

## BAB 3

### METODE PENELITIAN

#### 3.1 ALAT YANG DIGUNAKAN

Penelitian ini membutuhkan beberapa perangkat keras dan perangkat lunak untuk mendukung pembuatan prototipe AGV robot.

##### 3.1.1 Perangkat Keras

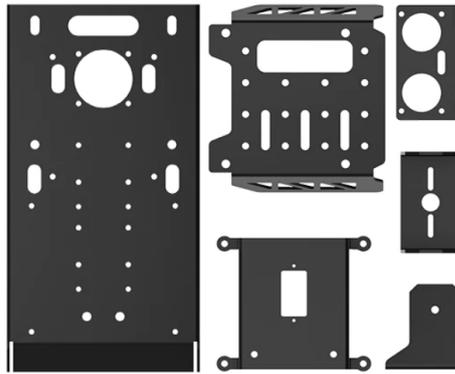
Perangkat keras (*Hardware*) yang digunakan dalam penelitian ini berupa perangkat fisik. Berikut keperluan perangkat keras yang dibutuhkan, yaitu :

**Tabel 3.1 Perangkat Keras yang Digunakan**

No	Nama Komponen	Jumlah
1	Kerangka Robot	1 Unit
2	<i>Raspberry Pi 4 Model B+</i>	1 Unit
3	<i>HD 120° Wide Angle Camera</i>	1 Unit
4	<i>Board Expansion Raspberry Pi</i>	1 Unit
5	Motor DC <i>Gearbox</i> Hitam dan Roda <i>Mecanum</i>	4 Unit
6	Baterai 18650 Li-Po 3.7v	2 Unit
7	Perangkat Penghubung dan Kabel	Secukupnya

##### 1) Kerangka Robot

Kerangka robot digunakan sebagai kerangka atau rangkaian utama yang menyatukan semua komponen dan perangkat yang dibutuhkan untuk membuat prototipe AGV robot. Dalam hal ini, kerangka yang berbahan *Hard Aluminium* dengan dimensi 200\*165\*140 mm digunakan sebagai basis atau *chassis* yang akan menopang dan mengintegrasikan semua komponen lainnya, seperti *Raspberry Pi*, roda *mecanum wheel*, modul kamera, dan komponen lainnya.



**Gambar 3.1 Kerangka Robot**

2) *Raspberry Pi 4B*

*Raspberry Pi* merupakan sebuah komputer kecil berbasis Linux yang sering digunakan dalam proyek-proyek elektronik. Pada penelitian ini, *Raspberry Pi* digunakan sebagai otak dari robot AGV, mengendalikan berbagai aspek operasional seperti mengendalikan motor, mengakses modul kamera. *Raspberry Pi* yang digunakan memiliki fitur utama dari *Raspberry Pi 4B* termasuk *prosesor quad-core* 64-bit berkinerja tinggi, dukungan layar ganda pada resolusi hingga 4K melalui sepasang *port micro-HDMI*, *dekode* video perangkat keras hingga 4Kp60, RAM hingga 8 GB, dual -band 2.4 / 5.0 GHz *wireless LAN*, *Bluetooth 5.0*, *Gigabit Ethernet*, USB 3.0, dan kemampuan PoE (melalui add-on PoE HAT terpisah).



**Gambar 3.2 Board Raspberry Pi 4B**

3) *HD 120° Wide Angle Camera USB*

Modul kamera akan digunakan untuk mengambil gambar dan video sebagai input visual untuk AGV robot. Pada penelitian ini, modul kamera yang

digunakan adalah modul kamera USB yang dilengkapi dengan *chip* OV2643 yang memiliki ukuran kecil. Modul kamera ini juga dilengkapi dengan lensa sudut lebar sebesar 120°, sehingga memberikan jangkauan pandang yang lebih luas. Resolusi kamera yang digunakan adalah 480P, yang menghasilkan efek gambar yang sangat baik.



**Gambar 3.3 HD 120° Wide Angle Camera USB**

4) Motor DC dan Roda *Mecanum*

Sistem *Mecanum Wheel* menggunakan roda khusus yang memungkinkan pergerakan lateral dan diagonal yang fleksibel. Motor DC dan roda *Mecanum* ini akan digunakan untuk menggerakkan AGV robot dalam berbagai arah dan memungkinkan manuver yang lebih presisi. Spesifikasi dari motor DC yang digunakan memiliki tegangan kerja motor: DC 3-6V. Motor DC ini memiliki kecepatan 200 RPM dan rasio 1:48. Gigi yang digunakan terbuat dari bahan plastik. Motor ini merupakan Double Axis, yang artinya dapat berputar ke dua arah. Rentang suhu operasional berkisar antara -10°C hingga +50°C, sedangkan rentang suhu penyimpanan berkisar antara -10°C hingga +40°C. Paket ini termasuk 1 motor DC.



