

Sistem Deteksi dan Penembak Target pada Robot Tank dengan Pengendali Nirkabel

Detection and Targets Shooter System on Robot Tank with Wireless Controller

Budi Herdiana, Yanuar Budi Pratama

Teknik Elektro, Fakultas Teknik Dan Ilmu Komputer

Universitas Komputer Indonesia

Jl. Dipati Ukur No 112, Bandung

Email : budi.herdiana@email.unikom.ac.id

Abstrak

Dalam penelitian ini memaparkan tentang sistem pendeteksi dan penembak target pada robot tank, target berupa pola dengan warna, sistem pendeteksian tersebut menggunakan kamera sebagai sensor yang akan mendeteksi target, ketika target telah terdeteksi maka selanjutnya akan diproses oleh pemrograman *OpenCV*, *OpenCV* akan memproses hasil deteksi kamera dengan menggunakan metode pengolahan citra, di mana metode pengolahan citra ini merupakan proses yang paling penting karena pada proses pengolahan citra akan diketahui atau dikenali warna serta pola dari pola yang menjadi dasar pendeteksian pada sistem ini, setelah *OpenCV* dapat mendeteksi target maka langkah selanjutnya adalah mengirim data hasil pendeteksian dan diproses Raspberry Pi, selanjutnya Raspberry Pi mengirimkan sinyal ke *servo controller* menggerakkan motor servo yang akan bergerak mengikuti di mana target berada. Pengujian sistem dilakukan dengan melakukan percobaan deteksi target dari setiap warna dan jarak antar target. Berdasarkan hasil pengujian sistem deteksi berdasarkan warna, sistem dapat mendeteksi semua target pada waktu siang dan malam. Sistem juga dapat mendeteksi warna utama jika digabungkan dengan warna lain, dari pengujian gabungan warna ini, sistem dapat mendeteksi seluruh warna utama pada saat siang dan malam. Sedangkan untuk jarak berbeda warna biru dapat mendeteksi target hingga jarak 70cm saat malam hari dan 140cm saat malam hari, warna hijau dapat mendeteksi target hingga jarak 80cm saat malam hari dan 130cm saat malam hari dan warna hijau dapat mendeteksi target hingga jarak 140cm saat malam hari dan 150cm saat malam hari. Sementara itu untuk respon *tracking*, tiap gerakan memiliki waktu rata-rata 2,56125 detik.

Kata kunci : Robot, Raspberry Pi, Arduino, OpenCV, Visi Komputer, Pengolahan Citra, Computer Vision

Absrtact

In this study explained the detection system and shooter target at the robotic tank, a target, a pattern with color detection system that uses the camera as a sensor that will detect the target, when the target has been detected, it will be processed by programming OpenCV, OpenCV will process the results of camera detection using the method of image processing, in which the method of image processing is the most important process for the image processing will be known or recognized by the color and pattern of a pattern that became the basis of the detection in this system, after OpenCV can detect a target then the next step is to send data the results of detection and processed Raspberry Pi, then Raspberry Pi sends a signal to servo controller to drive the servo motor will move to follow where the target is located. System testing is done by experimenting target detection of each color and the distance between the target. Based on test results based color detection system, the system can detect all targets by day and night. The system can also detect the main color when combined with another color, this color combination of testing, the system can detect all primary colors at day and night. As for the different distances blue can detect targets up to a distance of 70cm at night and 140cm at day, the green color can detect targets up to a distance of 80cm at night and 130cm at day and green colors can detect targets up to a distance of 140cm at night and 150cm at day. Meanwhile for tracking responses, each movement has an average time 2.56125 seconds.

Keywords: Raspberry Pi, Arduino, OpenCV, Computer Vision, Image Processing

I. PENDAHULUAN

Tracking objek bisa dipahami sebagai sebuah contoh kasus yang cukup istimewa dari sekian banyak permasalahan umum yang dipelajari dalam bidang computer vision. Perkembangan tingginya kemampuan dari suatu komputer saat ini dan meningkatnya kebutuhan akan analisa video dilakukan secara otomatis yang telah banyak sekali menghasilkan sesuatu yang hebat dan menarik dalam algoritma *tracking* objek. Penggunaan *tracking* objek ini merupakan sebuah permasalahan yang penting bagi sejumlah aplikasi yang mungkin bisa menguntungkan dalam penerapannya, seperti *traffic monitoring*, *automated surveillance*, sistem navigasi suatu kendaraan, militer, terutama dalam bidang robotika yaitu *mobile robot*, *robotic soccer* dan masih banyak lagi yang bisa diaplikasikan. [1]

Pada penelitian ini akan didesain kemampuan robot untuk melakukan deteksi dan menembak target yang dapat dikontrol secara nirkabel. Untuk membuat robot mampu melakukan itu dibutuhkan sensor dan penerapan metode kontrol yang khusus. Robot harus dapat melakukan pengolahan citra untuk mengenali objek apa yang harus dijadikan target sasaran tembak melalui pemrosesan data dengan bantuan kamera.

II. DASAR TEORI

A. Visi Komputer (*Computer Vision*)

Visi Komputer (*Computer Vision*) adalah salah satu ilmu pengetahuan yang mempelajari bagaimana penglihatan manusia dapat diterapkan pada komputer. Hal ini dimaksudkan agar komputer dapat mengenali lingkungan serta objek sekitarnya seperti manusia, Visi komputer terbagi menjadi 2 bidang ilmu lain yaitu

1. Pengolahan Citra (*image Processing*)

Pengolahan Citra adalah ilmu yang mempelajari bagaimana memperbaiki dan memodifikasi sebuah citra agar dapat menghasilkan citra yang baik dan mudah dikenali oleh manusia maupun oleh komputer.

