

BAB II

DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian ini dilakukan dengan mengadaptasi dari beberapa penelitian sebelumnya yang berkaitan dengan sistem kendali *hybrid pid-fuzzy logic* pada robot beroda *line follower*. Samet Oguten dan Bilal Kabas Pernah melakukan penelitian terhadap robot *line follower* pada tahun 2021 dengan menggunakan pengendali PID yang berjudul “**PID Controller Optimization for Low-cost Line follower Robots**”. Dalam penelitian tersebut robot *line follower* dirancang dengan menggunakan *microcontroller* STM32F103C8, untuk pembacaan garis menggunakan sensor QTR-8, dan menggunakan motor driver L298N sebagai pengendali kecepatan motor. Dalam perancangannya tersebut menggunakan pengendali PID dengan tambahan berupa pengendali open loop dikarenakan dalam hasil percobaan didapat hasil robot *line follower* yang hanya menggunakan pengendali PID tidak dapat menyelesaikan tikungan dengan sudut 45 derajat sehingga menyebabkan robot keluar jalur [2]. Dalam penelitian tersebut dijelaskan mengenai pengendali robot menggunakan PID akan tetapi tidak dijelaskan mengenai tuning parameter PID yang digunakan dalam perancangan sistem tersebut.

Penelitian yang berjudul “**Implementasi Fuzzy Logic Pada Robot Line follower**” dilakukan oleh Agusma Wijayansyah dan kawan-kawan pada tahun 2018. Penelitian ini merancang pengendali *fuzzy logic* dengan metode mamdani untuk robot *line follower*, akan tetapi dalam penelitiannya hanya dilakukan simulasi menggunakan matlab. Terdapat tiga fuzzy set dalam pengendali fuzzy tersebut dengan masing masing lima membership function. Tiga fuzzy set tersebut masing-masing terdiri dari dua fuzzy set *input* Sk1, Sk-1 dan satu fuzzy set *output*, dengan membership untuk fuzzy set *input* diantaranya Serong Kiri 2 (Ki2), Serong Kiri 1 (Ki1), Center (Cnt), Serong Kanan 1 (Kn1) dan Serong Kanan2 (Kn2), dengan semesta pembicaraan adalah 0 s/d 255. Sedangkan untuk *output* terdiri dari 5 membership yaitu Belok Kiri 2 (BK_{i2}), Belok Kiri 1 (BK_{i1}), Lurus, Belok Kanan 1 (BK_{n1}) dan Belok Kanan 2 (BK_{n2}), dengan semesta pembicaraan -127 s/d +127.

Hasil dari simulasi tersebut menunjukkan sistem dapat berjalan dengan baik sesuai dengan aturan yang sudah dibuat [4]. Kekurangan dalam jurnal tersebut diantaranya tidak ada implementasi secara real sehingga hasil pengujian tidak dapat mewakili kondisi real hanya dapat menjadi acuan mengenai perancangan sistem pengendali fuzzy dalam robot *line follower*.

Penelitian selanjutnya dilakukan oleh Muhammad Mukhlis Syeichu dan kawan-kawan dengan penelitian yang berjudul “**Desain Sistem Pengaturan Kecepatan Motor DC pada Prototipe Elevator Menggunakan Hybrid Fuzzy-PID Controller**” pada tahun 2021. Pada penelitian tersebut dibuat model pengendali *hybrid* PID-Fuzzy untuk dapat mengatur kecepatan motor DC pada prototype elevator. Pengendali PID dalam penelitian tersebut menggunakan metode tuning Ziegler Nichols, sedangkan pengendali fuzzy menggunakan metode mamdani. Sistem dengan pengendali Hybrid Fuzzy-PID tersebut dapat berjalan dengan baik nilai *rise time* = 0.35, *settling time* = 0.67, *peak time* = 2.16, dan *ess* = 0.02 % [5]. Dalam perancangannya sistem pengendali Fuzzy dan PID dilakukan secara hybrid dengan menjumlahkan *output* dari kedua pengendali tersebut, kemudian hasil dari penjumlahan tersebut menjadi *input* yang akan menggerakkan aktuator.

