

BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian pada [7] membahas perancang metode *fuzzy adaptive recursive terminal sliding mode control* pada robot *omnidirectional* berbasis roda *mecanum*. Hasil penelitian menunjukkan bahwa penerapan logika *fuzzy* yang digabungkan dengan *sliding mode control* mampu meningkatkan ketepatan pelacakan lintasan dan kestabilan sistem dibandingkan dengan pengendali seperti PID maupun *sliding mode control* standar. Namun, penelitian tersebut berfokus pada peningkatan performa pelacakan lintasan dan belum menggunakan parameter kondisi aktual kendaraan, seperti berat muatan dari *load cell* dan sudut kemiringan dari sensor IMU, sebagai masukan langsung dalam sistem kendali *fuzzy* untuk menghasilkan RPM motor pada AGV [7].

Penelitian pada [8] merancang *adaptive fuzzy sliding mode control* untuk robot *mobile nonholonomic* yang mengalami gangguan eksternal dan saturasi aktuator. Hasil penelitian menunjukkan bahwa metode kendali tersebut mampu menjamin presisi pelacakan posisi dalam waktu yang telah ditetapkan serta mengurangi efek masalah saturasi aktuator dengan *fuzzy* adaptif. Namun, meskipun fokus pada peningkatan ketepatan pelacakan dan robust terhadap gangguan eksternal, penelitian ini tidak membahas integrasi data beban kendaraan atau orientasi dari sensor IMU ke dalam desain kendali *fuzzy* yang menghasilkan nilai RPM motor untuk menyesuaikan respons motor terhadap variasi beban aktual dan kemiringan kendaraan [8].

Penelitian pada [9] merancang sistem navigasi *real time* untuk robot berbasis roda *mecanum* di lingkungan dinamis. Hasil penelitian menunjukkan bahwa sistem tersebut mampu mempertahankan akurasi navigasi dan kestabilan pergerakan robot dalam menghadapi rintangan. Namun, penelitian ini hanya menitikberatkan pada aspek navigasi dan perencanaan gerak, serta belum membuktikan penggunaan

sensor beban dan data orientasi kendaraan sebagai variabel masukan dalam sistem kendali *fuzzy* untuk penyesuaian kecepatan motor berbasis [9].

Berdasarkan kajian literatur yang telah diuraikan, menunjukkan bahwa metode kendali cerdas mampu meningkatkan akurasi gerak robot dalam menghadapi dinamika sistem yang tidak linier serta gangguan eksternal. Selain itu, penelitian lain juga menunjukkan bahwa sistem navigasi dan manuver *real time* pada robot *mecanum* dapat beroperasi secara efektif di lingkungan dinamis dengan strategi perencanaan jalur yang sesuai. Namun demikian, penelitian-penelitian tersebut masih berfokus pada aspek pelacakan lintasan, navigasi, dan peningkatan kestabilan sistem kendali secara umum, serta belum memanfaatkan parameter kondisi aktual kendaraan seperti berat muatan yang diukur secara langsung menggunakan *load cell* dan sudut kemiringan kendaraan dari sensor IMU sebagai masukan secara bersamaan dalam sistem kendali *fuzzy* untuk menghasilkan nilai RPM motor yang adaptif. Oleh karena itu, penelitian ini dilakukan untuk merancang sistem kendali berbasis logika *fuzzy* tipe Mamdani yang menggunakan data berat muatan dan data orientasi kendaraan sebagai variabel masukan untuk menyesuaikan kecepatan motor secara *real time* pada AGV berbasis roda *mecanum*, sehingga diharapkan mampu meningkatkan stabilitas dan respons sistem terhadap variasi beban dan kondisi operasional.

2.2 Automated Guided Vehicle (AGV)

Automated Guided Vehicle (AGV) merupakan komponen penting dalam sistem intralogistik modern yang telah berkembang sejak pertengahan tahun 1950-an. Perkembangan AGV tidak terlepas dari kemajuan teknologi sensor, sistem kendali, serta mikroelektronika yang memungkinkan kendaraan bergerak secara otomatis tanpa operator manusia. Secara historis, AGV pertama kali dikembangkan di Amerika Serikat, kemudian diadopsi dan dikembangkan lebih lanjut di Eropa. Selama lebih dari enam dekade, perkembangan AGV mengalami beberapa tahap evolusi yang ditandai dengan perubahan standar teknologi, peningkatan kemampuan sistem, serta perluasan ruang lingkup aplikasi [10].

AGV tidak beroperasi sebagai unit yang berdiri sendiri, melainkan sebagai bagian dari suatu sistem yang disebut *Automated Guided Vehicle System* (AGVS). Sejak tahap awal pengembangannya, sistem AGV telah memiliki elemen dasar yang tetap dipertahankan hingga saat ini, yaitu *master control system*, kendaraan dengan sistem kontrol *onboard*, sistem keselamatan, serta sistem navigasi atau *track guidance system*. *Master control system* berfungsi mengoordinasikan pergerakan kendaraan dalam jaringan transportasi internal, mengatur urutan perjalanan, serta mencegah konflik antar kendaraan. Kendaraan dilengkapi dengan sistem kontrol yang menggantikan fungsi operator manusia, sehingga proses penentuan keputusan dilakukan melalui kombinasi sistem mekanik, kelistrikan, serta perangkat elektronik. Sistem keselamatan dirancang untuk melindungi pekerja dan peralatan di lingkungan industri, sedangkan sistem navigasi memastikan kendaraan bergerak mengikuti jalur atau referensi yang telah ditentukan [10].

Pada tahap awal pengembangannya, teknologi AGV masih sederhana dan sangat bergantung pada jalur fisik tetap. Navigasi dilakukan menggunakan konduktor berarus yang ditanam di lantai (*wire guidance*) atau garis optik yang dideteksi sensor visual. Prinsip induktif dimanfaatkan untuk mendeteksi posisi kendaraan terhadap konduktor berarus melalui perbedaan tegangan pada kumparan sensor di bawah kendaraan. Sistem keselamatan direalisasikan menggunakan sensor taktil seperti bumper atau *safety edge* yang menghentikan kendaraan saat terjadi kontak fisik. Sistem kontrol pada periode ini masih berbasis tabung vakum, relay, dan kemudian transistor, dengan fleksibilitas operasional yang terbatas karena kendaraan bergerak mengikuti jalur yang telah ditentukan secara tetap [10].

Perkembangan selanjutnya ditandai dengan masuknya mikroprosesor dan *Programmable Logic Controller* (PLC) yang meningkatkan kemampuan komputasi sistem. *Master control system* mulai menyerupai sistem *block control* pada perkeretaapian untuk mencegah tabrakan antar kendaraan. *Inductive guidance* menjadi metode navigasi yang umum digunakan, sementara sistem komunikasi data mulai memanfaatkan inframerah dan radio. Pada periode ini, industri otomotif

menjadi pengguna utama AGV dalam skala besar, terutama untuk mendukung proses produksi dan perakitan [10].

Pada tahap berikutnya, standar teknologi AGV menjadi lebih optimal. Navigasi bebas seperti laser *triangulation* dan *magnetic grid* mulai menggantikan ketergantungan pada jalur fisik tetap. Komputer personal (PC) digunakan sebagai *master control system*, sedangkan kendaraan dilengkapi PLC atau mikrokomputer *onboard*. Komunikasi data banyak memanfaatkan jaringan nirkabel seperti WiFi. Pada periode ini, AGV diterima secara luas sebagai solusi intralogistik yang andal dan digunakan di berbagai sektor industri, tidak lagi terbatas pada industri otomotif [10].

Dalam perkembangannya, fungsi AGV juga mengalami perluasan. Jika pada awalnya AGV berfungsi terutama sebagai alat transportasi linear dari satu stasiun ke stasiun lain, maka pada tahap selanjutnya AGV mampu melakukan berbagai pergerakan, termasuk manuver maju, mundur, serta pergerakan lateral. AGV dapat digunakan untuk memindahkan berbagai jenis unit *load* seperti *pallet*, kontainer, kotak, dan beban lainnya dalam area produksi dan pergudangan. Lingkungan operasional AGV pada dasarnya adalah lingkungan pabrik atau gudang konvensional yang disesuaikan agar memenuhi persyaratan sistem otomatis, melalui penandaan jalur, pengaturan area bebas hambatan, serta penerapan langkah-langkah proteksi keselamatan [10].

Pada tahap perkembangan terkini, AGV tidak lagi hanya berfungsi sebagai alat transportasi internal, tetapi juga mengambil peran tambahan dalam sistem produksi dan intralogistik. Kompleksitas sistem meningkat seiring dengan bertambahnya fungsi, jumlah kendaraan, serta interaksi dengan sistem lain dalam lingkungan industri. Meskipun demikian, elemen dasar sistem meliputi pengendalian terpusat, kontrol kendaraan, sistem navigasi, dan sistem keselamatan, tetap menjadi fondasi utama dalam perancangan dan implementasi AGVS [10].

2.3 Logika Fuzzy

Logika fuzzy (*fuzzy logic*) merupakan metode komputasi yang digunakan untuk menangani kondisi sistem yang tidak dapat dinyatakan secara pasti serta hubungan

sistem yang tidak bersifat linear. Berbeda dengan logika Boolean yang hanya mengenal dua nilai kebenaran, yaitu 0 dan 1, logika *fuzzy* menjadikan suatu variabel memiliki nilai keanggotaan pada rentang 0 sampai 1. Dengan demikian, suatu kondisi tidak dinyatakan secara mutlak benar atau salah, tetapi dapat memiliki tingkat keanggotaan tertentu sesuai dengan nilainya.

