

Pengenalan Tanda Arah untuk Navigasi *Automatic Guided Vehicle* berbasis Raspberry Pi

Florentinus Budi Setiawan¹, Rachmat Hidayat², Leonardus Heru Pratomo³, Slamet Riyadi⁴

^{1,2,3,4} Program Studi Teknik Elektro, Fakultas Teknik Universitas Katolik Soegijapranata, Jl. Pawiyatan Luhur Selatan.IV No.1 Semarang 50234 INDONESIA (telp:024-8441555; ¹f.budi.s@unika.ac.id, ²rachmatha2508@gmail.com, ^{3,4}tu.elektro@unika.ac.id)

[Diterima: 4 Juli 2022, Revisi: 28 Oktober 2022]

Corresponding Author: Florentinus Budi Setiawan

INTISARI — Perkembangan zaman modern di sektor teknologi robotika dan mekanisasi telah meningkat sangat signifikan dalam beberapa dekade ini karena efisiensinya yang tinggi dari aspek waktu dan tenaga. Pada sistem mobilisasi barang untuk pemanfaatan perusahaan, khususnya bagian industri dan bagian pergudangan, salah satu robot yang digunakan adalah kendaraan berpemandu otomatis pengangkut barang atau yang biasa disebut sebagai *automatic guided vehicle* (AGV). Salah satu metode navigasi lama di AGV adalah penggunaan sebuah sensor untuk mengikuti pola garis pada objek yang terdeteksi, yaitu garis pada lantai. Metode tersebut kurang efektif karena lambat laun objek pola garis di lantai tersebut akan menghilang akibat efek dari gaya gesek roda AGV, sehingga tidak lagi dapat terdeteksi oleh sensor kamera. Oleh sebab itu, diperlukan sebuah peningkatan metode navigasi AGV agar dapat menjadi sebuah inovasi yang berkelanjutan. Metode navigasi ini menggunakan empat objek gambar yang diposisikan pada area yang dilintasi robot AGV dan kamera sebagai sensor yang menghadap ke depan, sehingga AGV dapat dengan presisi mendeteksi pola objek gambar menggunakan bantuan *computer vision* menggunakan *library* perangkat lunak OpenCV. Selanjutnya, pola objek gambar yang sudah terdeteksi diproses oleh sebuah program yang dirancang pada perangkat komputer mini Raspberry Pi 4 Model B. Hasil pengujian membuktikan bahwa metode ini mampu mendeteksi objek gambar yang berada di area yang terjangkau kamera dan berhasil menampilkan keluaran dari objek gambar tersebut. Sistem berhasil mengenali objek secara cukup akurat, dengan parameter jarak 10–95 cm dan melalui beberapa percobaan. Analisis kecepatan putaran roda depan dan belakang AGV dilakukan menggunakan osiloskop dan takometer sebagai alat pengukur kecepatan atau rotasi roda.

KATA KUNCI — Robotika, AGV, *Computer Vision*, OpenCV, Raspberry Pi.

I. PENDAHULUAN

Dalam sistem pengiriman otomatis [1], *automatic guided vehicle* (AGV) adalah bagian yang sangat penting. AGV merupakan jenis kendaraan pengiriman barang yang dapat mengarahkan dirinya sendiri dengan mengikuti program dan rute pemandu untuk melakukan gerakan, seperti bergerak maju, berhenti, dan berbelok, serta berhenti di stasiun kerja yang diprogram. Kendaraan ini digunakan untuk memuat maupun membongkar barang [2]. AGV dapat diklasifikasikan menjadi dua kategori. Kategori pertama adalah AGV dengan *guide path* yang dilengkapi dengan pemancar dan penerima inframerah reflektif untuk melacak *guide path*, sedangkan yang kedua adalah AGV tanpa *guide path* yang digerakkan menggunakan *remote control* dengan gelombang radio [3].

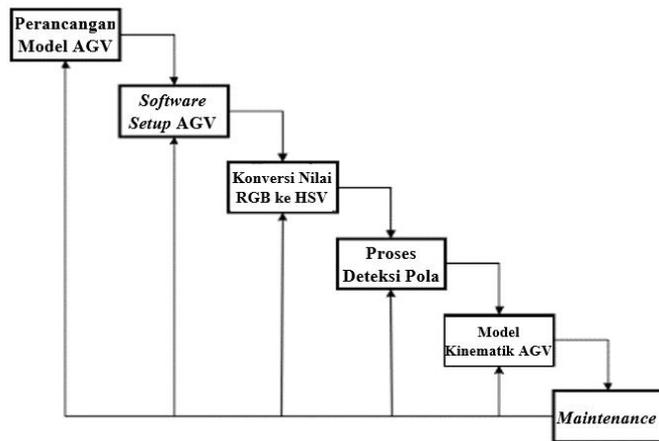
AGV dapat memanfaatkan penanda atau pola di lantai untuk melakukan navigasi dirinya sendiri di lingkungan tertentu [4]. Alat ini juga dapat memanfaatkan penglihatan menggunakan sebuah magnet atau laser untuk metode navigasi, tergantung pada aplikasinya, yang juga sering dimanfaatkan dalam aplikasi bidang industri untuk memindahkan bahan atau barang dari satu tempat ke tempat yang lain di fasilitas manufaktur atau gudang [5]. AGV dapat membantu meningkatkan produktivitas dan mengurangi biaya dengan memfasilitasi otomatisasi pabrik atau gudang. AGV memiliki beragam aplikasi yang sebagian besar digunakan untuk mengangkut berbagai jenis material, termasuk palet, gulungan, rak, gerobak, dan kontainer [6].

