

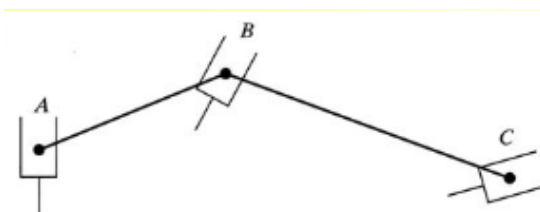
BAB II

TINJAUAN PUSTAKA

2.1 *Trajectory tracking*

Trajectory merupakan metode untuk menentukan pola gerakan dari sebuah robot dari titik awal menuju titik akhir. Algoritma dari *trajectory tracking* dikategorikan menjadi dua macam, yaitu *off-line path trajectory* dan *on-line trajectory*. *Off-line path trajectory* menghitung keseluruhan lintasan yang akan ditempuh sebelum terjadi pergerakan, sedangkan *on-line path trajectory* menghasilkan lintasan yang ditempuh dan terus bertambah selama adanya pergerakan (Edy Surya Prabowo et al., 2022).

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 seperti pada gambar 2.1 dibawah ini.



Gambar 2. 1 Titik jalur lintasan

Gambar 2.1 menunjukkan ilustrasi pola lintasan terpendek dari titik A menuju titik B, dan lintasan terpanjangnya titik A menuju titik C.

Secara umum metode *offline trajectory* adalah perencanaan yang dilakukan secara offline untuk menghitung seluruh jalur atau lintasan menuju tujuan sebelum gerakan dimulai (Shiller, 2015), pendeteksian lingkungan dimana suatu objek atau kendaraan akan bernavigasi sesuai dengan informasi algoritmanya. Sedangkan lintasan menuju tujuan dilakukan secara *online* selama pergerakan, untuk memungkinkan robot bereaksi terhadap perubahan lingkungan menuju target, dan terhadap kesalahan yang ditemui, dengan demikian jalur bebas dapat diselesaikan.

Trajectory tracking hanya signifikan dalam satu kali keadaan atau percobaan. *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 (Al Azhar et al., 2022).

2.2 *Mobile Robot*

Mobile Robot adalah sebuah robot yang mempunyai ciri khas dengan *actuator* berupa roda untuk menggerakkan seluruh badan robot, sehingga *Mobile Robot* tersebut mampu berpindah-pindah posisi dari satu titik koordinat ke titik koordinat lainnya (Zamronnan et al., 2021). Dalam

melakukan perpindahan *Mobile Robot* ini akan melakukan navigasi sesuai dengan *drive* yang mengendalikannya, pengendalian *Mobile Robot* dengan kendali dua roda di bagian kiri dan kanan disebut *Robot Diverential Drive*. *Robot diverential drive* menggerakkan roda dengan mengubah arah dari memvariasikan tingkatan rotasi relatif rodanya, sehingga masing-masing roda dapat dikontrol untuk berputar searah jarum jam maupun berlawanan jarum jam.

2.3 *Odometry*

Metode yang sering digunakan dalam menentukan posisi *Mobile Robot* secara *real time* salah satunya adalah metode *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 (Muldayani et al., 2021).

Robot mengandalkan sensor-sensor untuk membaca dan mengingat posisi robot dan sudut orientasi robot dari waktu ke waktu secara waktu nyata berdasarkan pada sistem *odometry*. *Odometry* mengestimasi perubahan posisi robot setiap saat saat bergerak berdasarkan pada data hasil pembacaan sensor-sensor posisi dan arah pergerakan robot, sehingga posisi robot dari titik awalnya dapat terpantau setiap saat. Sensor-sensor umum dipakai dalam sistem *odometry* adalah encoder yang terpasang pada poros roda (Kurniawati, 2017).

Untuk mendapatkan posisi relatif robot perlu adanya perhitungan jumlah pulsa yang dihasilkan dari sensor *Rotary encoder*, juga diperlukan konstanta peubah untuk mengkonversi jumlah panjang lintasan yang akan di tempuh. Perhitungan posisi tersebut dilakukan terus menerus sampai mencapai posisi tujuan (Rifandi, 2022).

Konstanta peubah didapatkan dari persamaan rumus berikut:

$$C = \frac{\text{Keliling Roda } (K)}{\text{Resolusi Encoder } (Re)} \quad (2.1)$$

$$\text{Keliling Roda } (K) = 2 \times \pi \times r \quad (2.2)$$

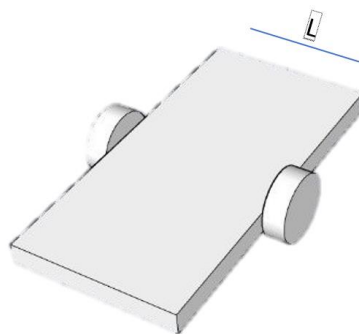
Keterangan:

C = Konstanta Peubah ($mm/counter$)

K = Keliling Roda (mm)

Re = Resolusi Encoder

Resolusi *encoder* adalah jumlah pulsa yang dihasilkan oleh *Rotary encoder* dalam satu putaran penuh. Jumlah pulsa dalam satu putaran sama dengan jumlah lubang pada piringan *encoder*.