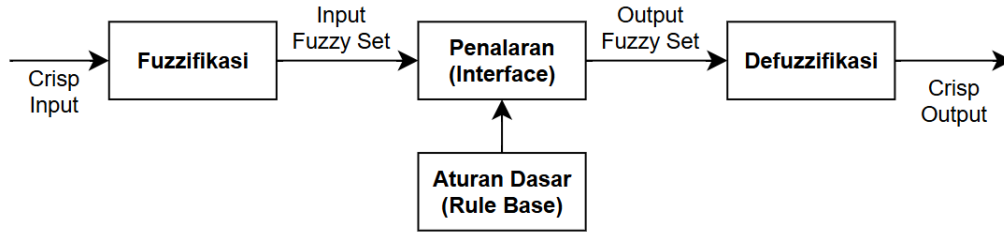
Konsep logika *fuzzy* diperkenalkan oleh Lotfi A. Zadeh pada tahun 1965 melalui teori *fuzzy set*, yang menjadi dasar pembentukan sistem berbasis aturan linguistik. Dalam sistem kendali, logika *fuzzy* digunakan untuk mengubah variabel masukan menjadi keluaran melalui aturan-aturan berbentuk pernyataan seperti “ringan”, “sedang”, dan “berat” yang dinyatakan dalam fungsi keanggotaan.

Metode ini tidak memerlukan model matematis yang rinci dari sistem yang dikendalikan, sehingga dapat digunakan pada sistem dengan kondisi yang berubah-ubah dan parameter yang sulit dinyatakan secara matematis secara tepat. Oleh karena itu, logika *fuzzy* sesuai diterapkan pada sistem kendali AGV yang membawa beban bervariasi dan beroperasi pada kondisi yang dinamis [11]

Secara umum ada beberapa unsur dalam logika *Fuzzy*, yaitu :

- a. Himpunan tegas (nilai *crisp*) adalah nilai keanggotaan.
- b. Himpunan *fuzzy* merupakan suatu himpunan yang berfungsi untuk merubah nilai *crisp* menjadi beberapa nilai dalam kelompok.
- c. Fungsi keanggotaan yang memiliki interval 0 sampai 1.
- d. Variabel linguistik adalah variabel yang dinyatakan dalam bentuk istilah atau kata-kata.
- e. Aturan (*rule*) *if-then fuzzy* merupakan aturan berbentuk pernyataan kondisional untuk mengatur *output* dari klasifikasi *input*.

Secara umum, sistem logika *Fuzzy* terdiri dari empat komponen utama, yaitu fuzzifikasi, basis aturan (*rule base*), inferensi (*inference*), dan defuzzifikasi, sebagaimana ditunjukkan pada Gambar 2.1.



Gambar 2. 1 Diagram Logika Fuzzy [11]

2.3.1 Fuzzifikasi

Fuzzifikasi merupakan proses mengubah nilai masukan tegas (*crisp input*) menjadi derajat keanggotaan dalam himpunan *fuzzy*. Pada tahap ini, setiap variabel *input* dipetakan ke dalam fungsi keanggotaan tertentu, seperti fungsi segitiga, trapesium, atau gaussian [12].

Fungsi Keanggotaan Segitiga (Triangular):

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x \leq b \\ \frac{c-x}{c-b} & b < x \leq c \\ 0 & x \geq c \end{cases} \quad (2.1)$$

Keterangan:

- a = batas bawah
- b = titik puncak
- c = batas atas

Fungsi Keanggotaan Trapesium (Trapezoidal):

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x \leq b \\ 1 & b < x \leq c \\ \frac{d-x}{d-c} & c < x \leq d \\ 0 & x \geq d \end{cases} \quad (2.2)$$

Fungsi ini digunakan untuk merepresentasikan kondisi seperti “beban ringan”, “beban sedang”, dan “beban berat”.

2.3.2 Aturan Dasar (Rule Base)

Aturan dasar merupakan kumpulan aturan berbentuk IF–THEN yang menghubungkan variabel *input* dengan variabel *output*. Aturan ini disusun

berdasarkan pengetahuan pakar atau karakteristik sistem. Contoh:
IF berat besar AND kemiringan tinggi THEN kecepatan rendah.

Contoh aturan dalam sistem AGV:

IF Beban Ringan AND Kemiringan Landai THEN RPM Cepat

IF Beban Ringan AND Kemiringan Agak Landai THEN RPM Sedang

Aturan ini disusun berdasarkan karakteristik sistem, yaitu semakin besar beban dan kemiringan, maka kecepatan motor perlu dikurangi untuk menjaga stabilitas.

2.3.3 Penalaran (*Inference*)

Tahap inferensi bertujuan mengevaluasi aturan berdasarkan derajat keanggotaan *input*. Operator logika seperti MIN (untuk AND) dan MAX (untuk agregasi aturan) digunakan untuk menentukan keluaran dalam bentuk himpunan *fuzzy*.

Operator AND (Minimum)

$$\alpha_i = \min(\mu A(x), \mu B(y)) \quad (2.3)$$

Operator OR (Maximum)

$$\alpha_i = \max(\mu A(x), \mu B(y)) \quad (2.4)$$

Implikasi pada metode Mamdani:

$$\mu_{output}(z) = \min(\alpha_i, \mu C(z)) \quad (2.5)$$

di mana:

- α_i = derajat aktivasi aturan ke-*i*
- $\mu_C(z)$ = fungsi keanggotaan *output*

Agregasi semua aturan:

$$\mu_{agregasi}(z) = \max(\mu_1(z), \mu_2(z), \dots, \mu_n(z)) \quad (2.6)$$

2.3.4 Defuzzifikasi

Defuzzifikasi merupakan proses mengubah keluaran *fuzzy* menjadi nilai numerik (*crisp output*). Nilai ini kemudian digunakan sebagai sinyal kendali, misalnya untuk mengatur kecepatan motor. Metode yang digunakan adalah metode centroid (*center of gravity*) [13].

Bentuk Kontinu:

$$z^* = \frac{\int_a^b z\mu(z)dz}{\int_a^b \mu(z)dz} \quad (2.7)$$

Keterangan

- $\int_a^b z\mu(z)dz = \text{Momen (M)}$
- $\int_a^b \mu(z)dz = \text{Luas (A)}$

Bentuk Diskrit (mikrokontroler):

$$z^* = \frac{\sum zi \cdot \mu(zi)}{\sum \mu(zi)} \quad (2.8)$$

Nilai z^* merupakan *output* berupa nilai RPM yang digunakan untuk mengatur kecepatan motor AGV.

2.3.5 Jenis Metode Logika Fuzzy

Dalam sistem inferensi *Fuzzy*, terdapat tiga metode utama yang umum digunakan, yaitu metode Tsukamoto, Sugeno, dan Mamdani. Ketiga metode ini memiliki perbedaan pada bentuk konsekuen aturan, proses inferensi, serta teknik defuzzifikasi yang digunakan.

1. Metode Tsukamoto

Metode Tsukamoto merupakan metode inferensi *Fuzzy* di mana setiap aturan IF–THEN memiliki konsekuen yang direpresentasikan dalam bentuk himpunan *Fuzzy* dengan fungsi keanggotaan monoton. Pada metode ini, setiap aturan menghasilkan *output* crisp berdasarkan nilai α -predikat (derajat kebenaran aturan). Nilai crisp dari setiap aturan kemudian digabungkan menggunakan metode rata-rata terbobot (*weighted average*) untuk memperoleh hasil akhir.

Derajat aktivasi aturan, seperti persamaan 2.3:

$$\alpha_i = \min(\mu A(x), \mu B(y))$$

2. Metode Sugeno

Metode Sugeno (Takagi-Sugeno) menggunakan aturan IF–THEN di mana bagian konsekuen tidak berupa himpunan *Fuzzy*, melainkan berupa konstanta atau persamaan linear dari variabel *input*. Dengan demikian, *output* setiap aturan langsung berupa nilai matematis.

Sebagai contoh:

IF eror kecil THEN $output = 5$

IF eror besar THEN $output = 0.8 \times error + 2$

Proses defuzzifikasi pada metode Sugeno dilakukan menggunakan rata-rata terbobot dari seluruh *output* aturan. Metode ini lebih efisien secara komputasi dan cocok untuk sistem *real time* seperti mikrokontroler.

3. Metode Mamdani

Metode Mamdani merupakan metode inferensi *Fuzzy* yang paling banyak digunakan dalam sistem kendali. Pada metode ini, konsekuen aturan berbentuk himpunan *Fuzzy*. Proses implikasi biasanya menggunakan operator MIN untuk operasi AND dan MAX untuk komposisi antar aturan. Metode Mamdani menghasilkan *output* dalam bentuk himpunan *Fuzzy* yang kemudian dikonversi menjadi nilai crisp melalui proses defuzzifikasi, umumnya menggunakan metode *centroid* (*Center of Gravity*).

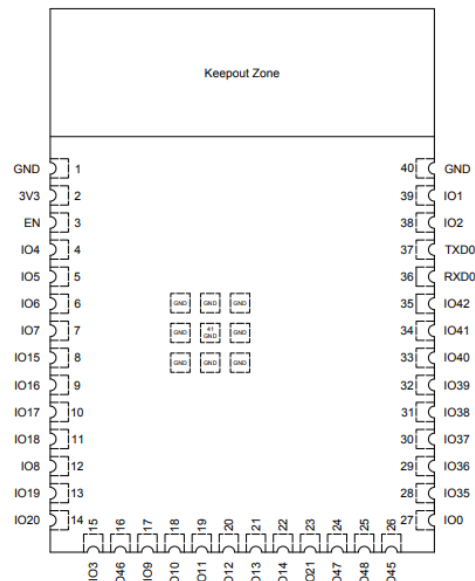
Secara matematis, metode *centroid* seperti pada persamaan 2.7 dinyatakan sebagai:

$$z^* = \frac{\int z\mu(z)dz}{\int \mu(z)dz}$$

Metode Mamdani memiliki keunggulan karena strukturnya intuitif dan menyerupai cara berpikir manusia, sehingga lebih mudah dipahami dan dirancang berdasarkan pengetahuan pakar. Oleh karena itu, metode ini sangat cocok digunakan dalam sistem pengendalian AGV, terutama untuk mengatur kecepatan, arah gerak, dan respons terhadap perubahan beban.