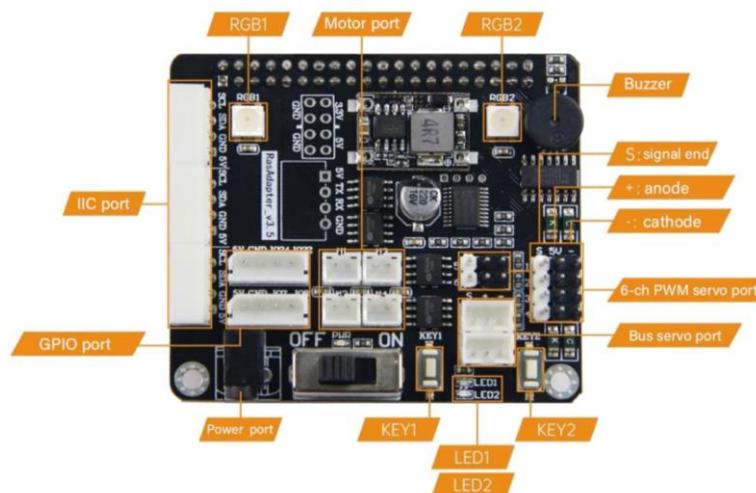
**Gambar 3.4 Motor DC dan Roda *Mecanum***

### 5) *Board extension Raspberry Pi*

*Board extension Raspberry Pi* adalah sebuah perangkat tambahan yang memungkinkan penambahan lebih banyak *port I/O (Input/Output)* pada *Raspberry Pi Board* ini dapat digunakan untuk menghubungkan komponen tambahan seperti sensor, aktuator, atau perangkat lainnya yang membutuhkan *interface I/O* tambahan.

Spesifikasi:

- Tombol saklar daya independen untuk pasokan daya yang stabil.
- 6 port PWM servo dengan perlindungan arus berlebih.
- MCU terpasang untuk mengurangi beban kerja *Raspberry Pi* dan mengurangi penggunaan sumber daya.
- Dilengkapi dengan lampu RGB, *buzzer*, lampu LED, dan sebagainya.
- Tiga *port IIC* untuk ekspansi.
- Menyediakan beberapa pin GPIO untuk pengembangan.
- Port 4PIN anti-*reverse* yang kompatibel dengan sensor.
- Empat port motor.



**Gambar 3.5 *Board extension Raspberry Pi***

### 6) Baterai 18650 Li-Po 3.7v

Baterai Li-Po (*Lithium Polymer*) digunakan sebagai sumber daya untuk mengoperasikan robot secara portabel. Baterai ini memberikan energi listrik

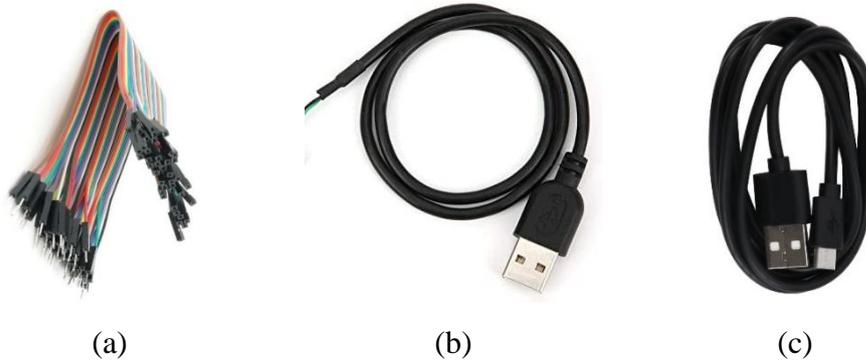
yang diperlukan untuk menggerakkan motor dan menjalankan komponen elektronik lainnya pada robot. Baterai yang digunakan adalah tipe 18650 dengan kapasitas sebesar 1800mAh dan tegangan sebesar 3.7V.



**Gambar 3.6 Baterai 18650 Li-Po 3.7v**

#### 7) Perangkat Penghubung

Untuk menghubungkan berbagai komponen dan modul, perangkat penyambung dan kabel seperti *jumper wire*, kabel USB, dan kabel *power* yang akan digunakan pada robot.



**Gambar 3.7 Perangkat Penghubung**

(a) *Jumper Wire* (b) Kabel USB (c) Kabel *Power*

### 3.1.2 Perangkat Lunak

Perangkat lunak (*Software*) yang digunakan dalam penelitian ini berupa perangkat non fisik yang nantinya akan dioperasikan pada Laptop dengan sistem operasi *Windows 10*. Berikut keperluan perangkat lunak yang dibutuhkan dalam penelitian ini, yaitu :

1) Sistem *Raspberry Pi OS*

*Raspberry Pi OS* atau Raspbian digunakan sebagai sistem operasi yang dijalankan oleh *Raspberry Pi*. Ini menyediakan lingkungan yang diperlukan untuk menjalankan aplikasi dan perangkat lunak pada robot AGV.

2) *Thonny IDE*

*Thonny IDE* digunakan sebagai *Integrated Development Environment* (IDE) untuk mengembangkan, mengedit, dan menguji kode program yang akan dijalankan pada robot AGV. *Thonny IDE* digunakan untuk mengintegrasikan dan mengimplementasikan kode program yang telah diuji coba menggunakan *Jupyter Notebook* pada robot AGV.

