



## Rancang Bangun Robot Penghindar Halangan Berbasis Kamera Menggunakan Deteksi Kontur

Koko Joni<sup>1</sup>, Zainal Abidin<sup>2</sup>, Achmad Fiqhi Ibadillah<sup>3</sup>

<sup>1,2,3</sup> Teknik Elektro, Universitas Trunojoyo Madura

<sup>1,2,3</sup> Jl. Raya Telang PO BOX 2 Kamal, Madura

Email korespondensi: [zainal.te.utm@gmail.com](mailto:zainal.te.utm@gmail.com)

Dikirim 31 Juli 2017, Direvisi 17 Agustus 2017, Diterima 19 Agustus 2017

**Abstrak** – Robot penghindar halangan pada umumnya dibangun dengan menggunakan acuan dari beberapa sensor jarak yang diletakkan pada badan robot, kelemahan dari sistem ini adalah robot hanya dapat menghindari halangan tanpa mengetahui posisi target berada sehingga untuk kasus yang kompleks penggunaan sensor jarak kurang relevan diterapkan pada robot penghindar halangan. Solusi yang ditawarkan pada penelitian ini adalah dibangun sebuah robot penghindar halangan berbasis kamera yang diletakkan 2 meter di atas arena sehingga posisi robot, halangan, dan target dapat diketahui secara *real time*. Citra dari kamera diolah oleh sistem *vision*, *output*-nya berupa sebuah karakter perintah pergerakan motor yang dikirimkan pada robot melalui komunikasi *bluetooth*. Deteksi kontur digunakan untuk membedakan objek yang dipilih dengan gangguan di sekitar. Dari hasil pengujian, sistem kalibrasi *range* warna HSV yang dikembangkan mampu menghasilkan *range* warna akurat untuk *color filtering* menggunakan metode *auto-click* dan *tuning-manual*. Kalibrasi pengujian jarak sebenarnya berdasarkan jarak piksel menghasilkan *trend* grafik yang *linier* sehingga persamaan *linier*-nya dapat digunakan untuk konversi jarak piksel menjadi jarak sebenarnya dalam satuan centimeter. Hasil pengujian laju robot, sistem yang dibangun dapat membuat robot berjalan mendekati target dan menghindari halangan di depannya dengan baik, menggunakan metode *confusion matrix* didapat nilai akurasi sebesar 95% dari 30 percobaan yang telah dilakukan.

**Kata kunci** – Robot Penghindar Halangan, Kamera, Deteksi Kontur

**Abstract** -The obstacle avoidance robot is generally built using the reference of some distance sensors placed on the robot body, the weakness of this system is that robots can only avoid obstacles without knowing where the target is located, so for complex cases is not relevant using distance sensor applied to the obstacle avoidance robot. The solution offered in this research develops a camera-based obstacle avoidance robot are placed 2 meters above the arena. So that the position of robot, obstacles, and target can be known in real time. The image of the camera will be processed using a vision system, the output is a character command for movement of a DC motor that is sent on the robot through Bluetooth communication. Contour detection is used to distinguish selected objects with surrounding disturbances. From the test results, the HSV range calibration system developed has been able to produce accurate color ranges for color filtering, using auto-click and tuning manual methods. The actual distance testing calibration is based on pixel distance produces a correlation graph trend, so the linear equations can be used to convert pixel distance to actual distance in centimeters. The robot test results, developed system can make the robot move close to the target and avoid obstacles in front of the robot well, using confusion matrix method obtained accuracy value 95% of 30 experiments that have been done.

**Keywords** - Obstacle Avoidance Robot; Camera; Contour Detection

### I. PENDAHULUAN

Robot merupakan suatu mekanik yang dapat diperintah secara otomatis untuk memproses suatu benda agar dapat bekerja sesuai dengan permintaan tertentu sehingga dapat meringankan pekerjaan

manusia [1], salah satu jenisnya adalah robot penghindar halangan. Sistem pendeteksi pada robot penghindar halangan pada umumnya menggunakan sensor jarak sebagai acuan dalam menghindari halangan [2]. Kelemahan dari robot penghindar

halangan berbasis sensor jarak adalah robot hanya mampu menghindari halangan disekitarnya tanpa mengetahui posisi target yang dituju. Sehingga diperlukan beberapa sensor tambahan agar robot dapat mengenali dan mengetahui target yang dituju. Untuk kasus yang kompleks penggunaan sensor jarak saja kurang relevan diterapkan pada robot penghindar halangan. Oleh sebab itu menjadi topik yang menarik untuk mengembangkan rancang bangun robot penghindar halangan berbasis kamera menggunakan deteksi kontur untuk identifikasi warna objek.

Telah banyak penelitian maupun artikel yang membahas tentang robot penghindar halangan, dan berikut sumber acuan yang mendukung penelitian ini:

Penelitian tentang robot penghindar halangan menggunakan 3 buah sensor ultrasonik yang diletakkan di bagian depan dan samping robot. Penelitian ini menggunakan metode PID untuk mengolah data dari sensor menjadi kontrol kecepatan pada motor DC Brushless. Hasil penelitian, robot mampu menghindari halangan dengan baik. Kelemahan dari sistem ini adalah robot hanya mampu menghindari halangan di depannya tanpa mengetahui posisi target yang dituju [2].

Penelitian lainnya mengembangkan robot penghindar halangan menggunakan kamera yang diletakkan di badan robot menghadap ke depan. Kendali PID digunakan untuk memuluskan pergerakan robot saat menghindari halangan. Hasil penelitian, robot mampu menghindari halangan dengan halus. Kelemahan dari sistem ini adalah jangkauan penglihatan robot sangat terbatas pada lebar *capture* kamera. Ketika halangan terlepas dari jangkauan kamera, robot akan maju terus tanpa henti [1].

Penelitian selanjutnya mengembangkan *robot soccer small size* dengan kamera yang diletakkan di atas lapangan, sehingga posisi robot, target dan halangan dapat diketahui secara *real-time*. Hasil penelitian robot mampu berjalan mendekati target berupa bola dan menghindari satu halangan berupa robot lawan. Kelemahan dari sistem yang dikembangkan adalah waktu respon robot mendeteksi sampai selesai menghindari halangan terbilang lama, yaitu sekitar 7 detik [3].

