BAB 2
DASAR TEORI

2.1 KAJIAN PUSTAKA

AGV robot (*Automated Guided Vehicle*) adalah jenis robot yang secara otomatis dapat melakukan transportasi barang dari satu lokasi ke lokasi lainnya. Dalam industri logistik, robot AGV sangat penting untuk meningkatkan efisiensi produksi dan mengurangi biaya operasional. *Mecanum wheel* digunakan untuk meningkatkan kemampuan robot AGV bergerak dalam semua arah dengan bebas. Modul kamera pada robot AGV digunakan untuk meningkatkan akurasi dalam pembacaan objek garis. Beberapa penelitian sebelumnya telah dilakukan mengenai desain dan implementasi robot AGV menggunakan sistem *mecanum* dan modul kamera untuk industri logistik diantaranya sebagai berikut:

2.1.1 Implementasi Modul Kamera

Penelitian yang dilakukan oleh J.H Li, Y.S Ho, dan J.J Huang pada tahun 2018 yang berjudul "***Line Tracking with Pixy Cameras on a Wheeled Robot Prototype***" bertujuan untuk mengembangkan sebuah prototipe robot beroda dengan sistem pelacakan garis menggunakan modul kamera *Pixy*. Penelitian ini menggunakan modul kamera *Pixy* dengan spesifikasi *processor* NXP LPC4330, 240 MHz, dual core dan image sensor Omnivision OV9715 dengan resolusi 1280x800. Modul kamera Pixy memiliki kemampuan pengolahan citra terintegrasi dengan kecepatan 50 FPS dan mampu melacak hingga 7 warna yang berbeda. Penelitian ini menggunakan metode penelitian eksperimental dengan merancang prototipe robot beroda yang dilengkapi dengan modul kamera *Pixy* dan menguji kemampuannya dalam mengikuti jalur garis. Hasil penelitian menunjukkan bahwa prototipe robot dapat bekerja dengan baik dalam mengikuti jalur garis yang telah ditentukan. Hal ini dapat terjadi karena modul kamera *Pixy* dapat mengenali dan mengidentifikasi jalur garis dengan baik, bahkan ketika jalur tersebut berada dalam kondisi yang sulit seperti berubah arah atau terputus [7].

Penelitian yang dilakukan oleh D. P. de Oliveira, W. P. N. Dos Reis, dan O. Morandin Junior pada tahun 2019 berjudul "***A Qualitative Analysis of a USB Camera for AGV Control***" membahas tentang penggunaan modul kamera pada robot AGV (*Automated Guided Vehicle*) untuk mendukung sistem pengendalian. Metode yang digunakan berupa eksperimental dengan melakukan pengujian pada prototipe robot AGV dengan sistem *mecanum wheel* dan modul kamera berbasis *Raspberry Pi*. Modul kamera USB yang digunakan dilengkapi dengan chip CMOS dan lensa sudut lebar 73° untuk memberikan jangkauan pandang yang luas. Modul kamera ini memiliki resolusi tertinggi 1280 x 720 dengan transfer rate 30 FPS dan ukuran yang kompak sehingga mudah dipasang dengan interface USB 2.0. Pengujian dilakukan dengan cara memvariasikan nilai *threshold* (T) pada pengolahan citra untuk mendapatkan hasil terbaik dalam mendeteksi garis-garis pada jalur pengendalian AGV. Hasil penelitian menunjukkan bahwa penggunaan modul kamera USB pada robot AGV dapat meningkatkan kemampuan kontrol dan navigasi, terutama pada area yang kompleks dan sulit dijangkau oleh sistem kontrol AGV. Selain itu, pengolahan citra pada modul kamera juga mampu mendeteksi dan mengikuti jalur pengendalian dengan lebih akurat dan efektif [8].

Penelitian yang dilakukan oleh M. Rizal, W. Djurianto, dan M. Rif'an pada tahun 2021 yang berjudul "**Implementasi Kamera OV7670 Sebagai Pendeteksi Garis Pada Robot Line Follower**" membahas tentang penggunaan kamera OV7670 sebagai sensor untuk mendeteksi garis pada robot *line follower*. Penelitian ini bertujuan untuk mengimplementasikan kamera OV7670 pada robot agar dapat mengenali dan mengikuti garis yang telah ditentukan. Modul kamera OV7670 yang digunakan memiliki spesifikasi *image* sensor Modul Kamera CMOS, dengan rentang frekuensi 10-48 MHz, resolusi VGA 640x480, dan *transfer rate* 30 FPS. Modul kamera OV7670 yang digunakan memiliki spesifikasi image sensor Modul Kamera CMOS, dengan rentang frekuensi 10-48 MHz, resolusi VGA 640x480, dan transfer rate 30 FPS. Dengan sudut tangkap vertikal sebesar 25° dan sudut tangkap horizontal 33°. Dalam penelitian ini, penulis menjelaskan pengaturan dan penggunaan kamera OV7670, termasuk pengaturan resolusi, kecepatan *frame*, dan komunikasi dengan mikrokontroler. Peneliti juga memaparkan tentang algoritma pengolahan citra yang digunakan untuk mendeteksi garis pada gambar yang diambil

oleh kamera. Dalam pengujian penelitian ini, kamera ditempatkan pada ketinggian 14,5 cm dengan sudut 25°, dan hasilnya menunjukkan bahwa konfigurasi tersebut telah mendukung navigasi line following. Selain itu, pengaturan kamera OV7670 oleh mikrokontroler melalui antarmuka SCCB berjalan lancar, terbukti dari data yang terbaca oleh mikrokontroler sesuai dengan informasi pada datasheet OV7670. Hasil penelitian ini menunjukkan bahwa kamera OV7670 mampu mendeteksi garis dengan akurasi yang memadai pada robot *line follower* yang diimplementasikan [3].

Kesimpulan dari ketiga referensi tersebut adalah bahwa penggunaan modul kamera, seperti modul kamera OV7670, modul kamera *Pixy*, dan modul kamera USB, pada robot AGV memiliki manfaat signifikan dalam meningkatkan kemampuan kontrol, navigasi, dan pelacakan garis. Penelitian menunjukkan bahwa resolusi pada modul kamera yang digunakan tidak memiliki dampak besar pada performa robot dalam melacak garis. Meskipun modul kamera dengan resolusi rendah dapat digunakan selama masih mampu menangkap gambar dengan jelas, hal ini dapat diimplementasikan pada robot AGV. Selain itu, penggunaan modul kamera USB lebih mudah diimplementasikan dibandingkan dengan modul kamera lainnya karena memiliki konfigurasi yang lebih sederhana. Modul kamera USB umumnya dapat langsung terhubung ke komputer atau perangkat mikrokontroler dengan antarmuka USB standar, sehingga tidak memerlukan komponen tambahan atau konfigurasi yang rumit.

2.1.2 Implementasi *Mecanum Wheel*

Penelitian yang dilakukan oleh Bayu, Budi, Sarwono, Tirza, Apriaskar, Esa, dan Fahmizal (2020), berjudul "**Desain Robot *Holonomic* berbasis Roda *Mecanum* dengan Arm Manipulator**". Penelitian ini membahas desain robot holonomik dengan roda *Mecanum* dan manipulator lengan. Penulis mengenalkan desain robot holonomik dengan roda *Mecanum* dan menjelaskan prinsip kerja serta keuntungannya dalam meningkatkan manuverabilitas dan mobilitas robot. Selain itu, penulis membahas implementasi manipulator lengan yang memungkinkan robot melakukan tugas manipulasi. Hasil penelitian menunjukkan bahwa desain robot holonomik dengan roda *Mecanum* dan manipulator lengan bekerja dengan

baik. Robot ini memiliki kemampuan gerakan bebas dan presisi dalam semua arah, serta mampu mengoperasikan manipulator lengan dengan baik. Penggunaan roda *Mecanum* pada robot holonomik memungkinkan gerakan lateral, diagonal, dan rotasi dengan mudah. Namun, penelitian ini belum menjelaskan metode pengendalian yang digunakan untuk menggerakkan robot holonomik dengan presisi. Oleh karena itu, penelitian dan implementasi metode pengendalian yang tepat perlu dipertimbangkan untuk mencapai presisi yang diperlukan dalam aplikasi industri logistik [9].

Penelitian yang dilakukan oleh T. Giurgiu, G. Bârsan, I. Virca, dan C. Pupăză pada tahun 2022 yang berjudul "***Mecanum Wheeled Platforms for Special Applications***" membahas tentang pengembangan platform beroda *Mecanum* untuk aplikasi khusus. Penelitian ini bertujuan untuk meningkatkan manuverabilitas dan kecepatan pada robot AGV (*Automated Guided Vehicle*) menggunakan sistem roda *Mecanum*. Para peneliti menguji *platform* roda *Mecanum* pada kendaraan truk dan mobil. Hasil pengujian menunjukkan bahwa *platform* roda *Mecanum* meningkatkan kemampuan manuver dan kecepatan pada kedua kendaraan tersebut. Penggunaan roda *Mecanum* memungkinkan gerakan lateral dan diagonal yang penting dalam pengiriman barang di area sempit dan berbelok-belok. Penelitian ini juga menunjukkan bahwa sistem pengolahan citra pada modul kamera *Raspberry Pi* mendukung fungsi robot AGV. Modul kamera ini dapat mengambil gambar barang dan membantu robot AGV dalam pengambilan keputusan [10].