3) *LAB Tool*

*LAB Tool* digunakan untuk pengolahan citra dan analisis visual dengan teori ruang warna LAB dalam penelitian ini. Dalam konteks pengolahan citra, *LAB Tool* digunakan untuk mengimplementasikan metode thresholding pada gambar yang berasal dari modul kamera. Pengguna dapat mengatur nilai MIN dan MAX pada setiap parameter LAB untuk menyesuaikan ambang batas dan meningkatkan kualitas hasil analisis citra.

4) *VNC Viewer*

*VNC (Virtual Network Computing) Viewer* adalah perangkat lunak yang digunakan untuk mengakses dan mengendalikan komputer dari jarak jauh melalui jaringan. Dalam penelitian ini, *VNC Viewer* digunakan untuk mengakses sistem *Raspberry Pi* melalui hotspot jaringan dari ponsel.

5) *Ms. Excel*

Dalam penelitian ini, *Microsoft Excel* berfungsi sebagai platform penyimpanan dan pengolahan data yang diperoleh dari *Thonny IDE* sebelum diubah menjadi representasi visual berupa grafik.

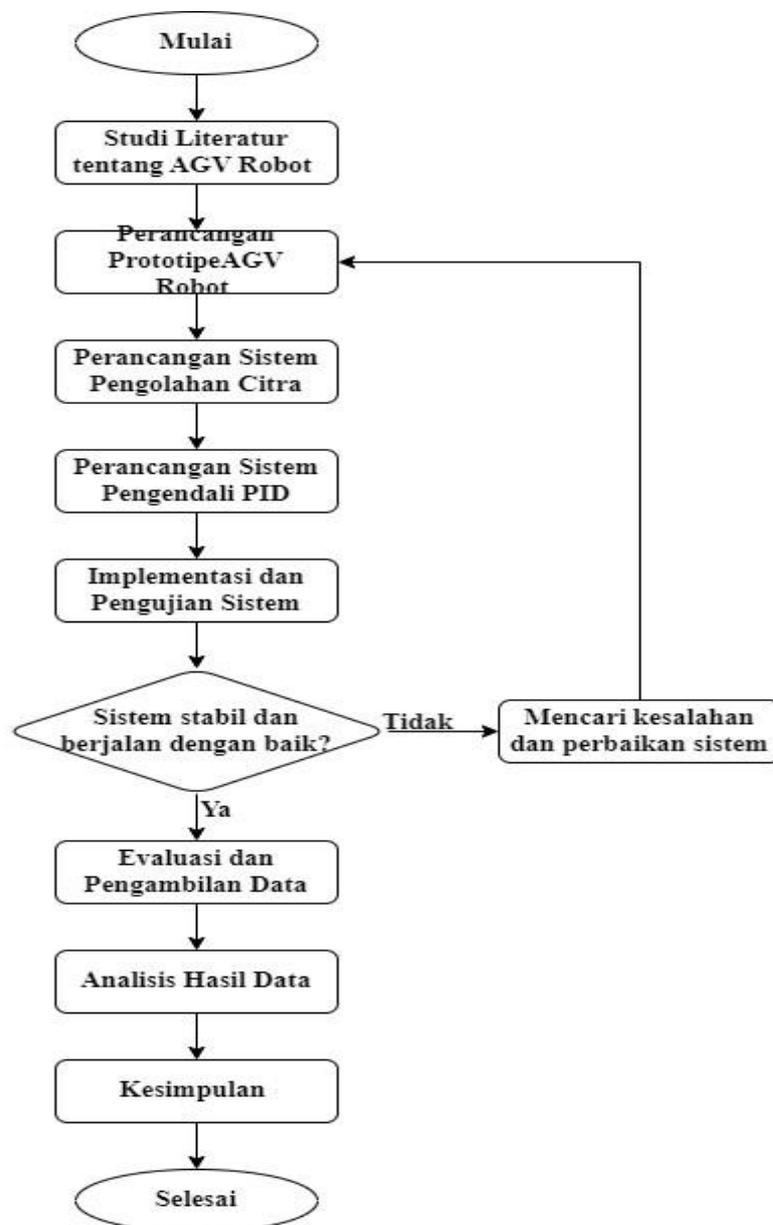
6) *Google Colab*

*Google Colab* digunakan untuk membantu dalam memahami respons robot terhadap perubahan parameter kontrol, yaitu dengan membaca data dan mengkonversinya ke dalam bentuk grafik. Hal ini memungkinkan untuk menguji dan memastikan bahwa kode program dapat diimplementasikan dengan sukses pada AGV robot.

## 3.2 ALUR PENELITIAN

### 3.2.1 *Flowchat* Penelitian

Penelitian mengenai AGV robot dengan menggunakan sistem *mecanum wheel* dan modul kamera berbasis *Raspberry Pi* dimulai dengan melakukan studi literatur, perancangan sistem, sampai dengan kesimpulan penelitian yang dapat dilihat gambar dibawah ini.