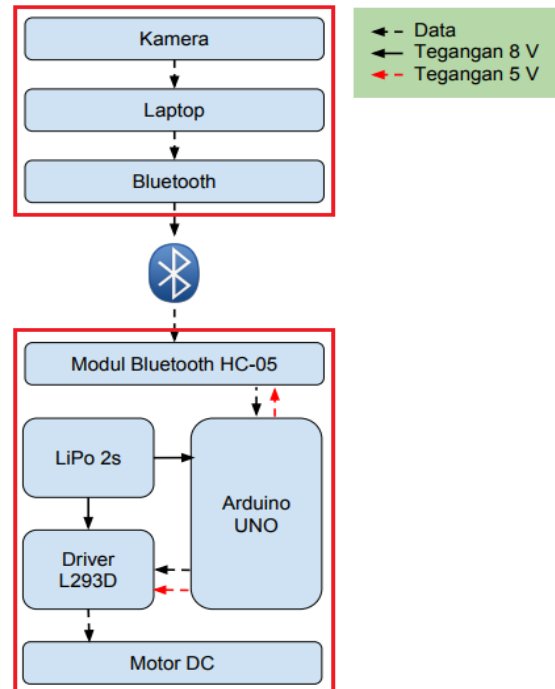
Pada penelitian ini kamera diletakkan di atas arena uji, sehingga posisi robot, target dan halangan dapat diketahui secara *real-time*. Data dari kamera akan diproses oleh sistem vision dengan output berupa karakter perintah pergerakan yang dikirim ke robot melalui komunikasi *bluetooth*. Harapannya hasil penelitian ini nantinya dapat dikembangkan menjadi proyek yang lebih kompleks, seperti robot sepak bola atau robot pemindah barang otomatis dalam suatu gudang.

## II. METODE PENELITIAN

Metodologi penelitian meliputi perancangan perangkat keras, perancangan sistem vision, strategi pergerakan robot, dan metode pengujian.

### A. Perancangan Perangkat Keras

Secara garis besar diagram kerja robot penghindar halangan, seperti pada Gambar 1.

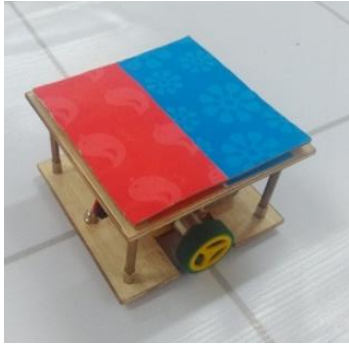


Gambar 1. Diagram Kerja Robot Penghindar Halangan

Pada Gambar 1, secara umum sistem pada robot penghindar halangan dibagi menjadi dua, yaitu sistem *vision* dan robot. Sistem *vision* terdiri dari kamera dengan lensa 0,67x *wide* yang diletakkan 2 meter di atas arena, dan laptop yang dilengkapi dengan aplikasi Codeblocks sebagai *compiler* dan OpenCV sebagai pustaka untuk pengolahan citra. Output dari sistem *vision* menghasilkan perintah pergerakan robot yang dikirim via *Bluetooth*.

Pada bagian robot dilengkapi dengan modul bluetooth HC-05 untuk menerima perintah dari sistem vision, mikrokontroler yang digunakan adalah Arduino UNO yang dilengkapi dengan Driver Motor L293D untuk memproses data dan mengontrol motor DC. *Supply* robot menggunakan baterai LiPo 2s dengan tegangan 8,40 V 1000 mAh.

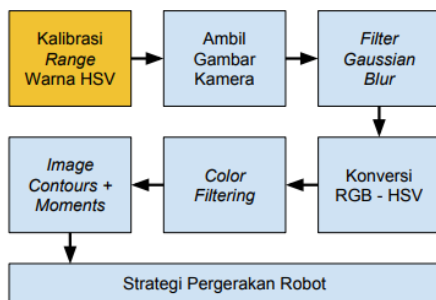
Rangka robot terbuat dari papan kayu tipis dengan dimensi keseluruhan robot adalah 11,5 cm x 11,5 cm x 8 cm. pada bagian atas robot diberi stiker dengan 2 warna, yaitu merah untuk bagian depan dan biru untuk bagian belakang robot. Tampilan robot seperti pada Gambar 2.



Gambar 2. Tampilan Robot Penghindar Halangan

### B. Perancangan Sistem Vision

Berikut merupakan diagram blok sistem *vision* untuk mendeteksi masing-masing objek dilapangan yang diterapkan pada robot penghindar halangan, seperti pada Gambar 3.



Gambar 3. Diagram Blok Sistem Vision Robot Penghindar Halangan

Sebelum dilakukan proses pengolahan citra dilakukan kalibrasi rentang warna HSV dari masing-masing warna objek yang digunakan, kalibrasi dilakukan dengan metode *auto-click* dan *tunning-manual*. Untuk metode *auto-click* menggunakan algoritma rerata dari area yang dipilih, seperti pada Persamaan 1 [4], kemudian dijumlahkan dan dikurangi dengan konstanta *ColorRadius* untuk mendapatkan nilai *High* dan *Low* HSV.

$$\text{rerata}(y, x) = \frac{1}{nm} \sum_{p=-n}^n \sum_{q=-m}^m f(y + p, x + q) \quad (1)$$

$$\text{Low}(HSV) = \text{rerata}(y, x) - \text{ColorRadius}(HSV) \quad (2)$$

$$\text{High}(HSV) = \text{rerata}(y, x) + \text{ColorRadius}(HSV) \quad (3)$$

Keterangan:

- $\text{rerata}(x, y)$  : Nilai rerata piksel yang di klik pada area  $(x, y)$
- $n$  : Lebar ketetangaan sumbu  $y$
- $m$  : Lebar ketetangaan sumbu  $x$
- $p, q$  : Variabel sementara
- $\text{ColorRadius}(HSV)$  : Konstanta radius *range* warna masing-masing HSV
- $\text{Low}(HSV)$  : Hasil batas bawah warna masing-masing HSV
- $\text{High}(HSV)$  : Hasil batas atas warna masing-masing HSV

Terkadang citra yang dihasilkan dari metode *auto-click* masih menimbulkan derau, maka perlu dikompensasi dengan metode *tunning-manual*, dengan sedikit menambah atau mengurangi nilai  $H, S$  atau  $V$ .

