

RANCANG BANGUN SELF-DRIVING CAR ROBOT BERBASIS PENGENALAN WARNA RGB DENGAN KAMERA SEBAGAI SENSOR MENGGUNAKAN ALGORITMA FUZZY LOGIC

DESIGN FOR SELF-DRIVING CAR ROBOT BASED RGB COLOR RECOGNITION WITH CAMERA AS A SENSOR USING FUZZY LOGIC ALGORITHM

Irfan Nurrahmat.¹, Agus Virgono², Randy Erfa Saputra³

^{1, 2, 3} Universitas Telkom, Bandung

irfannurrahmat@student.telkomuniversity.ac.id¹, avirgono@telkomuniversity.ac.id²,

resaputra@telkomuniversity.ac.id³

Abstrak

Penerapan robot telah banyak digunakan untuk berbagai industri dalam bekerja, dalam meningkatkan efisiensi pekerjaan dan meningkatkan berbagai aktivitas yang tidak dapat dilakukan oleh tenaga manusia. Robot yang dirancang dengan berbasis mengikuti objek warna *Red, Green, and Blue (RGB)* menggunakan kamera *webcam*, untuk menghasilkan aktivitas kegiatan yang diinginkan oleh pengguna. Robot dibuat untuk menangkap dan mengolah gambar pada objek warna yang tersedia. Pemrosesan dilakukan oleh *minicomputer* di *Raspberry Pi* sebagai kontrol utama yang menggunakan algoritma *Fuzzy Logic* untuk menggerakkan motor secara mobile robot. Nilai ekstraksi setiap warna diperlukan untuk membandingkan nilai ekstraksi yang diperoleh dikarenakan terdapat faktor kondisi cahaya yang berbeda. Perlu dilakukan beberapa kali percobaan untuk menemukan persentase keberhasilan dan kegagalan. Robot beroda akan mengenali 12 warna yaitu dengan warna merah, kuning, hijau, biru, coklat, merah marun, oranye, ungu, abu-abu, magenta, biru muda, dan putih. Setiap warna menghasilkan *output* yang berbeda sesuai dengan *rule based*.

Kata Kunci: *Fuzzy Logic, RGB Color Model, Raspberry Pi, Camera, Wheeled robot*

Abstract

The application of robots has been widely used for various industries in the work, in increasing the efficiency of work and increasing various activities that cannot be done by human power. The robot, which is designed based on following *Red, Green, and Blue (RGB)* color objects, uses a *webcam* camera, to generate the activity the user wants. Robots are created to capture and process images on available color objects. The processing is done by *minicomputers* in *Raspberry Pi* as the main control that uses *fuzzy logic* algorithms to drive the motor in a mobile robot. The extraction value of each color is required to compare the extracted values obtained due to different light condition factors. It is necessary to do several experiments to find the percentage of success and failure. The wheeled robot will recognize 12 colors, like red color, yellow, green, blue, brown, maroon, orange, purple, gray, magenta, light blue, and white. Each color produces a different output according to rule based.

Keywords: *Fuzzy Logic, RGB Color Model, Raspberry Pi, Camera, Wheeled robot*

1. Pendahuluan

Dunia robot telah mengalami kemajuan yang sangat cepat. Berbagai penelitian telah dilakukan selama proses pengembangan, dengan adanya robot dalam kehidupan manusia telah banyak membantu kegiatan aktivitas yang tidak

dapat dilakukan oleh manusia itu sendiri. Kendali robot dapat melakukan pekerjaan sesuai dengan perintah manual atau berada dalam system tertanam di robot. Penerapan robot dalam kehidupan manusia bahkan dalam aktivitas mengemudi, dapat terbantu oleh adanya kendaraan mobil tanpa pengemudi. Selain itu juga mobil dengan kemampuan mendeteksi warna merupakan suatu hal utama untuk membuat *self-driving* itu sendiri. Robot dapat mengenali perintah berdasarkan warna objek.