2. Pengenalan pola (*Pattern recognition*)

Pengenalan pola adalah salah satu cabang dari *computer vision* yang mempelajari proses pengelompokan data numerik dan simbolik (termasuk citra) secara otomatis, yang berfungsi untuk mengidentifikasi objek dari suatu citra.

B. OpenCV

Open Source Computer Vision (OpenCV) adalah suatu library computer vision yang dibuat oleh para developer Intel Corporation yang bersifat Open Source, library tersebut merupakan kumpulan fungsi-fungsi C dan beberapa kelas C++ yang mengimplementasikan banyak algoritma-algoritma Image Processing dan Computer Vision. Salah satu penerapan program OpenCV yaitu dalam dunia robotika

Library OpenCV mempunyai lebih dari 2500 algoritma yang telah dioptimalkan dimana meliputi sebuah himpunan menyeluruh dari keduanya yaitu klasik dan seni beberapa algoritma computer vision dan machine learning. Algoritma-algoritma tersebut dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi obyek, mengklasifikasi tindakan manusia dalam video, mengikuti jejak perpindahan obyek, mengekstrak model-model 3D obyek, menghasilkan titik awan 3D dari kamera stereo, dan lain sebagainya.

OpenCV dapat diterapkan pada pemrograman C++, C, Python, Java dan MATLAB. OpenCV mendukung untuk sistem operasi Windows, Linux, Android dan Mac OS. [7]

C. SimpleCV

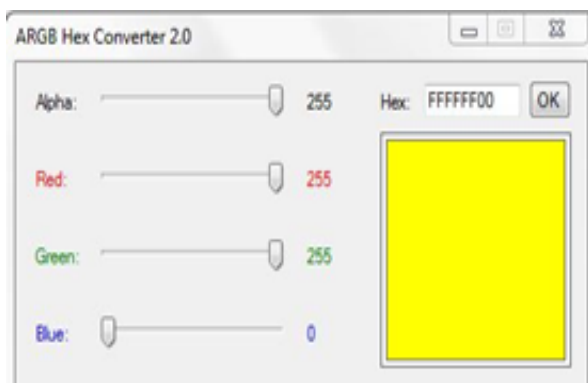
Simple Computer Vision (SimpleCV), merupakan framework bahasa Python yang mudah digunakan, serta membungkus library computer vision open source dan algoritma terkait untuk pemecahan masalah. Dengan SimpleCV, dapat mengakses ke library computer vision yang lebih tinggi, seperti OpenCV.

Beberapa fitur yang didukung oleh SimpleCV yaitu membaca gambar, konversi gambar ke RGB, konversi gambar ke HSL (Hue Saturation Lightness), konversi gambar ke HSV (Hue Saturation Value), konversi gambar ke Grayscale, membuat, menyalin, memperbesar, memperhalus, pencerminan gambar, edge detection, dan lain-lain. [8]

D. Model Warna RGB

Model warna RGB merupakan komponen yang terdiri dari warna dasar merah (Red), hijau (Green), dan biru (Blue) yang dapat dimodelkan atau diaplikasikan dengan cara mereduksi susunan warna untuk mendapatkan warna yang bervariasi.

Rentang nilai untuk warna RGB yaitu antara 0-255. Untuk dapat menghitung nilai warna dan menampilkannya dalam model RGB maka dapat dilakukan normalisasi dari ke tiga komponen warna. Dapat dilihat pada **Gambar 1** dari pencampuran warna diatas setiap warna memiliki rentang nilai dari 0-255, tingkat kehalusan dari sebuah citra dapat dilihat dari besar kecilnya satuan bit yang ditampilkan. Sebuah citra dengan ukuran 8 Bit memiliki jumlah warna sebesar $2^8 = 256$ warna. [14]



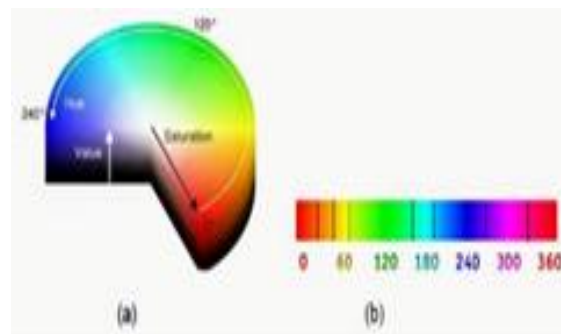
Gambar 1. Ilustrasi Pencampuran warna RGB

E. Model Warna *Hue Saturation Value* (HSV)

Model warna HSV merupakan kepanjangan dari Hue Saturation dan Value. Dari pengertian tersebut memiliki fungsi masing-masing yang berbeda. Model warna HSV ini dipandang lebih dekat dari model warna RGB dalam mendeskripsikan sensasi warna oleh mata manusia. Hue merupakan suatu ukuran panjang gelombang dari warna utama, hue mempunyai ukuran berkisar antara 0-255. 0 mewakili warna merah hingga melalui suatu spektrum kembali bernilai 256 atau kembali menjadi warna merah kembali. Saturation merupakan suatu proses untuk meningkatkan kecerahan warna yang di dasari dari jumlah hue murni pada warna akhir. Jika saturation bernilai nol maka warna akhir adalah bukan hue yang terbentuk hanya cahaya putih saja. Jika saturation bernilai 255 maka tidak ada pencahayaan tambahan pada warna akhir. Value merupakan sebuah ukuran seberapa besar kecerahan dari suatu warna. Apabila warna itu memiliki ukuran 100% maka akan terlihat sangat cerah, dan ketika 0% maka akan terlihat gelap.