*Gaussian blur* adalah salah satu *filter blur* yang menggunakan rumus matematika untuk mengurangi detail dan menciptakan efek berkabut [5], metode ini digunakan untuk menyamarkan derau pada citra ketika dilakukan *color filtering*. Pada penelitian ini digunakan *Gaussian blur* dua dimensi dengan konstanta nilai  $\sigma = 1$  dan ukuran matriks kernel sebesar  $5 \times 5$ . Berikut perintah menggunakan *Gaussian Blur* pada C++ OpenCV.

```
GaussianBlur(inFrame, outFrame, Size(5,5), 1, 0);
```

*Output* citra dari kamera berupa RGB, selanjutnya dilakukan konversi dari RGB menjadi HSV, HSV merupakan ruang warna yang sangat cocok untuk mengidentifikasi warna-warna dasar, dimana warna dasar ini digunakan dalam penelitian sebagai warna identifikasi robot. Selain itu, HSV menoleransi terhadap perubahan intensitas cahaya. Inilah yang menjadi keunggulan HSV dibandingkan dengan ruang warna lainnya [2]. Berikut perintah konversi RGB menjadi HSV pada C++ OpenCV.

```
cvtColor(rgbFrame, hsvFrame, CV_RGB2HSV);
```

Selanjutnya dilakukan proses *color filtering*, yaitu suatu teknik untuk memfilter atau mempertahankan warna citra berdasarkan dua batas warna yang telah ditentukan, sedangkan untuk warna yang lain akan diubah menjadi warna hitam [6]. *Thresholding* merupakan metode yang efektif pada citra menunjukkan pemisah yang jelas antara dua *cluster* pada histogram sehingga proses pemisahan objek dan *background* mudah dilakukan [7]. Proses *thresholding* dilakukan untuk memisahkan warna citra menjadi 2 bagian (terang dan gelap) untuk mendapatkan citra dengan warna yang kontras. Berikut perintah penerapan *color filtering* dan *thresholding* pada C++ OpenCV.

```
inRange(hsvFrame, LowRange, HighRange, resultImg);
```

*Contours* merupakan salah satu metode dalam pengolahan citra yang berguna untuk analisis bentuk dan pengenalan objek, untuk akurasi terbaik menggunakan citra *binary* yang dihasilkan dari metode *threshold*. *Contours* juga dapat digunakan untuk menghitung berapa banyak jumlah objek dalam sebuah frame menggunakan fungsi *findContours* [8]. Pada penelitian ini metode deteksi kontur digunakan untuk membedakan objek yang dipilih dari objek gangguan berdasarkan minimum kontur-area yang ditentukan, sehingga penentuan objek akan lebih akurat. Selanjutnya untuk mengetahui titik tengah dari sebuah objek terpilih, digunakan fungsi *contour-moments*. Titik pusat objek ini akan digunakan untuk mengetahui posisi dari robot, target dan halangan. Berikut penerapan *Contours* dan *Moments* pada C++ OpenCV.

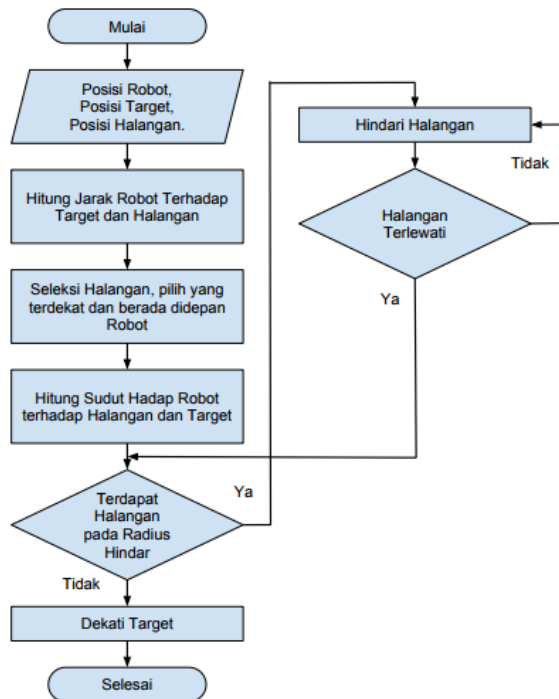
```

findContours(resultImg, contours, Hierarchy,
CV_RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE);
for(unsigned int i=0; i<contours.size(); i++){
    double a=contourArea(contours[i],false);
    if (a > 1000){IndexImg = i;}
}
Moments mu = moments(contours[IndexImg], false);
centerObject.x = mu.m10/mu.m00;
centerObject.y = mu.m01/mu.m00;

```

### C. Strategi Pergerakan Robot

Berikut merupakan *flowchart* strategi pergerakan robot penghindar halangan, seperti pada Gambar 4 di bawah ini.



Gambar 4. *Flowchart* Strategi Pergerakan Robot Penghindar Halangan

Posisi robot, halangan dan target dapat diketahui dari sistem vision, selanjutnya data posisi tersebut diolah untuk mendapatkan data jarak dari robot terhadap target dan masing-masing halangan, data jarak didapat dari Persamaan 4 [9], yaitu menghitung panjang dua buah vektor.

$$d_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4)$$

Keterangan:

$d_{12}$  : Jarak Robot terhadap Objek

$x_1, y_1$  : Posisi  $(x, y)$  Robot

$x_2, y_2$  : Posisi  $(x, y)$  Objek

Selanjutnya dilakukan seleksi dari beberapa halangan yang ada, hanya halangan dengan jarak terdekat dengan robot dan berada di antara robot dan target yang akan diproses atau ditandai.