Bidang industri meningkatkan penggunaan robot untuk transportasi dan pemrosesan material [7]. Sistem robotika ini menggunakan banyak teknologi pengindraan inovatif dan teknik kontrol untuk meningkatkan keserbagunaannya. Sistem transportasi ini bergantung pada akurasi dan pengulangan dari AGV yang digunakan untuk memindahkan material dan

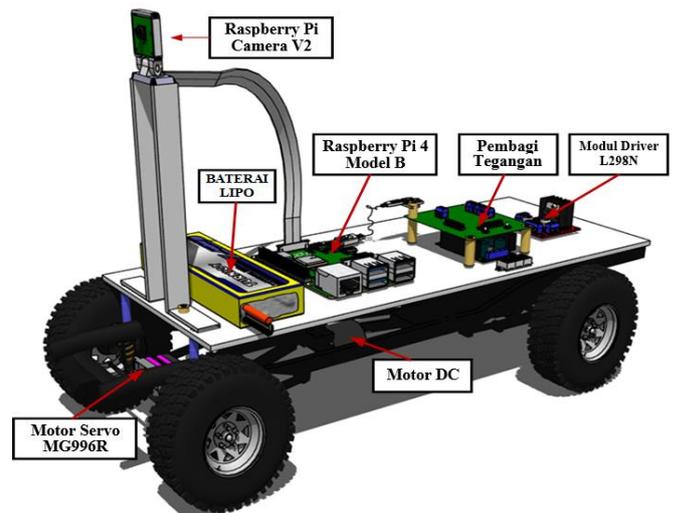
peralatan di sekitar ruang kerja. Kunci untuk mendapatkan akurasi yang diinginkan adalah kalibrasi AGV yang tepat dan terstruktur. Berbagai pendekatan yang berbeda untuk navigasi AGV telah banyak dibahas dalam literatur. Pendekatan navigasi dapat dipisahkan menjadi dua kategori besar, yaitu yang melakukan perhitungan secara manual dan yang mengandalkan referensi pemosisian eksternal. Perhitungan manual sebagian besar bergantung pada estimasi akurat dari jarak dan arah perjalanan AGV [8].

Kalibrasi AGV yang buruk dan tidak sesuai aturan dapat menyebabkan kesalahan pemosisian yang serius. Kesalahan kecil pada pengemudian dan jarak yang diukur akan terakumulasi dan menghasilkan kesalahan pemosisian yang besar dan tidak presisi [9]. Untuk AGV yang mengandalkan sebuah sumber daya pengaturan kendali navigasi eksternal, kemampuan mengarahkan servo secara akurat ke lokasi yang diinginkan dapat dipengaruhi oleh kualitas kalibrasi kendaraan. Penelitian ini mempresentasikan algoritme *computer vision* menggunakan model matematika untuk pengolahan citra [10] dalam memperoleh informasi dari gambar atau data.

Penglihatan manusia dalam menghadapi beberapa gambar yang berbeda dapat langsung dianalisis menurut pemahaman masing-masing. Akan tetapi, penggunaan komputer untuk analisis gambar akan menghasilkan lebih banyak metode interpretasi, yang secara keseluruhan lebih rumit. Oleh karena itu, dipilih penggunaan tipe data citra untuk analisis objek. Sebuah gambar yang diinterpretasikan melalui model probabilitas fisik, dan akhirnya algoritme *computer vision*, dapat terbentuk. Penelitian ini merupakan penyempurnaan dari sistem deteksi tanda arah [11]. *Computer vision* yang digunakan dalam penelitian ini akan menampilkan keluaran pada monitor, atau dapat dikatakan hanya tertampil hasil dari



Gambar 1. Bagan alur penelitian.



Gambar 2. Desain 3D prototipe AGV tampak samping.

pembacaan tanda arah saja. Maka, dalam penelitian ini diimplementasikan pembacaan tanda arah ini dengan sebuah mobil yang beroperasi dengan otomatis, atau dapat disebut sebagai AGV [11]. Dalam proses penerapan metode ini, algoritme *computer vision* dapat secara fleksibel mengidentifikasi gambar secara akurat dengan memanfaatkan empat buah gambar untuk navigasinya. Penggunaan metode *region of interest* (ROI) pengolahan citra ini dapat meningkatkan akurasi identifikasi yang diperoleh jika dibandingkan dengan beberapa metode lainnya. Makalah ini tersusun atas empat bagian: Bagian I adalah pendahuluan, lalu Bagian II membahas tentang metodologi, Bagian III membahas hasil dan pembahasan, serta Bagian IV berisi kesimpulan.

II. METODE YANG DIUSULKAN

Metode yang digunakan dalam penelitian ini adalah metode *waterfall*. Metode *waterfall* digunakan sebagai acuan untuk memastikan keberhasilan proyek penelitian [12]. Dalam metode *waterfall*, seluruh proses pengembangan perangkat dibagi menjadi beberapa fase terpisah yang saling melengkapi. Model ini merupakan hasil dari satu fase yang bertindak sebagai sebuah masukan untuk fase berikutnya secara berurutan. Bagian tersulit dari setiap penyelidikan ilmiah bukanlah memecahkan masalah tertentu, tetapi merumuskan pertanyaan yang memusatkan perhatian pada aspek-aspek penting dari fenomena yang sedang diselidiki [13]. Berikut ini adalah bagan tahapan penelitian dalam melakukan pembuatan robot AGV, seperti yang ditunjukkan pada Gambar 1.