Seperti pada **Gambar 2** dapat dijelaskan, untuk nilai hue dapat dilihat dari ukuran sudut yang melingkar hingga 360°, dari setiap perbedaan sudut hasil warnanya akan berbeda.

Saturation dapat dilihat dari ukuran horizontal ketika nilai warna itu di perkecil yang terlihat hanya warna putih. Value dapat dilihat dari ukuran vertikal nilai value akan berbanding dari nilai saturation yaitu ketika di perkecil akan tampak gelap. [15]



Gambar 2. Model warna HSV

F. Bahasa Python

Bahasa pemrograman ini dibuat oleh Guido van Rossum dari Amsterdam, Belanda. Pada awalnya, motivasi pembuatan bahasa pemrograman ini adalah untuk bahasa skrip tingkat tinggi pada sistem operasi terdistribusi Amoeba. Bahasa pemrograman ini menjadi umum digunakan untuk kalangan engineer seluruh dunia dalam pembuatan perangkat lunaknya, bahkan beberapa perusahaan menggunakan python sebagai pembuat perangkat lunak komersial.

Python merupakan bahasa pemrograman yang freeware atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan source codenya, debugger dan profiler, antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, GUI (antarmuka pengguna grafis), dan basis datanya. Python dapat digunakan dalam beberapa sistem operasi, seperti kebanyakan sistem UNIX, PCs (DOS, Windows, OS/2), Macintosh, dan lainnya. Pada kebanyakan sistem operasi linux, bahasa pemrograman ini menjadi standarisasi untuk disertakan dalam paket distribusinya.

G. Bahasa C

Pemrograman menggunakan Bahasa C adalah salah satu dari sekian banyak bahasa dasar untuk pemrograman mikrokontroler, misalnya bahasa *Assembly*, *Basic*, dan lain-lain. Bahasa pemrograman Bahasa C dikenal di seluruh dunia sebagai bahasa pemrograman handal, cepat,

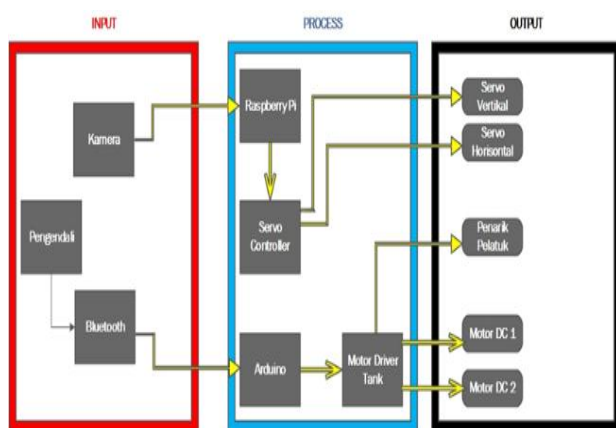
mudah, dan tergolong ke dalam bahasa pemrograman tingkat menengah. Bahasa C banyak digunakan untuk aplikasi mikrokontroler karena kemudahan dan kompatibel terhadap mikrokontroler jenis AVR.

III. PERANCANGAN

Pada bab ini akan membahas mengenai perancangan sistem. Perancangan ini akan terbagi beberapa bagian yaitu : blok diagram sistem, perancangan mekanik, perancangan hardware dan perancangan software.

A. Blok Diagram

Adapun gambaran dari blok diagram sistem yang akan dirancang adalah sebagai berikut.



Gambar 3. Blok Diagram Sistem

Secara umum cara kerja dari perancangan dari sistem deteksi yang akan dirancang adalah ketika adanya objek dan kamera mendeteksi objek tersebut, maka secara otomatis Raspberry Pi akan memproses data dari kamera dan Raspberry Pi juga melakukan komunikasi dengan *servo controller* melalui GPIO untuk memproses motor servo, sehingga motor servo dapat mengikuti gerak objek tersebut.

Fungsi dari setiap blok diagram :

1. Kamera mendeteksi target, lalu diproses melalui Raspberry Pi.
2. Raspberry Pi memberi perintah *Servo Controller* untuk menggerakkan servo Horizontal dan Vertikal, dimana Horizontal motor servo 1 gerak dari tengah ke kiri ke kanan dan sebaliknya, sedangkan Vertikal motor servo 2

bergerak dari tengah ke atas ke bawah dan sebaliknya.

3. Arduino mengoperasikan *motor driver*, dan menggerakkan motor 1 dan motor 2 yang bergerak maju dan mundur.
4. Arduino mengoperasikan motor penarik pelatuk senjata.
5. Pengontrol untuk mengendalikan gerak robot.