Pengenalan warna robot merupakan salah satu dari penelitian yang berhubungan dengan pengenalan objek untuk membuat robot tanpa pengemudi. Robot mampu melakukan pengenalan warna dengan menerapkan algoritma yang berada dan fungsi pengenalan warna yang ditangkap oleh kamera. Kamera yang dipasang dapat mengenali gambar berwarna *RGB*. Warna akan diproses dengan melakukan operasi seperti mengenali nilai setiap piksel gambar merah, hijau, dan biru pada kertas yang dicetak. Robot beroda dengan kemampuan mengenali warna perlu identifikasi tentang nilai piksel dari setiap gambar warna diterima.

Ekstraksi fitur untuk mendapatkan fitur dari setiap warna. Karakteristik citra akan didasarkan pada kondisi pengenalan berdasarkan logika. Objek warna gambar *RGB* yang telah ditangkap dengan menggunakan *webcam*, aturan-aturan disesuaikan berdasarkan logika yang sudah dibuat dengan kondisi tertentu dan telah ditetapkan. Robot beroda akan menjalankan perintah sesuai dengan kondisi yang ditentukan berdasarkan warna.

Penelitian ini menggunakan metode pengolahan citra yang mampu membedakan warna, sehingga setiap warna yang diekstraksi mampu memberikan *output* yang sesuai dengan perintah yang ditetapkan, warna yang akan diuji coba yaitu warna merah, kuning, hijau, biru, coklat, merah marun, oranye, ungu, abu-abu, magenta, biru muda, dan putih. Metode harus dilakukan beberapa kali percobaan, karena mencari nilai piksel setiap warna dikarenakan terdapat beberapa kondisi seperti pencahayaan yang kurang.

2. Landasan Teori

2.1 Warna

Kata Warna adalah elemen yang tidak berdiri sendiri, warna merupakan penampakan fisik dari berbagai bentuk yang dilihat oleh mata yang digunakan untuk benda mati dan benda hidup, beragam warna didasari dengan warna merah, biru, dan kuning. Berbagai warna dapat dibuat dengan ketiga element tersebut dengan proporsi tertentu [1]. Para ahli menyebutkan bahwa warna artinya, adalah suatu unsur penampilan terbuat dari pigmen atau pewarna yang dibentuk oleh pembiasan cahaya pada prisma yang menyusun spektrum warna pelangi. Warna primer adalah warna tanpa campuran, yang termasuk warna primer adalah merah, biru, kuning. Contoh warna sekunder campuran adalah warna merah dan kuning yang menghasilkan warna jingga. Warna tersier merupakan warna yang tercampur dengan warna sekunder menghasilkan warna baru seperti warna jingga dan warna kuning menghasilkan warna jingga kekuningan [2].

2.2 Citra

Citra disebut juga matriks dan setiap bidang indeks baris dan kolom nilai kecerahan titik pada gambar. Nilai matriks yang terdapat pada gambar mewakili kecerahan dari titik tersebut. Titik itu mendominasi suatu gambar yang disebut elemen gambar atau piksel. Pencitraan digital bisa digunakan untuk kondisi menganalisis fungsi $f(x,y)$ dengan x dan y sebagai koordinat ruang [3].

2.3 Logika Fuzzy

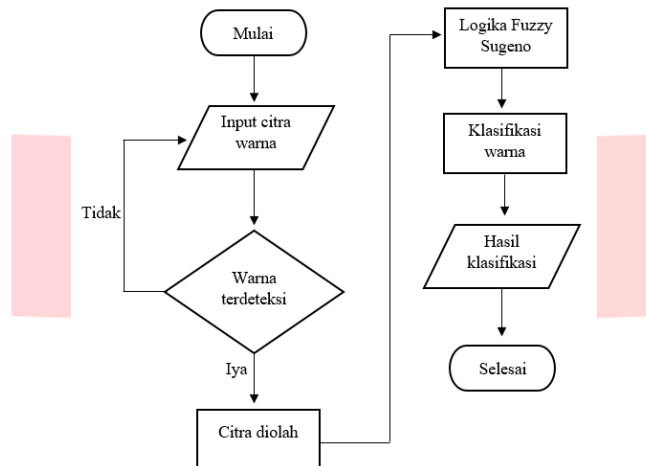
Logika *Fuzzy* adalah cara yang benar untuk menggunakan seperti pemetaan *input* dan pemetaan *output*. Penerapan sistem *fuzzy* didasarkan pada aturan-aturan atau pengetahuan, sistem *fuzzy* berdasarkan pada aturan *if-then* [3]. Berikut komponen umum sistem berbasis menggunakan aturan *fuzzy* terdiri tiga komponen, yaitu:

