

PENERAPAN DETEKSI GARIS PADA AGV MENGGUNAKAN METODE HSV

Yulius Dani Saputra*) dan Florentinus Budi Setiawan

¹Program Studi Teknik Elektro, Fakultas Teknik, Universitas Katolik Soegijapranata
Jl. Pawiyatan Luhur IV/1, Semarang 50234, Indonesia

*)E-mail: danisap755@gmail.com

Abstrak

Dalam perkembangan teknologi saat ini, khususnya di sektor industri, robot dengan kecerdasan buatan mengemban peran penting dalam meningkatkan efisiensi waktu kerja manusia. Penelitian ini menciptakan sebuah robot AGV yang dilengkapi sistem kecerdasan buatan, menggunakan kamera sebagai sensor visual untuk mendeteksi garis lintasan sebagai panduan navigasi. Robot AGV ini memanfaatkan OpenCV untuk pemrosesan citra dengan metode filter warna HSV. Metode ini mencakup teknik morfologi dan Gaussian blur untuk mengenali garis lintasan yang akan dilalui. Setelah proses identifikasi jalur, robot AGV akan bergerak sesuai jalur yang terdeteksi oleh kamera. Pengujian perangkat keras dilakukan di laboratorium, bahwa mode navigasi robot AGV berdasarkan metode HSV mampu berfungsi baik dan menghasilkan tingkat akurasi deteksi jalur yang tinggi.

Kata kunci: AGV, Open CV, Object Detection, HSV, Raspberry Pi

Abstract

In today's technological development, especially in the industrial sector, robots with artificial intelligence play an important role in improving the efficiency of human work time. This research creates an AGV robot equipped with an artificial intelligence system, using a camera as a visual sensor to detect track lines as a navigation guide. This AGV robot utilizes OpenCV for image processing with the HSV color filter method. This method includes morphology and Gaussian blur techniques to recognize the path line. After the path identification process, the AGV robot will move according to the path detected by the camera. Hardware testing was conducted in the laboratory, that the navigation mode of the AGV robot based on the HSV method was able to function properly and produce a high level of path detection accuracy.

Keywords: AGV, Open CV, Object Detection, HSV, Raspberry Pi

1. Pendahuluan

Dalam era kemajuan teknologi saat ini, banyak perusahaan tengah bersaing untuk menciptakan robot yang dapat membantu pekerjaan manusia, khususnya dalam upaya meningkatkan efisiensi waktu dan tenaga pada proses produksi [1], [2]. Salah satu contoh robot yang sedang dalam tahap pengembangan di sektor industri adalah Automated Guided Vehicle (AGV) atau kendaraan kendali otomatis. AGV (Automated Guided Vehicle) telah banyak digunakan sebagai untuk mengenali kendaraan yang memiliki kemampuan bergerak dan melaksanakan tugas-tugas tertentu secara mandiri tanpa memerlukan bantuan dari operator [3], [4]. Jenis AGV yang banyak diterapkan adalah AGV line follower dan banyak dunia industri menggunakan sistem AGV dengan bantuan jalur berupa garis warna, jalur magnet, atau laser [5][6][7]. Dengan sistem yang bergerak dari satu lokasi ke lokasi lainnya dengan bantuan panduan jalur yang terletak sepanjang rute AGV. Model navigasi menggunakan

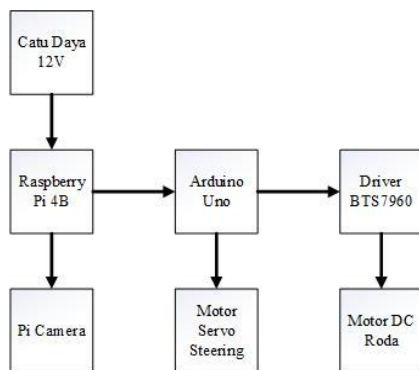
panduan jalur ini umumnya digunakan untuk sistem AGV yang sederhana [8].