Gambar 2. 2 Ilustrasi desain *Mobile Robot*

Gambar 2.2 menunjukkan ilustrasi mobile robot dengan roda diferensial drive. Untuk menghitung jarak tempuh roda menggunakan rumus berikut:

$$Drt = \sum crt \times C \quad (2.3)$$

$$Dlt = \sum clt \times C \quad (2.4)$$

$$D = \frac{Drt + Dlt}{2} \quad (2.5)$$

$$\theta = \frac{Drt - Dlt}{l} \quad (2.6)$$

Keterangan:

Drt = Jarak tempuh roda kanan

Dlt = Jarak tempuh roda kiri

$\sum crt$ = Jumlah counter roda kanan

$\sum clt$ = Jumlah counter roda kiri

D = Distance (jarak tempuh mobile robot)

l = Jarak antara kedua roda

θ = Sudut Orientasi

Karena θ masih dalam radian sedangkan sudut hadap robot ditentukan dalam derajat maka sudut hadap robot (*heading*).

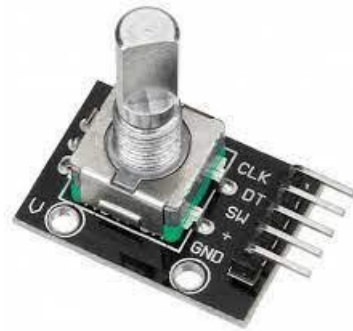
$$heading(\theta) = \theta \frac{180}{\pi} \quad (2.7)$$

dengan menyelesaikan persamaan diatas sehingga dapat mengetahui jarak tempuh dan sudut orientasi, maka dapat ditentukan koordinat posisi *Mobile Robot* dengan menggunakan persamaan trigonometri sebagai berikut:

$$x_{pos} = \text{jarak tempuh} \cdot \sin \theta \quad (2.8)$$

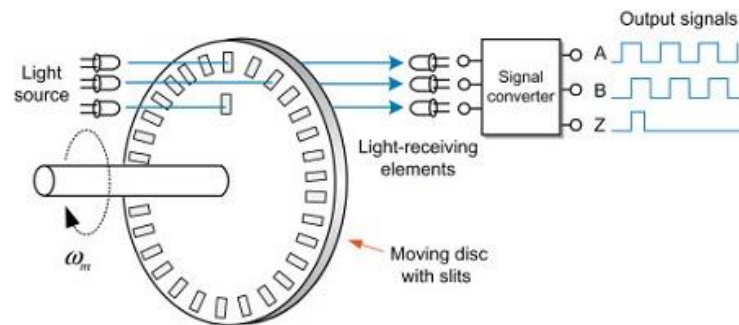
$$y_{pos} = \text{jarak tempuh} \cdot \cos \theta \quad (2.9)$$

2.4 Rotary encoder



Gambar 2. 3 Fisik *Rotary encoder*

Gambar 2.3 menunjukkan bentuk fisik *rotary encoder* dimana *Rotary encoder* adalah salah satu komponen elektromekanik dengan menggunakan sensor optik yang menghasilkan serial pulsa dan memiliki fungsi untuk mendeteksi gerakan, posisi, dan arah pada suatu pengendalian seperti robot, motor *drive*, dan lainnya. *Rotary encoder* digunakan untuk mengubah gerakan linear atau putaran menjadi sinyal digital putaran memonitor gerakan putar dari suatu alat (Zamronnan et al., 2021), dimana sensor *Rotary encoder* ini biasanya dipasang segaris dengan poros motor, *Gearbox*, sendi atau bagian putar lainnya. *Rotary encoder* akan membaca posisi sudut benda yang berputar kemudian dikonversi menjadi kode digital dalam bentuk pulsa elektrik, yang kemudian akan diteruskan ke rangkaian kendali untuk melakukan perintah berdasarkan kode yang diterimanya.