Proses selanjutnya adalah menghitung arah hadap robot, sebelumnya dilakukan perhitungan untuk mengetahui posisi tengah robot berdasarkan dua titik

pusat dari warna stiker pada badan robot menggunakan persamaan untuk mencari nilai rata-rata, sebagaimana Persamaan 5 dan 6.

$$center_x = \frac{x_{merah} + x_{biru}}{2} \quad (5)$$

$$center_y = \frac{y_{merah} + y_{biru}}{2} \quad (6)$$

Untuk mengetahui arah hadap robot menggunakan pengembangan dari persamaan trigonometri berdasarkan sumbu  $x$  citra dengan input berupa titik pusat robot terhadap titik pusat stiker merah, seperti pada persamaan 7 [10], *output* dari persamaan ini berupa sudut derajat terhadap sumbu  $x$ , 0 sampai 180 derajat untuk arah putaran CW dan 0 sampai -180 derajat untuk arah putaran CCW.

$$deg = atan\left(\frac{x_2 - x_1}{y_2 - y_1}\right) * \frac{180}{\pi} \quad (7)$$

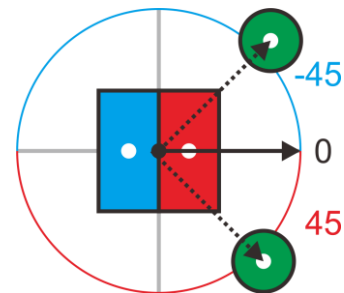
Keterangan:

$deg$  : Sudut Hadap Robot

$x_1, y_1$  : Posisi  $(x, y)$  Pusat Robot

$x_2, y_2$  : Posisi  $(x, y)$  Pusat Stiker Merah

Persamaan 7 juga dapat digunakan untuk mengetahui sudut robot terhadap objek (target maupun halangan) dengan mengubah  $x_2, y_2$  menjadi titik pusat objek. Dari perbedaan nilai sudut hadap robot dan objek dapat diketahui letak objek tersebut terhadap robot, jika nilai sudut objek lebih besar dari pada sudut hadap robot maka objek berada disebelah kanan robot, begitu pula sebaliknya. Ilustrasi perbedaan sudut untuk mengetahui posisi objek terhadap robot seperti pada Gambar 5.



Gambar 5. Ilustrasi Perbedaan Sudut Untuk Mengetahui Letak Objek Terhadap Robot

Strategi pergerakan robot mendekati target adalah mempertahankan selisih sudut  $0^\circ$  antara sudut hadap robot dan sudut robot terhadap target, ketika selisih sudut mendekati nilai  $0^\circ$ , maka robot akan maju mendekati target sampai radius henti robot. Jika terdapat halangan di depan robot, maka robot akan menghindari dengan cara mempertahankan selisih sudut sebesar  $90^\circ$  terhadap halangan.

Data yang dikirim dari laptop kepada robot berupa data serial dengan panjang 2 karakter, seperti " $x\n$ " dengan " $x$ " adalah perintah dan " $\n$ " adalah Enter. Selanjutnya data tersebut akan diterima dan diproses

oleh Arduino sebagai acuan untuk menggerakkan motor DC.

Tabel 1. Daftar Karakter Perintah Pada Robot

Karakter	PWM Motor Kiri	PWM Motor Kanan	Keterangan
'1'	-60	80	Pivot Kiri
'2'	65	80	Serong kiri 3
'3'	70	80	Serong kiri 2
'4'	75	80	Serong kiri 1
'5'	90	90	Maju
'6'	80	75	Serong kanan 1
'7'	80	70	Serong kanan 2
'8'	80	65	Serong kanan 3
'9'	80	-60	Pivot Kanan
'0'	0	0	Berhenti

Penggunaan metode pengiriman data perintah menggunakan karakter tunggal dapat memperjelas proses pengiriman dan pengolahan data, akan tetapi variasi perintah menjadi terbatas serta diperlukan penentuan pembobotan nilai karakter perintah sebelumnya.

#### D. Metode Pengujian

Secara umum pengujian yang dilakukan meliputi uji kalibrasi *range* warna HSV, uji kalibrasi jarak berdasarkan piksel, uji laju robot tanpa halangan dan uji laju robot dengan halangan.

Untuk uji kalibrasi rentang warna HSV menggunakan program kalibrasi dengan metode *auto-click*, jika masih terdapat derau, maka dikompensasi menggunakan metode *tunning-manual* sampai menghasilkan *range* warna yang tepat untuk 6 warna yang berbeda-beda.

Kalibrasi jarak dilakukan untuk mengetahui jarak sebenarnya (dalam satuan cm) robot terhadap target berdasarkan nilai jarak piksel. Pengujian akan dilakukan dari jarak 10 cm – 150 cm. Hasil pengujian akan diplot menggunakan aplikasi *microsoft excel* dan dianalisa, jika menghasilkan grafik yang linier antara nilai piksel dan jarak sebenarnya maka persamaan linier yang dihasilkan dapat digunakan untuk mengetahui jarak sebenarnya pada nilai piksel yang berbeda.

Pengujian selanjutnya merupakan uji laju robot tanpa halangan dan dengan halangan. Uji laju robot tanpa halangan dilakukan dengan memvariasi sudut hadap awal robot ( $0^\circ$ ,  $90^\circ$ , dan  $180^\circ$ ) terhadap target. Jika berhasil maka akan dilanjutkan dengan menambahkan halangan secara berkala (1, 3, dan 6 buah) di depan robot dengan posisi acak dan jarak antara halangan sebesar dua kali lebar robot, seperti pada Gambar 6 dibawah ini. Masing-masing pengujian akan diulang sebanyak 10 kali untuk mengetahui kehandalan sistem dan strategi yang telah dibangun. Halangan yang digunakan adalah plat berbentuk kotak dengan dimensi 6 cm x 6 cm yang diberi stiker warna berbeda-beda, untuk target menggunakan bahan yang sama dengan dimensi 9 cm x 9 cm.