Terdapat sistem navigasi AGV yang mendeteksi jalur menggunakan sensor garis berupa photodiode yang tersusun dari sejumlah LED. Dengan metode pembacaan jalur menggunakan sensor garis pada AGV dapat dipastikan jika ada salah satu LED yang mati maka akan terjadi error pada pembacaannya [9]. Dalam perkembangan metode CNN (Convolutional Neural Network) memiliki metode yang dilakukan dengan percobaan berdasarkan banyaknya dataset. Dalam metode CNN (Convolutional Neural Network) ini dipastikan membutuhkan banyak percobaan dataset sesuai yang diinginkan maka dari itu untuk meningkatkan efisiensi dalam merancang robot AGV yang dilengkapi kamera menggunakan metode HSV. Yang tidak membutuhkan banyak dataset dalam tahap uji coba robot AGV [10], [11]. Penelitian sebelumnya dalam menerapkan sistem AGV masih menggunakan Arduino dan belum dilengkapi

proses pengolahan citra gambar. Yang diketahui penggunaan arduino masih kurang dalam proses pengolahan citra gambar [9]. Permasalahan ini menjadi penghambat yang signifikan dalam hal waktu dan biaya jika diimplementasikan dalam dunia industri. Oleh karena itu, tujuan dari penelitian ini untuk mengembangkan AGV dengan memanfaatkan Raspberry Pi 4, yang memiliki kemampuan dalam memproses gambar secara real-time dengan cepat dan akurat. Karena itu, penerapan pengolahan citra dapat dijalankan pada sistem AGV sehingga dapat beroperasi dengan [12].

Pada umumnya penelitian sistem robot AGV yang diciptakan banyak yang masih menggunakan sensor infrared. Maka dari itu penelitian ini bertujuan untuk membahas mengenai sistem robot AGV yang dilengkapi dengan proses image processing. Dalam proses image processing bisa mengolah data berupa object detection [13]. Yang bisa mendeteksi wajah, warna, dan bentuk gambar [14]. Dengan sistem penggerak 4WD pada AGV dan dilengkapi camera untuk proses image processing. Pada sistem AGV ini menggunakan Arduino untuk pengaturan kecepatan motor DC dan mengatur gerak motor servo [9]. Untuk proses image processing menggunakan Raspberry Pi 4B. Komponen kamera berperan penting untuk mengambil gambar berupa video menggunakan Raspberry Pi Camera V2, sehingga penerapannya menjadi lebih mudah [15].

2. Metode



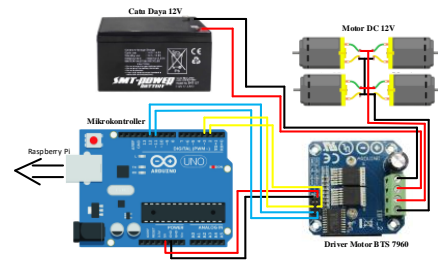
Gambar 1. Diagram Blok Sistem AGV

Pada Gambar 1 merupakan diagram blok sistem AGV. Dengan alur kinerja sistem secara berurutan, dari catu daya untuk menghidupkan Raspberry Pi yang dilengkapi dengan Pi Camera untuk proses pengolahan citra gambar atau image processing. Raspberry Pi juga dimanfaatkan sebagai catu daya untuk Arduino Uno yang sekaligus berfungsi sebagai media programming untuk Arduino Uno. Arduino Uno berfungsi untuk menjalankan motor servo sebagai steering dan motor DC untuk berjalan maju secara 4WD. Untuk menjalankan 4 motor DC yang

diparalel dibutuhkan driver motor BTS7960 yang berfungsi sebagai mengatur kecepatan motor DC.

Pada bagian metode penelitian ini terdapat beberapa tahapan, yaitu meliputi perancangan sistem penggerak AGV, pola jalur AGV, dan proses image processing.

2.1. Sistem Penggerak AGV



Gambar 2. Wiring Diagram Motor

Pada Gambar 2 menunjukkan sistem penggerak AGV ini menggunakan 4 motor DC 12V sebagai penggerak utama untuk berjalan maju. Pada AGV ini, pengaturan sistem steering mengandalkan motor servo yang berfungsi untuk melakukan perputaran ke kanan dan ke kiri. Motor DC yang diaplikasikan diberikan tegangan sebesar 12V yang berasal dari sumber daya aki, lalu motor servo diberi tegangan 5V dari pin 5V Arduino Uno. Untuk mengatur kecepatan motor DC menggunakan driver BTS7960 berdasarkan nilai PWM yang diatur [16]. Komponen utama pada sistem penggerak AGV ini adalah Arduino Uno [17], untuk mendukung kinerja dari driver BTS7960. Driver BTS7960 beroperasi dengan pengaturan nilai PWM maka menghasilkan kecepatan motor DC yang diinginkan. Sistem dari AGV ini akan berjalan sesuai jalur yang ditentukan tanpa kemudi. Motor DC berjalan dengan kecepatan stabil sesuai nilai PWM yang ditentukan. Jalur yang ditentukan untuk sistem AGV ini memanfaatkan Arduino Uno sebagai fungsi timer. Timer berfungsi untuk membatasi putaran roda pada saat AGV berjalan sesuai jalur. Jika timer yang diatur tidak sesuai dengan jalur yang ditentukan maka AGV akan berjalan tidak mengikuti jalur. Dari pembahasan diatas didapat persamaan untuk hasil perhitungan dari nilai PWM yang ditentukan.