2.4 ESP32-S3-WROOM-1-N16R8

ESP32-S3-WROOM-1 merupakan modul mikrokontroler yang dikembangkan oleh Espressif Systems dan dibangun menggunakan *System-on-Chip* (SoC) ESP32-S3. Modul ini mengintegrasikan unit pemrosesan, memori, rangkaian radio frekuensi, kristal osilator, antena, serta antarmuka masukan dan keluaran dalam satu modul. ESP32-S3-WROOM-1 dirancang untuk aplikasi *Internet of Things* (IoT), sistem kendali, robotika, pengolahan sinyal digital, serta aplikasi berbasis



Gambar 2. 3 Pin ESP32-S3-WROOM-1 [14]

Berdasarkan Gambar 2.3, ESP32-S3-WROOM-1 memiliki 41 pin yang terdiri atas pin catu daya, pin kendali, serta pin *General Purpose Input Output* (GPIO). Modul ini menyediakan hingga 36 GPIO yang dapat digunakan sebagai masukan maupun keluaran digital. Selain itu, beberapa GPIO juga mendukung fungsi alternatif seperti komunikasi UART, SPI, I²C, I²S, PWM, ADC, USB, dan *capacitive touch sensor*. Spesifikasi ESP32-S3-WROOM-1-N16R8 dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Spesifikasi ESP32-S3-WROOM-1-N16R8

Parameter	Spesifikasi
Modul	ESP32-S3-WROOM-1-N16R8
Produsen	Espressif Systems
Prosesor	Xtensa® Dual-core 32-bit LX7
Frekuensi CPU	Hingga 240 MHz
ROM	384 KB
SRAM	512 KB
RTC SRAM	16 KB
Flash Memory	16 MB
PSRAM	8 MB Octal PSRAM
Wi-Fi	IEEE 802.11 b/g/n 2,4 GHz
Bluetooth	Bluetooth® 5 LE
GPIO	Hingga 36 GPIO
ADC	2 × 12 bit SAR ADC
<i>Capacitive Touch Sensor</i>	14 kanal
UART	3 antarmuka
I ² C	2 antarmuka

Parameter	Spesifikasi
SPI	4 antarmuka
I ² S	2 antarmuka
MCPWM	2 kanal
USB	USB 2.0 OTG
Tegangan	3,0 V – 3,6 V
Kristal Osilator	40 MHz

Berdasarkan Tabel 2.1, ESP32-S3-WROOM-1-N16R8 memiliki spesifikasi yang memadai untuk digunakan sebagai mikrokontroler utama pada sistem AGV ini. Kemampuan prosesor dual-core hingga 240 MHz, kapasitas memori yang cukup, serta ketersediaan berbagai antarmuka komunikasi dan GPIO mendukung integrasi seluruh komponen sistem, mulai dari sensor, driver motor, hingga fitur pemantauan berbasis Wi-Fi [14].

2.5 Sensor *Load cell*

Load cell merupakan sensor transduser yang berfungsi untuk mengubah besaran mekanik seperti gaya, tekanan, atau berat menjadi sinyal listrik yang proporsional terhadap beban yang diberikan. Prinsip kerja *load cell* didasarkan pada perubahan resistansi elemen *strain gauge* akibat deformasi mekanis pada struktur elastis sensor. Ketika beban diberikan, elemen elastis akan mengalami regangan (*strain*), sehingga nilai resistansi strain gauge berubah dan menghasilkan perubahan tegangan keluaran yang sebanding dengan besar beban [15].

Secara matematis, regangan didefinisikan sebagai:

$$\varepsilon = \frac{\Delta L}{L} \quad (2.9)$$

Keterangan:

- ε = regangan (strain)
- ΔL = perubahan panjang
- L = panjang awal

Perubahan resistansi strain gauge dinyatakan sebagai:

$$\frac{\Delta R}{R} = GF \cdot \varepsilon \quad (2.10)$$

Keterangan:

- ΔR = perubahan resistansi

- R = resistansi awal
- GF = *Gauge Factor* (umumnya 2-2.2 untuk strain gauge logam)

Bentuk fisik *load cell* yang digunakan dalam penelitian ini ditunjukkan pada Gambar 2.4. Sensor tersebut memiliki kapasitas pengukuran maksimum sebesar 10 kg dan bekerja berdasarkan perubahan resistansi elemen *strain gauge* ketika menerima beban. Perubahan resistansi menghasilkan sinyal tegangan diferensial dengan amplitudo yang sangat kecil. Oleh karena itu, keluaran *load cell* memerlukan modul penguat dan konverter analog ke digital HX711 agar dapat dibaca dan diproses oleh mikrokontroler.



Gambar 2. 4 Load cell 10 kg [16]

2.5.1 Konfigurasi Empat *Load cell* dan Integrasi dengan HX711

Sistem penimbangan pada AGV ini menggunakan empat buah *load cell full bridge* dengan kapasitas masing-masing 10 kg yang ditempatkan pada keempat titik tumpu *platform* beban, sehingga total kapasitas penimbangan mencapai 40 kg. Karena setiap *load cell full bridge* sudah merupakan rangkaian jembatan *Wheatstone* lengkap dengan empat kabel keluaran (E+, E-, A+, dan A-), keempat sensor tidak digabungkan menjadi satu rangkaian jembatan, melainkan dibaca secara independen oleh masing-masing modul HX711. Nilai keluaran keempat modul HX711 kemudian dijumlahkan secara perangkat lunak untuk memperoleh berat total. Pendekatan ini memungkinkan distribusi beban terukur di setiap titik tumpu serta memudahkan kompensasi apabila terdapat ketidakrataan beban.

Karena tegangan keluaran *load cell* sangat kecil (dalam orde milivolt per volt), diperlukan modul penguat sinyal HX711 yang berfungsi sebagai *instrumentation amplifier* sekaligus *analog-to-digital converter* (ADC) 24-bit sebelum sinyal

diproses oleh mikrokontroler. Setiap *load cell* dihubungkan ke satu modul HX711 tersendiri. Spesifikasi *load cell full bridge* yang digunakan ditunjukkan pada Tabel 2.2.

Tabel 2. 2 Spesifikasi *Load cell Full Bridge* 10 kg

Spesifikasi	Keterangan
Kapasitas Maksimal	10 kg per sensor
Jenis	Full-Bridge Strain Gauge
Sensitivitas <i>Output</i>	$1,0 \pm 0,15 \text{ mV/V}^*$
Tegangan Eksitasi	$\pm 4,3\text{--}5 \text{ V DC}$ (dari HX711)
Material	Aluminium alloy
Rentang Suhu Operasi	$-10 \text{ }^{\circ}\text{C} - 40 \text{ }^{\circ}\text{C}^*$
Jumlah Kabel	4 (Merah, Hitam, Hijau, Putih)

2.5.2 Konfigurasi Pin dan Pembacaan Diferensial

Load cell full bridge memiliki empat terminal utama yang berasal langsung dari satu unit sensor, yaitu *excitation* positif (E+), *excitation* negatif (E-), *signal* positif (A+/S+), dan *signal* negatif (A-/S-). Keempat terminal ini umumnya diwakili oleh empat warna kabel, yaitu merah, hitam, hijau, dan putih. Konfigurasi pin selengkapnya ditunjukkan pada Tabel 2.3.

Tabel 2. 3 Konfigurasi Pin *Full Wheatstone Bridge*

Pin	Nama	Fungsi
E+	Excitation +	Tegangan suplai positif dari HX711
E-	Excitation -	Ground dari HX711
A+	Signal +	Sinyal diferensial positif
A-	Signal -	Sinyal diferensial negatif

Terminal A+ dan A- dihubungkan ke modul HX711 untuk diperkuat dan dikonversi menjadi data digital sebelum diproses oleh mikrokontroler (ESP32). HX711 bekerja dengan prinsip pembacaan diferensial, yaitu mengukur selisih tegangan antara kedua terminal sinyal, seperti dinyatakan pada Persamaan 2.11.

$$\text{Digital Output} = \text{Gain} \times (A^+ - A^-) \quad (2.11)$$

2.5.3 Perhitungan *Error* dan Kalibrasi *Load cell*

a) Proses Kalibrasi *Load cell*

Kalibrasi dilakukan untuk memperoleh hubungan linier antara nilai digital keluaran HX711 dan berat aktual (kg). Secara umum, hubungan antara tegangan keluaran *load cell* dan beban bersifat linier:

$$V_{out} \propto W \quad (2.12)$$

Karena HX711 mengubah sinyal analog menjadi data digital 24-bit, maka persamaan konversinya menjadi:

$$W = \frac{(ADC_{reading} - ADC_{offset})}{K} \quad (2.13)$$

Keterangan:

- W = berat (kg)
- $ADC_{reading}$ = nilai pembacaan HX711
- ADC_{offset} = nilai pembacaan saat tanpa beban (zero/tare)
- K = faktor kalibrasi

Karena sistem ini menggunakan empat *load cell*, berat total diperoleh dari penjumlahan pembacaan keempat modul HX711, seperti dinyatakan pada Persamaan 2.14.

$$W_{total} = \frac{\sum_{i=1}^4 ADC_i - ADC_{offset,total}}{K_{total}} \quad (2.14)$$

Keterangan:

- W_{total} = berat total
- $\sum ADC_i$ = jumlah pembacaan keempat sensor
- $ADC_{offset,total}$ = offset gabungan tanpa beban
- K_{total} = faktor skala gabungan

Langkah-langkah kalibrasi dilakukan sebagai berikut. Pertama, kalibrasi titik nol (*tare*), yaitu membaca nilai ADC pada kondisi tanpa beban, sesuai Persamaan 2.15.