Penelitian yang dilakukan oleh M. Hijikata, R. Miyagusuku, dan K. Ozaki pada tahun 2022 yang berjudul "***Wheel Arrangement of Four Omni Wheel Mobile Robot for Compactness***". Penelitian ini membahas tentang pengaturan roda pada *robot mobile* dengan empat roda *omnidirectional* untuk mencapai kompakitas. Tujuan dari penelitian ini adalah untuk mengoptimalkan pengaturan roda sehingga *robot mobile* dapat bergerak dengan efisien dan memenuhi kebutuhan ruang yang terbatas. Hasil penelitian menunjukkan bahwa pengaturan empat roda *omnidirectional* pada robot *mobile* memungkinkan gerakan yang lebih fleksibel dan kemampuan manuver yang tinggi. Robot ini dapat bergerak maju, mundur, dan berbelok secara independen, sehingga memungkinkan navigasi yang akurat dan efisien di ruang yang terbatas. Namun, penelitian ini juga memiliki beberapa

kekurangan yang perlu diperhatikan. Pertama, penelitian ini lebih fokus pada pengaturan roda dan aspek kompakitas, sedangkan aspek lain seperti sistem kontrol dan navigasi tidak tercakup secara rinci. Penelitian ini juga belum melibatkan implementasi robot dalam skenario industri logistik atau pengujian lapangan yang lebih realistis, serta konfigurasi roda *omnidirectional* jarang ditemukan atau digunakan dalam industri logistik [11].

Penelitian yang dilakukan oleh M. I. Riansyah, H. Isa, D. Adiputra, L. K. Amifia, dan A. Faricha pada tahun 2021 dengan judul "**Desain dan Simulasi Sistem Kendali PID Pada AGV (*Automated Guided Vehicle*) Pengikut Garis**". Penelitian ini mengfokuskan pada perancangan dan simulasi sistem kendali PID untuk AGV (*Automated Guided Vehicle*) sebagai pengikut garis. Metode kendali PID (*Proporsional-Integral-Derivative*) diterapkan untuk mengarahkan pergerakan AGV sesuai dengan posisi garis yang diinginkan. Pada uji awal, penulis mengevaluasi respons kecepatan posisi pusat momen gambar lintasan yang diambil oleh kamera terhadap titik referensi. Hasilnya menunjukkan bahwa nilai Kp = 0,03, Kp = 0,02, dan Kp = 0,01 memberikan respons kecepatan yang hampir identik. Namun, Kp = 0,03 menimbulkan osilasi tinggi, mengindikasikan ketidakstabilan sistem. Sementara Kp = 0,02 dan Kp = 0,01 memberikan osilasi yang lebih kecil dan mencapai referensi dalam 5 ms dengan kesalahan yang minim. Uji lebih lanjut melibatkan penambahan kontrol Integral pada sistem. Hasil optimal dicapai dengan parameter PID yang tepat, yaitu Kp = 0,01, Ki = 0,015, dan Kd = 0,000148. Sistem ini menunjukkan *overshoot* sebesar 25% dan kestabilan setelah 2 ms dengan steady state *error* sebesar 0,52%. Dengan demikian, sistem kendali PID dapat digunakan secara efektif untuk mengendalikan orientasi AGV dalam mengikuti lintasan garis [12].

Berdasarkan penelitian yang ada, penggunaan roda *mecanum* dan *omni wheel* pada robot AGV untuk industri logistik memiliki keunggulan masing-masing dan berpengaruh signifikan terhadap navigasi. Namun, penggunaan roda *mecanum* cenderung lebih mudah karena konfigurasinya yang lebih sederhana dan dapat lebih mudah dipahami serta diterapkan dibandingkan dengan roda *omni*. Selain itu, penggunaan roda *omnidirectional* dalam industri logistik jarang ditemukan atau digunakan secara luas.

Oleh karena itu, dalam penelitian ini, penulis akan merancang sebuah AGV robot menggunakan roda mecanum dan modul kamera. Selain itu, penelitian ini juga akan merancang sistem navigasi dengan menerapkan metode PID untuk meningkatkan performa robot dalam menjaga stabilitas gerakan, responsivitas terhadap perubahan jalur, dan ketepatan posisi robot pada jalur yang ditentukan.

2.2 DASAR TEORI

2.2.1 AGV Robot pada Industri Logistik

Robot logistik adalah sebuah perangkat cerdas berbasis robotika yang dirancang untuk membantu distribusi pergudangan dan pengiriman barang secara efektif dan efisien. Robot logistik beroperasi menggunakan teknologi informasi dan komunikasi, seperti *Artificial Intelligence* (AI) dan pengendali PID, untuk meningkatkan kinerja operasionalnya. Robot logistik memiliki kemampuan untuk membantu manusia dalam melakukan pemindahan barang dari satu lokasi ke lokasi lain secara otomatis dan semi-otomatis dengan menggunakan informasi sensor *onboard* yang terus diperbarui untuk menghindari rintangan dan memilih jalur pergerakan terbaik. Penggunaan robot logistik memberikan keuntungan berupa efisiensi waktu, tenaga, dan sumber daya yang lebih baik dibandingkan dengan metode konvensional yang masih mengandalkan tenaga manusia [13].

AGV Robot (*Automated Guided Vehicle*) menjadi representasi nyata dari robot otonom yang diterapkan di sektor industri, khususnya dalam proses material handling. Menyuguhkan alternatif modern terhadap sistem konvensional yang sebelumnya memerlukan tangan manusia, AGV memberikan keunggulan dengan tingkat akurasi posisi yang tinggi, waktu operasional yang lebih lama, biaya operasional dan perawatan yang rendah, serta tingkat keamanan yang optimal. Keunikan AGV robot terletak pada kemampuannya beroperasi secara otonom, mampu berpindah dan menjalankan berbagai tugas tanpa campur tangan manusia. Sistem navigasinya mengandalkan pemrosesan informasi dan pengetahuan yang diperoleh melalui sensor, memandu AGV melalui langkah-langkah seperti ekstraksi informasi lingkungan, identifikasi lokasi, pergerakan menuju target, dan navigasi. [14].

Kendaraan adalah komponen sentral dalam sistem AGV karena mereka bertanggung jawab untuk melakukan tugas transportasi, dan karakteristik kendaraan dapat bervariasi tergantung pada aplikasinya. Satu cara untuk mengkategorikan AGV adalah dengan mempertimbangkan muatan yang diangkut [15].

Berikut adalah uraian ringkas mengenai jenis kendaraan AGV [15]:

1) *Forklift vehicles* (Kendaraan *forklift*): Kendaraan ini dapat membawa palet atau wadah yang kompatibel. AGV *forklift* dapat beroperasi secara otonom atau dengan adanya tempat duduk untuk pengendaraan. Keuntungan utama dari kendaraan *forklift* adalah kemampuannya untuk mengangkat atau mengantarkan beban dari lantai atau ketinggian yang berbeda, menjadikannya AGV yang sangat fleksibel.
2) *Underride vehicles* (Kendaraan *underride*): AGV *underride* terletak di bawah gerobak atau wagon dan sedikit mengangkatnya. Kendaraan ini lebih kompak dan dapat membaca transponder di bagian bawah gerobak untuk mendapatkan petunjuk tentang isi gerobak tertentu dan tujuan pengiriman. Keunggulan kendaraan *underride* meliputi kemampuan manuver yang tinggi, ruang yang lebih sedikit yang diperlukan, dan dapat berfungsi sebagai kendaraan derek.
3) *Towing vehicles* (Kendaraan derek): Sesuai dengan namanya, kendaraan derek menarik gerobak beroda, dan muatannya ditempatkan dan dihentikan secara manual atau dengan bantuan mesin otomatis lainnya. Keuntungan dari kendaraan ini adalah kemampuannya untuk membawa beberapa gerobak sekaligus.

(a) (b) (c)

Gambar 2.1 Kendaraan AGV [15]

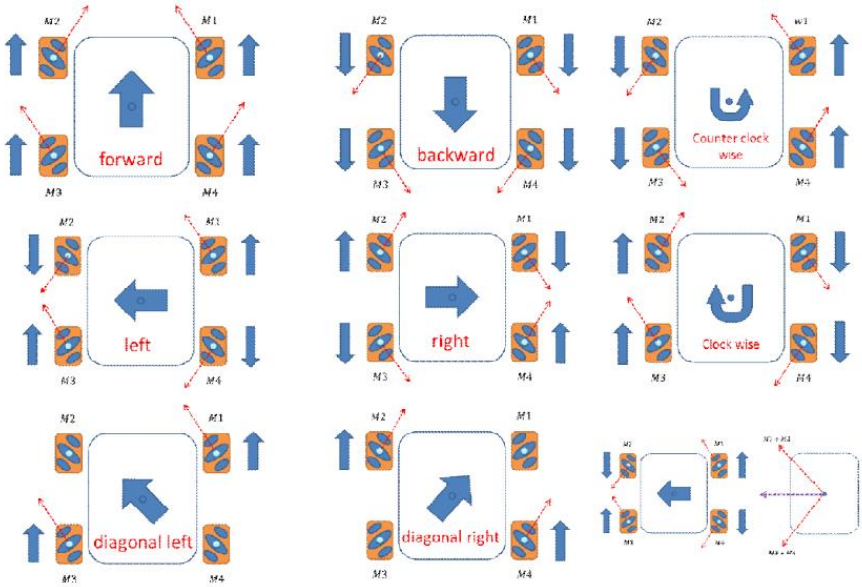
(a) *Forklift vehicles* (b) *Underride vehicles* (c) *Towing vehicles*

2.2.2 *Mecanum Wheel*

Mecanum Wheel terkateori roda yang dirancang untuk memiliki kemampuan bergerak ke segala arah.. Roda tersebut terbentuk dari hub dengan *roller* yang diposisikan pada sudut 45° terhadap sumbu rotasi. Setiap roda memiliki kemampuan putar mandiri, dan saat bergerak lateral, sepasang roda yang berlawanan akan berkolaborasi. Melalui kombinasi arah dan kecepatan gerakan pada masing-masing roda, robot dapat bergerak sesuai dengan vektor gaya total, memungkinkannya berpindah bebas tanpa mengubah orientasi roda itu sendiri [16].