Ada dua komponen utama dalam perancangan :

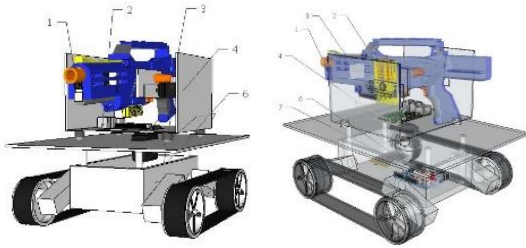
1. Raspberry Pi
 - Kontroler utama
 - Mengoperasikan kamera
 - Mengoperasikan motor servo
 - Komunikasi dengan Arduino
2. Arduino
 - Menjalankan motor DC
 - Mengontrol pelatuk senjata

B. Perancangan Mekanik

Perancangan mekanik ini, menggunakan platform Rover 5 Chassis, menggunakan kamera dibagian ujung laras senjata, Raspberry Pi pada bagian bawah senjata. Pada bagian dalam chassis terdapat 2 motor dc, Arduino, driver motor, servo controller, dan baterai.

Untuk tata letak komponen, servo horisontal diletakkan ditengah base plate atau plat dasar pada bagian bawah. Sedangkan untuk servo vertikal diletakkan ditengah pada bagian kiri turn plate atau plat putaran. Agar putaran seimbang dan memudahkan titik tumpu untuk berputar, pada bagian bawah plat putaran juga dipasang empat buah free wheel atau roda bebas pada bagian sudut plat putarannya.

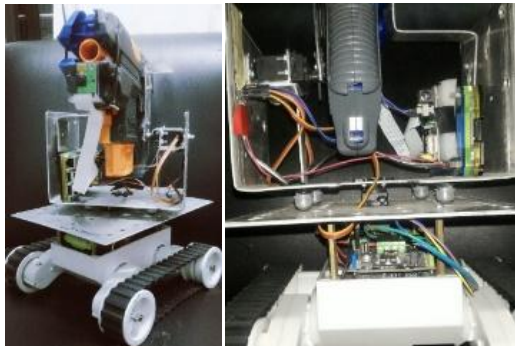
Pada tata letak senjata, dihubungkan dengan servo horisontal dengan dibantu satu buah bearing agar pergerakan senjata dapat seimbang dan halus. Sedangkan untuk tata letak komponen lain, seperti kamera diletakkan pada bagian depan senjata. Hal ini bertujuan agar pada saat pendeteksian target sasaran, senjata dapat menembak target tepat sasaran. Untuk Raspberry Pi, diletakkan pada bagian bawah senjata dan Arduino pada bagian kanan senjata. Komponen lain seperti driver motor, servo controller, motor dc, dan baterai diletakkan pada bagian bawah.



Gambar 4. Desain Sketsa Mekanik

Keterangan desain mekanik :

1. Pi Camera (Kamera utama penangkap objek)
2. Raspberry Pi
3. Servo vertikal
4. Servo horisontal
5. Arduino
6. Driver motor dc
7. Baterai



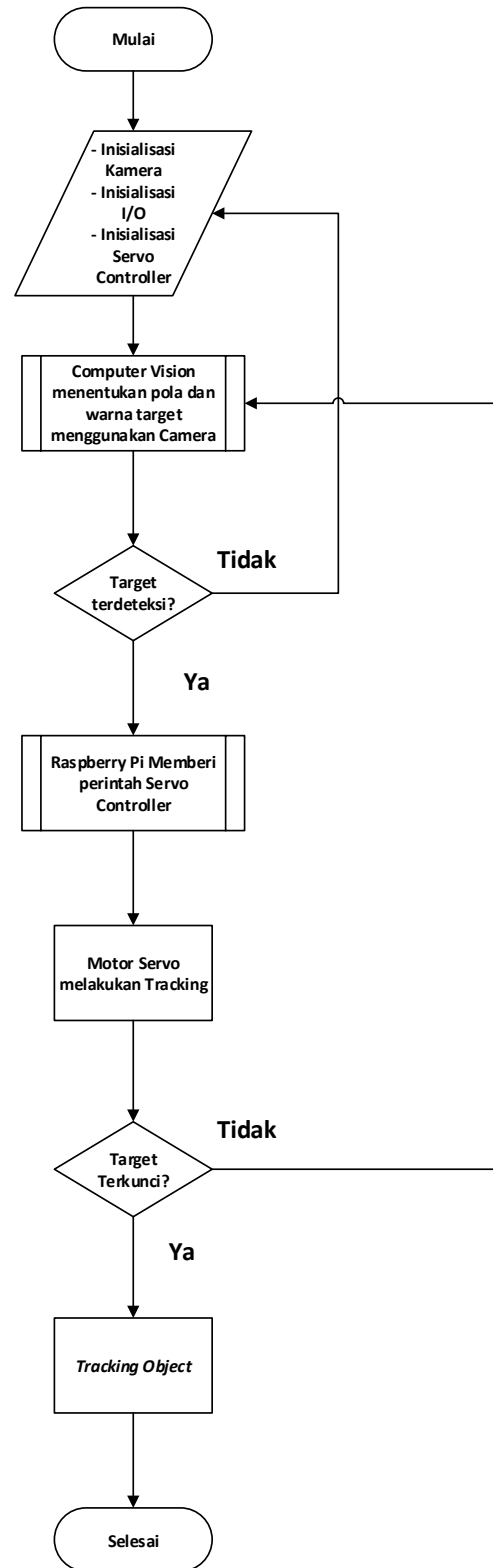
Gambar 5. Hasil Rancangan Mekanik

C. Flowchart Sistem Deteksi Target

Pada perancangan perangkat lunak ini bertujuan untuk menentukan setiap alur eksekusi dari perangkat sistem target yang dirancang. Setiap masukan yang diterima akan diatur oleh perangkat lunak yang selanjutnya akan diproses untuk menentukan eksekusi pada bagian keluaran. Berikut adalah alur kinerja (flowchart) dari sistem yang akan dirancang.