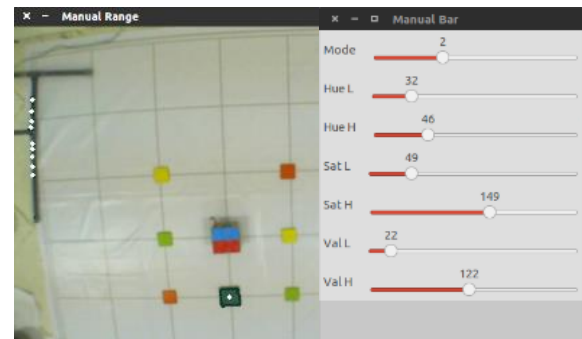
Gambar 6. Pengujian Laju Robot Dengan Posisi Halangan Acak

### III. HASIL PENELITIAN

Berikut merupakan serangkaian hasil uji dari penelitian yang telah dilakukan.

#### A. Uji Kalibrasi Range Warna HSV

Uji kalibrasi *range* warna HSV menggunakan sistem kalibrasi yang telah dikembangkan dengan metode *auto-click* dan *tunning-manual*, tampilan sistem kalibrasi seperti pada Gambar 7.



Gambar 7. Tampilan Sistem Kalibrasi Range Warna HSV

Dari hasil kalibrasi diperoleh nilai *range* warna HSV terbaik dari 6 warna objek yang berbeda-beda, hasilnya sebagaimana Tabel 2.

Tabel 2. Range Warna HSV Hasil Kalibrasi

Warna	Range LOW			Range HIGH			Metode
	H	S	V	H	S	V	
Merah	113	147	102	125	255	255	TM
Orange	102	191	107	109	255	255	TM
Biru	7	116	84	21	255	255	AC
Kuning	87	216	114	97	255	255	AC
Hijau Tua	35	75	57	50	255	255	TM
Hijau Muda	75	162	100	84	255	255	TM

Keterangan:

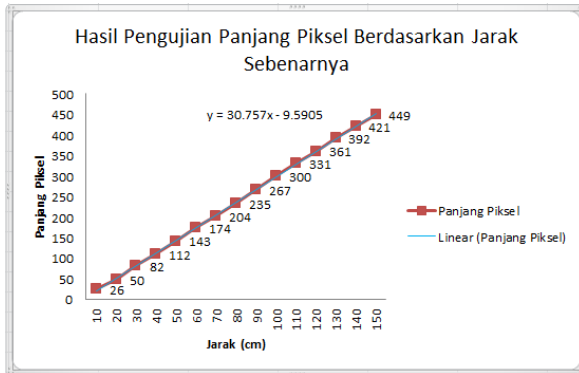
TM *Tunning-Manual*       AC *Auto-Click*

Dari data pada Tabel 2 dapat diketahui ada beberapa warna yang *range* warna HSV-nya didapat dari metode *auto-click* yaitu warna Biru dan Kuning. Sedangkan untuk Empat warna yang lain (Merah, Orange, Hijau Tua dan Hijau Muda) nilai *range* warna

HSV-nya diperoleh menggunakan metode *tunning-manual*.

**B. Uji Kalibrasi Jarak Berdasarkan Piksel**

Dari hasil uji kalibrasi jarak sebenarnya berdasarkan piksel menghasilkan data dengan grafik yang cukup linier, berikut grafik hasil pengujian panjang piksel terhadap jarak sebenarnya sebagaimana Gambar 8.

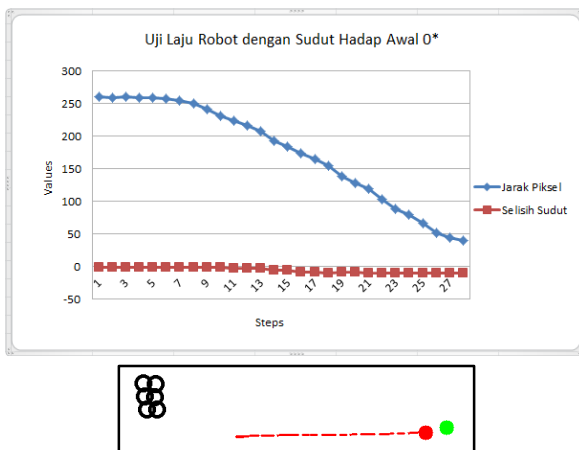


Gambar 8. Grafik Hasil Pengujian Panjang Piksel Berdasarkan Jarak Sebenarnya

Pada Gambar 8, dapat diketahui ketika jarak 10 cm menghasilkan panjang piksel sebesar 26, ketika jarak 20 cm menghasilkan panjang piksel 50, sampai pada pengujian jarak 150 cm menghasilkan nilai piksel dengan *trend* grafik yang linier, ini dibuktikan dengan sejajarnya garis linier terhadap nilai hasil pengujian.

**C. Uji Laju Robot Tanpa Halangan**

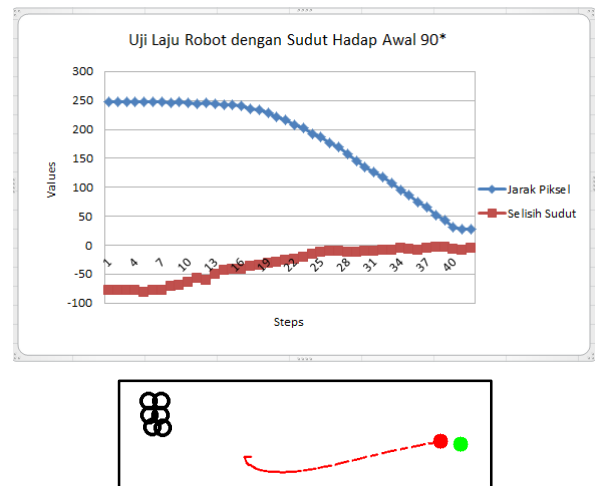
Berikut merupakan hasil uji laju robot menuju target tanpa halangan dengan variasi nilai sudut hadap awal robot 0°, 90°, dan 180°. Hasil uji berupa grafik jarak piksel dan selisih sudut terhadap langkah robot, serta trajektori robot sebagaimana Gambar 9 sampai Gambar 11.