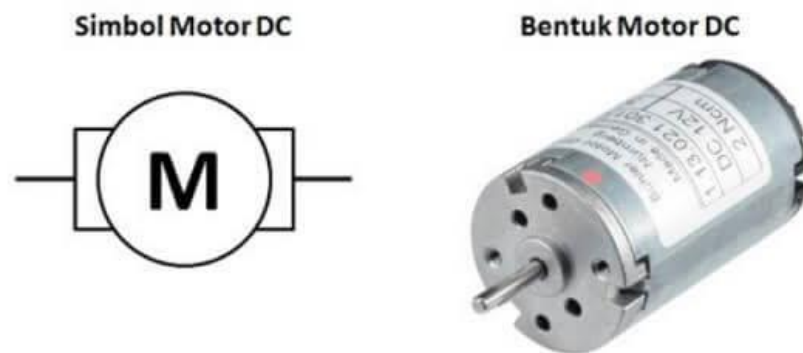
Gambar 2. 4 Penyusun *Rotary encoder*

Gambar 2.4 menjelaskan *Rotary encoder* tersusun dari 3 komponen yaitu, disk berlubang yang akan terhubung langsung dengan benda yang akan diukur, LED sebagai sumber cahaya, dan sensor optik *photo transistor* sebagai detektor cahaya. Sistem kerja dari ketiga komponen tersebut adalah sebagai penghasil pulsa pada *Rotary encoder*, dimana cahaya dari LED akan melewati dan menembus piringan berlubang yang kemudian terdeteksi oleh *photo transistor*. Karena cahaya yang menembus hanya pada lubang yang berputar sehingga membuat keluaran pulsa persegi menjadi 1 dan 0 sesuai dengan lubang dan kecepatan putaran motor yang diukur. Hasil dari keluaran pulsa tersebutlah yang nantinya akan diteruskan ke suatu sistem kendali sebagai informasi gerak, posisi sudut, dan kecepatan motor yang terhubung pada disk.

2.5 Motor DC

Motor DC adalah jenis motor listrik dengan suplai tegangan arus searah pada kumparan medan yang mengubah energi listrik menjadi energi gerak mekanik berupa putaran. Motor DC terdiri dari dua bagian yaitu

stator dan juga rotor. Stator merupakan bagian motor yang diam (badan motor atau kutub magnet berupa sikat-sikat), sedangkan rotor merupakan bagian yang bergerak (berputar) yang sering disebut juga sebagai jangkar lilitannya (Djahi et al., 2019).



Gambar 2. 5 Simbol dan Bentuk Motor DC

Gambar 2.5 menunjukkan simbol dan bentuk Motor DC yang berkerja berdasarkan prinsip gaya Lorentz yaitu ketika konduktor yang memiliki aliran arus diletakan dalam medan magnet maka akan timbul sebuah gaya secara ortogonal (berhubungan secara tegak lurus) diantara arah medan magnet dan aliran arus. Pada dasarnya motor DC menghasilkan tenaga mekanik berupa kecepatan atau putaran (berputar) yang dirumuskan sebagai berikut:

$$N = \frac{V_{TM} - I_A R_A}{K\phi} \quad (2.10)$$

Keterangan :

N = kecepatan motor

V_{TM} = tegangan terminal

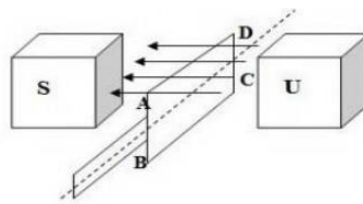
I_A = arus jangkar motor

R_A = hambatan jangkar motor

K = konstanta motor

\emptyset = fluks magnet yang terbentuk pada motor

Prinsip kerja motor DC



Gambar 2. 6 Kumparan pada motor

Gambar 2.6 adalah Kumparan ABCD terletak dalam medan magnet serba sama dengan kedudukan sisi aktif AD dan CB yang terletak tepat lurus arah fluks magnet. Sedangkan sisi AB dan DC ditahan pada bagian tengahnya, sehingga apabila sisi AD dan CB berputar karena adanya gaya lorentz, maka kumparan ABCD akan berputar.

Hasil perkalian jarak dengan gaya pada titik tertentu disebut dengan momen, pengaruh momen akan mengakibatkan perputaran sisi aktif AD dan CB. Setiap sisi kumparan aktif AD dan CB akan mengalami momen putar sebesar rumus dibawah.

$$T = f \cdot r \quad (2.11)$$

Keterangan:

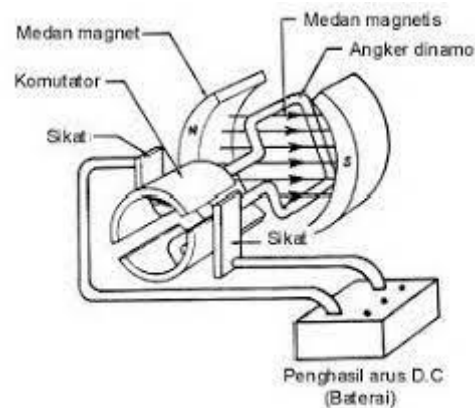
T = momen putar (N/m)

f = gaya tolak (Newton)

r = jarak sisi putaran pada sumbu putar (Meter)

Namun pada daerah kutub-kutub magnet besar momen putar akan tetap karena dipengaruhi besarnya gaya Lorentz, dengan demikian kedudukan garis netral sisi masing-masing kumparan akan berhenti berputar. Agar motor terus berputar dengan baik maka solusinya dengan ditambahkan jumlah kumparan. Kumparan-kumparan tersebut dililitkan pada suatu jangkar sehingga disebut jangkar lilitan, kumparan yang dililitkan dan diletakan sedemikian rupa akan menghasilkan putaran yang baik.