Penelitian selanjutnya dilakukan oleh H. Samsul Bachri M. dalam jurnal yang berjudul “**Sistem Kendali Hybrid PID-Logika Fuzzy Pada Pengaturan Kecepatan Motor DC**”. Dalam penelitian tersebut dibuat sebuah model pengendali yang menggabungkan antara pengendali PID dan Fuzzy untuk mengatur kecepatan sebuah motor dc yang diberi gangguan berupa beban yang akan mengganggu kinerja kecepatan motor dc tersebut. Hasil yang didapat dalam penelitian tersebut sistem hybrid PID-Logika Fuzzy dapat menghasilkan sebuah respon yang lebih baik dan dapat meningkatkan *recovery time* [6]. Sama halnya dengan penelitian yang dilakukan oleh Muhammad Mukhlis Syeichu dan kawan-kawan dengan penelitian yang berjudul “**Desain Sistem Pengaturan Kecepatan Motor DC pada Prototipe Elevator Menggunakan Hybrid Fuzzy-PID Controller**” sistem pengendali hybrid PID-Fuzzy dilakukan dengan cara menambahkan kedua *output* dari masing masing pengendali sebagai *input* dari sebuah aktuator.

Berdasarkan penelitian-penelitian tersebut penulis akan melakukan penelitian dengan merancang sebuah sistem kendali hybrid PID-Fuzzy Logic pada robot beroda *line follower*. Perancangan robot *line follower* mengadaptasi dari penelitian yang dilakukan Samet Oguten dan Bilal Kabas yang berjudul “**PID Controller Optimization for Low-cost Line follower Robots**”, sedangkan penerapan pengendali fuzzy pada *line follower* mengadaptasi dari penelitian yang berjudul “**Implementasi Fuzzy Logic Pada Robot Line follower**” dilakukan oleh Agusma Wijayansyah dan kawan-kawan, dan perancangan sistem pengendali hybrid PID-Fuzzy pada penelitian ini mengadaptasi dari penelitian yang dilakukan oleh Muhammad Mukhlis Syeichu dan kawan-kawan dengan penelitian yang berjudul “**Desain Sistem Pengaturan Kecepatan Motor DC pada Prototipe Elevator Menggunakan Hybrid Fuzzy-PID Controller**” dan penelitian yang dilakukan oleh H. Samsul Bachri M yang berjudul “**Sistem Kendali Hybrid PID-Logika Fuzzy Pada Pengaturan Kecepatan Motor DC**”.

Tabel 2.1 Kajian Pustaka

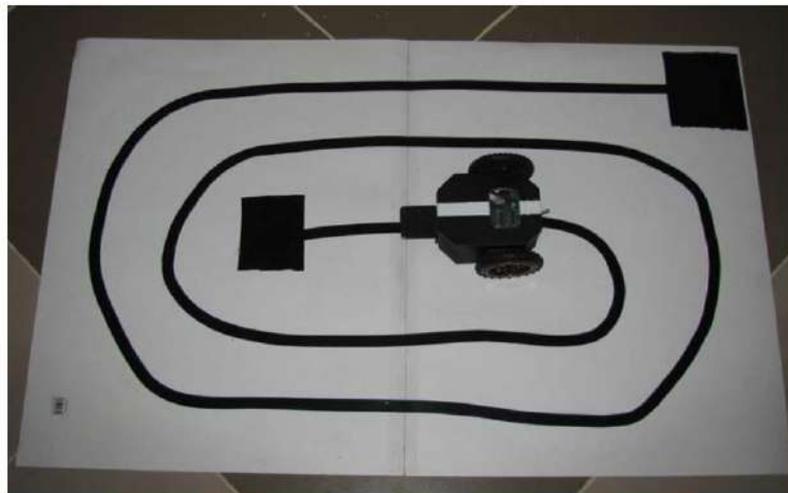
Peneliti	Judul	Perbedaan	Hasil Penelitian
Samet Oguten dan Bilal Kabas.	<i>PID Controller Optimization for Low-cost Line follower Robots</i> (2021).	Sistem pengendali hybrid PID-Fuzzy.	Robot tidak dapat melewati tikungan dengan besar sudut 45 derajat tanpa pengendali tambahan.
Agusma Wijayansyah, dkk.	Implementasi Fuzzy Logic Pada Robot <i>Line follower</i> (2018).	Sistem pengendali hybrid PID-Fuzzy, Fuzzy tsukamoto, Robot <i>line follower</i> .	Pengendali Fuzzy dapat berjalan dalam sistem robot <i>line follower</i> , akan tetapi penelitian ini dilakukan secara simulasi.
Muhammad Mukhlis Syeichu, dkk.	Desain Sistem Pengaturan Kecepatan	Robot <i>line follower</i> , Fuzzy tsukamoto.	Kinerja pengendali hybrid PID-Fuzzy logic lebih baik