$$ADC_{offset} = ADC_{tanpa\,beban} \quad (2.15)$$

Kedua, kalibrasi dengan beban referensi yang telah diketahui nilainya (W_{ref}) untuk memperoleh faktor skala, sesuai Persamaan 2.16.

$$K = \frac{ADC_{ref} - ADC_{offset}}{W_{ref}} \quad (2.16)$$

Keterangan:

- K = faktor skala
- ADC_{ref} = pembacaan pada beban referensi
- W_{ref} = berat beban referensi

Untuk meningkatkan akurasi, kalibrasi satu titik pada Persamaan 2.16 dikembangkan menjadi kalibrasi lima titik menggunakan metode regresi linier kuadrat terkecil (*least squares*). Beberapa variasi beban standar diukur, kemudian dibentuk model linier antara nilai ADC dan berat aktual seperti pada Persamaan 2.17, dengan nilai kemiringan (a) dan konstanta *offset* (b) diperoleh dengan meminimalkan jumlah kuadrat selisih antara berat terukur dan berat referensi [5], [6].

$$W = a \cdot ADC + b \quad (2.17)$$

Keterangan:

- W = berat hasil kalibrasi (kg)
- a = kemiringan garis kalibrasi
- ADC = nilai pembacaan HX711
- b = konstanta koreksi *offset*

Ketiga, verifikasi dilakukan dengan menguji beberapa variasi beban untuk memastikan linieritas sistem pengukuran.

b) Perhitungan *Error* Pengukuran

Dalam sistem pengukuran berat, terdapat beberapa jenis *error*:

- *Error* Absolut

$$Error_{abs} = |W_{ukur} - W_{sebenarnya}| \quad (2.18)$$

- *Error* Relatif

$$Error_{rel} = \frac{|W_{ukur} - W_{sebenarnya}|}{W_{sebenarnya}} \times 100\% \quad (2.19)$$

- Akurasi

$$Akurasi = 100 \% - Error (\%) \quad (2.20)$$

- Rata-rata *Error Absolut*

$$\bar{E} = \frac{\sum E_i}{n} \quad (2.21)$$

- Rata-rata *Error Persen*

$$\overline{E \%} = \frac{\sum E_i \%}{n} \quad (2.22)$$

- Rata-rata Akurasi

$$\bar{A} = \frac{\sum A_i}{n} \quad (2.23)$$

- *Mean Absolute Error* (MAE)

Untuk beberapa pengujian:

$$MAE = \frac{1}{n} \sum_{i=1}^n |W_{ukur,i} - W_{referensi,i}| \quad (2.24)$$

- *Root Mean Square Error* (RMSE)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (W_{ukur,i} - W_{referensi,i})^2} \quad (2.25)$$

RMSE lebih sensitif terhadap *error* besar dibanding MAE.

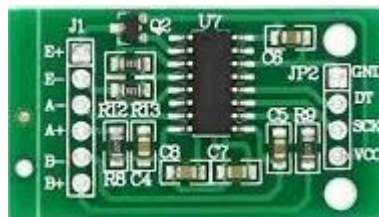
Keterangan:

- Wukur = berat hasil pengukuran
- Wsebenarnya / Wreferensi,i = berat acuan (timbangan standar)
- n = jumlah data pengujian

RMSE lebih sensitif terhadap *error* yang besar dibandingkan MAE, sehingga kombinasi keduanya memberikan gambaran yang lebih menyeluruh mengenai akurasi sistem.

2.6 Modul HX711

HX711 merupakan modul 24-bit *Analog to Digital Converter* (ADC) yang dirancang khusus untuk aplikasi timbangan digital dan sistem pengukuran berbasis sensor jembatan *Wheatstone*. Modul ini berfungsi untuk menguatkan sinyal keluaran *load cell* yang sangat kecil (dalam orde miliVolt) dan mengubahnya menjadi data digital yang dapat diproses oleh mikrokontroler. Pada sistem timbangan digital, *load cell* menghasilkan tegangan diferensial yang sangat kecil (sekitar ± 40 mV). Tegangan ini tidak dapat langsung dibaca oleh mikrokontroler, sehingga diperlukan penguat presisi seperti HX711 untuk meningkatkan resolusi pembacaan. Hal ini juga dijelaskan pada penelitian pengembangan timbangan digital berbasis *load cell* dan HX711 [15].



Gambar 2. 5 Modul HX711 [17]

Berdasarkan Gambar 2.5, HX711 bekerja dengan menerima sinyal diferensial berlevel rendah dari *load cell*, kemudian memperkuatnya menggunakan *Programmable Gain Amplifier* (PGA) internal dengan pilihan gain 32, 64, dan 128. Pada aplikasi timbangan, gain 128 umumnya digunakan karena sinyal keluaran *load cell* sangat kecil.

Sinyal yang telah diperkuat selanjutnya dikonversi menjadi data digital menggunakan Σ - Δ ADC beresolusi 24-bit, sehingga menghasilkan pembacaan dengan presisi tinggi dan *noise* rendah. Data digital tersebut dikirim ke mikrokontroler melalui komunikasi serial dua jalur, yaitu DT (*data output*) dan SCK (*clock*), sebagaimana diterapkan pada sistem timbangan berbasis IoT pada penelitian sebelumnya [18]. Spesifikasinya dapat dilihat pada Tabel 2.4 Spesifikasi HX711.

Tabel 2. 4 Spesifikasi HX711

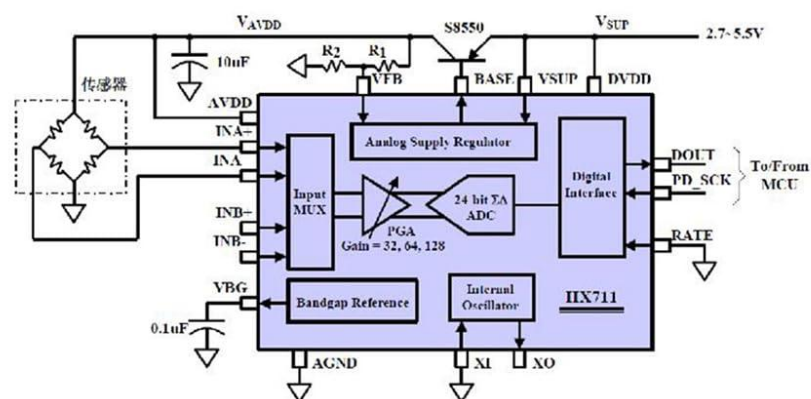
Spesifikasi	Keterangan
Tegangan <i>input</i> diferensial	$\pm 40\text{mV}$
Akurasi data	24 bit (<i>chip</i> konverter A / D 24)
Frekuensi	80 Hz
Tegangan Pengoperasian	5V DC
Operasi saat ini	<10 mA
Gain internal	32, 64, 128
Interface	2-wire serial (DT&SCK)
Ukuran	38mm * 21mm * 10mm

Berdasarkan spesifikasi tersebut, HX711 dapat diintegrasikan dengan *load cell* dan mikrokontroler untuk menghasilkan pembacaan berat yang presisi. Keterangan setiap pin pada modul HX711 dapat dilihat pada Tabel 2.5.

Tabel 2. 5 Konfigurasi Pin HX711

Pin	Keterangan
E+	Ch. E Positive <i>Input</i> (analog <i>input</i>)
E-	Ch. E Negative <i>Input</i> (analog <i>input</i>)
A-	Ch. A Negative <i>Input</i> (analog <i>input</i>)
A+	Ch. A Positive <i>Input</i> (analog <i>input</i>)
B-	Ch. B Negative <i>Input</i> (analog <i>input</i>)
B+	Ch. B Positive <i>Input</i> (analog <i>input</i>)
Gnd	Dihubungkan ke <i>ground</i>
DT	Digital <i>Output</i>
SCK	Digital <i>Input</i>
Vcc	Dihubungkan ke tegangan +5V

Diagram blok internal HX711 dapat dilihat pada Gambar 2.6.



Gambar 2. 6 Diagram Blok Internal HX711 [15]

Berdasarkan Gambar 2.6 ditunjukkan diagram blok internal HX711. Sinyal diferensial dari *load cell* masuk melalui kanal INA+ dan INA- (*Channel A*) atau INB+ dan INB- (*Channel B*), kemudian dipilih oleh *input multiplexer*. Sinyal

tersebut diperkuat menggunakan *Programmable Gain Amplifier* (PGA) dengan pilihan gain 32, 64, atau 128. Untuk aplikasi timbangan, gain 128 umumnya digunakan karena tegangan keluaran *load cell* sangat kecil (orde mV). Setelah diperkuat, sinyal analog dikonversi menjadi data digital menggunakan ADC 24-bit tipe sigma-delta (Σ - Δ) sehingga menghasilkan resolusi tinggi dan *noise* rendah. Data hasil konversi kemudian dikirim ke mikrokontroler melalui dua jalur komunikasi digital, yaitu DOUT sebagai keluaran data dan PD_SCK sebagai sinyal clock sekaligus kontrol daya.

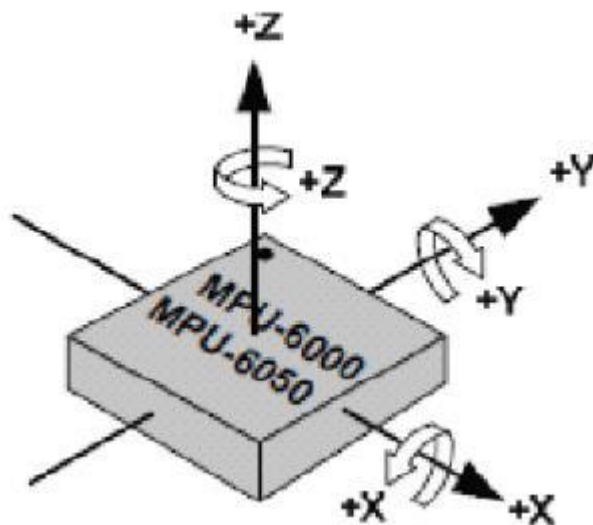
2.7 Sensor MPU6050

MPU6050 merupakan sensor *Inertial Measurement Unit* (IMU) yang menggabungkan 3-axis *accelerometer* dan 3-axis *gyroscope* dalam satu *chip* sehingga memiliki 6 derajat kebebasan (6-DOF). Sensor ini mampu mengukur percepatan linier dan kecepatan sudut pada sumbu X, Y, dan Z secara simultan [19]. Data keluaran sensor dikonversi menggunakan ADC 16-bit internal sehingga menghasilkan resolusi pengukuran yang tinggi.