- *Fuzzification* yakni mengubah masukan nilai kebenaran yang sifatnya pasti dengan nilai *crisp input* ke dalam *fuzzy* [4].

- *Inference* yakni penalaran menggunakan *fuzzy rule* dan *fuzzy input* yang ditentukan sebelumnya sehingga menghasilkan nilai *fuzzy output* [5].
- *Defuzzification* yakni himpunan tegas yang sesuai fungsi keanggotaan seperti yang ditentukan sebelumnya [6].

3. Pembahasan

3.1 Gambaran Umum Sistem

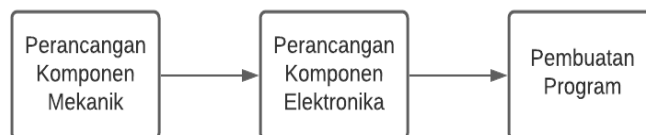


Gambar 1 Flowchart System klasifikasi warna

Secara umum sistem pengambilan warna yang terdapat pada *self-driving* menggunakan kamera *webcam* secara *real-time* dan setiap warna memiliki intruksi yang berbeda dengan menganalisis dua belas warna, yaitu Merah, Hijau, Kuning, Biru, Coklat, Merah marun, Oranye, Ungu, Abu-abu, Magenta, Biru muda, dan Putih. Pada gambar menjelaskan proses dari keseluruhan mulai dari pengambilan warna hingga pengolahan citra menjadi *grayscale*. Setelah itu proses klasifikasi warna dengan metode *Fuzzy Logic* dan mendapatkan *output* yang sesuai dengan intruksi.

3.2 Diagram Blok

Terdapat tiga tahapan dalam pengimplementasian pada robot *self-driving* berbasis pengenalan warna *RGB*, pada tahapan pertama yaitu merancang mekanik untuk robot, tahapan kedua merancang komponen elektronika robot, kemudian tahap ketiga membuat program yang sesuai untuk diterapkan pada robot.



Gambar 2 Diagram blok

3.3 Analisa Pengenalan Warna

Berdasarkan penelitian ini menggunakan warna yang sudah tercetak dan akan dibaca melalui kamera *webcam*. Warna yang digunakan terdapat dua belas jenis yakni warna merah, hijau, biru, kuning, coklat, merah marun, oranye, ungu, abu-abu, magenta, biru muda, dan putih. Berikut adalah gambar warna yang digunakan dalam penelitian *self-driving*. Citra warna akan direkam dengan kamera *webcam*, kemudian diproses sehingga mendapatkan intruksi yang

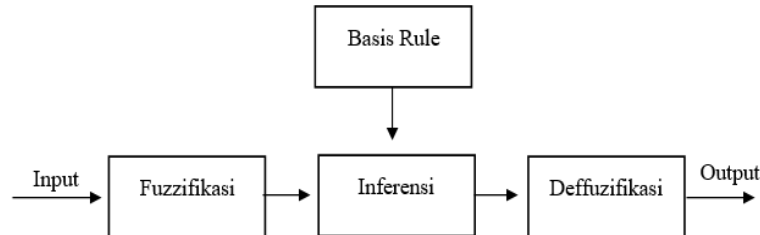
sesuai dengan program. Robot *self-driving* akan memenuhi pengambilan keputusan jika warna sesuai, terdapat dua belas intruksi dari masing-masing warna.