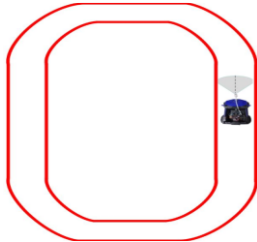
$$Duty Cycle = \frac{t_{ON}}{(t_{ON} + t_{OFF})} \quad (1)$$

$$V_{out} = Duty Cycle \cdot V_{in} \quad (2)$$

2.2. Pola Jalur AGV

Pola jalur yang terbentuk pada robot AGV ini ditunjukkan pada Gambar 3. Untuk membentuk pola jalur tersebut

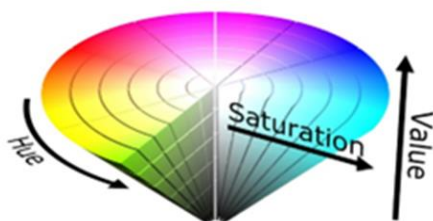
gerak motor DC dan motor servo yang digunakan diatur berdasarkan fungsi timer. Timer diatur sesuai panjangnya lintasan sehingga AGV berjalan sesuai jalur. Dalam menentukan program timer untuk sistem AGV ini harus tepat sehingga didapatkan hasil yang diinginkan sesuai jalur lintasan yang sudah ditentukan sebelumnya.



Gambar 3. Pola Jalur AGV

2.3. Proses Image Processing

Dalam proses image processing komponen yang digunakan adalah kamera Raspberry Pi V2. Kamera tersebut untuk menangkap gambar secara real time berupa video. Dalam proses penangkapan gambar berupa video ini digunakan hardware utama yaitu Raspberry Pi 4B. Komponen tersebut sudah mempunyai teknologi yang mampu untuk melakukan image processing. Kamera tersebut digunakan untuk mendeteksi objek berupa garis menggunakan metode HSV. Dalam mendeteksi objek tersebut melalui beberapa tahapan untuk mendapatkan hasil berupa color filter HSV. Metode HSV ini digunakan untuk menentukan warna garis lintasan yang digunakan. Filter HSV digunakan untuk mendeteksi perbedaan warna garis lintasan dan warna permukaan. Untuk mendapatkan hasil color filter HSV ini melalui beberapa tahapan. Model warna HSV mengandung komponen dengan fungsi yang berbeda untuk mengukur warna. Hue mengindikasikan panjang gelombang warna utama dan memiliki rentang antara 0-255 [18]. Saturasi merujuk pada intensitas kecerahan warna, yang ditingkatkan berdasarkan nilai saturasi murni dalam warna akhir. Value mengukur tingkat kecerahan warna, di mana 100% mewakili kecerahan maksimal dan 0% menghasilkan kegelapan penuh pada warna tersebut. Gambaran yang jelas mengenai warna model HSV terdapat pada Gambar 4.



Gambar 4. Komposisi HSV

Nilai HSV dapat diperoleh dari konversi nilai RGB. Konversi dari RGB ke HSV dapat diperoleh dengan persamaan.

$$h = \tan \left[\frac{3(g-b)}{(r-g)+(r-b)} \right] \quad (3)$$

$$s = 1 - \left[\frac{\min(r,g,b)}{v} \right] \quad (4)$$

$$v = \frac{r+g+b}{3} \quad (5)$$

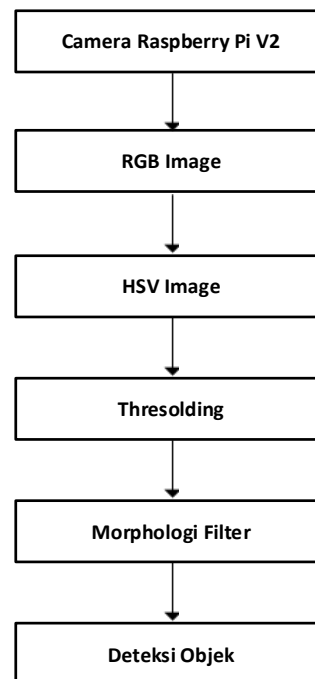
Untuk mendapatkan nilai HSV dapat dikonversi dari nilai RGB, RGB ke HSV dengan persamaan ini.