Motor DC memiliki tiga bagian utama untuk dapat berputar diantaranya:



Gambar 2. 7 Bagian Motor DC

Gambar 2.7 menunjukkan bagian-bagian dari motor DC, yaitu sebagai berikut:

1. Kutub Medan, kutub medan ini terdiri dari dua kutub yaitu, kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi ruang terbuka dari kutub utara ke kutub selatan, biasanya terdapat satu atau bahkan lebih elektromagnetik tergantung dengan ukuran

dan jenis motornya. Lebih kompleks dan lebih besar motor maka akan lebih banyak elektromagnet.

2. Kumparan motor DC, kumparan motor DC ini biasanya berbentuk silinder yang dihubungkan ke as penggerak untuk menggerakkan beban. Cara kerjanya jika arus masuk menuju kumparan maka arus tersebut akan menjadi elektromagnet, untuk kasus motor DC berukuran kecil, kumparannya berputar dalam medan magnet yang dibentuk oleh kutub-kutub sampai kutub-kutub tersebut berganti lokasi, sehingga arusnya akan terbalik dan menyebabkan kutub utara dan kutub selatan berubah.
3. Komutator Motor DC, sebuah komponen untuk membalikan arah arus listrik dalam kumparan juga membantu transmisi antara kumparan motor dan sumber daya.

2.5.1 *Gearbox*

Gearbox adalah suatu alat atau komponen transmisi utama motor yang berfungsi sebagai sistem pemindah tenaga, yaitu untuk menyalurkan tenaga atau daya mesin ke salah satu bagian mesin lainnya. Sehingga mesin tersebut dapat bergerak menghasilkan gerakan berupa putaran maupun pergeseran. *Gearbox* juga merupakan alat khusus untuk menyesuaikan daya atau torsi (momen/daya) dari suatu motor yang berputar, sehingga motor yang berputar tersebut memiliki tenaga yang lebih besar maupun lebih kecil. Transmisi pada *gearbox* juga memiliki fungsi lain yaitu untuk mengatur kecepatan gerak, menyesuaikan torsi,

serta berbalik putaran, sehingga objek dapat bergerak maju dan mundur.

Fungsi utama dari *gearbox* (transmisi manual) adalah sebagai berikut:

1. Merubah momen puntir yang akan diteruskan ke spindel mesin.
2. Menyediakan rasio gigi yang sesuai dengan beban mesin.
3. Menghasilkan putaran mesin tanpa selip.

Prinsip kerja *gearbox* yaitu putaran dari motor diteruskan ke input *shaft* (poros input) melalui hubungan antara *clutch*/kopling, kemudian putaran diteruskan ke *main shaft* (poros utama), torsi/momen yang ada di *main shaft* diteruskan ke *spindle* mesin, karena adanya perbedaan rasio dan bentuk dari gigi-gigi tersebut sehingga rpm atau putaran *spindel* yang di keluarkan berbeda, tergantung dari rpm yang di inginkan.

Komponen utama *gearbox* antara lain:

1. *Input shaft* (poros input), merupakan komponen yang menerima momen output dari kopling, dan juga berfungsi meneruskan putaran dari *clutch* kopling ke poros utama (*main shaft*), putaran tersebut disalurkan menuju *gear-gear*. Poros input juga sebagai dudukan pada *bearing* dan sebagai saluran oli yang melumasi bagian dari poros input itu sendiri. *Input shaft* ditunjukkan pada gambar 2.8.



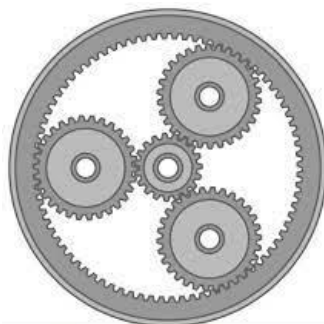
Gambar 2. 8 *Input Shaft*

1. *Main shaft* (poros utama), merupakan dudukan *gear*, *bearing*, dan komponen-komponen lainnya yang berfungsi sebagai poros penerus putaran dari *input shaft* untuk diteruskan ke *spindel*, dan juga sebagai saluran tempat jalannya oli. *Main shaft* ditunjukkan pada gambar 2.9.



Gambar 2. 9 *Main Shaft*

2. *Planetary gear* (gigi planetari), merupakan komponen yang berfungsi untuk mengubah arah putaran spindel, dan dapat merubah RPM pada suatu range tertentu sesuai dengan kebutuhan proses kerja motor. *Planetary gear* ditunjukkan pada gambar 2.10.