Gambar 3.8 *Flowchart* Penelitian

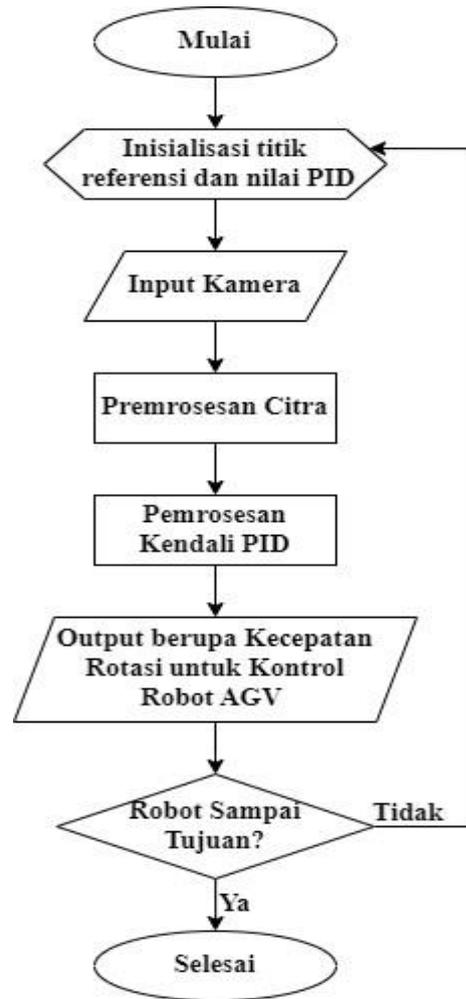
Penelitian ini terdiri dari beberapa tahap yang akan dilakukan untuk merancang dan mengimplementasikan AGV robot. Tahap pertama adalah studi literatur, di mana dilakukan pencarian dan studi terhadap referensi yang berkaitan dengan AGV robot untuk memperoleh pengetahuan dasar dan memahami perkembangan terkini serta teknologi terkait. Tahap selanjutnya adalah perancangan sistem, di mana dilakukan perancangan blok diagram dan rangkaian sistem AGV robot serta penggunaan metode yang akan digunakan.

Setelah perancangan, tahap implementasi dilakukan dengan memasang dan menghubungkan komponen-komponen yang telah dirancang, seperti *mecanum wheel*, modul kamera, *Raspberry Pi*, dan motor DC. Selain itu, perangkat lunak seperti pengolahan citra dan kontrol PID juga diimplementasikan pada *Raspberry Pi*. Tahap berikutnya adalah pengujian sistem, di mana sistem AGV robot yang telah diimplementasikan akan diuji untuk memastikan kinerjanya. Pengujian meliputi sistem *mecanum wheel*, sistem kamera, dan perancangan kendali PID, dan jika ditemukan kekurangan atau sistem tidak berjalan dengan baik, maka perbaikan atau pengembangan akan dilakukan.

Hasil uji akan dievaluasi dan dianalisis pada tahap evaluasi dan analisis hasil uji untuk dilakukan pengambilan data. Data pengujian akan dievaluasi untuk mengetahui sejauh mana kinerja AGV robot dan jika diperlukan, perbaikan atau pengembangan akan dilakukan. Tahap terakhir adalah analisis hasil data dan pengambilan kesimpulan, di mana jawaban atas pertanyaan penelitian akan dijabarkan. Kesimpulan ini mencakup keberhasilan sistem AGV robot, keterbatasan atau masalah yang ditemukan, serta saran untuk pengembangan dan perbaikan sistem di masa depan. Dengan demikian, tahapan tersebut membentuk alur penelitian yang komprehensif untuk mendesain dan mengimplementasikan AGV robot.

### **3.2.2 Flowchart Sistem**

Berikut adalah *flowchart* algoritma yang menjelaskan sistem kerja robot AGV dengan menggunakan sistem *Mecanum Wheel* dan modul kamera berbasis *Raspberry Pi*:



**Gambar 3.9 Flowchart Sistem**

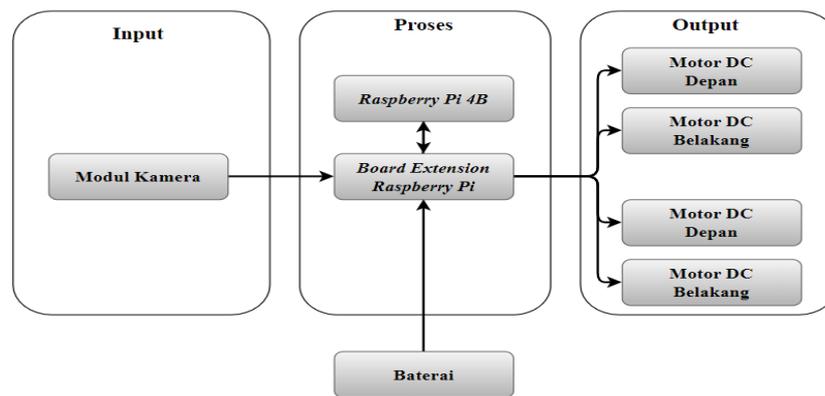
*Flowchart* algoritma diatas menjelaskan sistem kerja AGV robot dengan menggunakan sistem *Mecanum Wheel* dan modul kamera berbasis *Raspberry Pi* yang dimulai dengan inisialisasi titik referensi dan nilai-nilai PID yang menjadi parameter kunci dalam proses pengendalian AGV robot. Setelahnya, sistem memperoleh input dari kamera. Citra tersebut kemudian menjalani serangkaian proses pemrosesan, termasuk resizing, Gaussian Blur, konversi ke format LAB, dan segmentasi warna untuk mengidentifikasi objek atau fitur tertentu. Pada tahap ini, nilai *error* dari posisi garis tengah objek dihitung dan dijadikan masukan untuk pengendalian PID. Pengendalian PID menghasilkan *Output* berupa kecepatan rotasi yang dibutuhkan untuk mengarahkan AGV robot menuju posisi yang diinginkan. *Output* ini digunakan sebagai kontrol untuk menggerakkan dan mengarahkan robot sesuai dengan tugas atau tujuan yang telah ditentukan.