$$v = \max (R, G, B) \quad (6)$$

$$S = \begin{cases} 0 & \text{jika } v > 0 \\ v - \frac{\min(R,G,B)}{v} & \text{jika } v = 0 \end{cases} \quad (7)$$

$$H = \begin{cases} 0 & \text{jika } s = 0 \\ \frac{60x(g-b)}{sxv} & \text{jika } v = R \\ 60x \left[2 + \frac{(b-r)}{s \times v} \right] & \text{jika } v = G \\ 60x \left[4 + \frac{(r-g)}{s \times v} \right] & \text{jika } v = B \end{cases} \quad (8)$$

Dalam menggunakan filter HSV, harus mengkalibrasi setiap lintasan dan kondisi pencahayaan yang berbeda, jadi menggunakan fungsi ambang batas untuk menetapkan nilai HSV yang diterapkan pada sistem robot AGV.



Gambar 5. Pra-Pemrosesan Filter Warna HSV

Gambar 5 merupakan tahapan proses filter warna HSV yang digunakan untuk menghasilkan deteksi objek. Dalam proses transformasi citra, *thresholding* adalah salah satu teknik sederhana untuk mengubah citra dari

format grayscale menjadi citra biner, yang hanya memiliki dua kemungkinan warna pixel: hitam dan putih [19]. Dalam proses *thresholding*, metode segmentasi citra digunakan untuk memisahkan objek dari latar belakang dalam citra berdasarkan perbedaan tingkat kecerahan mereka.

Pada tahap filter morfologi juga melalui proses Dilasi, Erosi dan filter Gaussian Blur. Filter Morfologi merupakan teknik pengolahan gambar yang diterapkan untuk menganalisis dan mengubah struktur objek yang terdapat dalam gambar [20]. Pendekatan ini mengambil dasar dari bentuk dan pola yang ada pada objek, dan sangat bermanfaat dalam berbagai tugas seperti mereduksi gangguan citra (noise), mendeteksi tepi, meningkatkan kualitas gambar, serta memisahkan objek satu sama lain dalam citra. Kemudian pada tahap proses dilasi, dilasi merupakan salah satu tindakan dalam operasi morfologi pada pengolahan gambar yang bertujuan untuk membersihkan noise yang memiliki warna hitam menjadi warna putih serta memperluas segmen pada objek yang ada dalam citra. Fungsi dilasi digunakan untuk memperluas atau memperbesar kontur objek di dalam gambar dan membersihkan noise yang memiliki warna hitam (0) menjadi warna putih (1) serta memperluas segmen pada objek yang ada dalam citra [21]. Proses ini berbentuk matriks kecil guna mengelola piksel-piksel dalam gambar. Operasi dilasi sangat berguna untuk mengisi celah-celah kecil dalam objek atau untuk memperbesar objek. Kemudian proses erosi merupakan operasi mengecilkan atau mengikis batas-batas objek dalam gambar [22]. Hal ini dicapai dengan menempatkan kernel (matriks kecil) di atas setiap piksel pada gambar dan hanya mempertahankan nilai piksel jika semua piksel kernel yang sesuai juga benar memiliki nilai logika (1). Pada proses erosi dan dilasi dapat dituliskan dengan persamaan.

$$g(x, y) = f(x, y) \oplus SE \quad (9)$$

$$g(x, y) = f(x, y) \ominus SE \quad (10)$$

Kemudian pada proses filter Gaussian Blur bertujuan untuk menghasilkan efek keburaman atau pemulusan. Filter ini didesain untuk mengurangi noise frekuensi tinggi dan detail dalam gambar, sekaligus mempertahankan struktur keseluruhan dan komponen frekuensi rendah [23].