Gambar 2. 10 *Planetary Gear*

3. *Bearing*, merupakan komponen yang berfungsi untuk menjaga kerenggangan antar poros-poros, agar saat proses transmisi tidak

terjadi kejutan, sehingga transmisi dapat bekerja secara halus. *Bearing* ditunjukkan pada gambar 2.11.



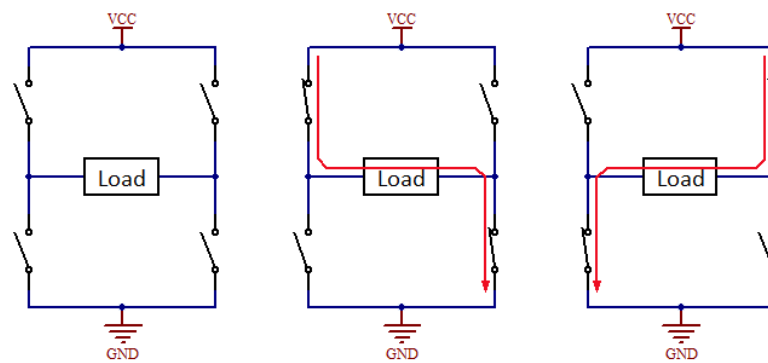
Gambar 2. 11 *Bearing*

2.6 *Driver Motor*

Driver motor merupakan rangkaian yang tersusun dari transistor yang berfungsi untuk mengatur gerakan motor DC. *Driver* motor yang sering digunakan yaitu L298N. *Driver* motor L298N adalah modul *driver* motor DC yang berfungsi mengontrol kecepatan dan arah putaran motor DC, arah putaran diatur oleh *H-Bridge* dan kecepatan diatur oleh pwm.

Didalam motor *driver* ini terdapat IC tipe *H-Bridge* yaitu IC L298 yang terdiri dari transistor-transistor logic (TTL) dengan gerbang NAND. IC ini menggunakan prinsip kerja *H-Bridge*. Tiap *H-Bridge* dikontrol menggunakan level tegangan TTL yang berasal dari output mikrokontroler. Tegangan yang dapat digunakan untuk mengendalikan robot bisa mencapai tegangan 46 VDC dan arus 2A untuk setiap kanalnya. L298 dapat mengontrol 2 buah motor DC, karena di dalam satu komponen L298N terdapat dua rangkaian *H-Bridge*.

Fungsi utama rangkaian *H-Bridge* yaitu mengubah arah aliran arus dengan mengubah polaritas tegangan inputnya sehingga dapat mengubah atau mengontrol arah putaran motor yang sedang bekerja. Sirkuit *H-Bridge* berisi empat elemen *switching*, berupa transistor (BJT atau MOSFET), dengan motor di tengah membentuk konfigurasi seperti hurup H.

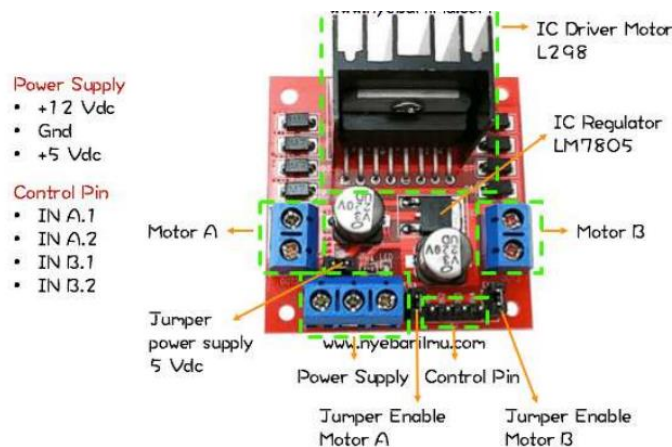


Gambar 2. 12 Rangkaian H-Bridge

Gambar 2.12 menunjukkan kondisi sebelah kiri semua sakelar terbuka maka tidak ada arus yang masuk ke terminal motor, sehingga pada kondisi ini motor dalam keadaan berhenti (tidak bekerja).

Pada kondisi tengah sakelar bagian atas sebelah kiri dan sakelar bagian atas sebelah kanan tertutup, maka terminal kiri motor mendapat tegangan positif dan terminal kanan motor mendapat tegangan negatif, sehingga motor berputar searah dengan arah jarum jam.

Pada kondisi sebelah kanan, sakelar bagian atas sebelah kanan dan bagian bawah sebelah kiri tertutup, maka terminal motor kanan mendapat tegangan positif dan terminal motor kiri mendapat tegangan negatif, sehingga motor dapat berputar berlawanan dengan arah jarum jam.