Sistem selanjutnya mengevaluasi apakah AGV robot telah mencapai tujuan akhirnya. Jika sudah, program dianggap selesai. Namun, jika belum, sistem kembali ke langkah pembacaan input kamera, memulai kembali siklus pemantauan dan pengendalian AGV robot.

### 3.3 PERANCANGAN SISTEM

#### 3.3.1 Blok Diagram Sistem

Blok diagram sistem untuk robot AGV dengan menggunakan sistem *Mecanum Wheel* dan modul kamera berbasis *Raspberry Pi* dapat dilihat pada gambar di bawah ini. Blok diagram ini memberikan visualisasi tentang rancangan keseluruhan sistem yang akan digunakan.

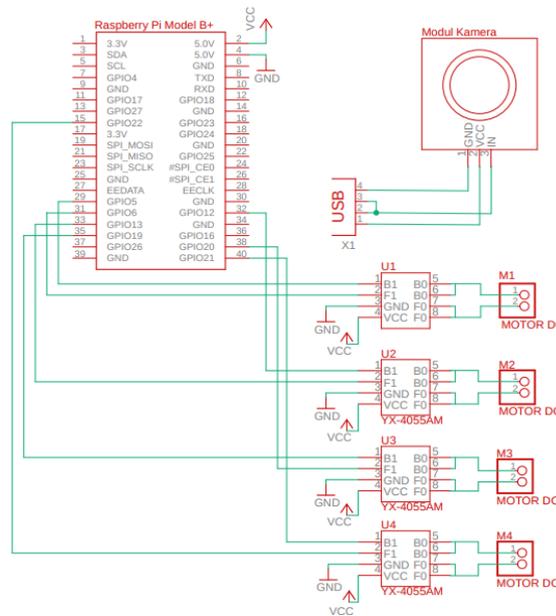


**Gambar 3.10 Blok Diagram Sistem**

Diagram blok ini menunjukkan bahwa sistem robot AGV menggunakan sistem *mecanum wheel* dan modul kamera berbasis *Raspberry Pi* terdiri dari *input*, proses, dan *Output* yang saling terhubung dan bekerja sama dalam menggerakkan robot serta mendeteksi objek di sekitarnya. Dalam diagram blok sistem tersebut, *input* berupa data dari modul kamera masuk ke proses yang dilakukan oleh *Raspberry Pi* dan *board extension*. Proses tersebut meliputi pemrosesan data, pengolahan citra, dan pengambilan keputusan berdasarkan input yang diterima. Hasil dari proses tersebut menghasilkan sinyal kontrol yang diteruskan ke *Output*, yaitu motor DC. Motor DC digunakan untuk menggerakkan pergerakan AGV robot.

### 3.3.2 Rangkaian Skematik Robot

Dalam sistem AGV robot, terdapat beberapa komponen yang saling terhubung melalui rangkaian skemati robot. *Raspberry Pi 4B* dan *board extension* berfungsi sebagai otak sistem, mengatur semua komponen lainnya, termasuk penggunaan motor DC dengan *Driver YX-5055AM*. Motor DC digunakan sebagai penggerak pada robot AGV, dan *Driver YX-5055AM* bertindak sebagai pengendali untuk motor DC tersebut. Rangkaian skematik ini menggambarkan hubungan koneksi antara setiap komponen, memastikan bahwa sinyal dan daya dapat mengalir dengan benar untuk mengoperasikan AGV robot secara optimal. Berikut rangkaian skematik pada AGV robot dan tabel pemetaan dari pin GPIO *Raspberry*



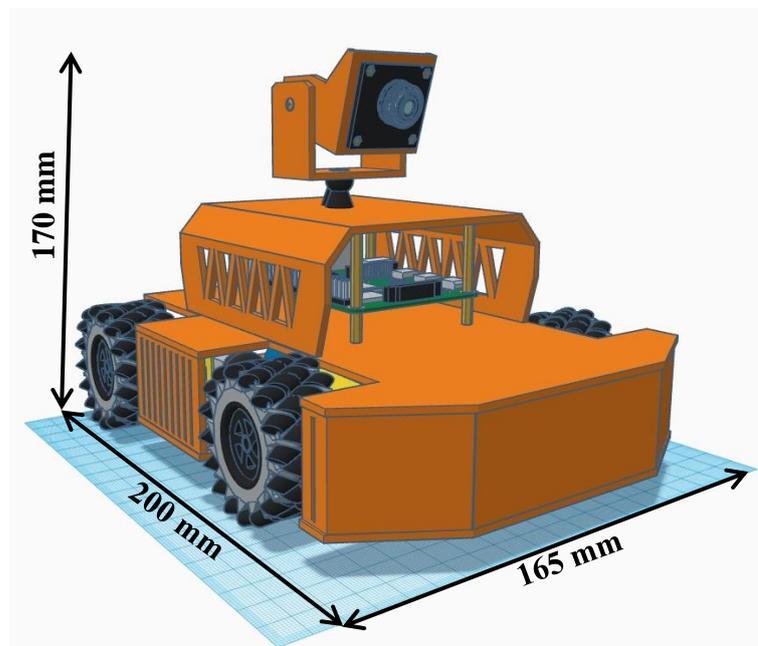
**Gambar 3.11 Rangkaian Skematik Robot**

