitar. Dalam implementasinya, sensor seperti *encoder*, IMU, dan GPS sering digunakan untuk menyediakan *feedback* yang akurat kepada sistem kontrol.

Pengujian dan validasi merupakan tahap kritis dalam perancangan sistem kontrol path following. Pengujian dilakukan di lingkungan simulasi dan juga di lingkungan nyata untuk memastikan bahwa sistem bekerja sesuai dengan spesifikasi. Selama tahap ini, parameter kontrol sering kali perlu disesuaikan untuk mencapai performa yang optimal. Uji coba dilakukan dengan berbagai scenario jalur dan kondisi lingkungan untuk memastikan bahwa sistem dapat beradaptasi dengan baik. Keberhasilan perancangan sistem kontrol ini

akan sangat bergantung pada kemampuan algoritma kinematika dan pengendali untuk bekerja secara harmonis dalam berbagai kondisi operasional. IDK menggunakan model kinematika berikut untuk mentransformasikan kecepatan translasi (V) dan kecepatan rotasi (ω) dari robot menjadi kecepatan roda kiri (θ_L) dan kanan (θ_R):

$$\begin{bmatrix} \theta_R \\ \theta_L \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & \frac{d}{2} \\ 1 & -\frac{d}{2} \end{bmatrix} \begin{bmatrix} V \\ \omega \end{bmatrix}$$

3.2. Prinsip Kerja

Prinsip kerja dari perancangan sistem kontrol *path following* dengan *inverse differential kinematics* melibatkan beberapa langkah kunci yang berfokus pada memastikan bahwa robot dapat mengikuti jalur yang telah ditentukan dengan presisi tinggi. Langkah pertama adalah menentukan jalur atau lintasan yang harus diikuti oleh robot. Jalur ini bisa berbentuk lintasan lurus, kurva, atau kombinasi dari keduanya. Jalur yang telah ditentukan ini kemudian digunakan sebagai referensi bagi robot untuk bergerak.

Setelah jalur ditentukan, data posisi robot dikumpulkan secara real-time menggunakan sensor-sensor seperti encoder, IMU (*Inertial Measurement Unit*), dan GPS. Data ini kemudian dibandingkan dengan jalur yang diinginkan

untuk menentukan deviasi atau perbedaan posisi robot dari jalur yang telah ditentukan. Algoritma inverse differential kinematics kemudian digunakan untuk menghitung kecepatan sudut dan linear yang diperlukan oleh roda-roda robot untuk mengoreksi deviasi ini. Algoritma ini memanfaatkan model kinematika robot yang memperhitungkan parameter seperti jarak antara roda dan radius roda untuk menghitung pergerakan yang diperlukan.

Setelah kecepatan yang diperlukan dihitung, sinyal kontrol dikirimkan ke motor-motor yang menggerakkan roda robot. Sistem kontrol, seperti *PID controller* atau *fuzzy logic controller*, digunakan untuk mengatur kecepatan motor sesuai dengan input dari algoritma kinematika. Sistem kontrol ini bertugas memastikan bahwa robot dapat mengikuti jalur yang diinginkan dengan kesalahan minimal, meskipun terdapat gangguan atau variasi di lingkungan sekitar. Sensor terus memberikan feedback secara real-time, memungkinkan sistem kontrol untuk melakukan penyesuaian secara terus-menerus.

Proses ini berulang secara terus-menerus dalam *loop* tertutup (*closed-loop*), di mana data sensor terus-menerus dikumpulkan, deviasi dihitung, dan kecepatan roda disesuaikan untuk menjaga robot tetap pada jalur yang diinginkan. Dengan demikian, robot dapat mengikuti jalur yang ditentukan dengan presisi tinggi. Tahap pengujian dan

validasi sangat penting untuk memastikan bahwa sistem bekerja sesuai dengan spesifikasi dan dapat beradaptasi dengan berbagai kondisi lingkungan. Pengujian dilakukan baik di lingkungan simulasi maupun di lapangan, dan parameter kontrol sering kali perlu disesuaikan untuk mencapai performa optimal. Keberhasilan perancangan sistem kontrol *path following* ini sangat bergantung pada kerjasama yang harmonis antara algoritma kinematika dan sistem kontrol dalam berbagai kondisi operasional.

BAB IV

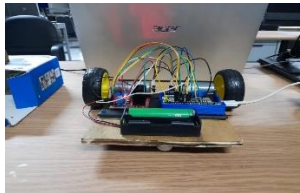
HASIL DAN PEMBAHASAN

4.1. Uji Coba

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

A. Desain mobile robot

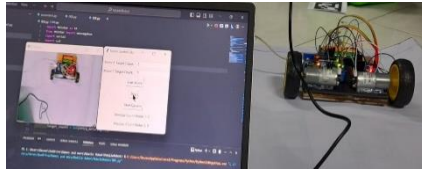
Berikut merupakan desain mobile robot yang digunakan pada percobaan *real-time* pada Gambar 4.1.