3. Hasil dan Pembahasan

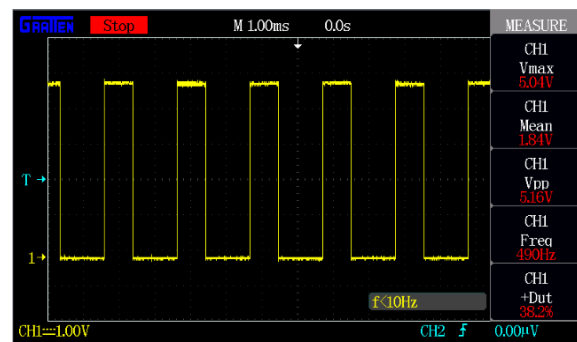
Berikut prototipe robot AGV yang digunakan dapat dilihat pada **Gambar 6**. Dalam pengujian alat ini dilakukan pada ruangan tertutup dengan pencahayaan yang cukup. Terdapat beberapa komponen penting yang digunakan pada pengujian alat ini yaitu Raspberry Pi 4B dengan kamera Raspberry V2 dan Arduino Uno. Penggunaan Raspberry Pi 4B beserta kamera Raspberry V2 ini berfungsi untuk mendeteksi pola jalur AGV yang sudah ditentukan. Kemudian penggunaan Arduino Uno

berfungsi untuk mengoperasikan motor DC yang menggerakkan roda depan dan belakang secara 4WD dengan disertai penggunaan motor servo yang berfungsi untuk steering dari robot AGV ini.



Gambar 6. Prototipe Robot AGV

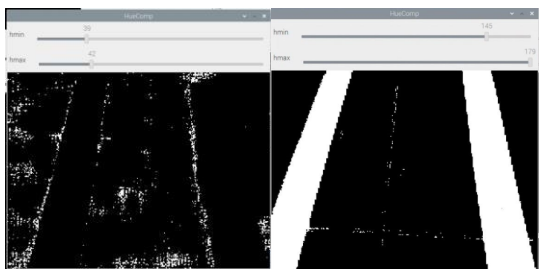
Pada sistem robot AGV dilakukan pengukuran sinyal PWM dengan menggunakan osiloskop. Motor DC hanya diatur pada 1 kondisi kecepatan dengan cara mengatur sinyal PWM pada Arduino Uno. Dengan kecepatan motor DC tersebut menghasilkan sinyal PWM yang ditunjukkan pada **Gambar 7**. Kecepatan motor DC ini diukur menggunakan oskiloskop yang menghasilkan sinyal PWM dengan duty cycle sebesar 38,2%. Dalam uji coba yang sudah dilakukan kecepatan motor DC dengan duty cycle 38,2% maka robot AGV yang mempunyai berat kurang lebih 15kg dapat berjalan dalam kecepatan 2m/s. Maka supaya proses *image processing* dapat berjalan dengan lancar, kecepatan maksimum robot AGV yang bisa dihasilkan yaitu 2m/s. Namun jika robot AGV diatur dalam kecepatan lebih dari 2m/s maka saat proses image processing berjalan menjadi terganggu dikarenakan tidak bisa mendeteksi objek dengan baik.



Gambar 7. Sinyal PWM

Kemudian dalam pengujian alat ini dilakukan proses uji coba filter HSV. Dalam uji coba filter ini memerlukan langkah awal dalam melakukan kalibrasi dari nilai

minimum hingga maksimum, serta dari nilai maksimum hingga minimum. Dalam uji coba ini, terdapat hasil dari kalibrasi nilai minimum dan maksimum *Hue*, *Saturation*, dan *Value*. Nilai minimum dan nilai maksimum dari uji coba filter HSV ini yaitu nilai $H_{min} = 145$ dan $H_{maks} = 179$, $S_{min} = 71$ dan $S_{maks} = 255$, dan $V_{min} = 67$ dan $V_{maks} = 123$ dari hasil nilai kalibrasi tersebut maka objek berupa garis dapat terdeteksi sesuai yang garis sudah ditentukan. Hasil dari sebelum kalibrasi nilai threshold dan setelah kalibrasi nilai threshold filter HSV tersebut terdapat pada **Error! Reference source not found.**, **Error! Reference source not found.**, dan **Error! Reference source not found.**



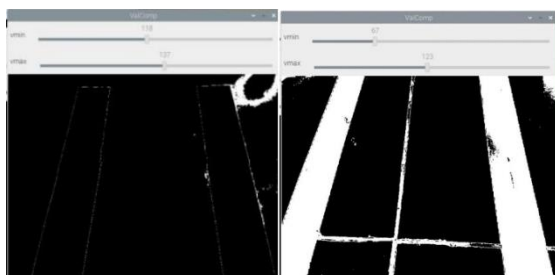
a) Sebelum *Thresholding*, b) Setelah *Thresholding*