Gambar 2.2 Roda *Mecanum* [17]

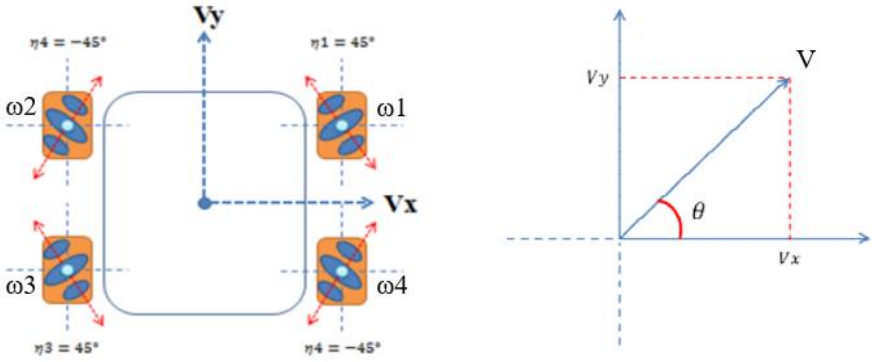
Pemilihan roda *mecanum* karena memberikan kebebasan pergerakan yang lebih besar daripada menggunakan roda konvensional seperti yang biasa digunakan dalam gerak diferensial beroda. Pergerakan robot menggunakan prinsip mecanum wheel untuk bergerak ke segala arah tanpa perlu mengubah arah menghadap terlebih dahulu. Hal ini dicapai dengan mengatur kecepatan masing-masing roda secara independen. *Mecanum Wheel* memiliki kemampuan gerak dinamis yang sangat cepat dalam setiap arah. Mereka mampu bergerak dengan cepat dan efisien, mentransfer putaran roda ke gerakan lateral. Roda *mecanum* sangat cocok untuk desain robot *indoor* dan *outdoor*, dan juga memiliki umur pakai yang lama [17]. Roda *mecanum* memiliki konfigurasi khusus yang harus diperhatikan dalam penggerakan robot. Robot yang menggunakanroda *mecanum* biasanya dilengkapi dengan empat roda *mecanum*. Setiap roda memiliki konfigurasi yang spesifik. Konfigurasi tersebut dapat dilihat pada gambar 2.3.



Gambar 2.3 Ragam Teknik Arah Putar Roda Pada *Mecanum* [17]

Roda *mecanum* menampilkan beragam kemampuan gerakan tergantung pada arah dan kombinasi gerakan setiap roda. Untuk gerakan maju, semua roda bergerak maju, sementara untuk gerakan mundur, arah gerakan roda berkebalikan dengan gerakan maju. Gerakan lateral ke kiri diimplementasikan dengan roda kiri depan dan roda kanan belakang bergerak mundur, sementara roda kanan depan dan roda kiri belakang bergerak maju. Sebaliknya, gerakan lateral ke kanan memiliki pola yang berkebalikan dengan gerakan ke kiri. Putaran berlawanan arah jarum jam dicapai dengan roda depan kiri dan roda belakang kiri bergerak mundur, sedangkan roda depan kanan dan roda belakang kanan bergerak maju. Putaran searah jarum jam memiliki pola yang berkebalikan dengan putaran berlawanan arah jarum jam. Gerakan diagonal maju ke kiri terjadi saat roda depan kanan dan roda belakang kiri diam, sementara roda depan kiri dan roda belakang kanan bergerak maju ke kiri. Pola gerakan diagonal maju ke kanan berkebalikan dengan gerakan diagonal maju ke kiri.

Dalam konfigurasi roda *mecanum*, diperlukan sebuah persamaan yang dapat mengontrol kecepatan sudut pada setiap roda agar robot *mecanum* dapat bergerak sejajar dengan sudut dan kecepatan yang diinginkan. Persamaan ini menjadi dasar pergerakan robot agar mampu bergerak ke segala arah dengan kebebasan yang lebih besar [18].



Gambar 2.4 Konfigurasi Dan Resultan Gaya Robot *Mecanum* [18]

Dalam konfigurasi desain robot *mecanum* seperti yang diperlihatkan dalam Gampar 2.4, untuk mencapai gerakan sejajar pada sudut tertentu (misalnya, satu arah) dengan kecepatan V pada sudut theta, diperlukan pengaturan kecepatan sudut ωi pada masing-masing roda (dengan i = 1...4). Kecepatan keseluruhan robot dalam sumbu x dan sumbu y dapat dihitung menggunakan metode trigonometri. Persamaan yang digunakan untuk mengestimasi kecepatan dapat diidentifikasi melalui persamaan yang tercantum di bawah ini:

$$\boldsymbol{V_x = V \cos \theta}$$
$$\boldsymbol{Vy = V \sin \theta} \qquad (2.1)$$

Dimana:

- Vx : kecepatan robot dalam sumbu x,
- Vy : kecepatan robot dalam sumbu y,
- V : kecepatan linier keseluruhan yang diinginkan,
- θ : sudut gerakan sejajar yang diinginkan.

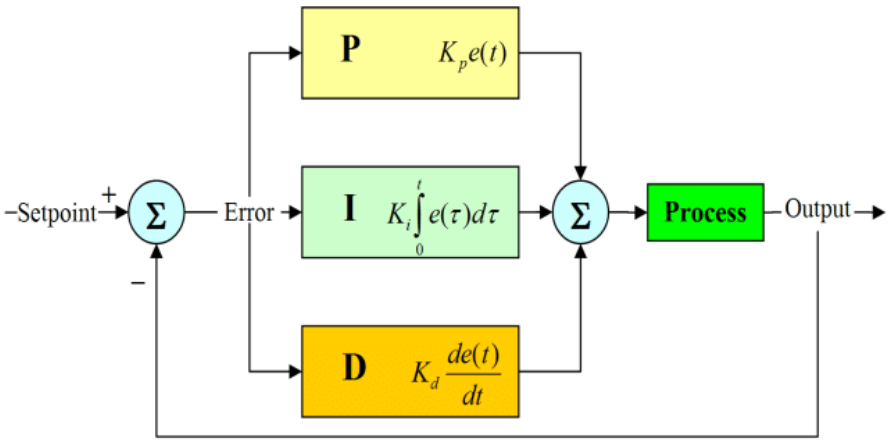
Dengan memanfaatkan persamaan di atas, kita dapat melakukan perhitungan untuk kecepatan masing-masing roda (ωi) yang diperlukan untuk mencapai gerakan sejajar pada sudut dan kecepatan yang diinginkan. Kecepatan sudut ωi pada masing-masing roda dapat dihitung menggunakan rumus:

$$\boldsymbol{w_1 = V_x - V_y} \qquad (2.2)$$

$$w_2 = V_x + V_y$$
$$w_3 = V_x - V_y$$
$$w_4 = V_x + V_y$$

2.2.3 PID (*Proportional Integral Derivative*)

Teknik kontrol PID (*Proportional-Integral-Derivative*), gabungan tiga jenis kontrol, yaitu Proporsional, Integral, dan Derivatif, sering diterapkan dalam rekayasa kontrol. Dalam penelitian sebelumnya, teknik kontrol PID telah banyak digunakan dengan berbagai metode, termasuk perancangan kendali kecepatan yang menghasilkan kendali PID sesuai dengan spesifikasi yang diinginkan dan stabil. Penggunaan mekanisme umpan balik (*feedback*) dalam kontrol PID memungkinkan koreksi kesalahan antara nilai pengukuran dan nilai kesalahan, dengan setiap jenis kontrol memiliki keunggulan masing-masing. Sebagai contoh, kontrol proporsional dapat mempercepat waktu naik (*rise time*), kontrol integral dapat mengurangi kesalahan (*error*), dan kontrol derivatif dapat mengurangi *overshoot* atau *undershoot* [19]. Diagram blok dari kontroler PID menunjukkan bagaimana sinyal kontrol dihasilkan dan digunakan untuk mengendalikan sistem instrumentasi.