A. PERANCANGAN MODEL AGV

Tahap yang pertama dilakukan adalah perancangan model desain AGV. Pembuatan alat ini dilakukan melalui proses desain 3D, seperti yang ditunjukkan pada Gambar 2. Penelitian ini membahas tentang perancangan pengenalan tanda arah yang memanfaatkan *computer vision* sebagai navigasi AGV [14]. Alat ini menggunakan komputer mini Raspberry Pi 4 Model B sebagai otaknya [15]. Komputer mini tersebut berperan sebagai komputer pusat yang memerintahkan gerakan AGV pada jalur yang telah ditentukan. Pengolahan citra juga dilakukan menggunakan komputer mini Raspberry Pi 4 model B. Lalu, untuk kemudi roda depan digunakan motor servo digital MG996R torsi tinggi [16]. Sensor AGV ini menggunakan Raspberry Pi Camera V2 yang merupakan *board* kamera resmi baru yang dirilis oleh Raspberry Pi, sedangkan untuk akselerasi di bagian roda depan dan belakang, digunakan *driver* L298N sebagai modul *driver* motor DC [17] dan pengendali kecepatan putaran roda dari motor DC tersebut [18]. Komputer *desktop*

digunakan sebagai penampil atau *screen mirroring* ketika AGV dioperasikan. Sebagai sumber daya kelistrikan AGV, digunakan baterai LiPo yang dapat di isi ulang, sedangkan bagian sasis AGV menggunakan sasis mobil *remote control* kecil 4WD dan papan akrilik sebagai penyangga robot AGV.

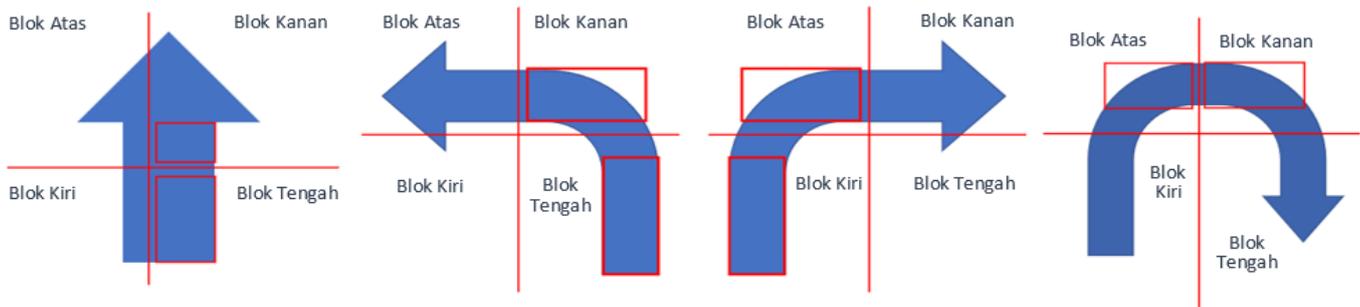
B. PENGATURAN PERANGKAT LUNAK AGV

Perancangan desain AGV dilakukan menggunakan perangkat lunak 3D SketchUp [19]. Algoritme dirancang dan diimplementasikan menggunakan Python 3.9.1 [20] dan *computer vision library* OpenCV [21] untuk perintah pengolahan citra. Kemudian, algoritme pengaturan kendali kecepatan dan kendali kemudi AGV diimplementasikan dalam bahasa Python [22], sedangkan *viewer board* Raspberry Pi 4 model B dijalankan dengan laptop menggunakan perangkat lunak VNC Viewer [23].

C. KONVERSI NILAI WARNA RGB KE NILAI HSV

Gambar yang ditangkap oleh kamera biasanya berada dalam ruang warna *red, green, blue* (RGB). Ruang warna ini perlu diubah menjadi ruang warna lain. Pendekatan yang paling populer dalam teknik segmentasi warna adalah model warna *hue, saturation, value* (HSV) [24]. Model warna RGB terutama digunakan dalam aplikasi grafik komputer. Model warna ini umumnya dibangun berdasarkan ruang koordinat Cartesian yang merepresentasikan sumbu *x, y, dan z*. Di sisi lain, model warna HSV dibangun berdasarkan representasi *hue, saturation, dan value*. *Hue* mewakili warna asli itu sendiri; *saturation* menunjukkan kemurnian warna yang berhubungan dengan simpangan baku di sekitar panjang gelombang dominan; sedangkan *value* mewakili kecerahan warna, dalam hal ini menggambarkan terang atau gelapnya warna dan berkisar dari 0 (gelap) hingga 1 (terang). *Value* terkait dengan jumlah warna putih [25].

Selain teknik berbasis warna, teknik berbasis bentuk juga telah digunakan dalam tahap deteksi objek. Teknik deteksi bentuk ini menggunakan *dataset common object in context* (COCO). Pada kenyataannya, teknik berbasis bentuk lebih sulit diterapkan karena tanda-tanda yang dideteksi memiliki tampilan dan bentuk yang beragam. Ini terjadi karena deteksi bentuk tanda sulit dilakukan jika gambar terlalu kecil dan buram akibat kamera menangkap gambar dalam jarak jauh pada kondisi dan sudut pencahayaan yang berbeda-beda [26]. Algoritme yang menggunakan deteksi warna HSV dan RGB dijelaskan sebagai berikut. Sebuah piksel ditandai berdasarkan



Gambar 3. Marking ROI.

rentang HSV. Batas-batas untuk algoritme ruang warna HSV digambarkan seperti pada (1).

$$(P_{hue_low} \geq 0,06) \text{ atau } (P_{hue_high} \leq 1) \text{ dan } (P_{saturation} \geq 0,75) \text{ dan } (P_{value} \geq 0,65). \quad (1)$$

Persamaan (1) menunjukkan bahwa P_{hue_low} dan P_{hue_high} mewakili nilai *hue*, $P_{saturation}$ adalah nilai *saturation*, dan P_{value} mewakili nilai *value*. Di sisi lain, segmentasi warna menggunakan algoritme RGB menggunakan batas warna yang berbeda. Selain itu, digunakan dekomposisi akromatik untuk mencari ROI kandidat gambar tanda arah. Persamaan dekomposisi akromatik ditunjukkan pada (2).

$$value(R, G, B) = \frac{(|B-R|+|G-B|+|R-G|)}{3E}. \quad (2)$$

Pada (2), nilai R, G , dan B mewakili kecerahan bayangan, sedangkan E adalah nilai total warna akromatik yang telah diekstraksi. Akromatik berarti kecerahan atau kegelapan warna dalam sebuah gambar yang berwarna hitam dan putih. Dalam implementasinya, nilai E diatur sebesar 20. Nilai ini dipilih karena memberikan hasil terbaik dalam segmentasi warna RGB. Berdasarkan algoritme yang diusulkan, citra tersebut kemudian diubah menjadi citra biner untuk memisahkan area ROI [27].