Pada flowchart diatas merupakan alur kerja sistem deteksi target menggunakan *OpenCV*, dimana penjelasannya sebagai berikut. Kamera sebagai pendeteksi target dan motor servo berfungsi sebagai penggerak kamera untuk melakukan *tracking*. Jika kamera mendeteksi ada suatu objek, maka *OpenCV* akan memroses untuk menentukan objek tersebut target yang jadi sasaran atau bukan. Setelah *OpenCV* mengidentifikasi objek menjadi target sasaran, selanjutnya, Raspberry Pi memberi perintah *Servo Controller* untuk menggerakkan motor servo agar

dapat melakukan *tracking* dan mengunci target. Setelah target terkunci, target dapat ditembak.



Gambar 6. Flowchart Sistem Deteksi Target

D. Flowchart Pergerakan Robot dan Pengoperasian Senjata

Pada perancangan perangkat lunak ini bertujuan untuk menentukan setiap alur eksekusi dari perangkat sistem target yang dirancang. Setiap masukan yang diterima akan diatur oleh perangkat lunak yang selanjutnya akan diproses untuk menentukan eksekusi pada bagian keluaran. Berikut adalah alur kinerja (flowchart) dari sistem yang akan dirancang.

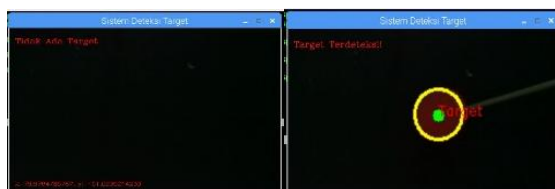
Pada flowchart diatas merupakan proses untuk mengendalikan robot dan mengoperasikan pelatuk senjata, dimana pada pengendalian robot menggunakan bluetooth sebagai komunikasi nirkabel. Sedangkan Arduino berperan sebagai pemroses, dan *motor driver* sebagai penggerak roda dan motor pelatuk senjata.

V. ANALISIS DAN PENGUJIAN

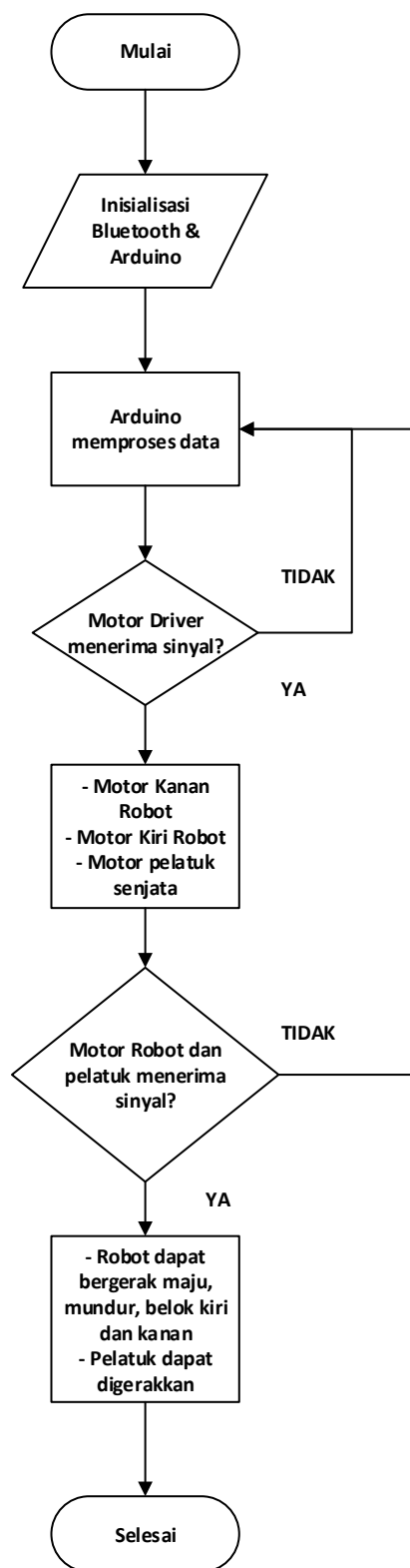
Pengujian Pengolahan citra ini bertujuan untuk mengetahui seberapa baik sistem saat mendeteksi target berupa lingkaran yang berwarna merah, hijau, dan biru. Pengujian Pengolahan citra dilakukan pada keadaan pencahayaan yang baik karena jika cahaya berubah redup atau terlalu terang akan mempengaruhi proses pendeteksian target terutama pada nilai Hue Saturation Value (HSV). Pengujian ini dilakukan dengan menggunakan kamera yang terhubung dengan Raspberry Pi dan terpasang pada robot.

A. Pengujian Kamera

Pengujian kamera merupakan awal untuk membuat pengembangan algoritma pengolahan citra untuk sistem deteksi target ini. Kamera telah dipasang pada mekanik robot dengan sedemikian rupa sehingga mendapatkan sudut yang sesuai saat melakukan object tracking.