**Tabel 3.2 Pemetaan Pin Motor DC dengan *Raspberry Pi***

Motor DC	Warna Kabel	<i>Raspberry Pi</i>	Motor DC	Warna Kabel	<i>Raspberry Pi</i>
Motor 1B	Merah	GPIO05	Motor 3B	Merah	GPIO19
Motor 1F	Hitam	GPIO06	Motor 3F	Hitam	GPIO20
Motor 2B	Merah	GPIO12	Motor 4B	Merah	GPIO21
Motor 2F	Hitam	GPIO13	Motor 4F	Hitam	GPIO22

### 3.3.3 Prototipe Robot

Prototipe AGV robot menggunakan bahan metal dengan dimensi 200 \* 165 \* 170mm. AGV robot ini menggunakan 2 baterai Lipo 18650 dengan tegangan 3.7V sebagai sumber daya. Baterai Lipo 18650 bertugas memberikan daya operasional pada robot AGV. Robot ini dilengkapi dengan kamera beresolusi 480p yang berfungsi untuk mengambil citra visual yang digunakan dalam pengolahan citra dan pengambilan keputusan oleh sistem navigasi robot. Sistem pergerakan pada robot ini menggunakan 4 roda *mecanum wheel* dan dikendalikan oleh mikrokontroler *Raspberry Pi 4B*.



**Gambar 3.12 Prototipe Robot**

Pada tampilan gambar diatas, prototipe robot menampilkan semua komponen utama dan fitur-fitur yang ada. Ini memberikan gambaran tentang dimensi robot, proporsi, dan penempatan komponen dalam desain prototipe secara keseluruhan.

### 3.3.4 Sistem Pembacaan Objek

Perancangan sistem pembacaan objek menggunakan modul kamera dengan citra biner yaitu metode pengolahan menggunakan *thresholding* adalah salah satu metode umum yang digunakan dalam pengolahan citra. Berikut adalah blok diagram untuk sistem pembacaan objek menggunakan modul kamera dengan pengolahan citra metode *thresholding*.



**Gambar 3.13 Blok Diagram Sistem Pembacaan Objek**

.Pada awal pemrosesan citra, dilakukan resizing untuk mengurangi beban komputasi. Selanjutnya, diterapkan operasi gaussian blur untuk menghilangkan noise dan meningkatkan kehalusan citra. Citra kemudian dikonversi ke format lab untuk mempermudah segmentasi warna. Proses segmentasi warna bertujuan mengidentifikasi area dalam citra sesuai dengan target warna. Hasilnya dibersihkan dari noise dan area kecil melalui operasi morfologi. Kontur pada citra hasil segmentasi diidentifikasi, dan dipilih kontur terbesar sebagai objek utama yang ingin dideteksi.

Pada tahap ini, citra hasil dievaluasi untuk menemukan kontur terbesar. Posisi garis tengah objek dihitung berdasarkan titik tengah kontur tersebut. Nilai *error* pada penulisan posisi garis tengah dihitung sebagai selisih antara nilai *setpoint* (pusat tengah citra) dan nilai aktual (posisi pusat garis) dari perhitungan. *Error* ini menjadi parameter utama dalam pengendalian PID, memungkinkan sistem menghasilkan *Output* yang mengarahkan kendaraan menuju posisi yang diinginkan.

### 3.3.5 Sistem Pengendali PID

Perancangan sistem pengendali PID pada robot AGV dengan metode *trial and error* bertujuan untuk mendapatkan nilai konstanta proporsional ( $K_p$ ), integral ( $K_i$ ), dan derivatif ( $K_d$ ) yang optimal untuk mencapai kinerja yang diinginkan. Metode *trial and error* melibatkan iterasi berulang dalam mencoba kombinasi nilai-nilai konstanta PID yang berbeda dan mengamati respons sistem terhadap perubahan tersebut.