Gambar 2. 7 Modul GY-521 [18]

Modul GY-521 yang digunakan pada penelitian ini dapat dilihat pada Gambar 2.7. Modul tersebut dilengkapi sensor MPU6050 yang mengintegrasikan *accelerometer* dan *gyroscope*. *Accelerometer* digunakan untuk mendeteksi percepatan dan sudut kemiringan berdasarkan pengaruh gravitasi, sedangkan *gyroscope* berfungsi mengukur kecepatan sudut saat terjadi gerakan dinamis. Kombinasi *accelerometer* dan *gyroscope* memungkinkan estimasi orientasi (*roll*, *pitch*, *yaw*) yang lebih stabil melalui teknik penggabungan data (*sensor fusion*).



Gambar 2. 8 Pergerakan Sumbu 3 Sudut [20]

Berdasarkan Gambar 2.8, orientasi suatu objek dalam ruang tiga dimensi direpresentasikan melalui tiga sudut rotasi, yaitu *roll*, *pitch*, dan *yaw*. Sudut *roll* menunjukkan rotasi terhadap sumbu X, *pitch* menunjukkan rotasi terhadap sumbu Y, sedangkan *yaw* menunjukkan rotasi terhadap sumbu Z. Ketiga sudut tersebut digunakan untuk mengetahui perubahan orientasi dan kemiringan objek selama bergerak [20].

Dalam implementasi, MPU6050 umumnya digunakan dalam bentuk modul GY-521 yang telah dilengkapi regulator tegangan dan menggunakan komunikasi I2C (SDA dan SCL), sehingga mudah diintegrasikan dengan mikrokontroler. Untuk meningkatkan akurasi dan mengurangi *noise*, pembacaan sensor biasanya dikombinasikan dengan metode *sensor fusion* seperti Kalman Filter [20]. Spesifikasi modul IMU MPU6050 dapat dilihat pada Tabel 2.6.

Tabel 2. 6 Spesifikasi dari modul IMU MPU 6050

Spesifikasi	Operasi
Chip	MPU-6050
Vin	3V ~ 5V
Communication	I2C
Gyroscopes ranges	+/- 250 500 1000 2000 degree/sec
Acceleration ranges	+/- 2g, +/- 4g, +/- 8g, +/- 16g
Pin Spacing Built-in 16bit AD converter	2.54mm (0.1in) 16 bit data output

Pada penelitian ini, data sudut yang diperoleh dari MPU6050 digunakan sebagai masukan sistem kendali untuk menentukan orientasi dan stabilitas sistem.

Sudut kemiringan dapat dihitung berdasarkan keluaran *accelerometer* menggunakan Persamaan (2.26) dan Persamaan (2.27).

$$Roll (\phi) = \tan^{-1} \left(\frac{A_y}{A_z} \right) \quad (2.26)$$

$$Pitch (\theta) = \tan^{-1} \left(\frac{-A_x}{\sqrt{A_y^2 + A_z^2}} \right) \quad (2.27)$$

Keterangan:

Keterangan variabel yang digunakan dalam perhitungan sudut kemiringan menggunakan sensor MPU6050 dapat dilihat pada Tabel 2.7.

Tabel 2. 7 Keterangan Variabel Perhitungan Sensor MPU6050

Variabel	Keterangan	Satuan
A_x	Percepatan pada sumbu X	m/s ²
A_y	Percepatan pada sumbu Y	m/s ²
A_z	Percepatan pada sumbu Z	m/s ²
ϕ_{roll}	Sudut kemiringan <i>roll</i>	°
θ_{pitch}	Sudut kemiringan <i>pitch</i>	°
ω	Kecepatan sudut gyroscope	°/s
Δt	Waktu sampling	s

Sedangkan sudut dari *gyroscope* diperoleh melalui integrasi kecepatan sudut terhadap waktu:

$$\theta = \theta_{sebelumnya} + \omega \Delta t \quad (2.28)$$

2.8 Buzzer

Buzzer merupakan komponen elektronika yang berfungsi mengubah sinyal listrik menjadi gelombang suara. Jenis yang umum digunakan adalah *piezoelectric buzzer*, yang bekerja berdasarkan efek piezoelektrik, yaitu perubahan bentuk material ketika diberi tegangan listrik sehingga menghasilkan getaran dan suara. *Buzzer* biasanya beroperasi pada tegangan 3–5 V DC dengan frekuensi sekitar 1–5 kHz serta memiliki konsumsi arus yang relatif kecil.



Gambar 2. 9 Komponen Buzzer Piezoelectric [21]

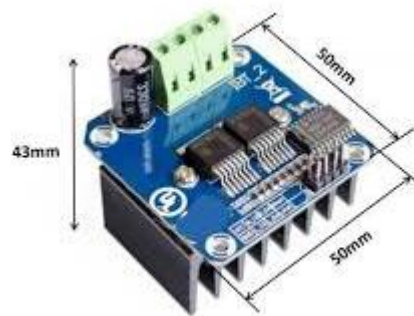
Berdasarkan Gambar 2.9, komponen *buzzer* secara umum dibedakan menjadi *active buzzer* dan *passive buzzer*. *Active buzzer* memiliki osilator internal sehingga dapat langsung berbunyi ketika diberi tegangan, sedangkan *passive buzzer* memerlukan sinyal frekuensi dari mikrokontroler untuk menghasilkan bunyi. Pada penelitian ini digunakan *active buzzer* karena pengendaliannya lebih sederhana. Spesifikasi *buzzer* yang digunakan dalam penelitian ini ditunjukkan pada Tabel 2.8 [21].

Tabel 2. 8 Spesifikasi buzzer Piezo

Spesifikasi	Operasi
Tipe	Active Piezo Buzzer
Tegangan Operasi	3 – 24V DC
Frekuensi Resonansi	±3 kHz
Konsumsi Arus	10 mA - 35 mA
Tingkat Kebisingan	±85 – 100 dB
Dimensi Fisik	12 mm × 9.7 mm

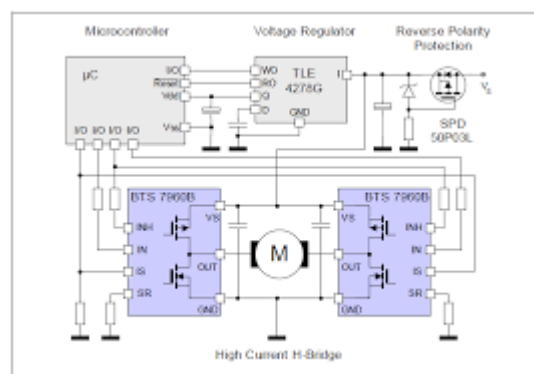
2.9 Driver BTS7960

BTS7960 merupakan modul driver motor DC berbasis H-Bridge arus tinggi yang dirancang untuk mengendalikan motor dengan kebutuhan daya besar. Modul ini menggunakan IC BTS7960 yang telah terintegrasi dengan rangkaian penguat daya serta sistem proteksi internal, sehingga mampu mengatur arah dan kecepatan motor secara efisien dan aman [22].



Gambar 2. 10 Tampilan Driver BTS7960 [22]

Berdasarkan Gambar 2.10 merupakan tampilan driver BTS7960, konfigurasi *H-Bridge* memungkinkan motor berputar dalam dua arah, yaitu searah jarum jam (*Clockwise/CW*) dan berlawanan arah jarum jam (*Counter Clockwise/CCW*), dengan mengatur polaritas tegangan pada terminal motor. Pengaturan kecepatan dilakukan menggunakan metode *Pulse Width Modulation* (PWM), di mana perubahan *duty cycle* akan mempengaruhi tegangan efektif yang diterima motor.



Gambar 2. 11 Skematik Driver BTS7960 [22]

Berdasarkan Gambar 2.11 skematik driver BTS7960 ditunjukkan konfigurasi rangkaian *H-Bridge* yang mengendalikan motor DC. Sinyal kontrol dari mikrokontroler berupa PWM dan direction masuk ke bagian *input* driver, kemudian diteruskan ke rangkaian penguat daya untuk mengatur arus yang mengalir ke motor. Jalur suplai tegangan motor terpisah dari suplai logika untuk menjaga kestabilan sistem. Konfigurasi ini memungkinkan pengendalian arah putar dan kecepatan motor secara aman dengan proteksi internal terhadap arus berlebih dan suhu berlebih.

Salah satu keunggulan utama BTS7960 adalah kemampuannya menangani arus hingga 43 A (arus puncak) dengan arus kerja normal sekitar 30 A. Selain itu, modul

ini dilengkapi proteksi terhadap arus berlebih, suhu berlebih, tegangan lebih, tegangan kurang, dan hubung singkat. Fitur ini sangat penting untuk menjaga keandalan sistem penggerak, terutama pada aplikasi motor berdaya besar. Berikut merupakan spesifikasi driver BTS7960 dapat dilihat pada Tabel 2.9.

