

BAB 2

TINJAUAN PUSTAKA

2.1 *Trajectory Tracking*

Trajectory tracking merupakan sebuah metode pelacakan lintasan pada suatu objek seperti *mobile robot* agar pergerakan pada objeknya (*Mobile Robot*) dapat terdeteksi atau terpantau sesuai dengan arahan atau perintah pada pengendaliannya. *Trajectory tracking* akan mencoba mengevaluasi keakuratan *mobile robot* dalam menuju titik tujuan, misalkan mencoba jalur terpendek, terpanjang, dan sebagainya.

Trajectory tracking pada pergerakan robot beroda dilakukan dengan tujuan untuk mengetahui posisi robot tersebut di area tempat robot tersebut diletakkan. Akurasi data pada *trajectory tracking* posisi robot sangatlah penting, karena semakin akurat data posisi dari robot yang dimiliki, semakin mudah untuk mengoperasikan robot tersebut secara otonomus. Proses *trajectory tracking* dapat dilakukan dengan menggunakan berbagai macam sensor, seperti penggunaan sensor encoder, sensor laser, sensor ultrasonic, serta sensor kamera. Dimana masing –masing penggunaan sensor tersebut memiliki kelebihan dan kekurangannya masing –masing.(Al Azhar et al., 2022)

Metode implementasi *trajectory tracking* dibagi menjadi dua kategori, yaitu *off-line trajectory* dan *online trajectory*. *Offline trajectory tracking* adalah metode perencanaan lintasan yang dilakukan sebelum robot mulai bergerak, di mana jalur lintasan dirancang berdasarkan informasi lingkungan

yang sudah diketahui secara statis. Metode ini cocok untuk lingkungan yang stabil dan terprediksi, seperti jalur produksi di pabrik, karena tidak memerlukan adaptasi selama operasi. Namun, kelemahannya adalah ketidakmampuannya untuk beradaptasi dengan perubahan lingkungan atau gangguan tak terduga. Sebaliknya, *online trajectory* memungkinkan lintasan direncanakan dan disesuaikan secara dinamis selama pergerakan robot. Robot menggunakan data sensor secara *real-time* untuk mengevaluasi lingkungan, bereaksi terhadap hambatan, atau memperbaiki kesalahan. Metode ini sangat fleksibel dan ideal untuk aplikasi di lingkungan yang dinamis atau tidak pasti, seperti kendaraan otonom atau robot yang bergerak di ruang publik. Meskipun lebih kompleks dan membutuhkan daya komputasi tinggi, *online trajectory* mampu memberikan solusi yang adaptif dan responsif terhadap perubahan kondisi operasional.

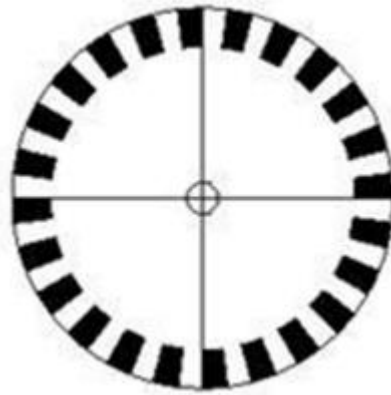
2.2 *Odometry*

Odometry adalah penggunaan data dari pergerakan aktuator untuk memperkirakan perubahan posisi dari waktu ke waktu. *Odometry* digunakan untuk memperkirakan posisi relatif terhadap posisi awal. Untuk memperkirakan posisi relatif robot, digunakan perhitungan jumlah pulsa yang dihasilkan oleh sensor *encoder* setiap satuan ukuran yang kemudian dikonversi menjadi satuan millimeter. Gambar 2.1 menunjukkan ilustrasi *rotary encoder*. Sedangkan Gambar 2.2 menunjukkan contoh desain robot yang menggunakan *odometry* dan parameter-parameter yang digunakan pada *odometry*. (Samsiana et al., 2020) Untuk mendapatkan jumlah pulsa setiap satu kali putaran roda

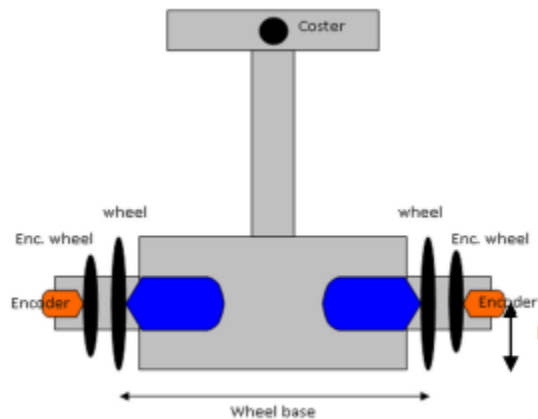
digunakan persamaan (2.1) dan persamaan (2.2). Untuk mencari jarak tempuh roda (D) dihitung dengan persamaan (2.3) sebagai berikut:

$$Keliling\ roda = 2\pi r \quad (2.1)$$

$$Pulse_per_mm = \frac{Resolusi\ encoder}{Keliling\ roda} \quad (2.2)$$



Gambar 2. 1 Ilustrasi Piringan Rotary Encoder



Gambar 2. 2 Desain Robot *Odometry*

Pada sistem penggerak diferensial terdapat dua roda,yaitu roda kanan dan roda kiri dan dimisalkan jumlah pulsa_per_mm untuk roda kanan adalah right_encoder dan roda kiri adalah left_encoder dan jarak antara dua roda