Gambar 8. Penentuan Nilai *Hue*



a) Sebelum *Thresholding*, b) Setelah *Thresholding*

Gambar 9. Penentuan Nilai *Saturation*



a) Sebelum *Thresholding*, b) Setelah *Thresholding*

Gambar 10. Penentuan Nilai *Value*

Tabel 1 menunjukkan uji coba *thresholding Hue*, Tabel 2 menunjukkan uji coba *thresholding Saturation*, dan Tabel 3 menunjukkan uji coba *thresholding Value*. Pada Tabel 1 sampai Tabel 3 merupakan data sebelum *thresholding* dan

setelah *thresholding*. Dimana pada proses ini untuk mengatur nilai yang sesuai sehingga ditemukan nilai yang akurat dengan menggunakan metode *thresholding*. Kemudian dilakukan uji coba nilai melalui perubahan nilai min ke max dan max ke min. Metode ini dilakukan untuk menentukan warna hitam maupun putih pada jalur yang sudah ditentukan, jika garis lintasan terdeteksi warna putih maka pengujian filter HSV sempurna

Tabel 1. Uji Coba *Threshold Hue*

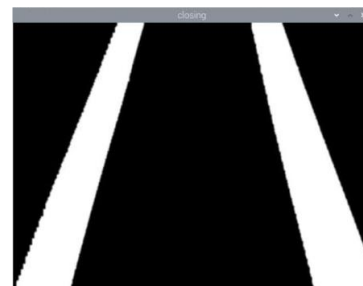
H_{min}	H_{max}	Warna Putih	Warna Hitam
39	42	Tidak Terdeteksi	Tidak Terdeteksi
145	179	Terdeteksi	Tidak Terdeteksi

Tabel 2. Uji Coba *Threshold Saturation*

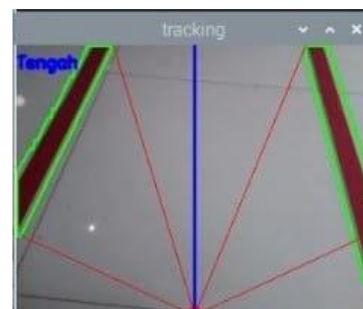
S_{min}	S_{max}	Warna Putih	Warna Hitam
58	43	Tidak Terdeteksi	Tidak Terdeteksi
71	255	Terdeteksi	Tidak Terdeteksi

Tabel 3. Uji Coba *Threshold Value*

V_{min}	V_{max}	Warna Putih	Warna Hitam
118	137	Tidak Terdeteksi	Tidak Terdeteksi
67	123	Terdeteksi	Tidak Terdeteksi



Gambar 11. Hasil Keluaran Proses Filter



Gambar 12. Hasil *Tracking*

Hasil akhir dari proses *thresholding* dan gaussian filter menghasilkan filter warna HSV tanpa noise. Sehingga filter HSV yang digunakan untuk mendeteksi objek berupa garis menghasilkan output yang sesuai dengan jalur yang ditentukan. Gambar 11 dan Gambar 12 merupakan gambar hasil keluaran dan hasil tracking.

Dalam uji coba kalibrasi nilai *thresholding* pada proses *image processing* menggunakan metode HSV yang terdapat pada **Error! Reference source not found.** sampai dengan **Error! Reference source not found.** Dalam uji coba tersebut saat nilai *thresholding* tidak tepat maka objek berupa garis lintasan tidak terdeteksi dengan baik bahkan objek bisa tidak terdeteksi sama sekali. Jika setelah proses *thresholding* dalam nilai yang tepat maka objek berupa garis lintasan terdeteksi dengan baik.