Gambar 9. Grafik dan Trajektori Pengujian Laju Robot Dengan Sudut Hadap Awal 0°

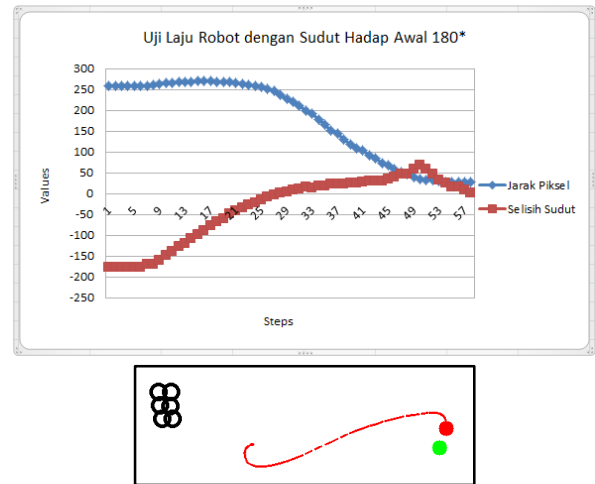
Pada pengujian dengan sudut hadap awal robot 0°, robot mampu mempertahankan nilai selisih sudut mendekati 0° terhadap target, selanjutnya robot

melakukan manuver maju mendekati target dan akhirnya berhenti.



Gambar 10. Grafik dan Trajektori Pengujian Laju Robot Dengan Sudut Hadap Awal 90°

Pada pengujian dengan sudut hadap awal robot 90°, mulanya robot melakukan manuver pivot berlawanan arah jarum jam, ini dibuktikan dengan grafik selisih sudut yang bergerak dari -90° menuju -45° dan jarak piksel yang tidak berubah. Selanjutnya robot berjalan mendekati target dan mempertahankan selisih sudut sebesar 0° hingga akhirnya robot berhenti.



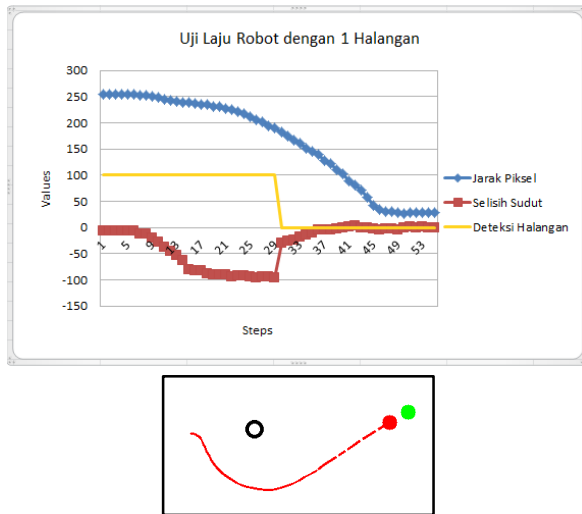
Gambar 11. Grafik dan Trajektori Pengujian Laju Robot Dengan Sudut Hadap Awal 180°

Pada pengujian dengan sudut hadap awal robot 180°, mulanya robot melakukan manuver *pivot* berlawanan arah jarum jam, ini dibuktikan dengan grafik selisih sudut yang bergerak dari -180° menuju -45° dan jarak piksel yang tidak berubah. Selanjutnya robot berjalan mendekati target, pergerakan robot sempat mengalami beresilasi, akan tetapi robot mampu memosisikan kembali menghadap target dan akhirnya berhenti.

**D. Uji Laju Robot Dengan Halangan**

Berikut merupakan hasil uji laju robot menuju target dengan variasi jumlah halangan di depan robot

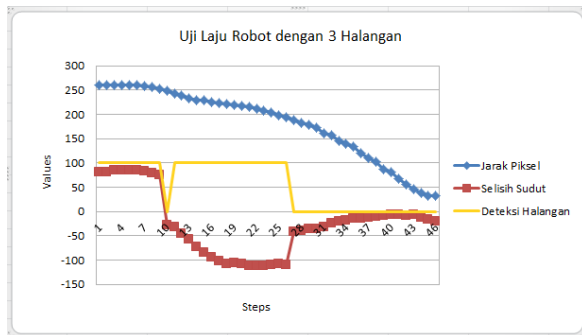
1, 3, dan 6 buah. Masing-masing variasi halangan dilakukan pengujian sebanyak 10 kali untuk mengetahui kehandalan sistem yang telah dibangun. Hasil uji berupa grafik jarak piksel, selisih sudut dan kondisi deteksi halangan (akan bernilai 100 jika halangan terdeteksi, jika tidak maka akan bernilai 0) terhadap langkah robot, serta trajektori robot sebagaimana Gambar 12 sampai Gambar 14.



Gambar 12. Grafik dan Trajektori Pengujian Laju Robot Dengan 1 Halangan Di Depan Robot

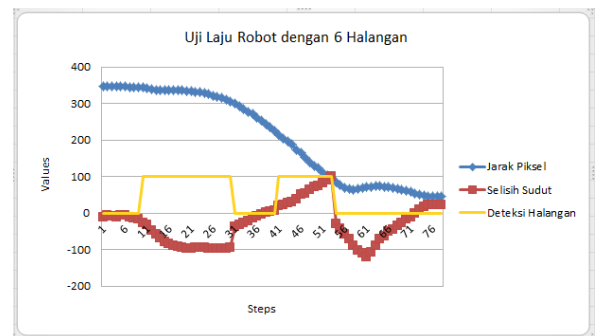
Pada pengujian dengan 1 buah halangan, mulanya robot mendeteksi halangan, ini dibuktikan dari grafik Deteksi Halangan yang bernilai 100, selanjutnya robot melakukan pergerakan menghindari halangan dengan berbelok ke kanan hingga selisih sudut terhadap halangan bernilai  $-90^\circ$ , lalu robot bergerak maju hingga halangan terlewati (grafik Deteksi Halangan bernilai 0). Selanjutnya robot bergerak menuju target dan akhirnya berhenti.