Gambar 2. 13 *Driver Motor*

Keterangan gambar 2.13 sebagai berikut:

- Enable A berfungsi untuk mengaktifkan bagian output motor A
- Enable B berfungsi untuk mengaktifkan bagian output motor B
- Jumper 5 Vdc sebagai mode pemilihan sumber tegangan 5Vdc, jika tidak dijumpers maka akan ke mode sumber tegangan 12 Vdc
- Control Pin Sebagai kendali perputaran dan kecepatan motor yang dihubungkan ke *Mikrocontroller*

Adapun untuk spesifikasi dari *driver* motor L298N dapat dijabarkan seperti berikut:

- Menggunakan IC L298N (*Double H-bridge Drive Chip*)
- Tegangan minimal untuk masukan power antara 5V-35V
- Tegangan operasional 5V
- Arus untuk masukan antara 0-36mA
- Arus maksimal untuk keluaran per Output A maupun B yaitu 2A
- Daya maksimal yaitu 25W
- Dimensi modul yaitu 43 x 43 x 26mm

- Berat: 26g

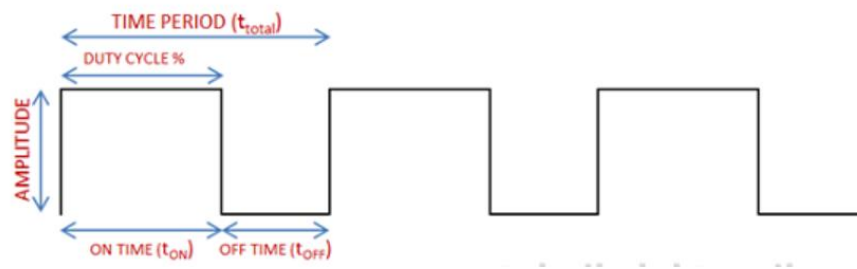
2.7 *Pulse Width Modulation*

Pulse width modulation (PWM) atau yang sering disebut modulasi lebar pulsa adalah suatu teknik untuk mengubah lebar pulsa dengan nilai frekuensi dan amplitude yang tetap, PWM digunakan untuk menghasilkan signal analog dari suatu perangkat digital (*mikrocontroller*). PWM juga adalah salah satu cara untuk mengatur kecepatan motor DC dengan menggunakan sumber tegangan yang tetap, yaitu dengan cara merubah secara cepat antara kondisi ON dan OFF pada frekuensi tertentu (Djahi et al., 2019).

Pulse Width Modulation (PWM) secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam suatu perioda, untuk mendapatkan tegangan rata-rata yang berbeda (Lubis & Yanie, 2022).

Pengaturan sinyal PWM untuk kondisi ON OFF disebut dengan siklus kerja PWM atau PWM Duty Cycle, PWM Duty Cycle merupakan pengkondisian waktu ON (kondisi tinggi 100% duty cycle) pada selang waktu tertentu dan akan terhenti OFF (mati atau 0% duty cycle) selama sisa periodenya. Rumus untuk menghitung duty cycle yaitu:

$$Duty\ Cycle = \frac{t_{on}}{t_{on} + t_{off}} = \frac{t_{on}}{t_{total}} \quad (2.12)$$



Gambar 2. 14 PWM dengan Duty Cycle

Gambar 2.14 merupakan bentuk gelombang dari *Duty Cycle* PWM. Frekuensi PWM menentukan seberapa cepat PWM menyelesaikan satu periode, satu periode adalah waktu total dari sinyal PWM. Untuk menghitung frekuensi menggunakan rumus berikut:

$$F = \frac{1}{P} \quad (2.13)$$

Rasio perbandingan kondisi ON dengan periode (duty cycle) dinyatakan dengan rumus:

$$duty\ cycle = \frac{t}{T} \times 100\% \quad (2.14)$$

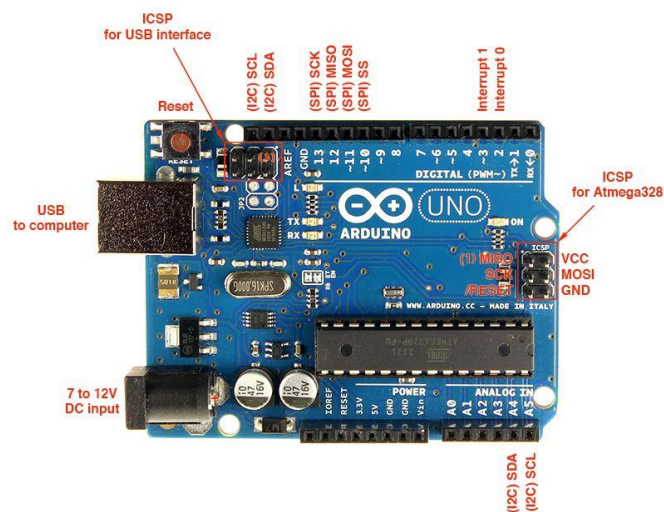
Besar persentase duty cycle berbanding lurus dengan kecepatan motor, dimana semakin besar duty cycle maka kecepatan motor semakin cepat karena arus rata-rata yang melalui motor semakin besar, begitupun sebaliknya jika duty cycle kecil maka arus rata-rata yang melalui motor juga kecil sehingga motor akan bergerak lambat.