Gambar 2.5 Diagram Blok Kontroler PID [20]

Dari diagram blok yang terlihat pada gambar di atas, fungsi transfer kontroler PID dapat diungkapkan melalui persamaan 2.3 [20]:

$$G_s(s) = K_p e(t) + K_i \int_0^1 e(t)\,dt + K_d \frac{de(t)}{dt} \quad (2.3)$$

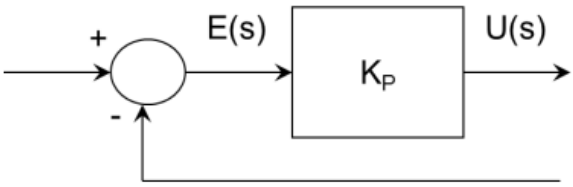
Persamaan 2.3 mengindikasikan bahwa *Output* dari Gs(s) adalah hasil penjumlahan dari kontrol proporsional (Kp), kontrol integral (Ki), dan kontrol derivatif (Kd), yang dipengaruhi oleh *error* (e) dan waktu (t) yang spesifik. Masing-masing konstanta pengendali PID memiliki kelebihan dan kekurangan dengan karakteristik sebagai berikut:

Tabel 2. 1 Karakteristik Pengendali PID [21]

Parameter	*Rise Time*	*Overshoot*	*Settling Time*	*Steady-State Error*
Proporsional	Menurunkan	Meningkatkan	Perubahan Kecil	Menurunkan/ mengurangi
Integral	Menurunkan	Meningkatkan	Meningkatkan	mengeliminasi
Derivatif	Perubahan Kecil	Menurunkan	Menurunkan	Perubahan Kecil

1) Kontrol Proporsional (Kp)

Parameter proporsional (P) merupakan elemen dalam sistem kontrol umpan balik yang menghasilkan keluaran yang sebanding dengan besarnya sinyal *error* pada sistem. Keluaran proporsional dari kontroler diperoleh melalui perkalian antara konstanta proporsional dan nilai *error* sistem. Perubahan pada sinyal input secara langsung memengaruhi keluaran dari kontroler proporsional sesuai dengan nilai konstanta pengaliannya [20]. Diagram blok kontroler proporsional menggambarkan bagaimana sinyal *error* diproses oleh kontroler proporsional untuk menghasilkan keluaran kontroler yang sesuai dengan tujuan sistem kontrol yang diinginkan.



Gambar 2.6 Blok Diagram Kd [20]

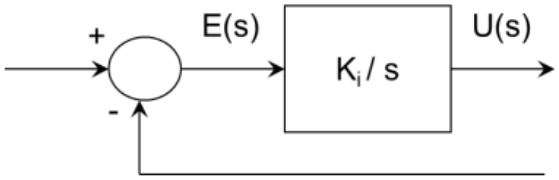
Persamaan kontroler proporsional dalam domain *Laplace* dapat dirumuskan pada persamaan 2.4 [20]:

$$\frac{\boldsymbol{U(s)}}{\boldsymbol{E(s)}} = \boldsymbol{K_p} \tag{2.4}$$

Parameter proporsional merupakan faktor kunci pada kontroler yang dapat mempercepat waktu respon sistem dan mengurangi kesalahan pada plant. Konstanta proporsional (Kp) menentukan sejauh mana *Output* kontroler akan merespons terhadap perubahan dalam *error* sistem. Semakin besar nilai Kp, semakin cepat sistem merespons terhadap perubahan, tetapi nilai yang terlalu besar dapat menyebabkan *overshoot*, di mana *Output* melebihi nilai *setpointt* yang diinginkan. Oleh karena itu, penentuan nilai Kp harus seimbang untuk mencapai kinerja optimal dalam sistem kontrol [20]. Karena itu, pemilihan nilai Kp harus dilakukan dengan cermat untuk memastikan respons sistem yang cepat dan tetap stabil.

2) Kontrol Integral (Ki)

Dalam pengendalian sistem, parameter integral digunakan untuk memperbaiki respons sistem dan memastikan *Output* sistem mencapai nilai yang diinginkan. Kontroler integral adalah jenis kontroler yang menggunakan parameter integral untuk meningkatkan *respons* sistem secara kontinyu, namun dengan batasan atas dan bawah tertentu [20]. Dengan menggunakan kontroler integral, *steady state error* dapat diminimalkan dan performa sistem dapat ditingkatkan.



Gambar 2.7 Blok Diagram Ki [20]

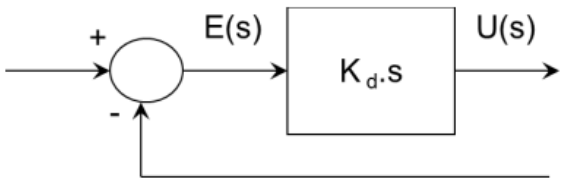
Rumus kontroler integral dalam domain Laplace dapat dijelaskan melalui persamaan 2.5 [20]:

$$\frac{\boldsymbol{U(s)}}{\boldsymbol{E(s)}} = \frac{\boldsymbol{K_i}}{\boldsymbol{s}} \tag{2.5}$$

Kontrol integral adalah suatu pengendali yang dapat mengurangi nilai *overshoot* dan meningkatkan *steady-state response* pada suatu sistem. Namun, perlu diingat bahwa penggunaan kontroler integral juga memiliki kelemahan yaitu memerlukan waktu untuk mencapai respon yang diinginkan sehingga terkesan memperlambat respon [20]. Oleh karena itu, dalam memilih jenis kontroler yang sesuai untuk suatu sistem, Perlu diperhatikan keuntungan dan kerugian dari masing-masing jenis pengendali, termasuk pengendali integral.

3) Kontrol Derivatif (Kd)

Semakin cepat terjadi perubahan pada nilai *error*, semakin besar pengaruh kontrol yang dihasilkan oleh komponen derivatif. *Respons impuls* akan terjadi saat terjadi perubahan tiba-tiba pada nilai *error* [20]. Namun, parameter derivatif tidak akan beraksi saat *error* dalam keadaan statis. Dalam praktiknya, derivatif selalu digunakan bersama dengan parameter Kp dan Ki untuk mengoptimalkan kinerja kontroler PID.



Gambar 2.8 Blok Diagram Kd [20]

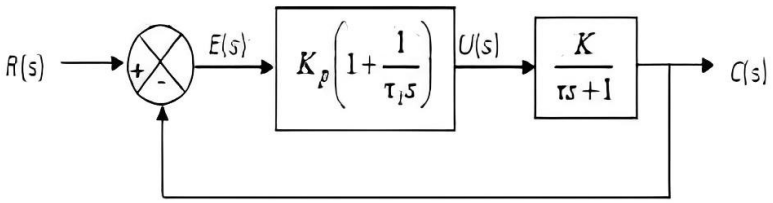
Rumus penontrol derivatif dalam domain *Laplace* dapat dinyatakan pada persamaan 2.6 [20]:

$$\boldsymbol{U(s).E(s) = K_d.s} \tag{2.6}$$

Dengan menggunakan parameter derivatif, sistem dapat menjadi lebih responsif terhadap perubahan *error*, yang pada akhirnya dapat meningkatkan kestabilan sistem. Selain itu, parameter derivatif juga dapat memberikan efek redaman pada sistem yang berosilasi, sehingga memungkinkan nilai Kp yang lebih besar untuk diberikan. Untuk mencapai nilai parameter PID (Kp, Ki, dan Kd) yang optimal, berbagai metode *tuning* PID telah diusulkan [20].

4) Kontrol PI

Aksi kontrol proporsional dan integral merupakan dua elemen kontrol yang digabungkan untuk membentuk kendali PI, penyesuaian parameter kontroler PI dapat dilakukan dengan memperhatikan keseimbangan antara *respons* sistem, *overshoot*, dan *steady state error*. Nilai Kp dapat dikurangi untuk menghindari *overshoot* yang berlebihan, sedangkan nilai Ki dapat ditingkatkan agar *steady state error* dapat diminimalkan. Proses penyesuaian parameter ini bertujuan untuk mencapai kinerja kontrol yang optimal sesuai dengan kebutuhan sistem yang dikendalikan. [20].



Gambar 2.9 Blok Diagram Kontroler PI [20]

Pengontrol PI, dapat dirumuskan pada persamaan 2.7 [20]:

$$\frac{\boldsymbol{C(s)}}{\boldsymbol{R(s)}} = \frac{\boldsymbol{K_p}(\mathbf{1} + \frac{\mathbf{1}}{\boldsymbol{\tau i s}})(\frac{\mathbf{1}}{\boldsymbol{\tau s + 1}})}{(\boldsymbol{K_p}(\mathbf{1} + \frac{\mathbf{1}}{\boldsymbol{\tau i s}})(\frac{\mathbf{1}}{\boldsymbol{\tau s + 1}})) + \mathbf{1}} \tag{2.7}$$

Keterangan :

C(s) : Keluaran

R(s) : Masukan

K : *Gain Overall*

$\boldsymbol{\tau i}$: Konstanta Waktu Tertentu

$\boldsymbol{\tau s}$: Konstanta Waktu

PI : Proporsional Integral

Pengontrol PI digunakan untuk mengatur respons sistem agar tidak mengalami *overshoot*, dengan syarat sistem tersebut harus memiliki orde satu yaitu nilai *steady-state error* (Ess) sama dengan 0% ketika sistem mencapai *set point*. Hal ini dilakukan untuk memastikan respon sistem yang stabil dan dapat diandalkan dalam penggunaannya [20].