Gambar 8. Tampilan Input Kamera Saat Ada dan Tidak Ada Target



Gambar 7. Flowchart Pergerakan Robot dan Pengoperasian Senjata

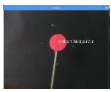
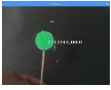

Dari **Gambar 8.** diatas, merupakan tampilan *frame* saat kamera aktif, dimana yang sebelah kiri merupakan input saat tidak ada target, dikiri atas tertulis Tidak Ada Target dan yang sebelah kanan

merupakan input saat ada target, dikiri atas tertulis Target Terdeteksi!. Sedangkan untuk indikasi target terdeteksi, ditandai lingkaran kuning di tepi objek yang terdeteksi. Resolusi *Frame* yang digunakan untuk pengujian ini adalah **Height = 480**, **Width = 320**.



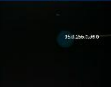
B. Pengambilan Nilai HSV

Proses pengujian berikut adalah pengambilan nilai Hue Saturation Value (HSV), karena pada saat melakukan object tracking yang diambil data warnanya adalah nilai HSV. Nilai HSV pada pengujian diambil pada dua kondisi waktu, yaitu siang dan malam.

Tabel 1. Nilai HSV saat siang hari

Warna	Proses Pengambilan Nilai HSV	Nilai HSV
Merah		H :176.0 S : 178 .0 V : 237.0
Hijau		H: 74.0 S : 204 .0 V : 194.0
Biru		H: 99.0 S : 230 .0 V : 196.0

Tabel 2. Nilai HSV saat malam hari

Warna	Proses Pengambilan Nilai HSV	Nilai HSV
Merah		H: 3.0 S : 215 .0 V : 76.0
Hijau		H: 63.0 S : 225 .0 V : 60.0
Biru		H: 95.0 S : 225 .0 V : 36.0

Dari perbandingan pada **Tabel 1** dan **Tabel 2**, memiliki perbandingan antara data HSV siang dan malam hari, di mana setiap warna memiliki selisih pada berikut ini :

Selisih nilai HSV = Data Siang – Data Malam

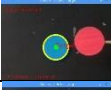

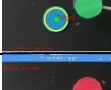
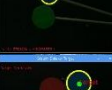


1. Merah : H = 173 , S = -37 , V = 161
2. Hijau : H = 11 , S = -21 , V = 134
3. Biru : H = 4 , S = 5 , V = 160

Selisih dari HSV di atas cukup untuk mempengaruhi hasil pengolahan citra, untuk itu jika melakukan *object tracking* berbeda waktu maka harus dilakukan pengambilan ulang nilai HSV.

C. Pengujian Deteksi Menggabungkan Dengan Warna Lain

Pengujian kali ini adalah percobaan menyatukan warna yang ingin dideteksi dengan warna lain. Hal ini bertujuan untuk memastikan warna utama dapat terdeteksi dengan baik tanpa terganggu dengan warna lain. Percobaan ini dilakukan dengan waktu berbeda, yaitu siang dan malam.

Tabel 3. Pengujian Dengan Warna Utama Biru

Warna	Proses Deteksi	
	Siang	Malam
Biru – Merah		
Biru – Hijau		
Biru – Merah – Hijau		


Tabel 4. Hasil Pengujian Dengan Warna Utama Biru

Warna Utama	Warna Lain	Keterangan		Keterangan
		Siang	Malam	
Biru	Merah	Terdeteksi	Terdeteksi	Berhasil
Biru	Hijau	Terdeteksi	Terdeteksi	Berhasil
Biru	Merah & Hijau	Terdeteksi	Terdeteksi	Berhasil

Hasil pengujian sistem pendeteksi warna utama biru digabungkan dengan warna lain, warna biru masih tetap terdeteksi dengan baik pada saat siang dan malam.

Tabel 5. Pengujian Dengan Warna Utama Hijau

Warna	Proses Deteksi	
	Siang	Malam
Hijau – Biru		


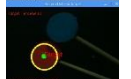


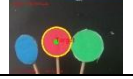

Hijau –Merah		
Hijau – Biru – Merah		

Tabel 6. Hasil Pengujian Dengan Warna Utama Hijau

Warna Utama	Warna Lain	Keterangan		Keterangan
		Siang	Malam	
Hijau	Biru	Terdeteksi	Terdeteksi	Berhasil
Hijau	Merah	Terdeteksi	Terdeteksi	Berhasil
Hijau	Merah & Biru	Terdeteksi	Terdeteksi	Berhasil

Hasil pengujian sistem pendeteksi warna utama hijau digabungkan dengan warna lain, warna biru masih tetap terdeteksi dengan baik pada saat siang dan malam.

Tabel 7. Pengujian Dengan Warna Utama Merah

Warna	Proses Deteksi	
	Siang	Malam
Merah – Biru		
Merah –Hijau		
Merah – Hijau – Biru		

Tabel 8. Hasil Pengujian Dengan Warna Utama Biru

Warna Utama	Warna Lain	Keterangan		Keterangan
		Siang	Malam	
Merah	Biru	Terdeteksi	Terdeteksi	Berhasil
Merah	Hijau	Terdeteksi	Terdeteksi	Berhasil
Merah	Hijau & Biru	Terdeteksi	Terdeteksi	Berhasil

Hasil pengujian sistem pendeteksi warna utama merah digabungkan dengan warna lain, warna biru masih tetap terdeteksi dengan baik pada saat siang dan malam
























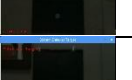











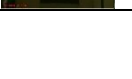
Dari pengujian deteksi warna di atas dapat dilihat bahwa seberapa baik dan akuratnya dari pendeteksian lingkaran yang dihasilkan. Pengambilan posisi target pada koordinat yang berbeda bertujuan untuk proses *tracking*.