4. Kesimpulan

Dalam uji coba dan analisis, berhasil merancang prototipe robot AGV sesuai dengan spesifikasi sistem AGV visi komputer. Pemrosesan citra dengan metode HSV digunakan sebagai deteksi lintasan, sehingga robot AGV mempunyai kemampuan mengikuti lintasan marka dengan baik. Kemampuan robot AGV ini dirancang hanya pada satu kecepatan saja sehingga ketika robot AGV bergerak dalam kondisi jalan lurus maupun belok dalam kecepatan yang sama. Metode HSV yang ditanamkan pada robot AGV ini mempunyai akurasi tinggi dalam mendeteksi objek jika intensitas cahaya dalam kondisi yang stabil. Ambang batas dari kalibrasi mengizinkan metode HSV mengolah objek lintasan jalan secara efektif dengan rentang nilai $H_{min} = 145$, $H_{max} = 179$, $S_{min} = 71$, $S_{max} = 255$, dan $V_{min} = 67$, $V_{max} = 123$. Hasil menunjukkan pendekatan HSV efektif dalam mendeteksi marka jalan. Proses melibatkan Raspberry Pi untuk transmisi data ke Kamera Raspberry, mewujudkan visualisasi gambar real-time. Driver BTS7960 berfungsi baik. Sudut belok maksimum tergantung pada kecepatan kendaraan, semakin cepat, sudut belok semakin kecil, dan sebaliknya. Program dibuat dan diteliti menunjukkan akurasi tinggi dan hampir tanpa kesalahan dalam deteksi lintasan. Penelitian berfokus pada penerapan industri dengan tujuan menerapkan teknologi ini pada robot AGV di lingkungan industri.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Universitas Katolik Soegijapranata atas fasilitas dan tempat yang telah diberikan untuk melakukan penelitian di laboratorium. Semoga dengan adanya alat ini dapat bermanfaat bagi kemajuan teknologi di dunia. Selain itu, penulis juga ingin mengucapkan terima kasih kepada Rektor Universitas atas dukungannya.

Referensi

[1]. F. B. Setiawan, O. J. Aldo Wijaya, L. H. Pratomo, and S. Riyadi, "Sistem Navigasi Automated Guided Vehicle Berbasis Computer Vision dan Implementasi pada Raspberry Pi," *Jurnal Rekayasa ElektriKa*, vol. 17, no. 1, pp. 7–14, 2021, doi: 10.17529/jre.v17i1.18087.

- [2]. M. De Ryck, M. Versteijhe, and F. Debrouwere, "Automated guided vehicle systems, state-of-the-art control algorithms and techniques," *J Manuf Syst*, vol. 54, no. December 2019, pp. 152–173, 2020, doi: 10.1016/j.jmsy.2019.12.002.
- [3]. Sumardi, M. Taufiqurrahman, and M. A. Riyadi, "Street mark detection using raspberry pi for self-driving system," *Telkonnika (Telecommunication Computing Electronics and Control)*, vol. 16, no. 2, pp. 629–634, 2018, doi: 10.12928/TELKOMNIKA.v16i2.4509.
- [4]. C. Liu, J. Tan, H. Zhao, Y. Li, and X. Bai, "Path planning and intelligent scheduling of multi-AGV systems in workshop," *Chinese Control Conference, CCC*, pp. 2735–2739, 2017, doi: 10.23919/ChiCC.2017.8027778.
- [5]. B. Xu and D. Wang, "Magnetic Locating AGV Navigation Based on Kalman Filter and PID Control," *Proceedings 2018 Chinese Automation Congress, CAC 2018*, pp. 2509–2512, 2019, doi: 10.1109/CAC.2018.8623691.
- [6]. S. H. Bach and S. Y. Yi, "An Efficient Approach for Line-Following Automated Guided Vehicles Based on Fuzzy Inference Mechanism," *Journal of Robotics and Control (JRC)*, vol. 3, no. 4, pp. 395–401, 2022, doi: 10.18196/jrc.v3i4.14787.
- [7]. Á. Cservenák, "Motion Planning for Automated Guided Vehicle," *Acta Technica Corviniensis - Bulletin of Engineering*, vol. 11, no. 4, pp. 33–38, 2018, [Online]. Available: https://search.proquest.com/docview/2132669330?accountid=26540%0Ahttp://link.periodicos.capes.gov.br/sfxlc141?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&genre=article&sid=ProQ:ProQ%3Aengineeringjournals&article=MOTION+PLANNING+FOR+AUTOMAT
- [8]. A. Khedkar, K. Kajani, S. Banthia, B. N. Jagdale, M. Kulkarni, and B. E. Student, "Automated Guided Vehicle System with Collision Avoidance and Navigation in Warehouse Environments," *International Research Journal of Engineering and Technology*, no. May, pp. 5442–5448, 2020, [Online]. Available: www.irjet.net
- [9]. F. Irsyadi, D. N. Pratomo, S. Julianto, M. S. Anwar, and A. A. Paripurna Barus, "Desain dan Implementasi Sistem Navigasi pada Automated Guided Vehicle (AGV)," *Jurnal Listrik, Instrumentasi dan Elektronika Terapan (JuLIET)*, vol. 2, no. 1, 2021, doi: 10.22146/juliet.v2i1.64830.
- [10]. Ajib Susanto, Yupie Kusumawati, Ericsson Dhimas Niagara, and Christy Atika Sari, "Convolutional Neural Network Dalam Sistem Deteksi Helm Pada Pengendara Motor," *Seminar Nasional Teknologi dan Multidisiplin Ilmu (SEMNASTEKMU)*, vol. 2, no. 1, pp. 91–99, 2022, doi: 10.51903/semnastekmu.v2i1.158.
- [11]. J. Naranjo-Torres, M. Mora, R. Hernández-García, R. J. Barrientos, C. Fredes, and A. Valenzuela, "A review of convolutional neural network applied to fruit image processing," *Applied Sciences (Switzerland)*, vol. 10, no. 10, 2020, doi: 10.3390/app10103443.
- [12]. K. Selvaraj, S. Alagarsamy, and M. Dhilipkumar, "Raspberry Pi based automatic door control system," *2021 3rd International Conference on Signal Processing and Communication, ICSPC 2021*, no. May, pp. 652–656, 2021, doi: 10.1109/ICSPC51351.2021.9451687.