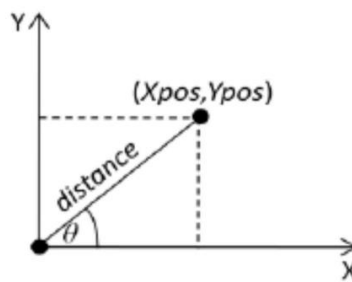
adalah $wheel_base$ maka didapatkan jarak tempuh ($distance$) dan sudut orientasi (θ) dalam satuan radian dengan persamaan (2.4) dan persamaan (2.5) berikut.

$$distance = \frac{(left_{enc} + right_{enc})}{2} \text{ atau } distance = \frac{(D_R + D_L)}{2} \quad (2.4)$$

$$\theta = \frac{(left_{enc} - right_{enc})}{Wheel\ base} \text{ atau } \theta = \frac{(D_L - D_R)}{wheel\ base} \quad (2.5)$$

Dari persamaan (2.4) dan persamaan (2.5) didapatkan bahwa nilai heading akan bernilai negatif (-) ketika robot berputar melawan arah jarum jam dan akan bernilai positif (+) ketika robot berputar searah dengan jarum jam. Dengan mengetahui jarak dan sudut ($distance$ dan θ) maka kita dapat mengetahui koordinat X dan koordinat Y dengan persamaan trigonometri seperti ilustrasi pada gambar berikut.

Dari ilustrasi pada Gambar 2.3. maka koordinat dari robot dapat kita ketahui dengan persamaan (2.6) dan persamaan (2.7) berikut.



Gambar 2. 3 Ilustrasi pada Sumbu Cartesian

$$X_{pos} = distance \cos (\theta) \quad (2.6)$$

$$Y_{pos} = distance \sin (\theta) \quad (2.7)$$

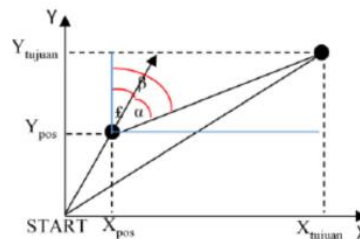
Untuk menentukan error arah hadap dari robot terhadap titik tujuan maka digunakan teorema pythagoras yang akan menghasilkan posisi (path) saat ini dan jarak terhadap titik tujuan seperti persamaan (2.8) sampai persamaan (2.10) berikut.

$$x = X_{tujuan} - X_{pos} \quad (2.8)$$

$$y = Y_{tujuan} - Y_{pos} \quad (2.9)$$

$$target_distance = \sqrt{x^2 + y^2} \quad (2.10)$$

Heading dari robot yang telah diketahui sehingga kita dapat menghitung error arah hadap (*heading error*) robot terhadap titik tujuan dengan persamaan (2.11) berikut serta ilustrasi yang ditunjukkan pada Gambar 2. 4 berikut.



Gambar 2. 4 Ilustrasi Sudut α , β , dan ϵ

Pada Gambar 2. 4 menunjukkan ilustrasi untuk mencari heading error (α) dimana β adalah target bearing yaitu sudut antara posisi robot saat ini terhadap titik tujuan. Sedangkan garis berwarna biru adalah garis bantu yang masing-masing sejajar dengan sumbu X dan sumbu Y. Untuk mendapat nilai dari β , maka digunakan persamaan sebagai berikut.

$$\beta = \arctan \frac{(Y_{Tujuan} - Y_{pos})}{(X_{Tujuan} - X_{pos})} \quad (2.11)$$

Sehingga heading error dapat dicari dengan menggunakan persamaan berikut.

$$\alpha = \beta - \epsilon \quad (2.12)$$

Untuk mengestimasi posisi *mobile robot* metode *odometry* menggunakan persamaan sebagai berikut.

$$x(t + \Delta t) = x(t) + v \cdot \cos(\theta(t)) \cdot \Delta t \quad (2.13)$$

$$y(t + \Delta t) = y(t) + v \cdot \sin(\theta(t)) \cdot \Delta t \quad (2.14)$$

$$\theta(t + \Delta t) = \theta(t) + \omega \cdot \Delta t \quad (2.15)$$

2.3 *Mobile Robot*

Mobile robot adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan-badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain. (Khairunnas & Mulyanto, 2021)

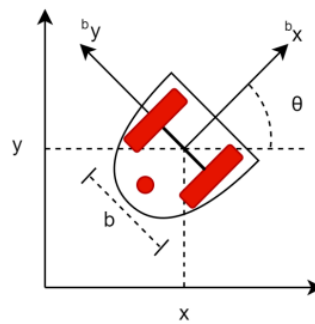
Mobile robot adalah robot yang bagiannya mempunyai aktuator. Sementara menurut Widodo Budiharto (2010:45) menyatakan bahwa aktuator adalah bagian peranti elektronik yang menghasilkan daya gerakan. (Hidayat & Hidayah, 2020)

2.4 *Differential Drive Kinematics*

Sebuah *mobile robot* dengan penggerak diferensial seperti yang diilustrasikan pada Gambar 2.5 memiliki dua roda tetap roda dengan sumbu rotasi yang sama dan roda kastor dengan fungsi menjaga menjaga

keseimbangan robot secara statis.(Angga Priyatmoko, Igi Ardiyanto, 2024)

Kedua roda tetap dikendalikan melalui sudut kecepatannya, sedangkan roda kastor bersifat pasif. *Differential drive kinematics* merupakan sebuah metode untuk menentukan kecepatan *mobile robot* atau kecepatan roda berdasarkan parameter dimensi robot serta jarak dan orientasi yang perlu ditempuh robot dalam lintasan.