	Motor DC pada Prototipe Elevator Menggunakan <i>Hybrid Fuzzy-PID Controller</i> (2021).		dibandingkan dengan pengendali PID saja.
H. Samsul Bachri M.	Sistem Kendali <i>Hybrid PID-Logika Fuzzy</i> Pada Pengaturan Kecepatan Motor DC (2004).	Robot <i>line follower</i> , Fuzzy tsukamoto.	Kinerja pengendali hybrid PID-Fuzzy logic lebih baik dibandingkan dengan pengendali PID saja.

2.2 DASAR TEORI

2.2.1 Robot *Line follower*

Robot *line follower* merupakan robot yang dirancang untuk mengikuti jalur atau garis tertentu pada permukaan yang memiliki kontras warna, seperti garis hitam pada latar belakang putih. Robot ini menggunakan sensor untuk mendeteksi garis dan mengambil keputusan berdasarkan informasi yang diterima untuk tetap berada di atas jalur tersebut [7].



Gambar 2.1 Robot *Line follower* [7].

Prinsip kerja robot *line follower* didasarkan pada penggunaan sensor inframerah (IR) atau sensor reflektif lainnya. Biasanya, robot dilengkapi dengan beberapa sensor IR yang diposisikan secara sejajar di bagian bawah robot. Ketika robot bergerak, sensor akan terus memindai permukaan di bawahnya. Jika sensor mendeteksi garis putih, berarti robot telah keluar dari jalurnya, dan robot harus mengoreksi arahnya untuk kembali ke jalur tersebut [8].

Untuk melakukan koreksi arah, robot *line follower* biasanya menggunakan logika sederhana. Jika sensor sebelah kiri mendeteksi garis hitam, maka robot harus berbelok ke kiri. Jika sensor tengah mendeteksi garis hitam, berarti robot sudah berada di jalur yang benar dan bisa melanjutkan perjalanan lurus. Jika sensor kanan mendeteksi garis hitam, robot harus berbelok ke kanan.

Proses koreksi arah ini berulang secara terus-menerus sehingga robot bisa mengikuti jalur dengan presisi. Dalam penelitian ini robot *line follower* tidak menggunakan logika sederhana untuk mencapai kestabilan sistem dan analisis tanggapan waktu yang baik robot *line follower* tersebut akan dirancang dengan menggunakan sistem pengendali [9].

2.2.2 Pengendali PID

Pengendali PID berfungsi untuk menentukan presisi dalam suatu sistem instrumentasi dengan karakteristik adanya *feedback* (umpan balik) yang terdapat pada sistem tersebut. PID (Proporsional, integral, Derivatif) merupakan pengendali gabungan yang terdiri dari tiga sistem, yaitu pengendali Proporsional, pengendali Integral, dan pengendali Derivatif [10]. Masing-masing dari pengendali tersebut memiliki konstanta yang harus diset dengan melakukan tuning terlebih dahulu. Terdapat kelebihan dan kekurangan pada masing-masing konstanta pengendali PID, hal tersebut terlihat dalam tabel berikut:

Tabel 2.2 Karakteristik Pengendali Proporsional, Integral, dan Derivatif [11]