Jika algoritme menemukan piksel-piksel yang tergolong pola tanda arah, piksel-piksel tersebut akan diubah menjadi piksel warna putih, sedangkan piksel warna hitam mewakili latar belakang. Kemudian, piksel putih akan disegmentasi menggunakan kotak pembatas dan proses morfologi biner akan menghilangkan derau yang tidak diinginkan serta menggabungkan beberapa celah yang mungkin muncul karena proses *thresholding* yang tidak lengkap [28].

D. PROSES TAHAPAN PROGRAM

Dalam pembuatan alat ini, program terbagi menjadi beberapa subprogram, yaitu pemanggilan *library*, inialisasi kamera, definisi tanda arah, identifikasi tanda arah, dan pergerakan jalan AGV. Subprogram pemanggilan *library* berisi tentang daftar *library* yang digunakan dalam penyusunan program. Apabila *library* tidak dipanggil, program tidak akan berfungsi sebagaimana mestinya. Inialisasi kamera berfungsi untuk mengatur resolusi kamera, kecerahan kamera, arah rotasi kamera, dan *frame rate* kamera. Definisi tanda arah berisi beberapa prapemrosesan yang digunakan untuk mendefinisikan gambar yang hendak dibaca. Pada subprogram identifikasi tanda arah dilakukan pembacaan arah dengan metode ROI yang ada di dalamnya, sedangkan subprogram pergerakan arah AGV berisi mekanika pergerakan motor servo dan motor DC. Kedua motor ini saling terintegrasi sesuai dengan pola yang diinginkan. Semua subprogram ini menjadi satu kesatuan hingga menjadi algoritme pembacaan tanda arah.

E. PROSES DETEKSI POLA PADA GAMBAR NAVIGASI

Tujuan utama penelitian ini adalah untuk mencapai keberhasilan dalam mengidentifikasi tanda arah ataupun objek gambar agar prototipe AGV dapat berpindah posisi dengan memanfaatkan *computer vision* dan sensor Raspberry Pi Camera V2. Gambar dari video *real-time* dari kamera diubah menjadi citra *grayscale* untuk mengurangi kompleksitas dimensi warnanya, karena masukan warna RGB saja tidak cukup untuk mendeteksi objek di depan kamera. Untuk mencapai akurasi yang maksimal, derau pada citra *grayscale* harus dikurangi. Perlu ditentukan ROI dalam proses pengurangan derau tersebut agar pola terdeteksi. Selanjutnya, citra tersebut harus didefinisikan atau diubah ke nilai HSV.

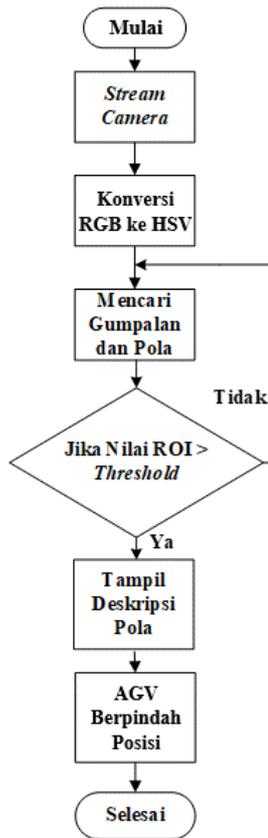
Jika nilai HSV untuk warna biru telah terdefinisi, dilakukan pengenalan pola yang berwarna putih di area warna biru. Lalu, gambar objek akan diperhalus menggunakan algoritme *kernel*. Setelah melalui proses algoritme *kernel*, pola gambar dideteksi menggunakan operasi morfologi dan kemudian fungsi *contours*, setelah melewati keempat ROI.

Nilai *tuple* satu dan nol dibandingkan, kemudian pola dideteksi menggunakan *look up table* yang merupakan bagian dari metode pengenalan ROI. Jika nilai segmen lebih besar dari nilai *threshold*, nilai segmen ditetapkan sebagai 1; dan jika tidak, nilai segmen ditetapkan sebagai 0. Nilai segmentasi harus sesuai dengan *look-up table* agar dapat menentukan arah navigasi pada AGV.

Jika terdeteksi gambar *move forward*, AGV akan bergerak maju; jika terdeteksi gambar *turn back*, AGV akan putar balik; jika terdeteksi gambar *turn left*, AGV akan berbelok ke kiri; dan jika terdeteksi gambar *turn right*, AGV akan berbelok ke kanan. Gambar 3 memperlihatkan *marking ROI* pada deteksi tanda arah menggunakan metode ROI. Diagram alir proses pengolahan citra ditunjukkan pada Gambar 4.