#### 1) Perhitungan *Error*

Kesalahan (*error*) merupakan selisih antara *setpoint* (posisi tengah citra) dan nilai aktual (posisi pusat objek) pada suatu waktu tertentu. Dalam konteks ini, citra gambar diambil dengan ukuran 640x480 piksel, sehingga *setpoint* diambil pada sumbu *y* dan nilai *setpoint* yang digunakan adalah 320 piksel.

$$Error = Setpoint - Nilai Aktual \quad (3.1)$$

#### 2) Perhitungan Integral

Integral mengukur akumulasi dari kesalahan sepanjang waktu. Ini membantu mengatasi kesalahan sistem yang terakumulasi seiring waktu. Integral (integral) pada suatu waktu *t* dihitung dengan menjumlahkan kesalahan saat itu dengan nilai integral pada waktu sebelumnya.

$$Integral = Integral\ Sebelumnya + Error \quad (3.2)$$

3) Perhitungan *Derivative*

Turunan mengukur kecepatan perubahan kesalahan. Ini membantu mengurangi *overshoot* dan menyesuaikan respons sistem terhadap perubahan yang cepat. Turunan (*Derivative*) pada suatu waktu  $t$  dihitung dengan mengurangkan kesalahan saat itu dengan kesalahan pada waktu sebelumnya.

$$\text{Derivative} = \text{Error} - \text{Derivative Sebelumnya} \quad (3.3)$$

4) Perhitungan *Output* PID

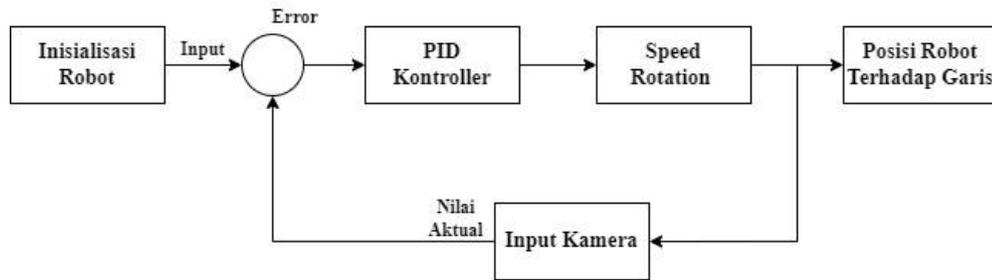
*Output* PID adalah hasil dari kombinasi proporsional (P), integral (I), dan *Derivative* (D), dihitung dengan rumus kontrol PID.

$$\text{PID} = (P \times \text{Error}) + (I \times \text{Integral}) + (D \times \text{Derivative}) \quad (3.4)$$

5) Penentuan Kecepatan Rotasi

Kecepatan Rotasi (*Rotation Speed*) adalah nilai *Output* PID. *Output* PID diatur seminimal mungkin agar sistem tetap stabil dan berada pada posisi jalur lintasan yang diinginkan, dengan rentang nilai untuk kecepatan rotasi antara -2 hingga 2. Dalam konteks ini, perlu dicatat bahwa nilai 320 piksel menunjukkan bahwa kecepatan rotasi robot pada saat itu adalah 0 rad/s. Dengan kata lain, robot sedang mempertahankan posisi tepat pada setpoint visualnya tanpa mengalami perubahan kecepatan rotasi pada saat tersebut. Ketika *Output* PID bernilai negatif, ini menunjukkan bahwa objek yang terdeteksi berada di piksel  $>320$ , sedangkan jika *Output* PID positif, hal tersebut mengindikasikan bahwa objek yang terdeteksi berada di piksel  $<320$ .

Berikut merupakan blok diagram sistem pengendali PID dengan input berupa nilai titik referensi, proses perhitungan nilai *error*, dan *Output* berupa nilai *speed rotation* (rad/s) dapat dilihat pada gambar berikut:



**Gambar 3.14 Blok Diagram Sistem Pengendali PID**

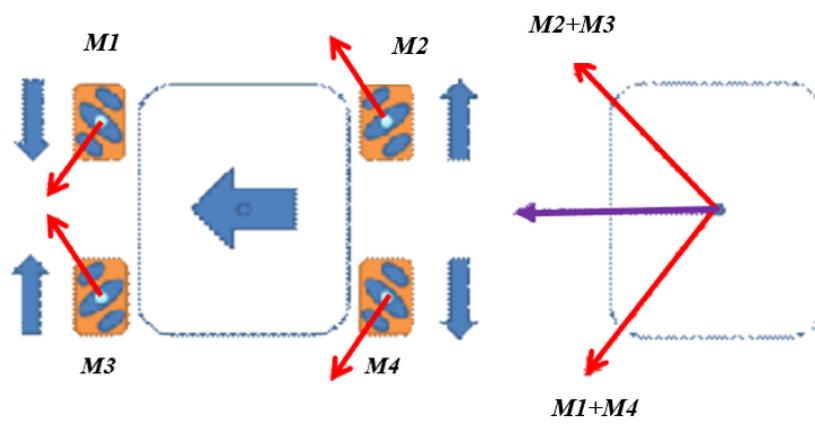
Dalam blok diagram sistem pengendali PID, terdapat beberapa komponen utama. Input dari sistem adalah inisialisasi nilai PID dan titik referensi (*setpoint*) yang digunakan untuk menentukan posisi robot terhadap garis. Nilai *error* dihasilkan dari perbedaan antara posisi aktual robot (posisi pusat objek) dan *setpoint* citra (posisi tengah citra). *Error* tersebut digunakan oleh kontroler PID untuk menghasilkan sinyal kendali yang menggerakkan motor DC dan mengubah orientasi robot. *Output* dari sistem adalah pergerakan robot, yang berfungsi sebagai umpan balik (*feedback*) untuk memperbarui nilai titik pusat gambar. Dengan demikian, sistem dapat terus memperhitungkan perubahan posisi dan menghasilkan sinyal kendali yang lebih akurat setiap kali kontrol dilakukan. Sistem ini menggunakan umpan balik dari modul kamera untuk secara kontinu menghitung nilai *error*. Sebagai hasilnya, sistem menghasilkan nilai *speed rotation* yang sesuai untuk menggerakkan motor DC dengan presisi. Proses ini berulang secara iteratif, memastikan AGV robot tetap berada pada lintasan garis yang diinginkan.