Gambar 2. 5 Differential Drive *Mobile Robot*

Persamaan (2.16) menyatakan gerakan translasi dan rotasi robot dalam sistem koordinat sebagai fungsi dari kecepatan translasi v dan kecepatan angular ω serta perubahan posisi robot pada sumbu x dan y .

$$\begin{cases} \dot{x} = v \cos(\theta) \\ \dot{y} = v \sin(\theta) \\ \dot{\theta} = \omega \end{cases} \quad (2.16)$$

Pada persamaan (2.17) kecepatan linier v dan kecepatan angular ω dihitung berdasarkan *distance error* dan *heading error*.

$$\begin{aligned} v &= K_p \cdot \text{Distance error} \\ \omega &= K_p \cdot \theta \text{ error} \end{aligned} \quad (2.17)$$

Kecepatan angular robot roda kanan dan roda kiri didapatkan dengan persamaan (2.18).

$$\begin{aligned}\omega_R &= \frac{v + \omega \cdot b/2}{r} \\ \omega_L &= \frac{v - \omega \cdot b/2}{r}\end{aligned}\quad (2.18)$$

Dengan menggabungkan persamaan v dan ω dari roda kanan dan kiri ke dalam model kinematika, diperoleh persamaan baru yang mengaitkan langsung kecepatan sudut roda kanan ω_R dan roda kiri ω_L dengan perubahan posisi robot pada persamaan (2.19).

$$\begin{cases} \dot{x} = \frac{r}{2} \cos(\theta)(\omega_r + \omega_l) \\ \dot{y} = \frac{r}{2} \sin(\theta)(\omega_r + \omega_l) \\ \dot{\theta} = \frac{r}{b} (\omega_r - \omega_l) \end{cases} \quad (2.19)$$

2.5 Motor DC

Motor DC merupakan suatu perangkat elektronis pertama yang digunakan dalam sejarah industri perkembangan sistem transmisi daya untuk mengubah besaran listrik menjadi besaran mekanik. Prinsip kerjanya apabila motor diberi daya maka arus akan mengalir ke dalam motor dan menghasilkan torsi putar yang sebanding dengan arusnya. Model motor DC yang digunakan adalah representasi fungsi alih (transfer function). Input sistem adalah tegangan yang diberikan pada motor DC yaitu V . Sedangkan output sistem berupa kecepatan sudut motor DC yaitu omega (ω). (MA'ARIF et al., 2021)

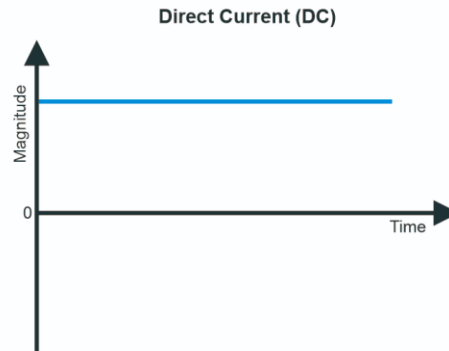
Motor *Direct Current* (DC) adalah perangkat yang merubah energi listrik DC menjadi energi mekanik berupa putaran. Motor DC adalah salah satu aktuator yang banyak digunakan pada dunia industri dan robotika.(Putri, 2022)

Cara kerja sebuah motor listrik mengubah energi listrik menjadi energi mekanik. Motor DC adalah jenis motor listrik yang penggunaannya memerlukan jenis arus DC atau arus searah. Jadi pada motor DC, arus searah yang dihasilkan akan diubah menjadi energi mekanis yang berupa putaran atau gerak. Jumlah putaran yang dihasilkan oleh motor tersebut disebut sebagai RPM (*Revolutions Per Minute*). Untuk sebuah motor DC kecil-standar, biasanya putaran yang dihasilkan adalah gerakan dengan kecepatan sekitar 100-5000 RPM. Dan biasanya juga memiliki tegangan operasional dengan kisaran sebesar 1,5 sampai dengan 24 volt.

Gambar 2.6 menunjukkan bentuk gelombang arus searah. Arus DC memiliki karakteristik utama yaitu nilai besar (magnitude) yang konstan terhadap waktu, Artinya arus mengalir hanya dalam satu arah dan tidak berubah-ubah, berbeda dengan arus bolak-balik (AC) yang nilai dan arahnya berubah secara periodik.

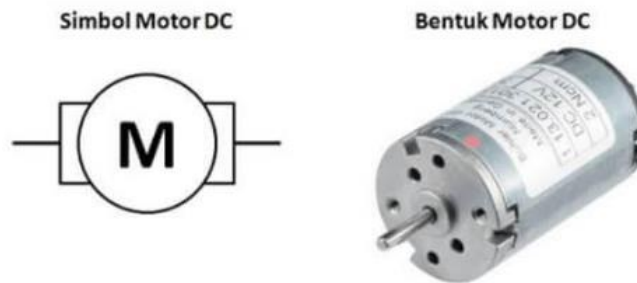
Pada motor DC, arus searah digunakan untuk menggerakkan motor. Ketika arus DC dialirkan ke lilitan kumparan motor, medan magnet yang dihasilkan akan bereaksi dengan medan magnet stator (tetap), menghasilkan torsi yang memutar rotor. Salah satu komponen penting dalam motor DC adalah komutator, yaitu mekanisme yang bertugas membalik arah arus pada

kumparan setiap setengah putaran, sehingga arah torsi tetap konstan dan motor dapat terus berputar dalam satu arah.