D. Pengujian Deteksi Dengan Jarak Berbeda

Proses pengujian berikut adalah proses menedeteksi target berdasarkan jarak yang berbeda, dengan rentang jarak 30 – 200cm dan

kondisi waktu yang berbeda. Berikut hasil pengujian deteksi berdasarkan jarak.

Tabel 9. Pengujian Jarak Deteksi Dengan Warna Biru

Jarak (cm)	Proses Deteksi	
	Siang	Malam
30		
40		
50		
60		
70		
80		
90		
100		
110		
120		
130		
140		
150		
160		
170		
180		
190		
200		

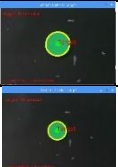
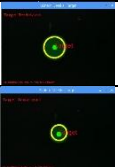
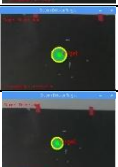
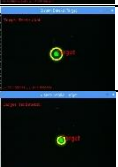




Tabel 10. Hasil Pengujian Pengujian Jarak Deteksi Dengan Warna Biru














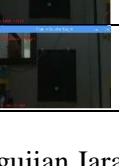



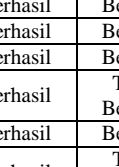

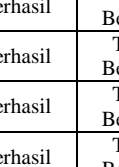

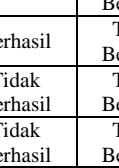




Jarak (cm)	Keterangan		Indikasi	
	Siang	Malam	Siang	Malam
30	Terdeteksi	Terdeteksi	Berhasil	Berhasil
40	Terdeteksi	Terdeteksi	Berhasil	Berhasil
50	Terdeteksi	Terdeteksi	Berhasil	Berhasil
60	Terdeteksi	Terdeteksi	Berhasil	Berhasil
70	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
80	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
90	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
100	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
110	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
120	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
130	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
140	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
150	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
160	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
170	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
180	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
190	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
200	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil

Analisis Hasil Pengujian :

Dari pengujian di atas, warna biru mulai tidak terbaca oleh sistem pada jarak 70cm ke atas saat malam hari, sedangkan pada siang hari warna biru tidak terdeteksi saat masuk jarak 140cm ke atas. Hal ini diakibatkan oleh perbedaan cahaya yang mempengaruhi nilai HSV.

Tabel 11. Pengujian Jarak Deteksi Dengan Warna Hijau

Jarak (cm)	Proses Deteksi	
	Siang	Malam
30		
40		
50		
60		
70		

80		
90		
100		
110		
120		
130		
140		
150		
160		
170		
180		
190		
200		

Tabel 12. Hasil Pengujian Pengujian Jarak Deteksi Dengan Warna Hijau

Jarak (cm)	Keterangan		Indikasi	
	Siang	Malam	Siang	Malam
30	Terdeteksi	Terdeteksi	Berhasil	Berhasil
40	Terdeteksi	Terdeteksi	Berhasil	Berhasil
50	Terdeteksi	Terdeteksi	Berhasil	Berhasil
60	Terdeteksi	Terdeteksi	Berhasil	Berhasil
70	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
80	Terdeteksi	Terdeteksi	Berhasil	Berhasil
90	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
100	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
110	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
120	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
130	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
140	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
150	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
160	Tidak	Tidak	Tidak	Tidak

	Terdeteksi	Terdeteksi	Berhasil	Berhasil
170	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
180	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
190	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
200	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil

Analisis Hasil Pengujian :

Dari pengujian di atas, warna hijau mulai tidak terbaca oleh sistem pada jarak 80cm ke atas saat malam hari, sedangkan pada siang hari warna biru tidak terdeteksi saat masuk jarak 130cm ke atas. Hal ini diakibatkan oleh perbedaan cahaya yang mempengaruhi nilai HSV.

Tabel 13. Pengujian Jarak Deteksi Dengan Warna Merah

Jarak (Cm)	Proses Deteksi	
	Siang	Malam
30		
40		
50		
60		
70		
80		
90		
100		
110		
120		
130		
140		
150		
160		

170		
180		
190		
200		

Tabel 14. Hasil Pengujian Pengujian Jarak Deteksi Dengan Warna Merah

Jarak (Cm)	Keterangan		Indikasi	
	Siang	Malam	Siang	Malam
30	Terdeteksi	Terdeteksi	Berhasil	Berhasil
40	Terdeteksi	Terdeteksi	Berhasil	Berhasil
50	Terdeteksi	Terdeteksi	Berhasil	Berhasil
60	Terdeteksi	Terdeteksi	Berhasil	Berhasil
70	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
80	Terdeteksi	Terdeteksi	Berhasil	Berhasil
90	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
100	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
110	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
120	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
130	Terdeteksi	Tidak Terdeteksi	Berhasil	Tidak Berhasil
140	Terdeteksi	Terdeteksi	Berhasil	Berhasil
150	Tidak Terdeteksi	Terdeteksi	Tidak Berhasil	Berhasil
160	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
170	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
180	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil
190	Tidak Terdeteksi	Terdeteksi	Berhasil	Tidak Berhasil
200	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Berhasil	Tidak Berhasil

Analisis Hasil Pengujian :

Dari pengujian di atas, warna hijau mulai tidak terbaca oleh sistem pada jarak 140cm ke atas saat malam hari, sedangkan pada siang hari warna biru tidak terdeteksi saat masuk jarak 150cm ke atas. Hal ini diakibatkan oleh perbedaan cahaya yang mempengaruhi nilai HSV.