Grafik selisih sudut pada *steps* 30 menghasilkan perubahan yang signifikan, ini dikarenakan ketika terdapat halangan, selisih sudut yang ditampilkan mengacu pada halangan. Sebaliknya ketika halangan tidak terdeteksi selisih sudut yang ditampilkan mengacu pada target.



Gambar 13. Grafik Dan Trajektori Pengujian Laju Robot Dengan 3 Halangan Di Depan Robot

Pada pengujian dengan 3 buah halangan, mulanya robot mendeteksi halangan di sebelah kanan robot, ini dibuktikan dengan grafik Deteksi Halangan yang bernilai 100 dan selisih sudut yang bernilai  $90^\circ$ . Kemudian robot melaju dan mendeteksi halangan di depannya, lalu robot menghindarinya dengan berbelok ke kanan sampai sudut  $-90^\circ$  dan maju sampai halangan terlewati. Selanjutnya robot bergerak menuju target dan akhirnya berhenti.



Gambar 14. Grafik dan Trajektori Pengujian Laju Robot Dengan 6 Halangan Di Depan Robot

Pada pengujian dengan 6 buah halangan, robot mampu mendeteksi halangan pertama dan berbelok ke kanan, sedangkan untuk halangan yang kedua robot berbelok ke kiri karena arah hadap robot lebih cenderung menghadap sebelah kiri halangan. Selanjutnya robot bergerak menuju target meskipun sempat terjadi osilasi pada pergerakannya.

Berikut merupakan hasil dari pengujian secara keseluruhan dengan variasi jumlah halangan di depan robot, sebagaimana Tabel 3 di bawah ini.

Tabel 3. Hasil Pengujian Laju Robot Dengan Variasi Jumlah Halangan Di Depan Robot

Pengujian	Variasi		
	1 Halangan	3 Halangan	6 Halangan
Uji ke-1	Berhasil	Berhasil	Berhasil
Uji ke-2	Berhasil	Berhasil	Berhasil
Uji ke-3	Berhasil	Berhasil	Berhasil
Uji ke-4	Berhasil	Berhasil	Berhasil

Pengujian	Variasi		
	1 Halangan	3 Halangan	6 Halangan
Uji ke-5	Berhasil	Berhasil	Berhasil
Uji ke-6	Berhasil	Berhasil	Berhasil
Uji ke-7	Berhasil	Berhasil	Gagal
Uji ke-8	Berhasil	Gagal	Berhasil
Uji ke-9	Berhasil	Berhasil	Berhasil
Uji ke-10	Berhasil	Berhasil	Gagal

Dari hasil pengujian, robot sempat mengalami gagal laju dan menabrak halangan pada pengujian dengan variasi jumlah halangan di depan robot sebesar 3 dan 4 buah.

#### IV. PEMBAHASAN

Pada pengujian kalibrasi *range* warna HSV, data *range* warna dari Tabel 2 diterapkan pada sistem vision robot penghindar halangan, hasilnya semua objek dengan warna yang telah terkalibrasi dapat dideteksi dan ditandai oleh sistem, sebagaimana Gambar 15.



Gambar 15. Hasil Deteksi Objek yang Telah Terkalibrasi

Dari data pada Tabel 2 juga dapat diketahui bahwa, metode *auto-click* dapat diterapkan hanya pada warna yang mempunyai perbedaan yang mencolok, seperti warna kuning dan biru. Sedangkan untuk warna merah yang mempunyai kemiripan dengan warna *orange*, dan warna hijau tua yang mempunyai kemiripan dengan warna hijau muda, harus menggunakan metode *tuning-manual* untuk mendapatkan nilai *range* HSV yang tepat.

Dari grafik pengukuran jarak piksel terhadap jarak sebenarnya pada Gambar 8, menghasilkan *trend* grafik yang sangat linier, sehingga jarak piksel antara robot dan objek yang dihasilkan dari Persamaan 4 dapat dikonversi menjadi jarak sebenarnya dalam satuan *centimeter* menggunakan Persamaan 10 di bawah ini.

Hasil dari persamaan liner dari grafik adalah:

$$y = 30.757x - 9.5905 \quad (8)$$

Dengan  $y$  adalah jarak piksel dan  $x$  adalah jarak sebenarnya dalam satuan centimeter, karena data  $x$  bernilai kelipatan 10 sedangkan persamaan linier yang dihasilkan dari *microsoft excel* masih menggunakan kelipatan 1, sehingga 30.757 perlu dibagi 10.

$$\text{jarak}_{\text{piksel}} = 3.0757 * \text{jarak}_{\text{cm}} - 9.5905 \quad (9)$$

Sehingga ketika dicari nilai jarak sebenarnya menjadi.

$$\text{jarak}_{\text{cm}} = \frac{\text{jarak}_{\text{piksel}} + 9,5905}{3,0757} \quad (10)$$

Persamaan 10 hanya dapat digunakan pada penelitian ini saja dengan jenis kamera, dan lensa serta ketinggian penempatan kamera yang sama, selain itu diperlukan kalibrasi ulang untuk menentukan jarak sebenarnya berdasarkan panjang piksel.

Dari hasil pengujian laju robot tanpa halangan, robot mampu berjalan mendekati halangan dan mempertahankan nilai selisih sudut hadap robot terhadap sudut robot pada target sekitar  $0^\circ$ . Pada uji sudut hadap  $180^\circ$  robot sempat mengalami osilasi, tapi robot dapat kembali stabil menghadap target, ini dikarenakan sistem pergerakan robot tidak menggunakan metode khusus seperti PID untuk meredam osilasi.

Pada pengujian laju robot dengan halangan, robot mampu melaju mendekati target dan menghindari halangan dengan variasi jumlah halangan yang diletakkan secara acak di depan robot. Robot bermanuver belok ketika menemui halangan di depannya sampai selisih sudut sekitar  $90^\circ$ , lalu robot bermanuver maju sampai halangan terlewati, selanjutnya robot bergerak maju mendekati target. Dari hasil trajektori dapat dilihat robot mampu berjalan melewati dua halangan dengan jarak antara halangan sebesar dua kali lebar robot.