Parameter	<i>Rise Time</i>	<i>Overshot</i>	<i>Settling Time</i>	<i>Error Steady-State</i>
K_p	Menurunkan	Meningkatkan	Perubahan Kecil	Menurunkan / Mengurangi
K_i	Menurunkan	Meningkatkan	Meningkatkan	Mengeliminasi

K_d	Perubahan kecil	Menurunkan	Meningkatkan	Perubahan Kecil
-------	-----------------	------------	--------------	-----------------

1. Pengendali Proporsional

Pengendali proporsional merupakan pengendali yang sederhana, hal ini dikarenakan sinyal kontrol umpan balik $u(t)$ sebanding dengan *error* umpan balik $e(t)$ [12].

$$u(t) = K_p e(t) \quad (2.1)$$

Keterangan: $u(t)$ = Sinyal kontrol umpan balik.

K_p = Konstanta proporsional / Penguatan proporsional.

$e(t)$ = *Error* umpan balik berdasarkan waktu (t).

Pengendali proporsional dapat mengurangi *Rise Time* dan *error steady state*, akan tetapi menambah *overshoot*. Jika konstanta pada pengendali proporsional diperbesar maka *overshoot* yang terjadi akan semakin besar, tetapi *Rise Timenya* menjadi kecil. Begitupun sebaliknya jika konstanta pada pengendali proporsional diperkecil [13].

2. Pengendali Proporsional Integral

Pengendali Proporsional Integral (PI) biasanya digunakan untuk pengontrolan proses yang memiliki dinamika relatif cepat [12]. Pengendali proporsional integral dapat diperoleh dengan cara mengatur nilai gain (waktu) derivatif sama dengan nol, atau secara matematis dapat ditulis:

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt \right] \quad (2.2)$$

Keterangan: $u(t)$ = Sinyal kontrol umpan balik.

K_p = Konstanta proporsional.

T_i = Konstanta waktu integral.

$e(t)$ = *Error* berdasarkan waktu (t).

dt = Nilai perubahan waktu (detik).

Seperti halnya dengan nilai gain proporsional, pemilihan parameter waktu (atau gain) integral pada pengendali PI secara praktis akan sangat mempengaruhi dinamika keluaran variabel *output* proses yang sedang dikendalikan. Jika kedua parameter pengendali (KP dan TI) dipilih secara tepat, maka respon yang didapat akan optimal (sesuai harapan). Pembesaran atau pengecilan nilai KP dan TI dari

nilai optimalnya dapat menyebabkan respon menjadi terlalu lambat atau terlalu beresilasi [14].

3. Pengendali Proporsional Derivatif

Secara umum dapat dikatakan bahwa fungsi dari pengendali derivatif adalah memperbaiki stabilitas dan mempercepat respon *output* proses karena dapat mengurangi *overshoot* dan *Settling time*, akan tetapi tidak memberikan dampak terhadap *error steady state* [12].

$$u(t) = K_p \left[e(t) + T_d \frac{de(t)}{dt} \right] \quad (2.3)$$

Keterangan: $u(t)$ = Sinyal kontrol umpan balik.

K_p = Konstanta proporsional.

T_d = Konstanta waktu derivatif.

$e(t)$ = *Error* berdasarkan waktu (t).

$de(t)$ = Nilai perubahan *error* (t).

dt = Nilai perubahan waktu (detik).

4. Pengendali Proporsional Integral Derivatif

Pengendali Proporsional Integral dan Derivatif (PID) pada dasarnya bertujuan untuk menggabungkan kelebihan komponen-komponen dasar yang dimiliki masing-masing pengendali seperti pengendali proporsional berfungsi untuk mempercepat respon, pengendali integral berfungsi untuk menghilangkan *error steady state*, dan pengendali derivative berfungsi untuk memperbaiki sekaligus mempercepat respon transien [14].

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (2.4)$$

Keterangan: $u(t)$ = Sinyal kontrol umpan balik.

K_p = Konstanta proporsional.

T_i = Konstanta waktu integral.

T_d = Konstanta waktu derivatif.

$e(t)$ = *Error* berdasarkan waktu (t).

$de(t)$ = Nilai perubahan *error* (t).

dt = Nilai perubahan waktu (detik).