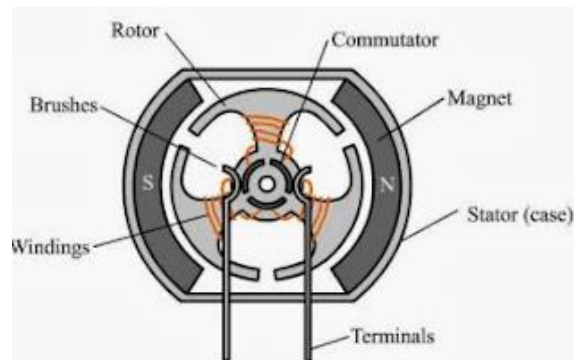


Gambar 2. 6 Prinsip Arus Searah

Motor DC memiliki beberapa bagian bagian penting. Pada Gambar 2. 7 menunjukkan bagian-bagian dari sebuah motor DC (motor arus searah) yang bekerja berdasarkan prinsip elektromagnetik. Motor ini terdiri dari rotor, yaitu bagian berputar yang dililitkan dengan kawat tembaga (*windings*) untuk menghasilkan medan magnet saat dialiri arus listrik. Rotor berinteraksi dengan medan magnet yang dihasilkan oleh magnet permanen pada stator, yaitu bagian diam dari motor yang juga berfungsi sebagai kerangka pelindung. Arus listrik dialirkan ke rotor melalui komponen sikat (*brushes*) yang terbuat dari karbon atau logam dan bersentuhan langsung dengan komutator, yaitu cincin yang bertugas membalik arah arus pada lilitan rotor sehingga putaran motor tetap kontinu. Terminal pada motor digunakan untuk menghubungkannya dengan sumber daya arus searah (DC). Motor DC ini menghasilkan rotasi yang stabil dengan memanfaatkan interaksi medan magnet dari rotor dan stator sesuai dengan prinsip gaya Lorentz.(Anwar & Ferdilla, 2023)



Gambar 2. 7 Bagian Motor DC

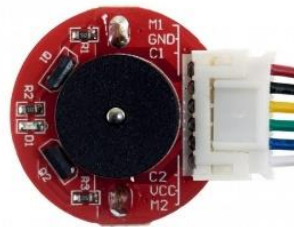


Gambar 2. 8 Bagian-bagian Penting Motor DC

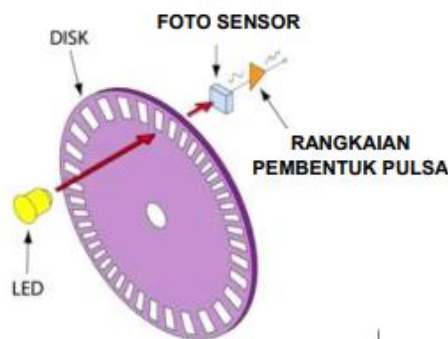
2.6 Encoder

Encoder merupakan perangkat elektromekanis putar yang mengubah putaran pada porosnya menjadi serangkaian output pulsa. Bentuk fisik dari encoder dapat dilihat pada Gambar 2. 9 Komponen-komponen di dalam encoder, dapat dilihat pada Gambar 2. 10, secara umum terdiri dari LED sebagai sumber cahaya, piringan berlubang (*Disk*) sebagai pelalu cahaya LED, foto sensor sebagai pendeteksi cahaya yang lewat celah piringan dan rangkaian pembentuk gelombang kotak atau pulsa sebagai output dari encoder. LED memancarkan cahaya ke piringan berlubang, dan di sebalik piringan tersebut terdapat foto sensor yang akan menerima cahaya dari LED yang melewati celah lubang piringan. Perubahan dari kondisi ada cahaya ke kondisi tidak ada

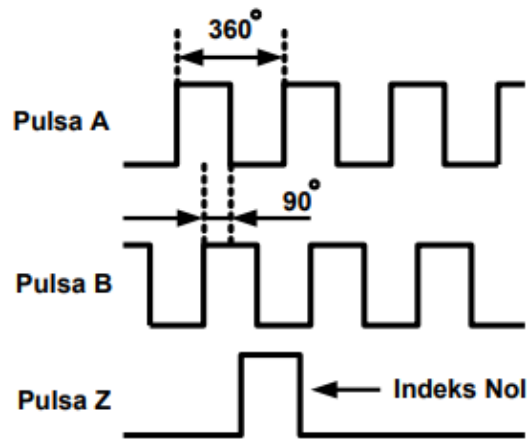
cahaya yang diseleksi oleh piringan berlubang tersebut akan menyebabkan rangkaian pemroses fotosensor mengeluarkan pulsa. *Encoder* umumnya memiliki dua sinyal keluaran pada kanal A dan B, yang mengeluarkan sinyal berupa pulsa apabila poros encoder tersebut diputar. Ada kalanya encoder juga memiliki satu kanal tambahan sebagai Indeks, yaitu kanal Z yang digunakan sebagai referensi saat poros encoder menunjuk pada posisi tertentu, contoh sebagai referensi posisi awal (titik nol) sebelum poros bergerak. Keluaran sinyal pulsa pada kanal A berbeda fasa 90 derajat terhadap pulsa pada kanal B, sedangkan kanal Z mengeluarkan hanya satu pulsa untuk setiap satu putaran. Gambar pulsa pada kanal A, B dan Z dapat dilihat pada Gambar 2.11.(Supriyo et al., 2024)



Gambar 2. 9 Encoder



Gambar 2. 10 Struktur Sederhana Encoder



Gambar 2. 11 Output Kanal A, B, Z

2.7 Driver Motor

Driver motor adalah rangkaian berbasis transistor yang digunakan untuk mengontrol pergerakan motor DC, termasuk arah dan kecepatannya. Salah satu driver motor yang populer adalah L298N, yang memanfaatkan IC L298 berbasis *H-Bridge*. Modul ini dirancang untuk mengatur arah putaran motor melalui prinsip kerja H-Bridge dan mengendalikan kecepatannya menggunakan sinyal PWM.