Gambar 4.1. Desain Mobile Robot

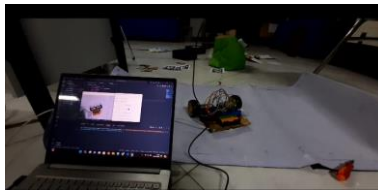
B. Uji coba *Invers differensial kinematic*

Berikut merupakan hasil uji coba *Invers differensial kinematic* menggunakan gui Tkinter yang ditunjukkan pada Gambar 4.2.



Gambar 4.2. *Invers differensial kinematic* Menggunakan gui Tkinter

Berikut merupakan hasil uji coba *Invers differensial kinematic* menggunakan *gui Tkinter* yang ditunjukkan pada Gambar 4.3.



Gambar 4.3. *Invers differensial kinematic* gui Tkinter

Pada percobaan IDK dengan *encoder*, *input* nilai dapat diubah dengan nilai θ_L dan θ_R yang berbeda menghasilkan derajat orientasi target. Sehingga kedua motor DC menghasilkan putaran roda yang berbeda sehingga derajat posisi juga berbeda sesuai dengan derajat target *input* yang ditentukan.

4.2. Pembahasan

Penggunaan Tkinter untuk mengontrol dua motor melalui Arduino menawarkan solusi yang efisien dan interaktif dalam automasi dan kendali motor. Tkinter, sebagai pustaka GUI standar untuk Python, memungkinkan pengguna untuk membuat antarmuka yang intuitif dan mudah digunakan. Dalam konteks ini, Tkinter digunakan untuk membuat GUI yang memungkinkan pengguna memasukkan target hitungan untuk masing-masing motor, memulai dan menghentikan motor, serta menampilkan gambar dari kamera secara *real-time* menggunakan OpenCV. Kombinasi antara Tkinter dan Arduino melalui koneksi serial memberikan kontrol yang akurat dan responsif, di mana perintah yang diberikan melalui GUI langsung diteruskan ke Arduino untuk menggerakkan motor sesuai dengan input yang diberikan. Hal ini memungkinkan pengguna untuk mengatur dan memantau kinerja motor dengan mudah dan efisien.

Selain fitur kontrol motor, integrasi dengan OpenCV memungkinkan GUI untuk menampilkan gambar dari kamera secara *real-time*, menambah dimensi visualisasi dan pemantauan yang lebih komprehensif. Proses ini melibatkan pengambilan gambar menggunakan OpenCV dan memperbarui tampilan GUI secara terus-menerus, memberikan umpan balik visual yang penting dalam aplikasi yang membutuhkan pengawasan langsung. Selain itu, program

ini juga dirancang untuk terus-menerus membaca nilai hitungan *encoder* dari Arduino, yang kemudian diperbarui dalam GUI secara *real-time*. Fitur ini tidak hanya meningkatkan akurasi dalam pemantauan posisi motor, tetapi juga memberikan transparansi penuh terhadap kinerja sistem, memungkinkan pengguna untuk melakukan penyesuaian yang diperlukan secara tepat waktu. Dengan demikian, kombinasi Tkinter, Arduino, dan OpenCV dalam satu sistem kontrol motor memberikan solusi yang menyeluruh dan efisien untuk berbagai aplikasi otomasi. Berikut merupakan program dari *Invers differensial kinematic gui Tkinter*

```
void setup() {  
    Serial.begin(9600);  
  
    pinMode(motorPin1, OUTPUT);  
    pinMode(motorPin2, OUTPUT);  
  
    // Inisialisasi motor dalam kondisi berhenti  
    analogWrite(motorPin1, 0);  
    analogWrite(motorPin2, 0);  
}  
  
void loop() {
```

```

if (Serial.available() > 0) {
    String input = Serial.readStringUntil('\n');
    input.trim();
    if (input.startsWith("ID=0")) {
        parseAndSetTarget(input);
    } else if (input == "stop") {
        stopMotors();
    }
}

if (motorRunning) {
    long currentCount1 = encoder1.read();
    long currentCount2 = encoder2.read();

    // Panggil fungsi IDK untuk menghitung kecepatan motor
    calculateIDK(currentCount1, currentCount2);

    if (currentCount1 >= targetCount1 && currentCount2 >=
targetCount2) {
        stopMotors();
    }

    sendEncoderValues(currentCount1, currentCount2);
}

```

```

    }
}

void parseAndSetTarget(const String &input) {
    int commaIndex = input.indexOf(',');
    if (commaIndex > 0) {
        String target1 = input.substring(commaIndex + 1,
input.indexOf(',', commaIndex + 1));
        String target2 = input.substring(input.indexOf(',',
commaIndex + 1) + 1);

        targetCount1 = target1.toInt();
        targetCount2 = target2.toInt();

        motorRunning = true;
    }
}

void stopMotors() {
    motorRunning = false;
    analogWrite(motorPin1, 0);
    analogWrite(motorPin2, 0);
}