Untuk menghitung tegangan output signal PWM digunakan rumus:

$$V_{out} = Duty\ Cycle \times V_{in} \quad (2.15)$$

2.8 Arduino

Arduino adalah sebuah *mikrocontroller single board* yang bersifat *open source* yang dapat digunakan untuk merancang sebuah perangkat elektronik pengendali peralatan pintar. Arduino memiliki prosesor Atmel AVR yang terdiri dari sebuah *mikrocontroller* ATmega, pin digital, pin analog, dan beberapa konektor. Arduino memiliki beragam jenis salah satunya Arduino UNO. Arduino UNO adalah *mikrocontroller* berbasis ATmega328P yang memiliki 14 pin digital yang dapat digunakan sebagai pin input dan output, serta dapat berkomunikasi dengan perangkat lain seperti PC, Arduino UNO lain, dan ATmega328 dengan komunikasi serial yang tersedia pada pin digital 0 (RX) dan pin 1 (TX). Bentuk fisik arduino ditunjukkan pada gambar 2.15.



Gambar 2. 15 Mikrocontroller Arduino UNO

2.8.1 Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah suatu aplikasi pemrograman untuk *board* arduino yang dilengkapi dengan bahasa pemrograman JAVA juga dilengkapi library C/C++. Dalam aplikasi ini programmer dapat membuat program, mengedit, serta mengupload program. Arduino IDE dapat di sambungkan pada berbagai mikrocontroller seperti ESP 32, ESP 826, dan seluruh mikrocontroller arduino. Tampilan aplikasi arduino IDE ditunjukkan pada gambar 2.16.



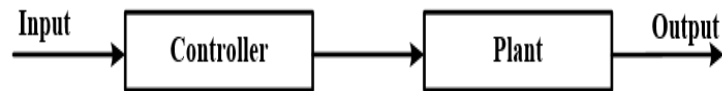
Gambar 2. 16 Tampilan *Software* Arduino IDE

2.9 Sistem Kontrol *Mobile Robot*

Sistem kontrol pada *Mobile Robot* terbagi menjadi dua kelompok yaitu sistem *open loop* dan juga sistem *close loop*, sistem *close loop* biasanya digunakan untuk mengendalikan posisi *Mobile Robot* dengan tipe penggerak diferensial yang mengacu pada suatu sasaran objek (menuju sasaran/titik akhir). Sedangkan sistem *open loop* biasanya digunakan

sebagai kendali pada kecepatan roda untuk mengatur pergerakan roda kanan maupun roda kiri.

2.9.1 Sistem Kontrol *Open Loop*



Gambar 2. 17 Sistem Kontrol *Open Loop*

Gambar 2.17 Sistem kontrol *open loop* merupakan sistem kontrol yang outputnya tidak berpengaruh terhadap aksi kontrol, sering disebut dengan *feed forward control* dimana output-nya tidak akan diperhitungkan ulang dan robot telah mencapai posisi target yang telah di atur. Pada sistem kontrol *open loop* untuk memperkecil kesalahan pada keluaran maka sistem ini memanfaatkan hasil kalibrasi dengan meninjau hubungan antara input dan output. kontrol ini sesuai untuk sistem operasi robot yang memiliki aktuator yang beroperasi berdasarkan umpan logika berbasis konfigurasi langkah sesuai urutan.

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

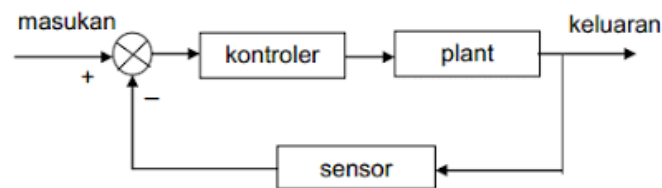
Tabel 2. 1 Kelebihan dan Kekurangan Sistem *Open Loop*

Sistem Kontrol <i>Open Loop</i>		
No.	Kelebihan	Kekurangan
1.	Konstruksi yang sederhana.	Diperlukan kalibrasi sistem secara teratur.

2.	Biaya konstruksi dan pemeliharaan yang relatif lebih murah.	Dapat beroperasi ketika hubungan input dan output sudah diketahui.
3.	Tidak memiliki masalah dalam hal stabilitas.	Jika terjadi gangguan sistem eksternal maupun internal dapat mengakibatkan nilai output tidak akurat.
4.	Cocok digunakan pada sistem yang outputnya sulit diukur.	Nilai output dapat berubah terhadap waktu.