Tabel 1 Intruksi robot dan warna

No	Warna	Respon	Kecepatan Robot
1	Merah	Berhenti	0 PWM
2	Kuning	Maju lambat	20 PWM
3	Hijau	Maju dipercepat	40 PWM
4	Biru	Maju dipercepat	60 PWM
5	Coklat	Maju dipercepat	80 PWM
6	Merah marun	Maju dipercepat	100 PWM
7	Oranye	Maju dipercepat	120 PWM
8	Ungu	Maju dipercepat	140 PWM
9	Abu-abu	Maju dipercepat	160 PWM
10	Megenta	Maju dipercepat	180 PWM
11	Biru muda	Maju dipercepat	200 PWM
12	Putih	Maju maksimal	220 PWM

3.4 Representasi Keanggotaan *Fuzzy*

Penelitian *self-driving* menggunakan *fuzzy logic* dalam menentukan *rule* pada setiap warna. Ketika citra warna selesai diolah maka akan lanjut ke proses pengambilan keputusan, untuk pengambilan keputusan berikut adalah proses logika intruksi [7].



Gambar 3 Proses *Fuzzy Logic*

4. Hasil dan Pembahasan

4.1 Pengujian Perangkat Keras Kamera

Pengujian perangkat keras kamera bertujuan untuk mengetahui perangkat kamera yang digunakan berfungsi dengan baik. Tahap pengujian ini dilakukan dengan menguji kamera *USB Webcam* untuk mendapatkan warna yang sesuai. Warna yang diuji pada tahap ini yaitu, merah, hijau, biru, kuning, coklat, merah marun, oranye, ungu, abu-abu, megenta, biru muda dan putih. Pengujian menggunakan MATLAB dengan mengetahui perbedaan nilai *RGB* antara pengujian menggunakan kamera dengan nilai *RGB* asli. Hasil pengujian memperoleh perbandingan pengambilan warna yang sudah ditentukan dengan pengambilan warna yang didapatkan oleh kamera. Berikut data pengujian kamera pada dua belas warna.

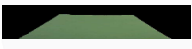
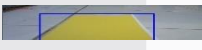
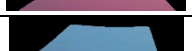
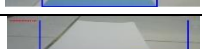
Tabel 2 Hasil pengujian kamera terhadap warna

Warna	Nilai Citra			Hasil Menggunakan Kamera			Keterangan
	R	G	B	R	G	B	
Merah	255	0	0	185	105	85	Nilai citra berbeda
Hijau	0	255	0	110	140	90	Nilai citra berbeda
Biru	0	0	255	80	95	145	Nilai citra berbeda
Kuning	255	255	255	190	175	70	Nilai citra berbeda
Coklat	150	75	0	115	100	85	Nilai citra berbeda
Merah- marun	128	0	0	115	100	85	Nilai citra berbeda
Oranye	255	127	0	170	120	80	Nilai citra berbeda
Ungu	191	0	255	125	100	135	Nilai citra berbeda
Abu-abu	128	128	128	125	125	125	Nilai citra berbeda
Magenta	255	0	255	165	95	125	Nilai citra berbeda
Biru- muda	0	255	255	100	150	185	Nilai citra berbeda
Putih	255	255	255	185	190	185	Nilai citra berbeda

4.2 Pengujian Pendeteksian Warna

Pendeteksian warna bertujuan untuk mendapatkan warna yang sesuai dari beberapa warna yang ada. Menggunakan aplikasi *pycharm* beserta *library open cv2* sehingga mendapatkan gambar atau warna yang sesuai dengan pencarian. Pengambilan warna melalui kamera secara *realtime*, akan dipengaruhi dengan kondisi ruangan atau pencahayaan. Kamera dapat mengambil warna secara akurat dengan mengatur nilai *HSV*. Pengujian dilakukan dengan mendeteksi warna objek dan memberi nama serta label pada warna target. Pemberian label menggunakan *ground truth* memungkinkan kamera lebih fokus pada target didepannya. Berikut pengujian pendeteksian warna melalui kamera secara *realtime*.