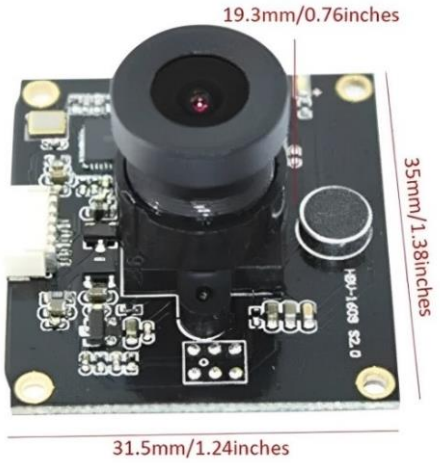
5) Kontrol PD

Kontrol PD (Proporsional-Derivatif) merupakan suatu pendekatan dalam sistem kendali yang mengintegrasikan teknik kendali proporsional (P) dengan teknik kendali derivatif (D). Gabungan ini bertujuan untuk mengatasi osilasi yang mungkin terjadi pada sistem kendali proporsional dengan menambahkan elemen pengendalian derivatif. Meskipun teknik PD efektif dalam mengurangi osilasi, namun perlu diperhatikan bahwa steady state *error* pada sistem tidak dapat sepenuhnya dihilangkan dengan pendekatan ini [20]:

$$\boldsymbol{U(t) = Kpe(t) + Kd\frac{de}{dt}} \tag{2.8}$$

2.2.4 Modul Kamera USB

HD 120° *Wide Angle Camera* adalah sebuah modul kamera yang mampu mengambil gambar dengan resolusi tinggi dan memiliki sudut pandang lebar sebesar 120 derajat. Sudut pandang yang luas ini memungkinkan kamera untuk menangkap area yang lebih besar dan lebih detail dalam satu bidikan. Modul kamera ini umumnya digunakan dalam berbagai aplikasi yang memerlukan pengambilan gambar dengan cakupan yang luas, seperti pemantauan keamanan, pemrosesan citra, pengenalan objek, dan lain sebagainya. Untuk dapat digunakan, modul kamera ini perlu dihubungkan ke perangkat lain, seperti komputer atau sistem pengolahan gambar, melalui koneksi USB. Koneksi USB ini memungkinkan transfer data gambar dari kamera ke perangkat tersebut. Setelah gambar berhasil ditransfer, perangkat dapat melakukan proses pengolahan dan analisis lebih lanjut terhadap gambar yang telah diambil oleh kamera [22].



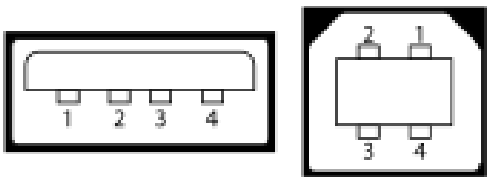
Gambar 2.10 HD 120° *Wide Angle Camera* [22]

Modul kamera USB ini dilengkapi dengan *chip* OV2643 yang kecil dan lensa sudut lebar 120° untuk memberikan jangkauan pandang yang lebih luas dan efeknya sangat bagus. Resolusi tertinggi pada modul ini adalah 1600 x 1200 dan memiliki ukuran kecil sehingga mudah dipasang. Spesifikasi lainnya dari modul ini mencakup bahan dari komponen elektronik, warna hitam, ukuran produk sekitar 35 x 31,5 x 19,3 mm, menggunakan *chip* OV2643, resolusi tertinggi 1600 x 1200 dengan 30 FPS, dan sudut pandang sebesar 120° [22].

2.2.5 Protokol USB

Protokol USB (*Universal Serial Bus*) adalah suatu protokol komunikasi yang digunakan untuk menghubungkan perangkat elektronik, berupa komputer, laptop, *printer*, *mouse*, *keyboard*, dan perangkat lainnya ke dalam jaringan atau sistem komputer. USB merupakan standar industri yang digunakan secara luas karena kecepatan transfer data yang tinggi, kemudahan penggunaan, dan kemampuan yang dapat diandalkan. Cara kerja protokol USB melibatkan beberapa komponen utama, yaitu perangkat USB, *host* USB, dan kabel USB. Perangkat USB adalah perangkat yang ingin berkomunikasi dengan host USB, seperti *printer* atau *mouse*. *Host* USB adalah perangkat yang mengendalikan komunikasi antara perangkat USB dan sistem komputer, seperti komputer atau laptop. Kabel USB berfungsi sebagai jalur fisik untuk mentransfer data dan memberikan daya listrik antara perangkat USB dan host USB [23].

Komunikasi USB dimulai dengan inisialisasi perangkat USB dan host USB yang saling mengidentifikasi dan melakukan negosiasi kemampuan. Setelah inisialisasi, perangkat USB dan host USB menggunakan protokol USB untuk mengatur komunikasi data. Komunikasi data menggunakan paket-paket yang terdiri dari data payload, header, dan token. Token digunakan untuk mengidentifikasi jenis transfer data, seperti transfer kontrol, *isochronous*, *interrupt*, atau *bulk*. Perangkat USB dan host USB bertukar paket-paket data melalui kabel USB, diatur dalam bentuk serangkaian bit sesuai dengan protokol USB. Protokol ini digunakan untuk sinkronisasi data, pengiriman ulang paket yang hilang atau rusak, serta memastikan keandalan komunikasi data antara kedua perangkat [23].



Gambar 2.11 Konektor USB tipe A dan B [24]

Berikut adalah penjelasan tentang keterangan pada penetapan kaki konektor USB:

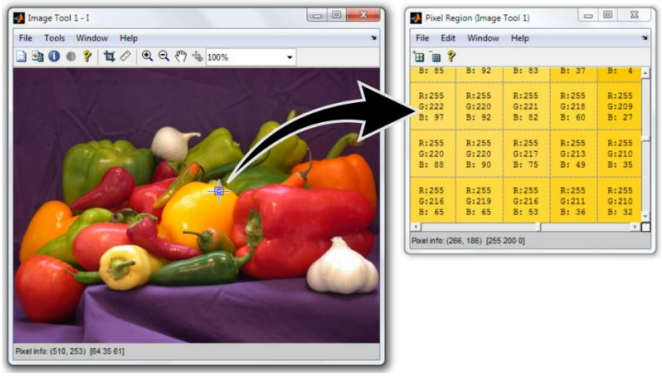
1) VBUS (Tegangan operasional 4,75 - 5,25 V): Kaki nomor 1 pada konektor USB, yang mengacu pada tegangan operasional sebesar 4,75 hingga 5,25 Volt. Tegangan ini digunakan untuk menyediakan daya listrik kepada perangkat yang terhubung melalui koneksi USB.
2) D- : Kaki nomor 2 pada konektor USB, yang merupakan jalur data negatif. Jalur ini digunakan untuk mentransmisikan sinyal data dalam arah tertentu.
3) D+ : Kaki nomor 3 pada konektor USB, yang merupakan jalur data positif. Jalur ini digunakan untuk mentransmisikan sinyal data dalam arah tertentu.
4) GND : Kaki nomor 4 pada konektor USB, yang merupakan koneksi *ground* atau tanah. Kaki ini digunakan untuk menghubungkan sinyal dan tegangan referensi ke *ground*, yang penting untuk menjaga integritas sinyal dan menjaga keselamatan.

Penetapan kaki konektor USB ini penting untuk memastikan koneksi yang benar antara perangkat dan komputer atau perangkat lainnya. Dengan mengetahui fungsi masing-masing kaki, pengguna dapat melakukan penghubungan yang tepat sesuai dengan spesifikasi yang diberikan.

2.2.6 Citra Digital

2.2.6.1 Definisi Citra Digital

Citra digital merupakan representasi data dua dimensi, di mana satuan terkecilnya disebut piksel. Setiap piksel memiliki koordinat (x, y) dan nilai intensitas I(x, y). Informasi warna dalam citra digital berasal dari intensitas cahaya yang diterima oleh sensor cahaya dalam kamera. Hasil tangkapan kamera direpresentasikan sebagai citra digital dengan format warna RGB (*Red, Green, Blue),* di mana setiap piksel memiliki komponen intensitas IR, IG, IB. Komponen intensitas ini tersusun dalam grid teratur atau array dua dimensi, seperti yang ditunjukkan pada gambar di atas [25].



Gambar 2.12 Gambar Citra dan Penyusunan Piksel [25]

2.2.6.2 Elemen Citra

2.2.6.3 Dalam pengolahan citra, fokus utama adalah pada sebuah gambar yang mencerminkan suatu objek dan mengandung sejumlah elemen dasar. Elemen-elemen tersebut, seperti warna, kecerahan, kontras, kontur, bentuk, tekstur, waktu, pergerakan, deteksi, dan pengenalan, dapat dimanipulasi selama proses pengolahan citra [26].

Berikut penjelasan mengenai masing-masing elemen citra [26]:

1) Warna: Persepsi warna adalah respons sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh suatu objek. Setiap warna memiliki panjang gelombang yang khas. Rentang warna yang paling luas dapat dihasilkan melalui kombinasi warna merah (R), hijau (G), dan biru (B).
2) Becerahan (Brightness): Kecerahan, atau yang juga dikenal sebagai intensitas cahaya, tidak mengacu pada intensitas sebenarnya pada titik piksel dalam citra. Sebaliknya, itu mencakup intensitas rata-rata dalam suatu area yang mencakup piksel tersebut.
3) Kontras (Contrast): Kontras menggambarkan distribusi antara wilayah terang dan gelap dalam sebuah gambar. Citra yang memiliki kontras rendah cenderung menunjukkan mayoritas area dengan tingkat kecerahan seragam atau mayoritas area dengan tingkat kegelapan seragam. Sebaliknya, citra dengan kontras yang baik menunjukkan distribusi yang merata antara area gelap dan terang.