- [13]. G. Chandan, A. Jain, and H. Jain, "Proceedings of the 3rd International Conference on Inventive Research in Computing Applications, ICIRCA 2021," *Proceedings of the 3rd International Conference on Inventive Research in Computing Applications, ICIRCA 2021*, no. Icirca, pp. 1305–1308, 2021.
- [14]. H. M. Ahmed and R. T. Rasheed, "A Raspberry PI Real-Time Identification System on Face Recognition," *Proceedings of 2020 1st Information Technology to Enhance E-Learning and other Application Conference, IT-ELA 2020*, pp. 89–93, 2020, doi: 10.1109/IT-ELA50150.2020.9253107.
- [15]. G. Anand and A. K. Kumawat, "Object detection and position tracking in real time using Raspberry Pi," *Mater Today Proc.*, vol. 47, no. xxxx, pp. 3221–3226, 2021, doi: 10.1016/j.matpr.2021.06.437.
- [16]. S. Syukriyadin, S. Syahrizal, G. Mansur, and H. P. Ramadhan, "Permanent magnet DC motor control by using arduino and motor drive module BTS7960," *IOP Conf Ser Mater Sci Eng*, vol. 352, no. 1, 2018, doi: 10.1088/1757-899X/352/1/012023.
- [17]. A. Maarif and N. R. Setiawan, "Control of dc motor using integral state feedback and comparison with pid: Simulation and arduino implementation," *Journal of Robotics and Control (JRC)*, vol. 2, no. 5, pp. 456–461, 2021, doi: 10.18196/jrc.25122.
- [18]. A. Aqthobilrobbany, A. N. Handayani, D. Lestari, Muladi, R. A. Asmara, and O. Fukuda, "HSV Based Robot Boat Navigation System," *CENIM 2020 - Proceeding: International Conference on Computer Engineering, Network, and Intelligent Multimedia 2020*, pp. 269–273, 2020, doi: 10.1109/CENIM51130.2020.9297915.
- [19]. N. K. Hamzidah, M. M. Parenreng, J. Teknik, P. Negeri, and U. Pandang, *PROSES IDENTIFIKASI OBJEK PADA CITRA SEL LEUKOSIT DARAH MENGGUNAKAN TEKNIK PENGOLAHAN CITRA DIGITAL*.
- [20]. F. Triatmojo, B. Sugandi, P. N. Batam, T. Mekatronika, and J. A. Yanin, "Robot Pengikut Posisi dengan Menggunakan Filter Warna HSV," 2018.
- [21]. A. Asvin MahersatillahSuradi, M. Furqan Rasyid, M. Rizal, M. Informatika, and T. Universitas Dipa Makassar, "Deteksi Tingkat Kematangan Buah Apel Menggunakan Segmentasi Ruang Warna HSV," 2023.
- [22]. R. Risfendra, A. A. Akbar, and F. Firdaus, "Sistem Pergerakan dan Deteksi Pada Robot Sepak Bola Beroda Berbasis Image Processing dengan Penerapan Multivision," *INVOTEK: Jurnal Inovasi Vokasional dan Teknologi*, vol. 20, no. 3, pp. 31–42, Oct. 2020, doi: 10.24036/invotek.v20i3.830.
- [23]. D. Zanuvar, E. Prastya, D. Putra Pamungkas, and R. K. Niswatin, "Implementasi Metode Gaussian Filter Dan Median Filter Untuk Penghalusan Gambar."