Tabel 3 Hasil pengujian pendeteksian warna

Warna Citra	Warna Citra	Ground Truth	Keterangan
Merah			Terdeteksi
Hijau			Terdeteksi
Biru			Terdeteksi
Kuning			Terdeteksi sama
Coklat			Terdeteksi sama
Merah marun			Terdeteksi sama
Oranye			Terdeteksi sama
Ungu			Terdeteksi sama
Abu-abu			Tidak terdeteksi, warna objek selaras dengan warna lantai
Magenta			Terdeteksi sama
Biru muda			Terdeteksi sama,
Putih			Terdeteksi sedikit berbeda

4.3 Pengujian Konversi RGB ke HSV

Pengujian konversi ke *HSV* dilakukan untuk mengetahui objek warna yang akan dipilih, dengan metode segmentasi deteksi nilai *HSV* terdapat pemilihan warna dengan membentuk segmen yang diinginkan. Citra warna *RGB* yang didapat akan menjadi proses konversi ke *HSV*, setiap dimensi warna *HSV* akan ditentukan nilai toleransi yang kemudian, nilai toleransi tersebut akan menjadi proses *thresholding*. Hasil dari *thresholding* membentuk segmen area dengan warna yang diinginkan.

Proses *thresholding* untuk mengetahui noise pada citra warna yang akan dikonversi ke *HSV*. Selanjutnya, pengaturan *upper* dan *lower* dibutuhkan untuk mencari minimum *noise* yang tertera pada hasil konversi.

Tabel 4 Hasil pengujian konversi *RGB* ke *HSV*

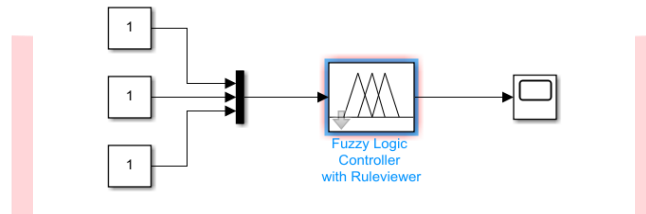
Uji Warna	<i>Thresholding</i>	Nilai <i>HSV</i>						Ket
		L-H	L-S	L-V	U-H	U-S	U-V	
Merah		0	38	114	15	159	188	Terdeteksi, terdapat <i>noise</i> sedikit
Hijau		45	56	117	80	225	255	Terdeteksi, terdapat <i>noise</i> sedikit
Biru		108	83	124	120	255	255	Terdeteksi, tidak ada <i>noise</i>
Kuning		20	120	130	255	255	255	Terdeteksi, tidak ada <i>noise</i>
Coklat		0	24	0	22	124	152	Terdeteksi, terdapat <i>noise</i> sedikit
Merah marun		0	24	85	24	128	152	Terdeteksi, terdapat <i>noise</i> sedikit
Oranye		12	97	75	22	153	225	Terdeteksi, tidak ada <i>noise</i>
Ungu		131	45	0	229	117	153	Terdeteksi, terdapat <i>noise</i> sedikit
Abu-abu		0	0	98	97	15	151	Terdeteksi, terdapat <i>noise</i> sedikit
Magenta		149	63	121	213	255	255	Terdeteksi, tidak ada <i>noise</i>
Biru muda		86	67	157	117	255	255	Terdeteksi, terdapat <i>noise</i> sedikit
Putih		0	0	163	167	45	255	Terdeteksi, ada <i>noise</i> sekitar objek

Keterangan:

L-H : Lower Hue **L-V** : Lower Value **U-S** : Upper Satue
L-S : Lower Satue **U-H** : Upper Hue **U-V** : Upper Value

4.4 Pengujian Perhitungan Manual

Pengujian perhitungan manual *fuzzy* bertujuan untuk memperoleh hasil yang sama dengan program yang dibuat. Perhitungan manual *Fuzzy Mamdani* akan dilakukan dengan *software* MATLAB, dengan perintah '*fuzzy*' yang diketikan pada *terminal* akan menampilkan *Fuzzy Inference System Editor*.