4) Kontur (Contour): Kontur muncul karena perubahan intensitas di antara piksel-piksel tetangga, memungkinkan mata manusia untuk mendeteksi tepi objek dalam citra.
5) Bentuk (Shape): Bentuk merupakan karakteristik intrinsik dari objek tiga dimensi dan menjadi properti utama sistem visual manusia. Citra yang terbentuk oleh mata umumnya berupa citra dwimatra (dua dimensi), meskipun objek yang diamati biasanya memiliki bentuk trimatra (tiga dimensi). Informasi mengenai bentuk objek dapat diekstraksi dari citra pada tahap awal prapengolahan dan segmentasi citra.
6) Tekstur (Texture): Tekstur diartikan sebagai pola spasial distribusi derajat keabuan dalam kumpulan piksel yang bertetangga. Dengan demikian, tekstur tidak dapat diatributkan pada satu piksel tunggal. Sistem visual manusia menerima informasi citra sebagai suatu kesatuan, dan resolusi citra yang diamati ditentukan oleh skala di mana tekstur tersebut dapat dipersepsi.
7) Waktu dan Pergerakan: Respon sistem visual tidak hanya dipengaruhi oleh faktor ruang, tetapi juga oleh faktor waktu dan pergerakan. Sebagai contoh, penyajian citra yang bergerak dengan cepat dapat menciptakan kesan bahwa citra tersebut sedang bergerak.
8) Deteksi dan Pengenalan: Dalam proses deteksi dan pengenalan citra, tidak hanya keterlibatan sistem visual manusia yang terlibat, tetapi juga melibatkan fungsi ingatan dan daya pikir manusia.

2.2.6.4 Jenis-Jenis Citra

Dalam tahapan deteksi dan pengenalan citra, keterlibatan sistem visual manusia tidaklah cukup, melainkan juga melibatkan fungsi ingatan dan daya pikir manusia:

1) Citra RGB

Citra RGB (*Red, Green, Blue*) adalah jenis citra yang terdiri dari tiga komponen warna utama yaitu merah, hijau, dan biru yang digunakan pada citra digital. Pada citra ini, setiap piksel memiliki nilai intensitas warna untuk ketiga komponen tersebut. Dengan menggabungkan intensitas warna ketiga

komponen, citra RGB dapat menghasilkan berbagai warna yang terlihat oleh mata manusia. Citra RGB umum digunakan dalam tampilan visual dan pemrosesan gambar umum seperti foto dan video.

Konsep dasar RGB bersumber dari respons mata manusia terhadap panjang gelombang tertentu, seperti 630 nm untuk warna merah, 530 nm untuk hijau, dan 450 nm untuk biru. Sebagai contoh, suatu piksel dengan nilai intensitas warna sebesar 255 pada kanal merah, 255 pada kanal hijau, dan 0 pada kanal biru akan menghasilkan warna kuning. Pada citra RGB truecolor 24-bit, kombinasi warna yang berbeda pada tiga kanal dapat menghasilkan hingga 16.777.216 warna piksel yang berbeda [27].

Gambar 2.13 Representasi Nilai Intensitas Piksel dengan Kombinasi Warna RGB [27]

2) Citra *Grayscale*

Citra *grayscale* merupakan tipe citra digital di mana setiap pikselnya memiliki nilai yang identik untuk komponen *Red*, *Green*, dan *Blue*. Nilai-nilai ini mencerminkan intensitas warna yang menggambarkan berbagai bagian dalam gambar. Hasilnya adalah citra berwarna abu-abu dengan variasi intensitas dari hitam hingga putih. Perbedaan antara citra *grayscale* dan citra hitam-putih terletak pada variasi warna yang ada di antara hitam dan putih. Citra *Grayscale* memiliki skala warna yang lebih terbatas dibanding citra RGB, namun sering digunakan dalam pemrosesan gambar yang membutuhkan analisis intensitas keabuan seperti deteksi tepi, pengenalan pola, dan pengolahan citra medis.

Citra *grayscale* terbentuk melalui perhitungan intensitas cahaya pada setiap piksel dalam satu band spektrum elektromagnetik. Biasanya, citra grayscale disimpan dalam format 8 bit untuk setiap sampel piksel, memungkinkan adanya 256 intensitas yang berbeda [27]. Persamaan yang digunakan untuk mengkonversi citra RGB *true color* 24-bit menjadi citra *grayscale* 8-bit dapat dijelaskan pada persamaan 2.9 berikut [27]:

$$Grayscale = (0.2989 * R) + (0.5870 * G) + (0.1140 * B) \quad (2.9)$$

Keterangan:

Grayscale : intensitas citra *grayscale*

R : intensitas piksel pada saluran warna merah

G : intensitas piksel pada saluran warna hijau

B : intensitas piksel pada saluran warna biru

Gambar 2. 14 Citra Hasil Konversi RGB menjadi *Grayscale* [27]

3) Citra Biner

Citra biner adalah jenis citra yang hanya memiliki dua nilai piksel, yakni hitam (0) dan putih (1). Umumnya, nilai 0 mewakili piksel yang gelap atau tidak aktif, sedangkan nilai 1 mewakili piksel yang terang atau aktif. Setiap piksel pada citra biner hanya memerlukan 1 bit untuk mewakili nilai, sehingga citra biner memerlukan ruang penyimpanan yang lebih sedikit dibandingkan dengan citra berwarna atau *grayscale*. Citra biner sering dihasilkan melalui proses pengubahan citra grayscale dengan menggunakan metode *thresholding*,

di mana ambang batas tertentu digunakan untuk memisahkan piksel yang terang dan gelap. Citra biner umum digunakan dalam pemrosesan citra yang melibatkan deteksi objek, segmentasi, dan analisis bentuk.

Citra biner dapat dibentuk dengan menggunakan nilai batas keabuan sebagai nilai patokan untuk membedakan piksel hitam dan putih. Proses binerisasi sering digunakan dalam berbagai macam pengolahan citra, seperti segmentasi dan pengenalan pola, karena mempermudah deteksi pola pada citra yang memiliki sedikit warna. Salah satu metode untuk mengubah citra grayscale menjadi citra biner adalah melalui proses *thresholding*, di mana nilai intensitas piksel yang lebih besar atau sama dengan nilai *threshold* akan diubah menjadi 1, sedangkan yang lebih kecil menjadi 0 [27].

Gambar 2.15 Citra Hasil Konversi *Grayscale* menjadi Biner [27]

2.2.6.1 Pengolahan Citra

Pengolahan citra digital dilakukan bertujuan untuk mengubah atau meningkatkan kualitas citra agar lebih mudah dalam melakukan pendeteksian objek yang menjadi fokus dan ekstraksi fitur-fiturnya. Hal ini bertujuan untuk membantu manusia dalam memahami citra tersebut dan mendapatkan informasi yang lebih berguna dari citra tersebut. Dengan melakukan pengolahan citra digital untuk memperbaiki citra, menemukan objek yang menjadi interest, dan mengekstrak informasi yang berguna dari citra tersebut [27].

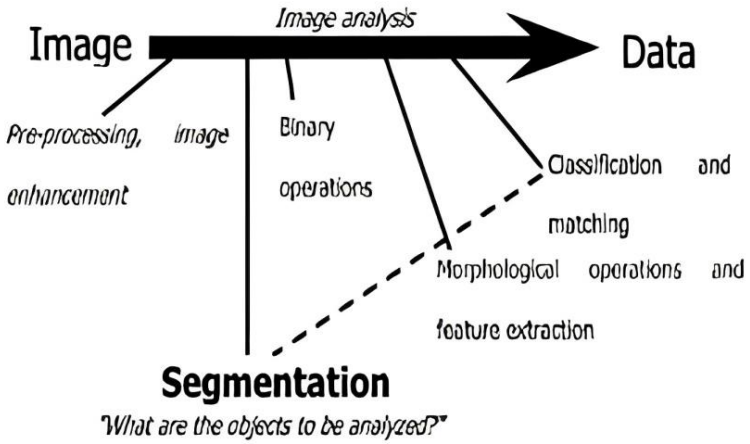
Dalam pengolahan citra terdapat 3 langkah utama yaitu [27]:

1) Konversi Warna

Konversi warna bertujuan untuk menampilkan citra dalam komponen lain agar mudah dideteksi. Pada tahap ini, citra yang awalnya direpresentasikan dalam satu format warna, seperti RGB (*Red, Green, Blue*), dapat diubah menjadi format warna lainnya, seperti *Grayscale* atau HSV (*Hue, Saturation, Value*). Konversi warna ini penting untuk mempermudah analisis dan manipulasi citra lebih lanjut.