F. MODEL KINEMATIK AGV

Model prototipe AGV didasarkan pada model kinematik yang baik untuk representasi kecepatan lambat [29]. Untuk merancang prototipe AGV, didefinisikan model kinematika yang mengatur mekanisme pergerakannya. Model kinematika kendaraan dimaksudkan untuk simulasi *real-time*. Oleh karena itu, waktu komputasi perlu dijaga serendah mungkin. Selanjutnya, akan dibahas pemodelan rinci robot penggerak diferensial, terutama model kinematika. Robot memiliki empat roda penggerak yang mengontrol gerak AGV. Prototipe robot AGV ini mempertimbangkan *differential drive kinematics* robot yang memanfaatkan kecepatan roda sebagai pengontrol gerak AGV. Roda depan dan belakang dihubungkan oleh poros logam atau sasis. Dari model AGV ini, konfigurasi empat roda mengilustrasikan derajat kebebasan AGV; yang pertama memungkinkan AGV bergerak maju dan mundur dan yang kedua memungkinkan AGV berputar atau berbelok kanan dan



Gambar 4. Diagram alir proses pengolahan citra.

kiri, dengan mengatur nilai γ (dalam π rad). Koordinat global (X, Y) AGV diilustrasikan pada Gambar 5. Parameter v_b adalah kecepatan roda belakang AGV (m/s); v_f adalah kecepatan roda depan AGV (m/s); l_a dan l_b mewakili jarak dari pusat massa AGV ke poros R dan P (m); sedangkan l_f adalah jarak dari poros AGV ke titik Q. Kinematika AGV dirumuskan seperti pada (3) sampai (5).

$$\dot{X} = V \cdot \cos(\Psi + \beta) \quad (3)$$

$$\dot{Y} = V \cdot \sin(\Psi + \beta) \quad (4)$$

$$\dot{\Psi} = \frac{V \cdot \cos \beta}{l_a + l_b + l_f} \quad (5)$$

Parameter β mewakili sudut slip AGV (dalam π rad), arah AGV ditentukan oleh poros yang ditentukan oleh sudut γ , dan kecepatan awal AGV ditentukan oleh kecepatan roda belakang v_b , dengan nilai sudut slip β dan kecepatan linear AGV dituliskan pada (6) dan (7).

$$\beta = \tan^{-1} \left(\frac{l_f \cdot \tan \gamma}{l_a + l_b + l_f} \right) \quad (6)$$

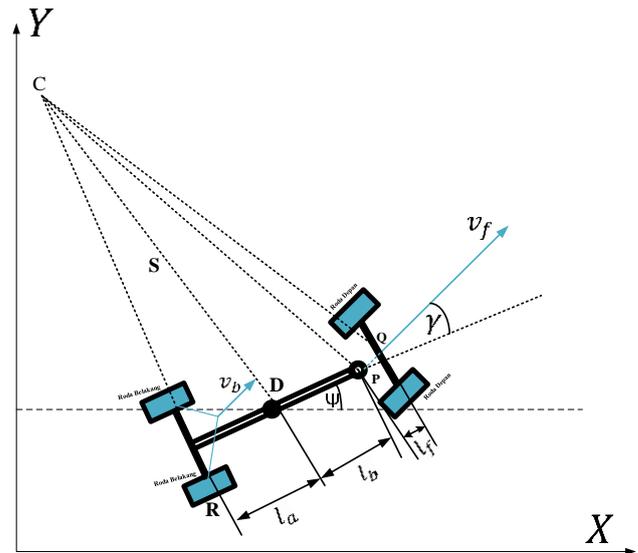
$$V = \frac{v_b \cdot \cos \gamma + v_b}{2 \cos \beta} \quad (7)$$

Posisi AGV kemudian ditentukan dengan menerapkan integrasi pada (3) sampai (5) terhadap waktu t (s) seperti yang ditunjukkan pada (8) sampai (10).

$$X = \int_0^t V \cdot \cos(\Psi + \beta) dt \quad (8)$$

$$Y = \int_0^t V \cdot \sin(\Psi + \beta) dt \quad (9)$$

$$\Psi = \int_0^t \frac{V \cdot \cos \beta}{l_a + l_b + l_f} (\tan \gamma) dt. \quad (10)$$



Gambar 5. Kerangka kinematik model AGV.

Nilai parameter l_a , l_b , l_f , v_b , $\cos \beta$, $\cos \gamma$, dan $\tan \gamma$ dianggap sebagai konstanta, sehingga

$$X = \frac{V \cdot \cos \beta}{l_a + l_b + l_f} \cdot (\tan \gamma) \cdot \left[\sin \left(\frac{V \cdot \cos \beta}{l_a + l_b + l_f} \cdot (\tan \gamma) \cdot t + \beta \right) - \sin \beta \right] \quad (11)$$

$$Y = \frac{V \cdot \cos \beta}{l_a + l_b + l_f} \cdot (\tan \gamma) \cdot \left[\cos \beta - \cos \left(\frac{V \cdot \cos \beta}{l_a + l_b + l_f} \cdot (\tan \gamma) \cdot t + \beta \right) \right] \quad (12)$$

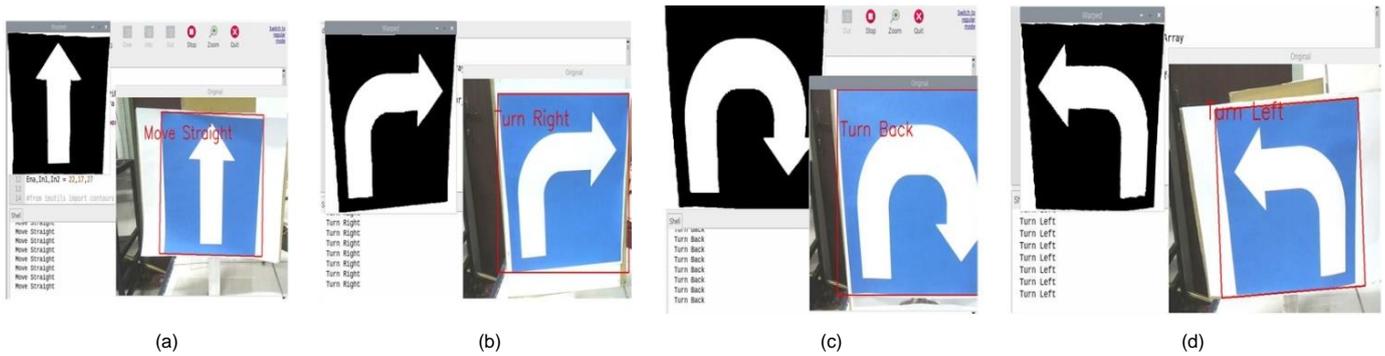
$$\Psi = \frac{V \cdot \cos \beta}{l_a + l_b + l_f} \cdot (\tan \gamma) \cdot t. \quad (13)$$