```

```

void sendEncoderValues(long count1, long count2) {
    Serial.print("ID=1,");
    Serial.print(count1);
    Serial.print(",");
    Serial.println(count2);
}

void calculateIDK(long currentCount1, long
currentCount2) {
    float Kp = 0.5; // Gain proportional

    long error1 = targetCount1 - currentCount1;
    long error2 = targetCount2 - currentCount2;

    int speed1 = Kp * error1;
    int speed2 = Kp * error2;

    speed1 = constrain(speed1, -255, 255);
    speed2 = constrain(speed2, -255, 255);

    analogWrite(motorPin1, abs(speed1));
    analogWrite(motorPin2, abs(speed2));
}

```

```
digitalWrite(motorPin1, speed1 >= 0 ? HIGH : LOW);  
digitalWrite(motorPin2, speed2 >= 0 ? HIGH : LOW);  
}
```

BAB V

PENUTUP

5.1. Kesimpulan

Penerapan *Inverse Differential Kinematics* (IDK) pada *Differential Drive Mobile Robot* (DDMR) telah terbukti sebagai metode yang efektif untuk mengontrol pergerakan robot secara presisi. Dengan menggunakan IDK, kecepatan linear dan sudut roda robot dapat dihitung dan dikendalikan secara optimal berdasarkan jalur yang telah ditentukan. Algoritma kinematika yang digunakan dalam IDK memperhitungkan parameter kritis seperti jarak antara roda dan radius roda, memungkinkan pergerakan robot yang akurat dan efisien. Implementasi ini sangat berguna dalam aplikasi-aplikasi di mana jalur yang presisi sangat penting, seperti dalam navigasi kendaraan otonom, robot pembersih, dan robot industri.

Keberhasilan penerapan IDK pada DDMR juga sangat bergantung pada sistem kontrol yang melibatkan sensor-sensor yang memberikan *feedback real-time*, seperti *encoder*,

IMU, dan GPS. Sensor-sensor ini mengumpulkan data tentang posisi dan orientasi robot secara terus-menerus, yang kemudian digunakan untuk menghitung deviasi dari jalur yang diinginkan. Sistem kontrol, seperti PID controller atau fuzzy logic controller, memproses data ini dan mengatur kecepatan roda untuk menjaga robot tetap pada jalur yang telah ditentukan. Proses loop tertutup ini memastikan bahwa robot dapat beradaptasi terhadap perubahan lingkungan dan gangguan, menjaga performa yang konsisten dan andal.

Pengujian dan validasi sistem kontrol IDK pada DDMR menunjukkan bahwa dengan penyesuaian parameter yang tepat, sistem ini dapat bekerja dengan baik di berbagai kondisi lingkungan. Pengujian di lingkungan simulasi dan nyata menunjukkan bahwa sistem kontrol ini mampu mengatasi variasi jalur dan kondisi operasional dengan efisiensi yang tinggi. Menjadikan IDK sebagai solusi yang fleksibel dan adaptif untuk berbagai aplikasi robotika, memastikan robot dapat beroperasi dengan akurasi tinggi dan keandalan yang diperlukan dalam berbagai situasi.

Secara keseluruhan, penerapan IDK pada DDMR menunjukkan potensi besar dalam meningkatkan kemampuan navigasi dan kontrol robot. Dengan optimasi lebih lanjut pada algoritma dan sistem kontrol, serta peningkatan pada kualitas sensor dan pengujian yang komprehensif, sistem ini dapat

lebih ditingkatkan lagi performanya. Implementasi IDK tidak hanya memberikan solusi teknis yang efektif tetapi juga membuka peluang untuk aplikasi yang lebih luas dalam industri dan teknologi robotika masa depan.

5.2. Saran

Untuk meningkatkan penerapan *Inverse Differential Kinematics* (IDK) pada *Differential Drive Mobile Robot* (DDMR), penting untuk mengoptimalkan algoritma kinematika dan mempertimbangkan teknologi sensor terbaru seperti LiDAR untuk data yang lebih akurat. Kalibrasi dan pemeliharaan rutin pada sensor dan sistem kontrol harus dilakukan untuk memastikan kinerja optimal. Pengujian di berbagai kondisi lingkungan diperlukan untuk memastikan adaptabilitas sistem, sementara antarmuka pengguna yang intuitif akan mempermudah pengaturan dan pemantauan. Integrasi metode kontrol lanjutan seperti kontrol adaptif atau robust dapat meningkatkan stabilitas dan performa sistem, serta pelatihan operator yang memadai akan mengurangi kesalahan operasional. Dokumentasi dan berbagi hasil penelitian dengan komunitas ilmiah juga akan mendorong inovasi lebih lanjut dalam bidang ini.

DAFTAR PUSTAKA

- Kim, J., Kim, D., & Park, J. (2021). A Survey on Wheeled Mobile Robots: Mechanisms and Control. *Journal of Robotics and Autonomous Systems*, 132, 103641.
- Kumar, A., Singh, P., & Rao, D. (2023). Recent Advances in Wheeled Mobile Robots. *International Journal of Advanced Robotic Systems*, 20(2), 1-18.
- Siciliano, B., & Khatib, O. (Eds.). (2022). *Springer Handbook of Robotics* (2nd ed.). Springer.
- Thrun, S., Burgard, W., & Fox, D. (2020). *Probabilistic Robotics*. MIT Press.
- 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.