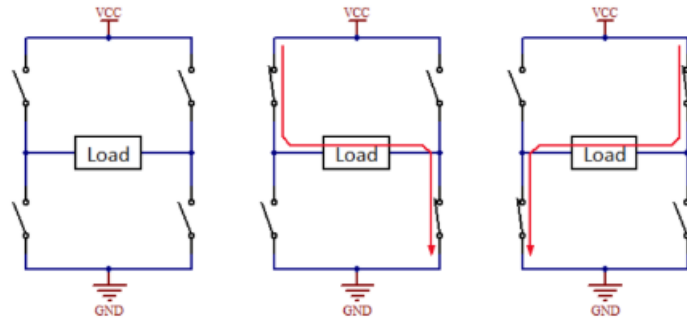
IC L298, yang merupakan inti dari driver ini, terdiri dari rangkaian transistor logika (TTL) dengan gerbang NAND. Prinsip *H-Bridge* memungkinkan pengendalian aliran arus dengan mengubah polaritas tegangan input, sehingga arah putaran motor dapat diatur. Setiap H-Bridge dikendalikan oleh sinyal tegangan logika TTL dari mikrokontroler.(Peerzada et al., 2021)

L298N mampu menangani tegangan hingga 46 VDC dan arus maksimal 2A per kanal, sehingga dapat digunakan untuk mengontrol dua motor DC secara bersamaan. Dalam konfigurasi *H-Bridge*, terdapat empat elemen switching berupa transistor (baik BJT maupun MOSFET), dengan motor ditempatkan di tengah, membentuk pola menyerupai huruf "H".

Gambar 2. 12 menunjukkan tiga kondisi kerja sakelar pada rangkaian H-Bridge. Pada kondisi pertama (sebelah kiri), semua sakelar dalam posisi terbuka sehingga tidak ada arus yang mengalir ke terminal motor. Akibatnya, motor dalam keadaan berhenti atau tidak beroperasi.

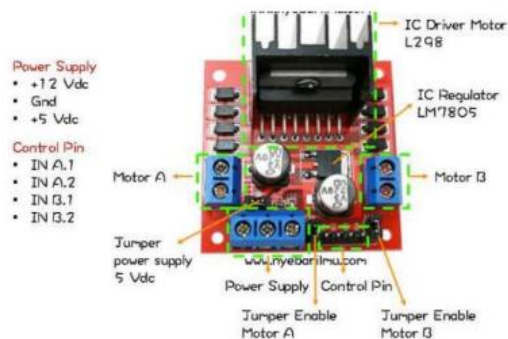
Pada kondisi kedua (tengah), sakelar bagian atas kiri dan sakelar bagian bawah kanan tertutup. Dalam keadaan ini, terminal kiri motor mendapatkan tegangan positif, sementara terminal kanan mendapatkan tegangan negatif. Hal ini menyebabkan motor berputar searah jarum jam.

Pada kondisi ketiga (sebelah kanan), sakelar bagian atas kanan dan sakelar bagian bawah kiri tertutup. Dengan konfigurasi ini, terminal kanan motor menerima tegangan positif, sedangkan terminal kiri menerima tegangan negatif. Akibatnya, motor berputar berlawanan arah jarum jam.



Gambar 2. 12 Rangkaian H-Bridge

Pada Gambar 2. 13 menunjukkan modul driver motor DC L298N beserta komponen-komponen utamanya. Modul ini memiliki terminal *power supply* yang terdiri dari +12 Vdc untuk tegangan utama motor, GND sebagai ground, dan +5 Vdc yang dapat berfungsi sebagai output tegangan 5V dari IC regulator LM7805 atau sebagai input tegangan 5V dari sumber eksternal jika jumper dilepas. Terdapat *control pin* berupa IN A1 dan IN A2 yang digunakan untuk mengontrol arah putaran motor A, serta IN B1 dan IN B2 untuk mengontrol arah putaran motor B, di mana logika *high/low* menentukan arah putaran masing-masing motor.



Gambar 2. 13 Driver Motor

Modul ini juga memiliki terminal output untuk motor, yaitu Motor A untuk motor pertama dan Motor B untuk motor kedua. Pada bagian jumper *power supply* 5 Vdc, pengguna dapat memilih untuk menggunakan tegangan 5V dari regulator internal LM7805 dengan memasang jumper, atau menggunakan tegangan eksternal dengan melepas jumper tersebut. Selain itu, terdapat jumper enable motor, yaitu *Jumper Enable Motor A* dan *Jumper Enable Motor B*, yang digunakan untuk mengaktifkan masing-masing motor. Jika jumper dilepas, pin *enable* dapat dikontrol langsung melalui mikrokontroler.

Komponen utama pada modul ini adalah IC driver motor L298, yang berfungsi untuk mengontrol motor DC menggunakan prinsip H-Bridge, dan IC regulator LM7805, yang berfungsi mengubah tegangan input 12V menjadi 5V untuk kebutuhan sirkuit logika. Modul ini dirancang untuk mengendalikan dua motor DC secara simultan dengan fleksibilitas dalam konfigurasi sumber daya dan kontrol.