2.9.2 Sistem Kontrol *Close Loop*

Sistem kontrol closed loop dimana kontroler bekerja sebagai penggerak plant (objek fisik yang digerakkan dalam sistem) dan mengontrol sifat plant (Saka Gilap Asa & Priyambodo, 2016).



Gambar 2. 18 Sistem Kontrol *Close Loop*

Gambar 2.18 Sistem kontrol *close loop* merupakan sistem kontrol yang outputnya berpengaruh pada aksi kontrol, sehingga hasil dari keluaran akan menjadi umpan balik pada input selanjutnya. Hasil keluaran pada sistem akan selalu memberikan *feedback* hingga hasil keluarannya

sesuai dengan yang diharapkan, sehingga keluaran akan berhenti memberikan *feedback* ketika hasil dari outputnya sudah sesuai. Beberapa fungsi sistem kontrol *close loop* terbagi menjadi beberapa bagian sebagai berikut:

1. Input adalah masukan yang diberikan pada sistem yaitu merupakan variabel nilai yang dikontrol.
2. Output adalah keluaran dari hasil sistem pengontrolan, output ini merupakan nilai yang akan dipertahankan dan juga menjadi umpan balik bagi siklus selanjutnya.
3. *Plant* adalah beban atau objek yang akan dikontrol.
4. *Controller* adalah alat kontrol yang memiliki fungsi sebagai pengendali beban pada sistem.
5. Elemen umpan balik adalah bagian yang menunjukkan nilai dari output ke detector agar dapat dibandingkan dengan nilai yang diinginkan.
6. *Error Detector* adalah bagian yang mendeteksi kesalahan dengan membandingkan selisih antara nilai input dan hasil umpan balik.
7. Gangguan adalah kemunculan sinyal tambahan yang tidak diperlukan tetapi dapat mengakibatkan keluaran berbeda dengan keluaran yang diharapkan, gangguan ini disebabkan karena adanya perubahan pada beban sistem.

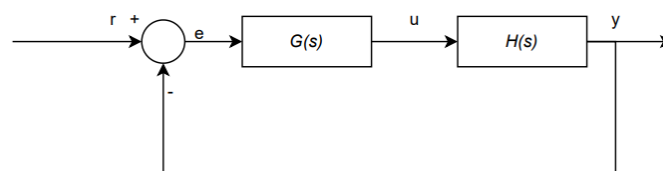
Kelebihan dan Kekurangan sistem kontrol *close loop* sebagai berikut:

Tabel 2. 2 Kelebihan dan Kekurangan Sistem *Close Loop*

Sistem Kontrol <i>Close Loop</i>		
No.	Kelebihan	Kekurangan
1.	Keakuratan dan ketelitian dapat terjaga.	Biaya yang relative lebih mahal dikarenakan banyaknya komponen yang diperlukan.
2.	Dapat mendeteksi perubahan pada beban dan output.	Biaya perawatan yang lebih mahal.
3.	Ketidaklinieran antar komponen tidak terlalu mengganggu sistem	Dapat mengakibatkan terjadinya osilasi.

2.9.3 Kontrol Proporsional

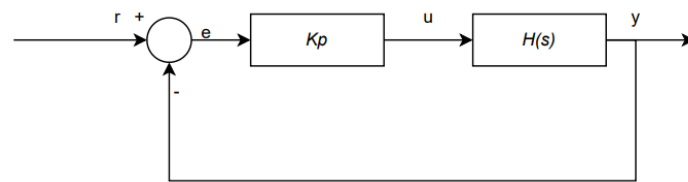
Cara kerja kontrol proportional ini yaitu dengan cara membandingkan variabel proses (variabel umpan balik) dengan set point sehingga dapat menghasilkan sinyal kesalahan yang digunakan untuk menyesuaikan nilai keluaran sistem. Proses tersebut terus berulang hingga nilai kesalahan mencapai titik nol dan nilai variabel proses sama dengan nilai keluaran yang diinginkan.



Gambar 2. 19 Diagram Kontrol PID

Dalam gambar 2.19 r adalah input, e adalah error, u adalah sinyal output *controller*, $G(s)$ adalah *controller*, $H(s)$ adalah dinamik robot (*plant*), dan y adalah output.

Kontrol P berlaku sebagai *gain* atau penguat yang memiliki keterbatasan karena sifat kontrol yang tidak dinamik, namun pada pengaplikasiannya dapat digunakan pada kontrol sederhana yang cukup mampu untuk mencapai konvergensi, khususnya cukup mampu memperbaiki respon transien *rise time* dan *settling time*, meskipun *error* keadaan tenangnya relatif besar.



Gambar 2. 20 Diagram Kontrol P

Kontrol P dengan konstanta K_p

$$u = K_p \cdot e \quad (2.16)$$

Pengontrolan atau tuning konstanta pada umumnya yang digunakan adalah metode trial-error, untuk menentukan parameter yang tepat untuk mendapatkan konstanta proportional (Colony & Time, 2021)