### 3.4 METODE PENGUJIAN

Metode pengujian yang akan dilakukan dalam penelitian ini meliputi pengujian terhadap sistem *mecanum wheel*, sistem modul kamera, dan perancangan sistem kendali PID pada robot AGV berbasis *Raspberry Pi* dengan tujuan untuk memvalidasi desain dan implementasi serta mengukur kinerja sistem secara keseluruhan. Berikut adalah penjelasan mengenai metode pengujian untuk setiap sistem.

### 3.4.1 Pengujian Sistem Mecanum Wheel

Pengujian sistem *mecanum wheel* akan dilakukan untuk mengevaluasi kinerja dan responsivitas roda *mecanum* dalam pergerakan robot AGV. Dilakukan pengujian untuk menguji kemampuan pergerakan robot AGV dengan sistem *mecanum wheel* dalam berbagai arah (maju, mundur, kanan, kiri, dan serong) serta kemampuan putar pada tempat. Pengujian ini akan melibatkan pengamatan visual terhadap pergerakan robot.



Gambar 3.15 Konfigurasi Pengujian Arah Gerak Robot

### 3.4.2 Pengujian Sistem Modul Kamera

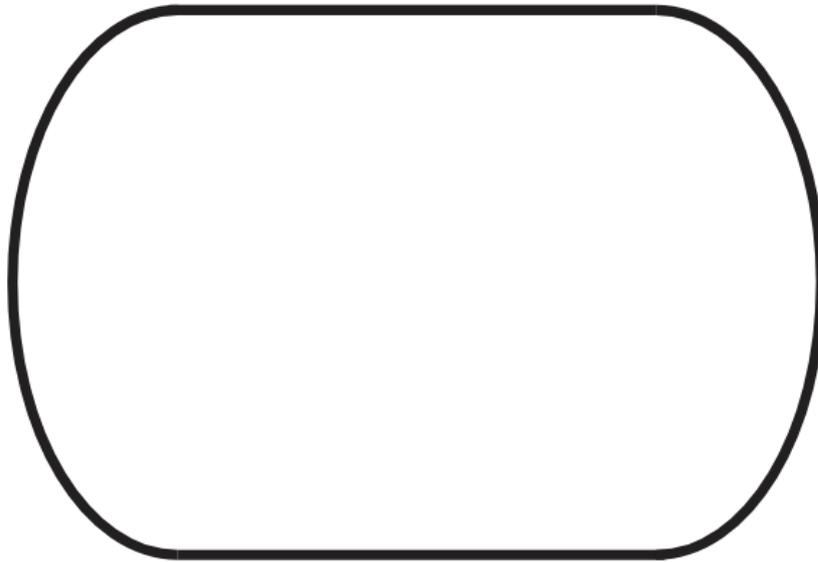
Pengujian sistem modul kamera menggunakan metode *thresholding* akan dilakukan untuk menguji kemampuan modul kamera dalam pengenalan objek, terutama objek berupa garis. Pengujian ini akan melibatkan pengambilan gambar dan pengolahan citra dengan menggunakan metode *thresholding* untuk mendeteksi garis pada lingkungan sekitar robot AGV. Pengujian ini akan melibatkan pengambilan gambar garis jalur.



Gambar 3.16 Sample Tangkapann Kamera

### 3.4.3 Pengujian Sistem Pengendali PID

Pengujian sistem navigasi menggunakan metode PID akan dilakukan untuk menguji kemampuan sistem navigasi dalam mengatur pergerakan robot AGV berdasarkan input dari modul kamera dan tujuan yang ditentukan. Pengujian ini akan melibatkan pengaturan PID untuk mengontrol kecepatan dan arah pergerakan robot AGV, serta melakukan perjalanan dari satu titik ke titik lain dengan mengikuti garis atau rute yang telah ditentukan. Pengujian akan melibatkan pengujian kestabilan gerakan robot dalam mengikuti jalur, responsifitas terhadap perubahan jalur, dan ketepatan posisi robot pada jalur yang ditentukan.



**Gambar 3.17 Jalur Lintasan Robot AGV**

Jalur lintasan pengujian memiliki ukuran persegi dengan panjang dan lebar sebesar 180 x 100 cm. Jalur tersebut dibuat dengan menggunakan material yang memiliki ketebalan sebesar 2 cm. Warna jalur yang digunakan adalah hitam dengan latar belakang berwarna putih.