2.8 *Pulse With Modulation*

Pulse Width Modulation (PWM), atau modulasi lebar pulsa, adalah teknik untuk mengatur lebar pulsa pada sinyal digital dengan nilai frekuensi dan amplitudo yang tetap. PWM digunakan untuk menghasilkan sinyal analog dari perangkat digital seperti mikrokontroler. Salah satu aplikasi utamanya adalah mengatur kecepatan motor DC dengan tegangan tetap. Teknik ini

dilakukan dengan mengubah kondisi sinyal antara ON dan OFF secara bergantian pada frekuensi tertentu.(Riski Aldiani et al., 2024)

Siklus kerja PWM dikenal sebagai *duty cycle*, yang merepresentasikan durasi sinyal dalam kondisi ON dibandingkan dengan total waktu dalam satu periode. *Duty cycle* dapat dihitung dengan persamaan (2.20).

$$\text{Duty Cycle} = \frac{t_{on}}{t_{total}} \times 100\% \quad (2.20)$$

Frekuensi PWM (F) menunjukkan kecepatan penyelesaian satu periode PWM. Hubungan antara frekuensi dan waktu periode dapat dihitung menggunakan persamaan (2.21).

$$F = \frac{1}{t_{total}} \quad (2.21)$$

Tegangan rata-rata yang dihasilkan oleh sinyal PWM dipengaruhi oleh nilai *duty cycle*. Tegangan keluaran (V_{out}) dapat dihitung dengan persamaan (2.22)

$$V_{out} = \text{Duty Cycle} \times V_{in} \quad (2.22)$$

Semakin besar nilai *duty cycle*, semakin besar tegangan rata-rata yang diterima oleh motor DC, sehingga menghasilkan kecepatan motor yang lebih tinggi. Sebaliknya, *duty cycle* yang lebih kecil menghasilkan tegangan rata-rata yang lebih rendah, menyebabkan kecepatan motor berkurang. Dengan demikian, PWM menjadi solusi efisien untuk mengatur kecepatan motor dalam berbagai aplikasi kontrol berbasis mikrokontroler.

2.9 Arduino

Arduino dikatakan sebagai sebuah platform dari *physical computing* yang bersifat *open source*. Pertama-tama perlu dipahami bahwa kata “platform” disini adalah sebuah pilihan kata yang tepat. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah kombinasi dari hardware, bahasa pemrograman dan *Integrated Development Environment* (IDE) yang canggih. Penggunaan arduino adalah untuk memproses, mengolah data memproses dan menampilkan pada komputer. (Raharja, 2024)

2.8.1 Arduino IDE

Arduino IDE adalah software yang digunakan untuk membuat sketch pemrograman atau dengan kata lain arduino IDE sebagai media untuk pemrograman pada board yang ingin diprogram. Arduino IDE ini berguna untuk mengedit, membuat, meng-upload ke board yang ditentukan, dan meng-coding program tertentu. Arduino IDE dibuat dari bahasa pemrograman JAVA, yang dilengkapi dengan *library C/C++(wiring)*, yang membuat operasi input/output lebih mudah.

Software Arduino Ide ini, kita dapat memodelkan sifat dari parameter rangkaian analog dan digital. Kemampuan yang disediakan Arduino Ide adalah dapat memodelkan berbagai rancangan rangkaian, menguji suatu rangkaian dengan berbagai kemungkinan komponen, memeriksa sifat dari keseluruhan rangkaian dengan melakukan analisis AC / DC atau *transient*.(Mahanin Tyas et al., 2023)



Gambar 2. 14 Tampilan Software Arduino IDE

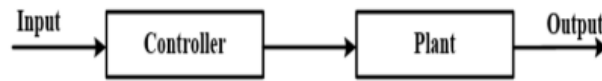
2.10 Sistem Kontrol

Sistem kontrol pada *mobile robot* dapat dibagi menjadi dua jenis, yaitu sistem *open loop* dan *close loop*. Sistem *close loop* biasanya diterapkan untuk mengendalikan posisi *mobile robot* dengan tipe penggerak diferensial, yang bertujuan untuk menuju atau mencapai suatu target tertentu (sasaran atau titik akhir). Di sisi lain, sistem *open loop* umumnya digunakan untuk mengontrol kecepatan roda, sehingga dapat mengatur pergerakan roda kanan dan kiri secara independen.

2.9.1 Sistem Kontrol Open Loop

Sistem kontrol *open loop* merupakan sistem kontrol di mana output tidak memengaruhi aksi kontrol yang dilakukan. Sistem ini sering disebut sebagai *feed-forward control*, karena outputnya tidak dihitung ulang, meskipun robot telah mencapai posisi target yang telah ditentukan seperti yang ditunjukkan pada Gambar 2. 15. Dalam sistem kontrol *open loop*, untuk mengurangi kesalahan pada *output*, sistem ini memanfaatkan hasil kalibrasi dengan menganalisis hubungan antara input dan output. Kontrol ini cocok

untuk sistem operasi robot yang menggunakan aktuator yang bekerja berdasarkan logika langkah-langkah terkonfigurasi secara berurutan.