Dari data pada Tabel 3, diketahui dari 30 kali percobaan yang dilakukan, robot mengalami kegagalan laju dan menabrak halangan sebanyak 3 kali. Dengan menggunakan metode *confusion matrix* didapat presentasi keberhasilan program strategi laju robot sebesar:

Total Pengujian : 30, Berhasil : 27, Gagal 3.

$$\text{Precision} = \frac{\text{Berhasil}}{\text{Berhasil} + \text{Gagal}}$$

$$= \frac{27}{27 + 3} = 0.9 = 90\%$$

$$\text{Accuracy} = \frac{\text{Berhasil} + \text{Total}_p}{\text{Berhasil} + \text{Gagal} + \text{Total}_p}$$

$$= \frac{57}{27 + 3 + 30} = 0.95 = 95\%$$

$$\text{Error Rate} = \frac{\text{Gagal}}{\text{Berhasil} + \text{Gagal} + \text{Total}_p}$$

$$= \frac{3}{27 + 3 + 30} = 0.05 = 5\%$$

Secara keseluruhan robot berhasil mendekati target dan menghindari halangan dengan baik, dengan presentasi tingkat presisi sebesar 90%, akurasi sebesar 95% dan eror sebesar 5 % dari 30 kali pengujian yang telah dilakukan.



## V. PENUTUP

### A. Kesimpulan

Sistem kalibrasi range warna HSV yang dikembangkan mampu menghasilkan nilai yang cukup akurat. Metode *auto-click* dapat diterapkan pada objek dengan warna yang kontras dengan *background* sekitar, sedangkan metode *tuning-manual* dapat diterapkan pada segala kondisi warna.

Hasil kalibrasi pengujian jarak sebenarnya berdasarkan jarak piksel menghasilkan *trend* grafik yang linier, sehingga persamaan liniernya dapat digunakan untuk konversi jarak piksel menjadi jarak sebenarnya dalam satuan centimeter.

Hasil pengujian laju robot, sistem yang dibangun dapat membuat robot berjalan mendekati target dan menghindari halangan di depan robot dengan baik. Dengan menggunakan metode *confusion matrix* didapat presentasi tingkat presisi sebesar 90% dan akurasi 95% dari 30 percobaan yang telah dilakukan.

### B. Saran

Sistem *vision* yang dibuat sangat sensitif terhadap perubahan cahaya sehingga diperlukan kalibrasi ulang untuk pencahayaan yang berbeda, diperlukan penelitian lebih lanjut untuk membuat sistem *vision* yang lebih handal terhadap perubahan cahaya.

Konstruksi *hardware* juga mempengaruhi laju manuver robot, selanjutnya dapat dikembangkan menggunakan 3 buah roda omni atau lebih, untuk menghasilkan sistem manuver yang lebih leluasa.

Proses komputasi pengolahan citra terbilang cukup berat, sehingga diperlukan spesifikasi laptop atau PC yang tinggi untuk mendapatkan hasil komputasi yang lebih baik dan cepat.

Sistem laju robot dapat dikembangkan dengan metode peredam osilasi seperti PID, untuk mendapatkan hasil pergerakan yang lebih halus.

pembimbing tugas akhir yang telah mendukung dan memberikan bimbingan penuh sehingga penelitian ini selesai.

## DAFTAR PUSTAKA

- [1] W. Basuki, Wahyu Meliyana, 2016, "Pengendalian Gerak Robot Penghindar Halangan Menggunakan Citra dengan Kontrol PID", Politeknik Negeri Madiun, JEECAE Vol. 1, No. 1, Oktober 2016.
- [2] Dadang Bayu, W. Helmy, Harianto, 2014, "Pengendalian Motor DC *Brushless* dengan PID pada Robot Penghindar Halangan", STMIK STIKOM Surabaya, JCONES Vol. 3, No. 1.
- [3] Yulianto Andik, 2012, "Penerapan *Behavior-Based Control* dan *Fuzzy Logic Controller* pada Sistem Navigasi *Robot Soccer*", Universitas Internasional Batam, Civil and Electrical Engineering Journal, Vol. 7 No. 1, Juni 2012.
- [4] Costa, Cesar, 2001, "*Shape analysis and classification: theory and practice*", Boca Raton, CRC Press, 659 p. ISBN 0849334934.
- [5] W. Andre, L. Herlina, S. Yanolanda, 2016, "Analisa Perbandingan Metode *Filter Gaussian*, *Mean* dan *Median* Terhadap Reduksi *Noise*", Universitas Dehasen Bengkulu, Jurnal Media Infotama Vol. 12 No. 1, Februari 2016.
- [6] Amri Syaful, S. Hadi, S. Onny, 2014, "Perbandingan CF dan k-NN untuk Identifikasi Warna pada *Robot Soccer*", Jurnal EECCIS Vol. 8, No. 1, Juni 2014.
- [7] G. Wawan, Z. Agus, 2017, "Lokal *Fuzzy Thresholding* Berdasarkan Pengukuran *Fuzzy Similarity* Pada Interaktif Segmentasi Citra Panoramik Gigi", Institut Teknologi Sepuluh Nopember, Surabaya, Jurnal Infotel Vol.9 No.1 Februari 2017.
- [8] B. Widodo, P. Djoko, 2015, "Robot Vision – Teknik Membangun Robot Cerdas Masa Depan", Penerbit Andi Yogyakarta, Hal. 178.
- [9] M. Izzati, W. H. Lestari, 2013, "Aplikasi Anthropometri Telapak Kaki Berbasis Pengolahan Citra", Universitas Islam Indonesia, Yogyakarta, Seminar Nasional Medis (SNIMed) IV, p. 1.
- [10] Zhang Yao, W. Jianxue, 2016, "*K-nearest neighbors and a kernel density estimator for GEFCom2014 probabilistic wind power forecasting*", Xi'an Jiaotong University, Xi'an, China, International Journal of Forecasting, Vol. 32, Issue 3, September 2016.

## UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada Bapak Koko Joni dan Bapak A. Fiqhi Ibadillah selaku dosen