Parameter X , Y , dan Ψ masing-masing adalah koordinat global (X, Y) dari AGV dan sudut belok AGV pada saat itu [30].

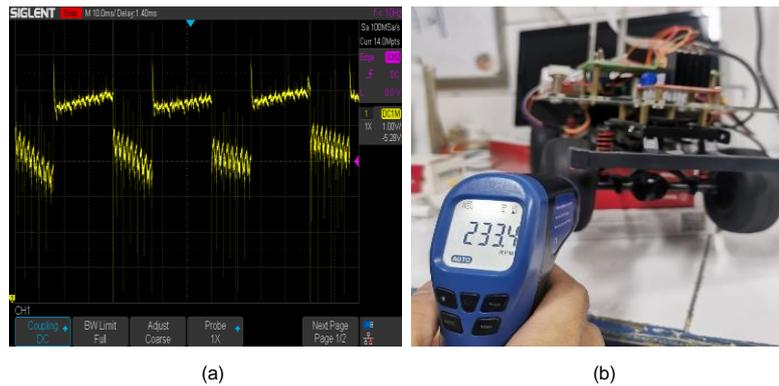
III. HASIL DAN PEMBAHASAN

A. PENGUJIAN PENGOLAHAN CITRA

Pada tahap ini, dilakukan pengujian saat kamera menangkap pola objek yang dideteksi. Jika pola terdeteksi benar, akan muncul kotak persegi berwarna merah sebagai pemisah antara objek yang diidentifikasi sebagai pola dan objek lain yang tidak terdeteksi sebagai pola tanda arah. Kemudian, dihasilkan teks keterangan yang akan muncul pada objek yang teridentifikasi sebagai pola. Empat gambar digunakan sebagai masukan navigasi AGV. Gambar 6 menunjukkan hasil identifikasi tanda arah untuk navigasi AGV. Pada Gambar 6(a) sampai Gambar 6(d) dapat dilihat bahwa ketika tanda arah dideteksi oleh kamera robot AGV, sistem akan langsung mendeteksi pola dan hasil ditampilkan pada layar monitor. Apabila tanda arah pada Gambar 6(a) dideteksi, akan dihasilkan teks "Move Straight", kemudian AGV akan berjalan lurus. Jika tanda arah pada Gambar 6(b) dideteksi,



Gambar 6. Hasil pengujian gambar pada navigasi dengan ukuran yang berbeda, (a) bergerak maju, (b) belok kanan, (c) berputar balik arah, (d) belok kiri.



Gambar 7. Hasil pengukuran menggunakan osiloskop dan takometer, (a) PWM dengan *duty cycle* 60%, (b) hasil kecepatan dari PWM dengan *duty cycle* 60%.

akan dihasilkan teks “Turn Back” dan AGV akan bermanuver putar balik. Selanjutnya, jika tandan arah pada Gambar 6(c) dideteksi, akan dihasilkan “Turn Left”, kemudian AGV akan bermanuver belok kiri, dan yang terakhir, apabila tanda arah pada Gambar 6(d) dideteksi, akan dihasilkan “Turn Right”, kemudian AGV akan bermanuver belok kanan. Jika tanda arah ini dibalik atau diputar arahnya, tanda arah tidak dapat terdeteksi. Pengujian hasil deteksi marka dan putaran roda pada AGV akan ditampilkan pada topik pembahasan selanjutnya.

Hasil pengujian deteksi arah untuk navigasi AGV dengan parameter jarak melalui beberapa percobaan pengujian ditunjukkan pada Tabel I. Pada jarak 10 cm sampai dengan 80 cm, didapatkan persentase keberhasilan deteksi sebesar 100% dengan total tiga kali percobaan. Pada jarak 90 cm dan 95 cm dengan tiga kali percobaan, diperoleh persentase keberhasilan sebesar 33,3%. Untuk menghitung total persentase keberhasilan, digunakan persamaan berikut.

$$\frac{\text{Total keberhasilan jarak}}{\text{Total percobaan jarak}} \times 100\% = \text{persentase keberhasilan} \quad (14)$$

$$\frac{20}{27} \times 100\% = 81,4\%$$

Total keberhasilan jarak adalah total keberhasilan dalam mendeteksi objek dari percobaan yang telah dilakukan, *Total percobaan jarak* adalah total percobaan yang telah dilakukan. dari (14), diperoleh *persentase keberhasilan* deteksi objek sebesar 81,4%.