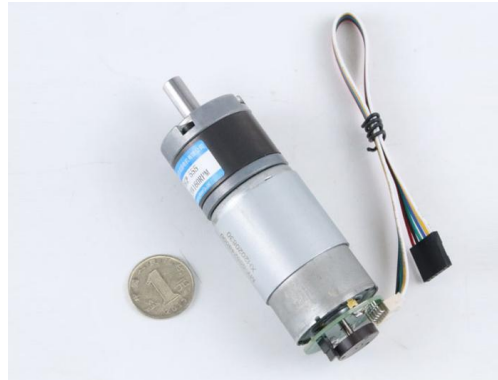
Tabel 2. 9 Spesifikasi driver BTS7960

Parameter	Spesifikasi
Tipe Modul	Driver Motor H-Bridge
Tegangan Motor	6,5 – 27 V DC
Tegangan Logika	5 V
Arus Maksimum	43 A
Arus Kerja Normal	± 30 A
Output	Bidirectional (maju + mundur)
Kontrol	PWM dan Direction

Pada penelitian ini, driver BTS7960 digunakan untuk mengendalikan motor DC penggerak roda AGV. Sinyal PWM dan direction dikirimkan dari mikrokontroler untuk mengatur kecepatan dan arah gerak kendaraan sesuai hasil keputusan sistem kendali. Dengan kemampuan arus besar dan sistem proteksi terintegrasi, modul ini mendukung pengoperasian AGV yang stabil dan aman saat membawa beban.

2.10 Motor Penggerak (PG Encoder PG36)

Motor DC (*Direct current*) merupakan aktuator listrik yang mengubah energi listrik menjadi energi mekanik berupa putaran poros melalui interaksi medan magnet dan arus listrik pada kumparan jangkar. Ketika arus dialirkan, timbul gaya elektromagnetik yang menghasilkan torsi sehingga rotor berputar. Motor DC banyak digunakan pada sistem robotika karena memiliki karakteristik torsi awal yang tinggi serta kemudahan pengaturan kecepatan menggunakan variasi tegangan atau metode PWM [4].



Gambar 2. 12 PG Encoder PG36 [23]

Pada penelitian ini digunakan motor DC gearbox tipe PG36 yang telah dilengkapi encoder incremental sebagai umpan balik kecepatan dan posisi. Gearbox berfungsi meningkatkan torsi keluaran, sedangkan encoder menghasilkan pulsa (PPR) yang digunakan untuk menghitung RPM dan mendukung sistem kendali tertutup (closed-loop control). Kombinasi gearbox dan encoder ini memungkinkan pengaturan kecepatan yang lebih stabil dan presisi, sehingga sesuai untuk aplikasi AGV berbasis roda *mecanum*[23]. Spesifikasi motor PG36 yang digunakan pada penelitian ini dapat dilihat pada Tabel 2.10.

Tabel 2. 10 Spesifikasi Motor PG36

Parameter	Spesifikasi	Satuan
Diameter Gearbox	36	mm
Tegangan Kerja	12	V
Kecepatan Tanpa Beban	160	RPM
Torsi	34.5	Nm
Shaft diameter	8	mm
Panjang	110	mm

Gaya total dihitung menggunakan Persamaan (2.29).

$$F = m \times g \quad (2.29)$$

Gaya yang bekerja pada setiap roda dihitung menggunakan Persamaan (2.30).

$$F_{roda} = \frac{F_{total}}{4} \quad (2.30)$$

Torsi minimum motor dihitung menggunakan Persamaan (2.31).

$$T = F \times r \quad (2.31)$$

Kebutuhan energi mekanik AGV dihitung menggunakan Persamaan (2.32).

$$P = F \times v \quad (2.32)$$

Arus motor dihitung menggunakan Persamaan (2.33).

$$I = \frac{P}{V} \quad (2.33)$$

Laju gerak motor dihitung menggunakan Persamaan (2.34).

$$v = \omega \times r \quad (2.34)$$

Konversi kecepatan putar motor dari satuan RPM menjadi kecepatan linear AGV dalam satuan m/s dihitung menggunakan Persamaan (2.35).

$$v_{konversi} = \frac{2\pi \cdot r \cdot n}{60} \quad (2.35)$$

Keterangan variabel yang digunakan dalam perhitungan kebutuhan mekanik dan elektrik AGV dapat dilihat pada Tabel 2.11.

Tabel 2. 11 Keterangan Variabel Perhitungan Mekanik

Variabel	Keterangan	Satuan
F	Gaya	N
m	Massa	kg
g	Gravitasi	m/s ²
T	Torsi	Nm
r	Jari-jari roda	m
P	Daya	W
v	Kecepatan linear	m/s
I	Arus	A
V	Tegangan	V
ω	Kecepatan sudut	rad/s
$v_{konversi}$	Kecepatan linear hasil konversi dari RPM	m/s
v	Kecepatan linear AGV	m/s
n	Kecepatan putar motor	RPM

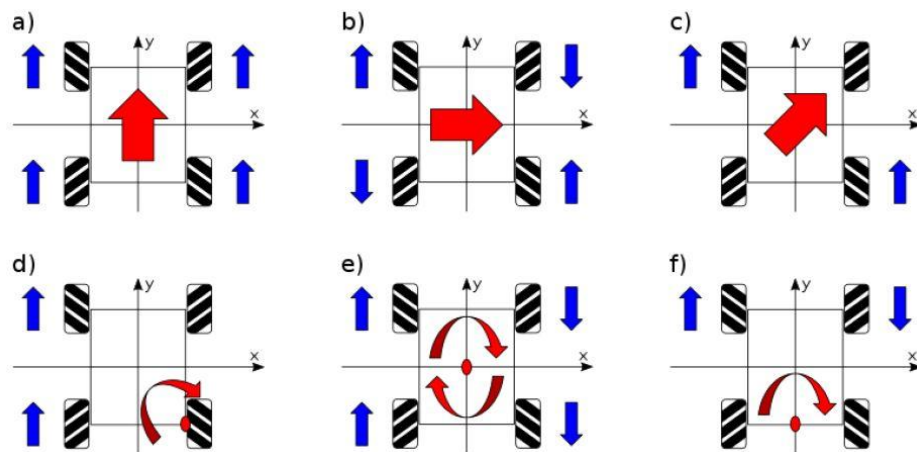
2.11 Mecanum wheel

Roda *mecanum* merupakan jenis roda *omni-directional* yang memungkinkan robot bergerak ke berbagai arah tanpa perlu mengubah orientasi bodi. Konsep ini pertama kali diperkenalkan oleh Bengt Ilon pada tahun 1970-an, di mana setiap roda dilengkapi *roller* kecil yang dipasang dengan sudut $\pm 45^\circ$ terhadap sumbu putar roda. Susunan *roller* tersebut menghasilkan kombinasi gaya vektor pada arah longitudinal dan lateral, sehingga kendaraan dapat bergerak maju, mundur, menyamping (lateral), maupun diagonal hanya dengan mengatur kecepatan dan arah putaran masing-masing roda [24] Roda *mecanum* yang digunakan pada sistem AGV dapat dilihat pada Gambar 2.13.



Gambar 2. 13 Roda *Mecanum* [25]

Pada sistem AGV yang dirancang, roda *mecanum* digunakan sebanyak empat buah dengan konfigurasi X-drive untuk memperoleh manuver yang fleksibel pada area sempit. Pergerakan translasi dan rotasi dihasilkan dari kombinasi kecepatan keempat motor yang dihitung menggunakan persamaan invers kinematika. Keunggulan utama roda *mecanum* adalah kemampuan manuver tinggi dan efisiensi pergerakan di ruang terbatas, meskipun membutuhkan sinkronisasi kecepatan yang presisi melalui sistem kendali tertutup agar stabilitas tetap terjaga [6].



Gambar 2. 14 Gerakan Omni Directional[25]

Berdasarkan Gambar 2.14 merupakan gerakan *omni directional*, pada sistem AGV yang menggunakan empat roda terdapat konfigurasi roda yang terdiri dari:

- *Front Left* (FL)
- *Front Right* (FR)
- *Rear Left* (RL)

- *Rear Right (RR)*

2.11.1 Model Kinematika AGV

Roda Model kinematika digunakan untuk menggambarkan hubungan antara kecepatan linier dan sudut robot terhadap kecepatan sudut masing-masing roda. Variabel yang digunakan pada perhitungan kinematika AGV beserta satuannya ditunjukkan pada Tabel 2.12.

Tabel 2. 12 Keterangan Variabel Kinematika AGV

Variabel	Keterangan	Satuan
V_x	Kecepatan linier arah x	m/s
V_y	Kecepatan linier arah y	m/s
ω	Kecepatan sudut robot	Rad/s
r	Jari-jari roda	m
L	Setengah panjang AGV	m
W	Setengah lebar AGV	m
v_1	Kecepatan linear roda 1	m/s
v_2	Kecepatan linear roda 2	m/s
v_3	Kecepatan linear roda 3	m/s
v_4	Kecepatan linear roda 4	M/s

Konstanta geometri robot dinyatakan sebagai: $(L+W)$

2.11.2 Persamaan Invers Kinematika

Invers kinematika digunakan untuk menentukan kecepatan sudut masing-masing roda berdasarkan kecepatan gerak robot yang diinginkan.

Persamaan invers kinematika roda *mecanum* dinyatakan sebagai berikut:

$$\omega FL = \frac{1}{r}(V_x - V_y - (L + W)\omega) \quad (2.36)$$

$$\omega FR = \frac{1}{r}(V_x + V_y + (L + W)\omega) \quad (2.37)$$

$$\omega RL = \frac{1}{r}(V_x + V_y - (L + W)\omega) \quad (2.38)$$

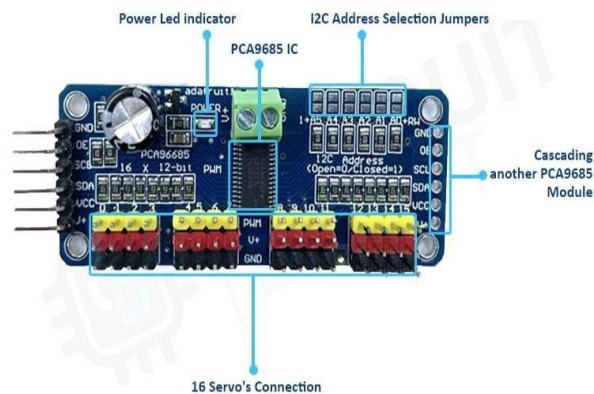
$$\omega RR = \frac{1}{r}(V_x - V_y + (L + W)\omega) \quad (2.39)$$

Keempat persamaan tersebut menunjukkan bahwa kecepatan sudut setiap roda merupakan kombinasi linier dari komponen translasi dan rotasi robot.