E. Pengujian Respon Waktu Tracking Target

Pengujian respon waktu diperlukan untuk mengetahui seberapa cepat robot dapat bereaksi ketika sistem mendeteksi target sehingga dapat diketahui seberapa baik sistem pendeteksi dan komponen *hardware* pada robot beroperasi.

Berikut adalah data hasil pengujian yang dilakukan.

Tabel 15. Pengujian Respon Waktu

Posisi (ke Arah)	Respon waktu (detik)	Derajat Servo (Horizontal & Vertikal)
Atas	2,38	H = 90 V = 130
Bawah	2,80	H = 90 V = 70
Kanan	1,69	H = 140 V = 90
Kiri	1,80	H = 40 V = 90
Kanan Atas	2,84	H = 140 V = 130
Kanan Bawah	3,27	H = 140 V = 70
Kiri Atas	2,63	H = 40 V = 130
Kiri Bawah	3,08	H = 40 V = 70

$$\begin{aligned}
 Respon_{avg} &= \frac{\sum_1^N Respon Waktu}{N} \\
 &= \frac{2,38 + 2,80 + 1,69 + 1,80 + 2,84 + 3,27 + 2,63 + 3,08}{8} \\
 &= 2,56125 \text{ detik}
 \end{aligned}$$

Dari hasil pengujian respon waktu di atas, waktu yang di dihasilkan merupakan pengaruh dari mekanik dan komponen yang robot, sehingga semakin baik mekanik dan komponen, semakin baik pula pergerakan *tracking* dan waktu yang didapat bisa lebih cepat.

VI. KESIMPULAN DAN SARAN

Setelah melakukan perancangan, analisis dan pengujian sistem deteksi target ini dapat diambil beberapa kesimpulan, yaitu :

1. Dengan mengambil nilai HSV tiap objek, dan menggunakan metode erosi dan dilasi sistem sudah mampu mendeteksi target.
2. Raspberry pi dengan bahasa Python menggunakan OpenCV mampu untuk membuat sistem deteksi target.
3. Raspberry Pi menggunakan pin *General Purpose Input Output* (GPIO) dapat mengontrol motor servo untuk melakukan *tracking*.
4. Program Arduino dengan modu bluetooth mampu mengendalikan robot secara nirkabel.

Ada beberapa saran untuk pengembangan penelitian ini adalah sebagai berikut.

1. Menambah warna dan metode pendeteksian agar dapat mendeteksi beragam warna dan pola.

2. Kamera bisa diganti dengan jenis yang lebih baik, sehingga saat perubahan intensitas cahaya sistem masih dapat mendeteksi target.
3. Desain mekanik robot yang lebih baik sehingga robot data melakukan *tracking* dengan lebih cepat dan lebih baik.

DAFTAR PUSTAKA

- [1]. Prabowo, Hermawan., Soelistono Soegianto (2010). *Pelacakan Objek Menggunakan OpenCV*, Surabaya :Jurnal Ilmiah Universitas Airlangga Surabaya.
- [2]. Rakhman, Adi., Faisal Candrasyah & Fajar D. Sutera (2014). *Raspberry Pi - Mikrokontroler Mungil yang Serba Bisa*, Yogyakarta : Penerbit Andi
- [3]. Kadir, Abdul (2013). *Panduan Praktis mempelajari Aplikasi Mikrokontroler dan Pemrogramannya menggunakan Arduino*, Yogyakarta : Penerbit Andi
- [4]. Budiharto, Widodo., Djoko Purwanto (2015). *Robot Vision : Teknik Membangun Robot Cerdas Masa depan*, Yogyakarta : Penerbit Andi.
- [5]. Fajar Septian, Ridwan (2013). *Belajar Pemrograman Python Dasar*, Bandung : POSS – UPI
- [6]. Garcia Bueno, Gloria (2015). *Learning Image Processing with OpenCV*, Brimingham : PACKT Publishing
- [7]. Howse, Joseph (2013). *OpenCV Computer Vision with Python*, Brimingham: PACKT Publishing
- [8]. Pajankar, Ashwin. *Raspberry Pi Computer Vision Programming*, Brimingham : PACKT Publishing
- [9]. Adi Nugraha, Yogi . 2014. Implementasi Sistem Otomatis Pada Robot Kapal Berbasis Komputer Vision Untuk Kontes Kapal Cepat Tak Berawak Nasional, Bandung : UNIKOM
- [10]. Nuran Gani, Fadlan (2012). *Pulse Widht Modulation (PWM)*. <http://robotic-electric.blogspot.co.id/2012/11/pulse-width-modulation-pwm.html>, 7 November 2012
- [11]. Anonym (2011). *Representasi Model Warna RGB Menggunakan HSL dan HSV*. <https://mhstekomp.wordpress.com/2011/05/07/representasi-model-warna-rgb-menggunakan-hsl-dan-hsv/>, 5 Juli 2011
- [12]. Anonym. *Motor Servo*. <http://elektronika-dasar.web.id/motor-servo/motor-servo/>
- [13]. Anonym. *Teori Motor DC*. <http://elektronika-dasar.web.id/teori-motor-dc-dan-jenis-jenis-motor-dc/>