B. HASIL PENGUKURAN KECEPATAN AGV

Pada tahap ini dilakukan pengukuran putaran roda belakang AGV dengan parameter ukur *pulse width modulation* (PWM) menggunakan osiloskop dan takometer digital. Pengukuran dilakukan dengan mengatur *duty cycle* PWM dari 25% sampai dengan 95%. Kecepatan putaran roda AGV dipengaruhi oleh besar kecilnya *duty cycle* PWM. Pengukuran pertama dilakukan dengan mengubah *duty cycle* PWM menjadi 60%. Gambar 7(a) menunjukkan isyarat PWM yang tertampil di

TABEL I
PENGUJIAN DETEKSI ARAH NAVIGASI

No.	Jarak	Percobaan ke-		
		1	2	3
1	10 cm	Berhasil	Berhasil	Berhasil
2	25 cm	Berhasil	Berhasil	Berhasil
3	65 cm	Berhasil	Berhasil	Berhasil
4	70 cm	Berhasil	Berhasil	Berhasil
5	75 cm	Berhasil	Berhasil	Berhasil
6	80 cm	Berhasil	Berhasil	Berhasil
7	85 cm	Berhasil	Berhasil	Gagal
8	90 cm	Berhasil	Gagal	Gagal
9	95 cm	Berhasil	Gagal	Gagal

osiloskop, sedangkan Gambar 7(b) menunjukkan kecepatan yang dihasilkan, yaitu sebesar 233,4 rpm.

IV. KESIMPULAN

Penelitian ini telah berhasil mengimplementasikan algoritme pengolahan citra menggunakan metode ROI dalam mendeteksi empat pola pada objek berupa gambar tanda arah. Deteksi dilakukan oleh prototipe AGV, dengan total persentase keberhasilan sebesar 81,4%. Tahapan proses navigasi AGV telah bekerja dengan baik. Pengolahan citra yang memanfaatkan *computer vision* dapat membaca perintah dengan baik pula. Parameter pengujian berupa jarak antara objek gambar dengan kamera. Selain itu, motor DC dan modul *driver* L298N pada AGV memiliki kinerja sangat baik dalam mengatur torsi, kecepatan, maupun *duty cycle* PWM. Robot AGV ini dapat menjadi contoh sebagai acuan untuk pengembangan robot selanjutnya. Penelitian lebih lanjut dapat memanfaatkan metode-metode pengolahan citra yang lain, seperti metode *pattern recognition* maupun metode *sign traffic recognition*.

KONFLIK KEPENTINGAN

Semua penulis menyatakan bahwa tidak terdapat konflik kepentingan dalam penelitian ini.

KONTRIBUSI PENULIS

Konseptualisasi, Florentinus Budi Setiawan dan Rachmat Hidayat; metodologi, Leonardus Heru Pratomo; desain sistem, Florentinus Budi Setiawan dan Rachmat Hidayat; perangkat lunak, Rachmat Hidayat; analisis, Rachmat Hidayat; penulisan-penyusunan draf asli, Florentinus Budi Setiawan; penulisan-peninjauan dan penyuntingan, Florentinus Budi Setiawan, Leonardus Heru Pratomo, dan Slamet Riyadi.