2.2.3 *Trial and Error Tuning Method*

Metode tuning pengendali PID terdapat beberapa jenis diantaranya metode *trial and error*. Metode tersebut akan diterapkan pada penelitian ini, metode tuning PID menggunakan pendekatan *trial and error* adalah salah satu pendekatan sederhana untuk menentukan nilai parameter kontrol PID (K_p , K_i , dan K_d) dengan mencoba nilai-nilai yang berbeda secara manual dan mengamati respons sistem hingga mencapai hasil yang diinginkan. Metode ini juga dikenal dengan sebutan "*heuristik tuning*" atau "*manual tuning*" [15].

Langkah-langkah dalam metode tuning PID menggunakan pendekatan *trial and error* adalah sebagai berikut [16]:

1. Inisialisasi Parameter

Langkah pertama menetapkan nilai awal untuk setiap parameter kontrol PID (K_p , K_i , dan K_d). Nilai-nilai awal ini dapat didasarkan pada pengetahuan awal tentang sistem atau dapat diambil dari tuning yang sebelumnya dilakukan.

2. Percobaan Pertama

Setelah nilai awal ditetapkan, sistem dikendalikan menggunakan nilai-nilai tersebut dan respons sistem diamati. Jika respons sistem masih tidak sesuai dengan yang diinginkan, maka nilai-nilai parameter kontrol dapat diubah dan percobaan selanjutnya dilakukan.

3. Pengamatan dan Perubahan

Proses percobaan dan pengamatan berulang-ulang dilakukan hingga mencapai respons yang diinginkan dari sistem. Selama proses ini, nilai-nilai parameter PID dapat diubah secara manual dan metode ini mengandalkan pengalaman dan intuisi operator untuk menentukan perubahan nilai parameter.

4. Evaluasi Kinerja

Selama percobaan dan pengamatan, kinerja sistem dievaluasi berdasarkan kriteria yang diinginkan, seperti respon yang cepat, *overshoot* yang minimal, *Settling time* yang baik, dan *error steady state* yang rendah.

5. Kesesuaian dan Koreksi

Metode *trial and error* melibatkan proses iteratif untuk mencari kombinasi nilai parameter yang paling sesuai dengan kebutuhan kinerja sistem. Ketika nilai-

parameter yang diubah menghasilkan respons yang sesuai, nilai-nilai tersebut dapat dianggap sebagai parameter tuning yang cocok untuk sistem tersebut.

2.2.4 Pengendali Fuzzy Tsukamoto

Logika fuzzy diperkenalkan pertama kali pada tahun 1965 oleh Prof Lutfi A. Zadeh seorang peneliti di Universitas California di Barkley dalam bidang ilmu komputer. Professor Zadeh beranggapan logika benar salah tidak dapat mewakili setiap pemikiran manusia, kemudian logika fuzzy dikembangkan sehingga dapat mempresentasikan setiap keadaan atau mewakili pemikiran manusia. Perbedaan antara logika tegas dan logika fuzzy terletak pada keanggotaan elemen dalam suatu himpunan. Jika dalam logika tegas suatu elemen mempunyai dua pilihan yaitu terdapat dalam himpunan atau bernilai 1 yang berarti benar dan tidak pada himpunan atau bernilai 0 yang berarti salah. Sedangkan dalam logika fuzzy, keanggotaan elemen berada di interval $[0,1]$ [17].

Terdapat beberapa metode dari logika fuzzy diantaranya metode fuzzy Tsukamoto. Pada metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-THEN harus direpresentasikan dalam suatu himpunan fuzzy dengan fungsi keanggotaan monoton. Nilai hasil pada konsekuen setiap aturan fuzzy berupa nilai crisp yang diperoleh berdasarkan fire strength pada antiseden-nya. Keluaran sistem dihasilkan dari konsep rata-rata terbobot dari keluaran setiap aturan fuzzy [18].

Misal terdapat 2 variabel masukan, yaitu x dan y serta sebuah variabel keluaran yaitu z. Variabel x terbagi atas 2 himpunan A1 dan A2, variabel y terbagi atas 2 himpunan B1 dan B2, dan variabel keluaran y terbagi atas 2 himpunan C1 dan C2. Jika terdapat 2 aturan fuzzy:

JIKA x adalah A1 dan y adalah B1 MAKA z adalah C1.