2) Segmentasi

Segmentasi bertujuan untuk mendapatkan lokasi, area atau tepian objek yang dicari atau akan dianalisis. Segmentasi citra merupakan proses yang memisahkan objek satu sama lain atau objek dari latar belakang dalam sebuah citra. Proses ini memungkinkan pengambilan objek secara individu untuk dapat digunakan dalam analisis dan interpretasi citra. Terdapat dua jenis segmentasi, yaitu *full segmentation* dan *partial segmentation*. *Full segmentation* memisahkan setiap objek secara individu dari latar belakang dan memberikan label pada setiap segmen, sedangkan *partial segmentation* hanya memisahkan sebagian data dari latar belakang, menyimpan data yang terpisah untuk mempercepat proses selanjutnya. Teknik segmentasi, seperti *thresholding*, *region growing*, atau pemisahan warna, dapat digunakan untuk mencapai tujuan ini. Segmentasi bertujuan mengidentifikasi dan memisahkan objek tertentu agar dapat dianalisis atau diolah lebih lanjut [28].



Gambar 2.16 Proses Pengolahan Segmentasi Citra [28]

Thresholding merupakan metode segmentasi citra yang memisahkan objek dari latar belakang dengan memperhatikan perbedaan tingkat kecerahan atau kegelapan. Bagian citra yang cenderung gelap akan diubah menjadi semakin gelap dengan nilai intensitas 0, sementara bagian yang cenderung terang akan diubah menjadi semakin terang dengan nilai intensitas 1. Hal ini menghasilkan citra biner dengan nilai intensitas piksel hanya berupa 0 atau 1. Setelah proses segmentasi ini, citra biner dapat digunakan sebagai mask untuk melakukan proses *cropping*, memperoleh tampilan citra asli tanpa latar belakang atau dengan latar belakang yang dapat diubah-ubah sesuai kebutuhan [29].

Metode ini menggunakan nilai ambang T sebagai acuan untuk menentukan apakah sebuah piksel harus diubah menjadi hitam atau putih. Nilai ambang T dihitung dengan persamaan $T = (f_{maks} + f_{min})/2$, di mana f_{maks} adalah nilai intensitas maksimum pada citra dan f_{min} adalah intensitas minimum pada citra [28]. Untuk setiap piksel pada posisi (x,y) dengan nilai intensitas f(x,y), piksel tersebut akan diubah menjadi putih atau hitam tergantung pada kondisi tertentu seperti pada persamaan 2.10 dan persamaan 2.11 [28]:

$$\boldsymbol{F(x,y) = 255, jika\ f(x,y) \geq = T} \quad (2.10)$$

$$\boldsymbol{F(x,y) = 0, jika\ f(x,y) < T} \quad (2.11)$$

Ketenrangan :

$\boldsymbol{F(x,y)}$: Citra hasil *Threshold*

$\boldsymbol{T}$: Nilai Pemetaan Piksel

Sebagai contoh, kita dapat mengambil citra *grayscale* berukuran 4 x 4 piksel dengan kedalaman 8 bit sebagai ilustrasi, seperti yang tertera pada gambar di bawah ini [28]:

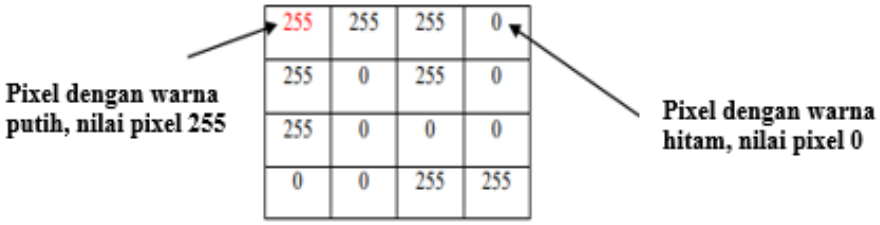
200	230	150	75
240	50	170	90
210	100	120	80
100	90	200	230

Gambar 2.17 Nilai Piksel Citra *Grayscale* 4x4 [28]

Perhitungan nilai *threshold* T adalah :

$$\boldsymbol{T = \frac{(fmax+fmin)}{2} = \frac{(240+50)}{2} = 145}$$

Jika nilai T adalah 145, maka saat diterapkan pada citra, akan menghasilkan *Output* seperti Gambar berikut:



Gambar 2.18 Nilai Piksel Citra setelah *Thresholding* [28]

3) Filter

Filter digunakan untuk memperbaiki kualitas citra atau menghilangkan noise yang tidak diinginkan. Filter dapat berupa filter spasial seperti filter mean atau filter median yang digunakan untuk menghaluskan citra, atau filter frekuensi seperti filter *high-pass* atau *low-pass* yang digunakan untuk memperbaiki kejelasan atau menghilangkan gangguan pada citra. Penggunaan filter dalam pengolahan citra membantu meningkatkan kualitas dan kejelasan objek yang terdapat dalam citra.

2.2.7 *Color Space*

Color space sebenarnya adalah suatu organisasi khusus dari warna yang, bersamaan dengan profil perangkat fisik, memungkinkan representasi warna yang dapat direproduksi baik dalam representasi digital maupun analog. Ini merupakan

alat khusus untuk memahami kemampuan warna dari suatu perangkat tertentu atau berkas digital. Ketika kita mencoba mereproduksi warna pada perangkat lain, *color space* inilah yang dapat menunjukkan apakah kita akan dapat mempertahankan detail bayangan atau sorot, saturasi warna, dan sejauh mana keduanya dapat dikompromikan [30]. Umumnya, warna dapat diukur melalui atribut berikut [30]:

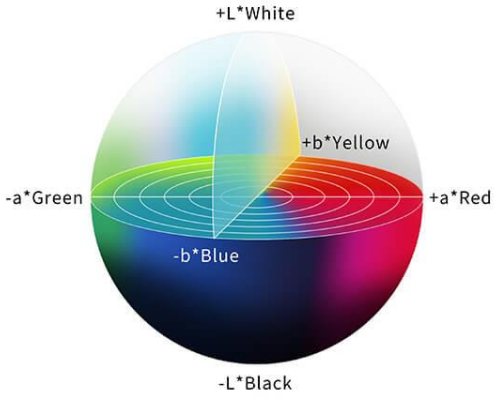
a. *Brightness*: Sensasi manusia terhadap seberapa banyak atau sedikit cahaya yang dimiliki suatu area.
b. *Hue*: Sensasi manusia yang menyatakan sejauh mana suatu area tampak mirip dengan satu atau dua proporsi warna yang terlihat seperti merah, kuning, hijau, dan biru.
c. *Colorfulness*: Sensasi manusia yang menyatakan sejauh mana suatu area tampak memiliki lebih atau kurang dari warna dasarnya.
d. *Lightness*: Sensasi kecerahan suatu area relatif terhadap warna putih referensi dalam adegan.
e. *Chroma*: Kewarnaan suatu area relatif terhadap kecerahan warna putih referensi.
f. *Saturation*: Kewarnaan suatu area relatif terhadap kecerahan.

Meskipun RGB (*Red, Green, Blue*) adalah yang paling umum digunakan, terdapat berbagai *color space* yang menyajikan informasi warna dengan cara yang berbeda. Ini membuat perhitungan tertentu lebih nyaman dan memberikan cara yang lebih intuitif untuk mengidentifikasi warna. RGB termasuk dalam kategori *color space* yang tergantung pada perangkat, sedangkan ada yang termasuk dalam kategori *color space* yang tidak tergantung pada perangkat. Sebagai contoh, LAB* adalah salah satu *color space* yang tidak tergantung pada perangkat. Karena fokus makalah ini adalah pada *color space* LAB* dan HSV, mari kita bahas keduanya [30].

2.2.7.1 LAB *Color Space*

LAB *Color space* awalnya didefinisikan oleh CAE dan ditentukan oleh Komisi Internasional untuk Pencahayaan. Dalam *color space* ini, kita memiliki satu saluran untuk Luminans (Kebercahayaan) dan dua saluran warna lainnya, yaitu a

dan b, yang menggambarkan kromatisitas. Saluran a menunjukkan di mana warna berada sepanjang sumbu merah-hijau, sementara saluran b menunjukkan di mana warna berada sepanjang sumbu biru-kuning. Warna hijau direpresentasikan oleh nilai negatif pada saluran a, sementara warna magenta direpresentasikan oleh nilai positif. Selain itu, warna biru direpresentasikan oleh nilai negatif pada saluran b, dan warna kuning direpresentasikan oleh nilai positif. Salah satu fitur penting dari *color space* ini adalah bahwa ia tidak tergantung pada perangkat, yang berarti dapat digunakan untuk berkomunikasi dengan warna yang sama di berbagai perangkat. Diagram koordinat dalam LAB *color space* dengan jelas menjelaskan bagaimana kita mengartikan warna dalam kerangka ini [30].

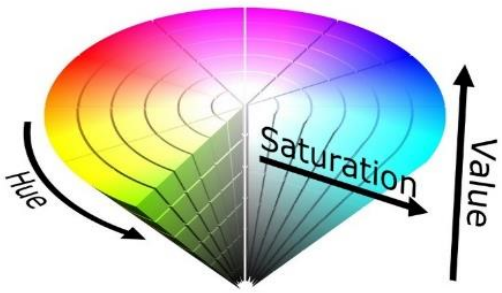


Gambar 2.19 Diagram LAB *Space Color* [30]

Dalam gambar di atas, sumbu vertikal sentral mewakili kecerahan (L*). L* dapat memiliki nilai dari 0 (hitam) hingga 100 (putih). Sumbu koordinat mengikuti kenyataan bahwa "suatu warna tidak dapat menjadi merah dan hijau, atau biru dan kuning, karena warna-warna ini saling bertentangan". Nilainya berjalan dari positif ke negatif untuk setiap sumbu. Nilai positif 'a' menunjukkan jumlah merah, sedangkan nilai negatif menunjukkan jumlah hijau. Nilai positif 'b' menunjukkan jumlah kuning, sedangkan nilai negatif 'b' menunjukkan jumlah biru. Nol mewakili abu-abu netral untuk kedua sumbu. Karena itu, nilai diperlukan hanya untuk dua sumbu warna dan untuk sumbu kecerahan atau skala abu-abu (L*) [30].