REFERENSI

- [1] J.H. Chou, "Automatic Guided Vehicle," *Proc. 1995 Int. IEEE/IAS Conf. Ind. Autom. Control Emerg. Technol.*, 1995, hal. 241-245, doi: 10.1109/iaacet.1995.527570.
- [2] E.M. Ngandu, N. Luwes, dan K. Kusakana, "Navigation System for an Automatic Guided Vehicle," *J. Phys. Conf. Ser.*, Vol. 1577, hal. 1-11, 2020, doi: 10.1088/1742-6596/1577/1/012030.
- [3] S. Legowik, R.V Bostelman, T. Hong, dan E.R. Messina, "Guideline for Automatic Guided Vehicle Calibration," National Institute of Standards and Technology, Gaithersburg, MD, AS, Interagency/Internal Report, NISTIR 8168, Mar. 2017., doi: 10.6028/NIST.IR.8168.
- [4] X. Zhang dan S. Xu, "Research on Image Processing Technology of Computer Vision Algorithm," *2020 Int. Conf. Comput. Vision, Image Deep Learn. (CVIDL)*, 2020, hal. 122-124, doi: 10.1109/CVIDL51233.2020.00030.
- [5] R. Bostelman, T. Hong, dan G. Cheok, "Navigation Performance Evaluation for Automatic Guided Vehicles," *IEEE Conf. Technol. Pract. Robot Appl. (TePRA)*, 2015, hal. 1-6, doi: 10.1109/TePRA.2015.7219684.
- [6] M. Antony dkk., "Design and Implementation of Automatic Guided Vehicle for Hospital Application," *2020 5th Int. Conf. Commun., Electron. Syst. (ICCES)*, 2020, hal. 1-6, doi: 10.1109/ICCES48766.2020.09137867.
- [7] M.N. Tamara dkk., "Electronics System Design for Low Cost AGV Type Forklift," *2018 Int. Conf. Appl. Sci., Technol. (iCAST)*, 2018, hal. 464-469, doi: 10.1109/iCAST1.2018.8751559.
- [8] X. Zhou, T. Chen, dan Y. Zhang, "Research on Intelligent AGV Control System," *Proc. 2018 Chinese Autom. Congr. (CAC)*, 2019, hal. 58-61, doi: 10.1109/CAC.2018.8623384.
- [9] J. Lee, C.H. Hyun, dan M. Park, "A Vision-Based Automated Guided Vehicle System with Marker Recognition for Indoor Use," *Sensors* Vol. 13, No. 8, hal. 10052-10073, Agu. 2013, doi: 10.3390/s130810052.
- [10] J.V. Stone, "Computer Vision: What Is The Object?," *Artif. Intell. Simul. Behav.*, 1993, hal. 199-208.
- [11] D. Priyanka dkk., "Traffic Light and Sign Detection for Autonomous Land Vehicle Using Raspberry Pi," *2017 Int. Conf. Invent. Comput. Inform. (ICICI)*, 2018, hal. 160-164, doi: 10.1109/ICICI.2017.8365328.
- [12] W. Nugraha, M. Syarif, dan W.S. Dharmawan, "Penerapan Metode SDLC Waterfall dalam Sistem Informasi Inventori Barang Berbasis Desktop," *JUSIM (J. Sist. Inf. Musirawas)*, Vol. 3, No. 1, hal. 22-28, Mei 2018, doi: 10.32767/jusim.v3i1.246.
- [13] C. Gao dan G. Hembroff, "Implications of Modified Waterfall Model to the Roles and Education of Health IT Professionals," *2012 IEEE Netw. Oper. Manage. Symp.*, 2012, hal. 1368-1369, doi: 10.1109/NOMS.2012.6212076.
- [14] A. Chakraborty, "Image Processing and Image Pattern Recognition a Programming Tutorial," *2018 First Int. Conf. Artif. Intell. Ind.*, 2018, hal. 122-123, 10.1109/AI4I.2018.8665702
- [15] R. Helbet, V. Monda, A.C. Bechet, dan P. Bechet, "Low Cost System for Terrestrial Trunked Radio Signals Monitoring Based on Software Defined Radio Technology and Raspberry Pi 4," *2020 Int. Conf., Expo. Elect., Power Eng. (EPE)*, 2020, hal. 438-441, doi: 10.1109/EPE50722.2020.9305536.
- [16] TowerPro, "MG996R High Torque Metal Gear Dual Ball Bearing Servo," MG996R datasheet, Agu. 2015.
- [17] B. Hamzi dkk., "Implementation of Data Driven Control System," *Electron.*, Vol. 8, No. 2, hal. 1801-1804, 2021.
- [18] S.S. Babu dan A. Suresh, "Current Programmed Controlled DC-DC Converter for Emulating the Road Load in Six Phase Induction Motor Drive in Electric Vehicle," *2020 Int. Conf. Power Electron., Renew. Energy Appl. (PEREA)*, 2020, hal. 1-6, doi: 10.1109/PEREA51218.2020.9339779.
- [19] A.A. Putri dan T. Aditya, "3D Modelling and Visualization of Drinking Water Supply System Using 3D GIS," *2017 7th Int. Annu. Eng. Sem. (InAES)*, 2017, hal. 1-6, doi: 10.1109/INAES.2017.8068574.
- [20] F. Liu dan Z. Yang, "Design of VMware vSphere Automatic Operation and Maintenance System Based on Python," *2018 Int. Conf. Adv. Mechatron. Syst. (ICAMEchS)*, 2018, hal. 283-286, doi: 10.1109/ICAMEchS.2018.8506789.
- [21] X. Farhodov dkk., "Faster RCNN Detection Based OpenCV CSRT Tracker Using Drone Data," *Int. Conf. Inf. Sci., Commun. Technol. (ICISCT)*, 2019, hal. 1-3, doi: 10.1109/ICISCT47635.2019.9012043.
- [22] A. Ohno, S. Matsumoto, M. Ohshita, dan K. Kaida, "A Learning Support System of C Programming Language for Novices as a Platform for Learning Analytics," *2020 9th Int. Congr. Adv. Appl. Inform. (IIAI-AAI)*, 2020, hal. 270-273, doi: 10.1109/IIAI-AAI50415.2020.00060.
- [23] H.N. Chi, P.J. Lee, dan C.L. Lo, "Visual Tracking Cleaner - A Robot Implements on the Whiteboard," *2020 Int. Conf. Syst. Sci., Eng. (ICSSE)*, 2020, hal. 1-4, doi: 10.1109/ICSSE50014.2020.9219288.
- [24] N.M. Ali, N.K.A. Md Rashid, dan Y.M. Mustafah, "Performance Comparison Between RGB and HSV Color Segmentations for Road Signs Detection," *Appl. Mech. Mater.*, Vol. 393, hal. 550-555, Sep. 2013, doi: 10.4028/www.scientific.net/AMM.393.550.
- [25] G. Kumar, P.P. Sarthi, P. Ranjan, dan R. Rajesh, "Performance of K-Means Based Satellite Image Clustering in RGB and HSV Color Space," *2016 Int. Conf. Recent Trends Inf. Technol. (ICRITT)*, 2016, hal. 1-5, doi: 10.1109/ICRITT.2016.7569523.
- [26] A. Ajmal, C. Hollitt, M. Freaan, dan H. Al-Sahaf, "A Comparison of RGB and HSV Colour Spaces for Visual Attention Models," *2018 Int. Conf. Image, Vis. Comput. New Zeal. (IVCNZ)*, 2018, hal. 1-6, doi: 10.1109/IVCNZ.2018.8634752.
- [27] S. Sudha, K.B. Jayanthi, C. Rajasekaran, dan T. Sunder, "Segmentation of Rol in Medical Images Using CNN- A Comparative Study," *TENCON 2019 - 2019 IEEE Reg. 10 Conf. (TENCON)*, 2019, hal. 767-771, doi: 10.1109/TENCON.2019.8929648.
- [28] C. Shi-Gang dkk., "Study on Segmentation of Lettuce Image Based on Morphological Reorganization and Watershed Algorithm," *2018 Chin. Control, Decis. Conf. (CCDC)*, 2018, hal. 6595-6597, doi: 10.1109/CCDC.2018.8408290.
- [29] A.J. Weinstein dan K.L. Moore, "Pose Estimation of Ackerman Steering Vehicles for Outdoors Autonomous Navigation," *2010 IEEE Int. Conf. Ind. Technol.*, 2010, hal. 579-584, doi: 10.1109/ICIT.2010.5472738.
- [30] N. Alfiany dkk., "Kinematics and Simulation Model of Autonomous Indonesian 'Becak' Robot," *2020 IEEE Reg. 10 Symp. (TENSYP)*, 2020, hal. 1692-1695, doi: 10.1109/TENSYP50017.2020.9230782.