JIKA x adalah A2 dan y adalah B2 MAKA z adalah C2.

α -predikat untuk aturan pertama adalah w_1 dan α -predikat untuk aturan ke dua adalah w_2 . Dengan penalaran monoton didapat keluaran aturan pertama adalah z_1 dan z_2 sebagai keluaran untuk aturan kedua. Untuk mendapatkan keluaran akhir digunakan konsep rata-rata terbobot.

$$Z = \frac{\sum w_i * z_i}{\sum w_i} \quad (2.5)$$

2.2.5 Tanggapan Sistem (*System Response*)

Karakteristik respon sistem dapat dikategorikan menjadi dua jenis berdasarkan bentuk sinyal uji yang digunakan, yaitu karakteristik respon waktu (*Time Response*) dan karakteristik respon frekuensi (*frequency response*) [19].

Berdasarkan penjelasan yang diberikan, terdapat dua macam karakteristik respon sistem:

1. Karakteristik Respon Waktu (*Time Response*)

Karakteristik respon waktu menggambarkan bagaimana sistem merespon terhadap perubahan sinyal *input* atau gangguan dari waktu ke waktu. Spesifikasi respon transient adalah spesifikasi respon sistem dari saat terjadinya perubahan sinyal *input* atau gangguan hingga respon masuk dalam keadaan steady state. Beberapa tolak ukur yang digunakan untuk mengukur kualitas respon transient adalah sebagai berikut:

- a. *Time Constant* (T) merupakan waktu yang dibutuhkan oleh sistem untuk mencapai 63,2% dari respon steady state setelah perubahan *input*.
- b. *Rise Time* (TR) merupakan waktu yang dibutuhkan oleh sistem untuk bergerak dari 5% hingga 95% dari respon steady state setelah perubahan *input*.
- c. *Settling time* (Ts) merupakan waktu yang dibutuhkan oleh sistem untuk masuk ke dalam rentang $\pm 0,5\%$, $\pm 2\%$, atau $\pm 5\%$ dari respon steady state setelah perubahan *input*.
- d. *Delay Time* (TD) merupakan waktu yang dibutuhkan oleh sistem untuk mencapai 50% dari respon steady state setelah perubahan *input*.
- e. *Overshoot* merupakan perbandingan nilai maksimum respon yang melebihi nilai steady state.
- f. *Time Peak* (Tp): Waktu yang dibutuhkan oleh sistem untuk mencapai nilai puncak maksimum setelah perubahan *input*.

Selain itu, spesifikasi respon steady state juga diamati saat sistem berada dalam keadaan steady state (setelah mencapai keseimbangan). Salah satu tolak ukur yang digunakan untuk mengukur kualitas respon steady state adalah *error steady state*.

2. Karakteristik Respon Frekuensi (*Frequency Response*):

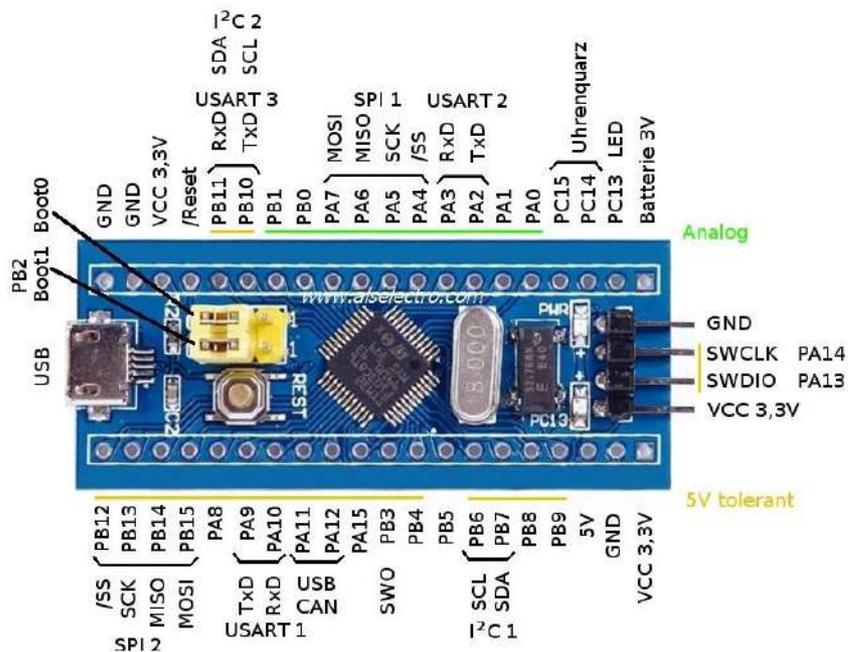
Karakteristik respon frekuensi menggambarkan performa sistem berdasarkan pengamatan terhadap magnitudo dan sudut fase penguatan (*output/input*) sistem