2.2.7.2 HSV *Color Space*

HSV *Color space* dapat direpresentasikan dalam bentuk heksakon tiga dimensi, dengan sumbu vertikal mewakili intensitas. Dalam *color space* ini, 'H' (*Hue*) adalah sudut dalam rentang [0,2π] relatif terhadap sumbu merah, dengan merah pada sudut 0, hijau pada 2π/3, biru pada 4π/3, dan merah lagi pada 2π. 'S' (*Saturation*) menggambarkan seberapa murni nuansa terhadap referensi putih dan diukur sebagai jarak radial dari sumbu sentral, dengan nilai antara 0 di tengah hingga 1 di permukaan luar. Ketika S=0, perpindahan ke atas sumbu intensitas menghasilkan perubahan dari hitam ke putih melalui berbagai nuansa abu-abu. Sementara itu, perubahan saturasi dari 0 hingga 1, pada intensitas dan hue yang sama, menghasilkan perubahan warna dari nuansa abu-abu menjadi bentuk paling murni dari warna yang direpresentasikan oleh hue-nya. 'V' (*Value*) adalah persentase intensitas cahaya, berkisar dari 0 hingga 100. Sebagai contoh, pada hue merah dan nilai tinggi, warna terlihat cerah, sementara pada nilai rendah, warna terlihat gelap [30].

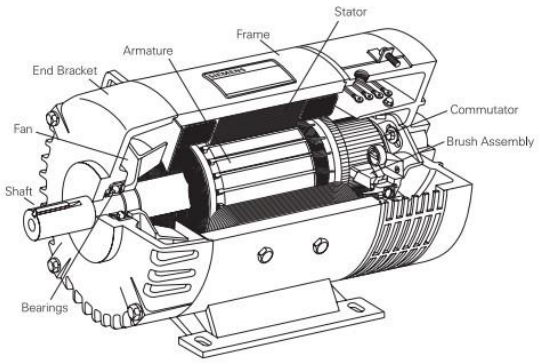


Gambar 2.20 Diagram HSV *Color Space* [30]

2.2.8 Motor DC

Motor DC adalah alat yang mengubah energi listrik menjadi energi gerak dengan menggunakan input tegangan searah. Prinsip kerjanya adalah ketika arus mengalir melalui konduktor, akan terbentuk medan magnet di sekitar konduktor tersebut. Arah medan magnet ditentukan oleh arah aliran arus pada konduktor. Medan magnet yang terbentuk di daerah kumparan medan dapat menyebabkan perubahan energi listrik menjadi gerak dalam medan magnet. Proses ini terjadi

ketika tegangan dari sumber lebih besar daripada gerak yang disebabkan oleh beban. Motor DC memiliki tiga kategori tergantung pada kecepatan yang diperlukan untuk keluaran tenaga putarnya: beban torsi konstan, beban dengan torsi variabel, dan beban dengan energi konstan [31].



Gambar 2.21 Motor DC [31]

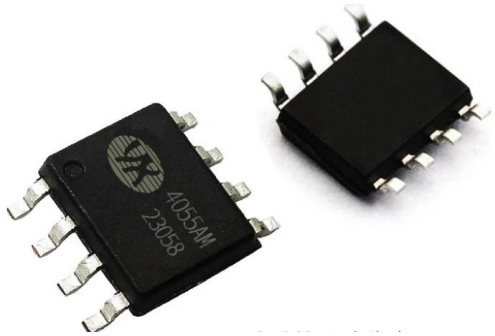
Gambar diatas menunjukkan motor DC dengan sistem umpan balik tertutup. Pada sistem ini, posisi rotor motor DC akan dikembalikan ke rangkaian kontrol yang ada di dalam motor. Motor servo terdiri dari rangkaian kontrol, *Driver*, motor DC, rangkaian *gear*, dan potensiometer. Motor DC dikendalikan oleh kontrol dan potensiometer yang terhubung ke gear.

2.2.9 *Driver* Motor DC

Driver motor adalah perangkat elektronik yang memiliki peran penting dalam mengontrol daya dan arus yang mengalir ke motor listrik. Fungsinya yang utama adalah mengatur putaran, kecepatan, arah pergerakan, dan pengendalian motor secara presisi sesuai dengan sinyal kendali yang diberikan. *Driver* motor biasanya digunakan dalam sistem kontrol motor yang kompleks, seperti pada robotika, automasi industri, kendaraan listrik, dan lain sebagainya.

Pada penelitian ini menggunakan *Driver* motor yang berjenis YX-4055AM. *Driver* motor ini memiliki spesifikasi yang mencakup tegangan kerja yang dapat disesuaikan antara 3V hingga 15V DC, dengan arus maksimum yang bisa mencapai 4A hingga 5.5A. Selain itu, *Driver* ini juga dilengkapi dengan fitur pengaturan kecepatan yang presisi, pengaturan arah putaran (*forward* dan *reverse*),

dan perlindungan terhadap tegangan tinggi, arus berlebih, serta perlindungan termal. *Driver* motor YX-4055AM dapat dikendalikan menggunakan sinyal PWM atau sinyal analog, dan kompatibel dengan berbagai jenis motor DC seperti motor brushed dan brushless. Selain itu, *Driver* ini juga mendukung fitur umpan balik seperti enkoder untuk memantau posisi atau kecepatan motor dengan akurasi [32].



Gambar 2.22 *Driver* Motor YX-4055AM [32]

2.2.10 *Raspberry Pi*

Raspberry Pi merupakan sebuah komputer papan tunggal (*Single-Board Computer*) dengan ukuran seukuran kartu kredit. *Raspberry Pi* dilengkapi dengan semua fungsi yang ditemukan dalam komputer lengkap, menggunakan *System-on-a-Chip* (SoC) berbasis ARM yang terintegrasi dalam PCB (papan sirkuit). *Raspberry Pi* mampu menjalankan sistem operasi *Linux* dan berbagai aplikasi seperti *LibreOffice*, multimedia (audio dan video), peramban *web*, dan pemrograman [34].

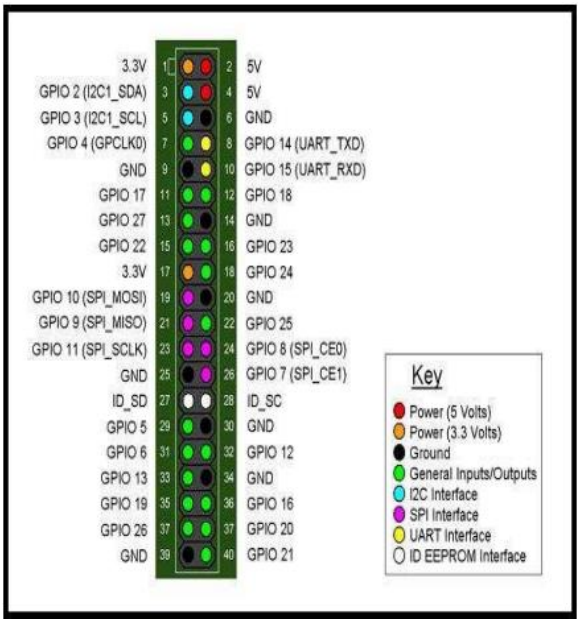
Gambar 2.23 *Board Raspberry Pi* [34]

2.2.11.1 Sistem Operasi *Raspbian*

Raspbian adalah sistem operasi berbasis *Linux* yang dikembangkan khusus untuk digunakan pada komputer mini *Raspberry Pi*. Sistem operasi ini didasarkan pada distribusi Debian *Linux* dan telah dioptimalkan untuk kompatibilitas dan kinerja maksimal dengan *Raspberry Pi*. Raspbian dilengkapi dengan berbagai program standar dan utilitas yang memungkinkan penggunaan lengkap dari perangkat keras *Raspberry Pi*. Dengan lebih dari 350.000 paket dan pustaka *precompiled* yang tersedia, *Raspbian* menyediakan beragam opsi untuk instalasi dan pengembangan pada *Raspberry Pi* [34].

2.2.11.2 *Raspberry Pi* GPIO Pin

Pin GPIO (*General Purpose Input/Output*) pada *Raspberry Pi* adalah antarmuka fisik yang memungkinkan koneksi dan integrasi dengan perangkat lain. Pada tingkat yang paling dasar, pin GPIO dapat digunakan sebagai saklar yang dapat diaktifkan atau dinonaktifkan (input) atau sebagai pengendali untuk menghidupkan atau mematikan perangkat lain (*Output*). *Raspberry Pi* memiliki 40 pin, di mana 26 pin di antaranya adalah pin GPIO, sedangkan yang lainnya digunakan untuk daya atau *ground* [34].



Gambar 2.24 *Raspberry Pi* GPIO Pin [34]