Gambar 4 Hasil *Export Fuzzy Inference System Editor*

4.5 Pengujian Intensitas Cahaya

Pengujian dengan intensitas cahaya sebesar (30 Lux) menghasilkan akurasi lebih besar dengan total 100%. Pengukuran intensitas cahaya menggunakan aplikasi *Light Meter* pada *smartphone* yang tersedia pada *App store*. Dari hasil pengujian bahwa intensitas cahaya yang mempengaruhi ruangan berpengaruh pada citra warna, semakin rendah intensitas maka warna yang dideteksi semakin lemah, namun sebaliknya semakin tinggi intensitas cahaya maka warna yang dideteksi semakin terlihat.

Tabel 5 Hasil pengujian konversi *RGB* ke *HSV*

Skema Intensitas Cahaya	Akurasi
5 lux	70%
15 lux	87%
30 lux	100%
≥ 50 lux	81%

5. Kesimpulan

Setelah Kesimpulan yang didapat dari penelitian tugas akhir ini yaitu:

1. Proses pengujian robot *self-driving* secara *real-time* dengan klasifikasi 12 warna dan intensitas cahaya pada ruangan yaitu 30 Lux, mendapatkan hasil akurasi yang paling tertinggi sebesar 100%.
2. Pengujian deteksi terhadap intensitas cahaya rendah akan menimbulkan objek citra warna yang lain dapat terdeteksi dan merekam 2 hingga 5 citra warna secara *real-time*.
3. Menurut perbandingan dengan 2 citra warna yang memiliki beda nilai *RGB*, kamera tetap bisa mendeteksi kedua warna meskipun menggunakan nilai *HSV* yang sama.
4. Pengujian pada konversi *RGB* ke *HSV*, terdapat noise pada citra warna putih, dan warna ruangan sekitar dapat berpengaruh pada konversi *HSV*.
5. Tahap implementasi membuktikan bahwa robot tidak memberikan *output* pada warna abu-abu, dikarenakan warna tersebut terpengaruh dengan warna lantai sekitar, dan robot memberikan *output*

kecepatan yang sama dengan warna yang hampir serupa yaitu biru muda dengan biru serta coklat dengan merah marun.

6. Hasil dari 3 percobaan klasifikasi warna jalur, robot mampu bergerak pada jalur yang bewarna selain warna *background* seperti warna lantai.
7. Akurasi pada robot saat mengikuti jalur lurus tidak sesuai dikarenakan robot tidak menggunakan *motor stepper*, serta tidak ada halangan dibagian kanan kiri jalur sehingga robot dapat keluar jalur.
8. Roda robot tidak dapat bergerak lurus kedepan secara presisi dikarenakan terdapat kecacatan bentuk roda antara roda kanan dan kiri.

Referensi

- [1] A. R. Chaidir, K. Anam, and G. A. Rahardi, "Lane Tracking pada Robot Beroda Holonomic menggunakan Pengolahan Citra," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 8, no. 1, p. 69, Jan. 2020, doi: 10.26760/elkomika.v8i1.69.
- [2] I. N. P. Mukhti, D. Suwandi, M. Si, and H. Bethaningtyas, "SISTEM OTOMASI DALAM PENYORTIRAN TOMAT DENGAN IMAGE PROCESSING MENGGUNAKAN METODE DETEKSI RGB," p. 8.
- [3] L. B. Prianggodo, "PERANCANGAN OBJECT TRACKING ROBOT BERBASIS IMAGE PROCESSING MENGGUNAKAN RASPBERRY PI," p. 23.
- [4] H. Novianto and D. Cahyono, "Sistem Pintu Air Otomatis Menggunakan Logika Fuzzy," p. 6, 2016.
- [5] R. A. Firmansyah, "Pembuatan Robot Penjejak Garis Berbasis Visual Menggunakan Fuzzy logic controller," p. 6.
- [6] A. Putri and F. Maspiyanti, "ROBOT LINE FOLLOWER PENGANTAR SURAT MENGGUNAKAN METODE FUZZY LOGIC STUDI KASUS FAKULTAS TEKNIK UNIVERSITAS PANCASILA," vol. 3, no. 1, p. 8, 2017.
- [8] R. Setiowati and N. C. Basjaruddin, "Active Steering Assistane For Turned Road Based On Fuzzy Logic," p. 7, 2017.