untuk masukan sinyal sinus dengan rentang frekuensi dari nol hingga tak terbatas ($0 \text{ s/d } \infty$).

Karakteristik respon waktu (*Time Response*) berfokus pada bagaimana sistem merespon terhadap perubahan sinyal *input* dari waktu ke waktu, sedangkan karakteristik respon frekuensi (frequency response) berfokus pada performa sistem terhadap sinyal sinusoidal pada berbagai frekuensi. Kedua karakteristik ini merupakan hal penting dalam analisis dan perancangan sistem kendali dan kontrol [20].

2.2.6 STM32F103C8T6

STM32F103C8T6 adalah mikrokontroler dari keluarga STM32 yang diproduksi oleh STMicroelectronics. Mikrokontroler ini termasuk dalam kelompok mikrokontroler dengan arsitektur ARM Cortex-M3 yang berkecepatan tinggi dan hemat daya. STM32F103C8T6 umumnya dikenal dengan nama "Blue Pill" karena bentuk fisiknya yang berwarna biru [21].



Gambar 2.2 Board STM32F103C8T6 Blue Pill [22]

Berikut penjelasan spesifikasi dari STM32F103C8T6 [22].

1. Arsitektur

STM32F103C8T6 merupakan mikrokontroler berbasis arsitektur 32-bit ARM Cortex-M3 yang merupakan arsitektur yang kuat dan efisien. Cortex-M3

mendukung instruksi 32-bit yang memungkinkan eksekusi cepat dan efisien dari kode.

2. Kecepatan

Mikrokontroler ini beroperasi pada kecepatan hingga 72 MHz. Ini menawarkan kinerja yang baik dan cocok untuk aplikasi yang membutuhkan kecepatan pemrosesan yang tinggi.

3. Memori

STM32F103C8T6 memiliki 64KB flash memory untuk menyimpan program (kode aplikasi) dan 20KB SRAM untuk menyimpan data saat aplikasi dijalankan. Chip ini juga memiliki 2KB EEPROM yang dapat digunakan untuk menyimpan data yang memerlukan retensi selama penghentian daya.

4. Periferal

STM32F103C8T6 dilengkapi dengan berbagai periferal yang berguna untuk berbagai aplikasi. Periferal ini termasuk GPIO (General Purpose *Input/Output*), USART (Universal Synchronous/Asynchronous Receiver/Transmitter), SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), ADC (Analog-to-Digital Converter), dan banyak lagi.

5. Sumber Daya

Mikrokontroler ini didukung oleh sistem tegangan 2,0 V hingga 3,6 V. Ini memungkinkan penggunaan sumber daya rendah atau baterai dalam aplikasi bergerak. STM32F103C8T6 memiliki beberapa mode hemat daya yang dapat membantu mengurangi konsumsi daya saat sistem sedang tidak aktif.

6. Antarmuka dan Komunikasi

STM32F103C8T6 memiliki berbagai pin GPIO yang dapat dikonfigurasi untuk berbagai tugas. Selain itu, juga menyediakan antarmuka komunikasi serial seperti USART, SPI, dan I2C untuk menghubungkan dengan perangkat eksternal atau sensor.