2.12 PCA9685

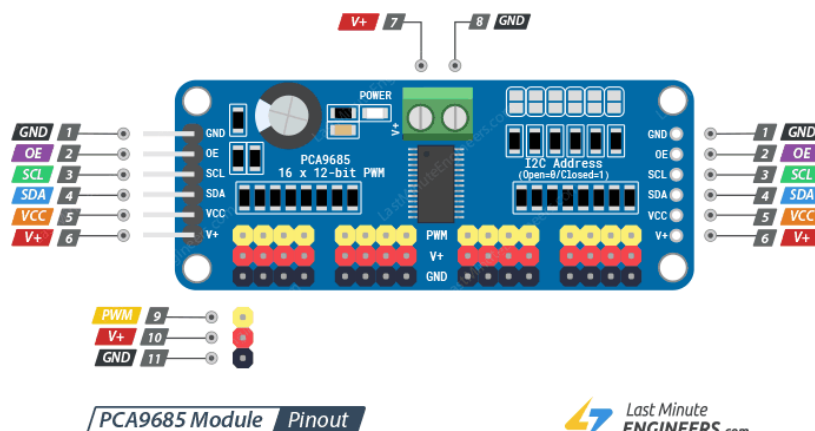
PCA9685 merupakan *Integrated Circuit (IC)* yang dikembangkan oleh NXP Semiconductors dan berfungsi sebagai pengendali sinyal *Pulse Width Modulation*

(PWM) sebanyak 16 kanal melalui komunikasi *Inter-Integrated Circuit* (I²C). IC ini dirancang untuk aplikasi pengaturan tingkat kecerahan LED, namun juga dapat digunakan untuk menghasilkan sinyal PWM pada aktuator lain seperti motor servo maupun driver motor DC. Pada setiap kanal keluaran, PCA9685 menyediakan resolusi PWM sebesar 12 bit atau setara dengan 4096 tingkat pencacahan dalam satu periode sinyal PWM. Frekuensi PWM yang dihasilkan dapat diatur melalui register internal dengan rentang frekuensi sekitar 24 Hz hingga 1526 Hz[26].



Gambar 2. 15 Modul PCA9685[26]

Berdasarkan Gambar 2.15, modul PCA9685 terdiri atas IC PCA9685, resistor *pull-up* untuk komunikasi I²C, terminal catu daya, serta enam belas kanal keluaran PWM yang diberi label PWM0 sampai PWM15. Modul ini memiliki alamat I²C bawaan sebesar 0x40 dan dapat diubah menggunakan pin alamat A0 sampai A5 sehingga memungkinkan beberapa modul PCA9685 digunakan pada satu jalur komunikasi I²C [27].



PCA9685 Module Pinout

Gambar 2. 16 Pin Modul PCA9685 [27]

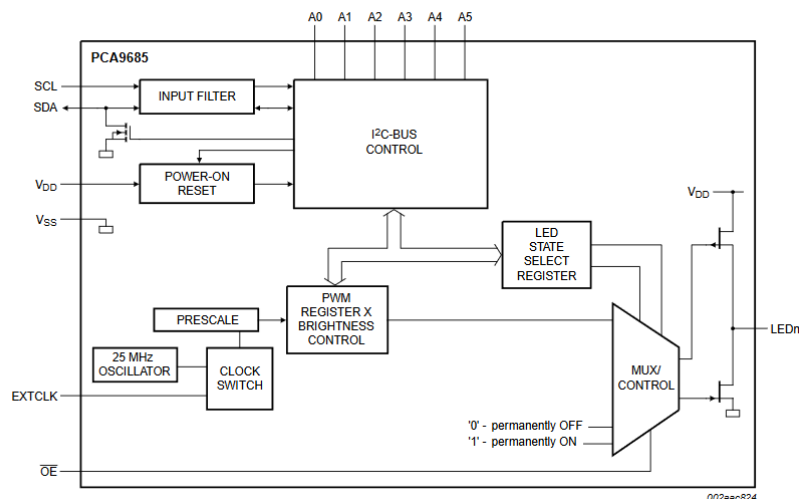
Berdasarkan Gambar 2.16, modul PCA9685 memiliki enam pin utama yang digunakan untuk komunikasi dan catu daya, yaitu pin GND, OE, SCL, SDA, VCC, dan V+. Selain itu tersedia enam belas kanal keluaran PWM yang masing-masing terdiri atas terminal PWM, V+, dan GND [28]. Spesifikasi modul PCA9685 dapat dilihat pada Tabel 2.13.

Tabel 2. 13 Spesifikasi PCA9685

Parameter	Spesifikasi
Nama IC	PCA9685
Arus source	10 mA
Produsen	NXP Semiconductors
Jenis komunikasi	Inter Integrated Circuit (I ² C)
Kecepatan I ² C	Hingga 1 MHz
Jumlah kanal PWM	16 kanal
Resolusi PWM	12 bit
Tingkat Pencacahan PWM	4096
Frekuensi PWM	24 Hz – 1526 Hz
Osilator internal	25 MHz
Tegangan operasi	2,3 V – 5,5 V
Jumlah alamat I ² C	Maksimum 62 perangkat
Arus sink	25 mA

Bagian-bagian utama modul PCA9685 dapat dilihat pada Gambar 2.17.

PCA9685



Gambar 2. 17 Bagian-bagian PCA9685 [26]

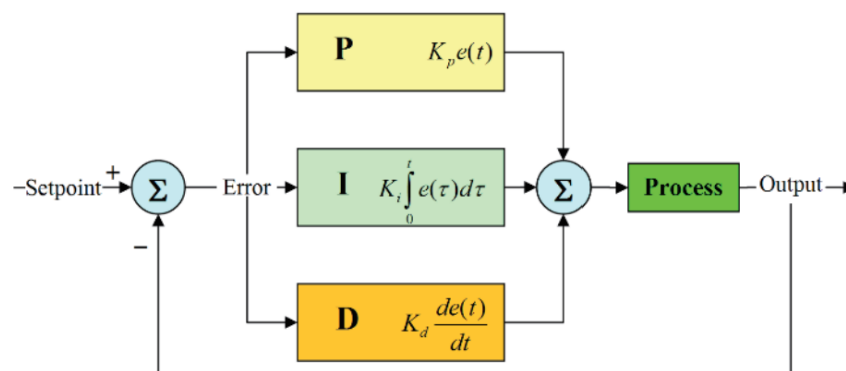
Berdasarkan Gambar 2.17, PCA9685 terdiri atas beberapa bagian utama, yaitu *Input Filter*, *Power-on Reset*, *I²C-Bus Control*, osilator internal 25 MHz, *Clock Switch*, *Prescale Register*, *PWM Register*, *LED State Select Register*, dan *MUX Control*. Komunikasi antara mikrokontroler dengan PCA9685 dilakukan melalui

jalur SDA dan SCL yang diterima oleh blok *I²C-Bus Control*. Data yang dikirimkan melalui komunikasi I²C akan disimpan pada *PWM Register* untuk menentukan nilai *duty cycle* pada masing-masing kanal keluaran [26].

Osilator internal 25 MHz digunakan sebagai sumber clock utama untuk menghasilkan sinyal PWM. Frekuensi PWM yang diinginkan dapat diatur melalui *Prescale Register*. Selanjutnya, nilai PWM yang tersimpan pada register akan diproses oleh *MUX Control* dan diteruskan menuju kanal keluaran LED0 hingga LED15. Selain itu, PCA9685 menyediakan pin *Output Enable (OE)* yang dapat digunakan untuk mengaktifkan maupun menonaktifkan seluruh keluaran PWM secara bersamaan [26].

2.13 Sistem Kendali PID

Sistem kendali *Proportional–Integral–Derivative* (PID) merupakan salah satu metode kendali umpan balik (*closed-loop control*) yang paling banyak digunakan pada pengaturan kecepatan motor karena strukturnya sederhana, mudah diimplementasikan pada mikrokontroler, dan memberikan respons yang stabil terhadap gangguan. Pada sistem AGV ini, keluaran logika *fuzzy* berupa nilai RPM tidak langsung dikirim sebagai sinyal PWM ke motor, melainkan diperlakukan sebagai nilai acuan (*setpoint*) kecepatan. Pengendali PID kemudian bertugas menjaga kecepatan aktual setiap roda agar mengikuti *setpoint* tersebut dengan membaca umpan balik dari encoder PG36 dan mengoreksi sinyal PWM secara terus-menerus [29]. Struktur dasar pengendali PID dapat dilihat pada Gambar 2.18.



Gambar 2. 18 Basic PID Controller Structure [29]

Pengendali PID bekerja berdasarkan nilai kesalahan (*error*), yaitu selisih antara kecepatan acuan (*setpoint*) dan kecepatan aktual motor yang terukur dari encoder.

Nilai *error* tersebut diolah melalui tiga komponen kendali, yaitu proporsional, integral, dan derivatif, sehingga menghasilkan sinyal kontrol untuk mengatur *duty cycle* PWM. Secara umum, sinyal kesalahan dinyatakan pada Persamaan 2.40.

$$e(t) = R_{sp}(t) - R_{act}(t) \quad (2.40)$$

Dengan $e(t)$ adalah nilai kesalahan, $R_{sp}(t)$ kecepatan acuan (*setpoint*) hasil logika *fuzzy*, dan $R_{act}(t)$ kecepatan aktual motor yang diperoleh dari pembacaan encoder.