Gambar 2. 15 Sistem Kontrol Open Loop

Kelebihan dan Kekurangan sistem kontrol *open loop* sebagai berikut:

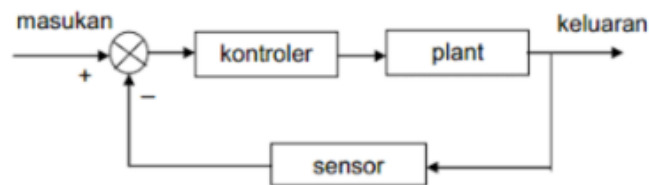
Tabel 2. 1 Kekurangan dan Kelebihan Sistem Open Loop

Sistem Kontrol Open Loop		
No.	Kelebihan	Kekurangan
	Desain lebih sederhana dan mudah diimplementasikan.	Tidak memiliki mekanisme feedback, sehingga kesalahan tidak dapat dikoreksi secara otomatis.
2.	Biaya lebih rendah karena tidak membutuhkan sensor atau komponen umpan balik.	Akurasi bergantung pada kalibrasi awal dan lingkungan kerja, sehingga kurang andal dalam kondisi variabel.
3.	Cepat dalam memberikan respons karena tidak ada siklus umpan balik.	Tidak cocok untuk sistem yang membutuhkan presisi tinggi atau respons terhadap perubahan beban.

4.	Konsumsi daya lebih rendah karena kontrol dilakukan tanpa banyak komponen tambahan.	
----	---	--

2.9.2 Sistem Kontrol *Close Loop*

Sistem kontrol close loop merupakan sistem kontrol di mana outputnya memengaruhi aksi kontrol yang dilakukan. Hasil keluaran dari sistem ini akan menjadi umpan balik bagi input selanjutnya, seperti yang digambarkan alurnya pada Gambar 2. 16. Hasilnya, sistem terus memberikan feedback hingga keluaran sesuai dengan yang diharapkan.



Gambar 2. 16 Sistem Kontrol Close Loop

Proses feedback akan berhenti ketika output telah mencapai nilai yang diinginkan. Sistem kontrol close loop memiliki beberapa fungsi utama yang terbagi menjadi beberapa komponen, yaitu:

1. **Input:** Masukan yang diberikan ke sistem, berupa variabel nilai yang akan dikontrol.
2. **Output:** Keluaran dari sistem kontrol yang menjadi nilai target untuk dipertahankan dan juga digunakan sebagai umpan balik untuk siklus berikutnya.
3. **Plant:** Beban atau objek yang akan dikendalikan dalam sistem.

4. Controller: Alat kontrol yang berfungsi sebagai pengendali *plant* atau beban dalam sistem.
5. Elemen Umpan Balik: Komponen yang mengirimkan nilai output ke *detector* agar dapat dibandingkan dengan nilai yang diinginkan.
6. *Error Detector*: Bagian yang mendeteksi kesalahan dengan membandingkan selisih antara nilai input dan umpan balik dari output.
7. Gangguan: Sinyal tambahan yang tidak diinginkan, yang dapat menyebabkan keluaran berbeda dari hasil yang diharapkan. Gangguan ini biasanya terjadi akibat perubahan pada beban sistem.

Kelebihan dan Kekurangan sistem kontrol *Closed loop* sebagai berikut:

Tabel 2. 2 Kelebihan dan Kekurangan Sistem Close Loop

Sistem Kontrol <i>Close Loop</i>		
No.	Kelebihan	Kekurangan
1.	Mampu mengoreksi kesalahan secara otomatis melalui umpan balik.	Desain lebih kompleks, membutuhkan sensor, dan algoritma kontrol yang lebih rumit.
2.	Lebih andal dalam menghadapi gangguan atau perubahan beban.	Biaya implementasi lebih tinggi karena memerlukan tambahan perangkat keras seperti sensor.
3.	Akurasi lebih tinggi karena keluaran selalu dikontrol agar sesuai dengan nilai yang diinginkan.	Membutuhkan waktu lebih lama dalam proses pengendalian karena adanya siklus umpan balik.

4.	Cocok untuk aplikasi yang memerlukan presisi tinggi, seperti robotik atau otomasi industri.	Potensi overcompensation atau osilasi jika parameter kontrol tidak diatur dengan baik.
----	---	--

2.9.3 Sistem Kontrol PID

Sistem Kontrol PID (*Proportional-Integral-Derivative*) adalah metode pengendalian sistem yang menggunakan tiga parameter utama untuk mengatur respon atau keluaran sistem agar mencapai nilai yang diinginkan dengan presisi tinggi. Pada Gambar 2.17 menunjukkan Ketiga parameter *Proportional* (P), *Integral* (I), dan *Derivative* (D), yang bekerja secara bersama-sama untuk meminimalkan kesalahan (*error*) antara respon aktual dan nilai referensi atau setpoint. PID adalah salah satu metode kontrol yang paling banyak digunakan karena kesederhanaan dan fleksibilitasnya dalam berbagai aplikasi, termasuk robotika, proses industri, dan sistem mekanik. Persamaan kontrol PID untuk menghitung sinyal kendali (*control output*) $u(t)$ digunakan persamaan

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (2.16)$$

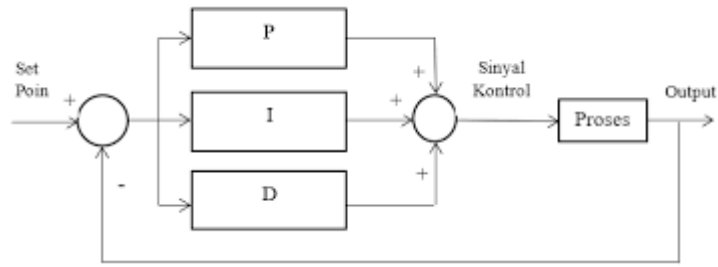
Dimana :

$e(t)$: error antara nilai setpoint dan respon aktual

K_p : konstanta proporsional

K_i : konstanta integral

K_d : konstanta derivative



Gambar 2. 17 Diagram Block Kontrol PID

1. Komponen *Proportional*

Komponen ini menghasilkan sinyal kendali yang proporsional terhadap besarnya error. Semakin besar error, semakin besar juga sinyal kendali yang dihasilkan. Fungsi utama komponen ini adalah mempercepat respons sistem untuk mendekati setpoint. Namun, jika hanya komponen P yang digunakan, sistem dapat mengalami osilasi.