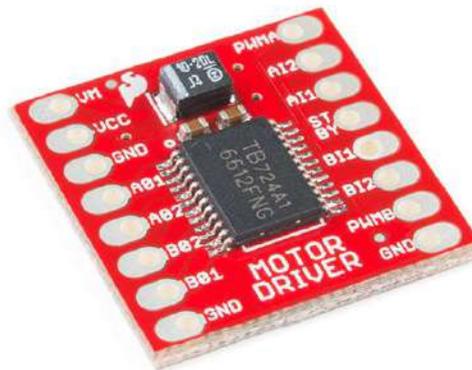
7. Pengembangan dan Pemrograman

Pengembangan perangkat lunak untuk STM32F103C8T6 dapat dilakukan menggunakan berbagai lingkungan pengembangan seperti STM32CubeIDE atau Arduino IDE dengan bantuan papan "Blue Pill". Mikrokontroler ini mendukung

berbagai mode pemrograman termasuk menggunakan ST-Link, USB bootloader, atau menggunakan programmer dan debugger eksternal.

2.2.7 TB6612FNG

TB6612FNG adalah chip driver motor dual H-bridge yang dikembangkan oleh Toshiba. Sebagai sebuah H-bridge, chip ini memiliki dua jalur pengendalian yang independen, memungkinkan kendali dua arah dari sebuah motor DC atau motor stepper. Setiap H-bridge memiliki empat terminal: IN1, IN2, PWM, dan STBY. *Input* tegangan kerja TB6612FNG adalah dari 2,7V hingga 5,5V, motor driver tersebut dapat mendrive sebuah motor hingga tegangan 15V [23].



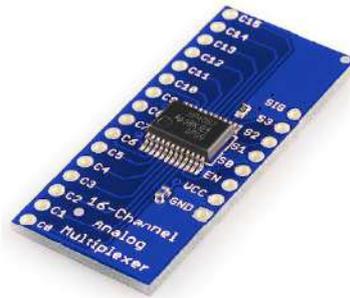
Gambar 2.3 Motor Driver TB6612FNG [24]

Pengaturan kecepatan motor DC dapat dilakukan menggunakan teknik *Pulse Width Modulation* (PWM) pada pin PWM. Dengan mengatur siklus kerja PWM, kecepatan motor dapat diatur sesuai kebutuhan. Selain itu, kontrol arah putaran motor dapat dicapai dengan mengatur kombinasi sinyal pada pin IN1 dan IN2. Proteksi overcurrent merupakan fitur yang penting pada TB6612FNG. Jika arus yang mengalir melampaui batas yang ditetapkan, chip secara otomatis akan memutuskan sumber daya ke motor, melindungi motor dan sirkuit dari kerusakan.

TB6612FNG juga dirancang dengan baik secara termal, memungkinkan operasi yang aman dan efisien bahkan pada beban tinggi. Dengan ukuran fisik yang relatif kecil, TB6612FNG mudah diintegrasikan ke dalam berbagai perangkat dan aplikasi. Sebagai hasilnya, TB6612FNG banyak digunakan dalam berbagai proyek robotika, kendali motor, kendaraan otomatis, drone, mainan bergerak, dan aplikasi kreatif lainnya yang membutuhkan kontrol motor DC atau stepper.

2.2.8 CD74HC4067

CD74HC4067 merupakan perangkat yang bersifat sakelar digital untuk sinyal analog yang menggunakan CMOS untuk mencapai kecepatan operasi seperti LSTTL, dengan konsumsi daya yang rendah dari CMOS standar sirkuit terintegrasi.



Gambar 2.4 Board Multiplexer CD74HC4067 [25]

Multiplexer/demultiplexer analog ini mengontrol tegangan analog yang dapat bervariasi di seluruh rentang suplai tegangan. Perangkat ini berfungsi sebagai sakelar dua arah sehingga memungkinkan digunakan sebagai *input* atau sebagai *output* [25]. Dalam penelitian ini multiplexer digunakan sebagai saklar yang menghubungkan *input* sensor *line follower* dengan mikrokontroler, sehingga penggunaan pin mikrokontroler tidak terlalu banyak.