Sinyal kontrol PID dalam bentuk kontinu diperoleh dari penjumlahan ketiga komponen kendali seperti dinyatakan pada Persamaan 2.41.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d e(t)}{d t} \quad (2.41)$$

Dengan $u(t)$ adalah sinyal kontrol keluaran PID (koreksi nilai PWM), K_p konstanta proporsional, K_i konstanta integral, dan K_d konstanta derivatif.

Ketiga komponen pada Persamaan 2.41 memiliki fungsi yang berbeda dalam membentuk respons sistem. Komponen proporsional memberikan koreksi yang sebanding dengan besar kesalahan saat ini, sehingga mempercepat respons; jika terlalu besar dapat menimbulkan osilasi. Komponen integral menjumlahkan kesalahan dari waktu ke waktu untuk menghilangkan kesalahan tunak (*steady-state error*) agar kecepatan aktual benar-benar sama dengan *setpoint*. Komponen derivatif memperhitungkan laju perubahan kesalahan untuk meredam *overshoot* dan menghaluskan respons sistem [29].

Karena pengendali diimplementasikan pada mikrokontroler ESP32-S3 yang bekerja secara diskret dengan periode pencuplikan (*sampling time*) tertentu, Persamaan 2.41 diubah ke dalam bentuk diskret seperti pada Persamaan 2.42. Integral didekati dengan penjumlahan kesalahan dan derivatif didekati dengan selisih kesalahan antara dua pencuplikan berurutan.

$$u(k) = K_p e(k) + K_i T \sum_{j=0}^k e(j) + K_d \frac{e(k) - e(k-1)}{T} \quad (2.42)$$

Dengan $u(k)$ adalah sinyal kontrol pada pencuplikan ke- k , $e(k)$ kesalahan pada pencuplikan ke- k , $e(k-1)$ kesalahan pada pencuplikan sebelumnya, dan T periode pencuplikan (sampling time) sistem.

Variabel yang digunakan dalam perhitungan kecepatan motor berdasarkan pembacaan encoder beserta satuannya dapat dilihat pada Tabel 2.14.

Tabel 2. 14 Keterangan Variabel Encoder

Variabel	Keterangan	Satuan
ΔP	Jumlah pulsa encoder selama periode sampling	pulsa
PPR	Jumlah pulsa encoder dalam satu putaran	pulsa/rev
T_s	Periode sampling	s
n	Kecepatan putar motor	RPM

Kecepatan aktual motor sebagai umpan balik dihitung dari jumlah pulsa encoder yang terbaca dalam satu periode pencuplikan. Encoder PG36 menghasilkan sejumlah pulsa per putaran (pulse per revolution/PPR), sehingga kecepatan putar dalam satuan RPM dapat dihitung menggunakan Persamaan 2.43.

$$RPM = \frac{\Delta P}{PPR} \times \frac{60}{T} \quad (2.43)$$

Dengan ΔP adalah jumlah pulsa encoder yang terbaca dalam satu periode pencuplikan, PPR jumlah pulsa per putaran, dan T periode pencuplikan (detik).

Penentuan nilai parameter kp, ki , dan kd pada pengendalian PID dapat dilakukan menggunakan metode Ziegler–Nichols berdasarkan respon open-loop system [29]. Metode ini dilakukan dengan memberikan masukan berupa sinyal langkah (*step input*) pada sistem tanpa umpan balik, kemudian mengamati respons keluarannya. Dari kurva respons tersebut dibuat garis singgung untuk memperoleh karakteristik sistem, yaitu waktu tunda (*dead time, L*) dan konstanta waktu (*time constant, T*). kedua parameter tersebut selanjutnya digunakan sebagai dasar dalam menentukan nilai awal parameter PID sesuai aturan penalaran (tuning) Ziegler-Nichols. Pada penelitian ini, proses penalaan parameter PID mengacu pada metode Ziegler-Nichols tipe 1, dengan aturan perhitungan yang ditunjukkan pada Tabel 2.15.

Tabel 2. 15 Perhitungan Parameter PID dengan Ziegler Nichols 1

Metode Kontrol	Kp	Ti	Td
P	$\frac{T}{L}$	∞	0

Metode Kontrol	Kp	Ti	Td
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PD	$1.2 \frac{T}{L}$	$2L$	$0.5 L$

Berdasarkan Tabel 2.15, untuk menentukan parameter kontrol PID dapat digunakan persamaan seperti pada rumus. Berikut

$$Kp = 1.2 \frac{T}{L} \quad (2.44)$$

$$Ti = 2 L \quad (2.45)$$

$$Td = 0.5 L \quad (2.46)$$

Berdasarkan persamaan 2.44 hingga 2.45 diperoleh L sebagai *dead time*, T sebagai *time constant*, sedangkan Ti digunakan untuk mendapatkan kecepatan integral guna mengeliminasi *Steady-state error*. Nilai Ki dapat diperoleh dengan menggunakan rumus sebagai berikut.

$$Ki = \frac{Kp}{Ti} \quad (2.47)$$

Dengan Kp merupakan konstanta proporsional, Ki merupakan konstanta integral dan Ti merupakan waktu integral. Setelah itu parameter Kd dapat ditentukan dengan menggunakan nilai Td dengan menggunakan rumus sebagai berikut.

$$Kd = Kp \cdot Td \quad (2.48)$$

Dengan Kp merupakan konstanta proporsional, Kd merupakan konstanta derivative dan Td merupakan waktu derivative. Setelah mendapatkan semua parameter Kp , Ki , dan Kd , sistem PID dapat dioperasikan dengan baik dan optimal.

Dengan demikian, pada sistem AGV ini logika *fuzzy* Mamdani berperan sebagai kendali tingkat tinggi (high-level) yang menentukan berapa RPM yang sesuai berdasarkan kondisi beban dan kemiringan, sedangkan pengendali PID berperan sebagai kendali tingkat rendah (low-level) yang memastikan kecepatan aktual setiap roda benar-benar mengikuti nilai RPM tersebut. Kombinasi dua

tingkat kendali ini banyak digunakan pada robot mobile beroda *mecanum* untuk meningkatkan akurasi dan kestabilan gerak, sebagaimana ditunjukkan pada penelitian mengenai penalaan PID kecepatan motor berumpan balik encoder, serta penggabungan logika *fuzzy* dengan PID pada robot *omnidirectional* beroda *mecanum* [29].

2.14 Antarmuka Web (*Web Interface*)

Antarmuka web merupakan media interaksi antara pengguna dan sistem tertanam (*embedded system*) yang diakses melalui peramban (*browser*) tanpa memerlukan pemasangan aplikasi khusus pada perangkat pengguna. Pada penelitian ini, antarmuka web digunakan sebagai pengganti panel tampilan fisik untuk memantau kondisi AGV secara *real time* serta memberikan perintah arah gerak. Pendekatan ini dipilih karena dapat diakses dari berbagai perangkat, baik telepon pintar maupun komputer, serta tidak menambah beban perangkat keras pada AGV.

2.14.1 *Web Server* pada Mikrokontroler

Web server merupakan perangkat lunak yang menerima permintaan (*request*) dari klien dan memberikan tanggapan (*response*) berupa halaman web maupun data. ESP32 memiliki modul *Wi-Fi* terintegrasi sehingga mampu menjalankan *embedded web server* secara mandiri tanpa memerlukan komputer sebagai server eksternal. Berkas antarmuka berupa HTML, CSS, dan JavaScript disimpan pada memori *flash* mikrokontroler, kemudian dikirimkan kepada klien saat terjadi permintaan akses [1]. Pada sistem tertanam, penggunaan *asynchronous web server* memungkinkan mikrokontroler melayani beberapa permintaan klien tanpa menghentikan proses utama. Dengan pendekatan tersebut, pembacaan sensor, perhitungan logika *fuzzy*, serta pengendalian PID tetap dapat berjalan secara bersamaan meskipun terdapat komunikasi antara *browser* dan mikrokontroler [30].

2.14.2 Mode *Access Point* dan *Station*

ESP32 mendukung dua mode operasi jaringan *Wi-Fi*, yaitu mode *Station* (STA) dan *Soft Access Point* (*SoftAP*). Pada mode STA, ESP32 bergabung ke jaringan *Wi-Fi* yang telah tersedia, sedangkan pada mode *SoftAP* ESP32

membangun jaringan *Wi-Fi* sendiri sehingga perangkat pengguna dapat terhubung secara langsung tanpa memerlukan *router* tambahan [30]. Pada penelitian ini digunakan mode *SoftAP* agar sistem AGV dapat dioperasikan secara mandiri tanpa bergantung pada infrastruktur jaringan yang tersedia di lokasi pengujian.

2.14.3 Protokol Komunikasi dan Format Pertukaran Data

Komunikasi antara *browser* dan *embedded web server* menggunakan protokol *Hypertext Transfer Protocol* (HTTP) untuk mengirimkan halaman antarmuka, sedangkan pengiriman data pemantauan secara *real time* menggunakan protokol *WebSocket*. Berbeda dengan HTTP yang bekerja berdasarkan mekanisme *request-response*, *WebSocket* menyediakan komunikasi dua arah (*full-duplex*) sehingga server dapat mengirimkan data kepada klien tanpa harus menunggu permintaan baru [31]. Data pemantauan dikirimkan dalam format *JavaScript Object Notation* (JSON), yaitu format pertukaran data berbasis teks yang ringan, mudah diproses oleh JavaScript pada sisi klien, serta sesuai untuk komunikasi *real time* pada sistem tertanam berbasis ESP32. Pada penelitian ini, setiap paket data JSON memuat informasi berat muatan, sudut *pitch*, *roll*, *yaw*, serta nilai RPM masing-masing roda untuk ditampilkan pada antarmuka web [30].