2. Komponen *Integral*

Komponen ini berfungsi untuk menghilangkan error tetap dengan mengakumulasi kesalahan selama periode waktu tertentu. Dengan menambahkan kontribusi dari komponen I, sistem mampu mencapai setpoint meskipun terdapat gangguan.

3. Komponen *Derivative*

Komponen ini berfungsi untuk mengantisipasi perubahan error di masa depan dengan melihat laju perubahan error. Komponen D membantu meningkatkan stabilitas sistem dan mengurangi osilasi, terutama dalam sistem yang dinamis.

2.9.4 Metode Tuning PID Ziegler Nichols

Metode Ziegler-Nichols *step response* merupakan salah satu pendekatan dalam melakukan tuning parameter kontroler PID (Proportional-Integral-Derivative). Metode ini digunakan ketika sistem memiliki respons terhadap perubahan mendadak (*step input*) yang dapat digambarkan dalam bentuk kurva S atau *S-shaped curve*.

Dalam metode ini, dilakukan identifikasi dua parameter utama dari kurva respons: waktu tunda (L – *dead time*) dan konstanta waktu (T – *time constant*), yang diperoleh dari analisis titik tangensial pada kurva. Setelah nilai L dan T didapatkan, parameter PID (K_p , K_i , K_d) dapat dihitung menggunakan rumus yang telah ditentukan oleh Ziegler dan Nichols berdasarkan jenis kontrol yang digunakan (P, PI, atau PID) yang ditunjukkan pada Tabel 2. 3. Meskipun metode ini cukup sederhana dan cepat, akurasinya tergantung pada linearitas sistem serta kualitas kurva respons yang diperoleh. Namun, metode ini tetap banyak digunakan karena memberikan titik awal tuning yang cukup baik dalam berbagai aplikasi kendali.

Tabel 2. 3 Parameter Metode *Ziegler Nichols*

Pengendali	K_p	T_i	T_d
P	$\frac{T}{L}$	–	0
PI	$0,9 \frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{T}{L}$	$2L$	$0,5L$

2.11 Penelitian Terkait

Tabel 2. 4 Penelitian Terkait

No.	Judul	Penulis	Tahun	Pembahasan
1.	Heading Control on Differential Drive Wheeled Mobile Robot with <i>Odometry</i> for Tracking Problem	F Fahmizal, A Priyatmoko, E Apriaskar, A Mayub	2019	Penelitian ini membahas desain dan implementasi sistem kontrol heading untuk robot mobile beroda differential drive menggunakan odometri dalam konteks pelacakan trajektori. Penelitian ini mengidentifikasi masalah umum yang muncul selama proses pelacakan, seperti kesalahan heading yang sering disebabkan oleh selip pada roda robot atau ketidakteraturan kecepatan motor DC.
2.	Dynamic <i>Odometry</i> for Waypoint Tracking of Omnidirectional Mecanum Wheeled Mobile Robot for Efficient Movement	A Priyatmoko, I Ardiyanto, AI Cahyadi	2022	Penelitian ini membahas pengembangan sistem odometri dinamis yang dirancang untuk meningkatkan pelacakan waypoint pada robot mobile beroda Mecanum yang omnidirectional. Penelitian ini menyoroti pentingnya akurasi dalam navigasi robot, terutama dalam lingkungan yang kompleks dan terbatas. Dengan memanfaatkan algoritma odometri yang adaptif, jurnal ini menunjukkan bagaimana robot dapat

				secara efisien menghitung posisi dan orientasi saat bergerak menuju titik waypoint yang ditentukan.
3.	Differential Drive Mobile Robot Motion Accuracy Improvement with <i>Odometry</i> -Compass Sensor Fusion Implementation	S Sungkono, G Al Azhar, AC Kusuma, S Izza	2023	Jurnal ini membahas implementasi sensor fusion antara odometri roda dan sensor kompas untuk meningkatkan akurasi gerakan robot mobile differential drive. Hasil percobaan menunjukkan bahwa penggunaan sensor fusion berhasil mengurangi kesalahan posisi rata-rata menjadi di bawah 0,15 meter.
4.	Trajectory Planning and Tracking Control of a Differential-Drive Mobile Robot in a Picture Drawing Application	Ching-Long Shih and Li- Chen Lin	2017	Jurnal "Trajectory Planning and Tracking Control of a Differential-Drive Mobile Robot in a Picture Drawing Application" mengeksplorasi pengembangan robot mobile diferensial yang mampu menggambar berdasarkan citra digital. Penelitian ini fokus pada perencanaan trajektori dan kontrol pelacakan untuk memastikan robot dapat mengikuti kontur gambar dengan akurasi tinggi. Dengan menggunakan teknik pemodelan kinematik dan algoritma kontrol, robot menunjukkan

				kesalahan pelacakan rata-rata hanya 1.0 mm, serta waktu menggambar yang bervariasi tergantung pada kompleksitas gambar. Hasilnya menunjukkan kemampuan robot dalam mereproduksi detail gambar secara efektif, yang membuka peluang untuk aplikasi lebih lanjut dalam bidang seni dan robotika kreatif.
--	--	--	--	---